National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cloud Computing

Aniket Ashok Shetty
Student ID: x23217529

School of Computing
National College of Ireland

Supervisor: Prof.Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Aniket Ashok Shetty
Student ID:	x23217529
Programme:	MSc in Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Prof.Vikas Sahni
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	2211
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Aniket Ashok Shetty
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Aniket Ashok Shetty
x23217529

1 Introduction

In this document a detailed overview of all the tools and software are being specified, and how those tools are integrated with each other has been explained for this project. The main objective of this project is to provide a more enhanced security testing tool that needs to be integrated in the CI/CD pipeline to increase the efficiency, performance and effectiveness of the software development lifecycle. So, by integrating Static, Dynamic & Interactive Application Security Testing this was possible. In this manual a detailed steps and screenshots are mentioned so that it will be helpful for understanding how this tools are integrated and can be demonstrated at the same time. The required URLs and Readme files are added to the document as footnotes so do go through that if needed.

2 System Specification & Requirements

This section tells a detailed specification of the operating system that was used for the implementation of this project, which contains the system & software tools:

2.1 Operating System Specifications

- Processor: Intel Evo i7
- RAM: 16GB
- Storage: 512 SSD
- Operating System: Windows 11

2.2 Software Tools Specifications

- Visual Studio: Version-1.92.0 x64
- Java: Version- 8

3 Installation & Setput for Local Machine

3.1 Visual Studio Setup

VScode is a open source software which works as an Integrated Development Environment (IDE) where you can built a software and integrate it with any online platform like

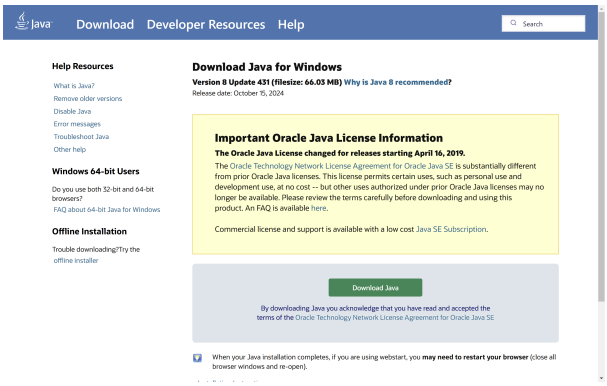
GitHub and many more, also it supports various programming language which needs to be installed with the help of the extension and can clone any project from Github Repository and work on it, the steps for installing and setting it up on the lcoal machine are showcased below:

- First download Visual Studio for Windows or Mac from Here. [1]
- Get it installed on your local machine.
- Once installed Launch and Sign-up.
- Configure VS code by installing Java Development Kit.

3.2 Java Installation

Java is installed for this project as the application which is used is java based so to makes any changes and work on this project java is required, so the steps are given below:

- Download Java for Windows from Here. [2]
- Install java packages with the help of extension in VS code.



Once the environment setup and installation is completed on the local machine, then the next step is platform setup.

[1] https://code.visualstudio.com/download
[2] https://www.java.com/download/ie$_manual.jsp$

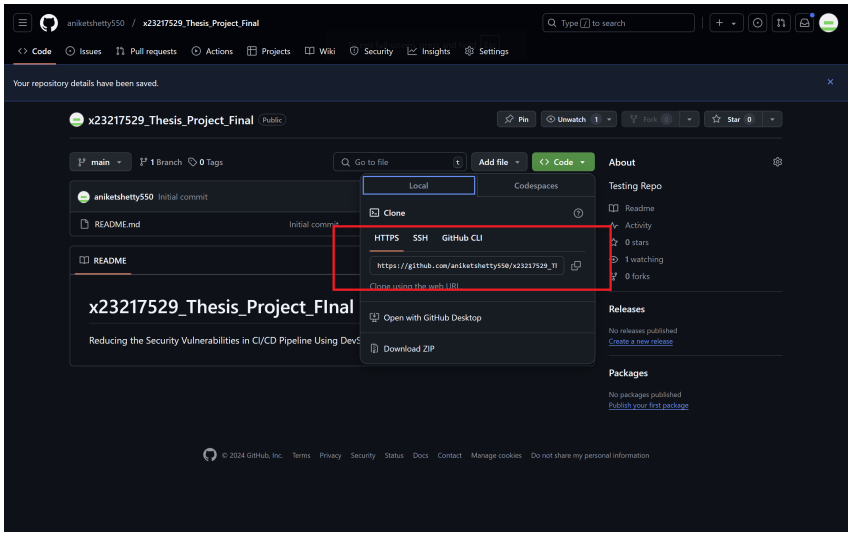
4 Platform Setup

Over here, all the open-source platforms which are used for this project are mentioned below and how they are integrated with VS code are showcased:

