**Tutoriel simulation Edge Computing**

Télécharger et installer Eclipse IDE : [Eclipse Downloads | The Eclipse Foundation](https://www.eclipse.org/downloads/)

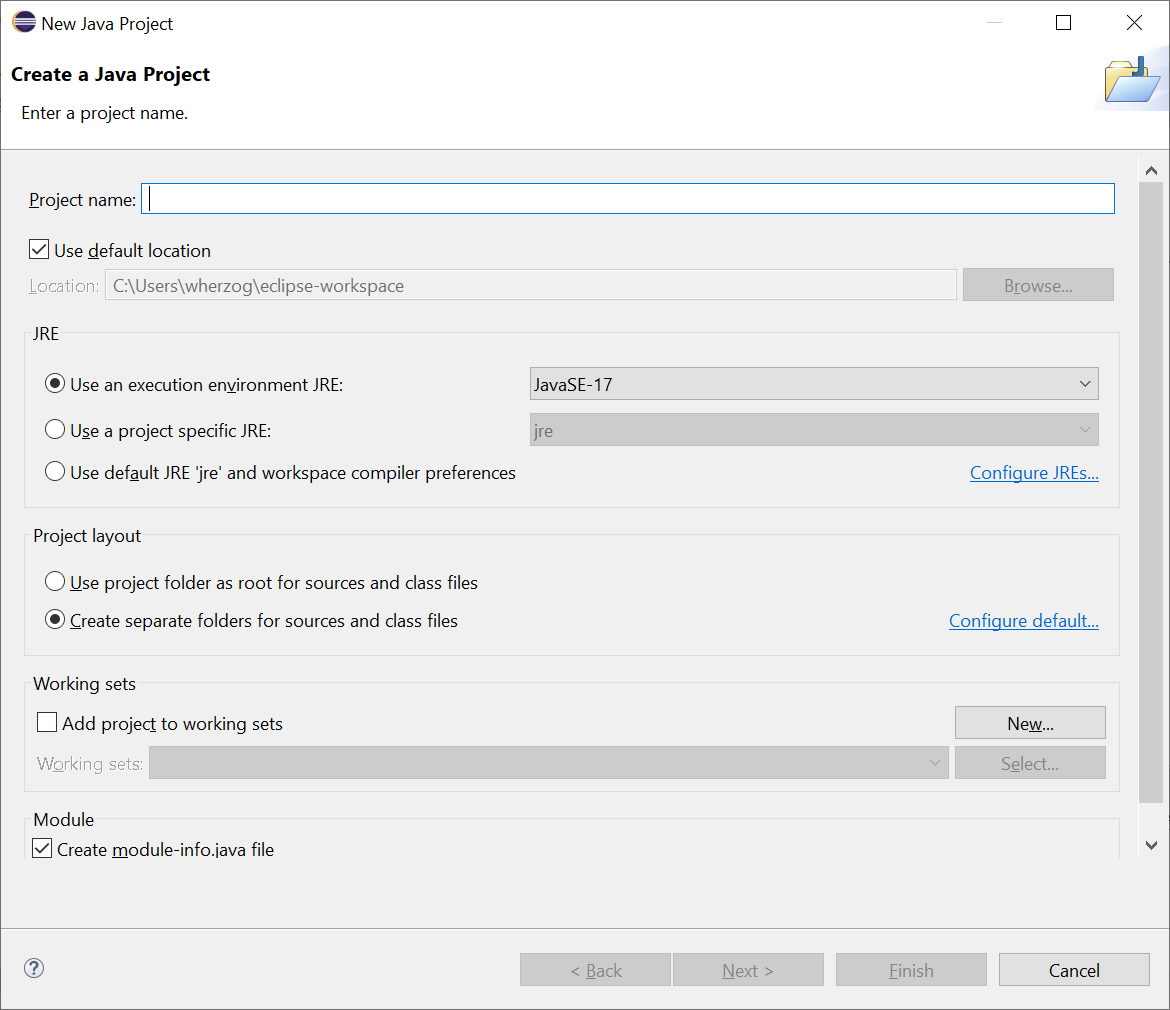
Télécharger EdgeCloudSim :

