National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Ankit Kumar
Student ID: 22113886

School of Computing
National College of Ireland

Supervisor: Dr. Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Ankit Kumar
Student ID:	22113886
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Shivani Jaswal
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	9

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:	
Date:	13th December 2023

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

Ankit Kumar
22113886

1 Prerequisites

The aim of the research is to improve workflow makespan and optimize resource use on Infrastructure as a Service (IaaS) cloud platforms, which would improve Quality of Service (QoS) for organizations with high computing demands. This research is performed as a part of the National College of Ireland's MSc in Cloud Computing Research Project.

The initial step in starting the research is to install the Java Development Kit (JDK) in your system's Java Runtime Environment (JRE). Afterwards, in order to run the CloudSim framework, an Integrated Development Environment (IDE) must be set up such as Eclipse IDE, Netbeans, etc. It is important to remember that this context uses Java version 17, and that in order to build the CloudSim project into Eclipse IDE, Maven needs to be installed.

To guarantee a smooth setup procedure, adhere to the comprehensive guidelines provided below:

Step 1: Installing Java Development Kit (JDK)

- Visit the official Oracle website and download the JDK 17 version by choosing the version on the website.
- Refer the official document present on the website to download and install according to your operating system.

Step 2: Download Apache Maven

- Visit the official Apache Maven and download the Binary Zip archive and unzip the file.
- Refer the official document present on the website to further install and setup paths according to your operating system.

Step 3: Verify the Installation

- For verifying Java installation check its version by using below code.
  `java -version`
- For verifying Maven installation check its version by using below code.
  `mvn -v`
- Figure 3 illustrate the authors result upon checking the versions of java and maven.

