MULUNGUSHI UNIVERSITY

Masters in Cyber Security

LAB REPORT

GitHub Setup, CI/CD Pipeline & Collaboration

Student: Chilufya Chimfwembe
Date: February 2026

1) Introduction

- This report provides detailed documentation of all steps performed during the GitHub project setup.
- It covers repository creation, Python script development, CI/CD workflow automation, and collaboration.

2) Creating a GitHub Account

- Typed github.com in the browser to reach the GitHub homepage.
- Clicked the 'Sign up' button to begin creating the account.
- Provided email, strong password, and a unique username.
- Verified the account through a confirmation email.
- Enabled Two-Factor Authentication (2FA) for added security.
- Logged into the GitHub dashboard for access to repositories and settings.

3) Creating the Project Repository

- Clicked the '+' icon on the top right of the GitHub page.
- Selected 'New repository' from the menu.
- Named the repository 'my-first-pipeline'.
- Enabled 'Add a README file' for initialization.
- Clicked 'Create repository' to finalize setup.
- Viewed the repository homepage which acts as the project workspace.

4) Creating the Python Application (hello.py)

- Clicked 'Add file' then 'Create new file'.
- Named the new file hello.py.
- Inserted the script: print("Hello, Secure World!").
- Clicked 'Commit new file' to save it in the repository.
- Verified the commit inside the history tab.

5) Setting Up the CI/CD Workflow

- Clicked 'Add file' and selected 'Create new file'.
- Named the workflow file .github/workflows/pipeline.yml.
- Added YAML code defining a pipeline triggered on each push.
- Specified ubuntu-latest as the execution environment.
- Committed the workflow file enabling GitHub Actions.

6) Observing the Pipeline Execution

- Clicked the 'Actions' tab to open workflow runs.
- Saw that a workflow was automatically triggered by the commit.
- Opened the run logs to observe execution.
- Confirmed successful pipeline output including printed message.

7) Collaboration With Peers

- Opened repository settings then selected 'Collaborators and teams'.
- Added friends' GitHub usernames and sent invites.
- Friends accepted and gained push/edit permissions.
- Friend updated hello.py with a new message.
- Viewed commit differences showing added and removed lines.
- Observed GitHub Actions triggering automatically after their commit.

8) Conclusion

- This lab demonstrated practical DevSecOps principles using GitHub.
- CI/CD pipelines ensured continuous validation of all code changes.
- Collaboration tools supported traceability and secure group work.