

A Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key.



Secrets



Kubernetes can store sensitive information (passwords, keys, certificates, etc.)

Avoids storing secrets in container images, in files, or in deployment manifests

Mount secrets into pods as files or as environment variables

Kubernetes only makes secrets available to Nodes that have a Pod requesting the secret

Secrets are stored in tmpfs on a Node (not on disk)



Enable encryption at rest for cluster data
(<https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data>)

Limit access to etcd (where Secrets are stored) to only admin users

Use SSL/TLS for etcd peer-to-peer communication

Manifest (YAML/JSON) files only base64 encode the Secret

Pods can access Secrets so secure which users can create Pods. Role-based access control (RBAC) can be used.

<https://kubernetes.io/docs/concepts/configuration/secret/#best-practices>

Secrets Best Practices



Creating a Secret

Secrets can be created using `kubectl create secret`

```
# Create a secret and store securely in Kubernetes
```

```
kubectl create secret generic my-secret  
  --from-literal=pwd=my-password
```

```
# Create a secret from a file
```

```
kubectl create secret generic my-secret  
  --from-file=ssh-privatekey=~/.ssh/id_rsa  
  --from-file=ssh-publickey=~/.ssh/id_rsa.pub
```

```
# Create a secret from a key pair
```

```
kubectl create secret tls tls-secret --cert=path/to/tls.cert  
  --key=path/to/tls.key
```

Question:

Can I declaratively define secrets using YAML?

Answer:

Yes – but any secret data is only base64 encoded in the manifest file!



```
apiVersion: v1
kind: Secret
metadata:
  name: db-passwords
type: Opaque
data:
  app-password: cGFzc3dvcmQ=
  admin-password: dmVyeV9zZWNYZXQ=
```

◀ Define a Secret

◀ Secret name

◀ Keys/values for Secret



Get secrets

kubectl get secrets

```
iMac-3:~ danwahlin$ k get secrets
+ kubectl get secrets
NAME                                TYPE                                DATA  AGE
db-passwords                        Opaque                             2      34m
default-token-rxmjb                 kubernetes.io/service-account-token 3      66d
```

Get YAML for specific secret

kubectl get secrets db-passwords -o yaml

```
iMac-3:~ danwahlin$ k get secrets db-passwords -o yaml
+ kubectl get secrets db-passwords -o yaml
apiVersion: v1
data:
  mongodb-password: cGFzc3dvcmQ=
  mongodb-root-password: cGFzc3dvcmQ=
kind: Secret
metadata:
  creationTimestamp: "2019-03-22T00:40:05Z"
  name: db-passwords
  namespace: default
  resourceVersion: "3481795"
  selfLink: /api/v1/namespaces/default/secrets/db-passwords
  uid: 0982413e-4c3b-11e9-b7f0-025000000001
type: Opaque
```

Listing Secret Keys

A list of secrets can be retrieved using **kubectl get secrets**



Accessing a Secret: Environment Vars

Pods can access Secret values through environment vars

DATABASE_PASSWORD environment var created

```
apiVersion: v1
```

```
kind: Secret
```

```
metadata:
```

```
  name: db-passwords
```

```
type: Opaque
```

```
data:
```

```
  db-password: cGFzc3dvcmQ=
```

```
  admin-password: dmVyeV9zZWNYZXQ=
```

```
apiVersion: apps/v1
```

```
...
```

```
spec:
```

```
  template:
```

```
    ...
```

```
  spec:
```

```
    containers: ...
```

```
    env:
```

```
      - name: DATABASE_PASSWORD
```

```
        valueFrom:
```

```
          secretKeyRef:
```

```
            name: db-passwords
```

```
            key: db-password
```



Accessing a Secret: Volumes

Pods can access secret values through a volume

Each key is converted to a file - value is added into the file

```
apiVersion: v1
kind: Secret
metadata:
  name: db-passwords
type: Opaque
data:
  db-password: cGFzc3dvcmQ=
  admin-password: dmVyeV9zZWNyZXQ=
```

```
apiVersion: apps/v1
...
spec:
  template:
    ...
    spec:
      volumes:
        - name: secrets
          secret:
            secretName: db-passwords
          containers:
            volumeMounts:
              - name: secrets
                mountPath: /etc/db-passwords
                readOnly: true
```

