Secrets In Kubernetes

A Kubernetes Secret is a Kubernetes object that allows you to store

- sensitive data, such as passwords, API keys, and certificates.

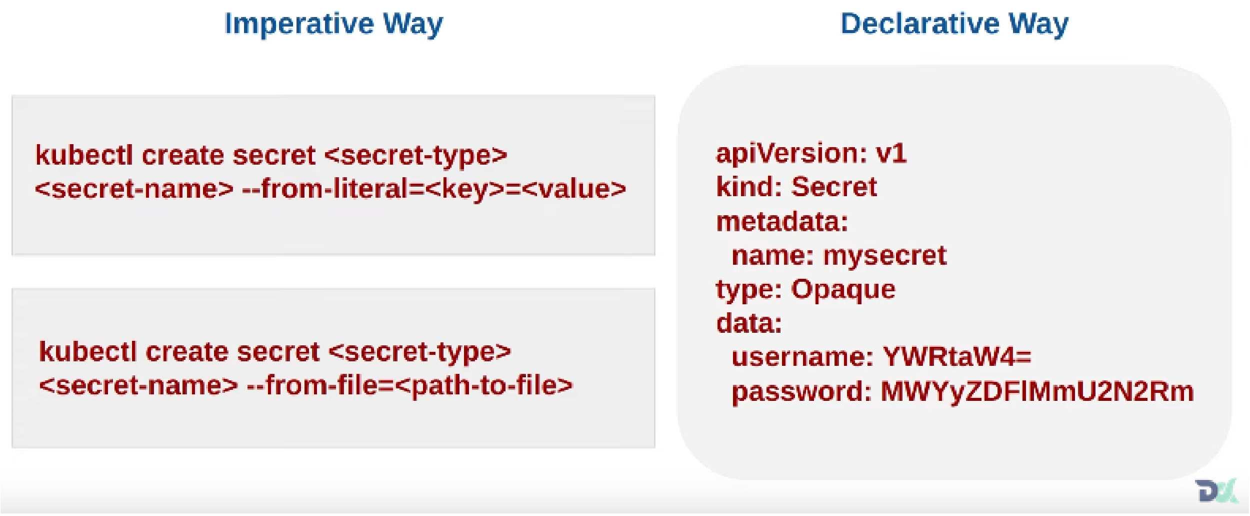
Kubernetes encode information in secret then store it as a object. Secrets are stored in a secure way, and they can only be accessed by pods that have the appropriate permissions.

Secrets are similar to ConfigMaps, but they are designed for storing sensitive data. ConfigMaps are designed for storing arbitrary data, such as environment variables and configuration files or non sensitive information.

There are Three types of secrets in Kubernetes:

- **Opaque(generic) secrets:** Opaque secrets store data in a binary format. This type of secret is used for storing data that cannot be easily represented as text, such as passwords and certificates.
- **Docker secrets:** Docker secrets store data in a format that can be used by Docker containers. This type of secret is used for storing data that needs to be passed to Docker containers, such as database passwords and SSH keys.
- **TLs** : To store tls cet and key

How to create Secret :



Imperative Way

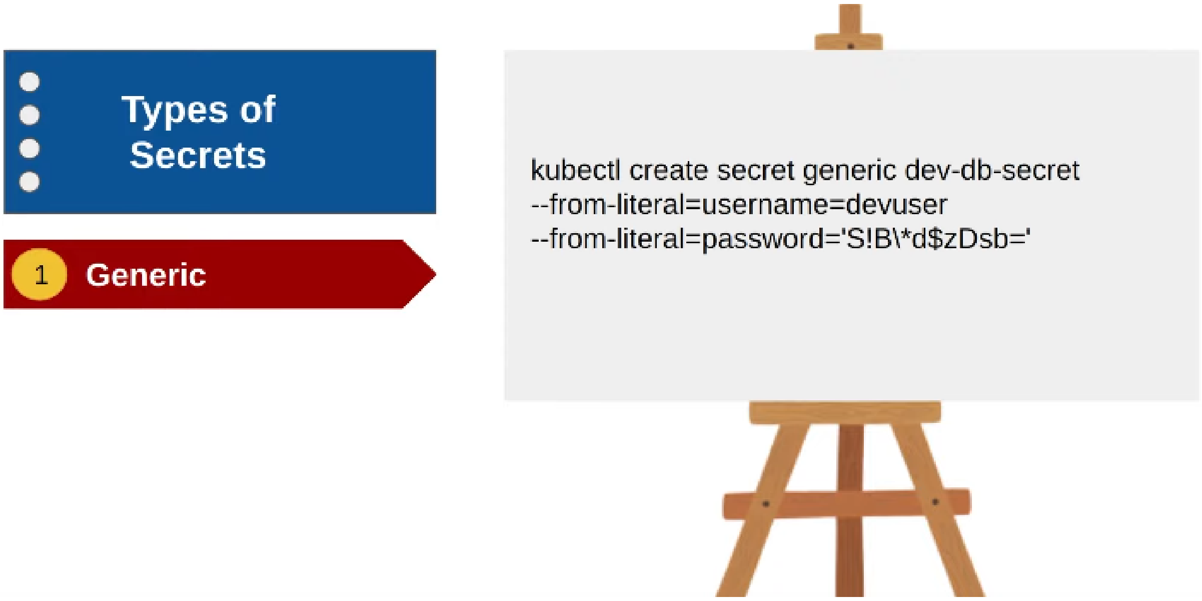
```
kubectl create secret <secret-type>
<secret-name> --from-literal=<key>=<value>
```

