Secret

Ow

Secret in kubernetes

- A Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key.
- Pod to use a Secret:
  - As files in a volume mounted on one or more of its containers.
  - As container environment variable.
- You can specify the data and/or the stringData field when creating a configuration file for a Secret. The data and the stringData fields are optional.
- Individual secrets are limited to 1MiB in size.

Secret in Data/StringData

- The kubernetes.io/basic-auth type is provided for storing credentials needed for basic authentication. When using this Secret type, the data field of the Secret must contain one of the following two keys:
  - username: the user name for authentication
  - password: the password or token for authentication
- Both values for the above two keys are base64 encoded strings. You can, of course, provide the clear text content using the stringData for Secret creation.

Manifest file

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-basic-auth
type: kubernetes.io/basic-auth
stringData:
  username: admin      # required field for kubernetes.io/basic-auth
  password: t0p-Secret # required field for kubernetes.io/basic-auth
```

Ssh authentication secuirty

- The builtin type kubernetes.io/ssh-auth is provided for storing data used in SSH authentication. When using this Secret type, you will have to specify a ssh-privatekey key-value pair in the data (or stringData) field as the SSH credential to use.

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret-ssh-auth
type: kubernetes.io/ssh-auth
data:
  # the data is abbreviated in this example
  ssh-privatekey: |
          MIIEpQIBAAKCAQEAulqb/Y ...
```

TLS Secret

- Kubernetes provides a builtin Secret type kubernetes.io/tls for storing a certificate and its associated key that are typically used for TLS.
- Using this type of Secret, the tls.key and the tls.crt key must be provided in the data field of the Secret configuration

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: secret-tls
type: kubernetes.io/tls
data:
  # the data is abbreviated in this example
  tls.crt: |
        MIIC2DCCAcCgAwIBAgIBATANBgkqh ...
  tls.key: |
        MIIEpgIBAAKCAQEA7yn3bRHQ5FHMQ ...
```

BootStrap Token

- A bootstrap token Secret can be created by explicitly specifying the Secret type to bootstrap.kubernetes.io/token. This type of Secret is designed for tokens used during the node bootstrap process.

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: bootstrap-token-5emitj
  namespace: kube-system
type: bootstrap.kubernetes.io/token
data:
  auth-extra-groups: c3lzdGVtOmJvb3RzdHJhcHBlcnM6a3ViZWFkbTpkZWZhdWx0LW5vZGUtdG9rZW4=
  expiration: MjAyMC0wOS0xM1QwNDozOToxMFo=
  token-id: NWVtaXRq
  token-secret: a3E0Z2lodnN6emduMXAwcg==
  usage-bootstrap-authentication: dHJ1ZQ==
  usage-bootstrap-signing: dHJ1ZQ==
```

..

token-id: A random 6 character string as the token identifier. Required.

token-secret: A random 16 character string as the actual token secret. Required.

description: A human-readable string that describes what the token is used for. Optional.

expiration: An absolute UTC time using RFC3339 specifying when the token should be expired. Optional.

usage-bootstrap-<usage>: A boolean flag indicating additional usage for the bootstrap token.

auth-extra-groups: A comma-separated list of group names that will be authenticated as in addition to the system:bootstrappers group.

Secret using kubectl command

- echo -n 'admin' > ./username.txt
- echo -n '1f2d1e2e67df' > ./password.txt
- https://kubernetes.io/docs/tasks/configmap-secret/managing-secret-using-kubectl/
- kubectl create secret generic db-user-pass \

  --from-file=./username.txt \

--from-file=./password.txt