# Configuration Manual

MSc Research Project
Cloud Computing


Anil Judhistira Gauda
Student ID: x18180663

School of Computing
National College of Ireland


Supervisor: Vikas Sahni

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Anil Judhistira Gauda |
| **Student ID:** | x18180663 |
| **Programme:** | Cloud Computing |
| **Year:** | 2018 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Vikas Sahni |
| **Submission Due Date:** | 17/08/2020 |
| **Project Title:** | Configuration Manual |
| **Word Count:** | 800 |
| **Page Count:** | 7 |

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.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

| | |
|---|---|
| **Signature:** | |
| **Date:** | 15th August 2020 |

**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

Anil Judhistira Gauda
x18180663

# 1 Cloud Environment Setup

## 1.1 Prerequisite

The scheduler is developed using JADE framework [1], this framework is developed on Java programming language and prominently used for agent development. Agent created using JADE are deployed on a cloud environment to execute the workload. For communication between the agents deployed on the cloud environment and scheduler, JSCH [2] library is used. JSCH by jcraft is used to implement SSH protocol (Secured Shell) to transfer logs and workloads into the cloud environment.
Virtual machines that are to be used for scheduling the workloads should have valid credentials (i.e. domain name, private key and username). Virtual machines can be from any cloud service provider or even on-premise private cloud. For this research, Openstack is considered to be a private cloud and AWS is used as a public cloud. As for the database, the scheduler supports Postgres database, so before attempting to schedule the workloads, make sure that the database is installed in the machine and credentials (i.e. connection URL, user name and password) are saved.

## 1.2 Java Installation

Login into the virtual machine using credentials. Then install JDK using following command.
"sudo yum install java-1.8.0-openjdk-devel.x86_64 -y"
Verification:

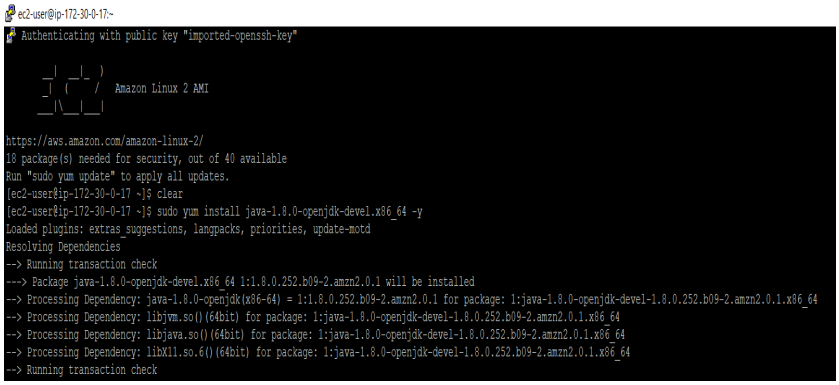



Figure 1: Java Installation

Set Java in the environment variable for the Linux environment using the following command.
“export JAVA_HOME=/usr/lib/jvm/java”
“export PATH=$PATH:$JAVA_HOME/bin”
Verification:

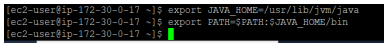



Figure 2: Environment variable

## 1.3 Maven Installation

The agent is developed using a build tool in to manage dependency, so it is necessary to install and configure Maven. Install maven build tool in the virtual machine using the following command.
“sudo yum install maven -y” Verification:

