

Configuration Manual

MSc Research Project
Cloud Computing

Abimbola Ogungbe
Student ID: 19124813

School of Computing
National College of Ireland

Supervisor: Vikhas Sahni

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Abimbola Ogungbe
Student ID:	19124813
Programme:	Cloud Computing
Year:	2019
Module:	MSc Research Project
Supervisor:	Vikhas Sahni
Submission Due Date:	17/08/2020
Project Title:	Configuration Manual
Word Count:	490
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.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	16th August 2020

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):	

Configuration Manual

Abimbola Ogungbe
19124813

1 Introduction

The configuration manual consists of the process for implementing the energy aware load balancing algorithm. It also consists of the general set up used for installing the necessary tools required for the project. This configuration manual will aid academic students and other researchers to have a deeper understanding on the method involved in implementing this research project.

2 Software Tools Used

The software tools used for this project are:

- Eclipse IDE - This was used in the project to code the implementation in Java.
- IfogSim: It is a java file that consists of cloudsims framework.
- Excel - It was used to visualize and study the results.
- JDK 14.0.1 - It contains Java libraries needed to run the Java program.

3 Hardware Specification

- Operating system - Windows/ Linux/ Mac) (Any Operating System is fine).
- RAM: More than 8GB is required
- Processor: Any CPU from Intel Core.

4 Software Installation

This is step by step process for the implementation.

