## Injecting Configurations from Environment Files

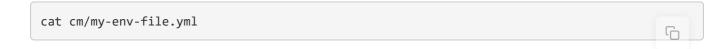
In this lesson, we will use environment files to inject configurations.

## WE'LL COVER THE FOLLOWING ^

- Looking into the Definition
- Creating the ConfigMap

## Looking into the Definition #

Let's take a look at the cm/my-env-file.yml file.



The **output** is as follows.

```
something=else
weather=sunny
```

The file has the same key/value pairs as those we used in the example with --from-literal.

## Creating the ConfigMap #

Let's see what happens if we create a ConfigMap using that file as the source.

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

kubectl get cm my-config -o yaml
```

We created the ConfigMap using the --from-env-file argument, and we retrieved the ConfigMap in yaml format.

The **output** of the latter command is as follows (metadata is removed for brevity).

```
apiVersion: v1
data:
    something: else
    weather: sunny
kind: ConfigMap
...
```

We can see that there are two entries, each corresponding to key/value pairs from the file. The result is the same as when we created a ConfigMap using -from-literal arguments. Two different sources produced the same outcome.

If we used --from-file argument, the result would be as follows.

```
apiVersion: v1
data:
  my-env-file.yml: |
    something=else
    weather=sunny
kind: ConfigMap
...
```

All in all, --from-file reads the content of one or more files, and stores it using file names as keys. --from-env-file, assumes that content of a file is in key/value format, and stores each as a separate entry.

In the next lesson, we will explore how to convert the output of configMap into environment variables.