Fabric8 Integration

Fabric8 Integration

Chapter 1. Introduction

This document will help you get started building applications with Apache Camel or ActiveMQ that run on Kubernetes and various distributions of Kubernetes (ie, OpenShift v3.x).

1.1. Apache Camel and microservices

Apache Camel is a lightweight integration library that makes integration much easier. Any microservice is not an island and you very often need to integrate your microservices with disparate systems which comes in many forms or shapes. Apache Camel comes with almost 200 components out of the box that makes it easy to connect to a lot of different systems.

With Camel you can find yourself only having to write a few lines of code, or one or more class to integrate with a system using a Camel component. We think Camel makes a great fit for microservices as you can just snap in the Camel components you need to use.

In recent time a few runtimes have gained popularity for running microservices. In the following we will show you how you can use Camel with Spring Boot and WildFly Swarm. Then followed by the various Camel tooling that comes with fabric8.

1.2. Camel and Spring Boot

Apache Camel works really great with Spring Boot which in turn works awesome on Kubernetes.

1.3. Do I need to do anything different for my Spring Boot app?

No! Just create your Spring Boot app (or use an existing one if you've got one) and add:

- Add Apache Camel BOM to dependency management
- Add Apache Camel dependencies you are using
- Add fabric8 maven plugin to be able to build Docker images and run the application on Kubernetes

That's it! Here's how we get started when using Spring Boot and Camel.

1.4. Camel and WildFly Swarm

Apache Camel works really great with WildFly Swarm which in turn works almost awesome on Kubernetes.

1.5. Do I need to do anything different for my WildFly Swarm app?

No! Just create your WildFly Swarm app (or use an existing one if you've got one) and add:

- Add WildFly Camel fragment dependency
- Add Apache Camel dependencies you are using
- Add fabric8 maven plugin to be able to build Docker images and run the application on Kubernetes

That's it! Here's how we get started when using WildFly Swarm and Camel.

1.6. Camel Tooling with fabric8

The power with Apache Camel is that it does not require any tooling. However developers can benefit from the lightweight Camel tooling that comes with fabric8. For example you can run a Maven plugin that scans your source code and report any invalid Camel configurations you may have.

Camel developers may have experienced that configuring Camel endpoints or components often would require the developer to browse the Camel online documentation to find information about the options a Camel component supports. Then having to figure out to configure the options in the Camel endpoint uris. And any mistake you may type would cause Camel to fail starting at runtime. Now behold fabric8 provides a lightweight toll that allows you to type-safe edit your Camel components and endpoints using a wizard driven UI.

Spring Boot users will also appreciate that every Camel component provides Spring Boot auto configuration metadata. This means IDEA or Eclipse can assist you while you edit the `application.properties` file and provide a type-safe editor for configuring Camel components.

We will show how all this works in more details and provide links to some videos demonstrating this. TODO: refer to existing spring boot getting started TODO: use my fabric8-hello project as sample it has both SB and WF-Swarm TODO: how to add Camel Unresolved directive in index.adoc - include::inc/camel-maven-tooling.adoc[] Unresolved directive in index.adoc - include::inc/camel-forge-tooling.adoc[]