selves. In addition to the containers, this Pod also defines two volumes: one volume for the template parameters, backed by a ConfigMap, and an `emptyDir` volume used to share the processed templates between the init container and the application container.

With this setup, the following steps are performed during startup of this Pod:

1. The init container is started and runs the template processor. The processor takes the templates from its image, and the template parameters from the mounted ConfigMap volume, and stores the result in the `emptyDir` volume.
2. After the init container has finished, the application container starts up and loads the configuration files from the `emptyDir` volume.

The following example uses an init container for managing a full set of WildFly configuration files for two environments: a development environment and a production environment. Both are very similar to each other and differ only slightly. In fact, in our example, they differ only in the way logging is performed: each log line is prefixed with `DEVELOPMENT:` or `PRODUCTION:`, respectively.

You can find the full example along with complete installation instructions in our example GitHub repo. (We show only the main concept here; for the technical details, refer to the source repo.)

The log pattern in Example 21-1 is stored in *standalone.xml*, which we parameterize by using the Go template syntax.

Example 21-1. Log configuration template

```
....
<formatter name="COLOR-PATTERN">
  <pattern-formatter pattern="{{(datasource "config").logFormat}}"/>
</formatter>
....
```

Here we use Gomplate as a template processor, which uses the notion of a *data source* for referencing the template parameters to be filled in. In our case, this data source comes from a ConfigMap-backed volume mounted to an init container. Here, the ConfigMap contains a single entry with the key `logFormat`, from where the actual format is extracted.

With this template in place, we can now create the Docker image for the init container. The Dockerfile for the image *k8spatterns/example-configuration-template-init* is very simple (Example 21-2).