National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Sameer Nandkishor Kshirsagar
Student ID: 21219141

School of Computing
National College of Ireland

Supervisor: Dr. Ahmed Makki

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sameer Nandkishor Kshirsagar
Student ID:	21219141
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Ahmed Makki
Submission Due Date:	14/12/2023
Project Title:	Dynamic Prewarming Strategy using Reinforcement Learning and LSTM for Cold Start Mitigation in Serverless Computing
Word Count:	923
Page Count:	6

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:	Sameer Nandkishor Kshirsagar
Date:	14th 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):	

Dynamic Prewarming Strategy using Reinforcement Learning and LSTM for Cold Start Mitigation in Serverless Computing

Sameer Nandkishor Kshirsagar
21219141

1 Introduction

This document offers detailed instructions for configuring and carrying out the project's research tests on t2.large AWS EC2 instances. It lists the platforms and software required for the experiments to be implemented successfully. The principal aim of the study was to evaluate the solution's capacity to improve cold start latency, specifically focusing on the Apache OpenWhisk serverless framework. Table 1 provides an overview of the essential instruments and systems required for the study, providing information about the technological basis of the tests. Simultaneously, Table 2 lists the particular software packages and libraries that were used during the study, offering a thorough summary of all the computer resources that were used for the project.

Type	Tool/Platform
Operating System	Linux Ubuntu Server 18.04 LTS (HVM)
Virtual Machines	Amazon Web Services (AWS) EC2
Machine Learning	Keras Tensorflow 2.15.0
Serverless Platform	Apache OpenWhisk 1.0.0 (open-source)
Container Technology	Docker 20.10.17 CE
Programming Language	Python 3.11.5
Performance/Load Testing	Java (openjdk-11) & Apache JMeter 5.6.2

Table 1: Used Platforms and Tools

Python	matplotlib, numpy andpandas
Bash	gnupg, curl, make, containerd, npm, ca-certificates, zip, lsb-release, docker-compose 1.21.2, python-pip, docker-ce-cli and wsk (openwhisk cli v1)

Table 2: Pre-requisites software

2 Experiment

2.1 Setting Up Amazon Resources

1. Open the Amazon Web Services Portal and log in.

2. Search for the EC2 service and start it.

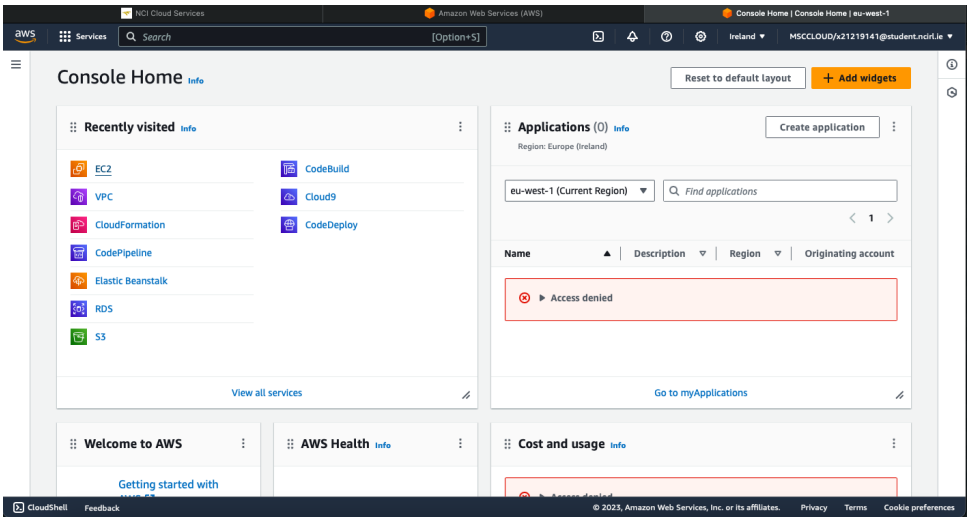

Figure 1: EC2 service.