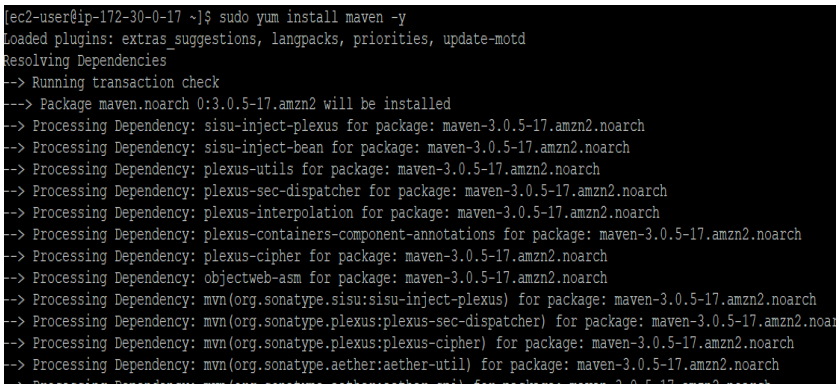



Figure 3: Maven Installation

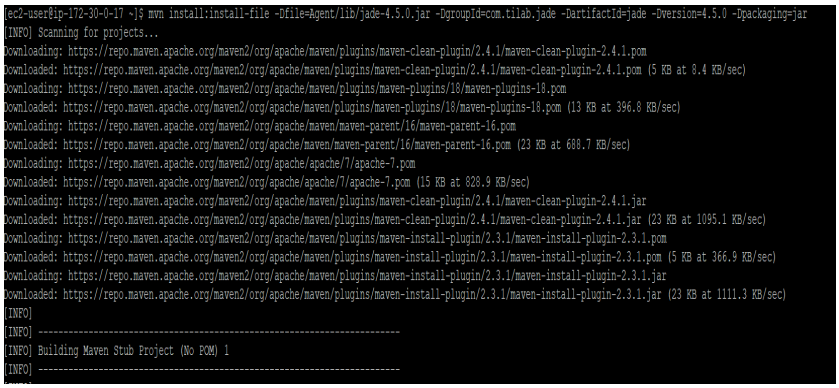
## 1.4 Agent Setup

Transfer “Agent.tar” to a virtual machine in the user’s home directory. Extract the agent in the same directory using the following command.
“tar -xvf Agent.tar”
Verification:

```
[ec2-user@ip-172-30-0-17 ~]$ tar -xvf Agent.tar
Agent/
Agent/.classpath
Agent/.project
Agent/.settings/
Agent/.settings/org.eclipse.core.resources.prefs
Agent/.settings/org.eclipse.jdt.core.prefs
Agent/.settings/org.eclipse.m2e.core.prefs
Agent/APDescription.txt
Agent/lib/
Agent/lib/jade-4.5.0.jar
Agent/lib/jade-test-suite-1.13.0.jar
Agent/MTPs-Main-Container.txt
```

Figure 4: Agent Extract

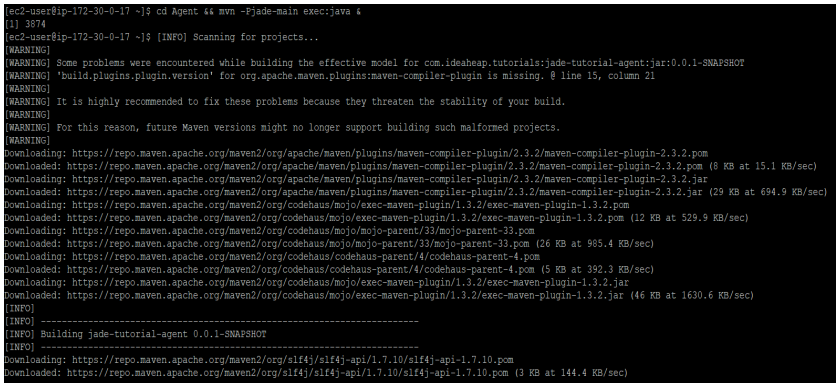
Execute following commands to install JADE dependency in maven
"mvn install:install-file -Dfile=Agent/lib/jade-4.5.0.jar -DgroupId=com.tilab.jade -DartifactId=jade -Dversion=4.5.0 -Dpackaging=jar"
"mvn install:install-file -Dfile=Agent/lib/jade-test-suite-1.13.0.jar -DgroupId=com.tilab.jade -DartifactId=jade-test-suite -Dversion=1.13.0 -Dpackaging=jar"
Verification:



```
[ec2-user@ip-172-30-0-17 ~]$ mvn install:install-file -Dfile=Agent/lib/jade-4.5.0.jar -DgroupId=com.tilab.jade -DartifactId=jade -Dversion=4.5.0 -Dpackaging=jar
[INFO] Scanning for projects...
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.4.1/maven-clean-plugin-2.4.1.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.4.1/maven-clean-plugin-2.4.1.pom (5 KB at 8.4 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/18/maven-plugins-18.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/18/maven-plugins-18.pom (13 KB at 396.8 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/16/maven-parent-16.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/16/maven-parent-16.pom (23 KB at 688.7 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/apache/7/apache-7.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/apache/7/apache-7.pom (15 KB at 828.9 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.4.1/maven-clean-plugin-2.4.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.4.1/maven-clean-plugin-2.4.1.jar (23 KB at 1095.1 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.3.1/maven-install-plugin-2.3.1.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.3.1/maven-install-plugin-2.3.1.pom (5 KB at 366.9 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.3.1/maven-install-plugin-2.3.1.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.3.1/maven-install-plugin-2.3.1.jar (23 KB at 1111.3 KB/sec)
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] ------------------------------------------------------------------------
```

