National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Sri Lakshmi Durga Pavedemukala
Student ID: 22174681

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sri Lakshmi Durga Pavedemukala
Student ID:	22174681
Programme:	Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	272
Page Count:	7

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:	Sri Lakshmi Durga Pavedemukala
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

Sri Lakshmi Durga Pavedemukala
22174681

1 Introduction

This guide provides step-by-step instructions to help you set up and customize your system effectively. Whether you're a beginner or an expert, you'll find clear explanations and helpful tips for optimal configuration. Let's get started on making your setup seamless and efficient.

2 System Requirements

2.1 Hardware Requirements

CPU: Dual-core processor Intel Core i3
RAM: 8 GB
Storage: 10 GB free disk space
OS: Windows 10

2.2 Software Requirements

Python and PIP : Version 3.12 (*Python installation*; n.d.)
NPM : Version 9.00 (*NPM installation*; n.d.)
VSCode: Version 1.95
Docker Desktop Version: 4.3.60
Gitbash (*Gitbash*; n.d.)

2.3 Other Requirements

Private key from the Github App to be saved as privatekey.pem under the same folder as that of app.py
Active Azure subscription
Azure Container registry and AKS cluster provisioned

3 Installation

1. Download the Flask web app.
2. Extract the zip file to a desired location on your computer.

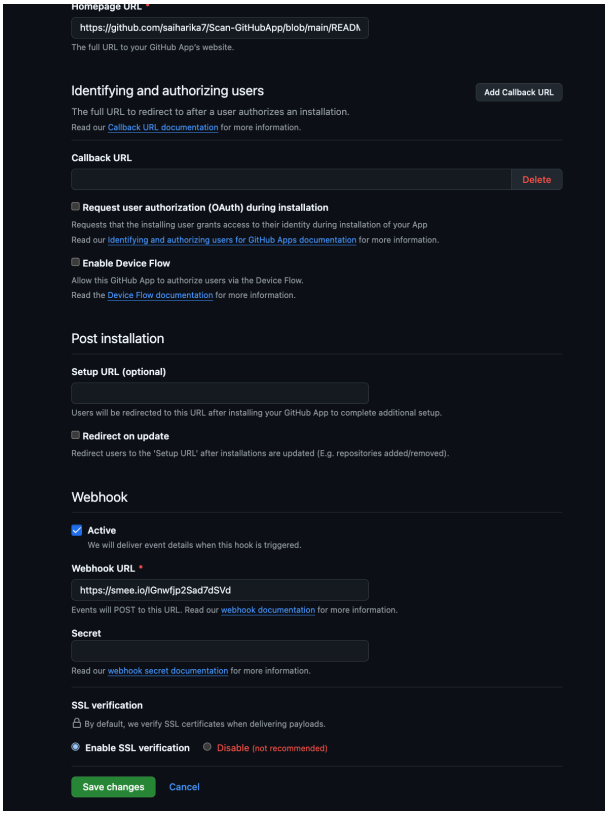

Figure 1: Configuring OAuth and Webhook Settings for GitHub App

3. Open VSCode and navigate to the extracted directory.
4. Install the requirements using pip3 install -r requirements.txt
5. Install the Smee for port forwarding npm install –global smee-client
4. Run the Flask web app Python file using Python 3.12 in VSCode:

sudo python3 app.py

5. Port forward using

smee -u https://smee.io/5AZhNDeV5aRs5U1D -t http://localhost:8080/webhook

3.1 Setup

1. Once the flask web app is running successfully on the local system with smee making our services available on internet a Github App needs to be created and installed on our repository which can be done by following the below steps
1) Visit https://github.com/settings/apps after logging in your account
2) Click on New Github App
3) Enter the required information by passing the smee URL in the Webhook URL param
4) Further the permissions needs to be provided which are as follows
Repository permissions –

Actions – Read Only
Checks - Read and Write
Metadata – Read only
Workflows – Read and Write

Organization permissions –
Events – Read only

Subscribe to events
Push
Check run
Check Suite
Workflow run
Repository

5) Create a repository containing the Dockerfile similar to this one along with the Github actions file available under .github/workflows which would be used to do the releases
6) Create branch rules sets by going under the repository settings

https://github.com/username/reponame/settings/rules

With the below rules configuration
- Enforcement status – active
- Target branches – "main"
- - "release-*"
- - Default branch

Branch Rules
- Restrict creations
- Restrict updates
- Restrict deletions
- Require pull request before merging
- Require status checks to pass with additional settings with a selection of the Github App

- Blocks force pushes

7) Create Repo secrets by going to

https://github.com/username/reponame/settings/secrets/actions

which are as follows
1) ACR_SERVER – Azure container registry server URL
2) REGISTRY_PASSWORD – ACR PASSWORD
3) REGISTRY_USERNAME – ACR USERNAME

