National College of Ireland

Configuration Manual

MSc Research Project
Master of Science in Cloud Computing

Muhammad Abu Bakar Sani
Student ID: 17112044

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Muhammad Abu Bakar Sani

Student ID: 17112044

Programme: Master of Science in Cloud Computing **Year:** 2022

Module: MSc Research Project

Lecturer: Shivani Jaswal

Submission Due Date: 19/09/2022

Project Title: Configuration Manual

Word Count: 758 **Page Count:** 10

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: Muhammad Abu Bakar Sani

Date: 19/09/2022

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

Muhammad Abu Bakar Sani
17112044

1 Mobile Application Development

Android Studio 2020.3.1 IDE [1] is used to develop the android application as shown in figure 1. It been used for both, frontend, and backend of application development.

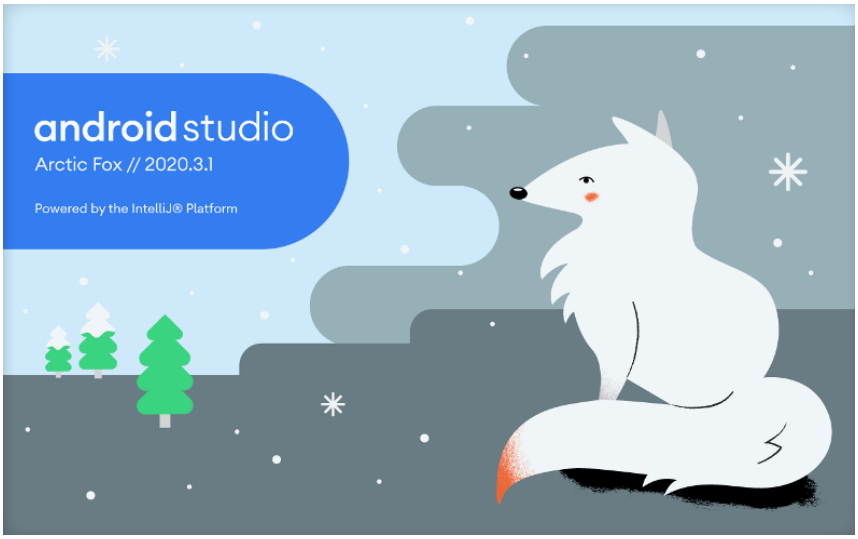

Figure 1: Android Studio

For version control GitHub [2] has been used which is integrated in Android Studio. New repository has been created in GitHub after creating a project in Android Studio then they both have been linked to make Push & Pull process easy, as shown in figure 2 & 3.

