Mansi Shah
19526
Week 11

7.5. USING SECRETS TO PASS SENSITIVE DATA TO CONTAINERS

Introducing the default token Secret

```
mansishah@macbookpro ~ % kubectl get secrets
NAME                  TYPE                                  DATA   AGE
default-token-lp9tk   kubernetes.io/service-account-token   3      3d6h
mansishah@macbookpro ~ % kubectl describe secrets
Name:         default-token-lp9tk
Namespace:    default
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: default
              kubernetes.io/service-account.uid: c1fa3898-51c1-48e3-a3ac-09938a7f2887

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1066 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IjdPT3ppSUtpNFUyZ3NTQWRlVlhDWjFMbzZTbnA4RF9SWFU5S1NIVzV6QlEifQ.eyJpc
iJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6ImRlZmF1bHQtdG9rZW4tbHA5dGsiLCJrdWJlc
2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI6ImMxZmEzODk4LTUxYzEtNDhlMy1hM2FjLTA5OTM4YTdmMjg4NyIsInN1YiI6I
XDaaz7QT2FcwoltO6FvjyQrZ0nmZfXlJxbXfS3aH8T_8t2BqwfvDOhil_i5R9sgtqFMzdpxyYLYSd-JCDDByEVoivrpQYH73sQUD7hUsYm4-
N-roNUF1EVLZ84xeXqBBEdQV7LHO1auy9ATXVqbTEdPGksPo7fua_9gQOxz5c-KbIHUBpSWyORBWh-XVOXMtaLu25sPTuUmw
```

```
mansishah@macbookpro ~ % kubectl get po
NAME                       READY   STATUS    RESTARTS   AGE
fortune-configmap-volume   2/2     Running   2          26h
mansishah@macbookpro ~ % kubectl exec fortune-configmap-volume ls /var/run/secrets/kubernetes.io/serviceaccount/
Defaulting container name to html-generator.
Use 'kubectl describe pod/fortune-configmap-volume -n default' to see all of the containers in this pod.
ca.crt
namespace
token
mansishah@macbookpro ~ %
```

Creating a Secret

```
mansishah@macbookpro ~ % openssl genrsa -out https.key 2048
Generating RSA private key, 2048 bit long modulus
.........+++
.....................+++
e is 65537 (0x10001)
mansishah@macbookpro ~ % openssl req -new -x509 -key https.key -out https.cert -days 3650 -subj /CN=www.kubia-example.com
mansishah@macbookpro ~ % echo bar > foo
mansishah@macbookpro ~ % kubectl create secret generic fortune-https --from-file=https.key --from-file=https.cert --from-file=foo
secret/fortune-https created
mansishah@macbookpro ~ % kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-lp9tk   kubernetes.io/service-account-token   3      3d6h
fortune-https         Opaque                                3      23s
```