* Aller sur la page [GitHub - CagataySonmez/EdgeCloudSim: EdgeCloudSim: An Environment for Performance Evaluation of Edge Computing Systems](https://github.com/CagataySonmez/EdgeCloudSim)
* Dans code -> Download ZIP
* Une fois le fichier téléchargé, le décompresser sur le bureau windows
* Ouvrir le fichier décompressé *EdgeCloudSim\_master* et créer un fichier *sim\_results*
* Dans *sim\_results*, créer un fichier *ite1* (attention à la casse)

**Utilisation du Simulateur EdgeCloudSim**

Ouvrir Eclipse IDE

Dans File-> New choisir Java Project. L’écran suivant s’affiche :

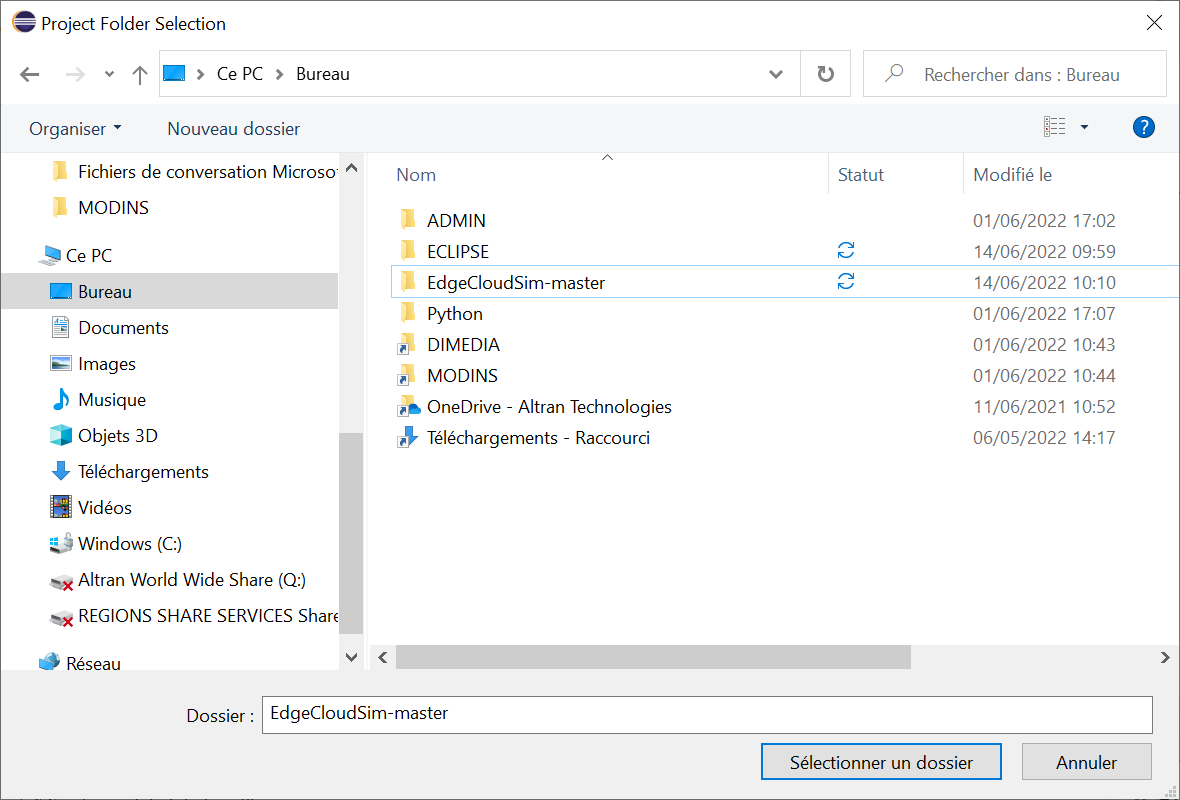


A remplir

Project Name : ECSIM\_test\_+ vos initiales

Décocher *Use default location*  puis *Browse* -> clique sur *Bureau* puis sélectionner le fichier décompressé *EdgeCloudSim-master*

