

Configuration Manual

MSc Research Project
Cloud Computing

Kamrun Nahar Ali
Student ID: 21139474

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Kamrun Nahar Ali

Student ID: 21139474.....

Programme: MSc. In Cloud Computing..... **Year:** 2022-2023

Module: Research Project.....

Lecturer: Vikas Sahni.....

Submission Due Date: 15/12/2023.....

Project Title: A comprehensive Secure Serverless Container-based Architecture (SSCAR)

Word Count:1017..... **Page Count:**11.....

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: Kamrun Nahar Ali.....

Date: 15/12/2022.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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):	

SSCAR Configuration Manual

Kamrun Nahar Ali
Student ID: 21139474

1 Introduction

The research paper introduces a secure serverless containerized architecture as a way to lessen the infrastructure management burden placed on web applications. The research project setup and testing would be made easier with the help of this configuration manual. It gives a thorough explanation of the required installations, steps, and prerequisites. The organization of this document is as follows. The requirements for system configuration are described in Section 2. Sections 3 to 6 explain how to code and configure the application, and Section 7 shows how to test the completed program.

2 Prerequisite Installation for Development Environment

- a. Download and install Visual code studio editor in a local machine¹².
- b. Create Azure account ³.
- c. Create GitHub account and repository⁴.
- d. Create an account in Docker Hub⁵.

3 Connect Azure with local VS code editor.

Setup of Azure account extension done using official Azure document ⁶.

4 Create File system and push to GitHub.

The file structure had been created in local with VS Code and activated virtual environment ⁷. For baseline the blank file system had been moved to the Git repository as shown with following commands:

¹ <https://visualstudio.microsoft.com/vs/>

² <https://learn.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2022>

³ <https://portal.azure.com/>

⁴ <https://github.com/>

⁵ <https://docs.docker.com/desktop/get-started/>

⁶ <https://learn.microsoft.com/en-us/azure-stack/user/azure-stack-dev-start-vscode-azure?view=azs-2206#set-up-the-azure-account-extension>

⁷ <https://code.visualstudio.com/docs/python/environments>

```
git clone <repository link from github>
git branch
git checkout <branch name>
git checkout Master
git pull <branch name>
git pull Master
git status
git add <file name>
git commit -m "Message"
git push origin <branch name>
git push origin Master
git tag -a latest 51db77f or git tag -a latest
git push origin latest
git push origin *
git tag -d <tag name>
```

Figure 1a

```
git checkout Master
git add *
git status
git commit -m "Azure Blob listener and
file movement"
git push origin Master
```

Figure 1b

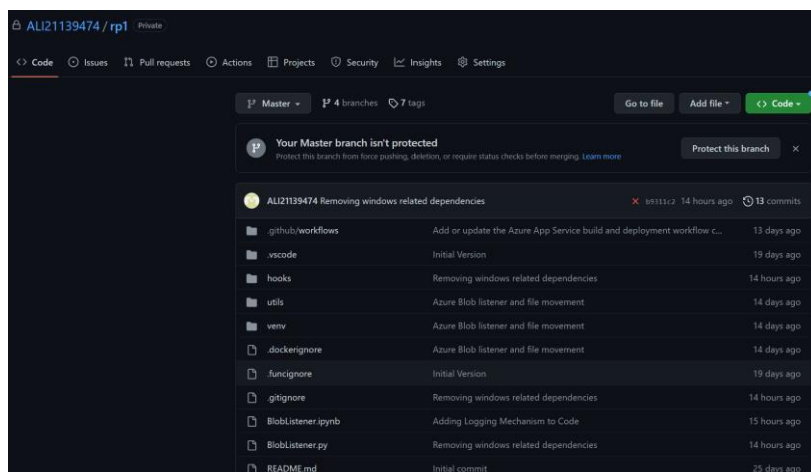


Figure 1c

The GitHub repository is at <https://github.com/ALI21139474/rp1>

4.1 Code Editor and Code Repository

Virtual Studio Code had been installed in Windows OS to run and edit python code and docker files smoothly during **development, testing deployment** of the code and containers.