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

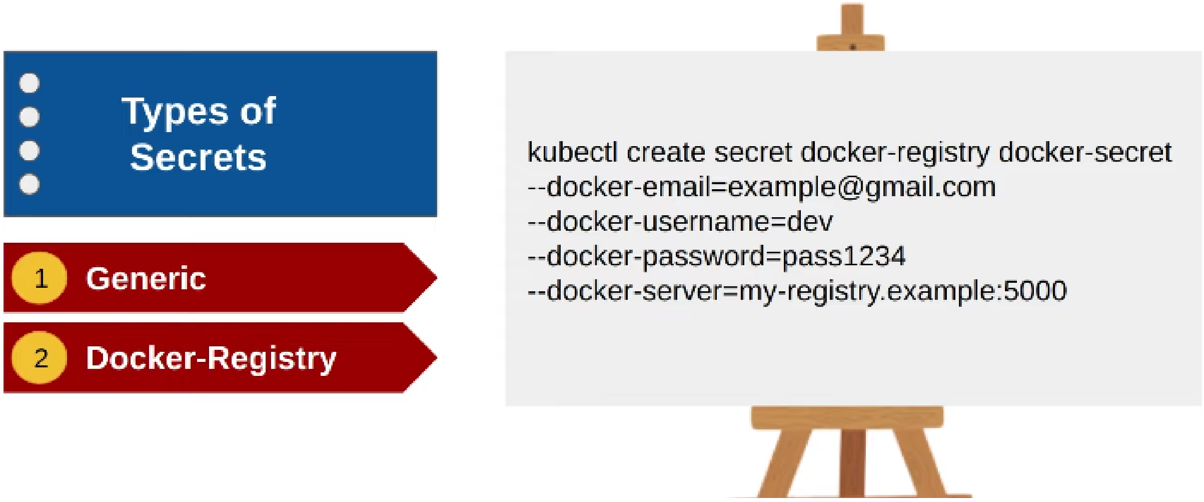
Declarative Way

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

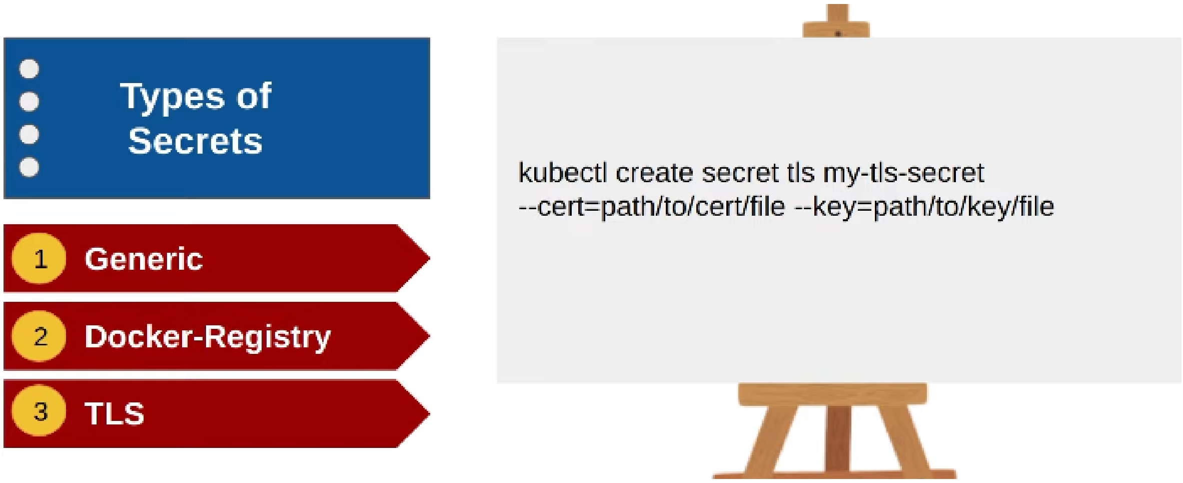
Types of Secret:



