Continuous Integration / Continuous Delivery – Practical Exam

Perform the below tasks in order. Create a separate commit per numbered task.

Due Date: 30th January 2026

1 – Create repository

Create a public GitHub repository named "ustp-cicd-final". Add the uploaded source code from *ecampus* to the root: https://ecampus.ustp.at/course/view.php?id=37434

2 – Settings

Create a branch ruleset in the repository settings to configure the following rules for the default branch (either main or master):

- Disallow commit pushes without a pull request
- Disallow force pushes
- Disallow the deletion of the branch

Additionally, the following **General** settings should be enabled in the repository:

- Allow Pull Requests to only be "squash" merged.
- Configure branches to automatically be deleted after being merged.
- Enabled "auto-merge" for Pull Requests.

Continue the next steps by submitting pull requests.

3 – Build GitHub Workflow

Create a GitHub workflow that builds the repository when submitting a pull request and when a commits gets pushed. Set the workflow as required when merging pull requests and also enable the "Require branches to be up to date before merging" sub-setting.

4 – Build Artifacts

Update the GH workflow and include a step that uploads the build artifacts. Make sure that the ready to use web application is included in the artifacts.

5 – Testing

Use AI with a model of your choice and ask it to generate unit tests for this web application. Make sure that the unit tests pass. Add a step that runs the tests to the GH workflow after the build and make sure to also upload the test results.

Add a matrix so that the tests run on different operating systems: "windows-latest" and "ubuntu-latest".

Optional: Add a code coverage measurement step with an action of your choice and upload the results.

6 – Publish

Add a publish workflow that can be manually triggered that publishes the built web application to GitHub pages. To avoid code duplicity, don't add steps to build the web application to this workflow. The logic for that already exists in the build workflow. Depend on that other workflow and download the build artifacts.

7 – Release and Tagging

Add a workflow that automatically publishes a release when a tag (in the format of vX.X.X, i.e. 1.0.0) gets pushed to the repository. Configure the release to include the following metadata:

- Title: “USTP CI/CD Tetris vX.X.X” (vX.X.X is the tag)
- Build Artifacts
- Automatically generated release notes based on the commits

Create a tag `v1.0.0` from the current *HEAD* commit and push it to the GH repository. Verify that the release got created correctly.

Add a “tag ruleset” in the repository settings for tags in the format of (vX.X.X) that disallows the deletions and force pushes.

8 – Dependabot

Configure GH Dependabot to update the dependencies in the repository via one combined PR with a weekly interval. Verify that it is working correctly and that it opens a Pull Request.

9 – CODEOWNERS

Add a CODEOWNERS file to your repository and make sure that you get notified (aka “tagged”) whenever a workflow file gets changed in a Pull Requests.

10 – REAMDE

Add a README.md file to the root of the repository that provides basic information:

- Link to GitHub Page
- Developer instructions how to build and test the application locally

Add status badges for all workflows. See https://docs.github.com/en/actions/how-tos/monitor-workflows/add-a-status-badge. The “build” workflow status badge should only show the result for “push” triggered runs, not Pull Requests.

Optional: Add an additional badge that shows the code coverage in percentage. For this you need to do the optional activity in step 5.

11 – Spellcheck workflow

Add a workflow that spell checks markdown files (.md files, i.e. the README.md file) and errors if there’s a spelling mistake. The workflow should always trigger when submitting a pull request, regardless of if markdown files were changed or not.

Configure the action to ignore a set of technical terms i.e. “vscode”. Make sure to use a spellcheck action that supports that.

Mark the workflow as required when merging Pull Requests.

Provide access and submit

Add the user `ViktorHofer` as a collaborator to your repository. Send a mail to lbhoferv@ustp.at with the link to your repository.