Workflow

When you have larger or more complex projects, you’ll want separate jobs to do separate things in paralell.

Up until now, we have had a job called `Build` both for the build and test, but that is not necessarily descriptive. The only reason we have done this, is because Github Actions **requires** you to have at least one job with the name fx. `Build`

We define each job as a collection of `jobs` key:

```
jobs:
  my_first_job:
    name: My first job
  my_second_job:
    name: My second job
```

By default all jobs are run in parallel. To control the order of jobs execution, we can add the `needs`key. A job can declare that it `needs` one or more (a list) of jobs to finish successfully before it is triggered.

To run the two jobs sequentially we define a workflow where job-2 "requires" job-1 to have run before it starts.

```
name: workflow
jobs:
  job-1:
  job-2:
    needs: job-1
```

This also ensures that `job-2` is not run if `job-1` fails. It is possible to add name to workflow fx. here: `name: workflow`.

To ensure that all files from previous jobs are available at new one, we have to make sure to upload artifact at the end of the job and download it at the beginning of a new one. The way to do it can be found in previous exercise `04-storing-artifacts.md`.

Tasks

Let's try to clean up our current build by utilizing workflows:

- Name your workflow `Java CI`.

```
name: JAVA CI
```

- Divide your job into two jobs: `Clone-down` and `Build`. `Clone-down` will checkout the repository, `Build` will build the code, run the tests, and stores the results.

```
jobs:
  Clone-down:
    ...
  Build:
    ...
```

- `Build` should be dependent on `Clone-down` job. Each of them also needs a running instance and container.

```
Clone-down:
    runs-on: ubuntu-latest
    container: gradle:6-jdk11
    steps:
      - ...
```

- Remember that to have information from previous job(s) the artifact with this information needs to be downloaded and respectively uploaded by using (`actions/upload-artifact@v2`and `actions/download-artifact@v2`).

Solution

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

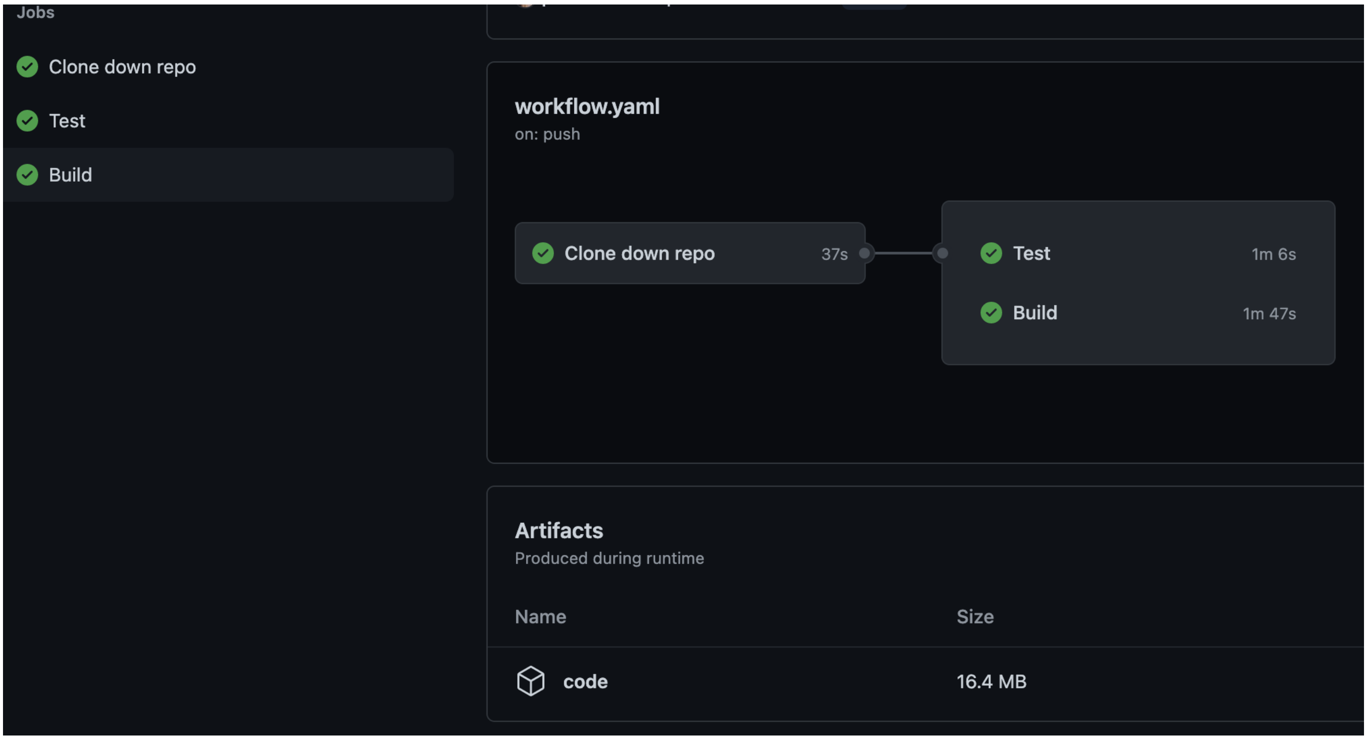
▶ Solution

```
name: Java CI
on: push
jobs:
  Clone-down:
    name: Clone down repo
    runs-on: ubuntu-latest
    container: gradle:6-jdk11
    steps:
    - uses: actions/checkout@v2
    - name: Upload Repo
      uses: actions/upload-artifact@v2
      with:
        name: code
        path: .
  Build:
      runs-on: ubuntu-latest
      needs: Clone-down
```

```
        container: gradle:6-jdk11
        steps:
        - name: Download code
          uses: actions/download-artifact@v2
          with:
            name: code
            path: .
        - name: Build with Gradle
          run: chmod +x ci/build-app.sh && ci/build-app.sh
        - name: Test with Gradle
          run: chmod +x ci/unit-test-app.sh && ci/unit-test-app.sh
        - name: Upload Repo
          uses: actions/upload-artifact@v2
          with:
            name: code
            path: .
```

Results

Opening it should show something like:



Artifacts

Produced during runtime

Name	Size
code	16.4 MB

More information about this topic can be found here: https://docs.github.com/en/actions/reference/workflow-syntax-for-github-actions

Resources

https://github.com/marketplace/actions/upload-a-build-artifact
https://github.com/marketplace/actions/download-a-build-artifact