5 Docker image and repository

- 5.1 Using VS Code Editor, a python function has been written with checkblob and moveblob library which will detect file in Azure blob storage and move from one source folder to another target folder. Figure 2a represents the code snippet.

```
with open(os.path.join('/rp1/logs', 'logfile.txt'), 'w') as f:
    run=0
    while(run<=3):
        run+=1
        print(f"Attempt #{run}: Checking for File")
        f.write(f"Attempt #{run}: Checking for File")
        if co.check_blob_obj()==1:
            print("File Detected in Source Azure Blob Container")
            f.write('File Detected in Source Azure Blob Container')
            mb.move_blob_obj(srcContainer = "containerrp1",
                             tgtContainer = "archivedfilerp1",
                             srcfile="srcFile.csv")
            print("File is archived now")
            f.write("File is archived now")
```

Figure 2a

- 5.2 Created docker container using VScode CLI⁸. Bellow set of commands are used to build, tag, and run docker images. Figure 2b shows the commands to build and run the docker image. Figure 2c is the content of dockerfile.

```
docker build --tag ali21139474/python-docker-rp1 .
docker tag ali21139474/python-docker-rp1 ali21139474/python-
docker-rp1:v1.0.0
docker tag ali21139474/python-docker-rp1 ali21139474/python-
docker-rp1:v3.0.0
docker image push --all-tags ali21139474/python-docker-rp1
docker run ali21139474/python-docker-rp1:latest
```

Figure 2a

```
rp1 > dockerfile > ...
1 FROM docker.io/library/python:3.8-slim-buster
2 WORKDIR /rp1
3 COPY requirements.txt requirements.txt
4 RUN pip install --upgrade pip
5 RUN pip install -r requirements.txt
6 RUN groupadd -r DEV
7 RUN useradd -r -g DEV DEV
8 USER DEV
9
10 COPY . .
11 CMD [ "python", "-m", "BlobListener"]
```

Figure 2b

⁸ <https://docs.docker.com/engine/reference/commandline/create/>

5.3 Implemented docker scan for vulnerability scan before docker build, shown in Figure 3.

```
PS C:\Users\Kamrun Ali\Documents\StudyMaterial\3rd Sem\rp1\rp1> docker scan --file Dockerfile --exclude-base ali21139474/python-docker-rp1:latest
WARNING! Partially defined environment, please ensure to provide both SNYK_INTEGRATION_NAME and SNYK_INTEGRATION_VERSION together!

Testing ali21139474/python-docker-rp1:latest...

Package manager: deb
Target file: Dockerfile
Project name: docker-image|ali21139474/python-docker-rp1
Docker image: ali21139474/python-docker-rp1:latest
Platform: linux/amd64
Base image: docker.io/library/python:3.8-slim-buster

✔ Tested 94 dependencies for known vulnerabilities, no vulnerable paths found.

Currently, we only offer base image recommendations for Official Docker images

For more free scans that keep your images secure, sign up to Snyk at https://dockr.ly/3ePqVcp
```

