

ConfigMap

Configmap ter permite configurar tus pods desde un fichero de configuración sin necesidad de hacerlo manualmente y facilitando su portabilidad.

Un configmap se define siempre con una llave y un valor. Además el Pod y el ConfigMap deben estar en el mismo Namespace.

Se puede hacer desde un fichero .conf o bien desde nuestro manifiesto

Por Ejemplo

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-config
  labels:
    app: front
data:
  #Definimos dos llaves una llamadas test y la
  otra nginx
  test: hola #Definimos la primera llave
  nginx: | #Definimos la segunda llave
    server {
      listen 9090;
      server_name localhost;
      location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
      }
      error_page 500 502 503 504 /50x.html;
      location = /50x.html {
        root /usr/share/nginx/html;
      }
    }
  ---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-test
  labels:
    app: front
spec:
  replicas: 1
  selector:
    matchLabels:
      app: front
  template:
    metadata:
      labels:
```

```

    app: front
  spec:
    containers:
      - name: nginx                                #nombre de la imagen
        image: nginx:alpine
        volumeMounts:
          - name: nginx-vol                        #nombre del volumen
            mountPath: /etc/nginx/conf.d
    volumes:
      - name: nginx-vol                            #Definimos como será el volumen y se
#tiene que llamar igual que el nombre que le pusimos anteriormente
        configMap:
          name: nginx-config
          items:
            - key: nginx
              path: default.conf

```

Ejemplo con variables y volúmenes

```

apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-config
  labels:
    app: front
data:
  nginx: |                                     #Definimos la primera llave de nuestro
ConfigMap
  server {
    listen      9090;
    server_name localhost;
    location / {
      root      /usr/share/nginx/html;
      index     index.html index.htm;
    }
    error_page  500 502 503 504  /50x.html;
    location = /50x.html {
      root      /usr/share/nginx/html;
    }
  }
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: vars
  labels:
    app: front
data:                                     #Definimos otro
configmap para las variables
  db_host: dev.host.local                #primera llave
  db_user: dev_user                      #segunda llave

```

```
script: | #Tercera llave
  echo DB host es $DB_HOST y DB user es $DB_USER >
/usr/share/nginx/html/test.html
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-test
  labels:
    app: front
spec:
  replicas: 1
  selector:
    matchLabels:
      app: front
  template:
    metadata:
      labels:
        app: front
    spec:
      containers:
        - name: nginx
          image: nginx:alpine
          env:
            - name: DB_HOST
              valueFrom:
                configMapKeyRef:
                  name: vars
                  key: db_host
            - name: DB_USER
              valueFrom:
                configMapKeyRef:
                  name: vars
                  key: db_user
          volumeMounts:
            - name: nginx-vol
              mountPath: /etc/nginx/conf.d
            - name: script-vol
              mountPath: /opt
      volumes:
        - name: nginx-vol
          configMap:
            name: nginx-config
            items:
              - key: nginx
                path: default.conf
        - name: script-vol
          configMap:
            name: vars
            items:
              - key: script
```

path: script.sh

Referencias

- <https://kubernetes.io/es/docs/concepts/configuration/configmap/>
- <https://github.com/ricardoandre97/k8s-resources/tree/master/configmaps>

From:

<http://wiki.intrusos.info/> - **LCWIKI**

Permanent link:

<http://wiki.intrusos.info/doku.php?id=virtualizacion:kubernetes:configmap>

Last update: **182023/01/ 13:37**

