In this module, we are building our application in an OCI container and build it on Gitlab.

Task 1: Build OCI Images

Based on the same sources, you should be able to build images: Read the Readme carefully and build:

1. Quarkus: One Image with a normal JVM and the Dockerfile
2. Quarkus: One Image with a native application and the Dockerfile
3. Quarkus: The application with buildpacks.
4. Flex: One with the Dockerfile.slim
5. Flex: One with the Dockerfile.ubi

Start the container locally and compare the:

- Image Size
- Resource Consumption

Information about starting can be found in the Readme. Information about how to determine the resource consumption can be found in the internet.

Task 2: Build Your Software with Gitlab-CI

Fork the templates if not yet done.

Based on the findings of Task 1, generate a Gitlab-CI Pipeline with the following attributes for your two forked repositories:

- Java-Applications should be built natively and as Java-Application in two OCI-Containers
- Flex should be built in an OCI-Container
- The Git-Reference (branch or tag) should act as tag for the OCI-Container
- The OCI-Container should be stored in the internal registry

You are free to use build packs or Dockerfiles.
You can use any templating you find (Wizards within Gitlab, Google, ChatGPT, etc.).

However, the releasing must after takes place upon creation of a git tag and the gitlab-ci and not via some autoreleasing in Gitlab. Furthermore, the pipeline should be documented: you are responsible for the correct outcome of the pipeline.

Task 3: Bonus: Create a Lib

Create a Library for the build. This is going to make the build in the next stept easier plus you only have to document once. Use this pipeline-lib into your quarkus and flex-builds.