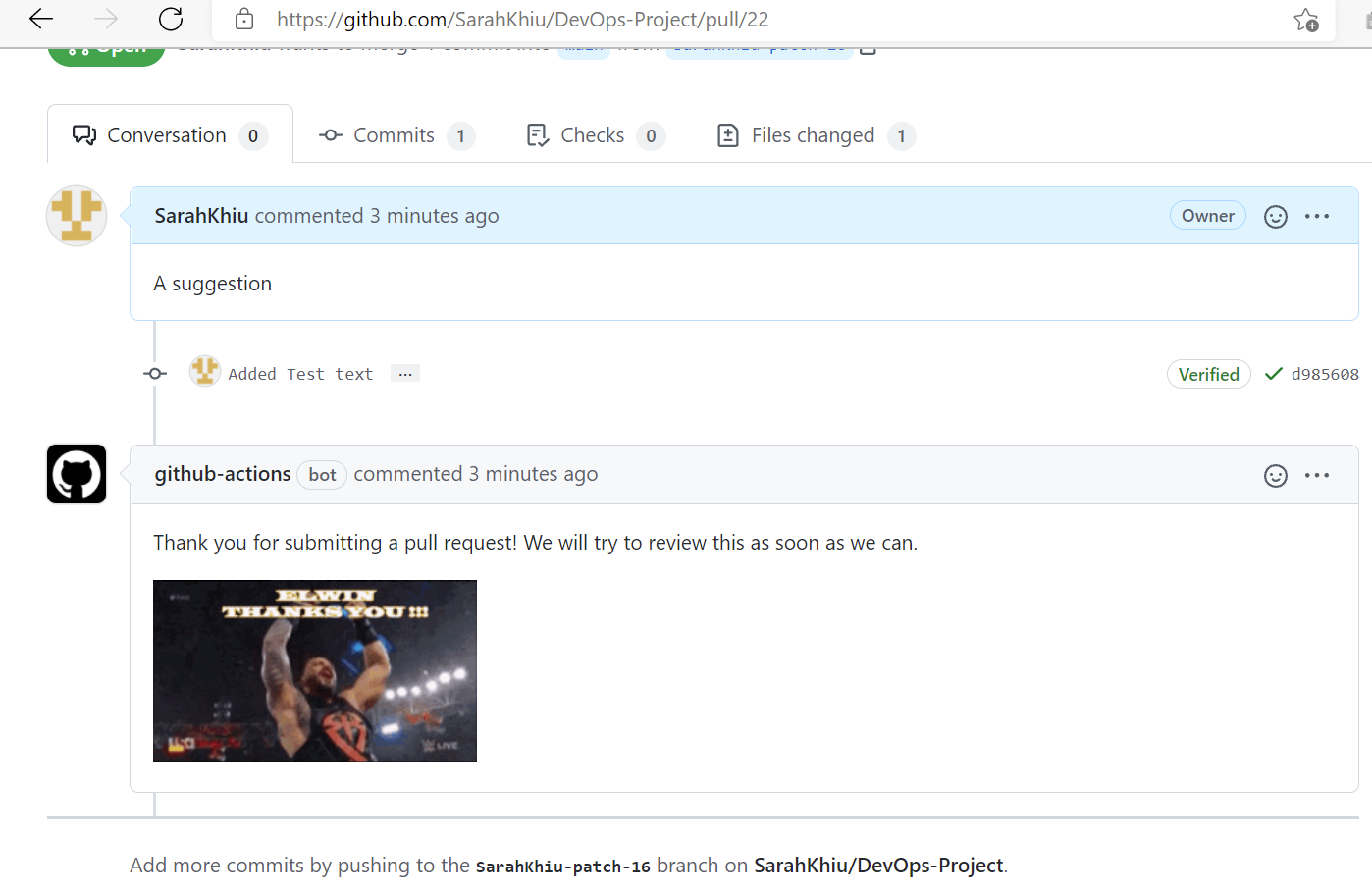
**Thought Processes and Learnings**

Repository: SarahKhiu/DevOps-Project

**What I tried:**

1. *Creating a customised GitHub Action*

Using node and JavaScript to add a new comment on every pull request created saying “Thank you!” to the person who created the request along with a GIF from the Tenor API to make it a little fun.

