CONFIGMAP

About Configmap:-

we can create a common environment variables by using configmap and can be included while creating a pod in kubernetes. basically, configmap and secrets are almost same but difference configmap is unencrypted and secrets is encrypted. secrets will have sensitive data whereas we can store environment variables in config map and can be included in mainfest yml file while creating pod.
we can save the information on configmap and its dynamic. when the changes made or added new info on configmap, it will update to pods which are attached to configmap.
for example, planning to setup the sql server means, we can have all configuration files and information on configmap and while creating sql pod, we can include configmap. so here kubelet is take care of applying the changes to all pods when configmap got updated with new information.

In three way we can create configmap.
1, we can put a data into file and can create configmap.
2, another option called literal, we can key pair value in command line.
3, mainfest yml method we can create configmap.

Actual readme:-

1) ConfigMaps are useful for storing and sharing non-sensitive, unencrypted configuration information.
2) ConfigMaps bind configuration files, command-line arguments, environment variables, port numbers, and other configuration artifacts to your Pods' containers and system components at runtime
3) ConfigMaps allow you to separate your configurations from your Pods and components, which helps keep your workloads portable, makes their configurations easier to change and manage, and prevents hardcoding configuration data to Pod specifications

##Ceate a configmap by using file method##
We can put a data into file and then can create configmap.

kubectl api-resources |grep configmap

```
[root@ansikube manifest]# kubectl api-resources |grep configmap
configmaps                          cm                                          true         ConfigMap
[root@ansikube manifest]#
```

@Create username/password text file.
#echo -n 'user1' > ./username.txt && echo -n 'hello123' > ./password.txt

kubectl create configmap user-pass --from-file=username=username.txt --from-file=password=password.txt
kubectl get configmap

```
[root@ansikube manifest]# kubectl get configmap
NAME        DATA   AGE
user-pass   2      29s
```

kubectl describe configmap user-pass

```
[root@ansikube manifest]# kubectl describe configmap user-pass
Name:         user-pass
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
password:
----
ebs123

username:
----
user1
Events:  <none>
[root@ansikube manifest]#
```

##Create pod by including above configmap.

cat configmap2.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: mysql-configmap2
spec:
  containers:
    - name: con1
      image: mysql:5.6
      ports:
        - containerPort: 3306
      env:
        # Define the environment variable
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            configMapKeyRef:
            # The ConfigMap containing the value you want to assign to SPECIAL_LEVEL_KEY
               name: user-pass
            # Specify the key associated with the value
               key: password
```

kubectl apply -f configmap2.yml

kubectl get pods -o wide

```
[root@ansikube manifest]# kubectl get pods -o wide
NAME              READY   STATUS    RESTARTS   AGE   IP           NODE                                  NOMINATED NODE   READINESS GATES
mysql-configmap2  1/1     Running   0          86s   10.32.1.19   gke-robo-default-pool-fbd04527-jznk   <none>           <none>
[root@ansikube manifest]#
```

kubectl exec mysql-configmap2 env

```
[root@ansikube manifest]# kubectl exec mysql-configmap2 env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=mysql-configmap2
MYSQL_ROOT_PASSWORD=hello123
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.36.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.36.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.36.0.1
KUBERNETES_SERVICE_HOST=10.36.0.1
KUBERNETES_SERVICE_PORT=443
GOSU_VERSION=1.7
MYSQL_MAJOR=5.6
MYSQL_VERSION=5.6.46-1debian9
HOME=/root
[root@ansikube manifest]#
```

Note: - above command env output shows that configmap info's are present in this pod.

##Create a configmap by using literal method##

Another option called literal, where we can pair the key value in command line.

kubectl create configmap db-user-pass --from-literal=username=admin --from-literal=password=admin123

kubectl get configmap

```
[root@ansikube manifest]# kubectl create configmap db-user-pass --from-literal=username=admin --from-literal=password=admin123
configmap/db-user-pass created
[root@ansikube manifest]# kubectl get configmap
NAME           DATA   AGE
db-user-pass   2      28s
user-pass      2      20m
[root@ansikube manifest]#
```

kubectl describe configmap db-user-pass

```
[root@ansikube manifest]# kubectl describe configmap db-user-pass
Name:         db-user-pass
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
password:
----
admin123
username:
----
admin
Events:  <none>
[root@ansikube manifest]#
```

@@ create a pod with db-user-pass configmap.

cat configmap1.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: mysql-configmap1
spec:
  containers:
    - name: conl
      image: mysql:5.6
      ports:
        - containerPort: 3306
      env:
        # Define the environment variable
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            configMapKeyRef:
            # The ConfigMap containing the value you want to assign to SPECIAL_LEVEL_KEY
              name: db-user-pass
            # Specify the key associated with the value
              key: password
```

kubectl apply -f configmap1.yml
kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME               READY   STATUS    RESTARTS   AGE
mysql-configmap1   1/1     Running   0          10s
mysql-configmap2   1/1     Running   0          20m
[root@ansikube manifest]#
```

kubectl exec mysql-configmap1 env

```
[root@ansikube manifest]# kubectl exec mysql-configmap1 env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=mysql-configmap1
MYSQL_ROOT_PASSWORD=admin123
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.36.0.1
KUBERNETES_SERVICE_HOST=10.36.0.1
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.36.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.36.0.1:443
GOSU_VERSION=1.7
MYSQL_MAJOR=5.6
MYSQL_VERSION=5.6.46-1debian9
HOME=/root
[root@ansikube manifest]#
```

##How to create configmap with manifest yml##

Mainfest yml method will be used ceate configmap.

cat configmap.yml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: special-config
  namespace: default
data:
  user: user1
  pass: ebs123
  rootpass: bs123
---
apiVersion: v1
kind: Pod
metadata:
  name: mysql-configmap3
spec:
  containers:
    - name: con2
      image: mysql:5.6
      ports:
        - containerPort: 3307
      env:
        # Define the environment variable
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            configMapKeyRef:
              # The ConfigMap containing the value you want to assign to SPECIAL_LEVEL_KEY
                  name: special-config
              # Specify the key associated with the value
                  key: pass
```

kubectl apply -f configmap.yml

kubectl get configmap

```
[root@ansikube manifest]# kubectl get configmap
NAME             DATA   AGE
db-user-pass     2      21m
special-config   3      74s
user-pass        2      41m
[root@ansikube manifest]#
```

kubectl get pods

```
[root@ansikube manifest]# kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
mysql-configmap1  1/1     Running   0          13m
mysql-configmap2  1/1     Running   0          33m
mysql-configmap3  1/1     Running   0          2m25s
[root@ansikube manifest]#
```

kubectl exec mysql-configmap3 env

```
[root@ansikube manifest]# kubectl exec mysql-configmap3 env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=mysql-configmap3
MYSQL_ROOT_PASSWORD=ebs123
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.36.0.1
KUBERNETES_SERVICE_HOST=10.36.0.1
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.36.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.36.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
GOSU_VERSION=1.7
MYSQL_MAJOR=5.6
MYSQL_VERSION=5.6.46-1debian9
HOME=/root
[root@ansikube manifest]#
```

Note: - with single manifest yml, both configmap and pod has been created.