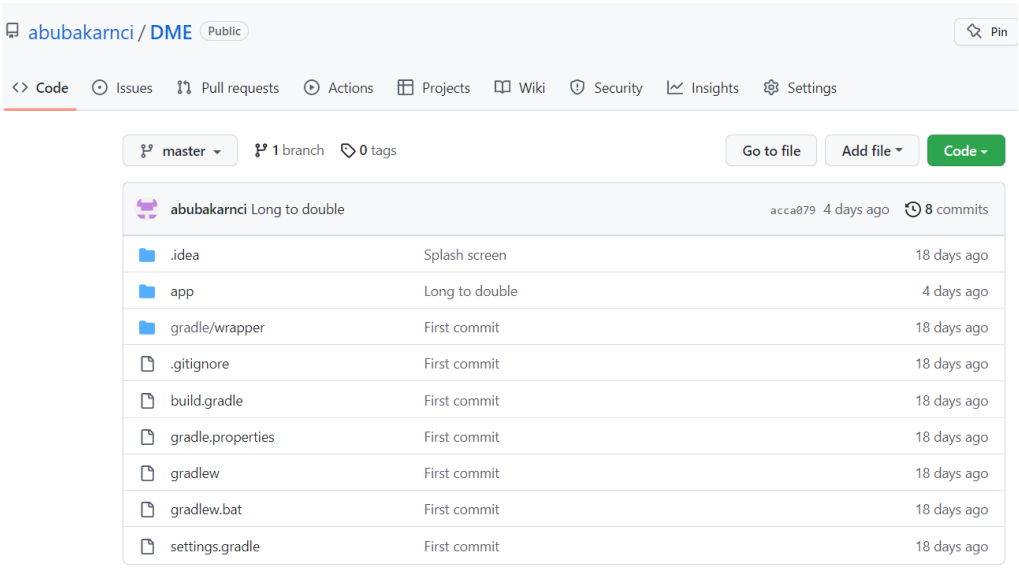

Figure 2: GitHub Repo

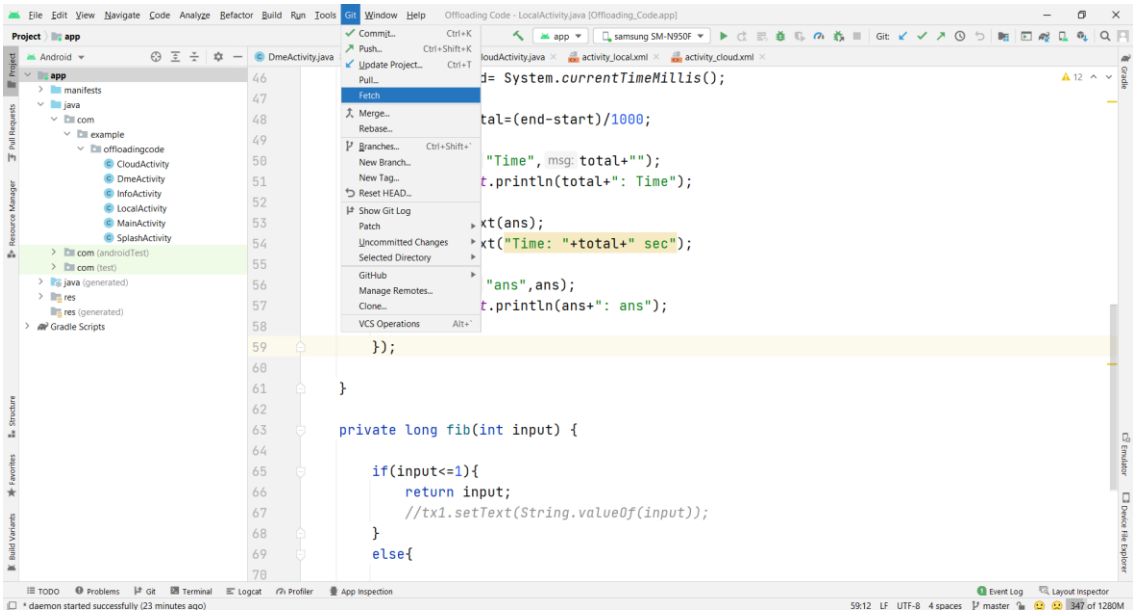

Figure 3: Android Studio and GitHub Integration

An android phone named Galaxy Note 8 has been used for the execution of this application as shown in figure 4. This IDE also gives an option to create multiple virtual devices.

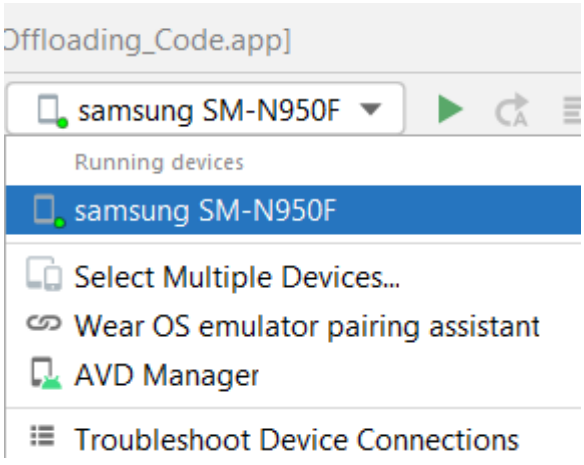

Figure 4: Device for execution

Default libraries are implemented in Application-level build.gradle but the volley library:1.2.1 has also been implemented as illustrated in figure 5 because this project involves performing HTTP requests and parsing JSON responses.