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



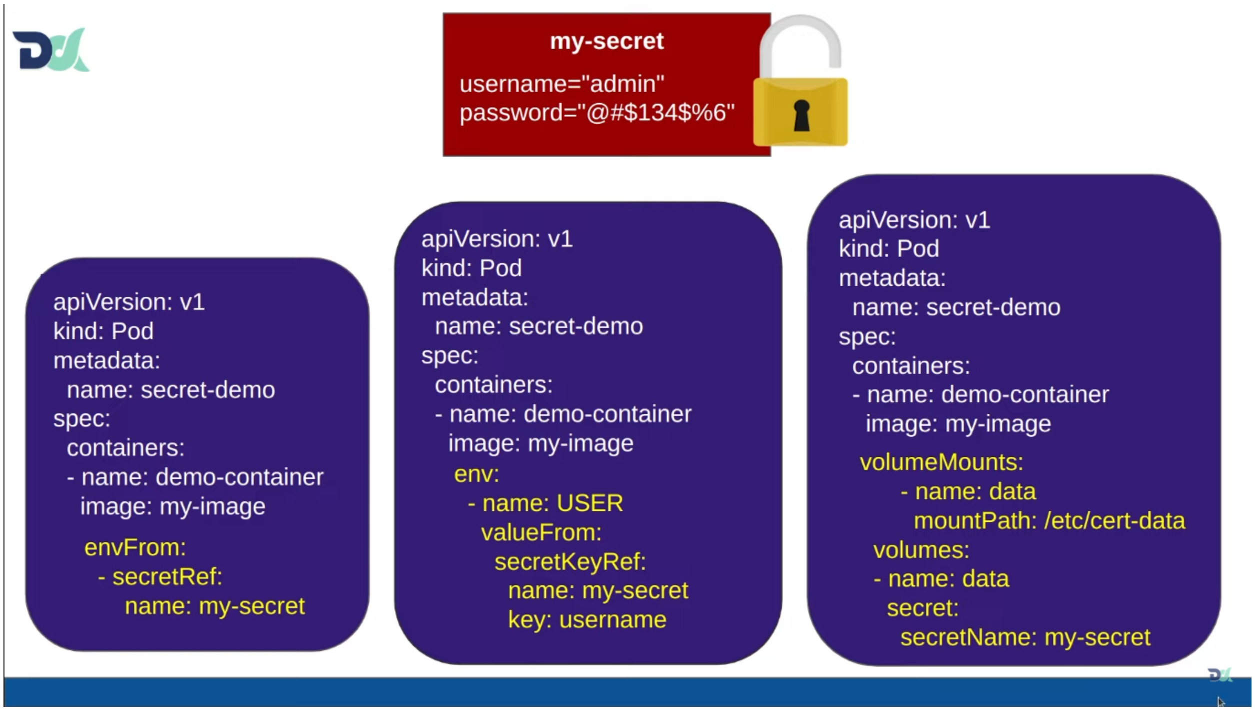
```
kubectl create secret docker-registry docker-secret
--docker-email=example@gmail.com
--docker-username=dev
--docker-password=pass1234
--docker-server=my-registry.example:5000
```



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

How to use secret in pods:

