## Secret Box in K8s

Launching pods with sql image needs environment variables.

We store some sensitive passwords which need to be secured. For that, we use the concept of secret box in kubernetes.

Start the service and check its status.

```
C:\Users\lenovo>minikube start
* minikube v1.16.0 on Microsoft Windows 10 Home Single Language 10.0.19041 Build
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Updating the running virtualbox "minikube" VM ...
* Preparing Kubernetes v1.20.0 on Docker 20.10.0 ...
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, dashboard, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namesp

C:\Users\lenovo>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
timeToStop: Nonexistent


C:\Users\lenovo>kubectl get pods
NAME     READY    STATUS     RESTARTS     AGE
mydb     1/1      Running    0            22h
mywp1    1/1      Running    0            22h

C:\Users\lenovo>kubectl get svc
NAME          TYPE         CLUSTER-IP      EXTERNAL-IP    PORT(S)        AGE
kubernetes    ClusterIP    10.96.0.1       <none>         443/TCP        23h
mywp1         NodePort     10.103.91.103   <none>         80:31740/TCP   22h

C:\Users\lenovo>
```

If I describe the pod, the password of database is visible. I don't want this, so I'll use the concept of secret box. It is similar to the vault in Ansible.

```
C:\Users\lenovo>kubectl describe pods mydb
Name:          mydb
Namespace:     default
Priority:      0
Node:          minikube/192.168.99.101
Start Time:    Tue, 19 Jan 2021 22:20:21 +0530
Labels:        run=mydb
Annotations:   <none>
Status:        Running
IP:            172.17.0.5
IPs:
  IP:   172.17.0.5
Containers:
  mydb:
    Container ID:   docker://3055d9c176d5195113cc1c707293b8e706537aed27df1ab052e38aa8d9fc0714
    Image:          mysql:5.7
    Image ID:       docker-pullable://mysql@sha256:b3d1eff023f698cd433695c9506171f0d08a8f92a0c8063c
1a4d9db9a55808df
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Tue, 19 Jan 2021 22:20:23 +0530
    Ready:          True
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:   shreya
```

Default:

Using the command kubectl get secrets, we get the default secret token of k8s.

```
C:\Users\lenovo>kubectl get secrets
NAME                  TYPE                                  DATA   AGE
default-token-mzkvs   kubernetes.io/service-account-token   3      12d
```

Launch a sql pod using the yml file.

sqlpod - Notepad

File  Edit  Format  View  Help

```
apiVersion: v1
kind: Pod
metadata:
  name: "mydb"
  labels:
    app: mydb
spec:
  containers:
  - name: "mydb"
    image: "mysql:5.7"
  env:
  - name: MYSQL_ROOT_PASSWORD
    value: redhat
  - name: MYSQL_USER
    value: shreya
  - name: MYSQL_PASS
    value: shreya
  - name: MYSQL_DATABASE
    value: wpdb
```
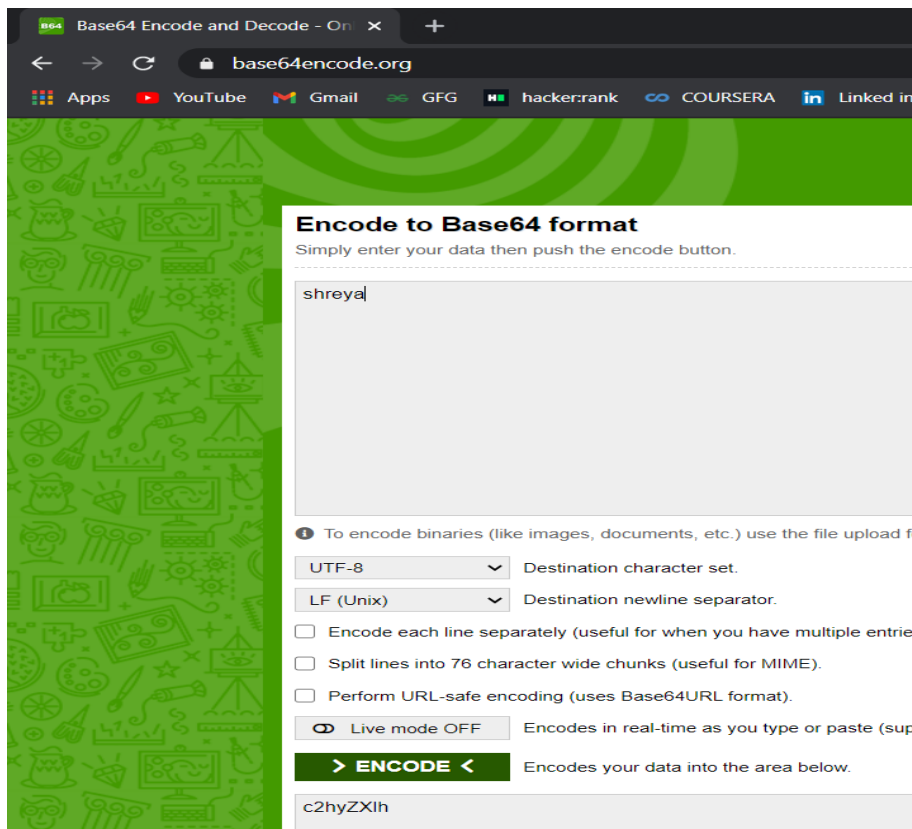
If we launch the pod using this yml file, it will get successfully launched. However, password is still available
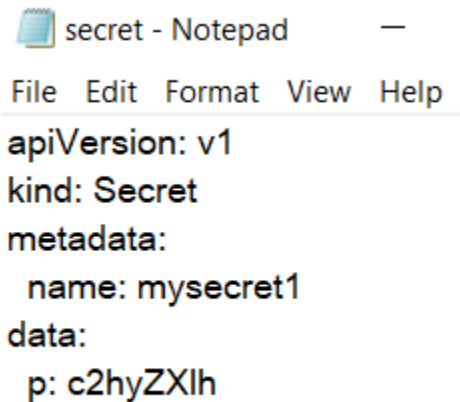
So, now we shall do encryption.

