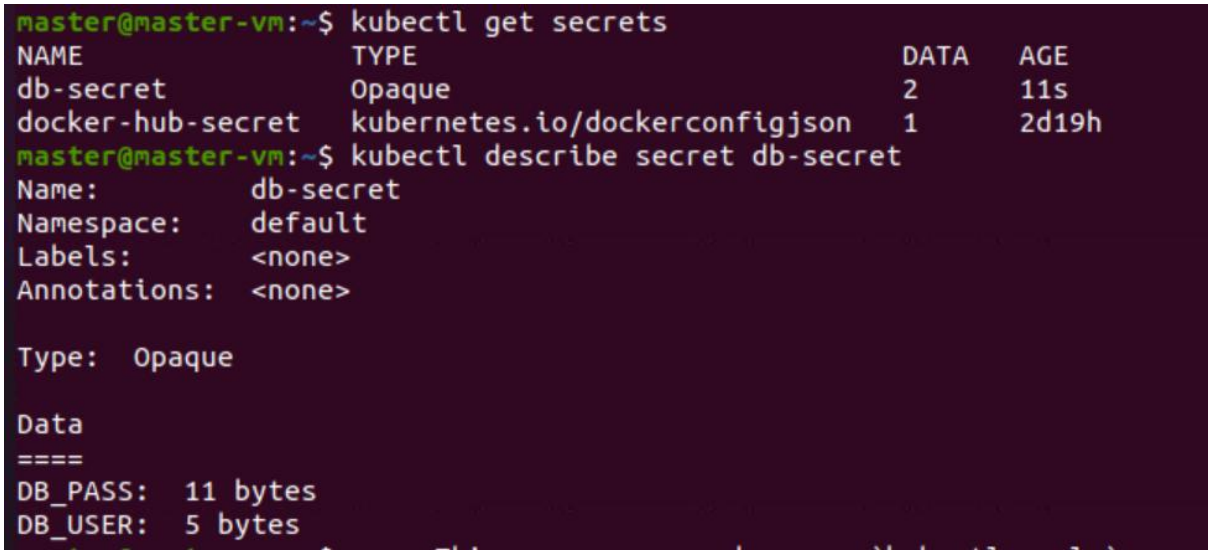
Exercise 5: Create and Use a Secret

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

Steps & Commands:

1. **Create a Secret for database credentials:**

2. **Verify the Secret:**



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

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

```
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
```

4. **Deploy the pod:**

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

5. **Check the pod and logs:**

```
master@master-vm:~$ kubectl get pods
NAME               READY   STATUS    RESTARTS   AGE
nginx-pod          1/1     Running   0          5m33s
nginx-secret-pod   1/1     Running   0          12s
master@master-vm:~$ kubectl logs nginx-secret-pod
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/03/14 04:51:05 [notice] 1#1: using the "epoll" event method
2025/03/14 04:51:05 [notice] 1#1: nginx/1.27.4
2025/03/14 04:51:05 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/03/14 04:51:05 [notice] 1#1: OS: Linux 5.15.0-134-generic
2025/03/14 04:51:05 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/03/14 04:51:05 [notice] 1#1: start worker processes
2025/03/14 04:51:05 [notice] 1#1: start worker process 29
2025/03/14 04:51:05 [notice] 1#1: start worker process 30
master@master-vm:~$ kubectl delete -f nginx-secret-pod.yaml
```

6. **Delete the pod and Secret:**

```
master@master-vm:~$ kubectl delete -f nginx-secret-pod.yaml
pod "nginx-secret-pod" deleted
master@master-vm:~$ kubectl delete secret db-secret
secret "db-secret" deleted
```

Exercise 6: Create and Expose a Service

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

Steps & Commands:

1. **Create a deployment:**

```
master@master-vm:~$ kubectl create deployment webapp --image=nginx
deployment.apps/webapp created
```

2. **Expose the deployment using a service:**

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

3. **Get service details:**

```
master@master-vm:~$ kubectl get svc webapp
NAME     TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
webapp   NodePort   10.96.214.204   <none>        80:32337/TCP   11s
master@master-vm:~$ minikube service webapp --url
```

4. **Access the service (Minikube users):**

```
master@master-vm:~$ minikube service webapp --url
http://192.168.49.2:32337
master@master-vm:~$ kubectl delete svc webapp
```

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
```