Secrets in Kubernetes

Secrets:

- Secrets are used to pass sensitive information to Pods in Kubernetes.
- They can store sensitive data such as usernames, passwords, TLS certificates, keys, and Docker registry credentials.
- Secrets encode all the data in Base64 and create an object that can be used in any Pod.

Types of Secrets:

a) Generic Secret: A basic type of secret used to store generic key-value pairs.

b) Docker Registry Secret: Docker registry secrets are used to authenticate and access private Docker repositories to pull images.

c) TLS Secret: TLS secrets are used to store TLS certificates and private keys required for secure communication.

Secret Creation:

- Secrets can be created in both declarative and imperative ways.
- Declarative: Use YAML or JSON manifest files to define the secret's properties.
- Imperative: Use the kubectl create secret <secret-type> <secret-name> --from-literal=<key>=<value> command.
- The --from-literal flag allows you to pass multiple key-value pairs to create the secret.
- Alternatively, you can use the --from-file flag to create a secret from a file's data.

Secrets in Pods:

There are three ways to use secrets in Pods:

a) Full Secret Pass:

- You can pass the entire secret to a Pod using the envFrom field.
- All the key-value pairs from the secret will be available as environment variables in the Pod.

b) Specific Key Reference:

- You can retrieve a specific value from a secret and assign it to an environment variable.
- This allows you to use only a specific piece of sensitive information from the secret.

c) Volume Mount:

- You can mount a secret to a specific location in the Pod's filesystem.
- The secret's key-value pairs will be accessible as files, allowing applications to read them from the mounted location