

Configuration Manual

MSc Research Project
MSc in Cloud Computing

Ganesh Ram Shanmugam
Student ID: 23244275

School of Computing
National College of Ireland

Supervisor: Aqeel Kazmi

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Ganesh Ram Shanmugam
Student ID: 23244275
Programme: MSc in Cloud Computing **Year:** 2025
Module: MSc Research Project
Lecturer: Aqeel Kazmi
Submission Due Date: 15-09-2025
Project Title: Configuration Manual
Word Count: 929 **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: Ganesh Ram Shanmugam

Date: 15-09-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	Yes
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	Yes
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.	Yes

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

Ganesh Ram Shanmugam
Student ID: 23244275

1 Purpose & Scope

This configuration manual provides information about the sequential procedure of setting up, configuring and implementing the artefact, which was developed as part of the MSc Research Project. The artefact helps to reproduce the Reinforcement Learning (RL) load-balancer to support live cloud streaming workloads using the CloudSim Plus simulation tool and two RL algorithms, Proximal Policy Optimization (PPO) and Deep Q-Network (DQN). The manual is such that it allows one to reproduce the experimental setting and the outcomes by any end user. This document has the following sections related to system requirements, dependencies, installation, configuration, execution, and troubleshooting.

2 Environment Matrix

Table 1: System Requirements

Area	Required / Used Version(s)	Notes
OS	Windows 10/11 or Ubuntu 22.04 LTS	Any modern CPU: no GPU required
Java JDK	Java 8 (1.8.x)	For CloudSim Plus compatibility
Build tools	Eclipse IDE 2025-06 M1, Jupyter notebook, Visual Studio Code.	IDE for using CloudSim Plus, training the model and python development.
Python	3.10.x	For predictor + SB3
Python libs	gym (0.26.x), stable-baselines3 (2.x), numpy, pandas etc.	See requirements.txt
Networking	No TCP port required	Predictor uses stdin/stdout pipe

3 Project Dependencies

3.1 CloudSim Plus

- Version: 6.3.0
- Included in artefact: /lib/cloudsimplus-6.3.0.zip
- File size (local): 16.9 MB

3.2 Python environment

- Predictor relies on Gym + SB3 models trained on the dataset.
- From the requirements.txt file you can see the list like this, gym==0.26. *, stable-baselines3==2. *, etc.

4 Artefact Layout

```
/Project Artefact
/java
  WorkloadSimulation.java
  TestingTrainedModel.java
/lib
  (cloudsimplus-6.3.0.zip)
/python
  streaming_env_multi_vm.py
  predict_server.py
  requirements.txt
/data
  TrainingDataset.csv
/models
  ppo_streaming_multiVM_final_fixed.zip
  dqn_streaming_multiVM_final_fixed.zip
/notebooks
  PPOFinal.ipynb
  PPOLoop.ipynb
  DQNFinal.ipynb
  DQNLoop.ipynb
/outputs
  Final Results.ipynb
  Configuration Manual.docx
```

5 Installation Instructions

1. Install Java JDK and set JAVA_HOME.
2. Install Python 3.10.
3. Install Eclipse IDE, Visual Studio Code, Jupyter Notebook.

4. Clone/download the CloudSim Plus project.
5. Import the project into Eclipse like shown in Figure 1.

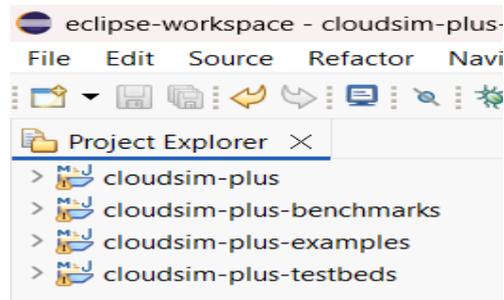


Figure 1: CloudSim Plus Project

6. Create a Python virtual environment and install dependencies using requirements.txt file.

6 Running the Artefact

6.1 Offline Training Phase

1. Run the WorkloadSimulation.java file in this directory, \cloudsimplus-6.3.0\cloudsimplus-6.3.0\cloudsim-plus-examples\src\main\java\org\cloudsimplus\examples\ to generate a dataset called TrainingDataset.csv. On a successful run, the output console will be shown as Figure 2.



Figure 2: Simulation Output

2. Use the streaming_env_multi_vm.py file to create a gym compatible python environment to train the RL models by integrating with the generated dataset.
3. Use this streaming_env_multi_vm.py file to train the PPO model in 2 stages. In the first stage, run the PPOLoop.ipynb file in a Jupyter notebook cell. In the second stage, run the PPOFinal.ipynb file in a another Jupyter notebook cell. And save the trained model as a ppo_streaming_multiVM_final_fixed.zip file. On successful training the output will be shown as Figure 3.

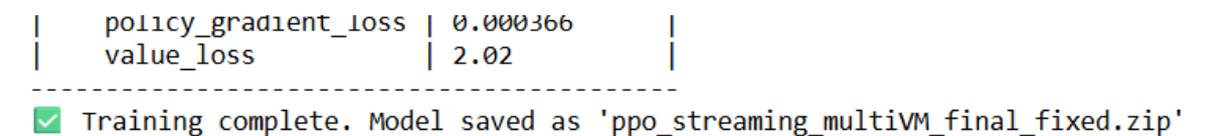


Figure 3: PPO Training Output

4. Use this `streaming_env_multi_vm.py` file to train the DQN model in 2 stages. In the first stage, run the `DQNLoop.ipynb` file in a Jupyter notebook cell. In the second stage, run the `DQNFinal.ipynb` file in another Jupyter notebook cell. And save the trained model as a `dqn_streaming_multiVM_final_fixed.zip` file. On successful training the output will show as Figure 4.

```
Using cpu device
Logging to ./dqn_streaming_tensorboard_fixed/DQN_1
✔ Training complete. Model saved as 'dqn_streaming_multiVM_final_fixed.zip'
```

Figure 4: DQN Training Output

6.2 Online Evaluation Phase

1. Use the `predict_server.py` file to make integration between the simulation environment and the trained RL models.

2. Run the `TestingTrainedModel.java` file in this directory `\cloudsimplus-6.3.0\cloudsimplus-6.3.0\cloudsim-plus-examples\src\main\java\org\cloudsimplus\examples\` to validate the RL models using `predict_server.py` file. Give program arguments names to run the file with the required model, like PPO, DQN or RANDOM as shown in Figure 5. And change the path accordingly in the `predict_server.py` file as shown in Figure 6.