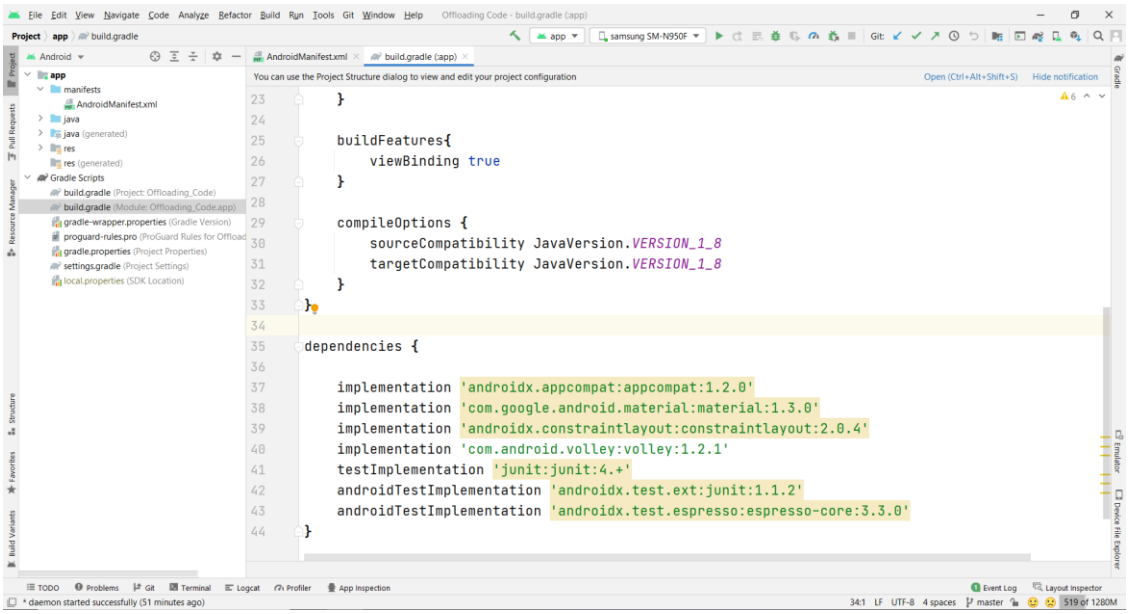

Figure 5: Android application - Build gradle (app level)

Permissions have been defined in Android Manifest file as shown in figure 6.

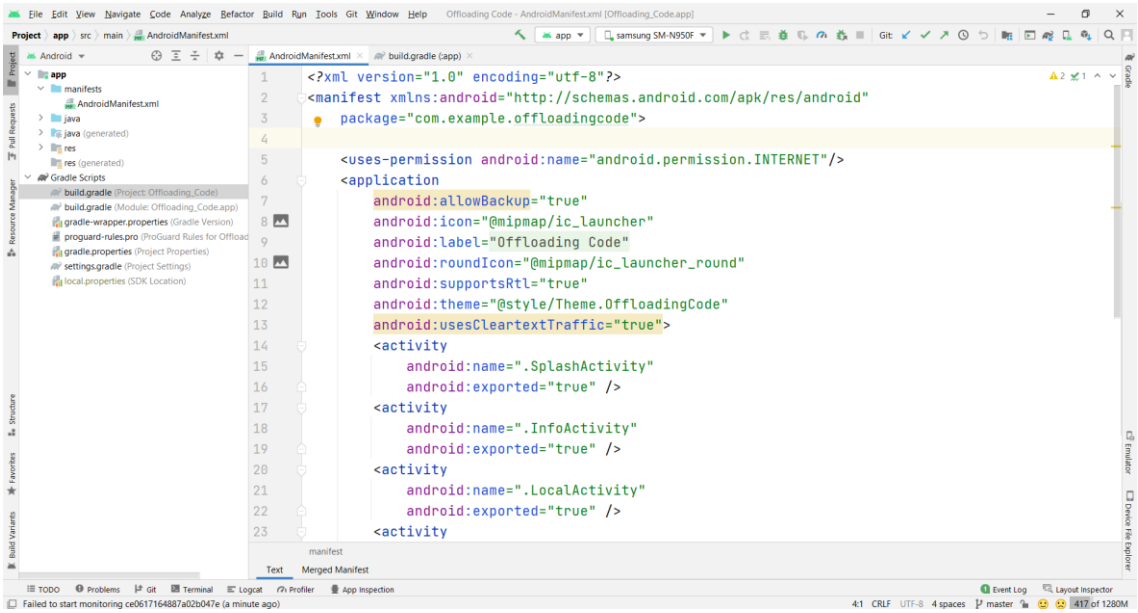

Figure 6: Android application - Manifest.xml

After setting up libraries and giving permissions, required Java and XML classes need to be created as Activities in Android application project structure.