4.1 Java Development Kit (JDK) installation

- Download the JDK 14.01 from the following link [JDK \(n.d.\)](#).¹

¹www.oracle.com/java/technologies/javase/14-0-1-relnotes.html

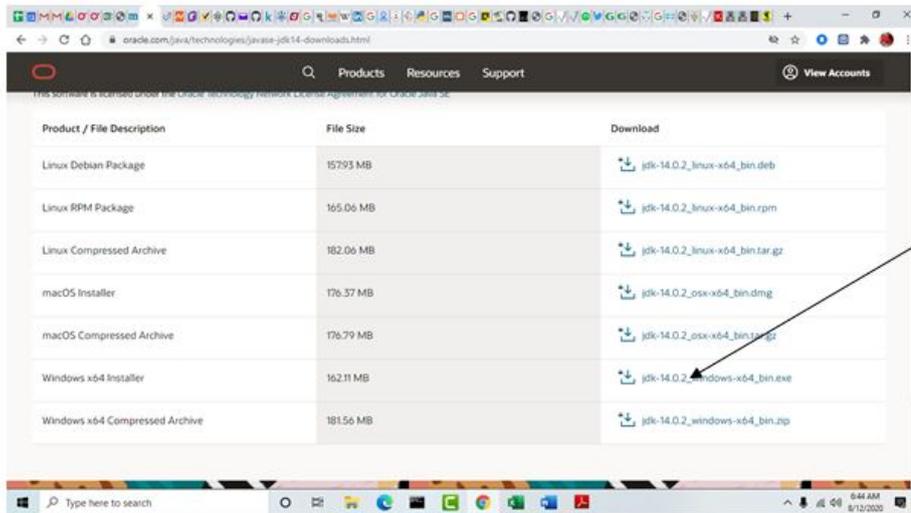


Figure 1: JDK Download

- Install Java JDK 14.0.1 on your system

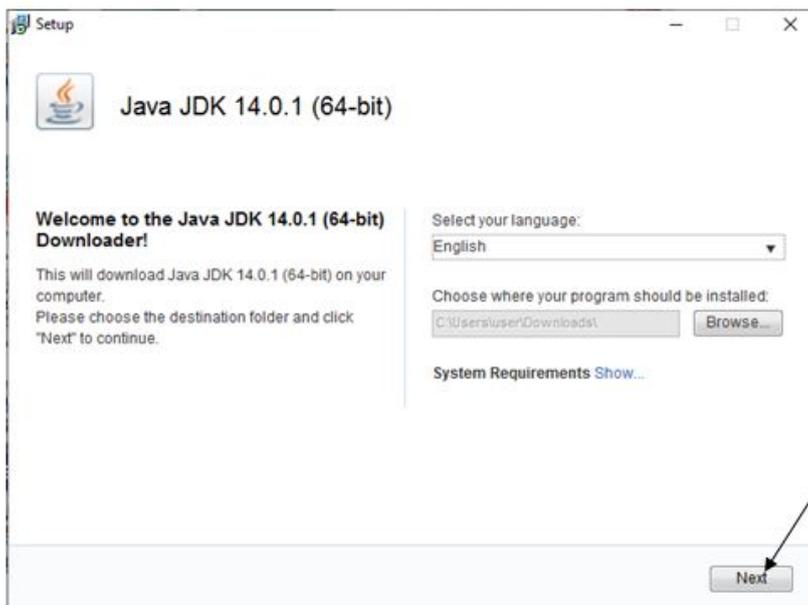


Figure 2: Jdk Installation

4.2 Eclipse Integrated Development (IDE) Installation

- Download Eclipse IDE 2020 -06 from this link 2020-06 (n.d.)

2

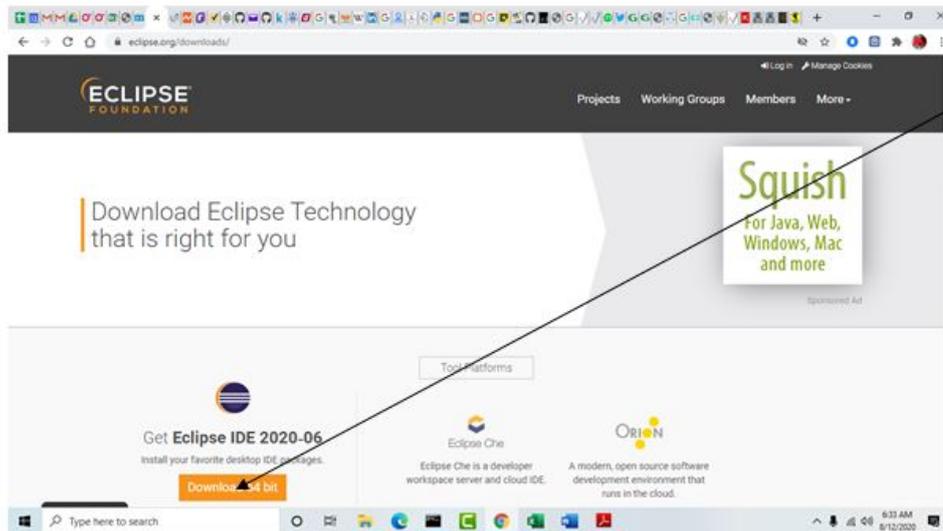


Figure 3: Eclipse Download

- Install Eclipse IDE 2020 -06 on your system

²www.eclipse.org/downloads/packages/release/2020-06/r/eclipse-ide-java-developers

