Techniques And Tools

I converted a compound interest calculator I made last year into a CI/CD-enabled project using GitHub Actions for continuous integration and delivery. Docker containerization packaged my Flask web app into a portable unit, while pytest created ten unit tests covering some calculation scenarios and edge cases. I used Flake8 to lint and maintain my Python code quality standards, and GitHub Container Registry handled automated image deployment. The workflow automation included build.yml, test.yml, lint.yml, and deploy.yml files that trigger on every push to my main branch.

Key Steps To Build My Release Package

The release package began with a Dockerfile based on Python 3.9-slim, installing the dependencies from requirements.txt, copying the application code, and configuring Gunicorn for production deployment. I took some security measures by creating a non-root user and exposing port 5000. The GitHub Actions pipeline did the build process, authenticating with GitHub Container Registry using repository secrets, tagging the Docker image with the latest version, and then pushing it for deployment.

Challenges Encountered

I had some version compatibility issues between Flask 2.2.3 and newer Werkzeug versions, which caused import errors and required an upgrade to Flask 3.0.3. I also ran into some Docker container connectivity problems during the build pipeline, which I resolved by correcting the Gunicorn command syntax and ensuring proper port binding. A minor problem I also had was with the tests. There were some floating-point calculation differences, which I fixed by updating expected values to match actual calculations.

How DevOps Methodologies Can Benefit My Future Projects

I believe DevOps methodologies are very useful and I plan to use them a lot in future projects. They can help me a lot, like with automated testing that maintains consistent code quality and catches errors early. Docker containerization also eliminates environment-specific issues, while CI/CD will help accelerate development cycles by automating tasks I have to do over and over. Infrastructure as code approaches allow deployment configurations to be versioned and improved iteratively, and Git will obviously help me a ton with my future work and collaboration in school.

Reflection On This Course

I genuinely believe this has been my most important class so far at Gonzaga. Before this class, all I ever did was write little programs on my local machine and never even really thought about scaling and testing, etc. I think this course has really helped me shape my perspective by emphasizing certain concepts that I will apply throughout my entire career. This course has also brought to my attention the number of technologies and different things that I have to learn and be skilled at in this market. I'm very happy I took this class as early as I did in my college journey, and I really appreciate the love Dr. Crandall has for these topics. Thanks for the great semester and good luck in Germany!