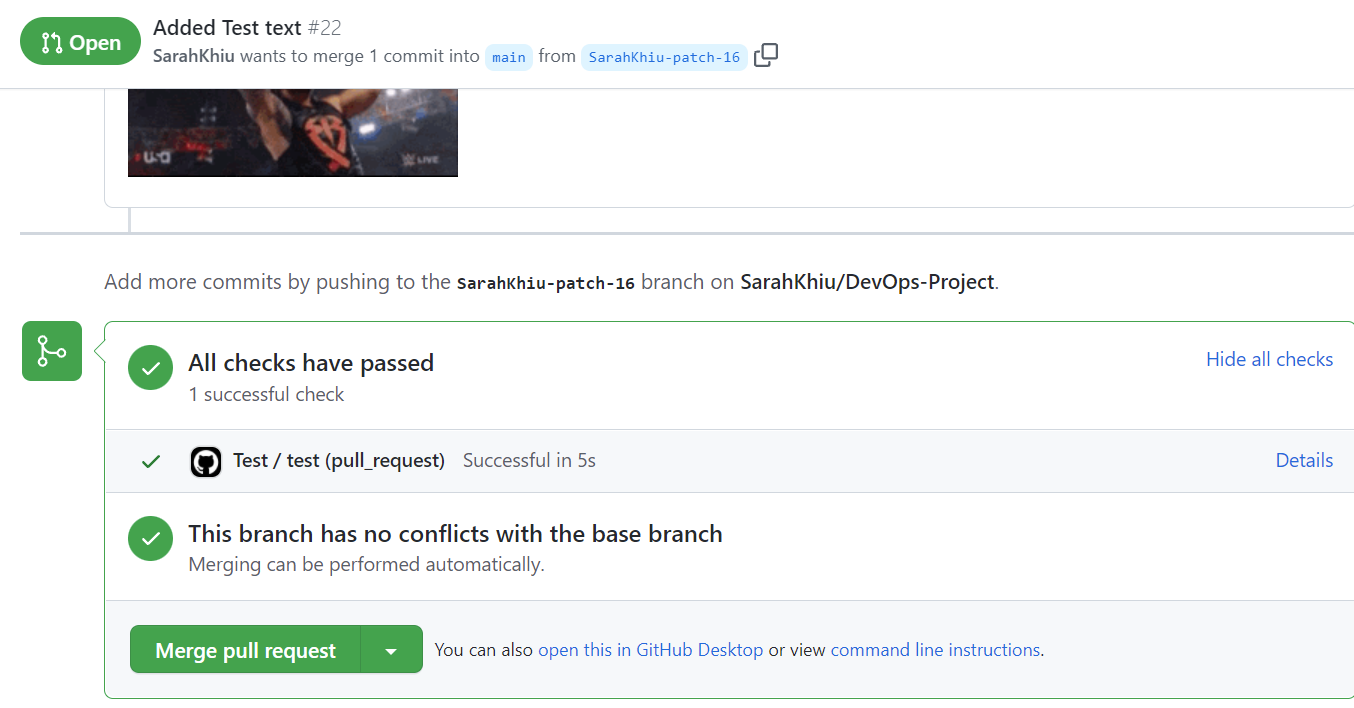


**What didn’t go quite as well initially:**

* Error message was “error: can't find 'action.yml', 'action.yaml' or 'dockerfile' under '/home/runner/work/devops-project/devops-project'. did you forget to run actions/checkout before running your local action?” when running workflow.

