

Config Maps and Secrets

- [Refer Here](#) for official docs of config maps

Activity: Config map for mysql details and use it in mysql container

- [Refer Here](#) for the config map written and apply to the k8s cluster

```
controlplane $ kubectl apply -f nop-cm.yaml
configmap/nop-cm created
controlplane $ kubectl describe configmaps nop-cm
Name:          nop-cm
Namespace:     default
Labels:        <none>
Annotations:   <none>

Data
====
MYSQL_DATABASE:
----
nop
MYSQL_PASSWORD:
----
nop123
MYSQL_ROOT_PASSWORD:
----
admin123
MYSQL_USER:
----
nop

BinaryData
====

Events:  <none>
controlplane $
```

- Now let's mount this configmap as a folder in k8s pod. [Refer Here](#) for the pod spec

```

Editor  Tab1  +
controlplane $ kubectl apply -f filemount.yaml
pod/filemountpod created
controlplane $ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
filemountpod   1/1     Running   0           8s
controlplane $ kubectl exec -it filemountpod -- /bin/sh
/ # cd /config/
/config # ls
MYSQL_DATABASE  MYSQL_PASSWORD  MYSQL_ROOT_PASSWORD  MYSQL_USER
/config # ls -al
total 12
drwxrwxrwx    3 root    root          4096 Jun 21 02:21 .
drwxr-xr-x    1 root    root          4096 Jun 21 02:21 ..
drwxr-xr-x    2 root    root          4096 Jun 21 02:21 ..2024_06_21_02_21_23.3845679748
lrwxrwxrwx    1 root    root           32 Jun 21 02:21 ..data -> ..2024_06_21_02_21_23.3845679748
lrwxrwxrwx    1 root    root          21 Jun 21 02:21 MYSQL_DATABASE -> ..data/MYSQL_DATABASE
lrwxrwxrwx    1 root    root          21 Jun 21 02:21 MYSQL_PASSWORD -> ..data/MYSQL_PASSWORD
lrwxrwxrwx    1 root    root          26 Jun 21 02:21 MYSQL_ROOT_PASSWORD -> ..data/MYSQL_ROOT_PASSWORD
lrwxrwxrwx    1 root    root          17 Jun 21 02:21 MYSQL_USER -> ..data/MYSQL_USER
/config # cat MYSQL_ROOT_PASSWORD
admin123/config #

```

- Now let's mount this configmap as environmental variables in a pod [Refer Here](#)

```

controlplane $ kubectl get cm
NAME          DATA   AGE
kube-root-ca.crt  1      12d
nop-cm         4      12m
controlplane $ kubectl apply -f envmount.yaml
pod/envmountpod created
controlplane $ kubectl exec -it envmountpod -- /bin/sh
/ # printenv
KUBERNETES_SERVICE_PORT=443
KUBERNETES_PORT=tcp://10.96.0.1:443
HOSTNAME=envmountpod
SHLVL=1
HOME=/root
MYSQL_ROOT_PASSWORD=admin123 ✓
TERM=xterm
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
MYSQL_PASSWORD=nop123 ✓
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
MYSQL_USER=nop ✓
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_SERVICE_HOST=10.96.0.1
PWD=/
TEST_OPTION_1=admin123
MYSQL_DATABASE=nop ✓
/ #

```

- now let's create a mysql pod with all values from config map loaded as environmental variables which are critical to start the mysql container

```
controlplane $ kubectl apply -f nopdb.yaml
pod/nopdb created
controlplane $ kubectl get po
NAME      READY   STATUS             RESTARTS   AGE
nopdb     0/1     ContainerCreating   0           5s
controlplane $ kubectl get po
NAME      READY   STATUS             RESTARTS   AGE
nopdb     0/1     ContainerCreating   0           8s
controlplane $ kubectl get po
NAME      READY   STATUS             RESTARTS   AGE
nopdb     0/1     ContainerCreating   0          11s
controlplane $ kubectl get po
NAME      READY   STATUS             RESTARTS   AGE
nopdb     0/1     ContainerCreating   0          13s
controlplane $ kubectl get po
NAME      READY   STATUS             RESTARTS   AGE
nopdb     1/1     Running            0          14s
controlplane $
```

```
controlplane $ kubectl exec -it nopdb -- mysql -u nop -p
mysql: [Warning] World-writable config file '/etc/mysql/my.cnf' is ignored.
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.37 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 |
| nop       |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql>
```

- [Refer Here](#) for the pod specification

- Secrets: [Refer Here](#) for official docs. [Refer Here](#) for the changes done

```

Editor  Tab 1  +
controlplane $ kubectl apply -f .
pod/envmountpod created
secret/nop-secret created
controlplane $ kubectl get secrets
NAME          TYPE      DATA   AGE
nop-secret    Opaque    4        9s
controlplane $ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
envmountpod   1/1     Running   0           13s
controlplane $ kubectl exec envmountpod -- printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=envmountpod
MYSQL_PASSWORD=nop123
MYSQL_ROOT_PASSWORD=admin123
MYSQL_USER=nop
MYSQL_DATABASE=nop
KUBERNETES_SERVICE_HOST=10.96.0.1
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
HOME=/root
controlplane $ 

```

Namespace

- [Refer Here](#) for official docs

Scopes in kubernetes

- In k8s we have two types of scopes
 - cluster wide scope

- Namespace wide scope

Editor Tab1 + 51 mi

```
controlplane $ kubectl api-resources
```

NAME	SHORTNAMES	APIVERSION	NAMESPACED	KIND
bindings		v1	true	Binding
componentstatuses	cs	v1	false	ComponentStatus
configmaps	cm	v1	true	ConfigMap
endpoints	ep	v1	true	Endpoints
events	ev	v1	true	Event
limitranges	limits	v1	true	LimitRange
namespaces	ns	v1	false	Namespace
nodes	no	v1	false	Node
persistentvolumeclaims	pvc	v1	true	PersistentVolumeClaim
persistentvolumes	pvc	v1	false	PersistentVolume
pod	po	v1	true	Pod
podtemplates		v1	true	PodTemplate
replicationcontrollers	rc	v1	true	ReplicationController
resourcequotas	quota	v1	true	ResourceQuota
secrets		v1	true	Secret
serviceaccounts	sa	v1	true	ServiceAccount
services	svc	v1	true	Service
mutatingwebhookconfigurations		admissionregistration.k8s.io/v1	false	MutatingWebhookConfiguration
validatingadmissionpolicies		admissionregistration.k8s.io/v1	false	ValidatingAdmissionPolicy
validatingadmissionpolicybindings		admissionregistration.k8s.io/v1	false	ValidatingAdmissionPolicyBinding
validatingwebhookconfigurations		admissionregistration.k8s.io/v1	false	ValidatingWebhookConfiguration
customresourcedefinitions	crd,crds	apiextensions.k8s.io/v1	false	CustomResourceDefinition

- Namespaced true belongs to namespace scope and false belongs to cluster scope

Exercises

- try running mysql by passing environmental variables from secrets
- How to use kubectl command to always view the items from dev namespaces