National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Rihand Parde
Student ID: 23115165

School of Computing
National College of Ireland

Supervisor: Sai Emani

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rihand Parde
Student ID:	23115165
Programme:	Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Sai Emani
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	709
Page Count:	5

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:	25th January 2025

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

Rihand Parde
23115165

1 Introduction

This is a configuration manual that describes the functionality of this project and the process of setting it up using the necessary software, tools and resources.

2 System requirements

- Operating System: Windows 11 (recommended), Linux or MacOS
- Eclipse IDE
- Java Runtime Environment: Version 8 or higher
- iFogSim2 Yousuf Khan and Rahim Soomro (2022)
- Amazon Web Services (AWS) JAR files

3 Installation and Configuration on Local System

1. Download and install Eclipse IDE from its official website at `https://www.eclipse.org/downloads/`
2. Download and install Java Runtime Environment (JRE) from `https://www.java.com/download/ie_manual.jsp`
3. Download the ZIP project named iFogSim-DART-Tsunami from the GitHub repository: `https://github.com/RihandParde/iFogSim-DART-Tsunami`

4 Project Setup on Local System

1. Unzip the iFogSim-DART-Tsunami-main.zip file and open it in Eclipse IDE.
2. Three directories will be present in the project - iFogSim-main/src, iFogSim-main and TsunamiWarningLambda
3. Navigate to iFogSim-main/src > org.fog.test.perfeval > smartdart.java
4. Right click anywhere in the code editor and select Run As Java Application to execute the simulation on local device. Same for the dart.java file in the same folder.

5. All the necessary JAR files will be present in the *lib* folder of both *iFogSim-main* and *TsunamiWarningLambda* folders. If they are not added automatically to the Classpath, then add them manually

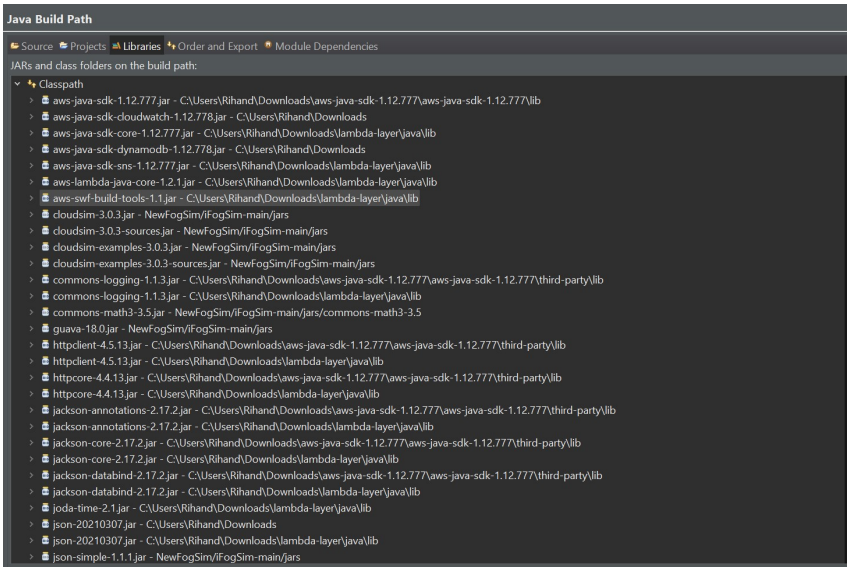

Figure 1: Classpath JAR files

6. In your C: drive or D: drive or any other drive, create the following directory structure: lambda-layer > java> lib

7. Copy all the JAR files from the lib folders of both the *iFogSim-main* and *Tsunami-WarningLambda* folders and paste them in the lib folder of java in lambda-layer folder

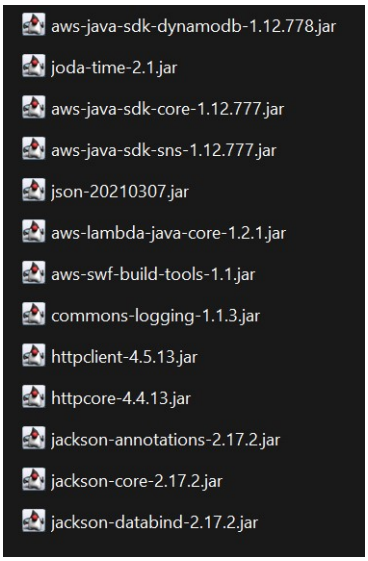

Figure 2: lambda-layer JAR files

8. Create a zip file of the java folder

9. Navigate to TsunamiWarningLambda > org > fog > test > perfeval > TsunamiWarningFunction.java. Replace the SNS topic ARN with the one from your SNS topic in TsunamiWarningFunction.java code, then export it to a JAR file

