

VOLUMES AND DATA

Volumes and Data

Portable Data with ConfigMaps

A similar API resource to Secrets is the ConfigMap, except the data is not encoded. In keeping with the concept of decoupling in Kubernetes, using a ConfigMap decouples a container image from configuration artifacts.

They store data as sets of key-value pairs or plain configuration files in any format. The data can come from a collection of files or all files in a directory. It can also be populated from a literal value.

A ConfigMap can be used in several different ways. A container can use the data as environmental variables from one or more sources. The values contained inside can be passed to commands inside the pod. A Volume or a file in a Volume can be created, including different names and particular access modes. In addition, cluster components like controllers can use the data.

Let's say you have a file on your local filesystem called **config.js**. You can create a ConfigMap that contains this file. The **configmap** object will have a data section containing the content of the file (see the command and output below):

```
$ kubectl get configmap foofoo -o yaml
```

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: foofoo
data:
  config.js: |
    {
...

```

ConfigMaps can be consumed in various ways:

- Pod environmental variables from single or multiple ConfigMaps
- Use ConfigMap values in Pod commands
- Populate Volume from ConfigMap
- Add ConfigMap data to specific path in Volume
- Set file names and access mode in Volume from ConfigMap data
- Can be used by system components and controllers.