**The current project how the values injected into AKS service.**

**SCENARIO**

* You're running a **Spring Boot application** in **AKS**.
* Your GitHub repo previously had some sensitive data like:
  + DB\_PASSWORD
  + AKS\_API\_KEY
  + APP\_ENV
* You want these to be securely **injected into the application** without hardcoding them.

**✅ 1. Create Kubernetes Secrets**

Create a secret YAML file:

yaml

CopyEdit

# secret.yaml

apiVersion: v1

kind: Secret

metadata:

name: my-app-secrets

type: Opaque

data:

db-password: cGFzc3dvcmQ= # base64 for 'password'

aks-api-key: YWtzX2tleV92YWx1ZQ== # base64 for 'aks\_key\_value'

To encode:

bash

CopyEdit

echo -n "password" | base64 # cGFzc3dvcmQ=

**✅ 2. Create Kubernetes ConfigMap**

Create a configmap YAML file:

yaml

CopyEdit

# configmap.yaml

apiVersion: v1

kind: ConfigMap

metadata:

name: my-app-config

data:

app.env: "prod"

aks.endpoint: "https://my-aks-cluster-endpoint"

**✅ 3. Deploy Secret and ConfigMap**

Apply them to your AKS cluster:

bash

CopyEdit

kubectl apply -f secret.yaml

kubectl apply -f configmap.yaml

**✅ 4. Update Spring Boot Deployment YAML**

Inject these as **environment variables**:

yaml

CopyEdit

# deployment.yaml (partial)

spec:

containers:

- name: spring-boot-app

image: my-registry/my-spring-app:latest

env:

- name: DB\_PASSWORD

valueFrom:

secretKeyRef:

name: my-app-secrets

key: db-password

- name: AKS\_API\_KEY

valueFrom:

secretKeyRef:

name: my-app-secrets

key: aks-api-key

- name: APP\_ENV

valueFrom:

configMapKeyRef:

name: my-app-config

key: app.env

- name: AKS\_ENDPOINT

valueFrom:

configMapKeyRef:

name: my-app-config

key: aks.endpoint

**✅ 5. Use in Spring Boot (application.yml)**

Spring Boot automatically maps environment variables:

yaml

CopyEdit

# application.yml

spring:

datasource:

password: ${DB\_PASSWORD}

app:

env: ${APP\_ENV}

aks:

api-key: ${AKS\_API\_KEY}

endpoint: ${AKS\_ENDPOINT}

You can also access them via @Value or @ConfigurationProperties:

java

CopyEdit

@Value("${app.aks.api-key}")

private String aksApiKey;

**📦 Summary**

| **Value** | **Source** | **Type** | **Injected As** |
| --- | --- | --- | --- |
| DB\_PASSWORD | Secret | Sensitive | env var |
| AKS\_API\_KEY | Secret | Sensitive | env var |
| APP\_ENV | ConfigMap | Public | env var |
| AKS\_ENDPOINT | ConfigMap | Public | env var |

**✅ Tips**

* **DO NOT store secrets in GitHub.**
* **Use base64** when storing secrets in Secret resources.
* If your cluster is Azure-integrated, consider **Azure Key Vault** + CSI Driver for secure secret mounting.