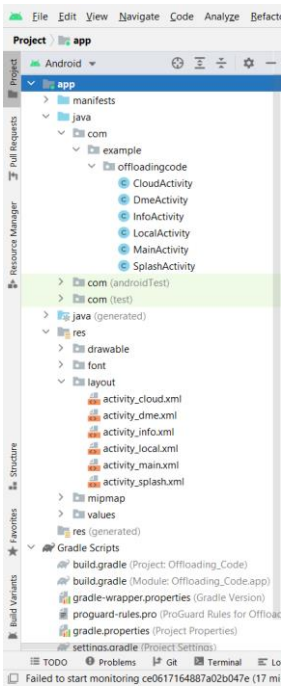

Figure 7: Android Application structure

Figure 8 shows android profiler that has been used to see live context of a device when an application is running. It's a prebuild feature in android studio.

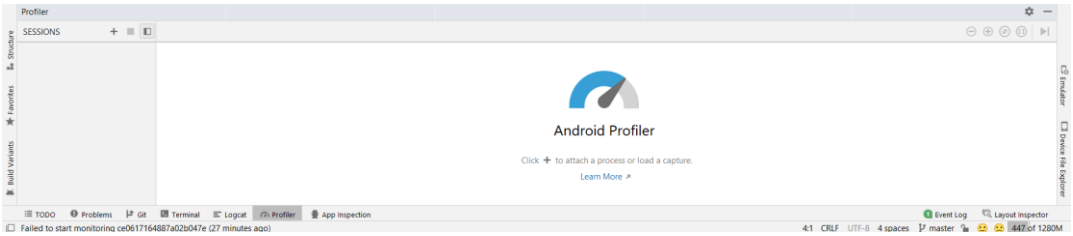

Figure 8: Android Profiler

2 Microservice Development

Eclipse IDE [3] is used to develop Fibonacci microservice for cloud execution. This version of Eclipse is specifically designed to develop web services. Maven needs to be installed as well for this part of the research.

Figure 9: Eclipse

Spring boot framework is used to develop this microservice. Figure 10 shows how to setup Spring Starter Project in eclipse. Appropriate application name needs to be given for good practice.

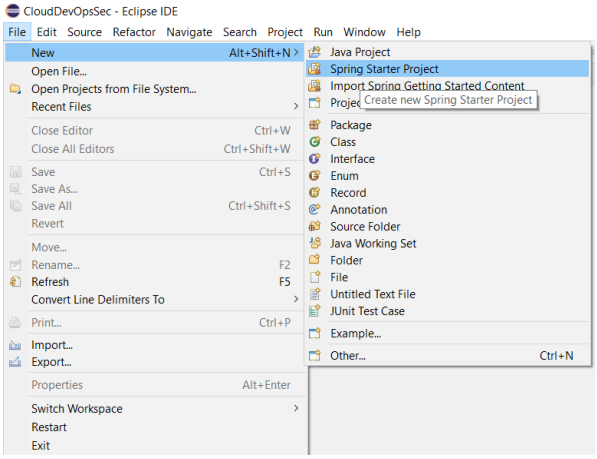

Figure 10: Spring Starter project

As figure 11 shows all packages and Java files created under main folder. These are all files which contain main content.

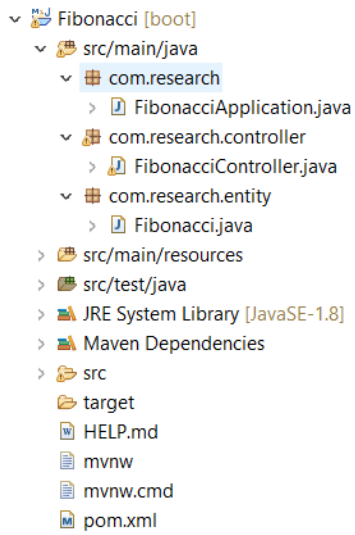

Figure 11: Core files

It can be executed locally on a localhost by selecting project and then by clicking on a play button, present in a nav bar as demonstrated in figure 12.