5 AWS Setup

1. Create an account on AWS and navigate to the Identity and Access Management (IAM) console to create a user.
2. Initialize AWS on your local system using your AWS Access Key ID and AWS Secret Access Key with AWS CLI. More details could be found on AWS's CLI documentation at `https://docs.aws.amazon.com/cli/latest/userguide/getting-started-quickstart.html`
3. In the IAM console, add Full Access roles for DynamoDB, Lambda and SNS

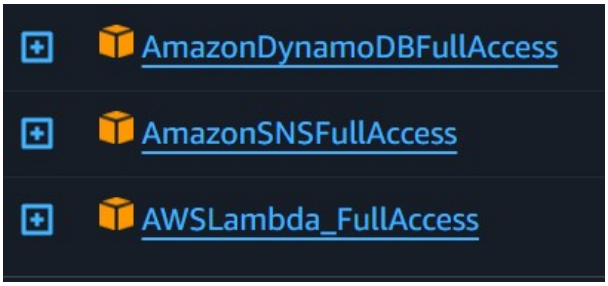

Figure 3: IAM permissions

4. Navigate to DynamoDB DeCandia et al. (2007) console and create a table named *SensorThresholds*.
5. Create two items named *pressure* and *temperature*. Name the partition key as *sensorType* and sort key as *thresholdLevel*
6. Add an additional attribute named *pressureThreshold* for pressure and *temperatureThreshold* for temperature. Add values to them as shown in Figure 3

sensorType *(String)*	thresholdLevel *(String)*	pressureThreshold	temperatureThreshold
temperature	High		10
pressure	High	1020	

Figure 4: DynamoDB table

7. Navigate to SNS Buddha and Beesetty (2019) console and create a new SNS topic named *TsunamiWarningTopic*
8. Create a Subscription and add your phone number to receive the alert SMS
9. Navigate to Lambda Poccia (2016) console and create a layer. Name it JavaDependenciesLayer and upload the zipped java folder from lambda-layer directory to it

10. In the Code tab of TsunamiWarningFunction, upload the TsunamiWarningFunction.java JAR file

11. Run a test event, as shown in Figure 4, to ensure the function invokes

Figure 5: Test event

12. Now run the smartdart.java project in iFogSim as a Java application. If successful, the DynamoDB table will show a successful read request and Lambda function will show a function invocation every time the threshold levels are crossed. An alert will also sent to the mobile number registered in the SNS topic

Figure 6: SensorThresholds table read requests

Figure 7: Lambda function invocation

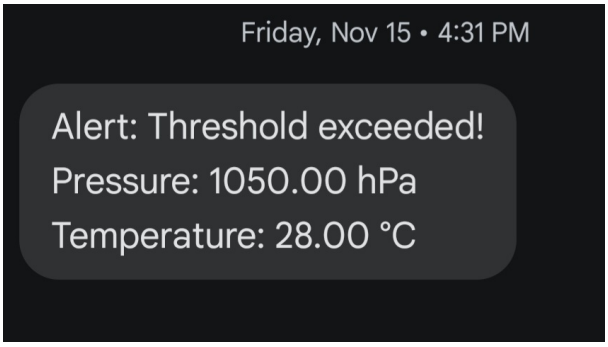

Figure 8: Alert sent to a mobile phone

References

Buddha, J. P. and Beesetty, R. (2019). *The Definitive Guide to AWS Application Integration: With Amazon SQS, SNS, SWF and Step Functions*, Apress.

DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P. and Vogels, W. (2007). Dynamo: Amazon's highly available key-value store, *ACM SIGOPS operating systems review* **41**(6): 205–220.

Poccia, D. (2016). *AWS Lambda in Action: Event-driven serverless applications*, Simon and Schuster.

Yousuf Khan, E. U. and Rahim Soomro, T. (2022). Overview of ifogsim: A tool for simulating fog networks of the future, *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, pp. 1–4.