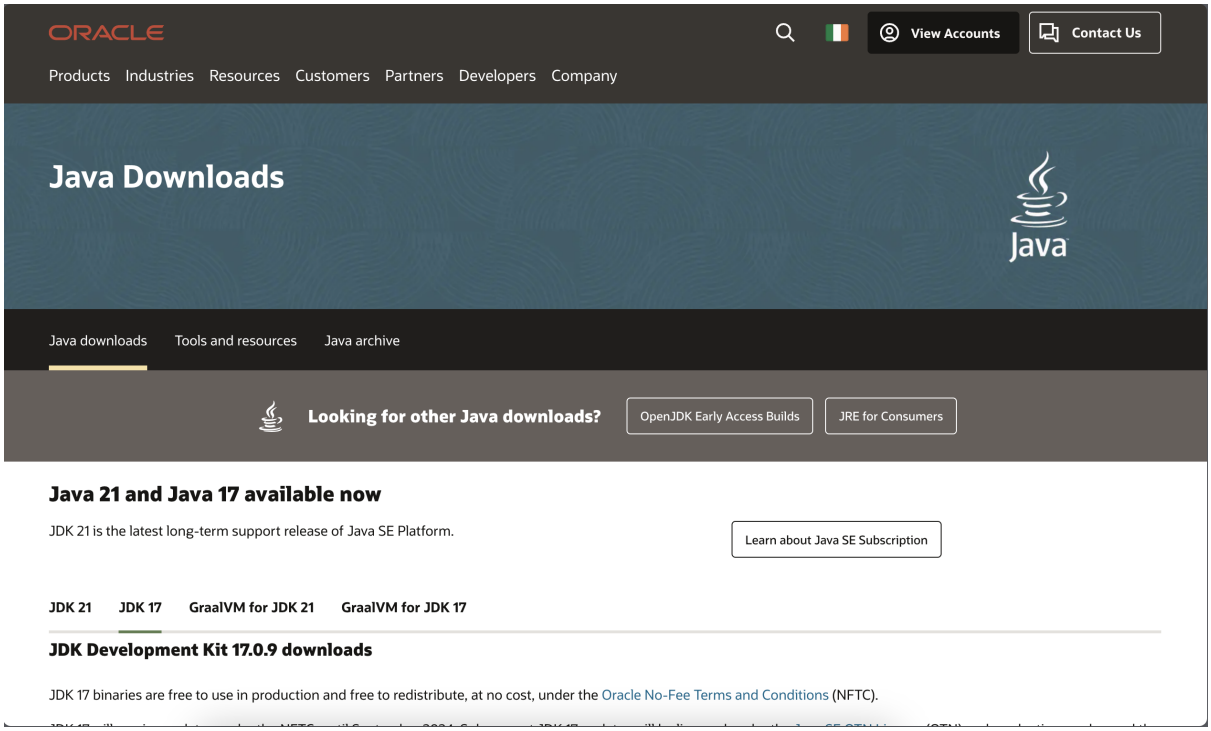

Figure 1: Oracle official webpage to download JDK 17

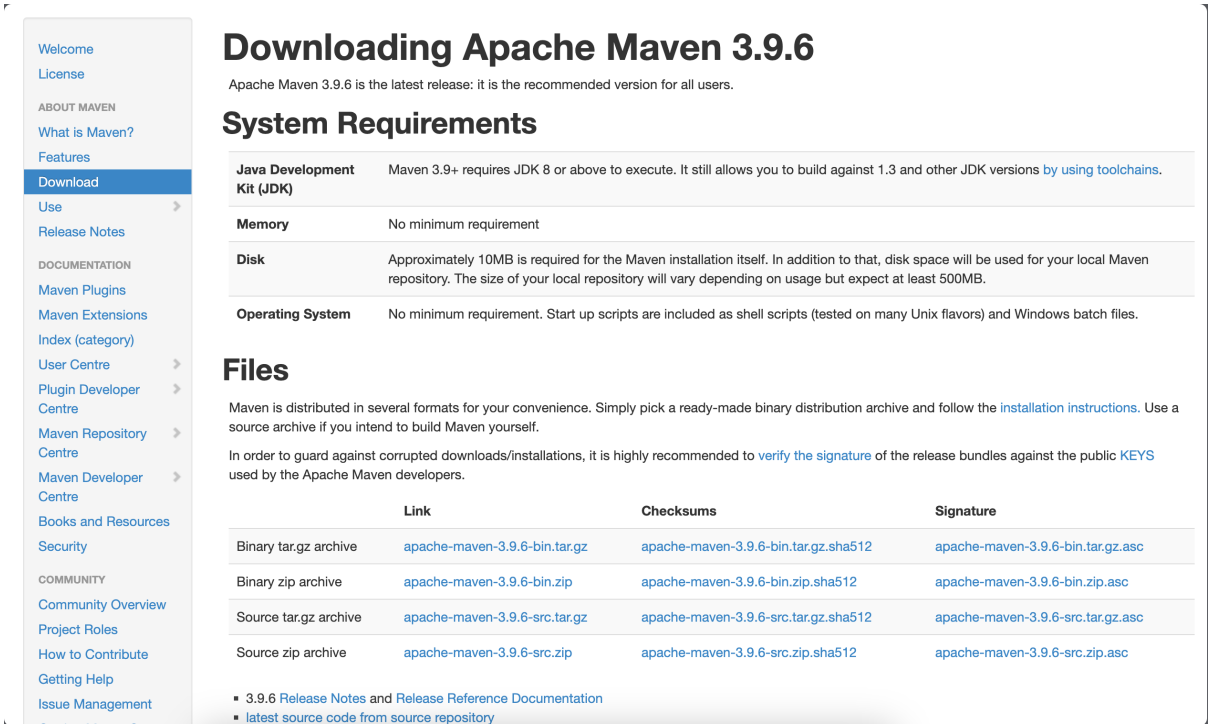

Figure 2: Apache Maven download page

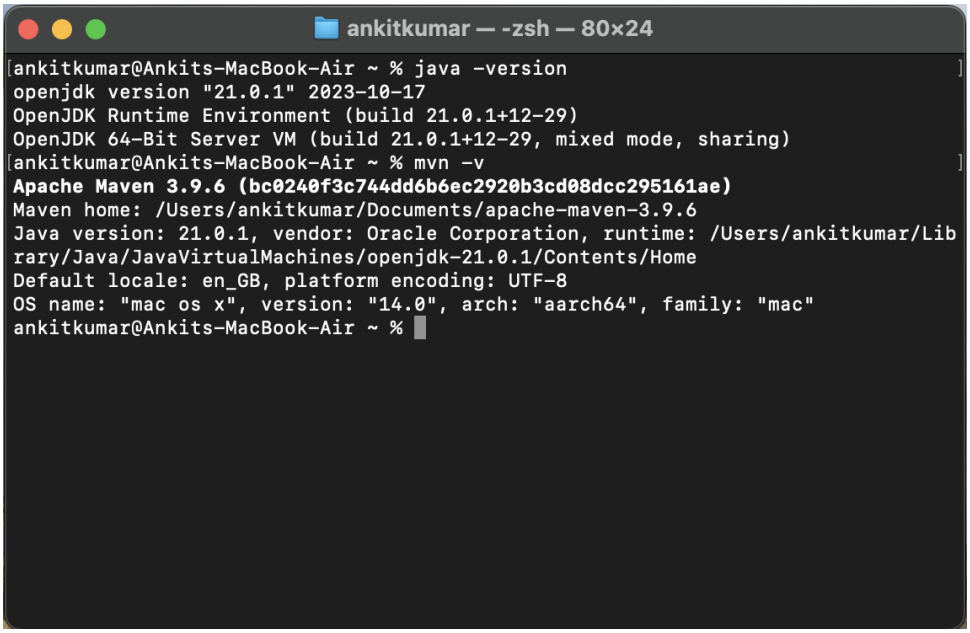

Figure 3: Results of Version verification

2 Development Environment

Development environment specification to simulate the Research project is shown in the below Figure 4.

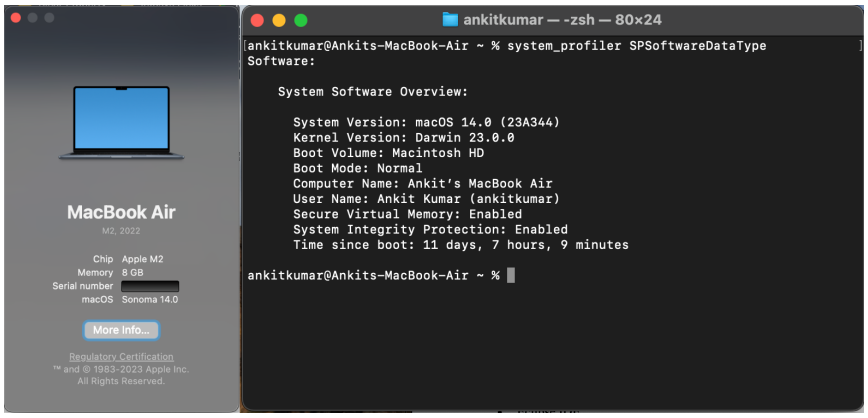

Figure 4: Author's System Specification.

3 Installation

Please follow the section 1 before installation.

Step 1: Installing Eclipse IDE