Figure 3

5.4 Docker image has been created and pushed to docker registry, as shown in Figure 4 and Figure 5.

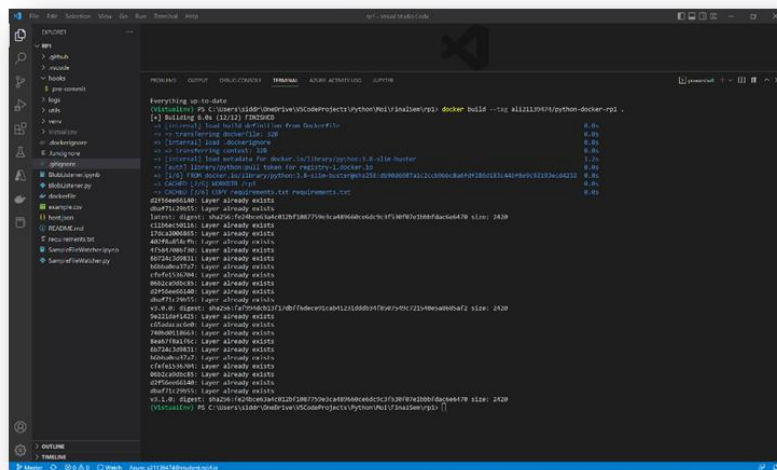


Figure 4

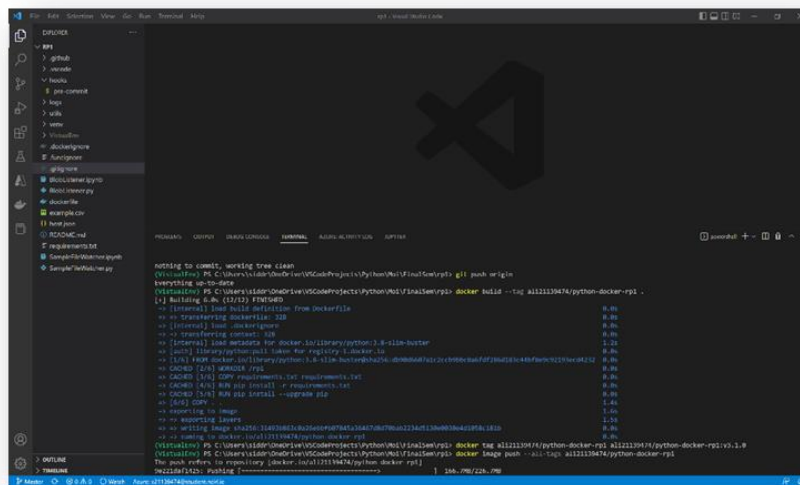


Figure 5

- 5.5 Docker file has been modified to create a non-root user while running the docker container, Figure 6 shows the code snippet.

```
FROM docker.io/library/python:3.8-slim-buster
WORKDIR /rp1
COPY requirements.txt requirements.txt
RUN pip install --upgrade pip
RUN pip install -r requirements.txt
RUN groupadd -r DEV
RUN useradd -r -g DEV DEV
USER DEV

COPY . .
CMD [ "python", "-m", "BlobListener" ]
```

Figure 6

6 Azure Environment Setup:

- 6.2 Created a student Azure account and created a new resource group and selected appropriate region as per Azure guidelines⁹ for this study. Local VS code editor has been linked to Azure account using Azure student account credentials.
- 6.3 Azure blob storage created for storing produced input and output files. A standard Locally Redundant Storage in Norway East region had been produced under a newly created region.
- 6.4 Azure Function has been created which will detect placement of file as an event (Figure 7).

