In Kubernetes, a secret is an object that allows you to store and manage sensitive information such as passwords, tokens, and private keys. Secrets are typically used to provide secure access to resources and services within a Kubernetes cluster.

Secrets can be created manually or automatically, and they can be mounted as files or environment variables in a container. Secrets are stored within etcd, the distributed key-value store used by Kubernetes, and are encrypted at rest using Kubernetes secrets encryption.

To create a secret in Kubernetes, you can use the kubectl create secret command

You can then use this secret in your Kubernetes deployment by referencing it in the deployment YAML file,

```yaml
apiVersion: v1
kind: Secret
metadata:
	name: mongodb-secret
type: Opaque
data:
  mongodb-root-username: bW9uZ29kYg==
  mongodb-root-password: bW9uZ29kYg==
```

```
#echo -n 'username'| base64
#echo -n 'password'| base64
```

The apiVersion: v1 and kind: Secret indicate that you are creating a Kubernetes secret object. The metadata field specifies metadata about the secret, including the name of the secret, which is mongodb-secret in this case.

The type: Opaque indicates that the secret data is an arbitrary byte array, and is not specifically encoded or decoded. This type of secret is typically used for storing sensitive information such as passwords, private keys, and other binary data.

The data field contains the actual secret data, which is base64-encoded. The mongodb-root-username and mongodb-root-password fields are keys that correspond to the values you want to store. In this case, the values are both bW9uZ29kYg==, which are the base64-encoded values of the username and password strings, respectively.