

1. Conversion

To convert the project to a CI/CD one, I copied the files from the simple counter service over to a new repository, put in my new primes file, and modified filenames and the shell scripts. I ensured the makefile worked fully on my system before attempting any CI/CD with the self-hosted runner.

2. Debian Package

Mostly, I re-used the build code for the simple counter project. The main changes I did to ensure the package built correctly was changing filenames and ensuring that none of the daemon or config files required for the counter service remained. Building the package was fairly simple. All of the required files for the program were put into a temporary folder and packaged up. The packaged product was uploaded as an artifact to the github repository during the workflow.

3. Challenges

The main challenge I had was adding testing to the program. Many tests would fail as I was unfamiliar with pytest and it took me some time to get everything working on my end. In the workflow, the testing failed as a file was in an incorrect spot which did not occur on my main machine. Originally, I was going to run a colab notebook with tensorflow but realized I did not have space for everything on my VM or on the google VM self-hosted runners. I could have fixed this, but the program would have taken ~5 minutes to run regardless and I wanted to have something that gave immediate feedback. The only other issues I encountered were simple naming mistakes or an incorrect `rm -rf` command in the make clean. Thankfully I had the presence of mind to push everything before running make clean.

4. DevOps Methodologies

After learning about and using devops methodologies this semester, I see a clear benefit: time. While the initial setup is somewhat time consuming and tedious, the time savings with respect to testing and creating a final version that works on *any* computer is well worth it. I believe I will be doing quite a bit of CI/CD at my internship this summer as it will give me guarantees that what I do works, and works for everyone.

5. Reflection

This course has shown me the light of automation. I have always enjoyed optimization games such as satisfactory or factorio, and this class scratches the same itch. None of the methods or technologies are that difficult to work with, and the payoff is incredible. Being able to

upload changes and get immediate feedback on whether the project works is huge for debugging and working on a team. In the future, I will work to automate my life away and try to avoid manually testing, building, deploying, or generally working. I have done things manually for too long, it's time to press a button and watch as things go right (hopefully).