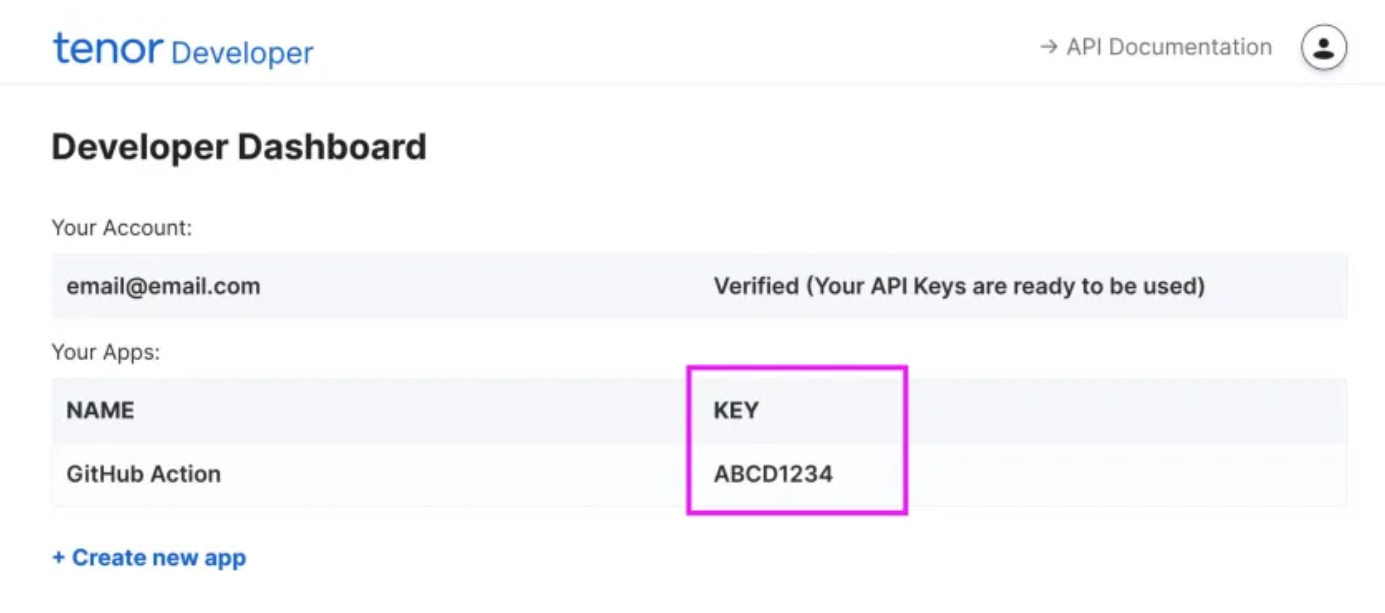
**\*\* Edited 7 Oct, 1:23AM:**

Finally managed to make it work: A ‘Thank you!’ message whenever a pull request is made to the repository with a random Thank You GIF each time using Tenor API.



Created another repository, ***tryAgain***, and tried to start from scratch again to see what was the issue that prevented the first attempt from working in ***DevOps-Project*** in order to troubleshoot the latter.

**Discovered issues of:** Did not install node-fetch package, which would make the browser’s fetch API available for us inside of node, so that it can interact with the Tenor API. Had to sign up for an account with Tenor API in order to get an API key. After which, had to save Tenor Key under Settings of your repository (Secrets>New Repository Secret>Type Tenor Token>Enter Tenor API Key under Value).

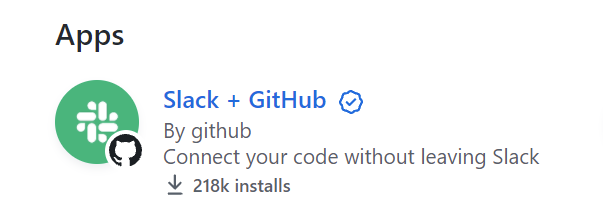


Last step, typed ‘npm build’ or ‘npm run-script build’ in terminal in VS code to get the last changes in your project. Test run on GitHub by creating a pull request when making edits on README.md file. Test should run correctly this time, and a ‘Thank You’ message plus ‘Thank You’ GIF will appear on screen. Another issue was also to check the alignment of text in yml file that could affect it from running successfully. \*\*end edit