4.1 GitHub Setup and Integration

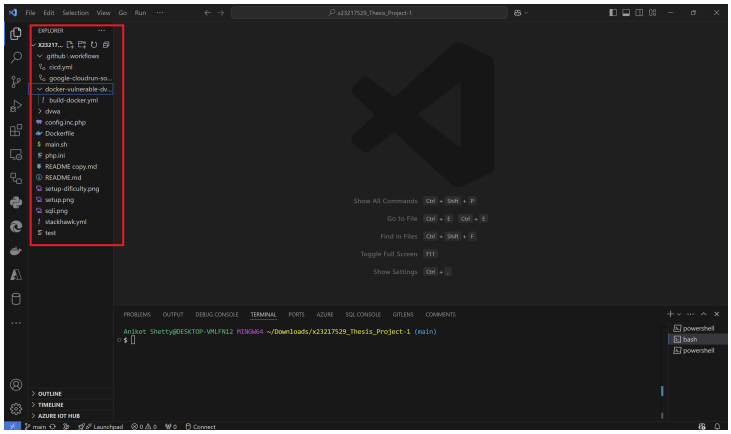
Github is an open source platform which works as a version control, as this service is used for running the application with the help of GitHub action CI/CD Pipeline, the steps are mentioned below for the set-up:

- Click on this link: [3] for accessing this platform.
- Create an account with your personal credentials.
- Create a private/public repository for your project by any name (example: Test Project) and after creating it might look something like the below figure:

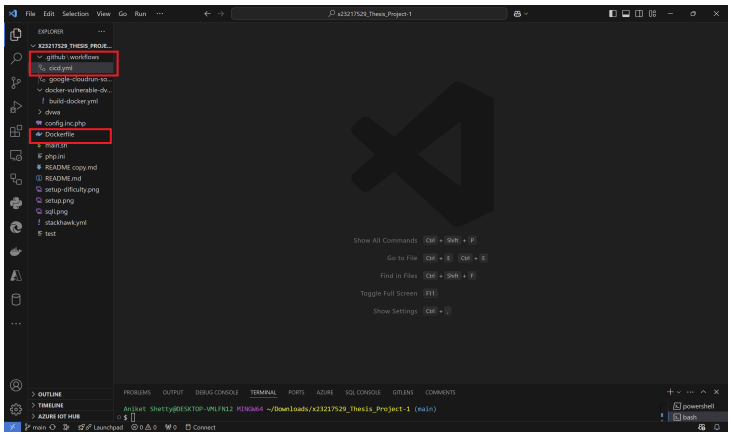


- Once your own repository (Test Project) is created, then the next step is to download the pre-build repository `https://github.com/aniketshetty550/x23217529_Thesis_Project` of the research project on the local machine, where all the source code and the integration of security tools, Google Cloud Platform is build.
- So, open the git repository which is provided above and follow the below process for downloading it to your local machine:
  - Click on the Code option which is green in colour as shown in the above diagram.
  - There is option for download at the last, click on it and download it.
- So, after downloading, unzip the folder and open it on your VScode.
- By following this process the source code of the project will be displayed on your VScode as shown in the below figure:

[3] https://github.com/



- So, once the source code is there on your local machine then the next step is to push that code to the newly created Github Repository (Test Project) which you have created previously.
- The steps for pushing the code to your private repository are showcased below:
  - git init
  - git add .
  - git commit -m “First Commit”
  - `git remote add origin <repository-url>` (Over here instead of you repository url you need to mention the github repo link which you have generated for your project)
  - git push -u origin main
  - By doing this, the code will be sent to your private github repository (Test Project).
- To make this project working you need to configure Github Action for running the CI/CD pipeline and deploying the application on cloud. So, for this project already a `.github\workflow` is created which contains cicd.yml file also a Dokcerfile which contains the docker image of the application (Altro Mutual) as shown in the figure:

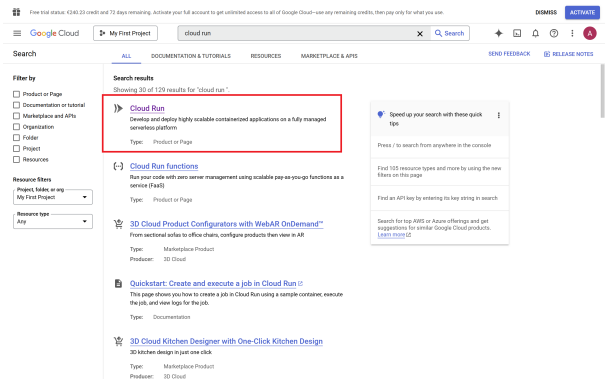


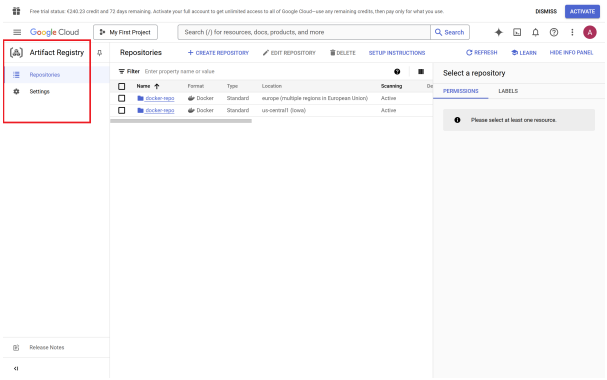
- So, now the next step is to integrate the Security tools and Google Cloud Platform by adding the secret key’s in the cicd.yml file of the VScode.

