

**Environment Variables**

When building your application stack to work on [Kubernetes](https://akomljen.com/tag/kubernetes/), the basic pod configuration is usually done by setting different environment variables. Sometimes you want to configure just a few of them for a particular pod, or to define a set of environment variables that can be shared by multiple pods.

You can set environment variables for containers running in a pod and in addition, Kubernetes exposes certain runtime infos via environment variables automatically.

## Define an environment variable for a container

When you create a Pod, you can set environment variables for the containers that run in the Pod. To set environment variables, include the env or envFrom field in the configuration file.

## set Environment Variables

To set the environment variables you can use env or envFrom key in the configuration file. A most basic option is to only set one or more of them using the simple key:value syntax:

spec:

containers:

- env:

- name: VARIABLE1

value: test1

It looks fine, but imagine a ten or more variables per Pod configuration. Isn't it better to have a separate file for that? You really want to use config map to set more environment variables at once. Here is one example:

apiVersion: v1

kind: ConfigMap

metadata:

name: my-env

data:

VARIABLE1: test1

VARIABLE2: test2

VARIABLE3: test3

With above configuration it is easy for multiple containers to share the same set of environment variables. Then we just need to reference the config map file with configMapRef (available from Kubernetes v1.6):

spec:

containers:

- envFrom:

- configMapRef:

name: my-env

In case you don't want for each container to have all environment variables from config map available, you can load only specific keys. For example, to only load VARIABLE1:

spec:

containers:

- env:

- name: VARIABLE1

valueFrom:

configMapKeyRef:

name: my-env

key: VARIABLE1