National College of Ireland

Configuration Manual

MSc Research Project
MSc Cloud Computing

Alice Angenette Rodgers
Student ID: X23210362

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……..……Alice Angenette Rodgers……

Student ID: …x23210362……..

Programme: …MSc Cloud Computing…… **Year:** ……2024….

Module: …Research Project……

Lecturer: …Vikas Sahni……

Submission Due Date: …12 December 2024……

Project Title: ……Enhancing CBTC System Efficiency with DRL and Cloud Computing…….....

Word Count: ……1300…… **Page Count:** ……09……

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: ……Alice Rodgers……

Date: ……12 December 2024……

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

Alice Angenette Rodgers
Student ID: X23210362

1 Simulating the Dataset

In this section, the modeling steps for simulating the dataset for Communication Based Train Control (CBTC) system. It consists of a dataset that simulates train operations, equipment status, task offloading, maintenance schedules and environmental conditions.

1.1 Dataset Overview

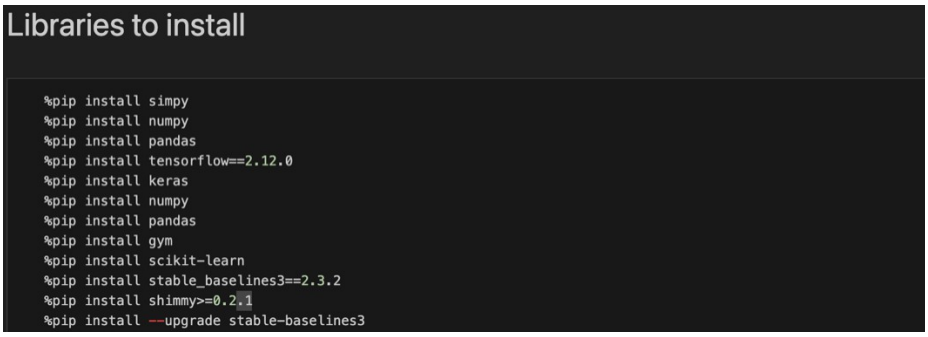
The SimPy library was used to create the dataset by simulating dynamic operations of trains and associated infrastructure. The key features include:

- Train speed, train position, onboard computation load.
- The costs for wayside equipment status and maintenance.
- Temperature, humidity, wind speed, environmental factors.
- Decisions of offloading tasks over network latency and bandwidth.

1.2 Simulate the dataset.

1.Set up the simulation environment in Google Colab:

- Open Google Colab, create a new notebook.
- Import the necessary libraries



2 Defining Parameter for Simulation

- Set edge device, number of trains, and time intervals simulation parameters.

• Category	Feature	Description
Train Operations Data	timestamp	Time when the data was recorded
Train Operations Data	train_id	Unique identifier for each train
Train Operations Data	speed_kmph	Speed of the train (km/h)
Train Operations Data	position_km	Distance from the starting point (km)

Train Operations Data	energy_consumption_kWh	Energy consumed by the train (kWh)
Wayside Equipment Data	equipment_id	Unique identifier for each piece of equipment
Wayside Equipment Data	equipment_status	Operational status of the equipment
Wayside Equipment Data	maintenance_cost_eur	Estimated cost of maintenance (EUR)
Network Communication	signal_strength_dbm	Signal strength between train and equipment (dBm)
Network Communication	latency_ms	Latency between train and equipment (ms)
Environmental Data	temperature_celsius	Ambient temperature (°C)
Environmental Data	humidity_percentage	Humidity level (%)
Task Offloading	task_offloaded	Size of task being offloaded (MB)
Task Offloading	energy_saved_kWh	Energy saved by offloading (kWh)
Maintenance Data	maintenance_type	Type of maintenance (scheduled/unscheduled)
Maintenance Data	downtime_minutes	Duration of downtime (minutes)

Table.1. Key Constraints for the simulated Dataset

Generate the data: Simulate data for each train, equipment, and environmental factors of trains, edge devices, and time intervals using a loop.