Comparing ConfigMaps and Secrets

```
mansishah@macbookpro ~ % kubectl get secret fortune-https -o yaml
apiVersion: v1
data:
  foo: YmFyCg==
  https.cert: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN2RENDQWFRQ0NRQ3RCd25UUktQNTFEQU5CZ2txaGtpRzl3
hjTk16QXdNekl3TVRJMApNelUzV2pBZ01SNHdIQVlEVlFRRERCVjNkM2N1YTNWaWFXRXRaWGhoYlhCc1pTNWpiMjB3Z2dFaU1BMEdE
Z2c0ZPSzJlQjdhSnMxaERsQlc3dnpnTmxFZXJQdGlpYk92MzBGbDc3bkFuV0g1TWo2V1NNQlRNOQp3OGkwdDBjMlR1UTM1U1lOQitN
hnbkx4NVZKR1NKRzF6VkdxZkRTT0pzTDRadUk1L3lVY1oKbXBuVi9MajZmeXpsSVNyUlc2aE14bmhGR2l6REc4bmtwN0RwWC9aM2h1
dNQkFBRXdEUVlKCktvWklodmNOQVFFTEJRQURnZ0VCQUxpK2d2Wm9DenJlUTBQR3JLL0NDQ2RpVEoxSEp4cll3d0dTTkpWdjdBUksk
dBNENsSkR1MDl4Yjh6dFFCaW9reHcxemxJdXRYcm9NYTM0engraVJoUTdaQ0FDRkNtRExqCjd3R1NQc3ZIRjBvQnJlU1BBaEVQeFNv
VXMEIxWVE2K2ViUjROR0tWTXhTcmlBMUN6eUxTOAprU2hKS0tjMVJNbVd0UzJ4c2xiUjUwV3grY0xOVjVZVWZZYmtpTmFaQTlVPQot
  https.key: LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb2dJQkFBS0NBUUVBeElrUlhMZ3gxWmU5VnhiWnZk
VelBjUEl0TGRITms3a04rVW0KRFFmaXluYjY3OURKTkZONGhIOUpjWitDUERtV3RodUlERllDeTVPV1JBQUVPdEV0TDBSblU4YUo0
UTVo0UlJvc3d4dko1S2V3NlYvMmQ0WHRtU1JaYlFaQUtkdXo0MnQ4WTBBREZ0SWRnWkl1d0FJdll1RDYKR09aZ0ViQmtXR2M4c05n
0bEpIdHBpUDNheWdnQ2hLQjZsZGlaQU1IT1pnRW02WkpjNlpDCkVjWHhvaUVxUFpBSC9vTHhYTTVIbUFpRVY2ajZHdk4xb2o0d1px
Db0JmTTgyOWZXUDJMNwowbHZFR201b3N6VmJ5QWR0eDkxelZqTE5YclZhRGlITDVuc2w0Z0FSN0I3SjNONXZJMUI3elRvUDc3N1ZTU
nWUVBN1VEQSs2NlRGMXpMKy9hUEJiYWE0Tkw1NzFrbFN5M1dKdjF6Z1BiNGxBRXppNHFpQVJBSgpVaGNkbUg2V2h6ck1tUHovS1pM
yMDRkZXl1NFpmaklidVJSTVdLdVdMdVVDZ1lFQTFCQ3AKOGpsK1h3RlpEZ2NPaXNXWFF1U2lWTVRrVkpTcmFEcGtRMmRKVHZZeUN5
pNy9wTWNpM2tnCnI4Mno2OSs2OXNLcjVYRkhXLzR2dmF5bWM4V3ppU3hZQ0lpTmx3a0NnWUExYzRwRzZhelA3YU16VC9nbTlGM3UK
rMmJmekNhMEJBbVdLTzFmZ29PVkNua3ljUWxzUlFPWnA1SkxzSXdCY1NuZjVJMzBOemw5CjdKaXBwWHp1MW9FeWNHVkc4TmwwRlFL
zRlovcjBVUjFrRnJUWVA5ZW9XSlRFaFRLVERFeAp3UzI0YUZiWjFlcGJiOU15VHRCTDV6Zk5NbWJtRkhMRUJ3d0xiM0VHZ2xvNGNJ
ueHM4NDMKTzFEd3hEYUNNTTNrRjY2dzYwTFRYd3dHbzhUamQzOS9GRXJsajJqYTBHSzczQ2Z1QWNWWFBndzJkbTQvREgwMApEUFY5
tCg==
kind: Secret
metadata:
  creationTimestamp: "2020-03-22T12:45:03Z"
  name: fortune-https
  namespace: default
  resourceVersion: "66572"
  selfLink: /api/v1/namespaces/default/secrets/fortune-https
  uid: cc76d2e9-acdc-4159-8685-0a711076d920
type: Opaque
```

```
mansishah@macbookpro ~ % kubectl get configmap fortune-config -o yaml
apiVersion: v1
data:
  my-nginx-config.conf: |
    server {
        listen              80;
        server_name         www.kubia-example.com;

        gzip off;
        gzip_types text/plain application/xml;

        location / {
            root   /usr/share/nginx/html;
            index  index.html index.htm;
        }
    }
  sleep-interval: |
    25
kind: ConfigMap
metadata:
  creationTimestamp: "2020-03-21T09:22:14Z"
  name: fortune-config
  namespace: default
  resourceVersion: "63392"
  selfLink: /api/v1/namespaces/default/configmaps/fortune-config
  uid: 51a3789e-5f79-433c-9a65-a7052674545e
```