**What was successful:**

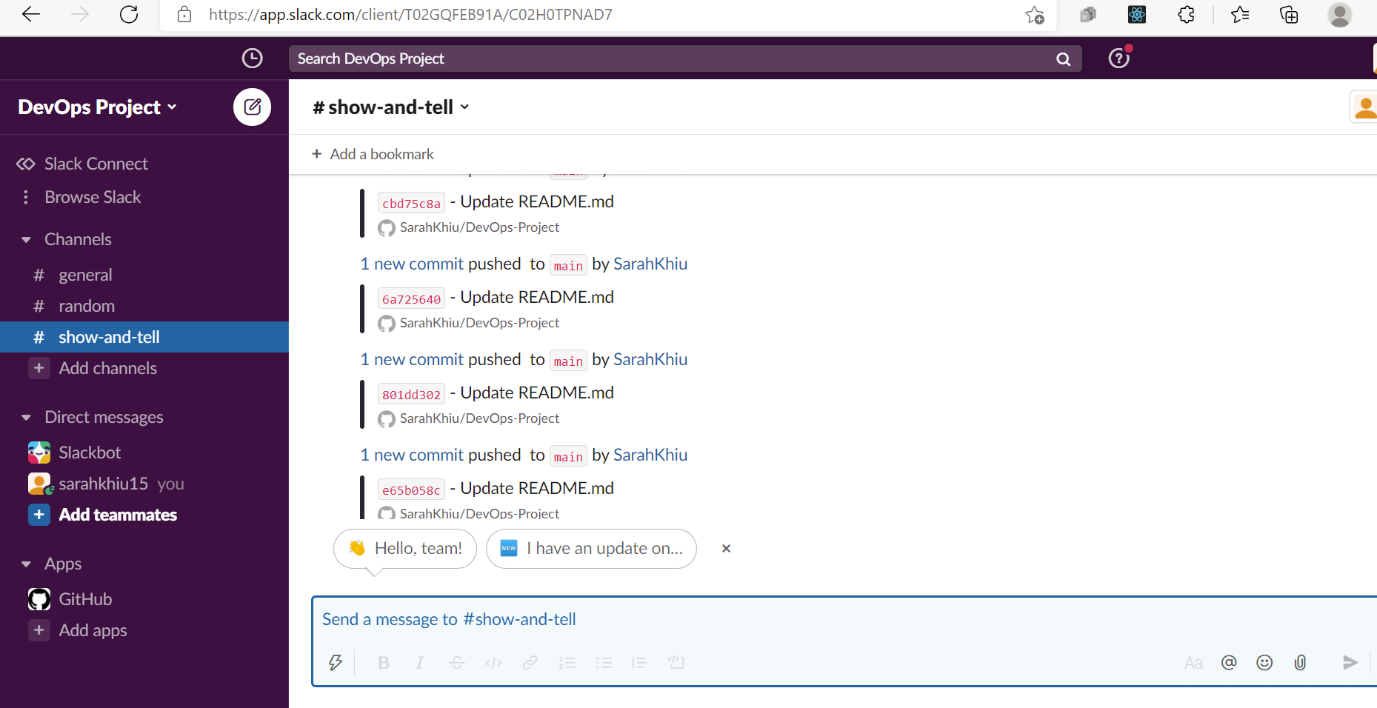
1. *Automation*

Linked Slack with GitHub such that notifications are automatically pushed to my Slack account/selected channel whenever there is a new commit in this Repository.



Steps:

* In GitHub Marketplace under Apps, search Slack + GitHub to install for free. Following which, go to your Slack application, connect to your GitHub Account then select a channel to subscribe to and push all notifications made from a repository. This channel will then receive all notifications from a particular repository or from all.