Figure 5: Jade dependency

Start agents in virtual machine using following commands.
"cd Agent  mvn -Pjade-main exec:java &"
"cd Agent  mvn -Pjade-agent exec:java &"
Verification:



```
[ec2-user@ip-172-30-0-17 ~]$ cd Agent && mvn -Pjade-main exec:java &
[1] 3874
[ec2-user@ip-172-30-0-17 ~]$ [INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for com.ideaheap.tutorials:jade-tutorial-agent:jar:0.0.1-SNAPSHOT
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-compiler-plugin is missing. @ line 15, column 21
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/2.3.2/maven-compiler-plugin-2.3.2.pom
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/2.3.2/maven-compiler-plugin-2.3.2.pom (8 KB at 15.1 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/2.3.2/maven-compiler-plugin-2.3.2.jar
Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-compiler-plugin/2.3.2/maven-compiler-plugin-2.3.2.jar (29 KB at 694.9 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/mojo/exec-maven-plugin/1.3.2/exec-maven-plugin-1.3.2.pom
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/mojo/exec-maven-plugin/1.3.2/exec-maven-plugin-1.3.2.pom (12 KB at 529.9 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/mojo/mojo-parent/33/mojo-parent-33.pom
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/mojo/mojo-parent/33/mojo-parent-33.pom (26 KB at 985.4 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/codehaus-parent/4/codehaus-parent-4.pom
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/codehaus-parent/4/codehaus-parent-4.pom (5 KB at 392.3 KB/sec)
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/mojo/exec-maven-plugin/1.3.2/exec-maven-plugin-1.3.2.jar
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/mojo/exec-maven-plugin/1.3.2/exec-maven-plugin-1.3.2.jar (46 KB at 1630.6 KB/sec)
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building jade-tutorial-agent 0.0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
Downloading: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.10/slf4j-api-1.7.10.pom
Downloaded: https://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.7.10/slf4j-api-1.7.10.pom (3 KB at 144.4 KB/sec)
```

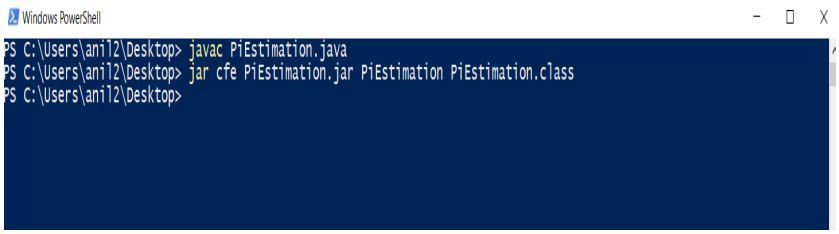
Figure 6: Start Agent

# 2 Scheduler Setup Guide

Developed scheduler is an Java application, So to run this application it is necessary to install and configure java to run the scheduler. Start application using following command in the in the Artefacts/scheduler/deployable "java -jar IntercloudScheduler.jar"

## 2.1 Workload Creation

Jobs are jars that will be executed by the agents. To create jars use following commands
Compile java program: javac <java_file_name>.java (e.g. javac PiEstimation.java)
Create jar: jar cfe <your_jar_name>.jar <java_file_name> <class_file_name>.class (e.g. jar cfe PiEstimation.jar PiEstimation PiEstimation.class)
Verification:



```
Windows PowerShell
PS C:\Users\anil2\Desktop> javac PiEstimation.java
PS C:\Users\anil2\Desktop> jar cfe PiEstimation.jar PiEstimation PiEstimation.class
PS C:\Users\anil2\Desktop>
```

Figure 7: Create workload

## 2.2 Database Creation

As application is storing job execution details in a postgres database, It is mandatory to have postgres database configured in the system with scheduler. Execute database scripts from project artefacts "artefacts/Database_scripts"

## 2.3 Scheduler Walk-through

### 2.3.1 Register Database:

The developed application allows user to register single database and it creates a “MAS.properties” file to store that credentials. for success full setup of the scheduler it necessary to register the database that will have all the database scripts present in Artefacts to be executed successfully.
Verification:

