A Plea NOT to Use ConfigMaps!

In this lesson, we will try to find out the best way to configure our applications.

WE'LL COVER THE FOLLOWING

Why No ConfigMaps?

ConfigMaps, in our experience, are overused.

If you have a configuration that is the same across multiple clusters, or if you have only one cluster, all you should do is include it in your Dockerfile and forget it ever existed. When there are no variations of a config, there's no need to have a configuration file. At least, not outside an immutable image.

Unfortunately, that is not always the case. To be more precise, it's almost never the case. We tend to make things more complicated than they should be. That, among other things, often means an endless list configuration options hardly anyone ever uses. Still, some things usually do change, from one cluster to another, and we might need to look into alternatives to configurations baked into images.

Finding out the Best Way

Design your new applications to use a combination of configuration files and environment variables. Make sure that the default values in a configuration file are sensible and applicable in most use-cases. Bake it into the image. When running a container, declare only the environment variables that represent the differences of a specific cluster. That way, your configuration will be portable and simple at the same time.

What if your application is not new and it does not support configuration through environment variables? Refactor it so that it does. It shouldn't be hard to add the ability to read a few environment variables. Keep in mind that you don't need all the settings, but only those that differ from one cluster to another. It would be hard to imagine that such a trivial request would be complex or time-consuming. If it is, you might have more significant issues to fix before even thinking about putting your application into a container.

When to Use ConfigMaps?

Still, configuration files will not disappear. No matter which strategy we choose, each image should have a copy of them with sensible default values. Maybe, we can put in an extra effort and change the application, so that configuration entries are loaded from two locations. That way, we can load the default values from one, and only the differences from the other. That would, at least, reduce the need to have to specify more than the minimum required for each cluster. In such a case, ConfigMap's `--from-literal` and `--from-env-file` sources are an excellent choice.

When everything else fails, the `--from-file` source is your friend. Just make sure that ConfigMap is not defined in the same file as the objects that mount it. If it is, it would mean that they could be used only inside one cluster. Otherwise, we'd be deploying the same config, and we should go back to the initial idea of having it baked into the image together with the application.

Do not let this pessimism discourage you from using ConfigMaps. They are very handy, and you should adopt them. Our intent to discourage you from doing so had the intention of making you think of alternatives, not to tell you never to use ConfigMaps.

In the next lesson, we will test your understanding of ConfigMaps with the help of a quick quiz.