- Visit the official Eclipse website and download the latest version of Eclipse by choosing the version on the website.

- Refer the official document present on the website to download and install according to your operating system.

Step 2: Installing CloudSim

- Visit the official CloudSim website and follow the link to GitHub where the new realease of CloudSim Framework can be found. CloudSim official website Figure 4.

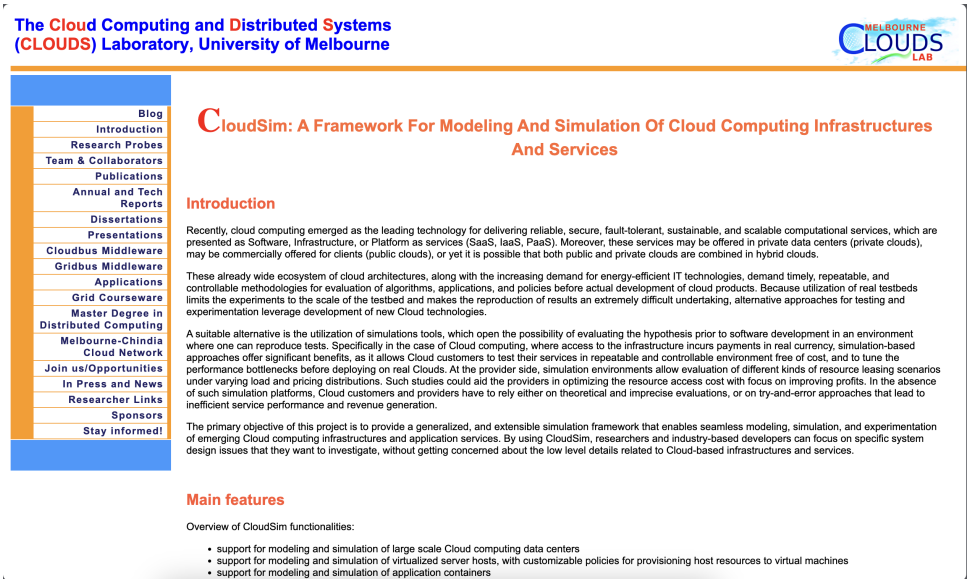

Figure 5: CloudSim official webpage

- From the GitHub download the Zip file, as indicated in Figure 6. But to implement this Research project download the project zip from author's GitHub repository.

