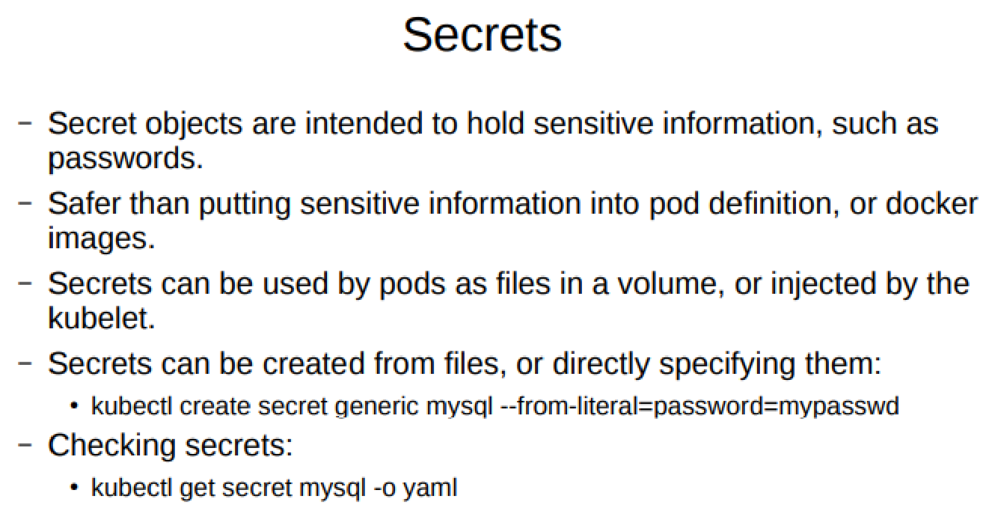
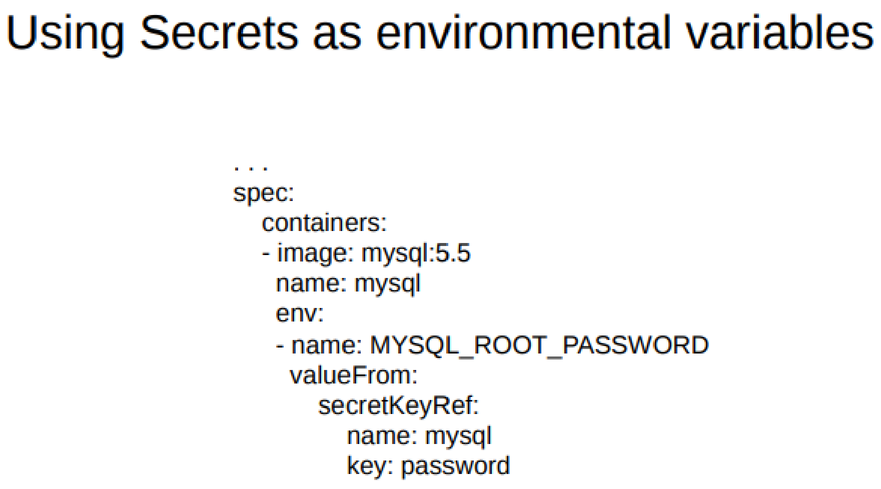
*Lab - kubernetes configmap and secrets

Wednesday, December 8, 2021 09:05 PM



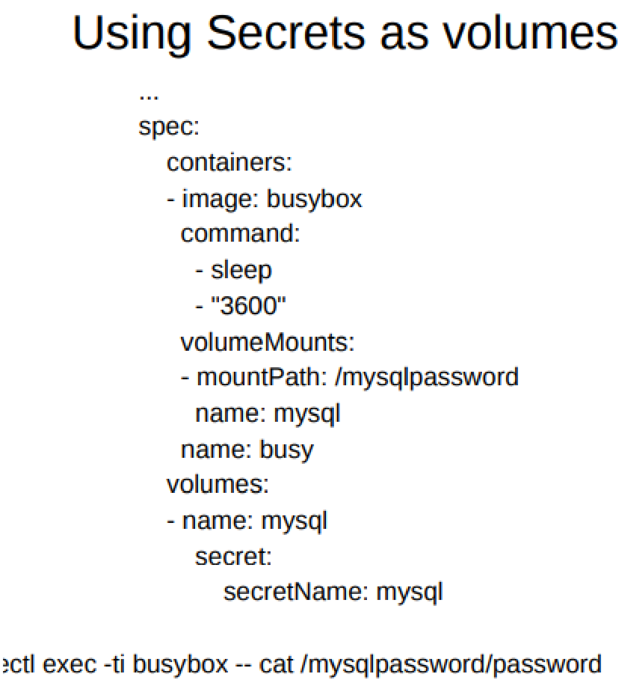
Secrets

- Secret objects are intended to hold sensitive information, such as passwords.
- Safer than putting sensitive information into pod definition, or docker images.
- Secrets can be used by pods as files in a volume, or injected by the kubelet.
- Secrets can be created from files, or directly specifying them:
  - kubectl create secret generic mysql --from-literal=password=mypasswd
- Checking secrets:
  - kubectl get secret mysql -o yaml



