

KUBERNETES CONFIGMAP

Create a Pod that uses the ConfigMap to set an environment variable.

Create yml file for ConfigMap

\$ vi mycon.yml

\$ kubectl create -f mycon.yml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mycon
data:
  key: spvp
~
```

```
controlplane $ k get configmaps
NAME          DATA   AGE
kube-root-ca.crt  1      18d
mycon          1      31m
controlplane $
```

Create yml file for pod

\$ vi mypod.yml

\$ kubectl create -f mysqlpod.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: mysqlpod
spec:
  containers:
  - name: mysqlcon
    image: mysql
    env:
    - name: MYSQL_ROOT_PASSWORD
      valueFrom:
        configMapKeyRef:
          name: mycon
          key: key
~
```

```
controlplane $ k get pod
NAME          READY   STATUS    RESTARTS   AGE
mysqlpod      1/1     Running   0           4m38s
controlplane $ vi mypod.yml
```

\$ kubectl exec -it mysqlpod -- bash

```
root@master:~# kubectl exec -it mysqlpod -- bash
bash-5.1# mysql -u r
```

bash5.1# mysql -u root -psvpv

```
controlplane $ k exec -it mysqlpod -- bash
bash-5.1# mysql -u root -psvpv
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

Update the ConfigMap and verify that the Pod is updated.

\$ kubectl edit configmap mycon

```
controlplane $ k get configmaps
NAME          DATA   AGE
kube-root-ca.crt    1       18d
mycon          1       35m
controlplane $ k edit configmaps mycon
configmap/mycon edited
```

```

# Please edit the object below. Lines beginning with
# and an empty file will abort the edit. If a
# reopened with the relevant failures.
#
apiVersion: v1
data:
  key: spvp@1234
kind: ConfigMap
metadata:
  creationTimestamp: "2024-11-24T18:19:26Z"
  name: mycon
  namespace: default
  resourceVersion: "2742"
  uid: d4185509-b549-45bf-9386-c0d03db35384

```

Delete the old pod and then create a new pod

```

controlplane $ k delete pod mysqlpod
pod "mysqlpod" deleted
controlplane $ k create -f mypod.yml
pod/mysqlpod created

```

\$ kubectl exec -it mysqlpod -- bash

```

root@master:~# kubectl exec -it mysqlpod -- bash
bash-5.1# mysql -u r

```

```

controlplane $ k exec -it mysqlpod -- bash
bash-5.1# mysql -u root -psvp@1234
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)

```

Lab 2

From a file: Let's say you have a configuration file named file:

\$ vi file

controlplane \$ k create configmap mycon --from-file file

configmap/mycon created

Now I view the created config map into the yml file

controlplane \$ k get configmaps mycon -o yaml > my.yml

```
apiVersion: v1
data:
  file: |
    key1=value1
    key2=value2
kind: ConfigMap
metadata:
  creationTimestamp: "2024-11-25T05:05:56Z"
  name: mycon
  namespace: default
  resourceVersion: "2878"
  uid: 73ae22b3-30fc-4466-95de-06f2c4e5df6d
~
~
```

By using this command we can also create an multiple files also

kubectl create configmap my-configmap --from-file=app.properties --from-file=database.properties

Now we create an config map without create an file we use directly literal value.

kubectl create configmap my-configmap --from-literal=key1=value1 --from-literal=key2=value2

```
controlplane $ kubectl create configmap my-configmap --from-literal=key1=value1 --from-literal=key2=value2
configmap/my-configmap created
controlplane $ k get configmaps
NAME          DATA   AGE
kube-root-ca.crt  1      18d
my-configmap    2       14s
mycon          1       10m
```

Now I view the config map into the yml file.

controlplane \$ k get configmaps my-configmap -o yaml > myconf.yml

controlplane \$ vi myconf.yml

```

apiVersion: v1
data:
  key1: value1
  key2: value2
kind: ConfigMap
metadata:
  creationTimestamp: "2024-11-25T05:16:05Z"
  name: my-configmap
  namespace: default
  resourceVersion: "3732"
  uid: e9bbfad9-022f-48c9-82ce-8fb892d7fcfe
~

```

Using a combination: Combine files and literals:

kubectl create configmap my-configmap --from-file=file --from-literal=extra-key=extra-value

```

controlplane $ kubectl create configmap my-configmap1 --from-file=file --from-literal=extra-key=extra-value
configmap/my-configmap1 created
controlplane $ k get configmaps my-configmap1 -o yaml > myconf1.yaml
controlplane $ vi myconf1.yaml

```

```

apiVersion: v1
data:
  extra-key: extra-value
  file: |
    key1=value1
    key2=value2
kind: ConfigMap
metadata:
  creationTimestamp: "2024-11-25T05:21:24Z"
  name: my-configmap1
  namespace: default
  resourceVersion: "4178"
  uid: a5de9707-543a-49db-91f7-b8dd9fb40d32
~

```