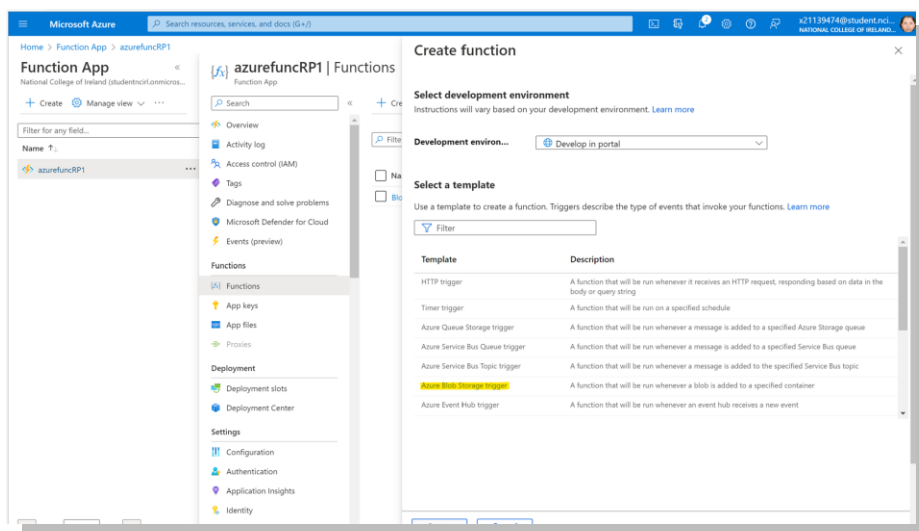


Figure 7

⁹ <https://azure.microsoft.com/en-us/explore/global-infrastructure/geographies/>

6.5 Azure Container Instance has been created as shown in Azure Figure 8.

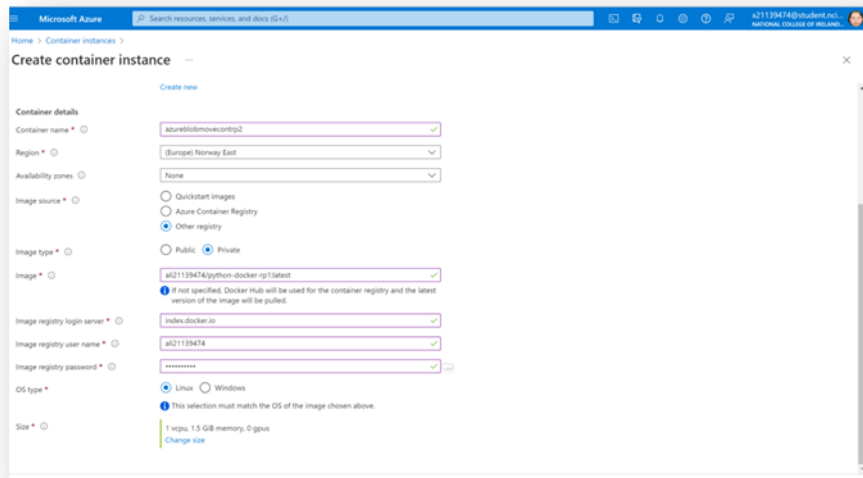


Figure 8

6.6 Azure log monitoring enabled using Azure portal (Figure 9a).

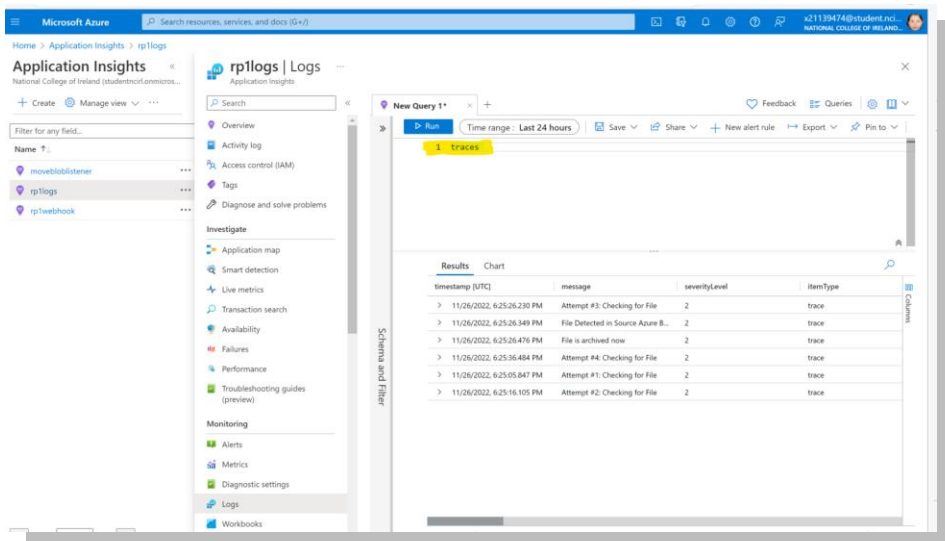


Figure 9a

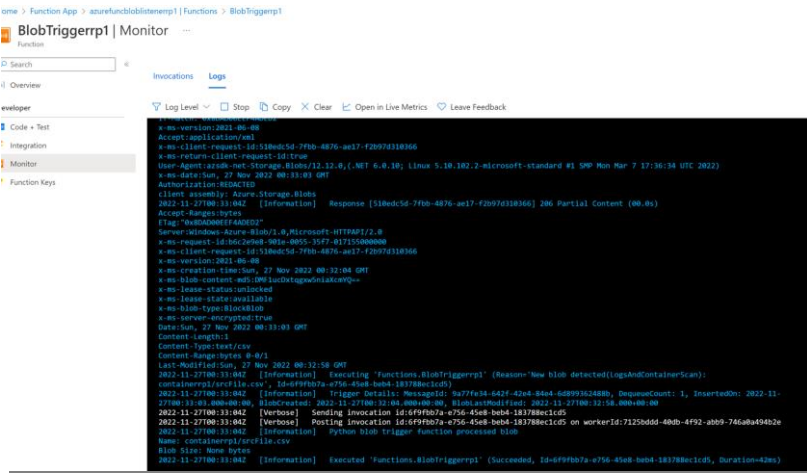