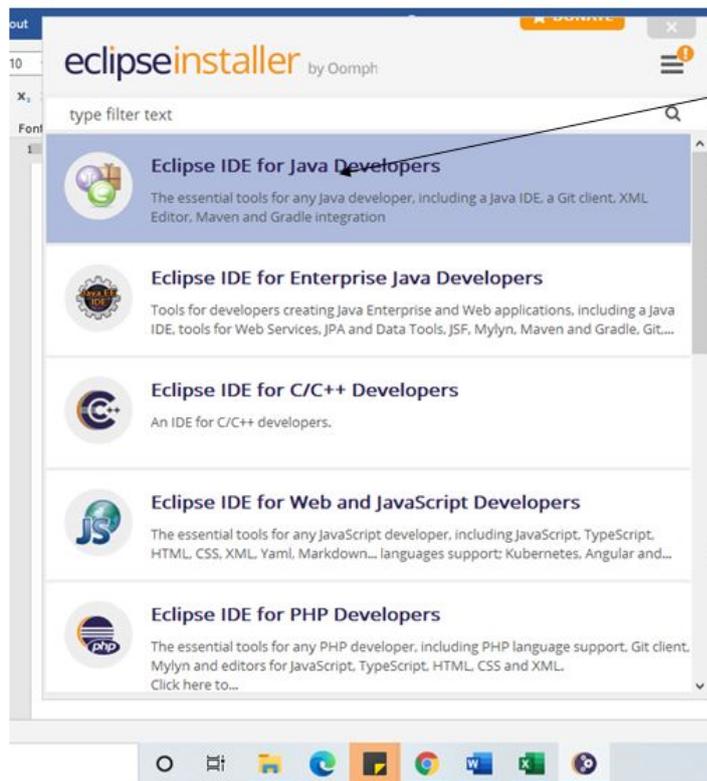


Figure 4: Eclipse Installation

- Open eclipse and create a new java project

- import the iFogSim into Java project

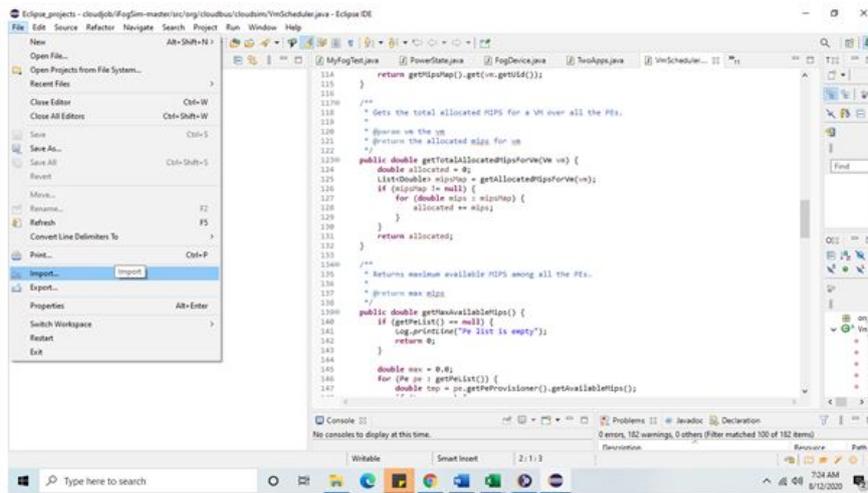


Figure 7: Importing the iFogSim into Eclipse IDE

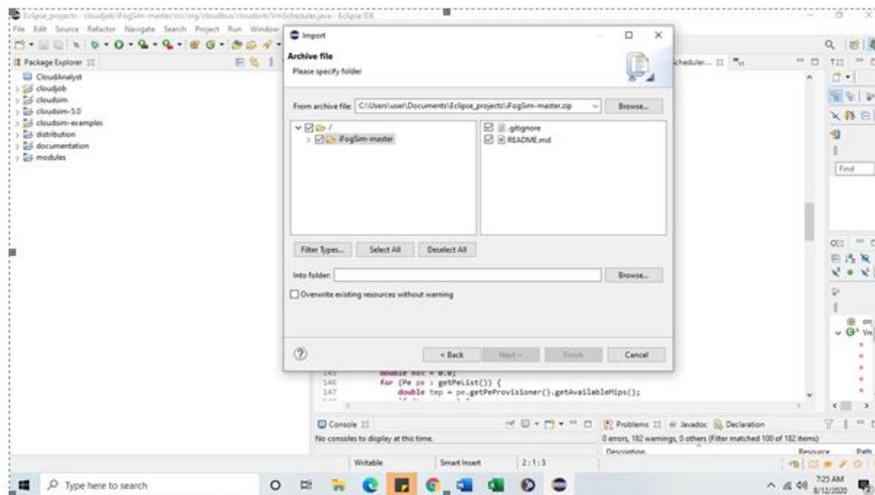


Figure 8: Importing the iFogSim into Eclipse IDE

5 Project Development

Our project development is done using the following steps

5.1 Creating a Fog environment

- Run FogGUI.java to show the graphical user interface for creating the network topology.

