Imagine deploying an app where you need to store database URLs and passwords securely. So, how does Kubernetes manage this?"

- **ConfigMaps and Secrets in Kubernetes**

ConfigMaps and **Secrets** are Kubernetes objects used to manage configuration data separately from application code.

- **ConfigMaps** store non-sensitive configuration data in key-value pairs.
- **Secrets** store sensitive information like passwords, API keys, and certificates, and they are encoded in Base64.

Example Demonstration

We'll create:

1. A **ConfigMap** with a database URL.
2. A **Secret** with a database password.
3. A simple **Pod** that uses these values.

Step 1: Create a ConfigMap

```
yaml
------
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-config
data:
  DATABASE_URL: "mysql://db-service:3306/mydb"
```

Apply it using:

```
sh
------
kubectl apply -f configmap.yaml
```

Step 2: Create a Secret

```
yaml
------
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
type: Opaque
data:
  DATABASE_PASSWORD: bXlwYXNzd29yZA==  # Base64 encoded value of "mypassword"
```

Apply it using:

```
sh
------
kubectl apply -f secret.yaml
```

To manually encode a password:

```
sh
------
echo -n "mypassword" | base64
```

Step 3: Use ConfigMap and Secret in a Pod

```
yaml
------
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
    - name: my-container
      image: nginx
      env:
        - name: DATABASE_URL
          valueFrom:
            configMapKeyRef:
              name: my-config
              key: DATABASE_URL
        - name: DATABASE_PASSWORD
          valueFrom:
            secretKeyRef:
              name: my-secret
              key: DATABASE_PASSWORD
```

Apply it using:

```
sh
------
kubectl apply -f pod.yaml
```

Step 4: Verify

Check if the pod is running:

```
sh
------
kubectl get pods
```

Check environment variables inside the pod:

```
sh
------
kubectl exec -it my-pod -- env | grep DATABASE
```

Summary

✅ ConfigMaps manage non-sensitive configuration.
✅ Secrets manage sensitive data securely.
✅ Both can be used as environment variables or mounted as files.

1. What is the primary purpose of a ConfigMap in Kubernetes?

A) Storing sensitive information like passwords and API keys
B) Storing non-sensitive configuration data like environment variables
C) Managing container networking in Kubernetes
D) Encrypting application data at rest

Answer: B) Storing non-sensitive configuration data like environment variables

---

2. How are Secrets stored in Kubernetes?

A) Plain text format
B) Base64-encoded format
C) AES-256 encrypted format
D) JSON format

Answer: B) Base64-encoded format