Using Secrets as environmental variables

```
. . .
spec:
    containers:
    - image: mysql:5.5
      name: mysql
      env:
      - name: MYSQL_ROOT_PASSWORD
        valueFrom:
            secretKeyRef:
                name: mysql
                key: password
```

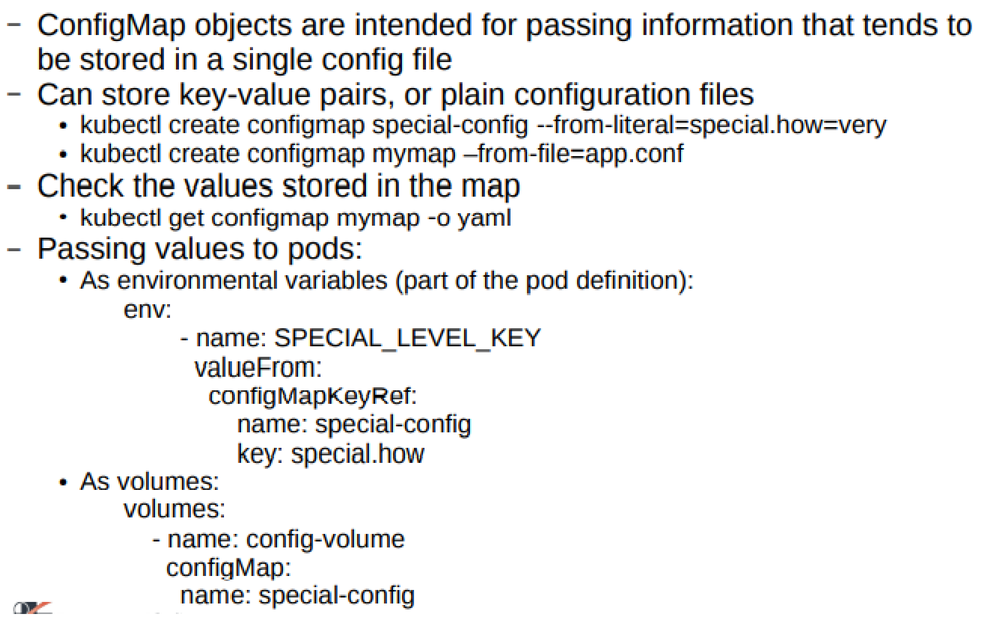


Using Secrets as volumes

```
...
spec:
    containers:
    - image: busybox
      command:
        - sleep
        - "3600"
      volumeMounts:
      - mountPath: /mysqlpassword
        name: mysql
      name: busy
    volumes:
    - name: mysql
        secret:
            secretName: mysql
```

```
ectl exec -ti busybox -- cat /mysqlpassword/password
```

ConfigMaps



- ConfigMap objects are intended for passing information that tends to be stored in a single config file
- Can store key-value pairs, or plain configuration files
  - kubectl create configmap special-config --from-literal=special.how=very
  - kubectl create configmap mymap –from-file=app.conf
- Check the values stored in the map
  - kubectl get configmap mymap -o yaml
- Passing values to pods:
  - As environmental variables (part of the pod definition):

```
env:
    - name: SPECIAL_LEVEL_KEY
      valueFrom:
        configMapKeyRef:
            name: special-config
            key: special.how
```

  - As volumes:

```
volumes:
    - name: config-volume
      configMap:
        name: special-config
```

Configmap

```
=======================================================================
 apiVersion: v1
kind: Pod
metadata:
  name: myvolconfig
spec:
  containers:
  - name: c1
    image: centos
    command: ["/bin/bash", "-c", "while true; do echo Hello world; sleep 5 ; done"]
    volumeMounts:
      - name: testconfigmap
        mountPath: "/tmp/config"   # the config files will be mounted as ReadOnly by default here
  volumes:
  - name: testconfigmap
    configMap:
       name: mymap   # this should match the config map name created in the first step
       items:
       - key: sample.conf
         path: sample.conf

=================================================================================

apiVersion: v1
kind: Pod
metadata:
  name: myenvconfig
spec:
  containers:
  - name: c1
    image: centos
    command: ["/bin/bash", "-c", "while true; do echo Hello World; sleep 5 ; done"]
    env:
    - name: MYENV         # env name in which value of the key is stored
      valueFrom:
        configMapKeyRef:
          name: mymap      # name of the config created
          key: sample.conf
===========================================================
 apiVersion: v1
```

```
kind: Pod
metadata:
  name: myvolsecret
spec:
  containers:
  - name: c1
    image: centos
    command: ["/bin/bash", "-c", "while true; do echo Hello World; sleep 5 ; done"]
    volumeMounts:
      - name: testsecret
        mountPath: "/tmp/mysecrets"   # the secret files will be mounted as ReadOnly by default here
  volumes:
  - name: testsecret
    secret:
       secretName: mysecret
```

==