Secrets

A Secret is a Kubernetes object used to store sensitive data, like:

- Passwords
- API keys
- Certificates
- Tokens

It's similar to a ConfigMap, but meant for confidential information, and the data is stored Base64-encoded.

Why Use Secrets

- Prevent exposing sensitive data in plain text.
- Securely pass credentials to pods.
- Avoid hardcoding secrets into application code or container images.
- Better access control (Secrets can be encrypted at rest and controlled via RBAC).

Types of secrets

Type	Description
`Opaque` (default)	Generic key-value pairs
`kubernetes.io/basic-auth`	Username/password
`kubernetes.io/dockerconfigjson`	Docker registry credentials
`kubernetes.io/tls`	TLS cert and key

Example: Creating a Secret (Opaque)

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret          # Name of the secret
type: Opaque               # Type of secret (generic key-value)
data:                      # All values must be base64-encoded
  username: dXNlcg==       # 'user' in base64
  password: cGFzc3dvcmQ=   # 'password' in base64
```

use this secret in a Pod as environment variables

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-env-pod         # Name of the Pod
spec:
  containers:
    - name: my-container       # Container name
```

```
        image: nginx             # Image to use (any container image)
        env:                     # Inject environment variables
          - name: USERNAME       # Name of ENV variable in container
            valueFrom:
              secretKeyRef:
                name: my-secret  # Refers to the Secret name
                key: username    # Key from the Secret to use
          - name: PASSWORD
            valueFrom:
              secretKeyRef:
                name: my-secret
                key: password
```