4. Save the dataset:

- Simulated dataset is exported to the CSV file for processing.
- Upload the CSV file generated to Google Drive for model train.

3 Model Training in Google Colab

In the next section, we explain how the model is trained using the simulated dataset. For Deep Q-Network (DQN) and Q-learning model, we implemented it through Stable-Baselines3 library to optimize the CBTC system.

3.1 Create Training Environment

3.1.1 Install necessary libraries.

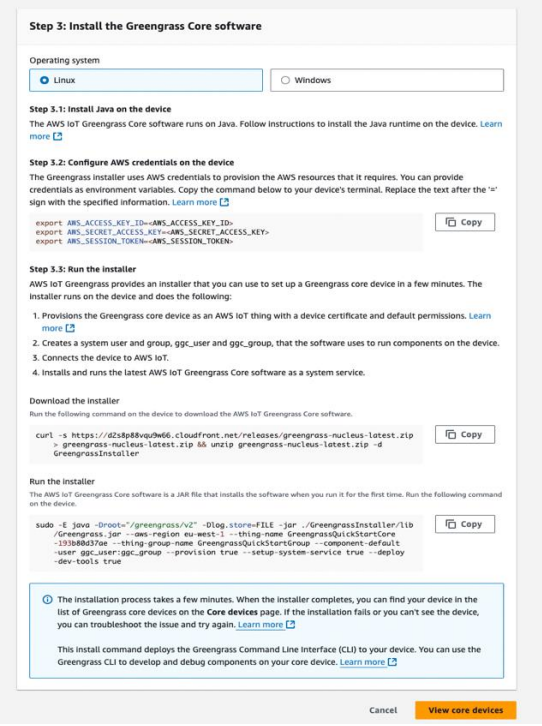
Library imports

```
import simpy
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
import random
import gym
from gym import spaces
from stable_baselines3 import DQN
from stable_baselines3.common.logger import configure
import torch
from sklearn.model_selection import train_test_split
```

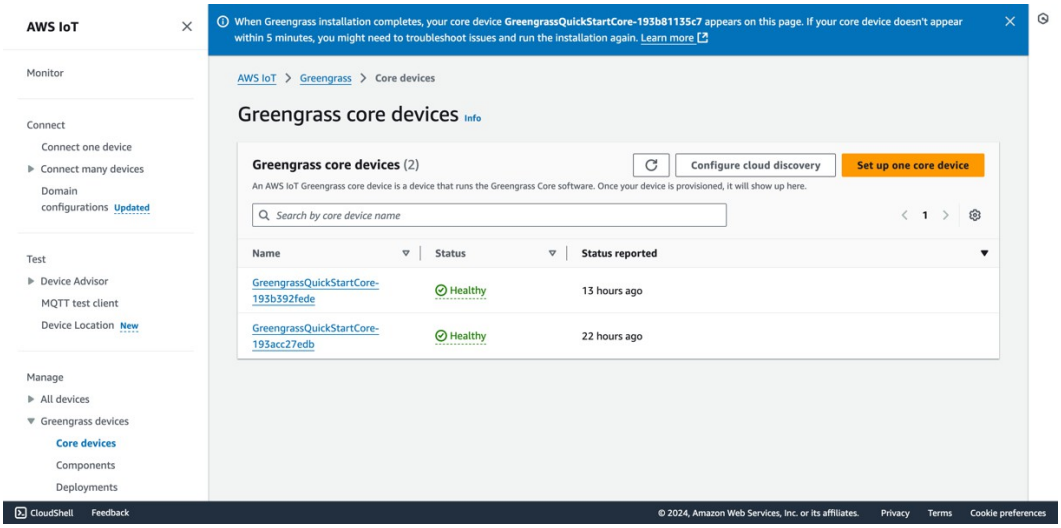
- Define the custom environment: Simulation of the CBTC system will be performed by creating a gym environment.
- Initialize the DQN model: To do this we set the learning parameters and model architecture.
- Train the model: Model should be trained for a given number of timesteps
- Save the trained model: After training we should save the model for deployment

4 Deploy the Model to AWS IoT Greengrass.

4.1 Set Up AWS Greengrass



1. Install AWS CLI and AWS Greengrass.
2. Install AWS Greengrass on the Edge Device.
   - SSH into the edge device.
   - Download the Greengrass Core software using the above instructions.