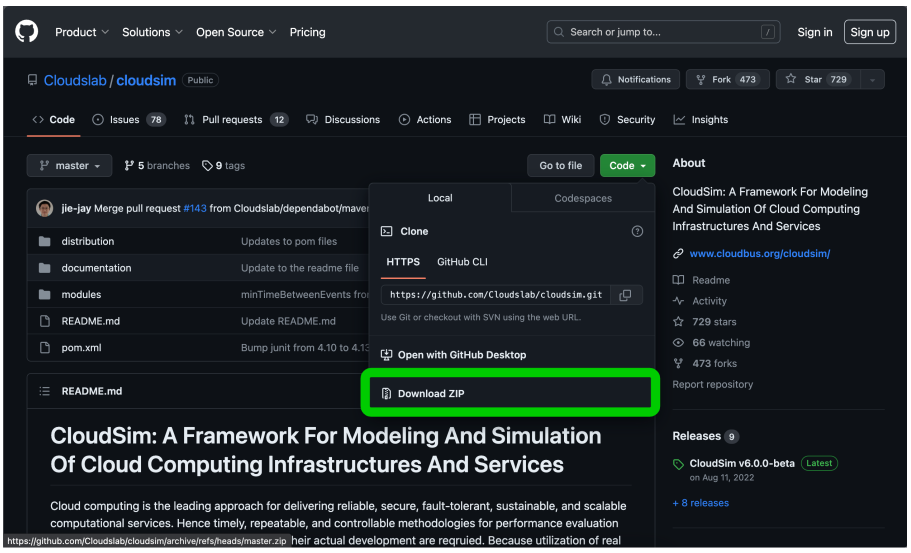

Figure 6: GitHub CloudSim Repository

4 Setting Up CloudSim

Step 1: Import CloudSim

- Import extracted file of CloudSim in the Eclipse IDE as shown in the Figure 7. Select Maven and then select Existing Maven projects and click on next.

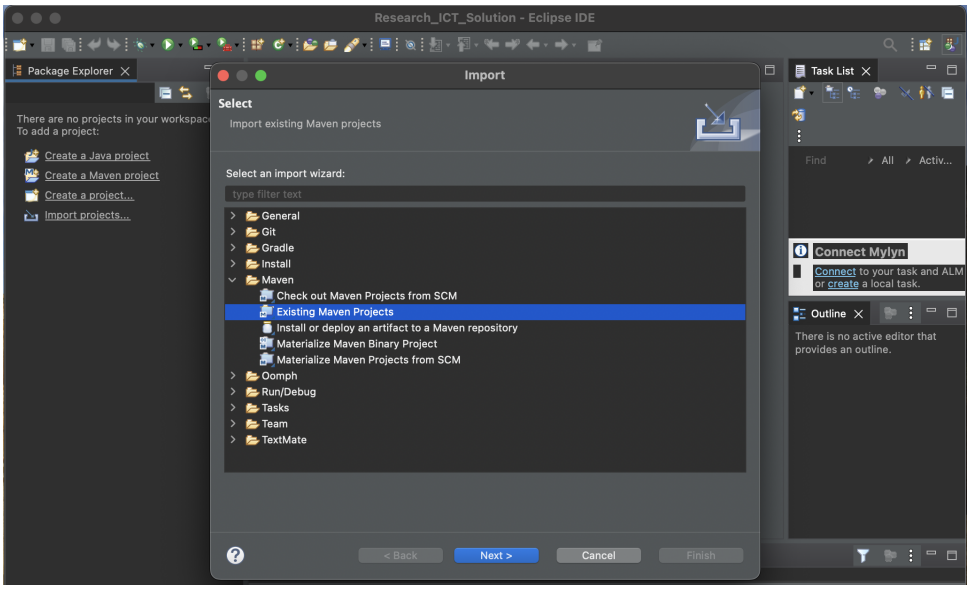

Figure 7: Importing Maven Project

- Now browse through the root directory to find the cloudSim folder and select that as shown in Figure 8. And then click on finish.

