In this module, we are building the chatbots and deploy them to kubernetes.

Task 1: Build Chatbot Images

Based your pipeline of the last tutorial, fork and build the chatbots:

1. Fork the chatbots from
    1. https://gitlab.fhnw.ch/cloud/devops/templates/chatbots/eliza
    2. https://gitlab.fhnw.ch/cloud/devops/templates/chatbots/roberta
    3. https://gitlab.fhnw.ch/cloud/devops/templates/chatbots/connecting-worlds
2. Based on the findings of the last lecture, build the application:

- Eliza and Connecting-Worlds should be built natively and as Java-Application in two OCI-Containers
- Roberta 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 your solution from the last lecture.

Releasing must after take place upon creation of a git tag and the gitlab-ci and not via some autoreleasing in Gitlab. Feel free to use any template / autogeneration tool you like. You are, however, responsible for the pipeline and need explain its doing in the report.

Of course you can re-use your own pipeline-lib if you have one created. However, do not rely on my gitlab-components as seen in the templates-folder, use your own solution. If you have questions, do not hesitate to contact me.

Task 2: Create Kubernetes Resources

Quarkus can generate entire kubernetes-resources, also based on predefined settings in `src/main/resources/application.yaml` with the help of a maven dependency already enabled. Use it and adapt them to deploy your applications to AKS:

1. Create a new repository called kubernetes in your project in gitlab called `kubernetes`.
2. Build your app locally. No native build is needed however this should generate resources under `target/kubernetes` in your workspace. Copy the resulting .yml into the new repository named after the repository (e.g. `eliza.yml, connecting-worlds.yml`). Do the same for both quarkus apps. Copy them for your python app as well (`roberta.yml`).

3. Adapt the yml.files:

    1. Search and replace the image tag if it is not fitting the link to your registry

    2. Adapt `connecting-worlds.yml` according to the Readme: the `application.yml` must be mounted into the `deployment.yml.`

    3. Copy one file to `roberta.yml` and adapt it:

        1. remove all quarkus metadata like timestamp, etc.

        2. Adapt the image

        3. Adapt the port

        4. remove liveness and readiness for now

4. Deploy on AKS

    1. Login into AKS

    2. Download kubeconfig via the az-cli: https://portal.azure.com

    3. Check if kubectl is working:
       ```
       kubectl get nodes
       ```

    4. Create a new Namespace `chatbots:`
       ```
       kubectl create ns chatbots
       ```

    5. To access images in the gitlab-registry, you need a credential from gitlab and register it to your cluster.

        1. Go to your subgroup (namely fs24-…)

        2. Go to Settings -> Access Tokens and create a new Token. Read Registry should be sufficient. Copy the token.

        3. Create the pull-credential in the namespace by hand:

```
     kubectl create secret docker-registry regcred --docker-
server=cr.gitlab.fhnw.ch --docker-username=NAME --docker-password=TOKEN
--docker-email=no-reply@fhnw.ch -n chatbots
```

    6. Go into the kubernetes-folder and deploy the pods.:
       ```
       kubectl apply -f eliza.yml
       kubectl apply -f roberta.yml
       kubectl apply -f connecting-worlds.yml
       ```

    7. Check if the application works by taking a look at the logs and the status of the pods. Lens might help