Pipeline with Dockerfile

Up until now, we have only made sure that Github Actions can reach the `hello-world.yaml` configuration file, but not really made it do anything useful.

As a next step, we want Github Actions to do two things:

- clone our project
- build the code using Gradle

Tasks

- Instead of printing "Hello World!", now we want to use a docker image that has both JDK and Gradle installed. After section `runs-on:` add `container: gradle:6-jdk11`

```
container: gradle:6-jdk11
```

- Under the `steps` part, insert a `- uses:` list item to the list (before the existing `- name: my-step` item) with action: `actions/checkout@v2`

```
- name: Clone-down
  uses: actions/checkout@v2
```

- Change the `run` command to run `ci/build-app.sh` as the command instead of `echo "Hello World!"`.

▶ :bulb: got issues with actions running the script? see here:

In case of issues with access denied add `chmod +x ci/build-app.sh` before the execution, so the entire run script looks like this:

```
 - run: chmod +x ci/build-app.sh && ci/build-app.sh
```

If you want to see what `build-app.sh` is doing, look into the script.

- Commit and push the changes. Github Actions should automatically detect your new commit and build again.
- Look into the logs of Github Actions and see if the build was successful.

▶ :bulb: If you strugle and need to see the whole ***Solution*** you can extend the section below.

```
on: push
jobs:
  Build:
```

```
    runs-on: ubuntu-latest
    container: gradle:6-jdk11
    steps:
      - name: Clone-down
        uses: actions/checkout@v2
      - run: chmod +x ci/build-app.sh && ci/build-app.sh
```

Results

See that the build runs green and outputs this in the step log:

```
Run ./ci/build-app.sh

Welcome to Gradle 6.9!

Here are the highlights of this release:
 - This is a small backport release.
 - Java 16 can be used to compile when used with Java toolchains
 - Dynamic versions can be used within plugin declarations
 - Native support for Apple Silicon processors

For more details see https://docs.gradle.org/6.9/release-notes.html

Starting a Gradle Daemon (subsequent builds will be faster)
> Task :clean UP-TO-DATE

> Task :compileJava
Note: Creating bean classes for 3 type elements

> Task :processResources
> Task :classes
> Task :shadowJar

Deprecated Gradle features were used in this build, making it incompatible with
Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See
https://docs.gradle.org/6.9/userguide/command_line_interface.html#sec:command_line
_warnings

BUILD SUCCESSFUL in 28s
4 actionable tasks: 3 executed, 1 up-to-date
```

Congratulations, you have now build the java application!