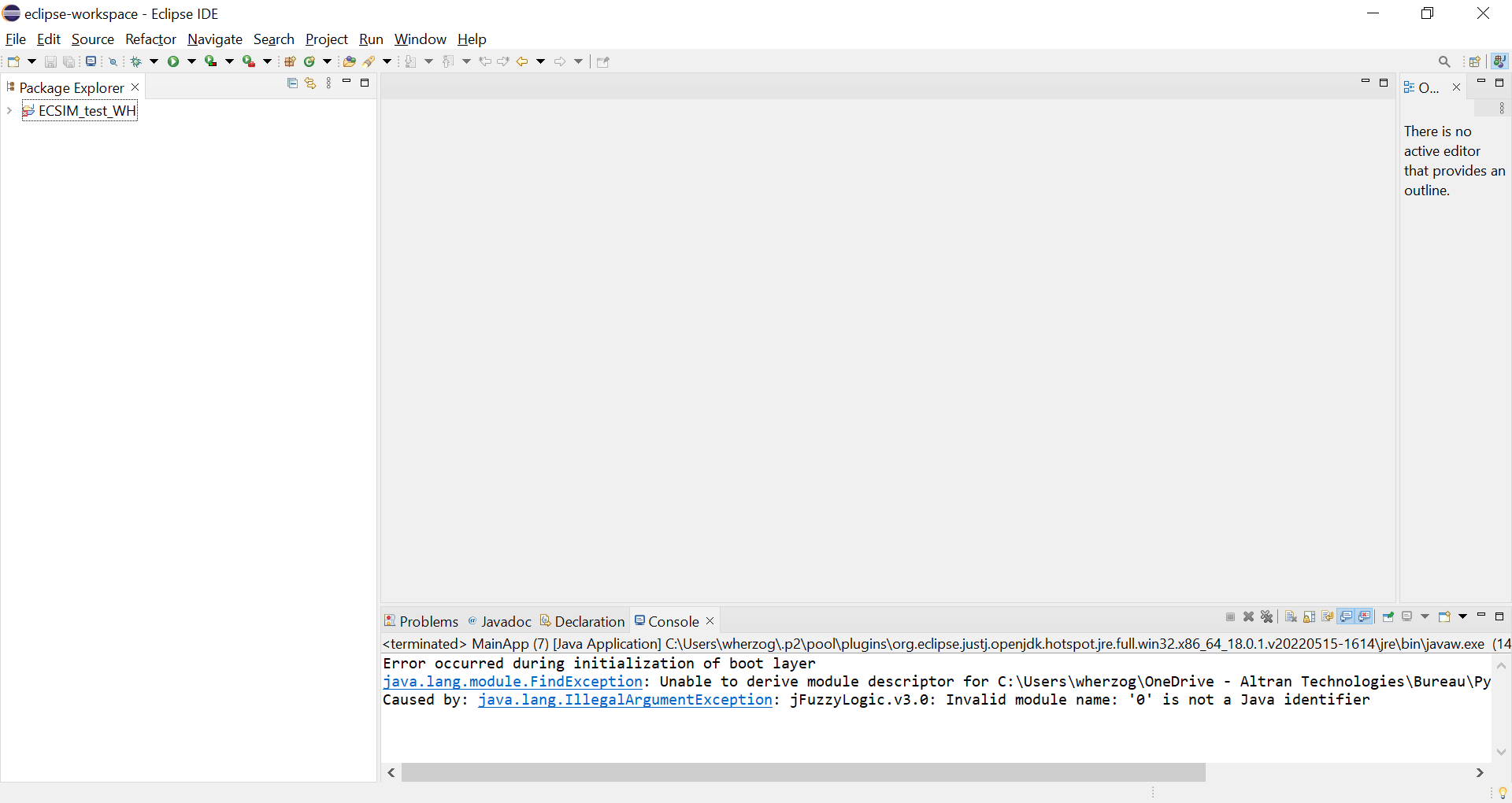
Cliquer sur sélectionner le dossier



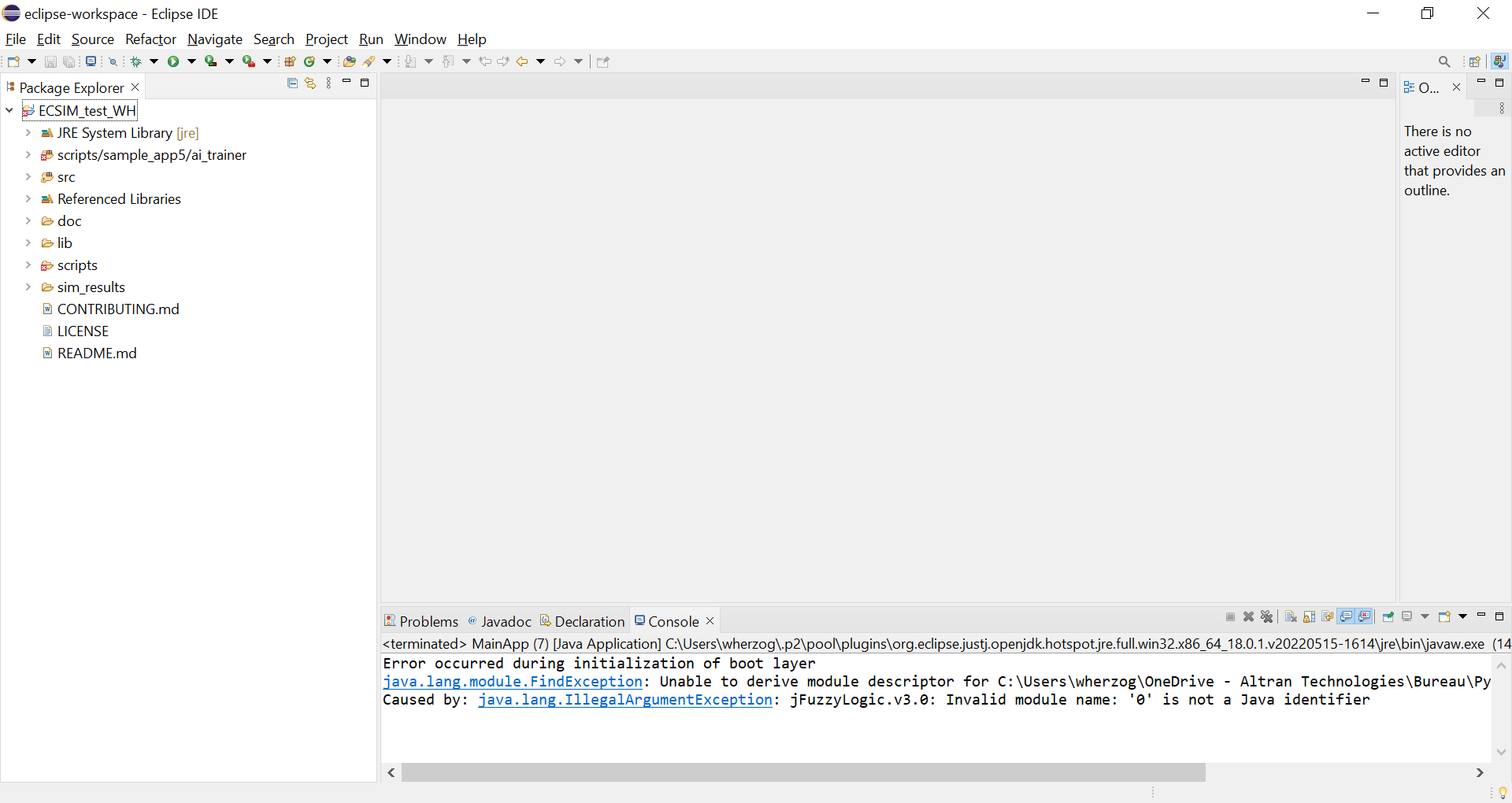
Cliquer sur *Finish* sur la page de création du projet Java

Sur la page suivante *Create module-info.java*, cliquer sur *Don’t Create*

Normalement cette page s’affiche avec le projet ECSIM\_test\_XX :