- Same like a configmap we can use a secret



my-secret

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

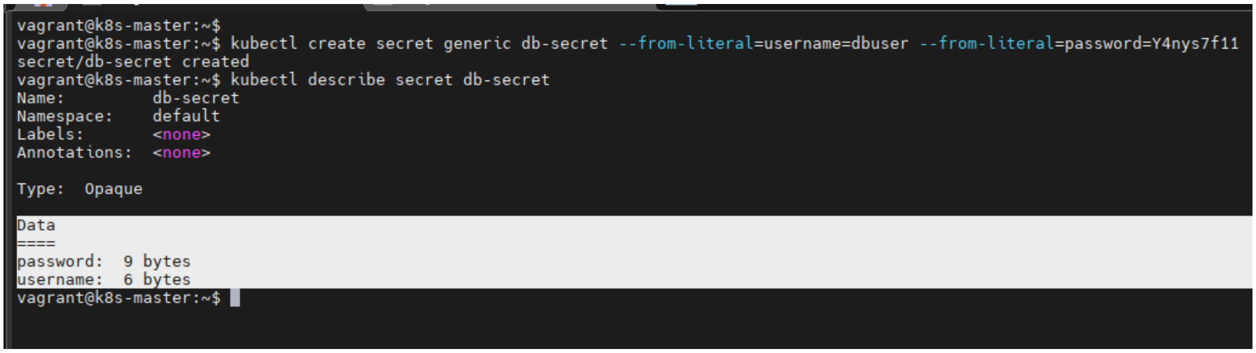
```
apiVersion: v1
kind: Pod
metadata:
  name: secret-demo
spec:
  containers:
  - name: demo-container
    image: my-image
     envFrom:
      - secretRef:
          name: my-secret
```

```
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
```

Generic Secret:

kubectl create secret generic db-secret --from-literal=username=dbuser --from-literal=password=Y4nys7f11



```
vagrant@k8s-master:~$
vagrant@k8s-master:~$ kubectl create secret generic db-secret --from-literal=username=dbuser --from-literal=password=Y4nys7f11
secret/db-secret created
vagrant@k8s-master:~$ kubectl describe secret db-secret
Name:         db-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
password:  9 bytes
username:  6 bytes
vagrant@k8s-master:~$
```

As you can see from above thing, we dont get anything what is inside of that secret instead we can only see how much of data it has. In configmap case , we get all the info store in configmap object but this is not the case in secret.

Get specific value from secret:

- vi specific-val.yml

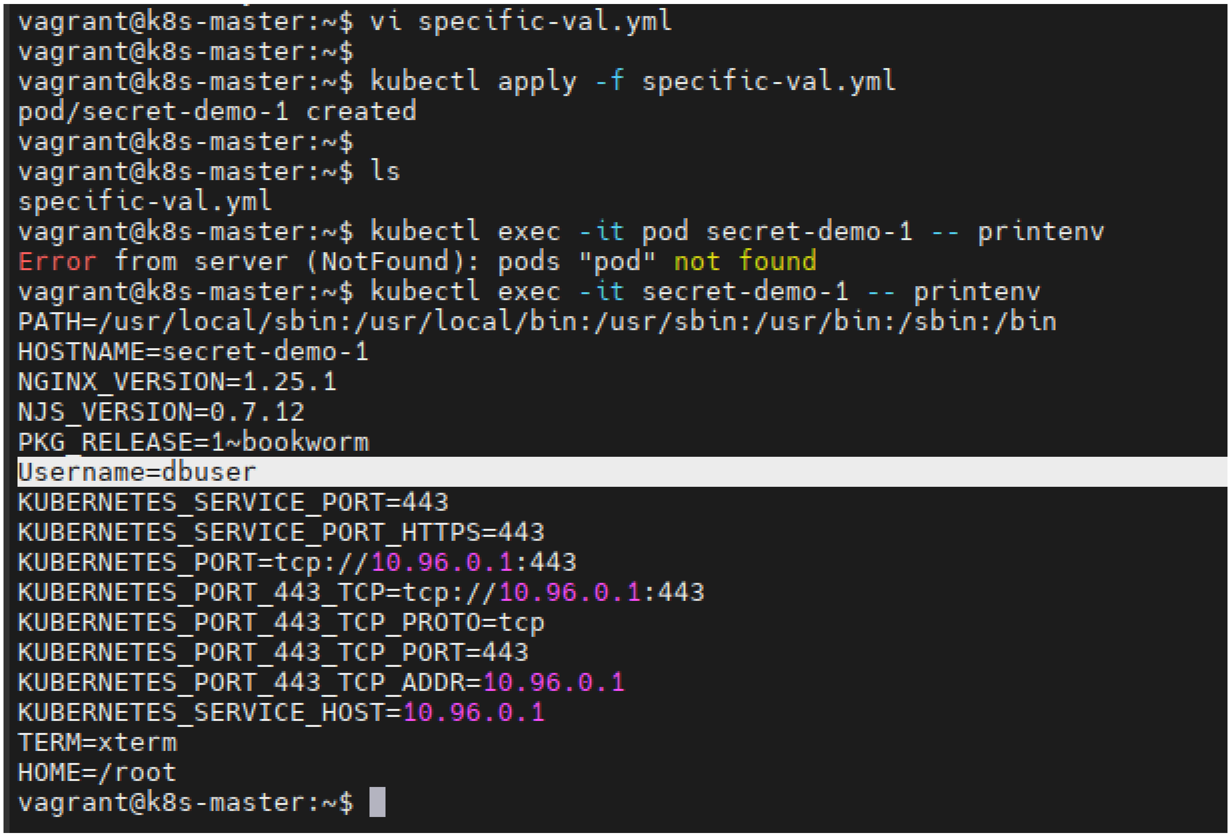
```
 apiVersion: v1
kind: Pod
metadata:
  name: secret-demo-1
spec:
  containers:
  - name: demo-container
    image: nginx
    env:
    - name: Username
      valueFrom:
        secretKeyRef:
          name: db-secret
          key: username
```

