

# Config LAB

07 January 2021 12:23

- ConfigMap is an API object used to store non-confidential data in key-value pairs
- Pods can consume ConfigMaps as environment variables, command-line arguments, or as configuration files in a volume.

## Creation of ConfigMaps and then inject in Pod as environment variable and Volume.

Two way to create Config Maps

### 1) Imperative

```
# kubectl create configmap app-config --from-literal=APP_COLOR=Blue --from-literal=APP_MOD=prod
```

```
[root@master ~]# kubectl create configmap app-config --from-literal=APP_COLOR=Blue --from-literal=APP_MOD=prod
configmap/app-config created
```

```
# kubectl get configmap
```

```
[root@master ~]# kubectl get configmap
NAME          DATA  AGE
app-config    2      68s
kube-root-ca.crt 1      3d12h
```

```
# kubectl describe configmap app-config
```

```
[root@master ~]# kubectl describe configmap app-config
Name:         app-config
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
APP_COLOR:
----
Blue
APP_MOD:
----
prod
Events:      <none>
```

```
[root@master ~]# cat appconfig.properties
APP_COLOR=red
APP_MOD=qa
```

```
# kubectl create configmap appconfig-file --from-file=appconfig.properties
```

```
[root@master ~]# kubectl create configmap appconfig-file --from-file=appconfig.properties
configmap/appconfig-file created
```

```
# kubectl get configmap
```

```
[root@master ~]# kubectl get configmap
NAME          DATA  AGE
appconfig-file 1      57s
kube-root-ca.crt 1      3d12h
```

```
# kubectl describe configmap appconfig-file
```

```
[root@master ~]# kubectl describe configmap appconfig-file
Name:         appconfig-file
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
appconfig.properties:
----
APP_COLOR=red
APP_MOD=qa
Events:      <none>
```

## 2) Declarative

```
[root@master ~]# cat configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-configmap
data:
  APP_COLOR: green
  APP_MOD: test
```

```
# kubectl create -f configmap.yaml
```

```
[root@master ~]# kubectl create -f configmap.yaml
configmap/my-configmap created
```

```
# kubectl get configmap
```

```
[root@master ~]# kubectl get configmap
NAME                DATA  AGE
appconfig-file      1      5m35s
kube-root-ca.crt    1      3d13h
my-configmap        2      53s
```

```
# kubectl describe configmap my-configmap
```

```
[root@master ~]# kubectl describe configmap my-configmap
Name:         my-configmap
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
APP_COLOR:
----
green
APP_MOD:
----
test
Events:      <none>
```

## Inject Config map in Pod as Environment variable

```
[root@master ~]# cat pod_configmap.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-configmap
spec:
  containers:
  - name: test-container
    image: k8s.gcr.io/busybox
    command: [ "/bin/sh", "-c", "env" ]
    envFrom:
    - configMapRef:
        name: my-configmap
  restartPolicy: Never
```

```
# kubectl get pods
```

```
[root@master ~]# kubectl get pods
NAME                READY  STATUS   RESTARTS  AGE
nginx-deploy-5fcd49fb86-cpc94  1/1    Running   0          7h30m
nginx-deploy-5fcd49fb86-ctrq8  1/1    Running   0          7h30m
nginx-deploy-5fcd49fb86-n8nws  1/1    Running   0          7h30m
nginx1               1/1    Running   0          7h33m
nginx2               1/1    Running   0          7h33m
pod-configmap        0/1    Completed 0          82s
```

```
# kubectl logs pod-configmap
```

```
[root@master ~]# kubectl logs pod-configmap
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=pod-configmap
SHLV=1
HOME=/root
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
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
APP_MOD=test
APP_COLOR=green
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT_HTTPS=443
PWD=/
KUBERNETES_SERVICE_HOST=10.96.0.1
```

## Populate a Volume with data stored in a ConfigMap

```
# vi configmap-multiplekeys.yml
apiVersion: v1
kind: ConfigMap
metadata:
  name: special-config
  namespace: default
data:
  SPECIAL_LEVEL: very
  SPECIAL_TYPE: charm
```

```
# kubectl create -f configmap-multiplekeys.yml
configmap/special-config created
```

```
[root@k8s-master Day-4]# cat pod-configmap-volume.yml
apiVersion: v1
kind: Pod
metadata:
  name: pod-configmap-volume
spec:
  containers:
    - name: test-container
      image: k8s.gcr.io/busybox
      command: [ "/bin/sh", "-c", "ls /etc/config/ ; cat /etc/config/*" ]
      volumeMounts:
        - name: config-volume
          mountPath: /etc/config
  volumes:
    - name: config-volume
      configMap:
        name: special-config
  restartPolicy: Never
```

```
# kubectl apply -f pod-configmap-volume.yml
```

```
[root@master ~]# kubectl apply -f pod-configmap-volume.yml
pod/configmap-vol created
```

```
# kubectl logs configmap-vol
```

```
[root@master ~]# kubectl logs configmap-vol
SPECIAL_LEVEL
SPECIAL_TYPE
```