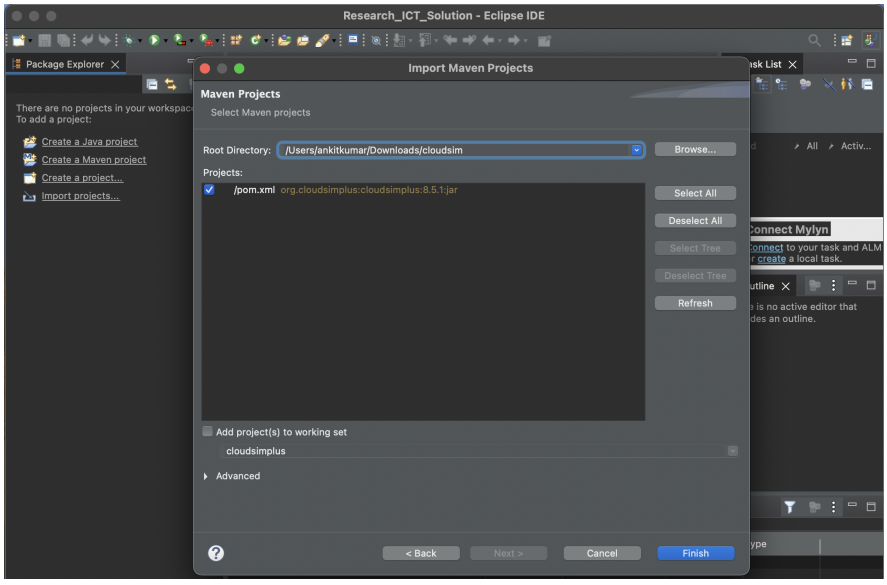

Figure 8: Select the root directory of Maven Project

- Now the CloudSim will be imported and build as an maven project. The imported CloudSim can be seen in the Figure 9.

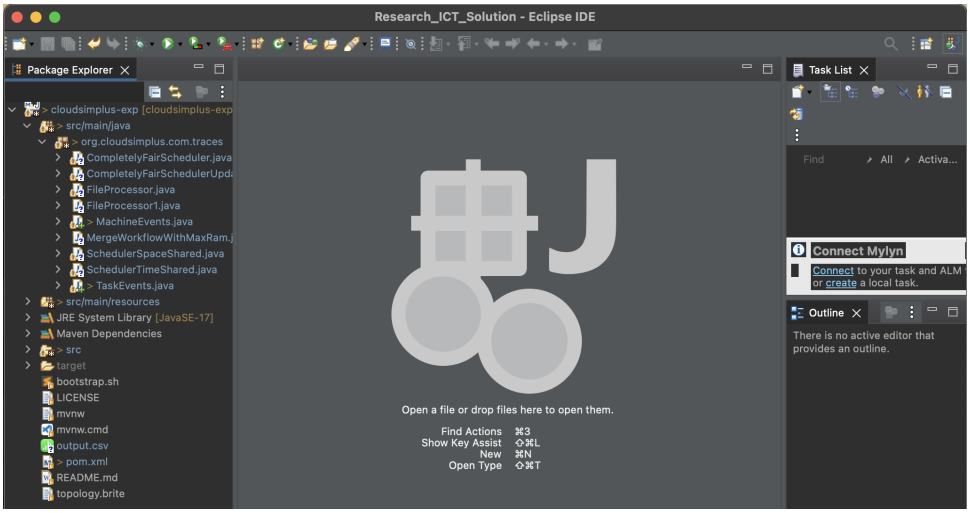

Figure 9: CloudSim imported in Eclipse IDE

5 Project Stages

Different Project stages along with the results.

5.1 Run CFS task scheduling policy

Figure 10 shows the execution of Completely Fair Scheduler.