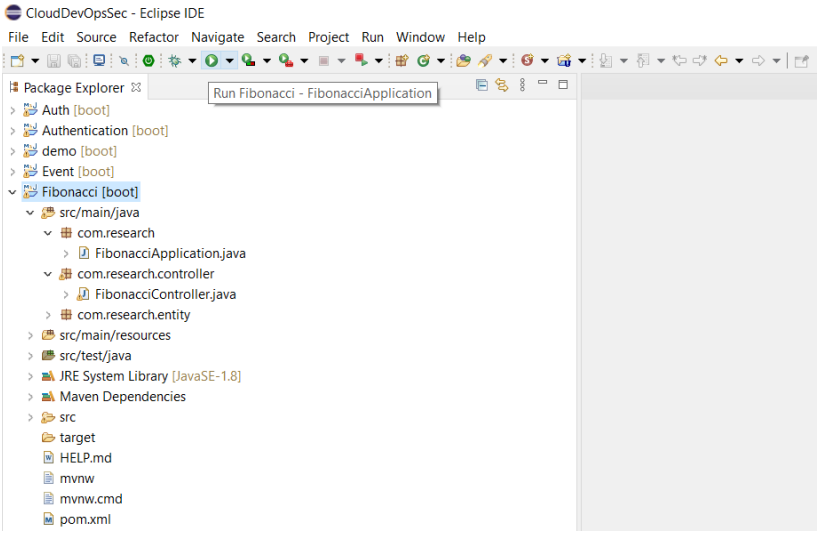

Figure 12: Microservice execution

It's necessary to change port number to 5000 before deploying it to cloud for security seasons. It needs to be done on application.properties file as shown in figure 13.

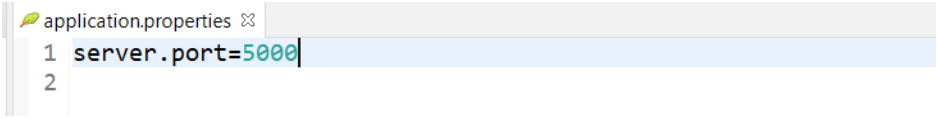

Figure 13: Port change

To deploy this microservice to cloud, its required to convert this spring boot project to JAR file. This can be done by running this command "mvn clean install" on cmd under project's folder as shown in figure 14. The JAR file will then be formed under target folder.