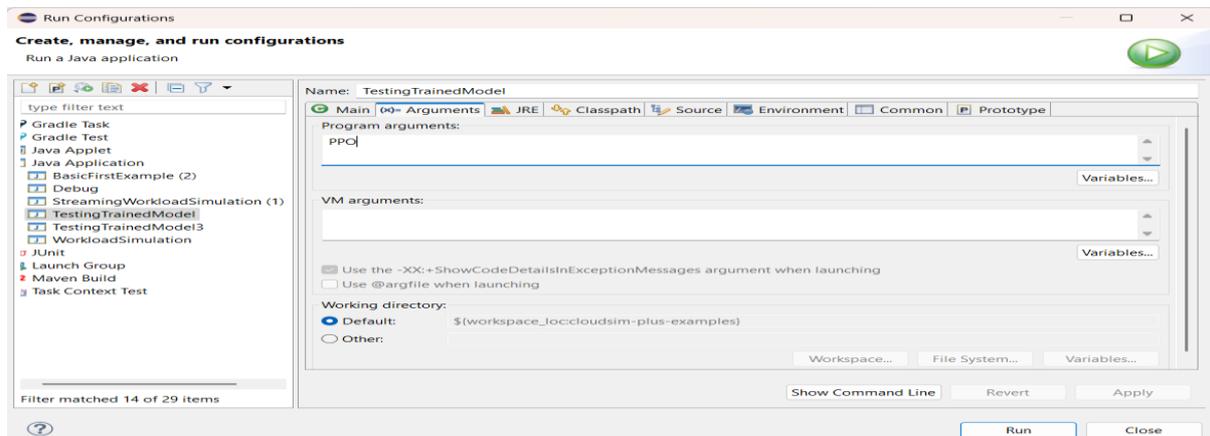


Figure 5: Program Arguments

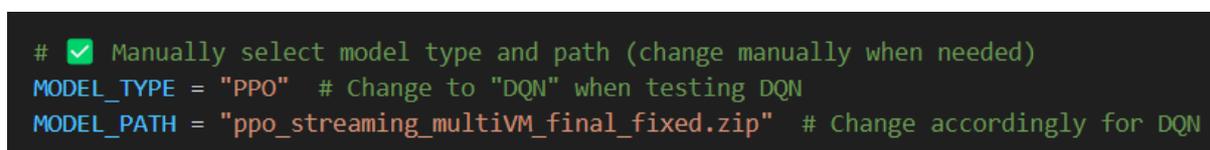


Figure 6: Model Type and Path

3. On a successful run the output will show as Figure 7. Simulation results will be generated into a CSV file named according to the different models. For example, a `model_eval_by_profile_dqn_multivm.csv` file in the artefact.

```
INFO CloudInformationService0: Notify all CloudSim Plus entities to shutdown.
INFO
===== Simulation finished at time 7.21 =====
DEBUG DeferredQueue >> max size: 2 added to middle: 0 added to tail: 22
Scenario: Heavy | Mode: RANDOM | Iteration: 99 | VM: 2 | Latency: 7.000 s | Energy: 0.000408 kwh | Jitter: 0.144 s | Throughput: 342.857 M
Validation completed. Results saved to model_eval_by_profile_random_multivm.csv
```

Figure 7: Evaluation Output

4. Use the Final Results.ipynb file to analyze and compare the performance of each model using graphs.

Note: Make sure all the files are on the same path.

References

CloudSim Plus (2025) *CloudSim Plus Documentation*. Available at: <https://cloudsimplus.org> (Accessed: 9 August 2025).

Eclipse Foundation (2025) *Eclipse IDE for Java Developers – Documentation*. Available at: <https://www.eclipse.org/ide/> (Accessed: 9 August 2025).

Java SE Documentation (2025) *Java Platform, Standard Edition 8 Documentation*. Oracle. Available at: <https://docs.oracle.com/javase/8/docs/> (Accessed: 9 August 2025).

OpenAI (2025) *OpenAI Gym Documentation*. Available at: <https://gym.openai.com> (Accessed: 9 August 2025).

Python Software Foundation (2025) *Python 3.10.12 Documentation*. Available at: <https://docs.python.org/3.10/> (Accessed: 9 August 2025).

Stable-Baselines3 (2025) *Stable-Baselines3 Documentation*. Available at: <https://stable-baselines3.readthedocs.io/> (Accessed: 9 August 2025).

NumPy Developers (2025) *NumPy Documentation*. Available at: <https://numpy.org/doc/> (Accessed: 9 August 2025).

Pandas Development Team (2025) *Pandas Documentation*. Available at: <https://pandas.pydata.org/docs/> (Accessed: 9 August 2025).

Matplotlib Development Team (2025) *Matplotlib Documentation*. Available at: <https://matplotlib.org/stable/contents.html> (Accessed: 9 August 2025).

Microsoft (2025) *Windows 10 – System Requirements and Specifications*. Available at: <https://learn.microsoft.com/en-us/windows/> (Accessed: 9 August 2025).