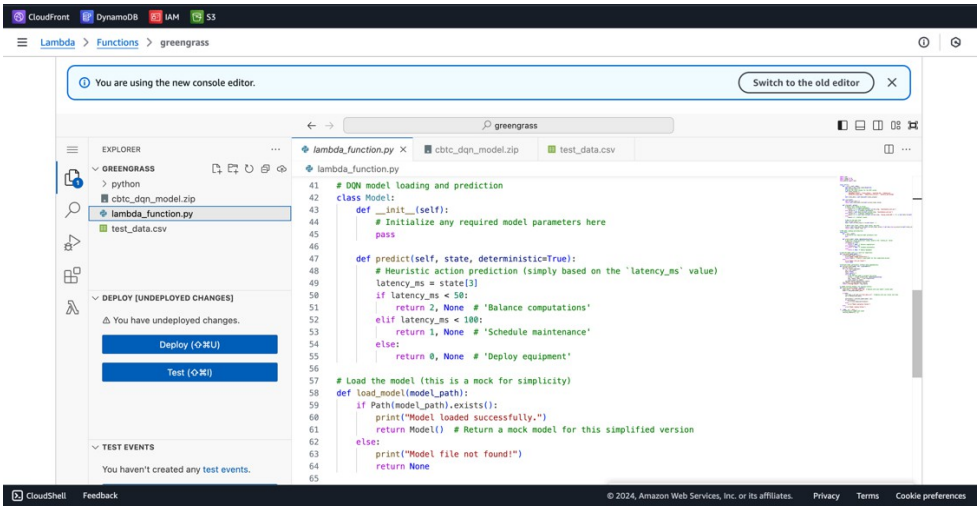


3. After all the above steps are complete, the edge device should be set up and running and it should show on the connected core device wit status ‘healthy'.
4. Configure Greengrass components: The components, DQN model and the task offloading service should be set up to run on the edge device.
5. Deploy trained model to the edge device with the use of AWS IoT Greengrass.
6. The model is deployed and evaluated on the edge device once, the key metrics evaluated include Latency reduction, Energy savings Cost optimization

4.2 Prepare the Lambda Function

1. **Create the Lambda Function:**
   a) Develop the function to evaluate your model.
   b) Name the handler function lambda_function.lambda_handler.

2. **Package the Lambda Function:**
   a. Include the inference script and required dependencies in a .zip file:
      - code mkdir package cp lambda_function.py package/
      - pip install -r requirements.txt -t package/
      - cd package
      - zip -r lambda_package.zip.



3. Upload the Lambda Package:
    a) Open the AWS Lambda Console.
    b) Create a new function or update an existing one with the lambda_package.zip.