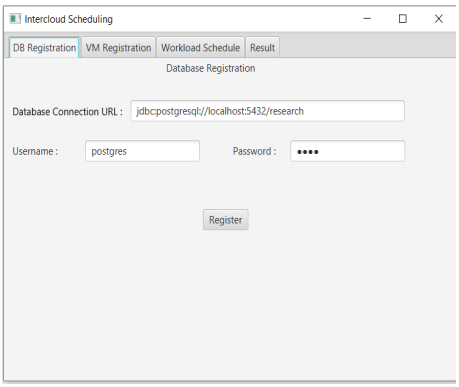



Figure 8: Register Database

### 2.3.2 Register Virtual Machines:

Virtual machines need to be configured and registered with the application so that it will be considered while scheduling the workloads. Register the virtual machine as shown below in the screenshot.
Verification:

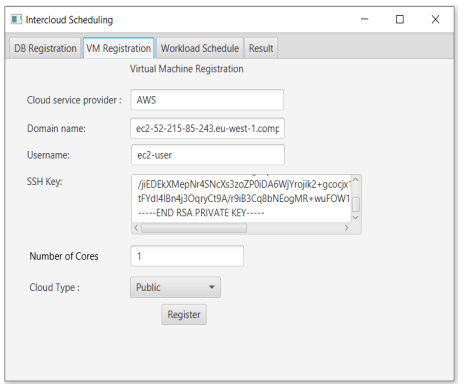



Figure 9: Register Virtual Machine

### 2.3.3 Workload Scheduling:

As soon as the application is started the ”workload” directory will be created in the current location of the scheduler. Then copy all the workloads into the workload folder

then following steps

1. Click on check for jobs on workload folder.
2. Click on check for the naming of jobs.
3. Assign user preference in between 0 to 100. 0 for complete private cloud scheduling and 100 for complete public cloud scheduling.

Verification:

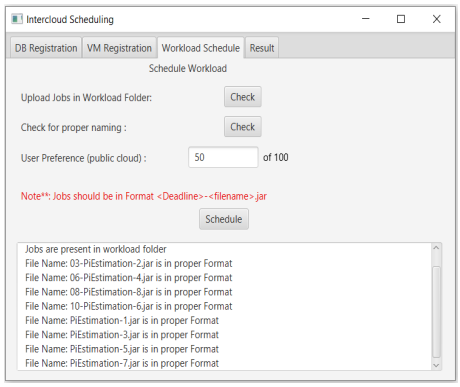



Figure 10: Workload Scheduling

### 2.3.4 Evaluation

As the workloads have completed the execution scheduler fetched the logs from the cloud environment. These logs are parsed by the scheduler and stored in the database with table name "jobdetails". The result obtained in also available in "Artefacts/Evaluation" directory.
Verification:

| | id [PK] numeric | name character varying | ttime numeric | ctime numeric | vm_id numeric | userpref double precision | deadline integer |
|---|---|---|---|---|---|---|---|
| 9 | 9 | 08-PiEstimation-8.jar | 3 | 861 | 1 | 50 | 8 |
| 10 | 10 | 02-PiEstimation-5.jar | 32 | 136 | 2 | 50 | 2 |
| 11 | 11 | 03-PiEstimation-2.jar | 22 | 139 | 2 | 50 | 3 |
| 12 | 12 | 04-PiEstimation-1.jar | 13 | 137 | 2 | 50 | 4 |
| 13 | 13 | 05-PiEstimation-3.jar | 3 | 135 | 2 | 50 | 5 |
| 14 | 14 | 06-PiEstimation-4.jar | 3 | 598 | 3 | 50 | 6 |
| 15 | 15 | 08-PiEstimation-8.jar | 3 | 617 | 1 | 50 | 8 |
| 16 | 16 | 10-PiEstimation-6.jar | 3 | 649 | 5 | 50 | 10 |
| 17 | 17 | 09-PiEstimation-7.jar | 3 | 711 | 4 | 50 | 9 |

Figure 11: Database

# References

[1] B. Fabio, P. Agostino, and R. Giovanni, "Developing multi-agent systems with jade.," *Intelligent Agents VII Agent Theories Architectures and Languages : 7th International Workshop, ATAL 2000 Boston, MA, USA, July 7–9, 2000 Proceedings*, p. 89, 2001.

[2] Jcraft.com, "Java Secure Channel." `http://www.jcraft.com/jsch/`, 2020. [Online; accessed 09-August-2020].