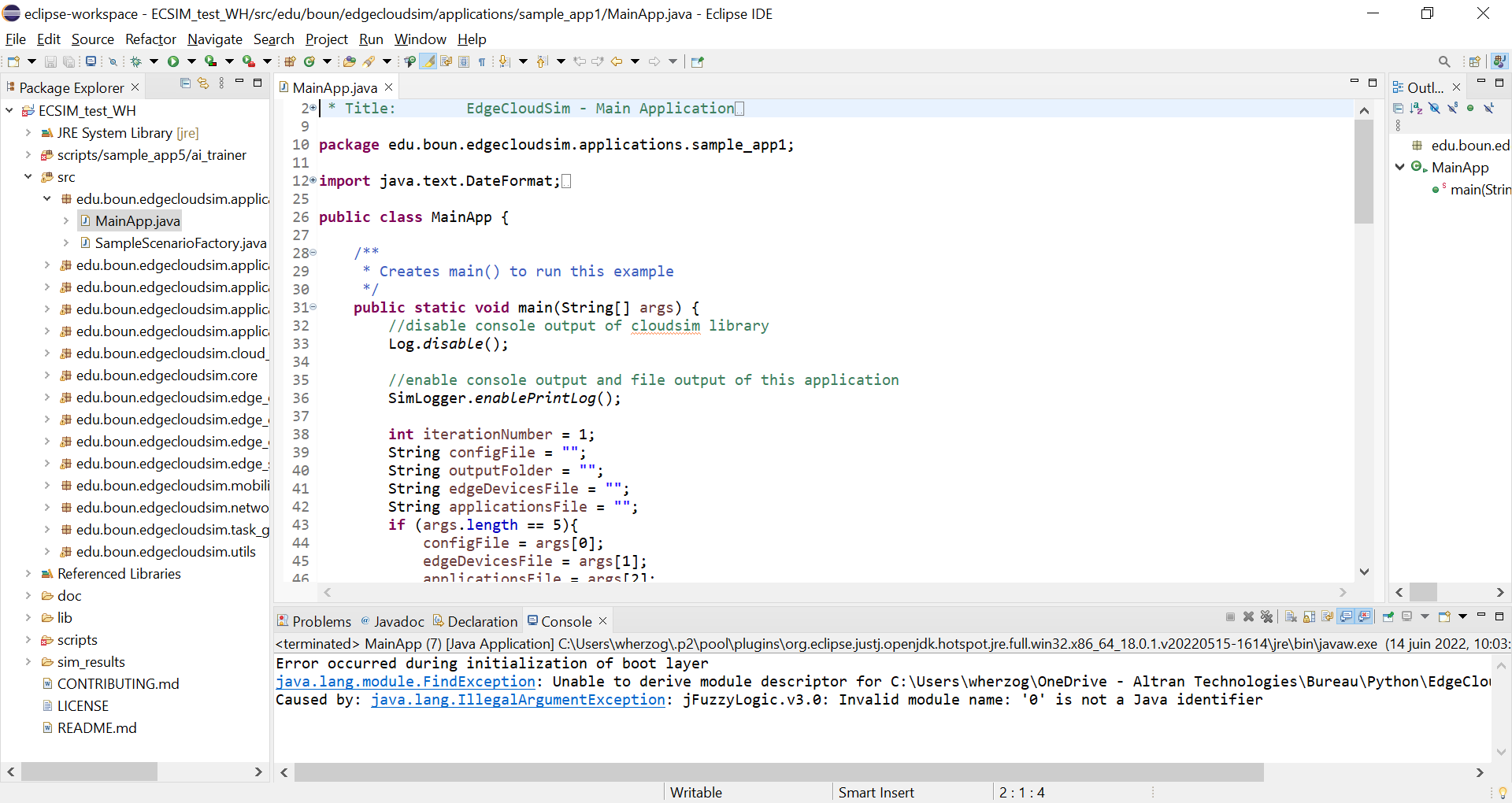


Cliquer sur la petite flèche à côté de ECSIM\_test\_XX pour dérouler le projet :



Dérouler de la même manière *Src* et double-cliquer sur : *Edu.boun.edgecloudsim.applications.sample\_app1* puis double-clic sur *MainApp.java.*

La page suivante s’affiche avec le code du simulateur :



Cliquez sur la flèche verte *Run*

Normalement le simulateur se lance avec un défilement d’information en bas de la page (la console)

**Ressources**

1. Machine Learning-Based Workload Orchestrator for Vehicular Edge Computing
   1. <https://ieeexplore.ieee.org/document/9208723>
   2. <https://github.com/CharafeddineMechalikh/FDT_based_workload_orchestration>
2. Deep Reinforcement Learning-Based Edge Computing Offloading Algorithm for Software-Defined IoT
   1. <https://assets.researchsquare.com/files/rs-2150294/v1_covered.pdf?c=1666752236>

**The download consists of 2 types of RUN programs :**

1. Eclipse Java RUN program : [A] and [B] below explain setup of the Eclipse Java RUN programs for sample\_app1 and sample\_app5. I only looked at sample\_app1 to understand the basic streaming program, and sample\_app5 to understand the last version with the Weka models.
2. Developement files with command line RUN -program :
   1. [C] explains the AI developement files C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\scripts – This is what I need to use to get X-matrix
   2. [D] explains how to use the shell files to RUN sample\_app5’s VehicularMainApp.java

**[A] Main data streaming program (sample\_app1)**

Steps for operating C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\_master\src\edu\boun\edgecloudsim\applications\sample\_app1/MainApp.java:

- 0) download cloudsim source files from <https://github.com/Cloudslab/cloudsim> - unzipped and put the folder cloudbus in the directory C:\Users\jafouche\Documents\Edge\_computing\Additional\_modules\org for a full import into Eclipse.