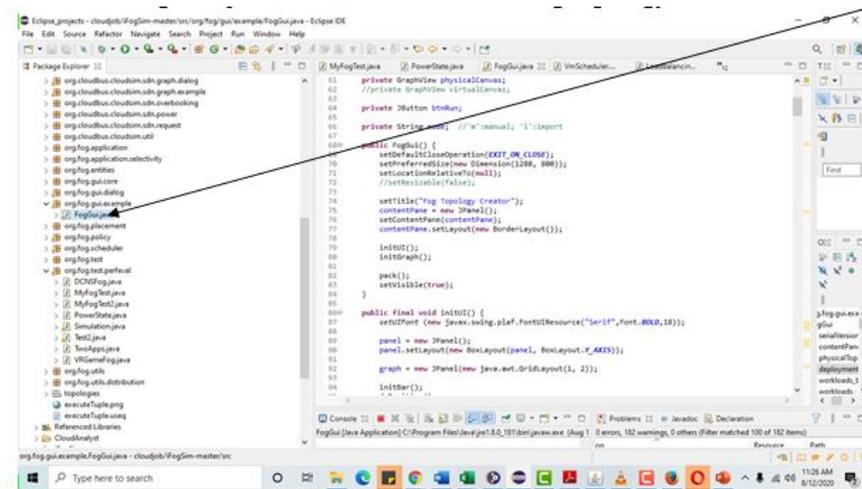


Figure 9: Run FogGUI.java to show GUI for creating a network topology

- Designed Topology

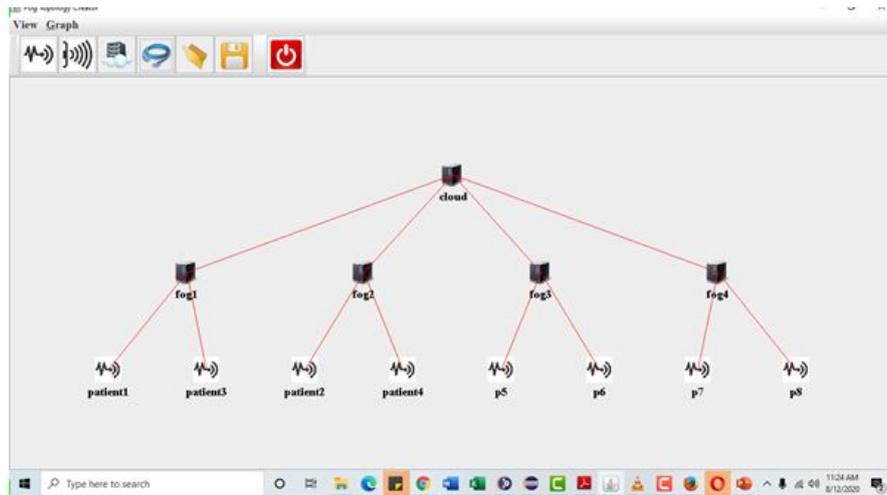


Figure 10: Designed Network Topology

5.2 Creating the Power Recharge Simulation

This shows the code for power state of the fog device that notifies the system if the device is low or charged. It also tells the device to connect to an energy source if it detects the weather condition when low.

```

34     for (int i=0; i<i2; i++) {
35         System.out.println("");
36     }
37     System.out.println("Battery level of fog Node 1 "+fog1+"% \n Battery level of fog Node 2 "+fog2+"%
38     \n Battery level of fog Node 3 "+fog3+"% \n Battery level of fog Node 4 "+fog4+"%");
39
40     if (fog1 <=30) {
41         System.out.println("Fog Node 1 currently running on Non-Renewable Energy");
42     }
43     else if (fog1 >30 && fog1 <60) {
44         System.out.println("Fog Node 1 currently running on Renewable Energy");
45     }
46
47     if (fog2 <=30) {
48         System.out.println("Fog Node 2 currently running on Non-Renewable Energy");
49     }
50     else if (fog2 >30 && fog2 <60) {
51         System.out.println("Fog Node 2 currently running on Renewable Energy");
52     }
53
54     if (fog3 <=30) {
55         System.out.println("Fog Node 3 currently running on Non-Renewable Energy");
56     }
57     else if (fog3 >30 && fog3 <60) {
58         System.out.println("Fog Node 3 currently running on Renewable Energy");
59     }
60
61     if (fog4 <=30) {
62         System.out.println("Fog Node 4 currently running on Non-Renewable Energy");
63     }
64     else if (fog4 >30 && fog4 <60) {
65         System.out.println("Fog Node 4 currently running on Renewable Energy");
66     }

```

Figure 11: Java Code for Power Management

5.3 Creating the Code for the Implementation

Attached below shows the code for the developing the energy aware load balancing algorithm.

```

46
47 package org.fogtest.perfeval;
48
49 import org.fogtest.perfeval.*;
50
51 public class MyFogTest {
52     public static void main(String[] args) {
53         Log.println("Starting Demo App...");
54
55         try {
56             Log.disable();
57             PowerState.showPowerLevel();
58             int num_user = 1; // number of cloud users
59             Calendar calendar = Calendar.getInstance();
60             boolean trace_flag = false; // mean trace events
61
62             CloudSim.init(num_user, calendar, trace_flag);