- Save , exit , apply

Check:

Print env variables :

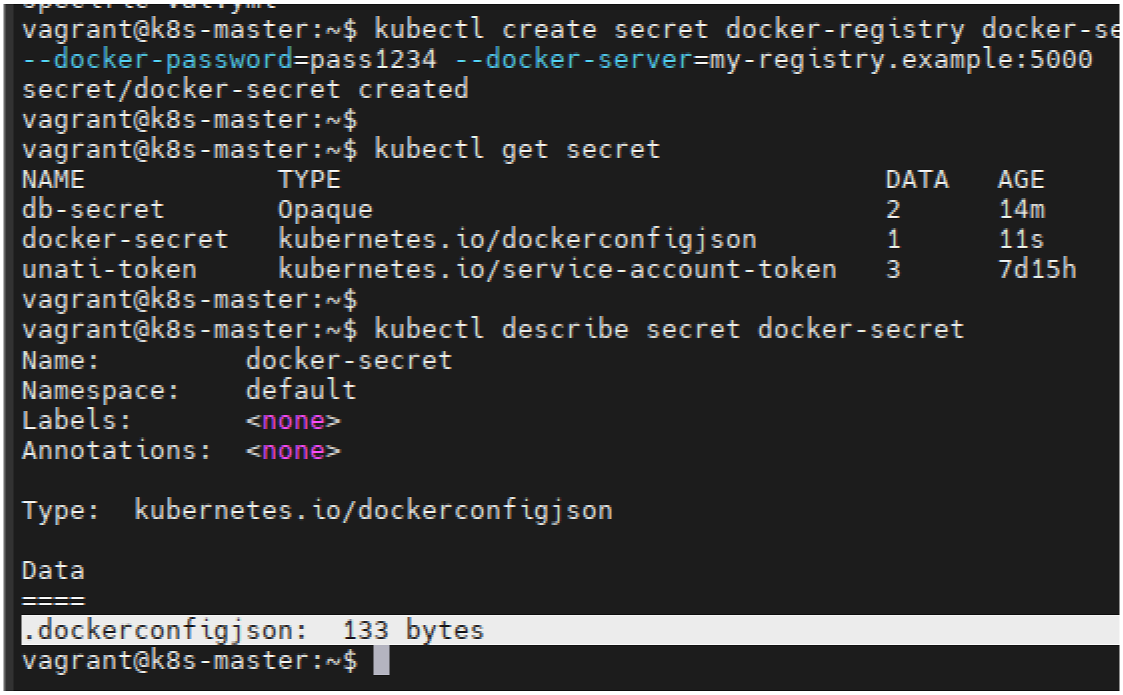
kubectl exec -it pod secret-demo-1 -- printenv



```
vagrant@k8s-master:~$ vi specific-val.yml
vagrant@k8s-master:~$
vagrant@k8s-master:~$ kubectl apply -f specific-val.yml
pod/secret-demo-1 created
vagrant@k8s-master:~$
vagrant@k8s-master:~$ ls
specific-val.yml
vagrant@k8s-master:~$ kubectl exec -it pod secret-demo-1 -- printenv
Error from server (NotFound): pods "pod" not found
vagrant@k8s-master:~$ kubectl exec -it secret-demo-1 -- printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=secret-demo-1
NGINX_VERSION=1.25.1
NJS_VERSION=0.7.12
PKG_RELEASE=1~bookworm
Username=dbuser
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
TERM=xterm
HOME=/root
vagrant@k8s-master:~$
```

Docker-registry Secret :

