Variations & Experiments

In this lesson, we'll look at some variations and experiments that can be conducted with Cloud Foundry.

WE'LL COVER THE FOLLOWING

Variations

This chapter refers to the PaaS concept. Cloud Foundry is not the only available PaaS.

- OpenShift supplements Kubernetes with support for different programming languages to automate the generation of the Docker containers.
- Amazon Elastic Beanstalk is only available on the Amazon Cloud. It can install applications in virtual machines and scale these virtual machines. Elastic Beanstalk represents a simplification compared to the IaaS approach. Since Beanstalk is based on IaaS and some additional features, Beanstalk rests on a very stable foundation. In Beanstalk applications, additional services from the Amazon offer can be used. These include databases and MOMs. Elastic Beanstalk benefits from the numerous components available in the Amazon Cloud.
- Heroku is only available in the public cloud. Similar to Cloud Foundry it has buildpacks for supporting different programming languages and a marketplace for additional services.

Experiments

- Supplement the Kubernetes system with an additional microservice.
    - As an example, you can take a microservice used by a call center agent to make notes about a conversation. The call center agent should be able to select the customer.
    - Calling the customer microservice has to use the host name `customer.local.pcfdev.io`.
    - You can copy and modify one of the existing microservices.
    - Enter the microservice in the file `manifest.yml`.
    - The deployment of a Java application can be derived quite easily from the existing `manifest. mf`. For other languages the documentation of the buildpacks might be helpful.
- Get acquainted with the possibilities of running Cloud Foundry. Start the example and have a look at the logs with `cf logs`. Log into the Docker container of an application with `cf ssh` and see the latest events of a microservice with `cf events`.

It is also possible to use the Cloud Foundry infrastructure in other areas of the application. However, these experiments are harder to do.

- Replace the integrated database in the microservices with MySQL. For

information on how to start MySQL with Cloud Foundry, see The Example with Cloud Foundry. However, the code and configuration of the

microservices must be changed in order for the applications to use MySQL. There is a Guide for that.

- Cloud Foundry also provides RabbitMQ in the marketplace. This MOM can be used for asynchronous communication between microservices. A guide shows how to use RabbitMQ with Spring so you can port the example from chapter 7 to RabbitMQ and then run it with a RabbitMQ instance created by Cloud Foundry.
- An alternative is user-provided service instances. With this approach, a microservice can obtain information about the Kafka instance with Cloud Foundry mechanisms. The Kafka instance is not running under the control of Cloud Foundry. Change the Kafka example from chapter 7 so that it uses a user-provided service Kafka instance. This concerns the configuration of the port and host for the Kafka server.
- Change one of the microservices and use Blue/Green Deployment to deploy the change so that the microservice does not fail during the deployment. The Blue/Green Deployment creates a new environment and then switches to the new version so that no downtime occurs.

We'll look at Serverless in the next lesson.