Configuring Applications with ConfigMaps and Secrets

Relevant Documentation

- ConfigMaps
- Secrets

Exam Tips

- A ConfigMap stores configuration data that can be passed to containers.
- A Secret is designed to store sensitive configuration data such as passwords or API keys.
- Data from both ConfigMaps and Secrets can be passed to containers using either a volume mount or environment variables.

Lesson Reference

Log in to the **control plane node**.

Create a ConfigMap.

```
vi my-configmap.yml
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
data:
  message: Hello, World!
  app.cfg: |
    # A configuration file!
    key1=value1
    key2=value2
```

```
kubectl apply -f my-configmap.yml
```

Create a Pod that uses the ConfigMap.

```
vi cm-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: cm-pod
spec:
  restartPolicy: Never
  containers:
  - name: busybox
    image: busybox:stable
    command: ['sh', '-c', 'echo $MESSAGE; cat /config/app.cfg']
    env:
    - name: MESSAGE
      valueFrom:
        configMapKeyRef:
          name: my-configmap
          key: message
    volumeMounts:
    - name: config
      mountPath: /config
      readOnly: true
  volumes:
  - name: config
    configMap:
      name: my-configmap
      items:
      - key: app.cfg
        path: app.cfg
```

```
kubectl apply -f cm-pod.yml
```

Check the Pod logs. You should see the message followed by the contents of the config file, both from the ConfigMap.

```
kubectl logs cm-pod
```

Get base64-encoded strings for some sensitive data.

```
echo Secret Stuff! | base64

echo Secret stuff in a file! | base64
```

Create a Secret using the base64-encoded data.

```
vi my-secret.yml
```

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
type: Opaque
data:
  sensitive.data: U2VjcmV0IFN0dWZmIQo=
  passwords.txt: U2VjcmV0IHN0dWZmIGluIGEgZmlsZSEK
```

```
kubectl apply -f my-secret.yml
```

Create a Pod that uses the ConfigMap.

```
vi secret-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-pod
spec:
  restartPolicy: Never
  containers:
  - name: busybox
    image: busybox:stable
    command: ['sh', '-c', 'echo $SENSITIVE_STUFF; cat /config/passwords.txt']
    env:
    - name: SENSITIVE_STUFF
      valueFrom:
        secretKeyRef:
          name: my-secret
          key: sensitive.data
    volumeMounts:
    - name: secret-config
      mountPath: /config
      readOnly: true
  volumes:
  - name: secret-config
    secret:
      secretName: my-secret
      items:
      - key: passwords.txt
        path: passwords.txt
```

```
kubectl apply -f secret-pod.yml
```

Check the Pod logs. You should see the data from the Secret, both the environment variable and the file.

```
kubectl logs secret-pod
```