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



```
vagrant@k8s-master:~$ kubectl create secret docker-registry docker-se
--docker-password=pass1234 --docker-server=my-registry.example:5000
secret/docker-secret created
vagrant@k8s-master:~$
vagrant@k8s-master:~$ kubectl get secret
NAME            TYPE                                  DATA   AGE
db-secret       Opaque                                2      14m
docker-secret   kubernetes.io/dockerconfigjson        1      11s
unati-token     kubernetes.io/service-account-token   3      7d15h
vagrant@k8s-master:~$
vagrant@k8s-master:~$ kubectl describe secret docker-secret
Name:         docker-secret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  kubernetes.io/dockerconfigjson

Data
====
.dockerconfigjson:  133 bytes
vagrant@k8s-master:~$
```

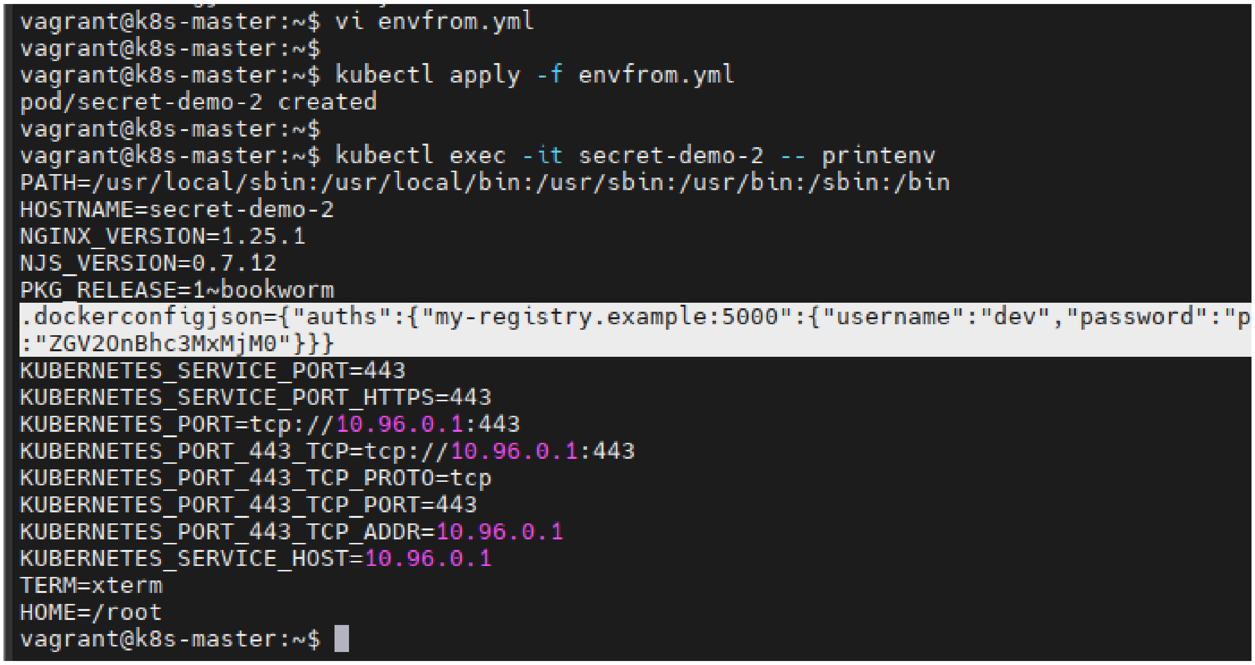
All info encoded and store in .dockerconfigjson

- vi envfrom.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-demo-2
spec:
  containers:
  - name: demo-container
    image: nginx
    envFrom:
    - secretRef:
        name: docker-secret
```

- Save , exit , apply

Check:



```
vagrant@k8s-master:~$ vi envfrom.yml
vagrant@k8s-master:~$
vagrant@k8s-master:~$ kubectl apply -f envfrom.yml
pod/secret-demo-2 created
vagrant@k8s-master:~$
vagrant@k8s-master:~$ kubectl exec -it secret-demo-2 -- printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=secret-demo-2
NGINX_VERSION=1.25.1
NJS_VERSION=0.7.12
PKG_RELEASE=1~bookworm
.dockerconfigjson={"auths":{"my-registry.example:5000":{"username":"dev","password":"p
:"ZGV2OnBhc3MxMjM0"}}}
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
TERM=xterm
HOME=/root
vagrant@k8s-master:~$
```

Declarative way secret :

For generic secret:

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