4.2 Google Cloud Platform Setup

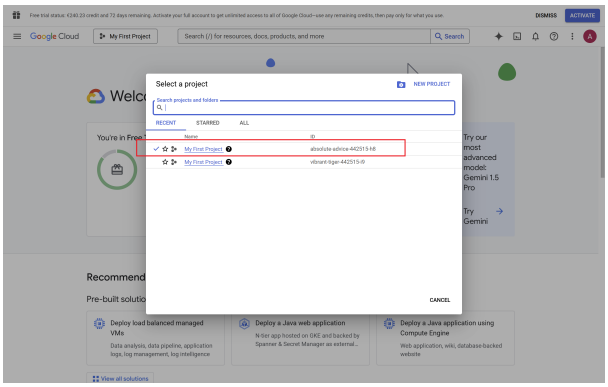
GCP is a open-source cloud platform which is easily accessible and is used for building, deploying and to run the application on cloud using various services which are provided by it. For this project two main services of GCP are used which are explained below with its steps for setting it up for this project:

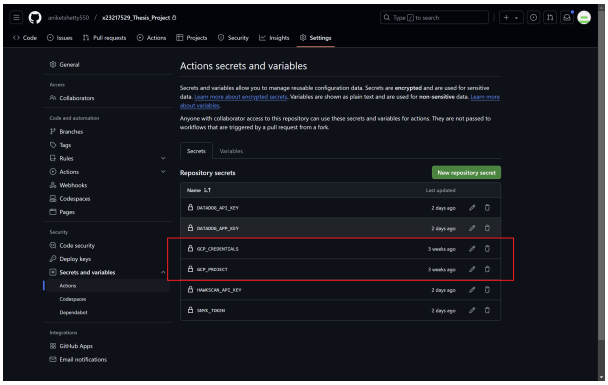
- First go to this link `https://cloud.google.com/?hl=en` where you need to login with your personal credentials.
- After that the next step is to enable the Cloud Run and Artifact Registry services. Which looks like this after enabling it:



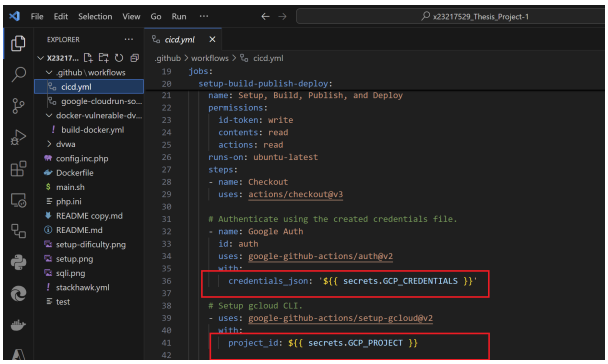


- Cloud Run is used for the deployment purpose and Artifact Registry is used for storing the docker images which are generated for the application.

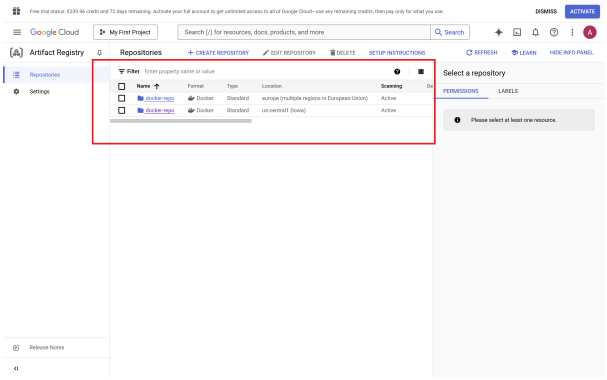




- The project id which is genearated (as shown in the figure) at the GCP console needs to be added to Github Secrete Key as a GCP_PROJECT.
- By doing this step the secrete key (GCP_PROJECT) can be used in the cicd.yml file of the project to integrate GCP services and authenticate the profile.



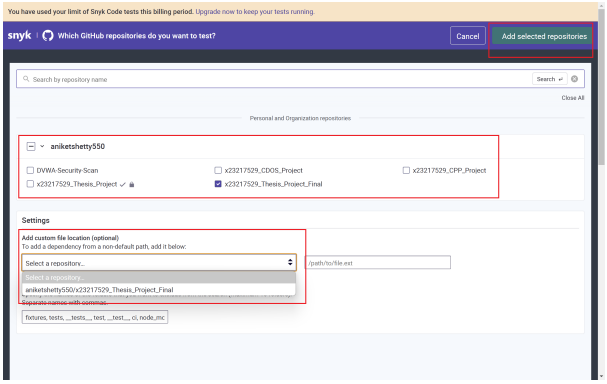
- The images which are build in the Dockerfile for the application are stored in the Artifact Registry of the GCP as shown above figure:



