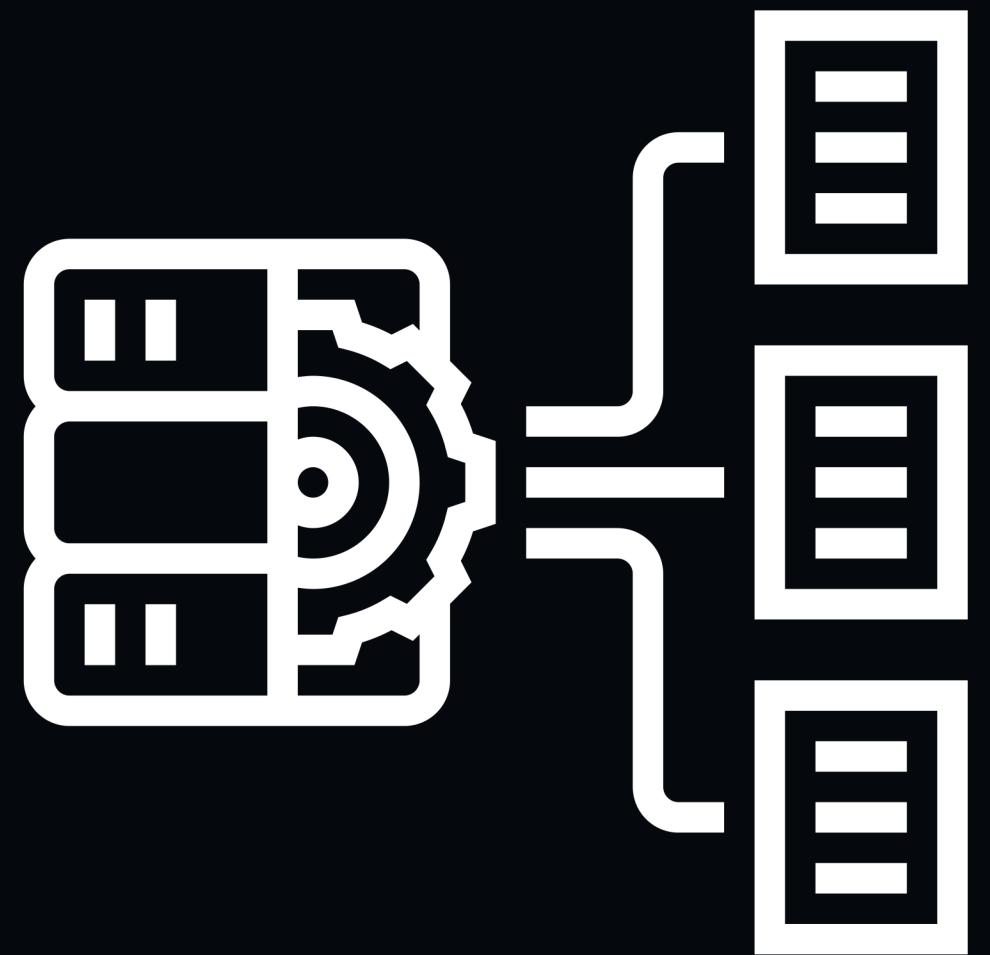


CONFIGMAP

- **ConfigMap** is a Kubernetes object that allows you to **store configuration data as key-value pairs**
- It is used by your containers at runtime, without having to rebuild your container images
- Use case
 - Environment variables
 - Configuration file



CONFIGMAP



```
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: django-app-configmap
5 data:
6   ENV: test
7   PUB_TOPIC: codemke
8   CON_TOPIC: codemke
9
```



```
1 spec:
2   containers:
3     - name: django-app
4       image: joonlee0228/django:0.0.1
5   resources:
6     requests:
7       cpu: "100m"
8   env:
9     - name: ENV
10    valueFrom:
11      configMapKeyRef:
12        name: django-app-configmap
13        key: ENV
```

SECRET

- Similar to a ConfigMap, but it is **designed for storing sensitive data**, such as passwords, API keys, or TLS certificates
- The data is stored in an **encoded or encrypted format** to help protect it from unauthorized access
- You can reference it using an **environment variable**, a volume mount, or a command-line argument
- Use case
 - Storing database credentials or other sensitive information that is required by your application at runtime
 - Storing TLS to access between Kubernetes objects
 - Storing external API keys or credentials



SECRET TYPE

- **Opaque**: The default **Secret** type, which allows you to store arbitrary data in an encoded format.



```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: my-secret
5   type: Opaque
6 data:
7   username: dXNlcm5hbWU=
8   password: cGFzc3dvcmQ=
9
```



```
1 # encoding
2 # "-n" prevent additional newline
3 # e.g. echo -n username | base64
4 echo -n <string> | base64
5
6 # decoding
7 echo <base64-encoded-string> | base64 -d
8
```

SECRET TYPE

- **kubernetes.io/tls**: A **Secret** type that is used to store TLS certificates and private keys.



```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: tls-secret
5 type: kubernetes.io/tls
6 data:
7   tls.crt: <base64-encoded-certificate>
8   tls.key: <base64-encoded-private-key>
```

SECRET TYPE

- **kubernetes.io/dockerconfigjson**: A Secret type that is used to store Docker registry credentials in JSON format



```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: docker-secret
5 type: kubernetes.io/dockerconfigjson
6 data:
7   .dockerconfigjson: <base64-encoded-JSON-data>
8
```

SECRET TYPE

- **kubernetes.io/service-account-token**: A Secret type that is automatically created by Kubernetes for each service account, and contains the service account's token and other metadata.



```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: my-serviceaccount-token
5   annotations:
6     kubernetes.io/service-account.name: my-serviceaccount
7   type: kubernetes.io/service-account-token
```

SECRET AS ENV



```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: django-app-secret
5 type: Opaque
6 data:
7   username: dXNlcm5hbWU=
8   password: cGFzc3dvcmQ=
9
```



```
1 spec:
2   containers:
3     - name: django-app
4       image: joonlee0228/django:0.0.1
5     resources:
6       requests:
7         cpu: "100m"
8   env:
9     - name: USERNAME
10    valueFrom:
11      secretKeyRef:
12        name: django-app-secret
13        key: username
```

SECRET AS MOUNT



```
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: tls-secret
5 type: kubernetes.io/tls
6 data:
7   tls.crt: <base64-encoded-certificate>
8   tls.key: <base64-encoded-private-key>
```



```
1 apiVersion: v1
2 kind: Pod
3 metadata:
4   name: my-pod
5 spec:
6   containers:
7     - name: my-container
8       image: my-image
9     volumeMounts:
10    - name: tls-volume
11      mountPath: /etc/tls
12      readOnly: true
13   volumes:
14    - name: tls-volume
15      secret:
16        secretName: tls-secret
```