63
64             String appId = "Patient"; // identifier of the application
65
66             FogBroker fb = new FogBroker("broker");
67
68             Application app = createApplication(appId, fb.getId());
69             app.setUserId(fb.getId());
70
71             createFogDevices(fb.getId(), appId);
72
73             Controller controller = null;
74
75             ModuleMapping moduleMapping = ModuleMapping.createModuleMapping(); // initializing a module mapping
76             for(FogDevice device : fogDevices){
77                 if(device.getName().startsWith("p")){ // names of all Smart phones start with 'p'
78                     moduleMapping.addModuleToDevice("motion_detector", device.getName()); // fixing 1 instance of the Motion Detector module
79                 }
80             }
81             moduleMapping.addModuleToDevice("user_interface", "cloud"); // fixing instances of User Interface module in the Cloud
82             if(CLOUD){
83                 // if the mode of deployment is cloud-based
84                 moduleMapping.addModuleToDevice("IOT_detector", "cloud"); // placing all instances of Object Detector module in the Cloud
85                 moduleMapping.addModuleToDevice("IOT_tracker", "cloud"); // placing all instances of Object Tracker module in the Cloud
86             }
87         }
88     }
89 }

```

Figure 12: Java Code for the Implementation

5.4 Simulation Output

This snapshot shows the output simulation of the designed network topology.

```

<terminated> MyFogTest [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Jul 30, 2020 10:25:45 PM - 10:25:50 PM)
Starting Demo App...
Battery level of Fog Node 1 =53%
Battery level of Fog Node 2 =59%
Battery level of Fog Node 3 =3%
Battery level of Fog Node 4 =57%
Fog Node 1 currently running on Renewable Energy
Fog Node 3 currently running on Renewable Energy
Fog Node 3 currently running on Non-Renewable Energy
Fog Node 4 currently running on Renewable Energy
Placement of operator object_detector on device proxy-server successful.
Placement of operator object_tracker on device proxy-server successful.
proxy-server is shifting object_detector north.
Creating user_interface on device cloud
Creating object_tracker on device cloud
Creating object_detector on device cloud
Creating motion_detector on device proxy-server
0.8 Submitted application Patient
===== RESULTS =====
*****

```

Figure 13: Simulation Output

5.5 Performing the test

- Run the code the simulation.

