# 1.1 ConfigMap 实验手册

## 1.1.1 创建 configmap

### 步骤 1 创建一个目录用于存放 configmap 和 secret 实验的文件

```
[root@k8s-master storagefile]# mkdir /labfile/configmap
[root@k8s-master storagefile]# cd /labfile/configmap
```

### 步骤 2 创建两个文件，用于保存配置文件。

```
[root@k8s-master configmap]# vim game.properties
```

```
enemies=aliens
lives=3
enemies.cheat=true
enemies.cheat.lever=noGoodRotten
```

```
[root@k8s-master configmap]# vim ui.properties
```

```
color.good=purple
color.bad=yello
allow.textmode=true
```

### 步骤 3 从这两个文件创建 configmap

```
[root@k8s-master configmap]# kubectl create configmap game-config --from-file=/labfile/configmap
```

```
configmap/game-config created
```

### 步骤 4 查看创建的 configmap

```
[root@k8s-master configmap]# kubectl get configmap
```

```
NAME             DATA   AGE
game-config        2     58s
```

查看 configmap 详细信息

```
[root@k8s-master configmap]# kubectl describe configmap game-config
```

```
Name:         game-config
Namespace:     default
Labels:       <none>
Annotations:   <none>

Data
====
game.properties:
----
enemies=aliens
lives=3
enemies.cheat=true
enemies.cheat.lever=noGoodRotten

ui.properties:
----
color.good=purple
color.bad=yello
allow.textmode=true

Events:  <none>
```

### 步骤 5 从文件创建 configmap

```
[root@k8s-master configmap]# kubectl create configmap game-config-3 --from-file=/labfile/configmap/game.properties --from-file=/labfile/configmap/ui.properties
```

```
configmap/game-config-3 created
```

### 步骤 6 查看 game-config-3 的详细信息，与 game-config 对比，内容一致。

```
[root@k8s-master configmap]# kubectl describe configmap game-config-3
```

```
Name:         game-config-3
Namespace:    default
Labels:       <none>
Annotations:  <none>

Data
====
game.properties:
----
enemies=aliens
lives=3
enemies.cheat=true
enemies.cheat.lever=noGoodRotten

ui.properties:
----
color.good=purple
color.bad=yello
allow.textmode=true

Events:  <none>
```

### 步骤 7 直接从参数创建 configmap

```
[root@k8s-master configmap]# kubectl create configmap special-config --from-literal=special.how=very --from-literal=special.type=charm
```

### 步骤 8 查看创建的 configmap 结果

```
[root@k8s-master configmap]# kubectl describe configmap special-config
```

```
Name:         special-config
Namespace:     default
Labels:       <none>
Annotations:  <none>

Data
====
special.how:
----
very
special.type:
----
charm
Events:  <none>
```

### 步骤 9 从 yaml 文件创建 configmap

```
[root@k8s-master configmap]# vim configmap2.yaml
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: specialconfig-2
data:
  key1: value1
  pro.property: |
    key2: value2
    key3: value3
```

### 步骤 10 创建 configmap 并查看结果。

```
[root@k8s-master configmap]# kubectl apply -f configmap2.yaml
```

查看创建的结果

```
[root@k8s-master configmap]# kubectl describe configmap specialconfig-2
```

```
Name:         specialconfig-2
Namespace:     default
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
```

```
                {"apiVersion":"v1","data":{"key1":"value1","pro.property":"key2: value2\nkey3: value3\n"},"kind":"ConfigMap","metadata":{"annotations":{},...

Data
====
key1:
----
value1
pro.property:
----
key2: value2
key3: value3

Events:  <none>
```

## 1.1.2 使用 configmap

### 步骤 1　创建 Pod 的 yaml 文件，引用 configmap 做为 env 参数。

```
[root@k8s-master configmap]# vim cmpod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: cm-test-pod
spec:
  containers:
    - name: cm-container
      image: busybox
      args: ["/bin/sh", "-c", "env"]
      env:
        - name: special-env
          valueFrom:
            configMapKeyRef:
              name: specialconfig-2
              key: key1
      envFrom:
        - configMapRef:
            name: specialconfig-2
  restartPolicy: Never
```

### 步骤 2　创建 pod

```
[root@k8s-master configmap]# kubectl apply -f cmpod.yaml
```

### 步骤 3　查看 pod 输出的日志

```
[root@k8s-master configmap]# kubectl logs cm-test-pod
```

```
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_SERVICE_PORT=443
HOSTNAME=cm-test-pod
SHLVL=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
HTTPD_SVC_PORT_8080_TCP_ADDR=10.105.253.35
HTTPD_SVC_SERVICE_HOST=10.105.253.35
KUBERNETES_PORT_443_TCP_PORT=443
key1=value1
KUBERNETES_PORT_443_TCP_PROTO=tcp
HTTPD_SVC_PORT_8080_TCP_PORT=8080
special-env=value1
HTTPD_SVC_PORT_8080_TCP_PROTO=tcp
HTTPD_SVC_PORT=tcp://10.105.253.35:8080
HTTPD_SVC_SERVICE_PORT=8080
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_SERVICE_HOST=10.96.0.1
PWD=/
pro.property=key2: value2
key3: value3

HTTPD_SVC_PORT_8080_TCP=tcp://10.105.253.35:8080
```

### 步骤 4 通过 volume 使用 configmap，创建 pod 的 yaml 文件

```
[root@k8s-master configmap]# vim cmpod2.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: cmpod2
spec:
  containers:
  - name: cmpod2
    image: busybox
    args: [ "/bin/sh", "-c", "sleep 3000" ]
    volumeMounts:
    - name: db
      mountPath: "/etc/db"
      readOnly: true
  volumes:
  - name: db
    configMap:
      name: specialconfig-2
```

### 步骤 5 创建 pod

```
[root@k8s-master configmap]# kubectl apply -f cmpod2.yaml
pod/cmpod2 created
```

### 步骤 6 进入 pod，查看 volume 内文件。

```
[root@k8s-master configmap]# kubectl exec -it cmpod2 /bin/sh
/ # cd /etc/db
/etc/db # ls
key1            pro.property
/etc/db # cat key1
value1

/etc/db # cat pro.property
key2: value2
key3: value3
exit 退出
```