Repository: SarahKhiu/Google-Cloud-Run-Git-Hub-Actions

**What I tried:**

1. *Deploying CI/CD to Google Cloud Run Using GitHub Actions*

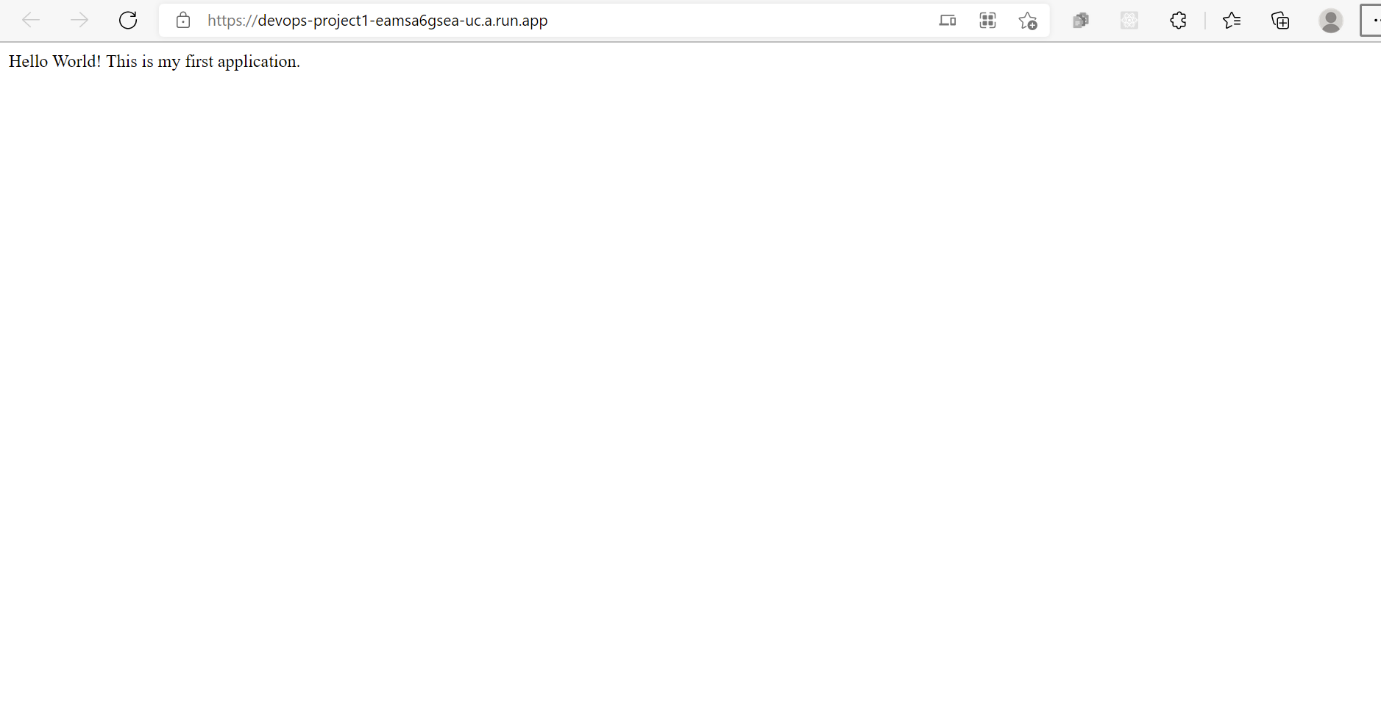
Used u1i/nodejs-api, Currency Exchange API – NodeJS, as a starting point for the project. Followed instructions from this [link](https://towardsdatascience.com/deploy-to-google-cloud-run-using-github-actions-590ecf957af0). The specific task was to have a simple Flask server deployed via GitHub Actions to Google Cloud Run.

Steps:

* What was required: GitHub Actions (an automation solution that allowed you to supply a yaml configuration to automate tasks like CI/CD operations), Google Cloud Run (a managed serverless platform provided by Google that scales down to zero. User just needs to provide a Dockerfile for the container to be run) and Flask (a Python micro web framework commonly used by Python developers).

