ConfigMap in Kubernetes?

A ConfigMap is a dictionary of configuration settings. This dictionary consists of key-value pairs of strings. Kubernetes provides these values to your containers. Like with other dictionaries (maps, hashes, ...)

kind: ConfigMap

**What is a ConfigMap used for?**

A ConfigMap stores configuration settings for your code. Store connection strings, public credentials, hostnames, and URLs in your ConfigMap.

**How does a ConfigMap work?**

Here's a quick animation I made showing how a ConfigMap works in Kubernetes.

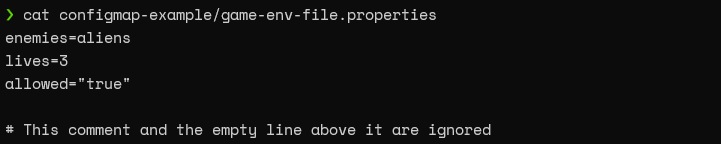
First, you have multiple ConfigMaps, one for each environment.

Second, a ConfigMap is created and added to the Kubernetes cluster.

Third, containers in the Pod reference the ConfigMap and use its values

### Creating ConfigMaps from an environment file

cat configmap-example/game-env-file.properties



kubectl create configmap game-config-env-file-example --from-env-file=configmap-example/game-env-file.properties

kubectl

### ConfigMap in a YAML file.

**apiVersion**: v1

**kind:** confifMap

**metadata:**

name: game-demo

**data:**

player\_initial\_lives:5

ui\_properties\_file\_name: "user-interface.properties"

game.properties:

enemy.types=aliens,monsters

player.maximum-lives=7

user-interface.properties:

color.good=green

color.bad=red

allow.textmode=true

kubectl create configmap game-data \  
    --from-file game-configs/

### #Setting POD for ConfigMap in a YAML file.

apiVersion: v1

kind: pod

metadata:

name: configMap\_game-demo-pod

spec:

containers:

- name: game-demo-container

image: alpine

env:

- name: PLAYER\_INITIAL\_LIVES

valueFrom:

configMapkeyRef:

name: game-demo

key: player\_initial\_lives

- name: UI\_PROPERTIES\_FILE\_NAME

valueFrom:

configMapkeyRef:

name: game-demo

key: ui\_properties\_file\_name

volumeMounts:

- name : config

mountPath: "/config"

readOnly: true

volumes:

- name: config

configMap:

name: game-demo

items:

- key: "game.properties"

path: "game.properties"

- key: "user-interface.properties"

path: "user-interface-properties"