# Secret Usage

Try the following case:

1. Consume Secret as a *folder*. Use 600 as default permissions.
2. Run Pod that uses the Secret with the following command:

while true; do cat /etc/path/to/secret; sleep 5; done

1. Attach using kubectl exec to the Pod and check how Secret is represented on a disk.
2. Update Secret. Wait about 2 minutes. Output of the Pod should print the new value.

Try the following case:

1. Consume Secret as an *environment variables*.
2. Run Pod that uses the Secret and run the command:

printenv

1. Get output from the Pod. You should see the secret decoded value.
2. Update Secret.
3. Restart Pod.
4. Get output from the Pod again. You should see updated value.

## Solution

### Case A

Try the following case:

1. Consume Secret as a *folder*. Use 600 as default permissions.

$ mkdir tmp-secret  
  
$ cat <<"EOF" > tmp-secret/users  
login1=pass1  
login2=pass2  
EOF  
  
$ echo a1s2d3f4 > tmp-secret/admin.credentials  
  
$ kubectl create secret generic mysecret --from-file tmp-secret  
secret/mysecret created  
  
$ kubectl describe secret mysecret  
Name: mysecret  
Namespace: msuslov  
Labels: <none>  
Annotations: <none>  
  
Type: Opaque  
  
Data  
====  
admin.credentials: 9 bytes  
users: 26 bytes

1. Run Pod that uses the Secret with the following command:

$ cat <<"EOF" | kubectl create -f -  
apiVersion: v1  
kind: Pod  
metadata:  
 name: demo-secret  
spec:  
 containers:  
 - name: busybox  
 image: busybox  
 imagePullPolicy: IfNotPresent  
 command:  
 - /bin/sh  
 - -c  
 - |  
 while true; do  
 echo $(date +'%Y.%m.%d %H:%M:%S')  
 cat /etc/secret/admin.credentials  
 sleep 5  
 done  
 volumeMounts:  
 - name: volume-mysecret  
 mountPath: /etc/secret  
 readOnly: true  
 volumes:  
 - name: volume-mysecret  
 secret:  
 secretName: mysecret  
 defaultMode: 0400  
 restartPolicy: Never  
EOF  
  
$ kubectl get pod demo-secret  
NAME READY STATUS RESTARTS AGE  
demo-secret 1/1 Running 0 11s

1. Attach using kubectl exec to the Pod and check how Secret is represented on a disk.

$ kubectl exec demo-secret -- ls /etc/secret  
admin.credentials  
users  
  
$ kubectl exec demo-secret -- cat /etc/secret/admin.credentials  
a1s2d3f4  
  
$ kubectl logs demo-secret  
2021.06.14 04:42:33  
a1s2d3f4  
2021.06.14 04:42:38  
a1s2d3f4  
...

1. Update Secret. Wait about 2 minutes. Output of the Pod should print the new value.

$ kubectl create secret generic mysecret --from-literal=admin.credentials=AABBCCDD -o yaml --dry-run=client | kubectl replace -f -  
secret/mysecret replaced  
  
$ kubectl describe secret mysecret  
Name: mysecret  
Namespace: msuslov  
Labels: <none>  
Annotations: <none>  
  
Type: Opaque  
  
Data  
====  
admin.credentials: 8 bytes  
  
$ kubectl logs demo-secret -f  
2021.06.14 04:45:48  
a1s2d3f4  
...  
2021.06.14 04:46:23  
AABBCCDD

Clean up:

$ kubectl delete pod/demo-secret  
pod "demo-secret" deleted

### Case B

Try the following case:

1. Consume Secret as an *environment variables*.

$ cat <<"EOF" > tmp-secret/pod.yaml  
apiVersion: v1  
kind: Pod  
metadata:  
 name: demo-secret  
spec:  
 containers:  
 - name: busybox  
 image: busybox  
 imagePullPolicy: IfNotPresent  
 command: ["/bin/sh", "-c", "printenv"]  
 envFrom:  
 - secretRef:  
 name: mysecret  
 restartPolicy: Never  
EOF

1. Run Pod that uses the Secret and run the command printenv:

$ kubectl create -f tmp-secret/pod.yaml  
pod/demo-secret created  
  
$ kubectl get pod demo-secret  
NAME READY STATUS RESTARTS AGE  
demo-secret 1/1 Running 0 11s  
  
$ kubectl exec demo-secret -- printenv  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin  
HOSTNAME=demo-secret  
admin.credentials=AABBCCDD  
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  
KUBERNETES\_SERVICE\_PORT\_HTTPS=443  
HOME=/root

1. Get output from the Pod. You should see the secret decoded value.

$ kubectl logs demo-secret | grep admin  
admin.credentials=AABBCCDD

1. Update Secret.

$ kubectl create secret generic mysecret --from-literal=admin.credentials=233-322-233-322 -o yaml --dry-run=client | kubectl replace -f -  
secret/mysecret replaced

1. Restart Pod.

$ kubectl delete pod demo-secret  
pod "demo-secret" deleted  
  
$ kubectl create -f tmp-secret/pod.yaml  
pod/demo-secret created

1. Get output from the Pod again. You should see updated value.

$ kubectl logs demo-secret | grep admin  
admin.credentials=233-322-233-322

Clean up:

$ kubectl delete secret/mysecret pod/demo-secret  
secret "mysecret" deleted  
pod "demo-secret" deleted  
  
$ rm -rf tmp-secret