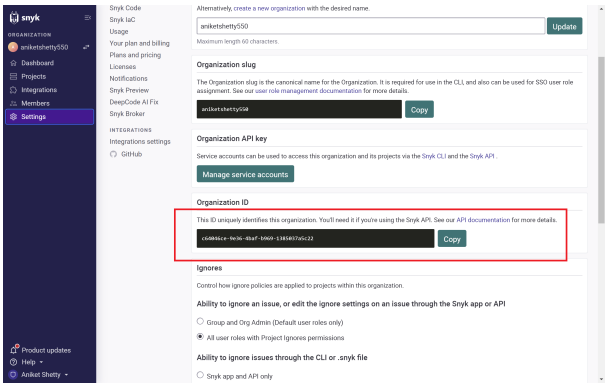
4.3 Synk Integration

Synk Tool is an open source security testing platfrom that is used for detecting the vulnerabilities of the source code in a application. The integration steps of Synk with Github repository and VScode is showcased below:

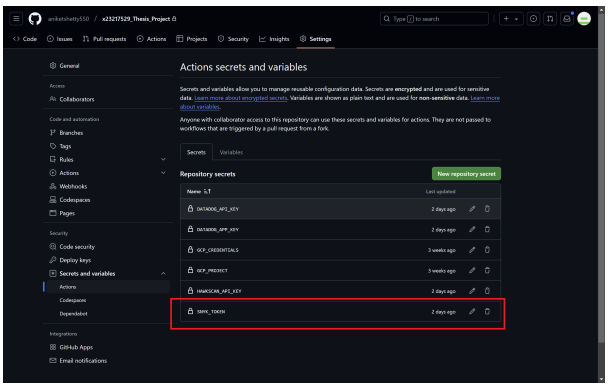
- Go to the Synk platform by clicking on this link `https://snyk.io/?utm_medium=paid-search&utm_source=google&utm_campaign=dm_goog-ps_aom_240916_emea-brand&utm_content=br_ex&utm_term=snyk&gad_source=1&gclid=CjwKCAiA6t-6BhA3EiwAltRFGMaWtp6M1NhRmQQ5lCZOlB-fZXiNu-vn9CKzoH6aG-9WL016choC__wQAvD_BwE`



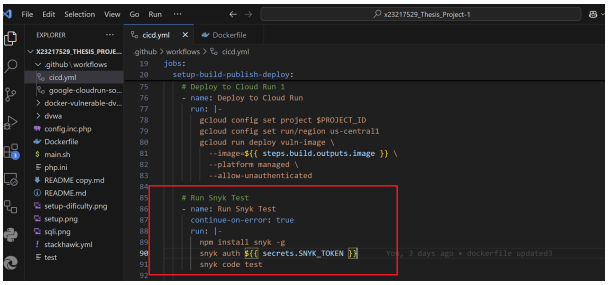
- You need to login with the help of your Github account and clone the repository (Test Project) which you have created for this project, so that the synk account will be created as shown in the figure above:



- Once you have cloned your Github Repository with Synk tool then the next step is to get the API key of Synk and add it to Github Secretes as shown in the figure:



- So, after taking the API key from Synk tool and adding it to Github repository, it also needs to be added in the cicd.yml file, so when the Github Action pipeline gets triggered, the Synk tool automatically gets activated and the logs are generated at the Synk dashboard.

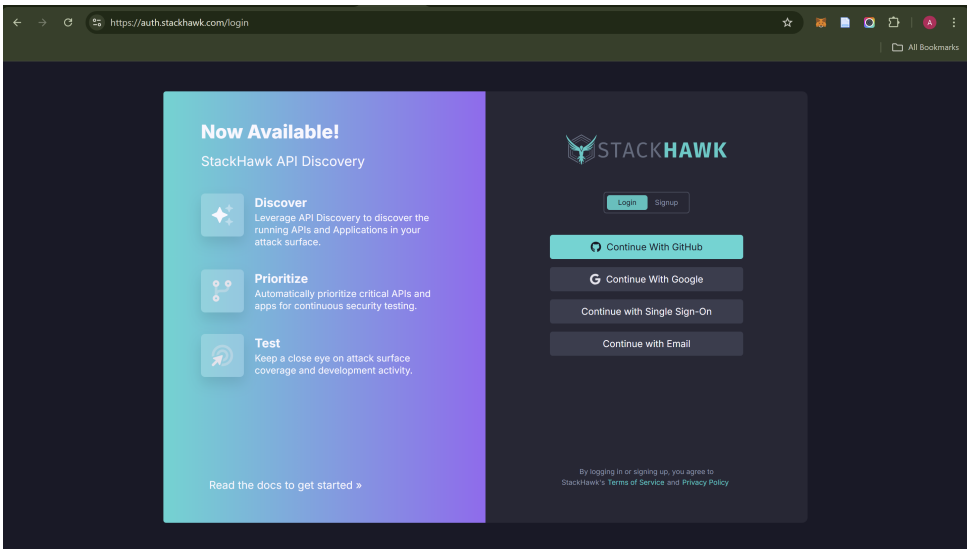


- So, above is the screenshot of VScode, where the Synk tool integration is done in the cicd.yml file and secrete API key is added.
- By following this steps the integration of Synk tools gets completed.

