One of the challenges in managing a complex application infrastructure is ensuring that sensitive data remains secure. It is always important to store sensitive data, such as tokens, passwords, and keys, in a secure, encrypted form. In this lesson, we will talk about Kubernetes secrets, a way of securely storing data and providing it to containers. We will also walk through the process of creating a simple secret, and passing the sensitive data to a container as an environment variable.

Relevant Documentation

- https://kubernetes.io/docs/concepts/configuration/secret/

Lesson Reference

Create a secret using a yaml definition like this. It is a good idea to delete the yaml file containing the sensitive data after the secret object has been created in the cluster.

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
stringData:
  myKey: myPassword
```

Once a secret is created, pass the sensitive data to containers as an environment variable:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-secret-pod
spec:
  containers:
  - name: myapp-container
    image: busybox
    command: ['sh', '-c', "echo Hello, Kubernetes! && sleep 3600"]
    env:
    - name: MY_PASSWORD
      valueFrom:
        secretKeyRef:
          name: my-secret
          key: myKey
```