

How do we make Containerize app Portable?



ConfigMaps

Concept

Objectives

Concept

Configuring Containerized Application

ConfigMaps

Creating Config Maps

Review Demo

Creating ConfigMaps from

- Directories
- Files
- Literal Values

Configuring Containerized Application

- Container images are build to be “portable”
- Containers expect configuration from
 - Configuration files
 - Command line arguments
 - Environment variables
- INI - XML - JSON - Custom Format

ConfigMaps

- Decouples configuration from pods and components
- Stores configuration data as Key-value pairs
 - Configuration files
 - Command line arguments
 - Environment variables
- Similar to Secrets but don't contain sensitive information
- You must create a ConfigMap before referencing it in a Pod spec

Creating ConfigMaps

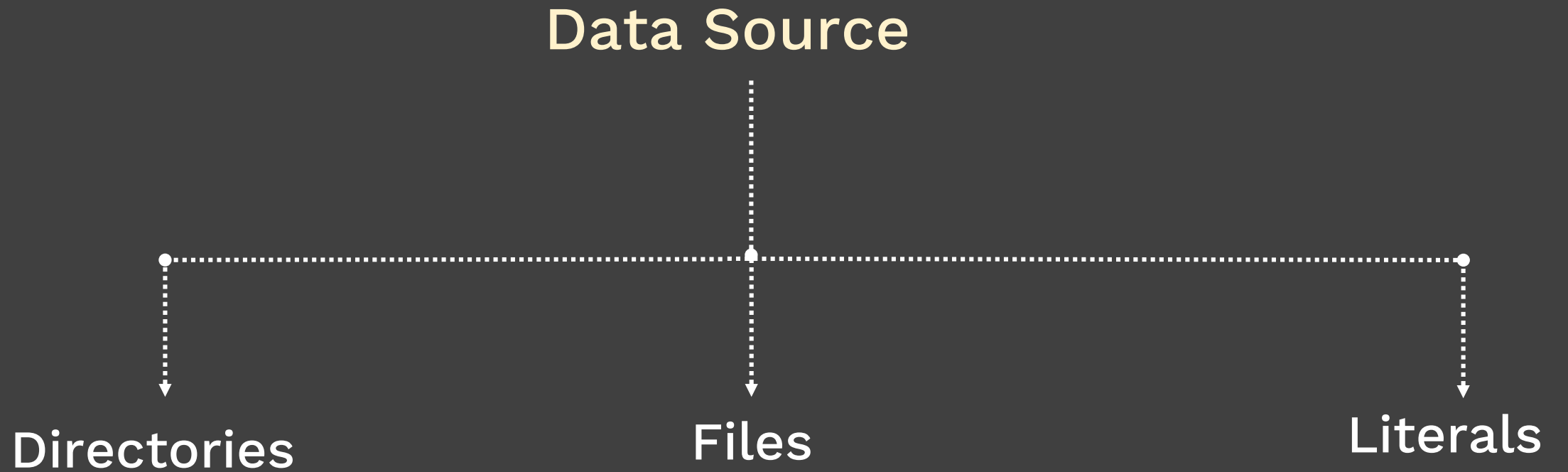
`kubectl create configmap <map-name> <data-source>`



The diagram shows the command `kubectl create configmap <map-name> <data-source>` with white brackets underneath each part. A white arrow points from the `<data-source>` bracket to the list of options below.

- Path to dir/file: `--from-file`
- Key-Value pair : `--from-literal`

Review Demo



Directories

ConfigMaps

Create ConfigMaps from directories

```
mkdir -p configure-pod-container/configmap/kubect1/
```

```
wget https://k8s.io/docs/tasks/configure-pod-container/configmap/kubect1/game.properties -O  
configure-pod-container/configmap/kubect1/game.properties
```

```
wget https://k8s.io/docs/tasks/configure-pod-container/configmap/kubect1/ui.properties -O  
configure-pod-container/configmap/kubect1/ui.properties
```

```
ls configure-pod-container/configmap/kubect1/  
game.properties  
ui.properties
```

```
[srinath@master ~]$ cat game.properties  
enemies=aliens  
lives=3  
enemies.cheat=true  
enemies.cheat.level=noGoodRotten  
secret.code.passphrase=UUDDLRLRBABAS  
secret.code.allowed=true  
secret.code.lives=30  
srinath@master ~]$
```

```
[srinath@master ~]$ cat ui.properties  
color.good=purple  
color.bad=yellow  
allow.textmode=true  
how.nice.to.look=fairlyNice  
[srinath@master ~]$
```