4.4 Stackhawk Integration

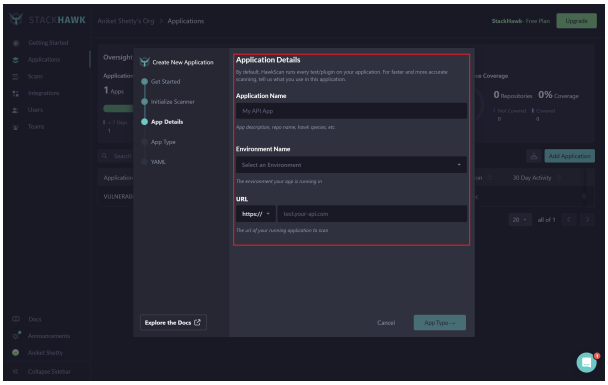
Stackhwak is aslo an open-source tool which is basically used for detecting the vulnerabilities in an application in real time as it works dynamically. So to integration steps for Stackhawk in the CI/CD pipeline are showcased below:

- To start the integration click on the link. [4]
- The UI for stackhawk looks like the picture which is attached below.



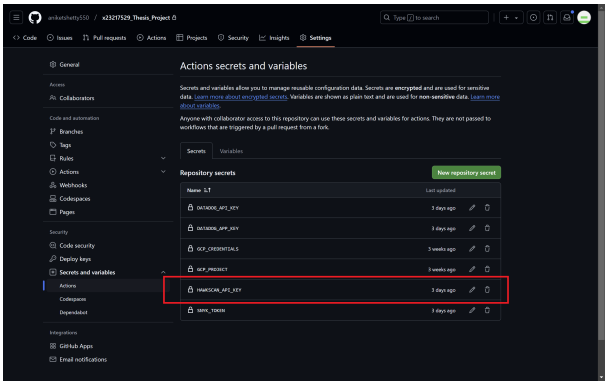
- So, to create an account for stackhawk there is an option to clone the github repository to the stackhawk account.
- Once the account is been cloned then the next step is to create a Application ID at the stackhawk portal for the application that is deployed on cloud.
- There are three steps for creating the Application id for the stackhawk which are mentioned below and there is a screenshot which will help to understand the process in more correct manner:

[4] https://www.stackhawk.com/

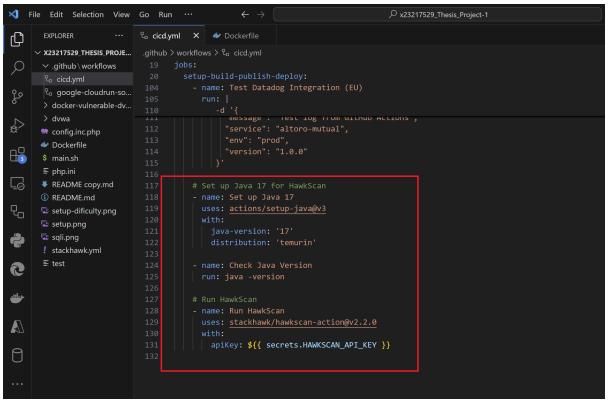


  - Select the Application Name.
  - Select the Environment (Production, Development & Pre-prodution)
  - At last select the URL of the deployed application. (This URL needs to be the url of application which is deployed on GCP Cloud Run)

- So after following this process on the Stackhawk platform then the Application ID which was created needs to be added to Github Secrete Key.



- Aslo after that, the secrete key which was created for stackhawk needs to be added in the cicd.yml file where the stackawk integration is been done as shown in the figure.

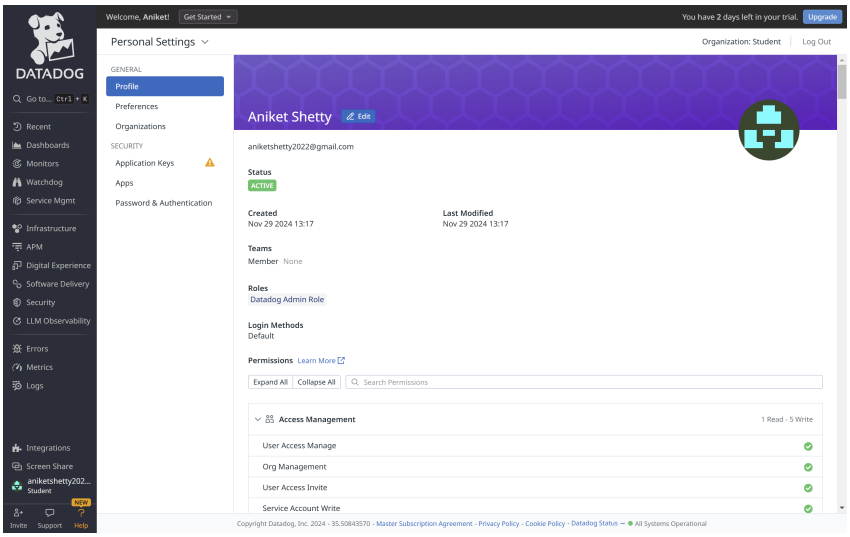


- By following this process the integration of the Stackhawk tool with the github and cicd.yml file at VScode is been done successfully.