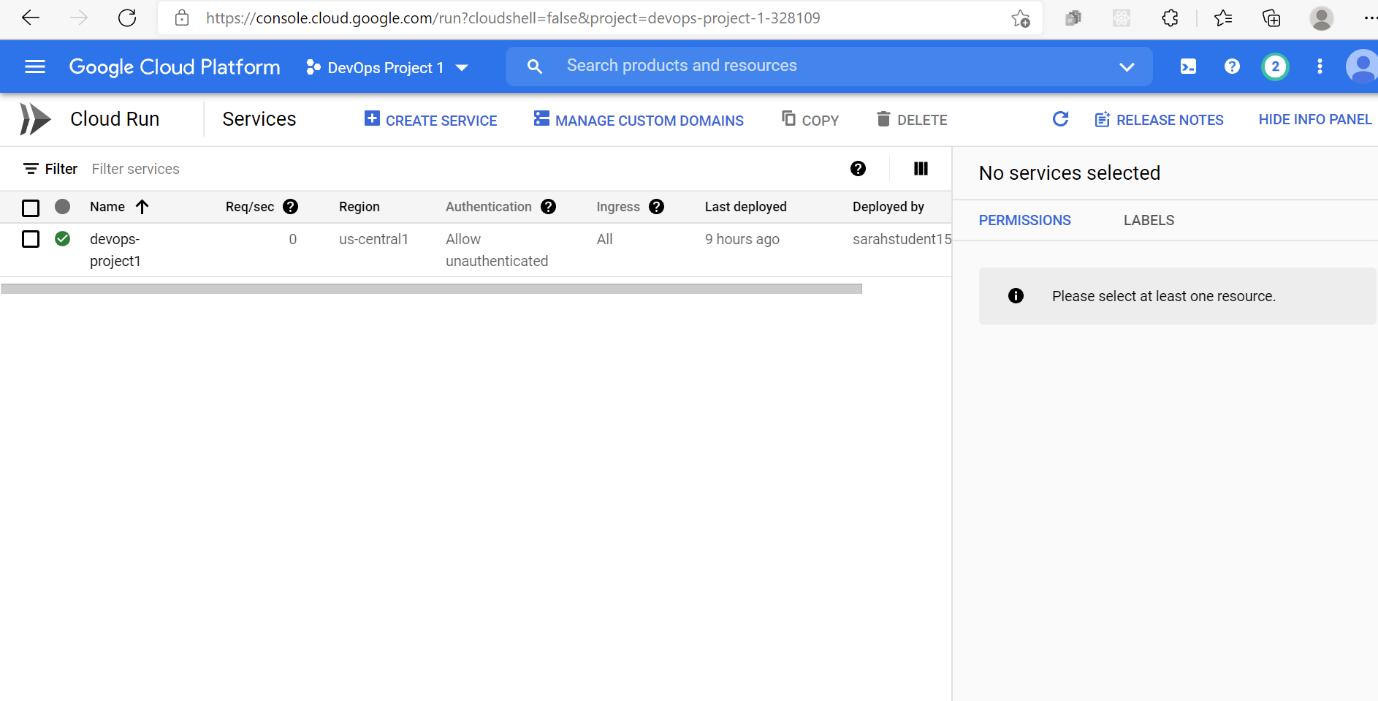
**What was successful:**

I got a simple Flask server running on Google Cloud Run, and deployed onto the web. [https://devops-project1-eamsa6gsea-uc.a.run.app](https://devops-project1-eamsa6gsea-uc.a.run.app/)