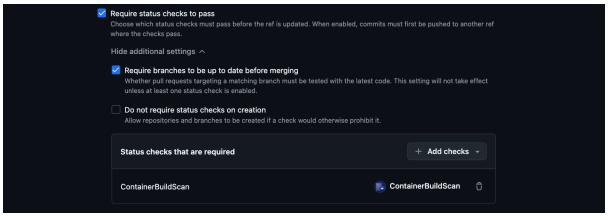

Figure 2: Branch Protection Rules and Status Check Requirements

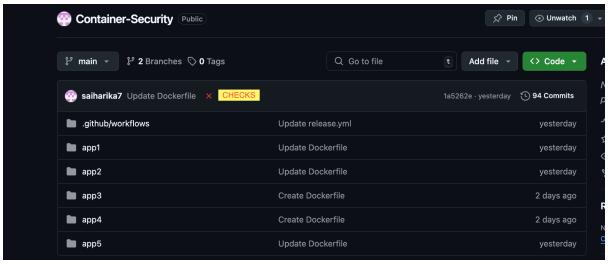

Figure 3: Container-Security Repository Overview with Recent Dockerfile Updates

4 Configuration

The Dockerfiles pushed to the Github needs to be first built locally ensuring they are able to be build properly precisely the files and folders required are also pushed properly in the respective app folder

The docker files can be tested by starting the docker service from the docker desktop by switching to linux or windows containers as per the requirement

5 Usage

Once the Github repository is created and doccker file is pushed checks will be shown at the top

When this check is clicked it will show the details about the failure and the path of dockerfile