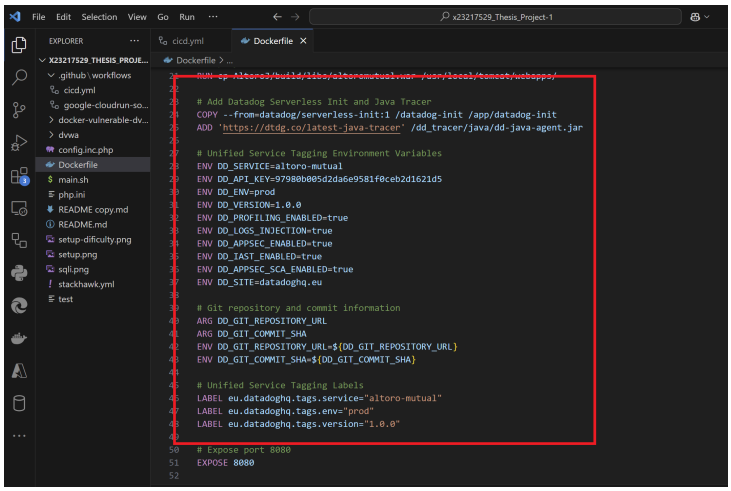
4.5 Datadog Integration

Datadog is an open source tool which works as a security testing and monitoring tool. In datadog IAST Code Security service needs to be enabled for detecting the vulnerabilities in the application, and the steps are showcased below:

- Check the link for Datadog portal. [5]
- Login with your personal details and the UI of datadog looks like this:

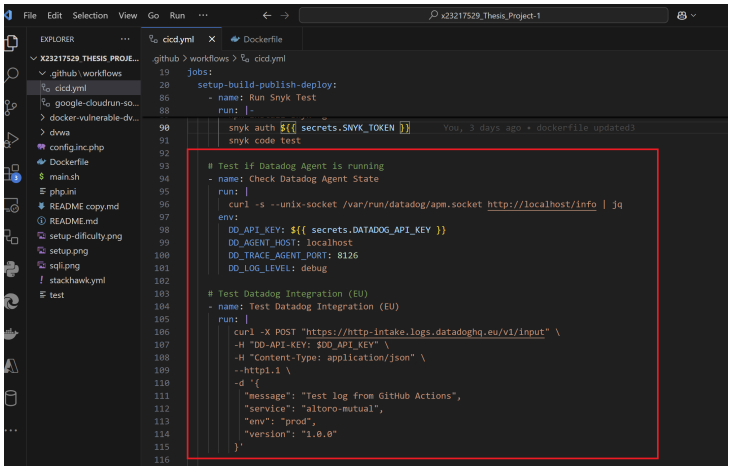


- The main part for integrating Datadog IAST is done in the VScode under Dockerfile and cicd.yml file where the agent is installed and all the datadog services are enabled and integration is done. Over here the service name which is mentioned is "altoro mutual" so which is showcased in the source code. So you need to mention the name as per your project.

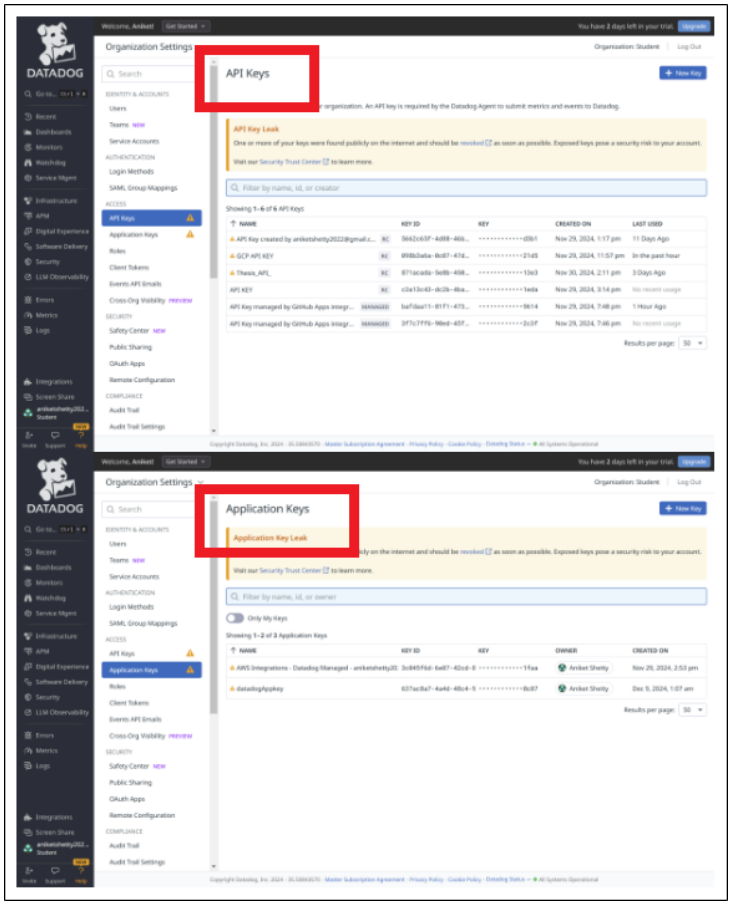


- In the first figure the source code of datadog is showcased which is located in Dockerfile, where the datadog agent is initialized and all the services like APM, Application Security: IAST, SCA are enabled.

