Injecting Configurations from Multiple Files

In this lesson, we will learn how to inject configuration from multiple files and from a directory.

WE'LL COVER THE FOLLOWING

Creating a ConfigMap from Multiple Files

Let's see what happens when we execute the commands that follow.

```
kubectl create cm my-config \
    --from-file=cm/prometheus-conf.yml \
    --from-file=cm/prometheus.yml

kubectl create -f cm/alpine.yml

#Run the following command separately
kubectl exec -it alpine -- \
    ls /etc/config
```

We created a ConfigMap with two files, and we created the same Pod based on the `alpine.yml` definition. Finally, we output the list of files from the `/etc/config` directory inside the Pod's only container. The **output** of the latter command is as follows.

```
prometheus-conf.yml  prometheus.yml
```

We can see that both files are present in the container. That leads us to the conclusion that a ConfigMap can **contain multiple files**, and all will be created inside containers that mount it.

Deleting the Objects

Let's delete the objects (again), and explore one more option behind the `--from-file` argument.

```
kubectl delete -f cm/alpine.yml
#Run the following command separately to delete the configmap
kubectl delete cm my-config
```

Creating a ConfigMap from a Directory

The `--from-file` argument might lead you to the conclusion that you can specify only a file path as its value. It works with directories as well. We can, for example, add all files from the `cm` directory to a ConfigMap.

```
kubectl create cm my-config \
    --from-file=cm
```

We created `my-config` ConfigMap with the directory `cm`. Let's describe it, and see what's inside.

```
kubectl describe cm my-config
```

The **output** is as follows (content of the files is removed for brevity).

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

Data
====
alpine-env-all.yml:
----
...
alpine-env.yml:
----
...
alpine.yml:
----
...
my-env-file.yml:
----
...
prometheus-conf.yml:
----
```

```
prometheus.yml:
----
...

Events:  <none>
```

We can see that all six files from the `cm` directory are now inside the `my-config` ConfigMap.

We're sure you already know what will happen if we create a Pod that mounts that ConfigMap. We'll check it out anyways.

```
kubectl create -f cm/alpine.yml
#Run the below command separately after the "alpine" container is created
kubectl exec -it alpine -- \
    ls /etc/config
```

The **output** of the latter command is as follows.

```
alpine-env-all.yml alpine.yml      prometheus-conf.yml
alpine-env.yml     my-env-file.yml prometheus.yml
```

All the files are there, and the time has come to move away from files and directories.

Deleting the Objects

So, let's remove the objects first, and discuss the other sources.

```
kubectl delete -f cm/alpine.yml
#Run the following command separately to delete the configmap
kubectl delete cm my-config
```

Now that we are done with creating ConfigMaps from files and directories. In the next lesson, we will learn to inject configurations from key/value literals.