Secret 存在意义

Secret 解决了密码、token、密钥等敏感数据的配置问题，而不需要把这些敏感数据暴露到镜像或者 Pod Spec 中。Secret 可以以 Volume 或者环境变量的方式使用

Secret 有三种类型：

- **Service Account**： 用来访问 Kubernetes API，由 Kubernetes 自动创建，并且会自动挂载到 Pod 的 `/run/secrets/kubernetes.io/serviceaccount` 目录中
- **Opaque**： base64编码格式的Secret，用来存储密码、密钥等
- **kubernetes.io/dockerconfigjson**： 用来存储私有 docker registry 的认证信息

Service Account

Service Account 用来访问 Kubernetes API，由 Kubernetes 自动创建，并且会自动挂载到 Pod的 `/run/secrets/kubernetes.io/serviceaccount` 目录中

```
$ kubectl run nginx --image nginx
deployment "nginx" created
$ kubectl get pods
NAME                     READY     STATUS    RESTARTS   AGE
nginx-3137573019-md1u2   1/1       Running   0          13s
$ kubectl exec nginx-3137573019-md1u2 ls /run/secrets/kubernetes.io/serviceaccount
ca.crt
namespace
token
```

Opaque Secret

Ⅰ、创建说明

Opaque 类型的数据是一个 map 类型，要求 value 是 base64 编码格式：

```
$ echo -n "admin" | base64
YWRtaW4=
$ echo -n "1f2d1e2e67df" | base64
MWYyZDFlMmU2N2Rm
```

secrets.yml

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  password: MWYyZDFlMmU2N2Rm
  username: YWRtaW4=
```

II、使用方式

1、将 Secret 挂载到 Volume 中

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    name: seret-test
  name: seret-test
spec:
  volumes:
  - name: secrets
    secret:
      secretName: mysecret
  containers:
  - image: hub.atguigu.com/library/myapp:v1
    name: db
    volumeMounts:
    - name: secrets
      mountPath: "
      readOnly: true
```

2、将 Secret 导出到环境变量中

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: pod-deployment
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: pod-deployment
    spec:
      containers:
      - name: pod-1
        image: hub.atguigu.com/library/myapp:v1
        ports:
```

```
        - containerPort: 80
        env:
        - name: TEST_USER
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: username
        - name: TEST_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: password
```

kubernetes.io/dockerconfigjson

使用 Kuberctl 创建 docker registry 认证的 secret

```
$ kubectl create secret docker-registry myregistrykey --docker-server=DOCKER_REGISTRY_SERVER --
docker-username=DOCKER_USER --docker-password=DOCKER_PASSWORD --docker-email=DOCKER_EMAIL
secret "myregistrykey" created.
```

在创建 Pod 的时候，通过 `imagePullSecrets` 来引用刚创建的 `myregistrykey`

```
apiVersion: v1
kind: Pod
metadata:
  name: foo
spec:
  containers:
    - name: foo
      image: roc/awangyang:v1
  imagePullSecrets:
    - name: myregistrykey
```