

Dealing with Configuration Data

Use Config maps to send dynamic values into deployment

Applications require a way for us to pass data to them that can be changed at deploy time. Examples of this might be log levels or URLs of external systems that the application might need at start-up time. Instead of hard-coding these values, we use configmaps in Kubernetes to pass these values as environment variables to the container.

I have a deployment where I have a container called karthequian reader and, as an environment variable, I want to pass log level of error. Let's take a look at the logs. We'll notice that this says that the log level passed via environment variable was error as per the application. What we'd like to do is we'd like to change this at deployment time.

```
PS C:\Users\msuban01\Desktop\kube\Exercise\05_03_Configmaps> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
logreader-7b746fc87-z4dzj           1/1     Running   0           117s
PS C:\Users\msuban01\Desktop\kube\Exercise\05_03_Configmaps> kubectl logs logreader-7b746fc87-z4dzj
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
Log level passed via env variables was: 'error'
PS C:\Users\msuban01\Desktop\kube\Exercise\05_03_Configmaps> kubectl create configmap logger --from-literal=log_level=debug
configmap/logger created
```

For this, we can use something called configmaps. We're going to call the configmap logger and we're going to take this from literal and we're going to call the actual value log_level=debug. This means that we are going to pass something called log_level=debug to the actual container as an environment variable.

We get configmap created and let's take a look at the yaml for the deployment and how that's going to change when we use configmap. Open this file up and we can see that this is similar to the last example that we opened up, except you'll notice that we have an environment variable named log_level, but the values are values from configmapRef, and the name of the configmap, and the key for this is the log_level. So this structure basically takes configmap, which is called logger, and reads the key, called log_level, out of the configmap.

After creating the configmap we can take a look at the actual configmaps that we have created. As we notice, we have the logger that we've created from before and we can actually introspect into this by doing a kubectl, get configmap/logger -o yaml, similar to the way we look at deployments. And this returns the entire configmap to us (you'll notice that in the data section, as a log_level=debug, which is what we had initially set it to).

The log level passed via environment variables was debugged. Successfully we have used configmaps to send dynamic values into our deployment.