Figure 9b: Detailed Azure Log when event detected

6.7 Using Azure portal Role-based-access-control blob storage has been secured (Figure 10a & 10b).

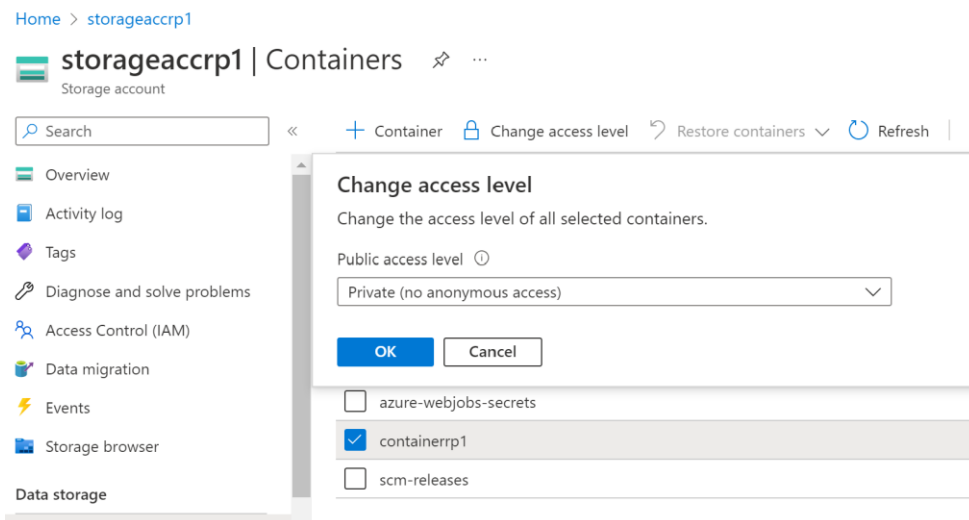


Figure 10a

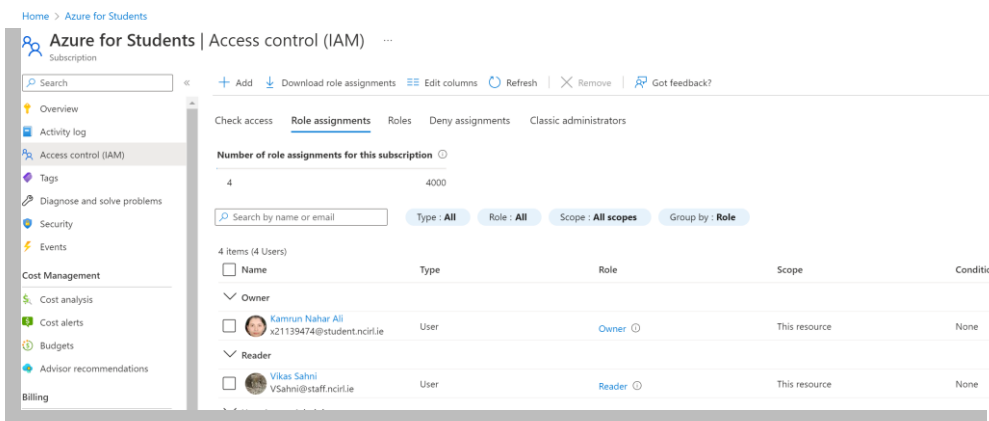


Figure 10b

7 Running Application

7.1 Once deployed start the container instance and Azure function in Azure portal as shown in Figure 11 and Figure 12, respectively.