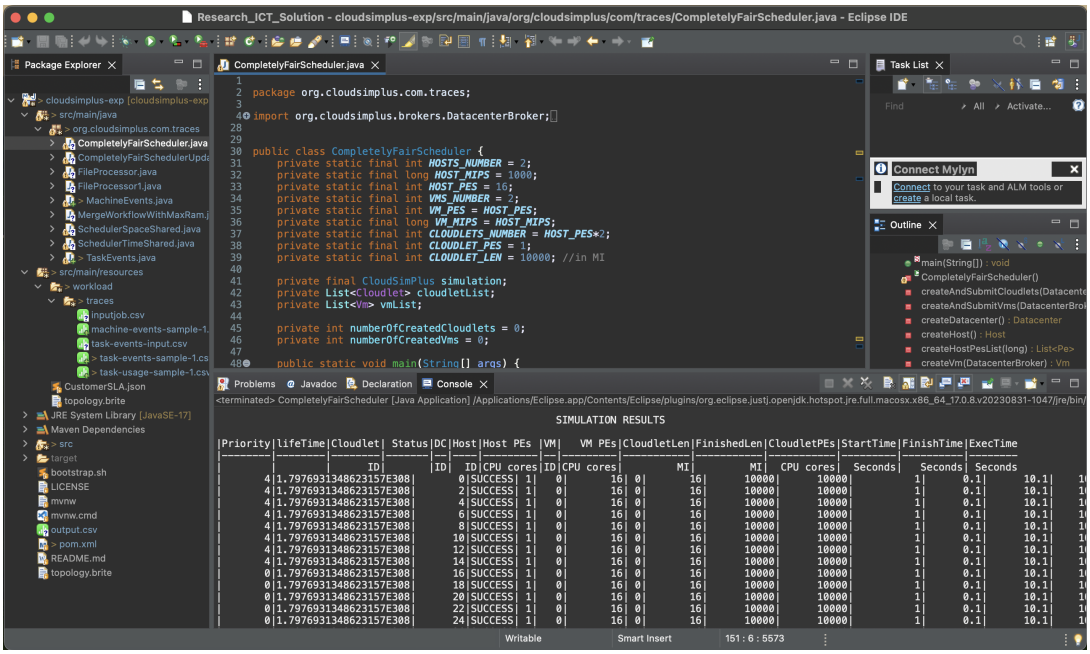

Figure 10: run CompletelyFairScheduler.java to get the results

5.2 Run SSS task scheduling policy

Figure 11 shows the execution of Scheduler Space Shared.

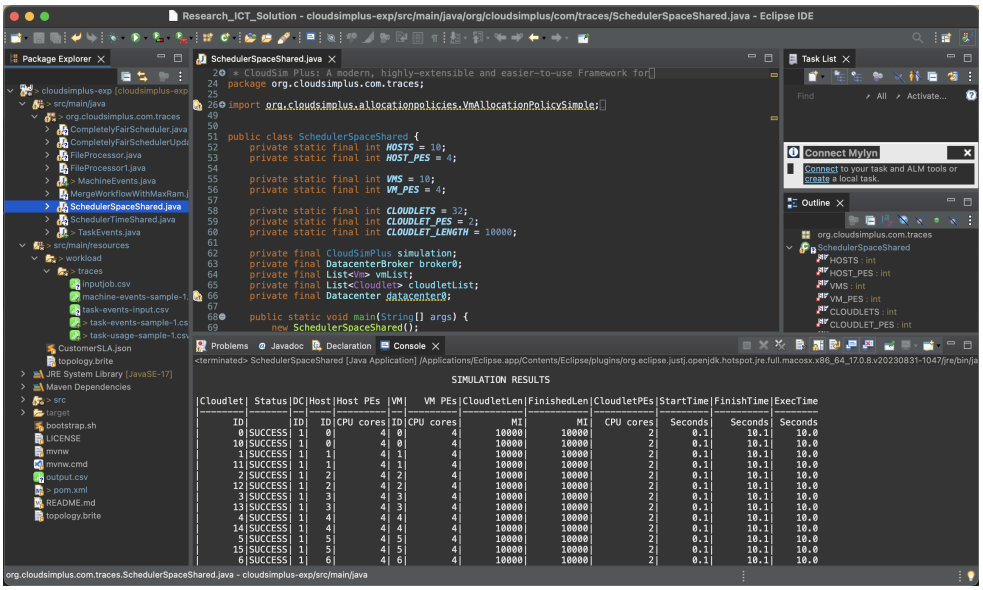

Figure 11: run SchedulerSpaceShared.java to get the results

5.3 Run STS task scheduling policy

Figure 12 shows the execution of Scheduler Time Shared.

