Extras

Reusing build cache

Github Actions has a few different methods for reusing files and artifacts produced in a job, in downstream jobs or even in subsequent builds.

This is needed because each job in Github Actions is running in a separate docker container or machine, and by default no files are shared between these.

Caching

The caching mechanism is persistent across multiple builds, and therefore a key is needed.

One example is to store downloaded dependencies in a cache, to avoid downloading the same dependencies over and over. And since dependencies typically are defined in one central file, this file is hashed and used as a key. At Github Actions `actions/cache@v2` can be used to do it. Possible input parameters are:

- `path` - (required) The file path on the runner to cache or restore. The path can be an absolute path or relative to the working directory.
- `key` - (required) The key created when cache was saved and later used to find it.
- `restore-keys` - (optional) An ordered list of alternative keys to use for finding the cache if no cache hit occurred for key.

The example for gradle can be seen below.

```
- name: Cache Gradle packages
  uses: actions/cache@v2
  with:
    path: |
      ~/.gradle/caches
      ~/.gradle/wrapper
    key: ${{ runner.os }}-gradle-${{ hashFiles('**/*.gradle*', '**/gradle-wrapper.properties') }}
    restore-keys: |
      ${{ runner.os }}-gradle-
```

To use cache at the consequtive jobs, the same code (from above example) has to be added as a step.

Limitations on using caching:

- Github will delete all unaccessed caches for 7 days.
- There is no limit on how many caches can be stored in the repository as long as it is below 5GB.

Tasks

In this exercise the artifacts management in the existing pipeline will be changed to use caching instead as follow:

1. Create a new Github Actions file under `./github/workflows/caching.yaml` with exact copy of the pipelimne from exercise 7.
2. Instead of using upload and download artifacts at each job, use caching with an above example for Gradle.
3. Should artifact be uploaded at the end of the workflow?

Resources

More information about caching can be found here: https://docs.github.com/en/actions/guides/caching-dependencies-to-speed-up-workflows Cache action tool: https://github.com/actions/cache