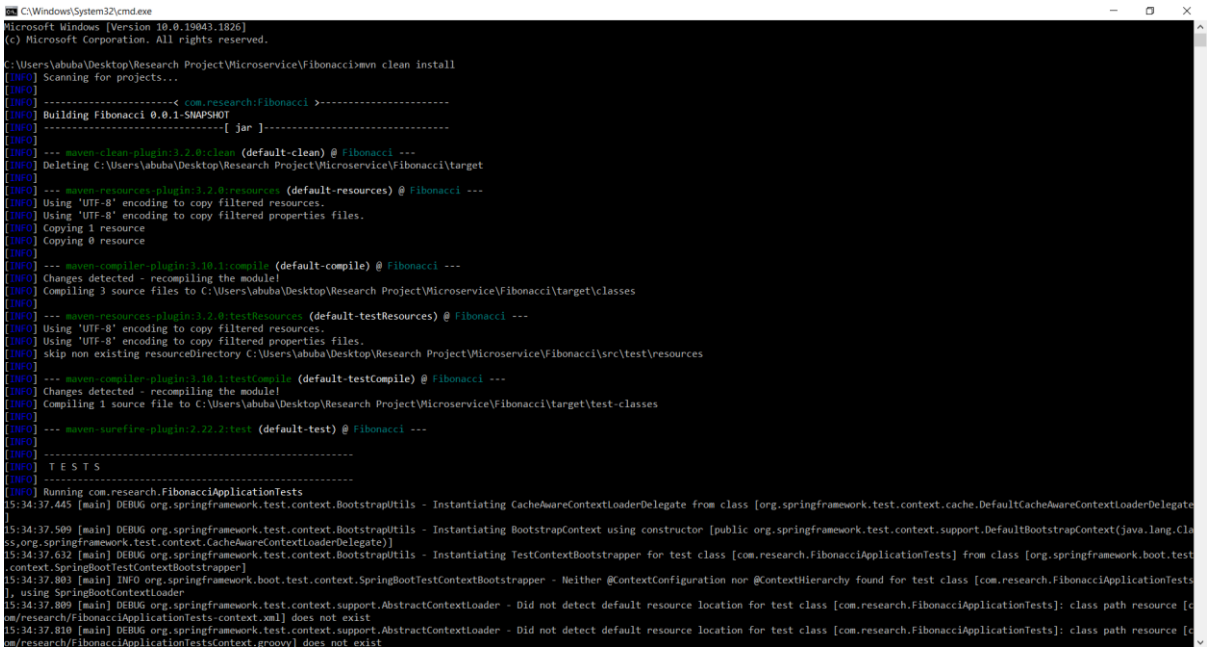

Figure 14: Script execution

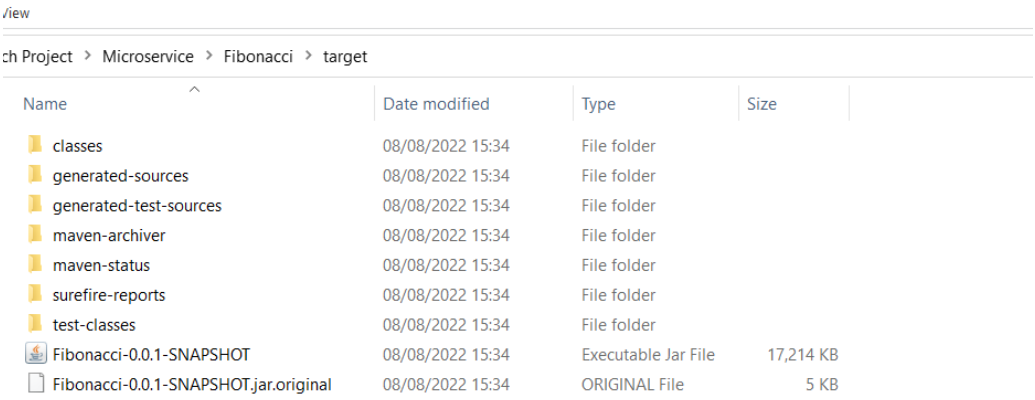

Figure 15: JAR file

3 Cloud Environment Setup and Deployment

The proposed model uses cloud for offloading, AWS cloud platform [4] is chosen for offloading tasks. After creating an account, search Elastic Beanstalk in Management console as shown in figure 16.

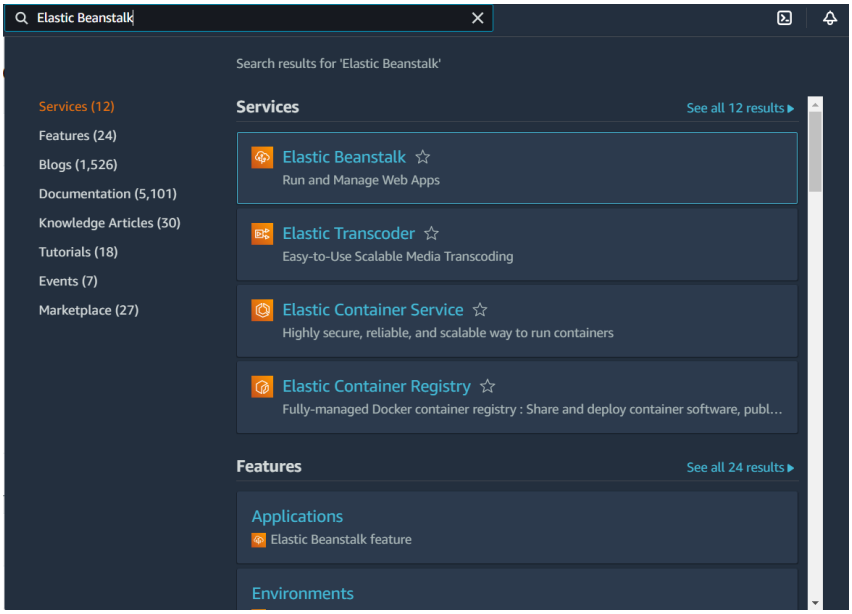

Figure 16: Elastic Beanstalk in Management console

Figure 17 shows the Elastic Beanstalk home page. This is where the deployment process of microservice starts. To start the deployment process first need to create the application and this can be done by clicking on a button called “Create Application”. After clicking, it will bring the page named Create a web app.