[5] https://www.datadoghq.com/



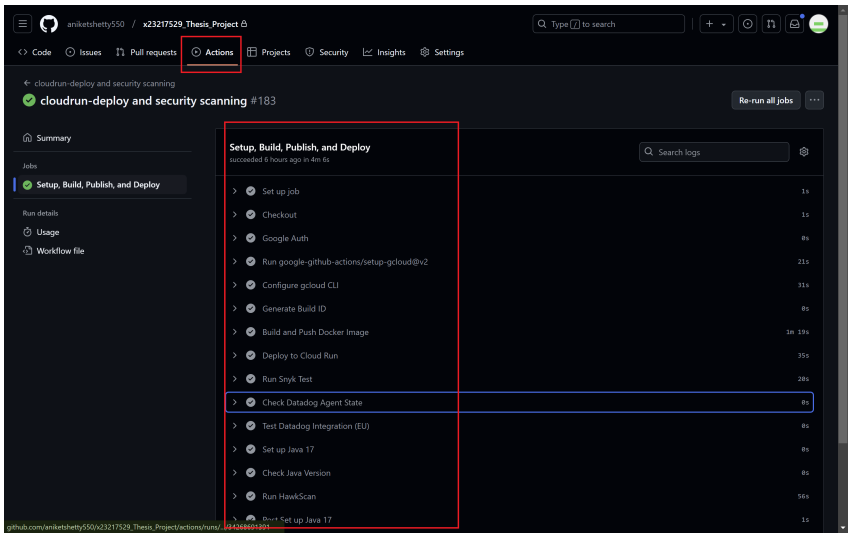
- And in the second picture the the datadog is been integrated in the cicd.yml file, where the API key and the APP Key is been added for datadog which was genearated while creating the account.
- The API and APP key for the datadog can be found from the below diagram and both this key needs to be added in the Github Secretes Key.



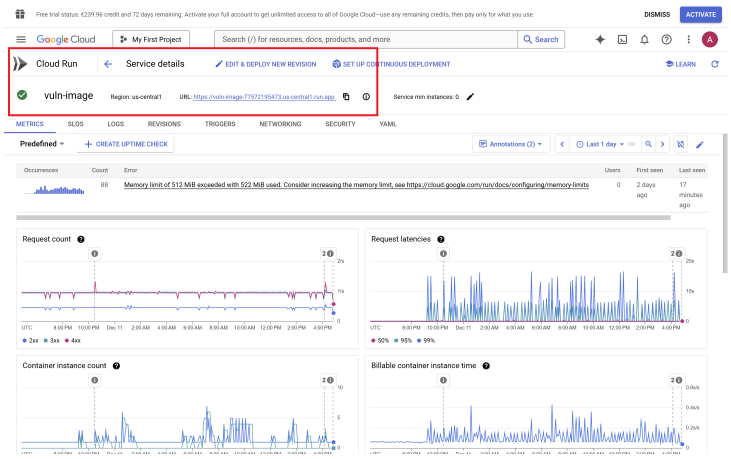
- So, once this key's are added to the Github account then the Datadog will be getting integrated for your project.
- By following this process the Datadog IAST service will be enabled.

5 Final Integration

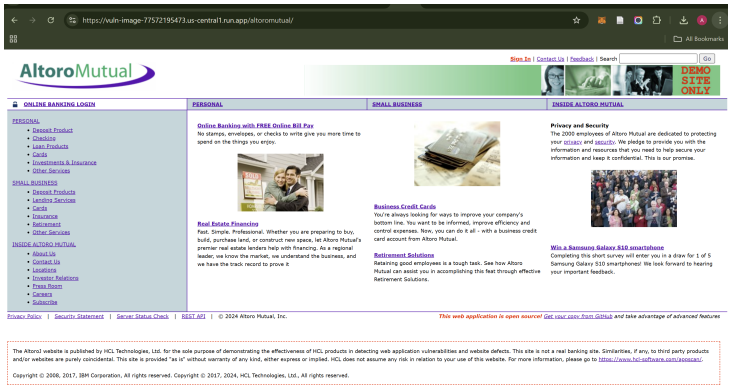
- After performing this entire process now all the tools and software platforms are integrated for this project. So, now the next step is to push the source code to the private Github repository that was been created previously (Test Project).
- Topush the code to the Github repository you need to use the terminal which is provided by the VScode and follow this commands:
  - git init
  - git add .
  - git commit -m “First Commit”
  - `git remote add origin <repository-url>` (Over here instead of repositort url you need to mention the github repo link which u have generated for your project)
  - git push -u origin main
  - By doing this, the code will be sent to your private github repository (Test Project).
- So once the source code is pushed to the github repository then the Github Action CI/CD pipeline gets triggered and the updated source code will be there on the git.
- By doing this process the CI/CD pipeline will run and all the steps which are mentioned in the cicd.yml file will be pushed and GCP, Synk, Stackhawk, Datadog tools will be getting triggered at the stages where they are integrated.
- The images shows the Github Action CI/CD pipeline where the pipeline is running:



