echo -n 'admin' > ./username.txt

echo -n 'mypwd1122&&88' > ./password.txt

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

kubectl get secrets

kubectl describe secrets/mysecret

kubectl get secret mysecret -o yaml

echo 'bXlwd2QxMTIyJiY4OA==' | base64 --decode

echo 'YWRtaW4=' | base64 --decode

~~~~

kubectl create secret generic mysecretlitral --from-literal=username=mylitraluser --from-literal=password='!@345^'

kubectl get secret mysecretlitral -o yaml

echo 'IUAzNDVe' | base64 --decode

echo 'bXlsaXRyYWx1c2Vy' | base64 --decode

~~~~~

Createing manually

echo -n 'admin' | base64

echo -n 'mypwd1122&&88' | base64

mysecretmanual.yaml

apiVersion: v1

kind: Secret

metadata:

name: mysecret

type: Opaque

data:

username: YWRtaW4=

password: bXlwd2QxMTIyJiY4OA==

kubectl create -f mysecretmanual.yaml

kubectl get secret mysecret -o yaml

~~~~~

Using it in Pod

usesecretinpod.yaml

apiVersion: v1

kind: Pod

metadata:

name: mypod

spec:

containers:

- name: mypod

image: redis

volumeMounts:

- name: foo

mountPath: "/etc/secret"

readOnly: true

volumes:

- name: foo

secret:

secretName: mysecret

kubectl apply -f usesecretinpod.yaml

~~~~~~~~~~~~`

kubectl exec -it mypod /bin/bash

ls /etc/secret/

cat

~~~~~

secretasenv.yaml

apiVersion: v1

kind: Pod

metadata:

name: secret-env-pod

spec:

containers:

- name: mycontainer

image: redis

env:

- name: SECRET\_USERNAME

valueFrom:

secretKeyRef:

name: mysecret

key: username

- name: SECRET\_PASSWORD

valueFrom:

secretKeyRef:

name: mysecret

key: password

restartPolicy: Never

kubectl apply -f secretasenv.yaml

~~~~~~`

kubectl exec -it secret-env-pod -- /bin/bash

inside container

echo $SECRET\_USERNAME

echo $SECRET\_PASSWORD

kubectl delete secret mysecret

kubectl delete secret mysecretlitral