Using the Secret in a pod

```
mansishah@macbookpro ~ % kubectl edit configmap fortune-config
error: configmaps "fortune-config" is invalid
error: configmaps "fortune-config" is invalid
error: configmaps "fortune-config" is invalid
configmap/fortune-config edited
mansishah@macbookpro ~ % kubectl describe configmap fortune-config
Name:         fortune-config
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
my-nginx-config.conf:
----
server {
    listen              80;
    listen              443 ssl;
    server_name         www.kubia-example.com;
    ssl_certificate     certs/https.cert;
    ssl_certificate_key certs/https.key;
    ssl_protocols       TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers         HIGH;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
}

sleep-interval:
----
25

Events:  <none>
mansishah@macbookpro ~ %
```

```
mansishah@macbookpro ~ % cat fortune-pod-https.yaml
apiVersion: v1
kind: Pod
metadata:
  name: fortune-https
spec:
  containers:
  - image: mansi2210/shahm888:fortune_env
    name: html-generator
    env:
    - name: INTERVAL
      valueFrom:
        configMapKeyRef:
          name: fortune-config
          key: sleep-interval
    volumeMounts:
    - name: html
      mountPath: /var/htdocs
  - image: nginx:alpine
    name: web-server
    volumeMounts:
    - name: html
      mountPath: /usr/share/nginx/html
      readOnly: true
    - name: config
      mountPath: /etc/nginx/conf.d
      readOnly: true
    - name: certs
      mountPath: /etc/nginx/certs/
      readOnly: true
    ports:
    - containerPort: 80
    - containerPort: 443
  volumes:
  - name: html
    emptyDir: {}
  - name: config
    configMap:
      name: fortune-config
      items:
      - key: my-nginx-config.conf
        path: https.conf
  - name: certs
    secret:
      secretName: fortune-https
```

```
mansishah@macbookpro ~ % kubectl create -f fortune-pod-https.yaml
pod/fortune-https created
mansishah@macbookpro ~ % kubectl get po fortune-https
NAME            READY   STATUS    RESTARTS   AGE
fortune-https   2/2     Running   0          16s
mansishah@macbookpro ~ %
```

```
mansishah@macbookpro ~ % kubectl port-forward fortune-https 8443:443 &
[1] 2417
mansishah@macbookpro ~ % Forwarding from 127.0.0.1:8443 -> 443
Forwarding from [::1]:8443 -> 443

mansishah@macbookpro ~ % curl https://localhost:8443 -k -v
*   Trying ::1...
* TCP_NODELAY set
* Connected to localhost (::1) port 8443 (#0)
Handling connection for 8443
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/cert.pem
  CApath: none
* TLSv1.2 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
* ALPN, server accepted to use http/1.1
* Server certificate:
*  subject: CN=www.kubia-example.com
*  start date: Mar 22 12:43:57 2020 GMT
*  expire date: Mar 20 12:43:57 2030 GMT
*  issuer: CN=www.kubia-example.com
*  SSL certificate verify result: self signed certificate (18), continuing anyway.
> GET / HTTP/1.1
> Host: localhost:8443
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.17.9
< Date: Sun, 22 Mar 2020 12:58:25 GMT
< Content-Type: text/html
```

Understanding secret volumes are stored in memory

```
mansishah@macbookpro ~ % kubectl exec fortune-https -c web-server -- mount | grep certs
tmpfs on /etc/nginx/certs type tmpfs (ro,relatime)
mansishah@macbookpro ~ %
```

Understanding image pull Secrets

Creating a Secret for authenticating with a Docker registry

```
mansishah@macbookpro ~ % kubectl create secret docker-registry mydockerhubsecret \
> --docker-username=mansi2210 --docker-password=[redacted]\
> --docker-email=[redacted]@gmail.com
secret/mydockerhubsecret created
mansishah@macbookpro ~ %
```

Using the docker-registry Secret in a pod definition

```
mansishah@macbookpro ~ % vim pod-with-private-image.yaml
mansishah@macbookpro ~ % cat pod-with-private-image.yaml
apiVersion: v1
kind: Pod
metadata:
  name: private-pod
spec:
  imagePullSecrets:
  - name: mydockerhubsecret
  containers:
  - image: mansi2210/shahm888:fortune_env
    name: main
mansishah@macbookpro ~ %
```

```
mansishah@macbookpro ~ % kubectl create -f pod-with-private-image.yaml
pod/private-pod created
mansishah@macbookpro ~ % kubectl get pod private-pod
NAME          READY   STATUS    RESTARTS   AGE
private-pod   1/1     Running   0          12s
```