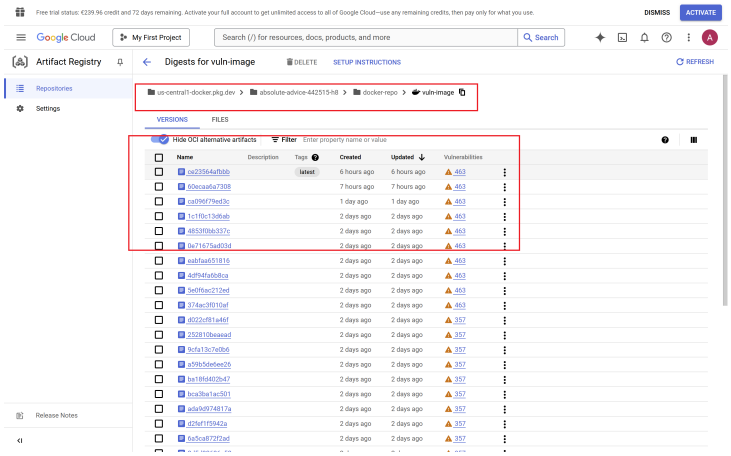
- Once the CI/CD pipeline runs completely then you need to check Google Cloud Platform, Synk, Stackhawk and Datadog tools, as logs will be generated and the application will be deployed on cloud.
- Starting with the GCP Cloud Run where the Application was deployed should look something like this which is highlighted in the picture:



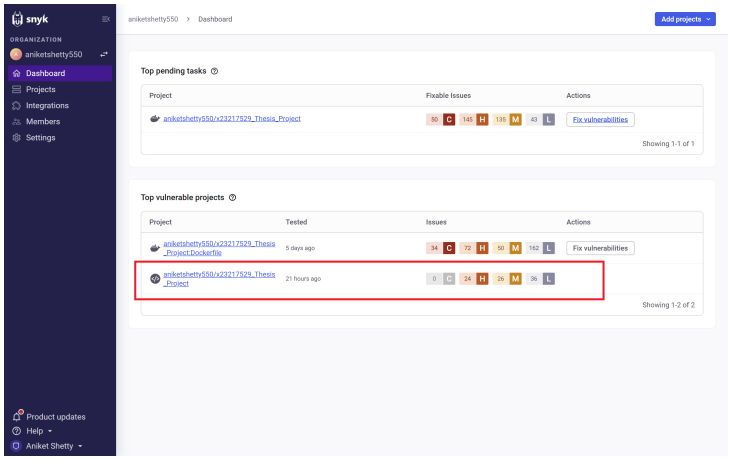
- The application which was deployed on cloud can be hosted with the help of the URL which was highlighted in the previous picture and it should look like this:



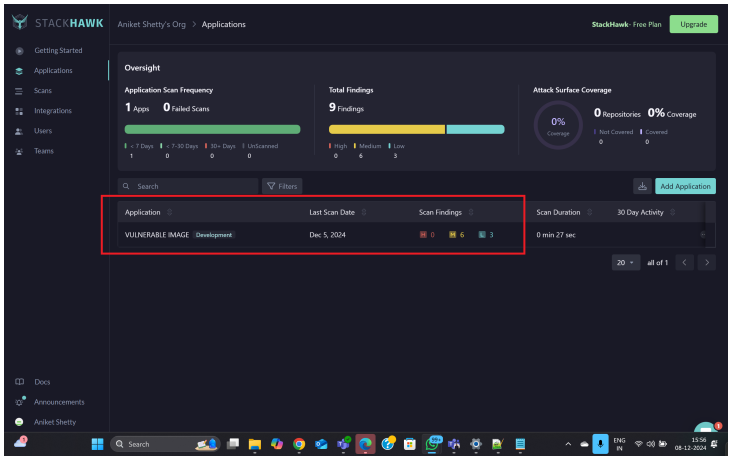
- Next need to check the Artifacts of the docker images which were generated by docker and were stored at the GCP artifact registry as shown in the figure:



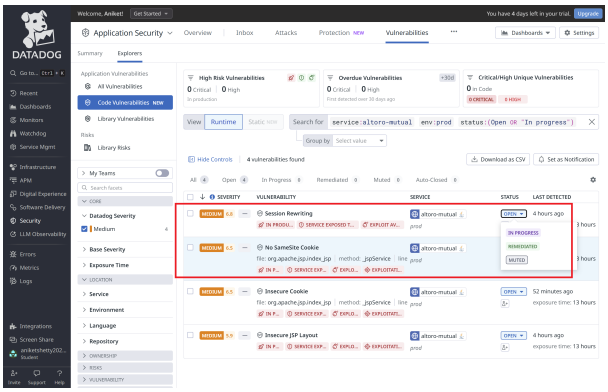
- Security tools Synk is first triggered and the static code analysis of the application is done on this platform and the results of detected vulnerabilities of the application are triggred over here which should look something like this:



- Next tool which was triggered was Stackawk and the results of detected vulnerabilities for the application should look something like this:



- The last tool that get triggered in this process is Datadog and the results should look similar to this, where the vulnerabilities are detected as shown in the figure:



By following this steps you will be able to get the provided results for this project once the implementation is completed.