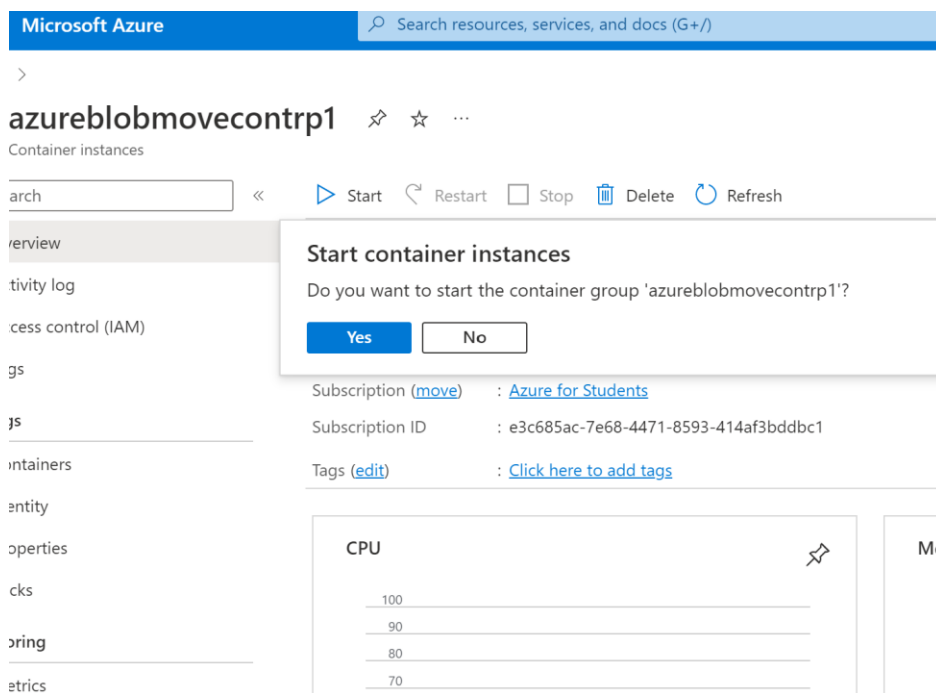


Figure 11

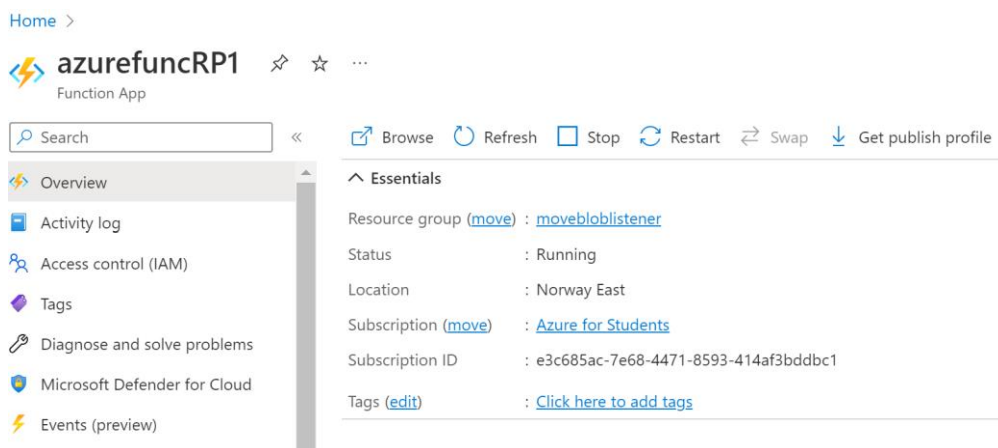


Figure 12

7.2 Now authorized blob file upload would automatically trigger the event driven serverless file processing and the processed file would land to target blob storage.