05Nov2022
Day**21**
Kubernetes Secret

Secrets
https://kubernetes.io/docs/concepts/configuration/secret/
A Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key.
Such information might otherwise be put in a Pod specification or in a container image.
Using a **Secret** means that you don't need to include confidential data in your application code.

Define variable in K8s workloads through 'env'

```
# kubectl run busybox --image busybox -o yaml --dry-run=client >busybox.yaml
# vim busybox.yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: busybox
  name: busybox
spec:
  containers:
  - image: busybox
    name: busyboxcnt
    command: ['sleep','4800']
    env:
    - name: username
      value: user1
    - name: password
      value: redhat123
:wq!
# kubectl create -f busybox.yaml
# kubectl get pods
# kubectl exec -it busybox -- /bin/sh
/ # env
username=user1
password=redhat123
# exit
```

Define variable in K8s workloads through secret
create secret
--from-literal

```
# kubectl create secret generic secmysql --from-literal password=test123 -o yaml --dry-run=client >mysqlsec.yaml
# cat secmysql.yaml
apiVersion: v1
stringData:
  password: dGVzdDEyMzQK
kind: Secret
metadata:
  creationTimestamp: null
  name: secmysql
# kubectl create -f secmysql.yaml
# kubectl get secret
# kubectl get  secrets secmysql -o yaml
```

create K8s workloads

```
# kubectl run mysql --image mysql -o yaml --dry-run=client >mysql.yaml
# vim mysql.yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: mysql
  name: mysql
spec:
  containers:
  - image: mysql
    name: mysql
    env:
      - name: MYSQL_ROOT_PASSWORD        ->this will define in MySql pod
        valueFrom:
          secretKeyRef:
            name: secmysql                ->secret name
            key: password                 ->secret parameters
:wq!
# kubectl apply -f mysql.yaml
# kubectl get pods
```

Mohammad Ali Naghval
ali@prodevans.com

```
# kubectl exec -it mysql -- /bin/sh
sh-4.4# env
MYSQL_ROOT_PASSWORD=dGVzdDEyMzQK
sh-4.4# mysql -u root -p
Enter password: dGVzdDEyMzQK
mysql> quit
sh-4.4# exit
root@master1:~#
```

Edit secret parameters value

```
# kubectl exec -it mysql -- env
MYSQL_ROOT_PASSWORD=dGVzdDEyMzQK
#
# echo "coss@2022" | base64
Y29zc0AyMDIyCg==
# kubectl edit secrets secmysql
data:
  password: Y29zc0AyMDIyCg==
:wq!
# kubectl get pods
# kubectl exec -it mysql -- env
MYSQL_ROOT_PASSWORD=coss@2022
or
# kubectl exec -it mysql -- env
MYSQL_ROOT_PASSWORD=aGF0cmVkMjAyMgo=
# kubectl exec -it mysql -- /bin/sh
sh-4.4# mysql -u root -p
Enter password: aGF0cmVkMjAyMgo=  or coss@2022
mysql>
mysql> exit
#

--from-file
# vim abc.txt
password=India321
:wq!
# kubectl create secret generic sec2 --from-file=/root/abc.txt -o yaml --dry-run=client >sec2.yaml
# cat sec2.yaml
apiVersion: v1
data:
  abc.txt: cGFzc3dvcmQ9SW5kaWEzMjEK
kind: Secret
metadata:
  creationTimestamp: null
  name: sec2
# kubectl apply -f sec2.yaml
# kubectl get secret
# vim mysql2.yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    run: mysql
  name: mysql
spec:
  containers:
  - image: mysql
    name: mysql
    envFrom:
      - secretRef:
          name: sec2     ->secretname
:wq!
# kubectl create -f mysql2.yaml --dry-run=client
# kubectl create -f mysql2.yaml
# kubectl get pods
```

```
# kubectl get nodes
NAME                STATUS     ROLES          AGE    VERSION
node1.example.com   NotReady   <none>         146d   v1.24.1
# ping node1.example.com
64 bytes from node1.example.com (192.168.29.105): icmp_seq=1 ttl=64 time=0.341 ms
# nslookup node1.example.com
Server:         127.0.0.53
Address:        127.0.0.53#53
Non-authoritative answer:
Name:   node1.example.com
Address: 192.168.29.105
# ssh node1.example.com
# journalctl -xe
Nov 05 15:41:44 node1.example.com systemd[1]: Stopped kubelet: The Kubernetes Node Agent.
# systemctl start kubelet.service

# kubectl describe <pod/deployment/secret/jobs>
# kubectl logs pod/<podname>
```