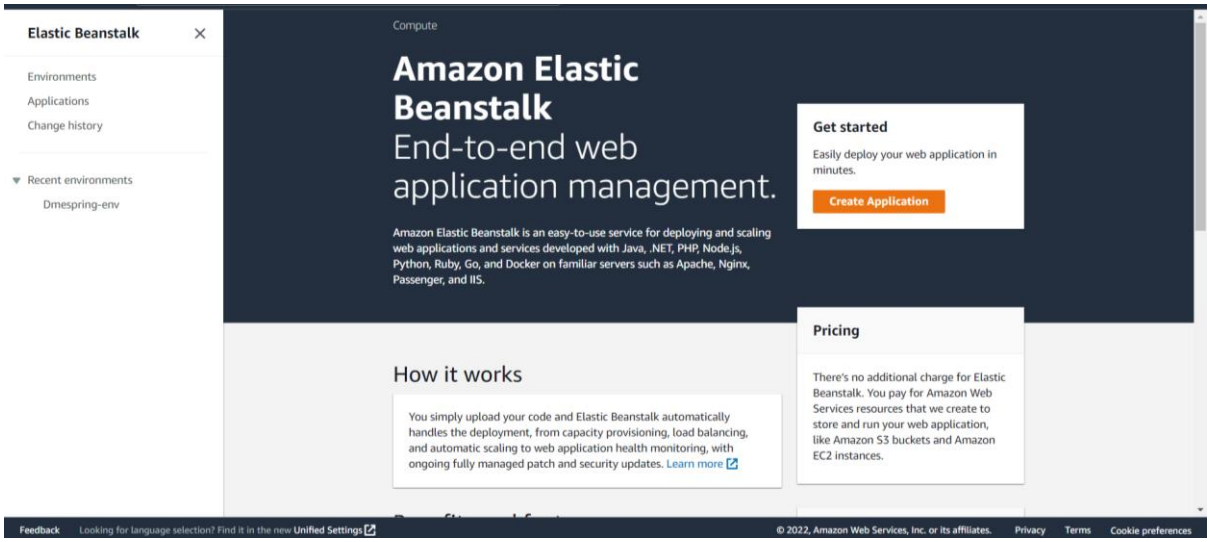

Figure 17: Elastic Beanstalk home page

Provide the appropriate application name because this name will also appear in API link. After typing name, scroll down to further option as shown in figure 18.

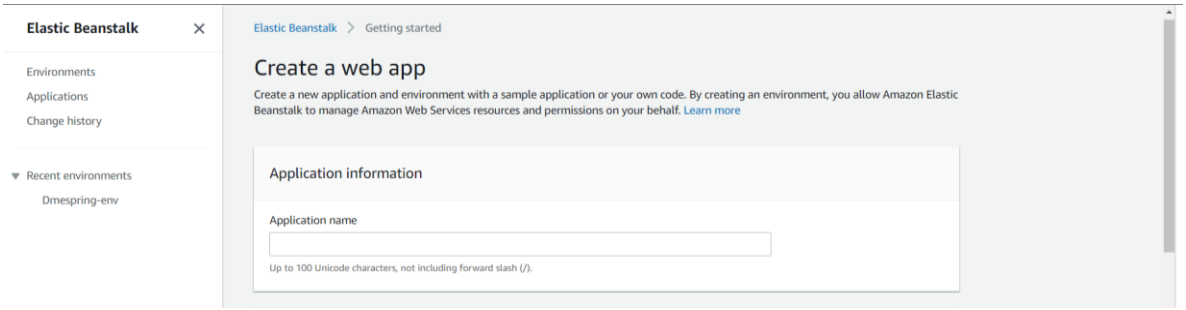

Figure 18: Application name

Next, need to specify platform for deployment as illustrated in figure 19. Java should be selected as a deployment platform not the Tomcat because we have JAR file which has embedded Tomcat server. Leave the rest options by default under platform section.

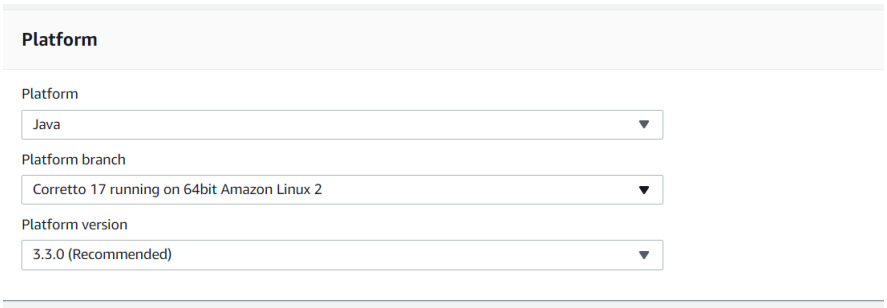

Figure 19: Platform selection

Under Application code section need to select “Upload your code” option and under Source code origin section need to select “Local file”. After this, should select the JAR file from system’s storage and finally click create application.

