

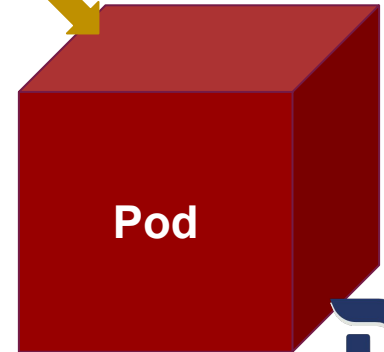
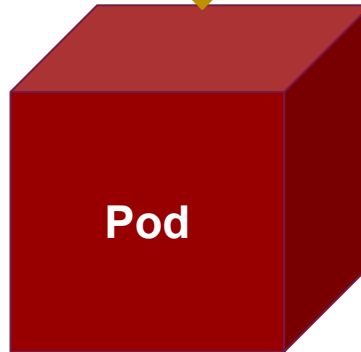
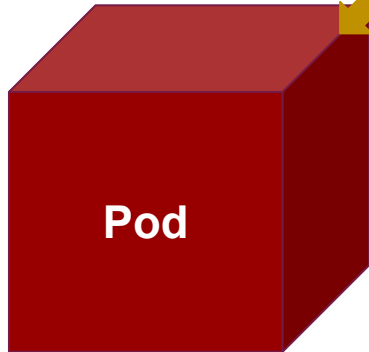
Secrets In Kubernetes



What is Secret?



username: admin
password: %\$^res!@#
tls.crt: |
MIIC2DCCAcCgAwIBAgIBATAN...
tls.key: |
MIIEpgIBAAKCAQEA7yn3bRHQ...



Secret Creation



Imperative Way

```
kubectl create secret <secret-type>  
<secret-name> --from-literal=<key>=<value>
```

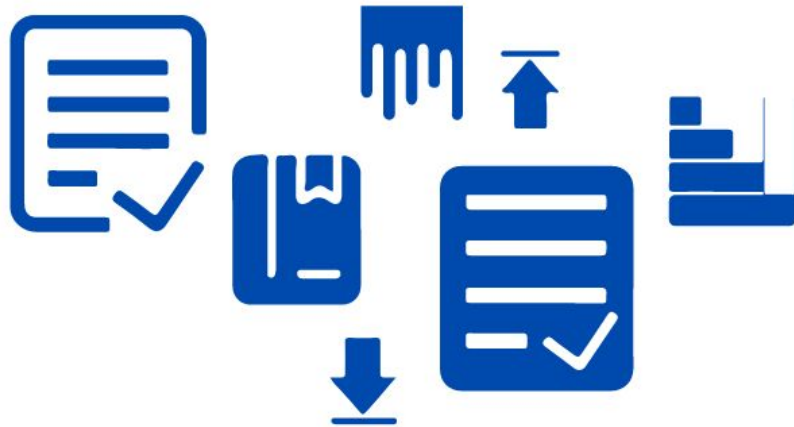
```
kubectl create secret <secret-type>  
<secret-name> --from-file=<path-to-file>
```

Declarative Way

```
apiVersion: v1  
kind: Secret  
metadata:  
  name: mysecret  
type: Opaque  
data:  
  username: YWRtaW4=  
  password: MWYyZDFIMmU2N2Rm
```



Types of Secrets



Types of Secrets

1

Generic

2

Docker-Registry

3

TLS

```
kubectl create secret generic dev-db-secret  
--from-literal=username=devuser  
--from-literal=password='S!B\*d$zDsb='
```





Types of Secrets

1

Generic

2

Docker-Registry

3

TLS

```
kubectl create secret docker-registry docker-secret  
--docker-email=example@gmail.com  
--docker-username=dev  
--docker-password=pass1234  
--docker-server=my-registry.example:5000
```





Types of Secrets

1

Generic

2

Docker-Registry

3

TLS

```
kubectl create secret tls my-tls-secret  
--cert=path/to/cert/file --key=path/to/key/file
```



Secret In Pods



my-secret

username="admin"
password="@#\$134\$%6"



```
apiVersion: v1
kind: Pod
metadata:
  name: secret-demo
spec:
  containers:
  - name: demo-container
    image: my-image
  envFrom:
  - secretRef:
      name: my-secret
```

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-demo
spec:
  containers:
  - name: demo-container
    image: my-image
    env:
    - name: USER
      valueFrom:
        secretKeyRef:
          name: my-secret
          key: username
```

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-demo
spec:
  containers:
  - name: demo-container
    image: my-image
    volumeMounts:
    - name: data
      mountPath: /etc/cert-data
  volumes:
  - name: data
    secret:
      secretName: my-secret
```