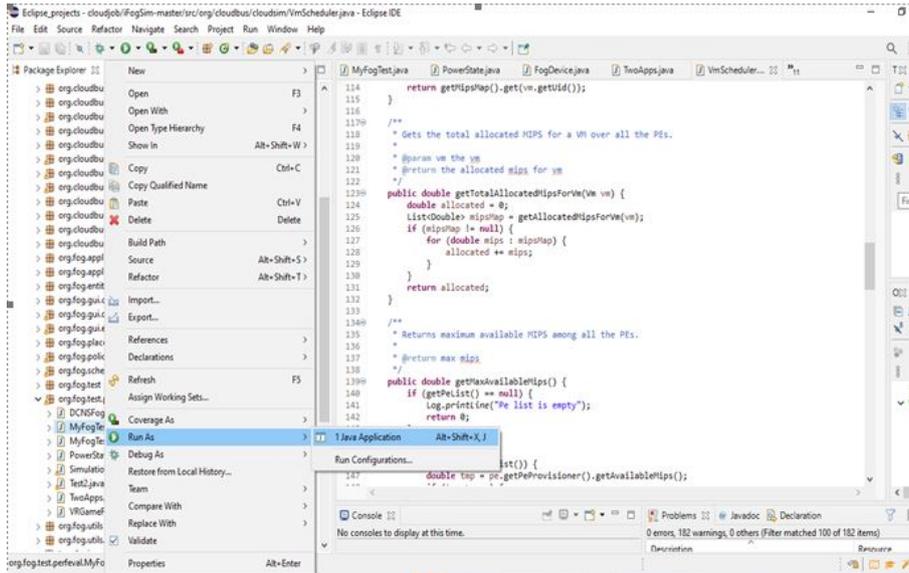


Figure 14: Running the Simulation Code

- Test Results After running the .java file. Shown below is the output that is generated in the console.

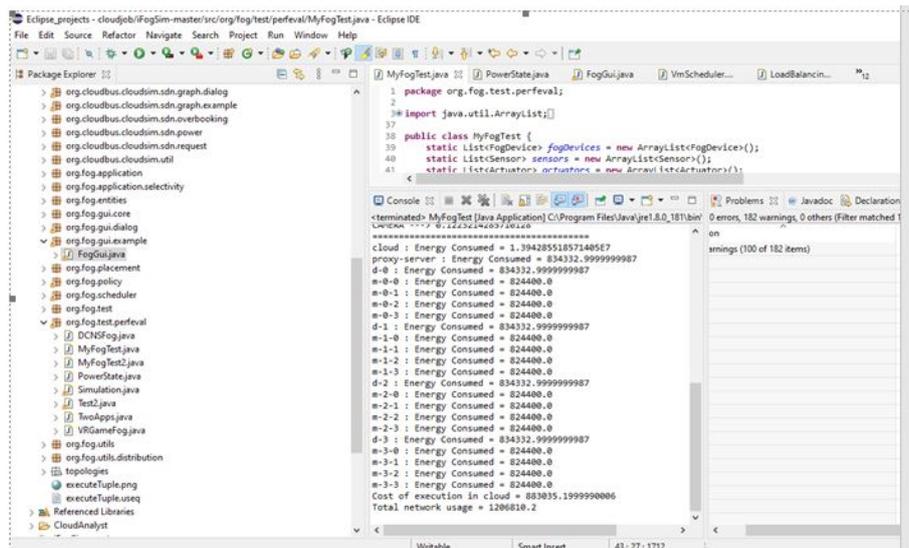


Figure 15: Test Result from the Console

6 Conclusion

This configuration manual illustrates the guidelines to reproducing the energy aware load balancing algorithm associated with the research. The process of installing Eclipse IDE, JDK, iFogSim, creating the task allocation are described.

References

2020-06, E. (n.d.). Eclipse ide for java developers.

URL: *www.eclipse.org/downloads/packages/release/2020-06/r/eclipse-ide-java-developers*

iFogSim (n.d.). ifogsim download.

JDK (n.d.). Java development kit 14.0.1.

URL: *www.oracle.com/java/technologies/javase/14-0-1-relnotes.html*