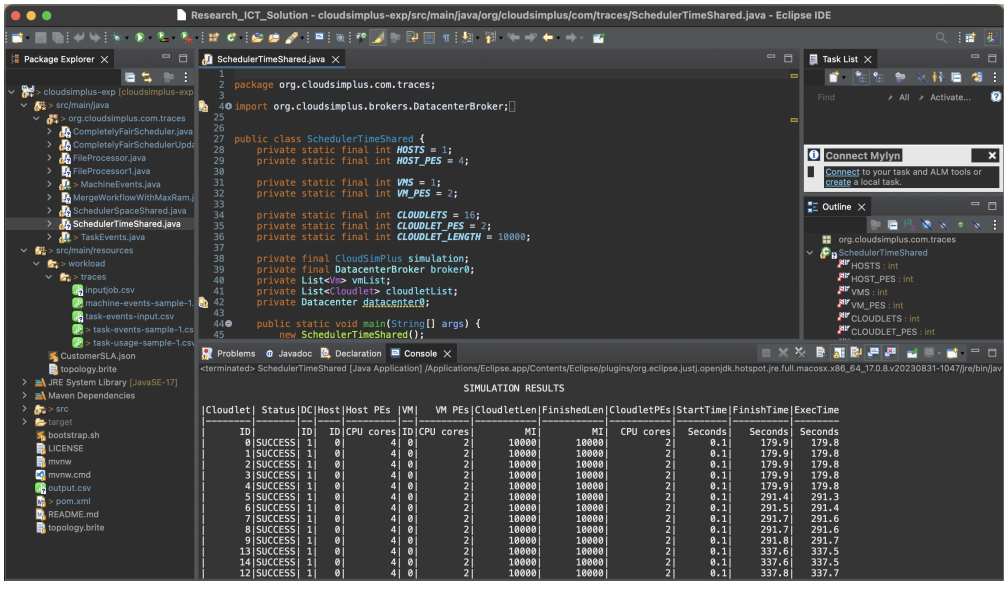

Figure 12: run SchedulerTimeShared.java to get the results

5.4 Run CPU Usage prediction

Figure 13 shows the execution of CPU usage prediction.

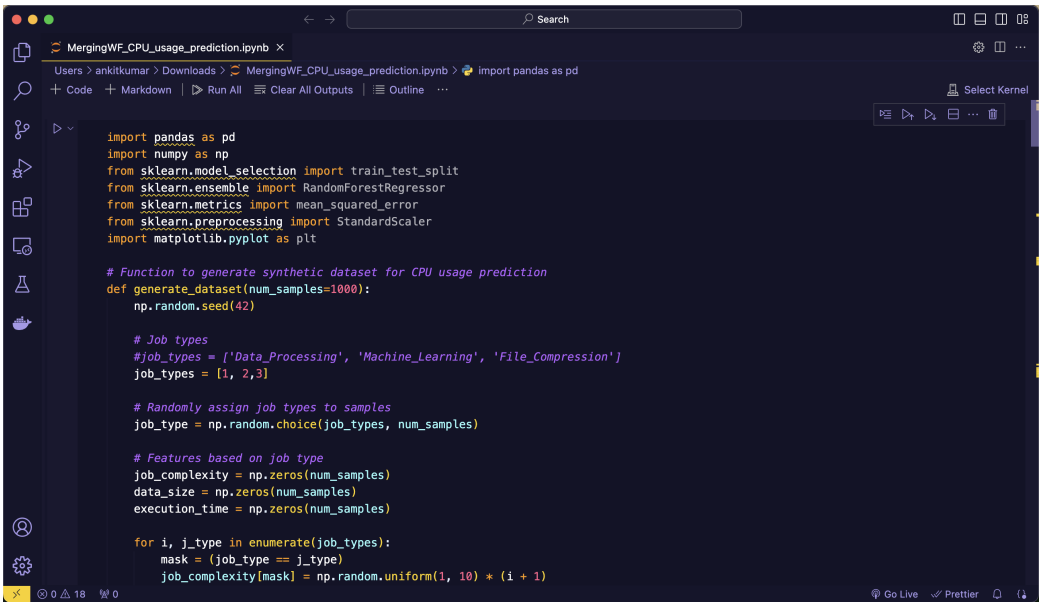

Figure 13: run MergingWF_CPU_usage_prediction.ipynb to get the results

5.5 Run MergeWF task scheduling policy

Figure 14 shows the execution of Merge workflow.