Further a similar result would be seen whenever a pr is raised

For scanning the images the workflow needs to be triggered from the main branch

the pipeline further detects the cve's and lists them in the logs

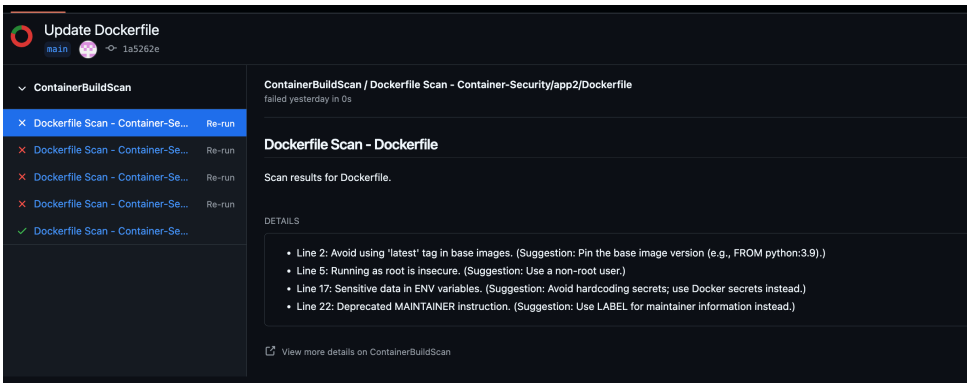

Figure 4: Dockerfile Scan Results for Container-Security/app2

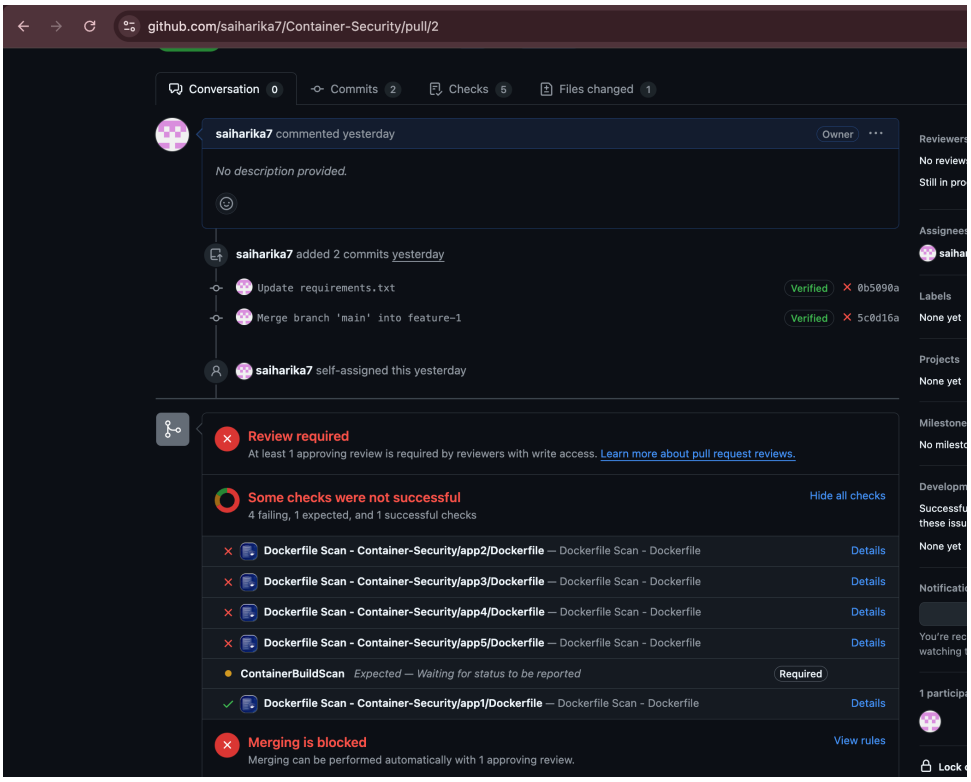

Figure 5: Pull Request Status with Dockerfile Scan Checks

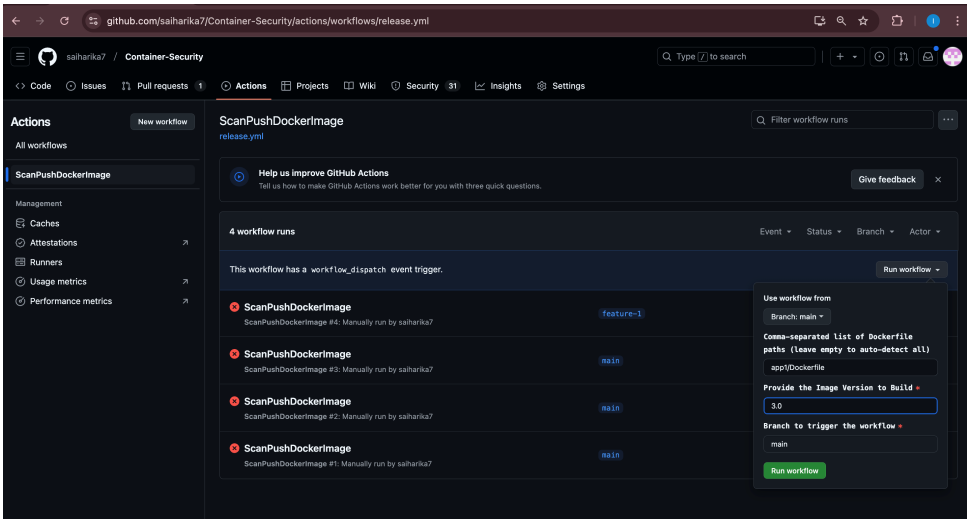

Figure 6: GitHub Actions Workflow - Scan and Push Docker Image

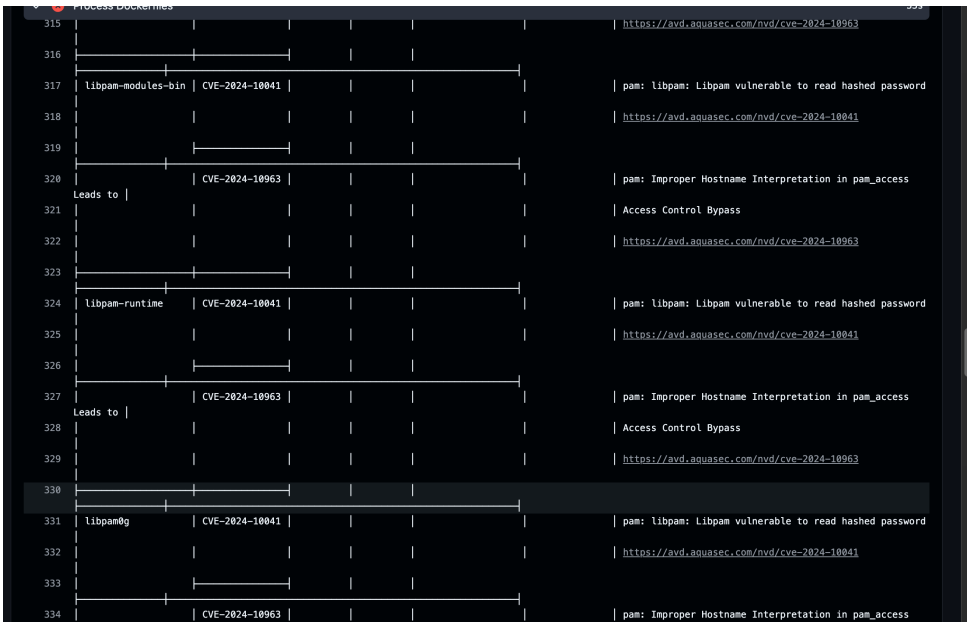

Figure 7: CVE Vulnerability Details for Docker Images

6 Troubleshooting

If you encounter any problems while using the Flask web app the following troubleshooting tips:
1. Make sure the required libraries (e.g., flask, etc) are installed before running the model..
2. Docker Issues: Ensure Docker Desktop is running, and the Docker daemon is properly initialized.
3. if any issues in checks not showing check for appprpriate permissions for github app

References

Gitbash (n.d.).
URL: *https://git-scm.com/downloads*

NPM installation (n.d.).
URL: *https://docs.npmjs.com/downloading-and-installing-node-js-and-npm*

Python installation (n.d.).
URL: *https://pip.pypa.io/en/stable/installation/*