- 1) download org.apache from https://commons.apache.org/proper/commons-math/download\_math.cgi, <https://dlcdn.apache.org//commons/math/source/commons-math3-3.6.1-src.zip> . Unzipped and put C:\Users\jafouche\Downloads\commons-math3-3.6.1-src\src\main\java\org\apache in in the directory C:\Users\jafouche\Documents\Edge\_computing\Additional\_modules\org for a full import into Eclipse.

- 2) Create folder org in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\src

- 3) File - Open Project from File - select C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim

- 4) In Package Explorer - right click on src [EdgeCloudSim master] - Import - General - File System - Next - Select the Additional file directory C:\Users\jafouche\Documents\Edge\_computing\Additional\_modules\org - check all directories under - \*\*\*Into folder: select src/org\*\*\* (KEY: the packages in C:\Users\jafouche\Documents\Edge\_computing\Additional\_modules\org will appear in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\src\org)

- 5) Add the following jar files from the lib folder to the Project: right-click on src [EdgeCloudSim master] - Build Path - Configure Build Path - click on Libraries tab - click on Modulepath - Add External JARs - go to C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\_master\lib - select colt.jar. Apply - Apply and Close.

- 6) Physically create the outputFolder location (make sure you created a folder called sim\_results in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim, and inside sim\_results make a folder called ite1)

- 7) Click on C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\_master\src\edu\boun\edgecloudsim\applications\sample\_app1/MainApp.java in the Package Explorer. Change the reference path to the .xml files to your absolute path (lines 51 to 60 in MainApp.java); so added C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\ before all the PATH names. Push RUN - the data will be saved to the outputFolder C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\sim\_results\ite1.

**[B] Machine Learning program that uses main data streaming program (sample\_app5)**

Steps for operating C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\_master\src\edu\boun\edgecloudsim\applications\sample\_app5\VehicularMainApp.java:

- 0) Click on C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\_master\src\edu\boun\edgecloudsim\applications\sample\_app5/VehicularMainApp.java in the Package Explorer. Change the reference path to the .xml files to your absolute path (lines 51 to 60 in MainApp.java); so added C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\ before all the PATH names.

- 1) Download the source files for weka from <https://github.com/bnjmn/weka>. I received errors using the given weka.jar (right-click on src [EdgeCloudSim master] - Build Path - Configure Build Path - click on Libraries tab - click on Modulepath - Add External JARs). In Downloads, unzip weka and go to C:\Users\jafouche\Downloads\weka-master\weka\src\main\java. Make a folder called weka in C:\Users\jafouche\Documents\Edge\_computing\Additional\_modules, copy the files under C:\Users\jafouche\Downloads\weka-master\weka\src\main\java into the weka folder. Create a folder called weka in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\src.

Go back to Eclispe, click on File – Refresh to make the weka folder that you created recognizable in the PATH for Eclispe. In the Package Explorer, right click on src [EdgeCloudSim master] - Import - General - File System - Next - Select the Additional file directory C:\Users\jafouche\Documents\Edge\_computing\Additional\_modules\weka - check all directories under - \*\*\*Into folder: select src/weka\*\*\* (KEY: the packages in C:\Users\jafouche\Documents\Edge\_computing\Additional\_modules\weka will appear in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\src\weka).

- 2) Change the reference path for weka in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\_master\src\edu\boun\edgecloudsim\applications\sample\_app5/VehicularMainApp.java. String wekaModelsFolder = "C:\\Users\\jafouche\\Documents\\Edge\_computing\\EdgeCloudSim\\scripts\\sample\_app5\\config\\weka\\";

-3) Push RUN - the data will be saved to the outputFolder C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\sim\_results\ite1.