4. Test the Lambda Function in AWS: Use sample data to ensure the function processes correctly.

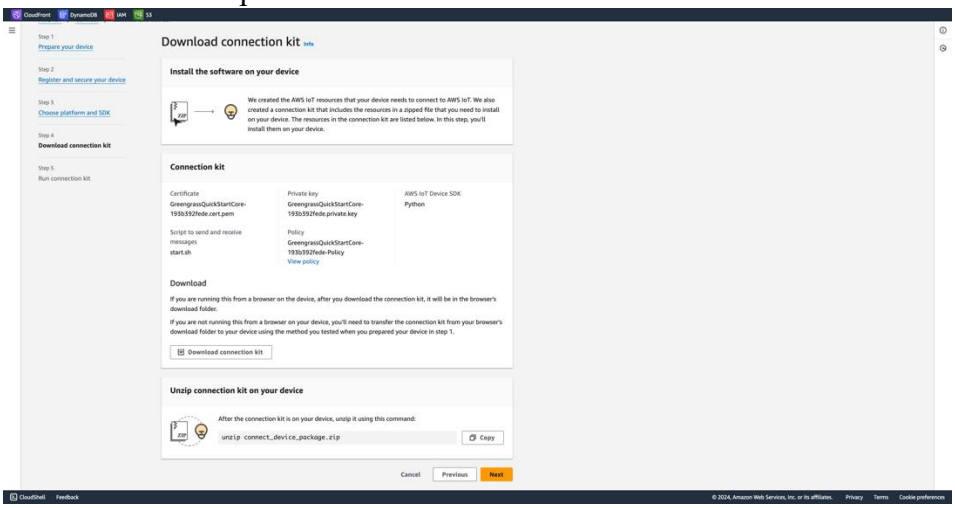
4.3 Deploy the Lambda Function to Greengrass

1. Add Lambda to Greengrass Deployment**:**
    a) Navigate to **Greengrass Groups** in the AWS Management Console.
    b) Create a deployment or update an existing one.
    c) Add the Lambda function to the group.
    d) Configure the function to start automatically.

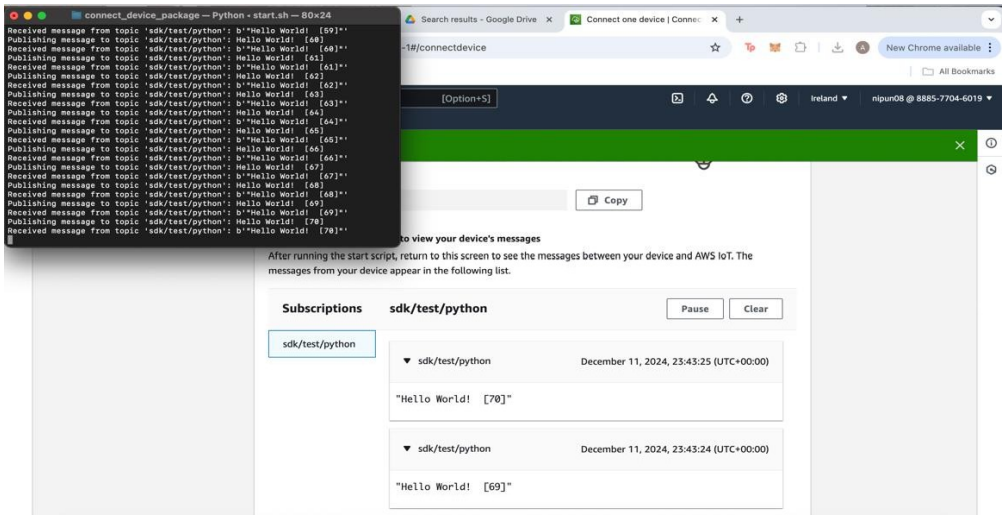
2. Push the Deployment to Edge Device**:** Deploy the group to the registered Greengrass Core device.