Create ConfigMaps from directories

```
srinath@master:$ kubectl create configmap game-config --from-file=configure-pod-container/configmap/kubectl/
```

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

```
srinath@master:$ kubectl get configmaps -o wide
```

NAME	DATA	AGE
game-config	2	1m

Create ConfigMaps from directories

```
srinath@master:$ kubectl get configmaps game-config -o yaml
apiVersion: v1
data:
  game.properties: |-
    enemies=aliens
    lives=3
    enemies.cheat=true
    enemies.cheat.level=noGoodRotten
    secret.code.passphrase=UUDDLRLRBABAS
    secret.code.allowed=true
    secret.code.lives=30
  ui.properties: |
    color.good=purple
    color.bad=yellow
    allow.textmode=true
    how.nice.to.look=fairlyNice
kind: ConfigMap
metadata:
  creationTimestamp: 2018-09-01T09:21:42Z
  name: game-config
  namespace: default
  resourceVersion: "598823"
  selfLink: /api/v1/namespaces/default/configmaps/game-config
  uid: 707531b3-adc8-11e8-b3de-42010a800003
```

Files

ConfigMaps

Creating ConfigMaps from-file

```
srinath@master:$ curl -OL https://k8s.io/examples/pods/config/redis-config
```

% Total	% Received	% Xferd	Average	Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left	Speed
100	185	100	185	0	0	239	0	--:--:-- --:--:-- --:--:-- 239
10043	100	43	0	0	29	0	0:00:01 0:00:01 --:--:-- 132	

```
srinath@master:$ cat redis-config
```

```
maxmemory 2mb
```

```
maxmemory-policy allkeys-lru
```

```
srinath@master:$ kubectl create configmap example-redis-config --from-file=redis-config
```

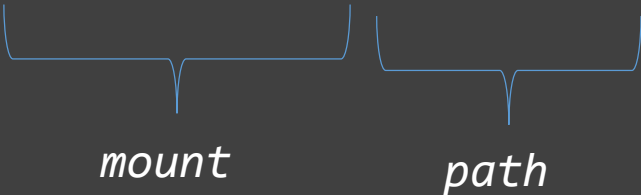
```
configmap/example-redis-config created
```

Accessing ConfigMaps in Pods

```
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: kubernetes/redis:v1
    volumeMounts:
    - mountPath: /redis-master
      name: config
  volumes:
  - name: config
    configMap:
      name: example-redis-config
      items:
      - key: redis-config
        path: redis.conf
```

Testing

```
srinath@master:$ kubectl exec redis cat /redis-master/redis.conf
```



maxmemory 2mb
maxmemory-policy allkeys-lru

mount *path*

```
srinath@master: kubectl exec -it redis redis-cli
127.0.0.1:6379> CONFIG GET maxmemory
1) "maxmemory"
2) "2097152"
127.0.0.1:6379> CONFIG GET maxmemory-policy
1) "maxmemory-policy"
2) "allkeys-lru"
```

Literals

ConfigMaps

Create ConfigMaps from literal values

```
srinath@master:$ kubectl create configmap special-config --from-literal=special.how=very  
configmap/special-config created
```

```
srinath@master:$ kubectl get configmaps
```

NAME	DATA	AGE
special-config	2	2m

Create ConfigMaps from literal values

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
  - name: test-container
    image: k8s.gcr.io/busybox
    command: [ "/bin/sh", "-c", "env" ]
    env:
      - name: SPECIAL_LEVEL_KEY
        valueFrom:
          configMapKeyRef:
            name: special-config
            key: special.how
  restartPolicy: Never
```

```
srinath@master:$ kubectl logs test-pod | grep SPECIAL
```

```
SPECIAL_LEVEL_KEY=very
```

Summary

Concept

Configuring Containerized Application

ConfigMaps

Creating Config Maps

Review Demo

Creating ConfigMaps from

- Directories
- Files
- Literal Values

Coming up...

Demo ConfigMaps