- 4) Download SMOTE package for class balancing

<https://sourceforge.net/projects/weka/files/weka-packages/SMOTE1.0.3.zip/download?use_mirror=kumisystems&download=>

Ensure that SMOTE.java is in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\src\weka\filters\supervised\instance\SMOTE.java

**[C] Shell files (They give the option of running the the program autonomously without Eclispe)**

The .sh files in C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\scripts\sample\_app5 are there to run the VehicularMainApp.java program by command line WITHOUT Eclispe, and collect the output log files.

compile.sh (just runs javac), run\_scenarios.sh (runs javac, creates folders and log files, Why use this program when the results are stored in sim\_results/ite1??), runner.sh (runs javac), stop\_scenarios.sh.

* None of these shell program Run correctly

What I did : MingWin64 terminal (you have to use absolute paths):

CLI (Command Line Interface) compile command :

javac -d /c/Users/jafouche/Documents/Edge\_computing/out -classpath "/c/Users/jafouche/Documents/Edge\_computing/EdgeCloudSim/lib/colt.jar;/c/Users/jafouche/Documents/Edge\_computing/EdgeCloudSim/lib/mtj-1.0.4.jar" -sourcepath /c/Users/jafouche/Documents/Edge\_computing/EdgeCloudSim/src /c/Users/jafouche/Documents/Edge\_computing/EdgeCloudSim/src/edu/boun/edgecloudsim/applications/sample\_app5/VehicularMainApp.java

- Runs, but has 100 errors - missing packages that need to be put in place (Eclispe is able to run the file without these packages)

import no.uib.cipr.matrix

import java\_cup.runtime

import com.googlecode.jfilechooserbookmarks

import javax.xml.bind

- If these files are downloaded and put into the src folder, the program can be compiled manually.

**[D] Sequentical Functioning of VehicularMainApp.java**

0) SimLogger - launches the simulation

1) Specify the simulation settings in the C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\scripts\sample\_app5\config\default\_config.properties file. Fields of interest to change are :

a) orchestrator\_policies = ["AI\_BASED", "AI\_TRAINER", "RANDOM", "MAB", "GAME\_THEORY", "PREDICTIVE", "Q\_LEARNING"]. The policy tells the simulation how to command the network to choose GSM, EDGE, or RSU. The file C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\src\edu\boun\edgecloudsim\applications\sample\_app5\VehicularScenarioFactory.java has the code for comanding the network based on an ACTION/prediction\_result.

b) simulation\_scenarios=ITS\_SCENARIO is set to this, and it seems like this corresponds to the type of application [

simulation\_scenarios=SINGLE\_TIER,TWO\_TIER,TWO\_TIER\_WITH\_EO]

1) VehicularEdgeOrchestrator.java

- Give the policy and it commands the network to select EDG, GSM, RSU

- For the AI\_BASED policy, it calls WekaWrapper.java which gets the feature data

a) The « AI\_BASED «   policy loads pre-trainned weka models (C:\Users\jafouche\Documents\Edge\_computing\EdgeCloudSim\scripts\sample\_app5\config\weka) – lr\_cloud\_gsm, lr\_cloud\_rsu, lr\_edge, mlp\_cloud\_gsm, mlp\_cloud\_rsu, mlp\_edge

b) The WekaWrapper.java has two main functions (handleRegression, handleClassification) – they get the data from [edge, cloud\_rsu, cloud\_gsm] which is called values (which is a new raw value for a particular attribute (attribute values are defined for each type\_of\_network and type\_of\_prediction) ) attributes, mean, std. He baseline shifts the raw value with respect to the mean and std of the specified attribute.

Lists of attributes :

* EDGE\_REGRESSION\_ATTRIBUTES = {"TaskLength","AvgEdgeUtilization","ServiceTime"};
* EDGE\_CLASSIFIER\_ATTRIBUTES = {"NumOffloadedTask","TaskLength","WLANUploadDelay","WLANDownloadDelay","AvgEdgeUtilization"};
* CLOUD\_RSU\_REGRESSION\_ATTRIBUTES = {"TaskLength","WANUploadDelay","WANDownloadDelay","ServiceTime"};
* CLOUD\_RSU\_CLASSIFIER\_ATTRIBUTES = {"NumOffloadedTask","WANUploadDelay","WANDownloadDelay"};
* CLOUD\_GSM\_REGRESSION\_ATTRIBUTES = {"TaskLength","GSMUploadDelay","GSMDownloadDelay","ServiceTime"};
* CLOUD\_GSM\_CLASSIFIER\_ATTRIBUTES = {"NumOffloadedTask","GSMUploadDelay","GSMDownloadDelay"};