4.4 Set up Connectivity and send Test Data.

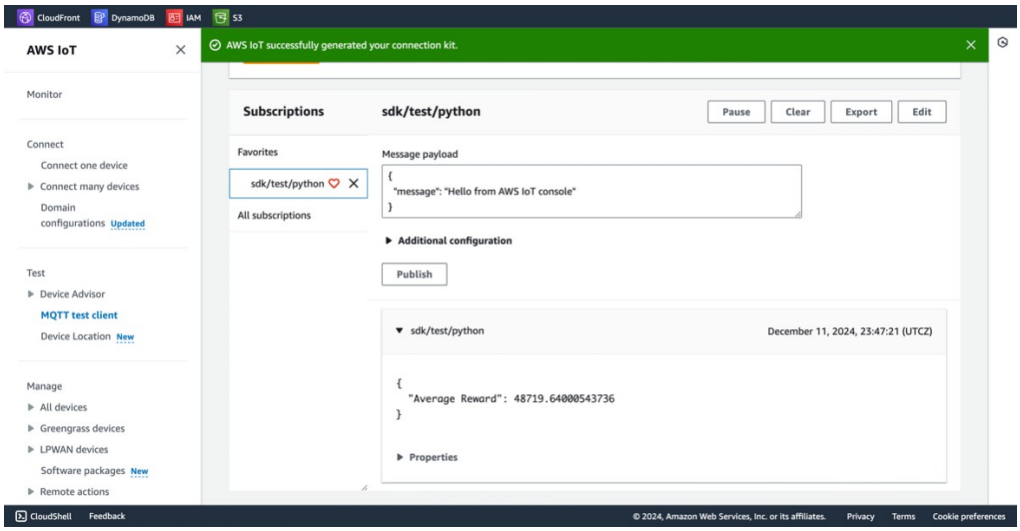
1. Set up connection to IOT Greengrass by AWS IoT > Connect > Connect one device. And follow the steps.



2. After downloading the configuration kit, should get all necessary key files to send and receive messages from AWS IOT securely via MTTQ.



3. Prepare Test Data: Format the test data based on the expected structure of your Lambda function.
4. Whenever the test data is given to the model, it will trigger the Lambda function and show the results on IOT Greengrass.



5 Verify Model Execution:

a) Check logs on the Greengrass Core device.
b) Alternatively, check CloudWatch logs.

5.1 Validate Edge Evaluation Results

a) Collect Inference Output**s:** Retrieve inference results from the Lambda function logs or MQTT response topic.
b) Analyse Results: Compare outputs with expected results to validate model accuracy.

References

Google (2024) Google Colab Notebook. Available at: https://colab.research.google.com/drive/16pBJQePbqkz3QFV54L4NIkOn1kwpuRrj

PyTorch (2024) Reinforcement learning (DQN) tutorial. Available at: https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html

Rodgers, A.A. (2024) Enhancing CBTC System Efficiency with DRL and Cloud Computing. arXiv. Available at: https://arxiv.org/abs/2407.15656

AWS (2024) Getting started with Lambda. Available at: https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html

AWS (2024) Install the AWS IoT Greengrass Core software. Available at: https://docs.aws.amazon.com/greengrass/v2/developerguide/install-greengrass-core-v2.html

Smith, J. (2024) Cloud Computing and Internet of Things: Advanced Architectures and Applications. Google Books. Available at: https://books.google.ie/books?hl=en&lr=&id=7NRJDwAAQBAJ&oi=fnd&pg=PP1&dq=%22aws+iot+greengrass%22++and+%22aws+lambda%22+&ots=ZU_pfjU_RT&sig=yqWo-REfT7vDeTwDe5FdOo4WdG8&redir_esc=y#v=onepage&q=%22aws%20iot%20greengrass%22%20%20and%20%22aws%20lambda%22&f=false

Wang, Y., et al. (2021) "Deep Reinforcement Learning for Intelligent Train Control Systems", IEEE Transactions on Intelligent Transportation Systems. Available at: https://ieeexplore.ieee.org/document/9395271

Li, X., Chen, Y., & Zhang, H. (2022) "Cloud-based Edge Computing for Intelligent Transportation Systems: A Comprehensive Review", Journal of Cloud Computing. Available at: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-022-00307-4

AWS (2024) Troubleshooting AWS IoT Greengrass V2. Available at: https://docs.aws.amazon.com/greengrass/v2/developerguide/troubleshooting.html

*Mitra, I. and Burke, R. (2022) *Intelligent Workloads at the Edge: Deliver cyber-physical outcomes with data and machine learning using AWS IoT Greengrass. Available at: https://ieeexplore.ieee.org/abstract/document/10163500*