3. Along the left panel, in the Network and Security group, select Open Key Pairs.

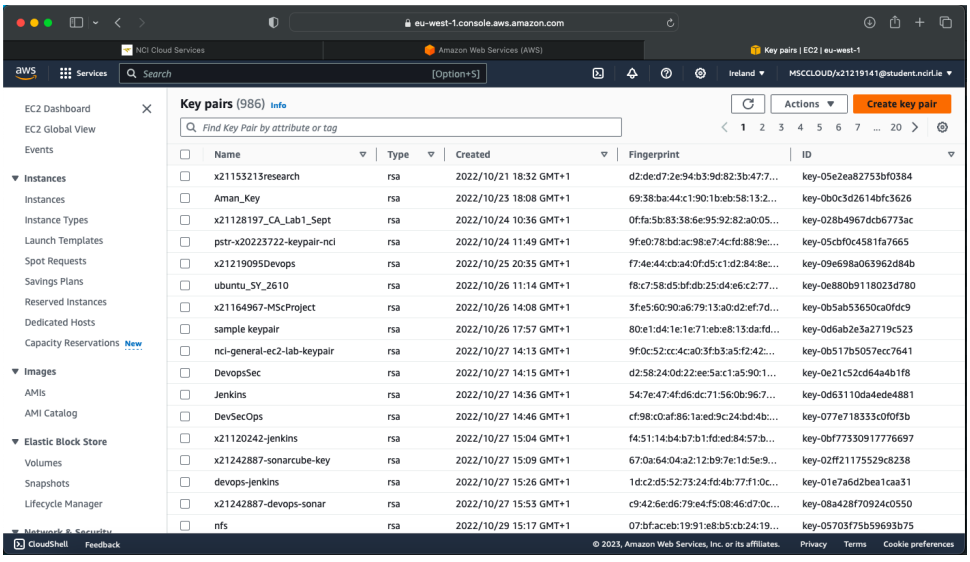

Figure 2: Key Pairs.

4. Using the name, the default RSA type, and exporting as a.pem file format, create a key pair.

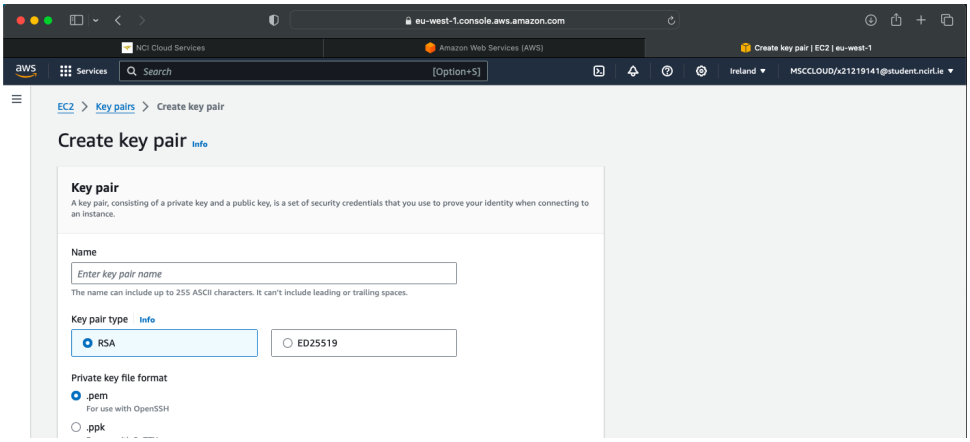

Figure 3: Create a Key Pairs.

5. To open a new instance, navigate to Instances and click.
6. Assign virtual machines with proper name.
7. Browse for and choose the 64-bit "Ubuntu Server 20.04 LTS (HVM), SSD Volume Type.
8. If you don't change the instance type to t2.large (2vCPUs/8GB RAM), the experiments will not succeed make sure that t2.large (2vCPUs/8GB RAM) is selected.
9. Choose the key pair that was previously produced.
10. Increase the virtual machines' storage space to 16 gigabit.
11. Start the instances.

2.2 How to connect to Instance

It is not mandatory but After the instances are up and running, append "studentid" or anything which you can remember to the end of the VM names to differentiate them from one another.

1. For easy access, copy the public IPv4 DNS from each instance's information tab to your clipboard or another location.
2. From your client, launch a terminal session (for convenience, in the same directory as your.pem file).
3. Open an SSH session to the EC2 instances by entering. `ubuntu@YOUR_EC2_PUBLIC_IP_DNS ssh -i "YOUR_KEYPAIR.pem"`.

2.3 Executing the Installation

1. Use the command git –version to confirm that git is installed on the virtual machine (which it should be). If git isn't installed already, use sudo apt install git.

