**Kubernetes Secrets**

* How do you store sensitive information? Should you include it in Docker image? How about in a pod spec? NEVER?
* Kubernetes Secrets let you store and manage sensitive information that your pods can access at runtime. Think passwords, OAuth tokens, and ssh keys.
* When using kubectl get, you won't see the contents of a secret. But they are accessible to those with access directly to the cluster.
* Its best to have secrets managed by a limited set of people who know how to keep them safe. And don’t just check them into source control alongside your resources.

kubectl create **secret** generic dbsecrets --**from-literal** user=admin **--from-literal** password=tiger1234

kubectl get secret dbsecret -o yaml

**Secret.yaml**

apiVersion: v1

kind: Secret

metadata:

  name: dbsecrets

type: Opaque

data:

  user: YWRtaW4= #base64 value of admin

  password: dGlnZXIxMjM0 #base64 value of tiger1234

**Referencing a secret:**

**pod.yaml**

apiVersion: v1

kind: Pod

metadata:

  name: test-pod

spec:

  containers:

    - name: test-container

      image: nginx

      env:

        - name: USERENV

          valueFrom:

**secretKeyRef**:

              name: dbsecrets

              key: user

        - name: MYPASSWORDENV

          valueFrom:

**secretKeyRef**:

              name: dbsecrets

              key: password

**Execute the following commands:**

1. Kubectl apply secret.yaml
2. Kubectl apply pod.yaml
3. Kubectl get secrets dbsecrets -o yaml

**Secrets from file:**

**credentials.txt**

username=admin

password=tiger1234

**Command:**

kubectl create secret generic mysecrets **--from-env-file** credentials.txt

**OR**

kubectl create secret generic mysecrets **--from-file**=ssh-privatekey=~/.ssh/id\_rsa **--from-file**=ssh-publickey=~/.ssh/id\_rsa.pub

**Using the Private Repository from Docker Hub using Secret**

**Step0: Push an image to Registry**

docker login

docker tag nginx coderden/mynginx

docker push coderden/mynginx

Go to docker.io and make the image repository as PRIVATE.

**Step1: Create a Secret:**

If DockerHub is used:

**kubectl create secret** **docker-registry** **mydockersecret** **--docker-username**="<USERNAME>" **--docker-password** "<PASSWORD>" --docker-server=docker.io

**Step2: Update the YAML file**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: myhelloapp-deployment

spec:

  replicas: 2

  selector:

    matchLabels:

      app: myhelloapp

  template:

    metadata:

      labels:

        app: myhelloapp

    spec:

      containers:

      - name: myhelloapp-container

        image: coderden/mynginx

        ports:

        - containerPort: 80

        imagePullPolicy: Always # IfNotPresent #Never

      imagePullSecrets:

      - name: mydockersecret

**Step3: Deploy**

kubectl apply -f deployment.yaml

Create a LoadBalancer Service and verify by visiting localhost:PORT (e.g. 8080)

**Secrets YAML:**

**kubectl get secrets mydockersecret -o yaml > secrets.yaml**

apiVersion: v1

data:

  .dockerconfigjson: eyJhdXRocyI6eyJodHRwczovL2luZGV4Lm...RPT0ifX19

kind: Secret

metadata:

  creationTimestamp: "2020-07-01T12:38:09Z"

  name: mysecret

  namespace: default

  resourceVersion: "369988"

  selfLink: /api/v1/namespaces/default/secrets/docker-registry-secret

  uid: cf146992-8896-48e7-b9df-80a9051036b2

type: kubernetes.io/dockerconfigjson

The value of the .dockerconfigjson field is a base64 representation of your Docker credentials

You can visit <https://www.base64decode.org/> and decode the Base64 value.

**More about Pulling an Image from Private Registry:**

<https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/>