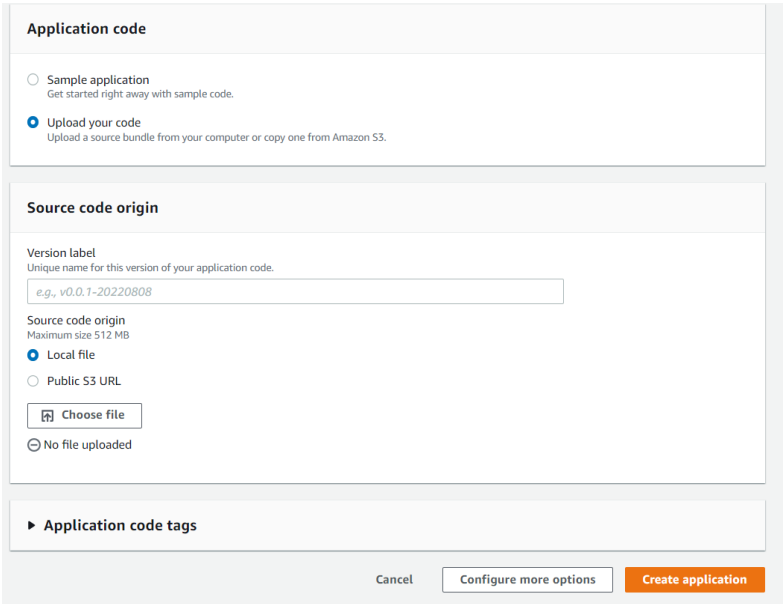

Figure 20: Create application

Application will take few minutes to get deployed and upon the successful completion of the deployment process, API link will be given along with some other details as shown in figure 21.

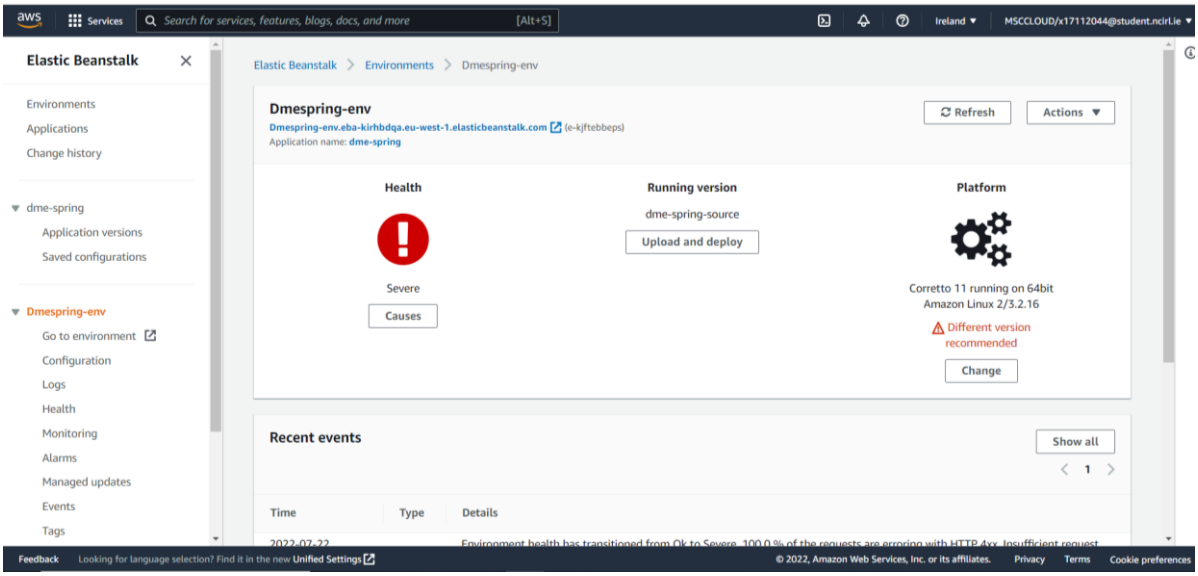

Figure 21: Application deployed

The last step is to take that link and put it in an application developed in an android studio.

References

[1] "Download Android Studio & App Tools - Android Developers", Android Developers, 2022. [Online]. Available: https://developer.android.com/studio. [Accessed: 08- May- 2022].

[2] "GitHub: Where the world builds software", GitHub, 2022. [Online]. Available: https://github.com/. [Accessed: 27- May- 2022].

[3] [Online]. Available: https://www.eclipse.org/ide/. [Accessed: 30- Jun- 2022].

[4] "Cloud Computing Services - Amazon Web Services (AWS)", Amazon Web Services, Inc., 2022. [Online]. Available: https://aws.amazon.com/. [Accessed: 17- Jul- 2022].