This is secret.yml file, which contains the password. However, before using it, we need to encrypt it. This is because, currently, the password is still visible.



Go to this website and encrypt the password.
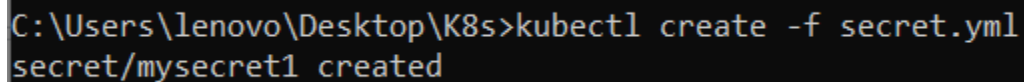
Get the encrypted password from this site and use it in the secret file.

```
secret - Notepad             —
File  Edit  Format  View  Help
apiVersion: v1
kind: Secret
metadata:
  name: mysecret1
data:
  p: c2hyZXlh
```

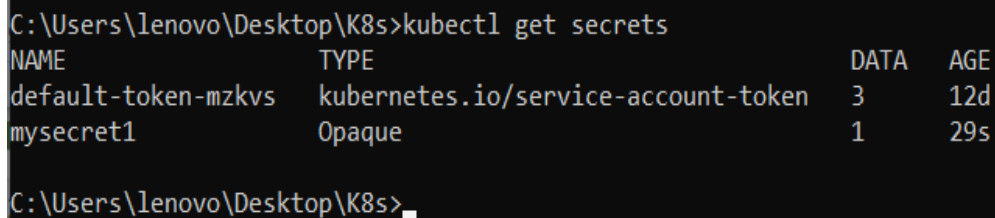Now we launch the secret file and check if it has been launched successfully.

kubectl create –f secret.yml

```
C:\Users\lenovo\Desktop\K8s>kubectl create -f secret.yml
secret/mysecret1 created
```

kubectl get secrets

```
C:\Users\lenovo\Desktop\K8s>kubectl get secrets
NAME                  TYPE                                DATA   AGE
default-token-mzkvs   kubernetes.io/service-account-token  3      12d
mysecret1             Opaque                               1      29s

C:\Users\lenovo\Desktop\K8s>
```

Now, apply the contents of the secret.yml file in the yml file which intends to launch the pod.

sqlpod - Notepad

File  Edit  Format  View  Help

```
apiVersion: v1
kind: Pod
metadata:
  name: "mydb"
  labels:
    app: mydb

spec:
  containers:
  - name: "mydb1"
    image: "mysql:5.7"
    env:
      - name: MYSQL_ROOT_PASSWORD
        valueFrom:
          secretKeyRef:
            name: mysecret1
            key: p
      - name: MYSQL_DATABASE
        value: wpdb
      - name: MYSQL_USER
        value: shreya
      - name: MYSQL_PASSWORD
        valueFrom:
          secretKeyRef:
            name: mysecret1
            key: p
```

Launch the pod using yml file.

```
C:\Users\lenovo\Desktop\K8s>kubectl create -f sqlpod.yml
pod/mydb created

C:\Users\lenovo\Desktop\K8s>_
```

Now, if we describe the pod, the password is not clearly visible.

```
Select Command Prompt
C:\Users\lenovo\Desktop\K8s>kubectl describe pods
Name:           mydb
Namespace:      default
Priority:       0
Node:           minikube/192.168.99.101
Start Time:     Thu, 21 Jan 2021 20:13:10 +0530
Labels:         app=mydb
Annotations:    <none>
Status:         Running
IP:             172.17.0.5
IPs:
  IP:  172.17.0.5
Containers:
  mydb1:
    Container ID:   docker://3964441b6a86b44fdceb3042810a64bc94f3c69e6b27d6d0f7850dc135ce5
    Image:          mysql:5.7
    Image ID:       docker-pullable://mysql@sha256:b3d1eff023f698cd433695c9506171f0d08a8f9
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 21 Jan 2021 20:13:12 +0530
    Ready:          True
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:  <set to the key 'p' in secret 'mysecret1'>  Optional: false
      MYSQL_DATABASE:       wpdb
      MYSQL_USER:           shreya
      MYSQL_PASSWORD:       <set to the key 'p' in secret 'mysecret1'>  Optional: false
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-mzkvs (ro)
Conditions:
  Type            Status
  Initialized     True
  Ready           True
```