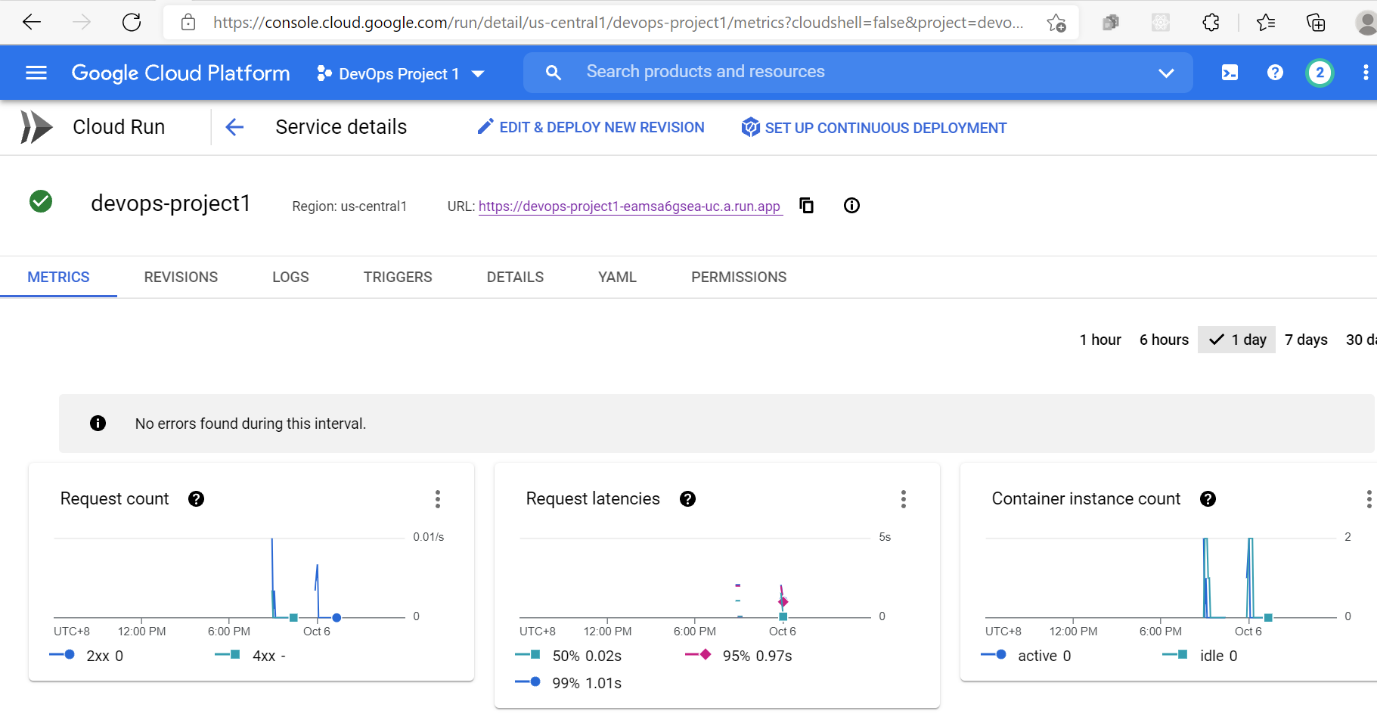


This was done on Google Cloud Platforms and enabling the Cloud Run API. Using the CloudShell Terminal, I imported Flask and was able to run the project locally. Import Python and gunicorn as a dependency so that Google Cloud Run can work. Next, a Dockerfile for Container was created.

Using the CloudShell Editor, we see our Dockerfile and python files created on the left then we run the files on the Cloud Run Emulator >> Authorise Cloud Shell and also have an enabled Billing Account. After successful run of our application, we are also to see a web preview on port 8080.

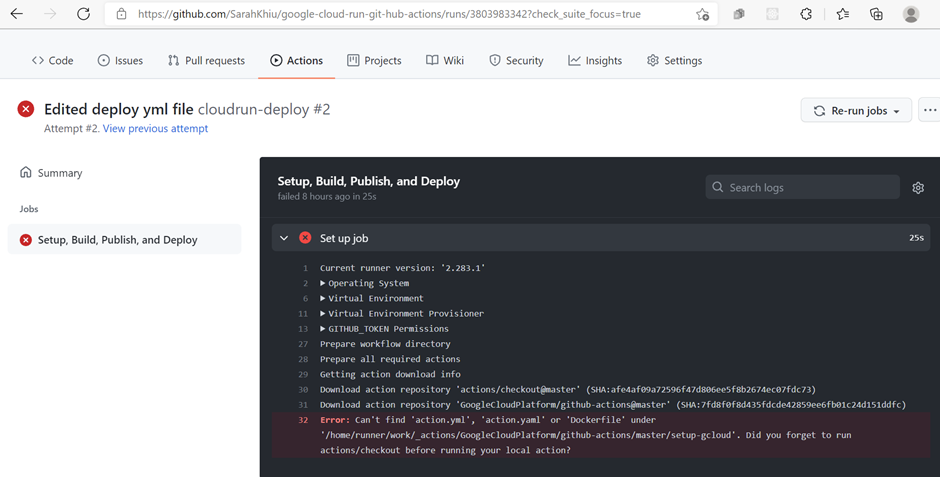


To deploy our service, we need to deploy to Cloud Run by selecting our Project name. We can choose the deployment platform and region you would like to deploy. After clicking on Deploy, our project appears on our Google Cloud Platform Projects Page and a URL link also appears that is accessible worldwide.



**What didn’t go as well:**

I didn’t manage to successfully run the GitHub Actions workflow despite following the instructions and creating GitHub secrets and following steps to set up Google Authentication and ‘credentials’. Error message was “error: Can't find 'action.yml', 'action.yaml' or 'Dockerfile' under '/home/runner/work/\_actions/GoogleCloudPlatform/github-actions/master/setup-gcloud'. Did you forget to run actions/checkout before running your local action?” when running workflow.



Learnings & Reflections:

I approached this project not with the intention of coding something new and fantastic, rather I wanted to know if I was able to connect the dots and put the pieces of the puzzle in DevOps together by using the different programmes and tools that would help in this automation.

It was quite a tedious process when attempting this project, as the scope was broad (we were free to showcase any CI/CD workflow we preferred). Lots of time was spent watching online tutorials and reading up for clarity on certain information with regards to DevOps and how GitHub Actions work.

Small successes motivated me to keep going, but I also encountered a fair share of workflow failures and not being able to achieve the entire project’s objective. Thankfully, we were not expected to show all successes in our attempts.

I don’t think I’m familiar enough yet with GitHub Actions workflow instructions such as ‘GitHub Token’, ‘secret credentials’ or API implementation, as I seem to encounter issues when I attempt to run these instructions.

Despite the bumps in the road, I would say overall, it was an enriching experience completing this project as you really learn to celebrate the small successes when things work and you see something like this.