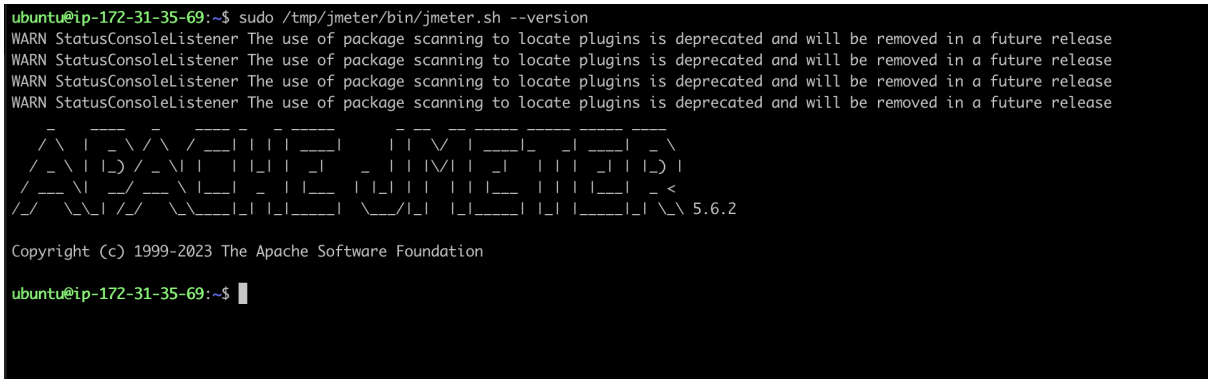

Figure 4: This is Apache Jmeter version output.

2. Use the following command to clone the project's codebase from the Github repository: sudo git clone https://github.com:sameer05jain/projectprewarm.git.

3. The install.sh bash file automates the download and installation of necessary software tools and dependencies. Enter sudo bash /openwhisk/install.sh to launch this file. Please wait till the installation gets complited

4. Verify the installation of both Docker and OpenWhisk by watching the containers that are now running `sudo docker ps --format '{{.Names}}'`.

5. Verify that the Jmeter work-load `sudo wsk -i` action list has the necessary Openwhisk actions created.

6. To enable JMeter to execute `sudo unzip`, use the following commands to relocate the downloaded executables and JMeter files to the proper location on the server.
   It is `sudo unzip ~/apache-jmeter-5.4.3.zip`.
   To move the Apache JMeter installation directory, use the following command: `sudo mv ~/apache-jmeter-5.4.3 ~/jmeter`.
   To move the JMeter directory to the '/tmp' directory, use the following command: `sudo mv ~/jmeter /tmp`.
   To add JMeter to the system's PATH, use the following command: `echo 'export PATH="$PATH:/tmp/jmeter/bin"' >> ~/.bashrc`.
   To apply the changes to the current terminal session, use the following command: `source ~/.bashrc`.

7. To check the JMeter version, use the following command:
   `sudo /tmp/jmeter/bin/jmeter.sh --version`.

   Figure 4 presents a **full-size** plot.

2.4 Running the Validation in Test Infrastructure

Run with no Machine learning predictions

Step 1 : Make sure you're in the home directory cd / Step 2 : Run the Jmeter test plan against `function_gateway` with this command `sudo nohup /tmp/jmeter/bin/jmeter.sh`

Figure 5: This is console output while running the script.

```
-n -t
"~/project_prewarm/ml_experiments/ml_default/Jmeter_testplan_default.jmx"
-l
"~/project_prewarm/ml_experiments/ml_default/Jmeter_run_logs_default.csv"
>
"~/project_prewarm/ml_experiments/ml_default/Jmeter_run_logs_default.log"
&
```

Run with LSTM Machine learning predictions
Step 1 : Make sure you're in the home directory cd / Step 2 :`sudo nohup python ~/project_prewarm/function_gateway.py lstm & /tmp/jmeter/bin/jmeter.sh -n -t "~/project_prewarm/ml_experiments/ml_lstm/Jmeter_testplan_lstm.jmx" -l "~/project_prewarm/ml_experiments/ml_lstm/Jmeter_run_logs_lstm.csv" > "~/project_prewarm/ml_experiments/ml_lstm/Jmeter_run_logs_lstm.log" &`

Run with QLearning Machine learning predictions
Step 1 : Make sure you're in the home directory cd / Step 2 : `sudo nohup python ~/project_prewarm/function_gateway.py qlearning & /tmp/jmeter/bin/jmeter.sh -n -t "~/project_prewarm/ml_experiments/ml_qlearning/Jmeter_testplan_qlearning.jmx" -l "~/project_prewarm/ml_experiments/ml_qlearning/Jmeter_run_logs_qlearning.csv" > "~/project_prewarm/ml_experiments/ml_qlearning/Jmeter_run_logs_qlearning.log" &`

2.5 Result Analysis

Logs files will be generated in `~/project_prewarm/ml_experiments` under the corresponding ML folder. This can be used in the Jupyter notebooks provided to visualize and verify the results.

For example :

`~/project_prewarm/ml_experiments/ml_qlearning/Jmeter_run_logs_qlearning.csv`.

Azure (n.d.a) Azure (n.d.b)

References

Azure (n.d.a). Azurepublicdataset/azurefunctionsinvocationtrace2021.md, https://github.com/Azure/AzurePublicDataset/blob/master/AzureFunctionsInvocationTrace2021.md. GitHub.

Azure (n.d.b). Azurepublicdataset/license, https://github.com/Azure/AzurePublicDataset/blob/master/LICENSE. GitHub.