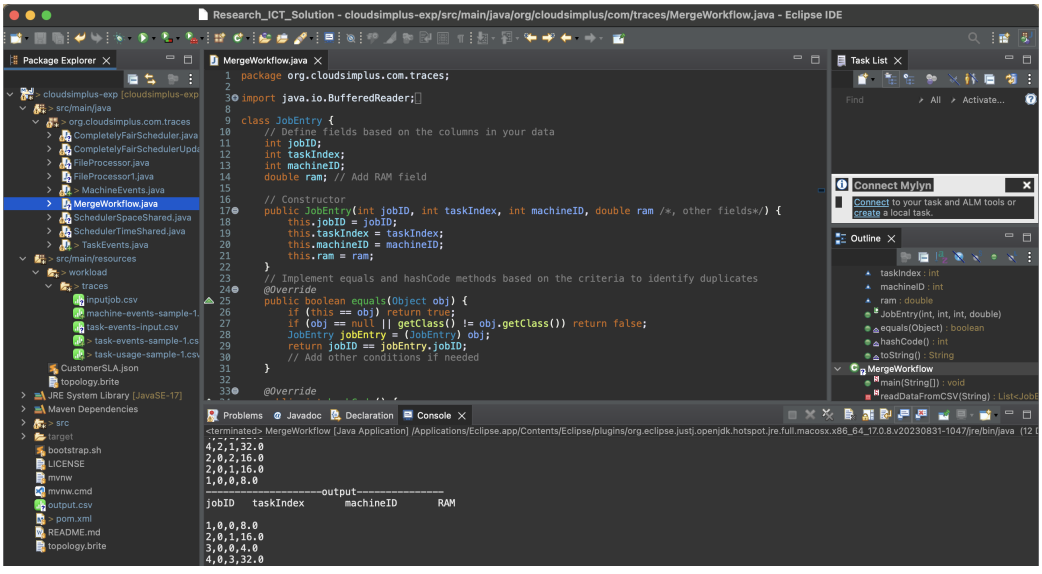

Figure 14: run MergeWF.java to get the results

5.6 Run Task Event

Figure 15 shows the execution of Task Event Simulation.

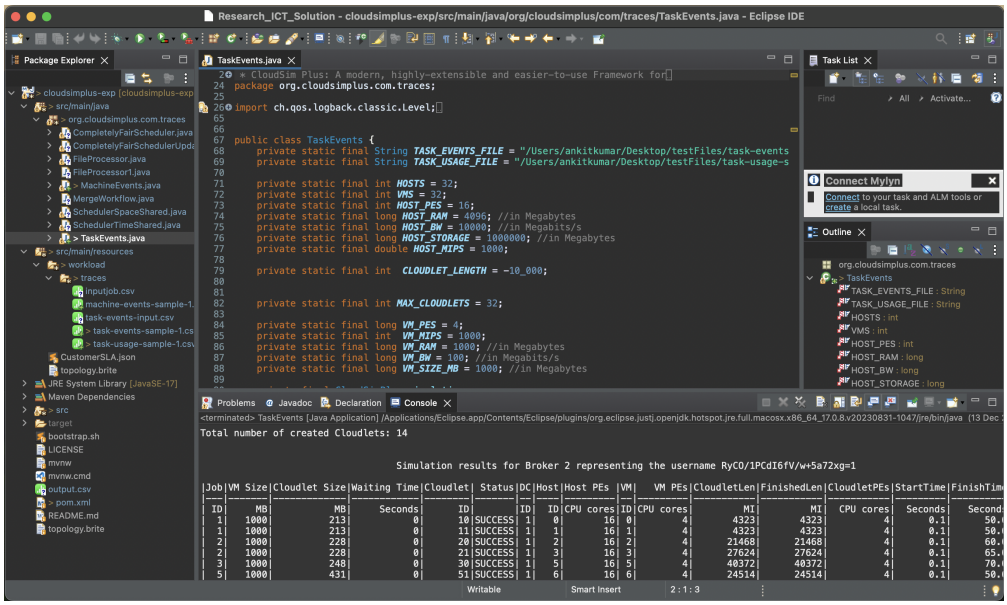

Figure 15: run TaskEvents.java to get the results

5.7 Iterative Simulation

Run the simulation of each policies several time while changing the HOST (Datacenter), Cloudlets (Tasks), VM (Virtual Machine). Now compare these results on the basis of Turn Around Time, Response Time, Flow Time (Makespan) and CPU utilization. These will produce graphs to compare the different policies with author's proposed policy.

6 Conclusion

The simulation findings, in summary, consistently highlight the improved performance of the suggested MergingWF approach over well-known Directed Acyclic Graph (DAG) scheduling methods like CFS, SSS, and TSS. In Infrastructure as a Service (IaaS) scenarios, MergingWF with task merging and a level-based allocation approach shows improved efficiency in makespan optimization and QoS improvement. The empirical analysis establishes MergingWF's effectiveness as the best option among rivals by demonstrating its versatility in managing a range of workflow conditions. For the purpose of improving execution times in complex computing environments, this research offers insightful information about developing workflow allocation strategies.