Hazelcast Certification – Code Challenge

Compile & Run

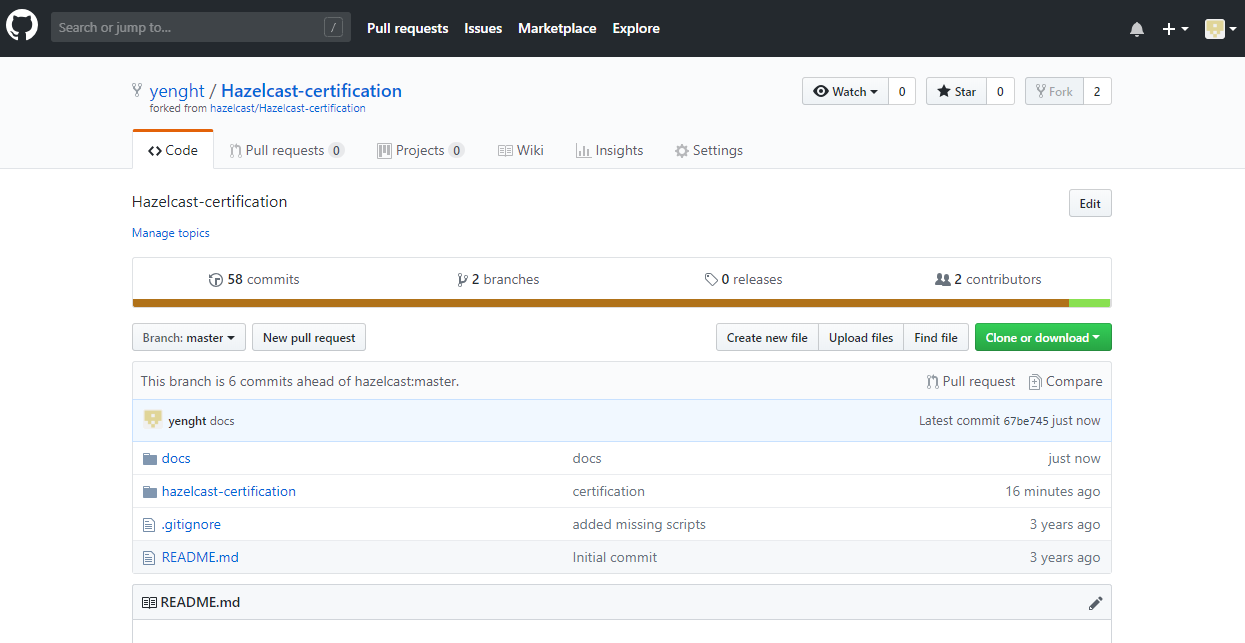
# Introduction

This document contains instructions to compile and run the entire solution for the Hazelcast Certification’s Code Challenge.

# Code compilation

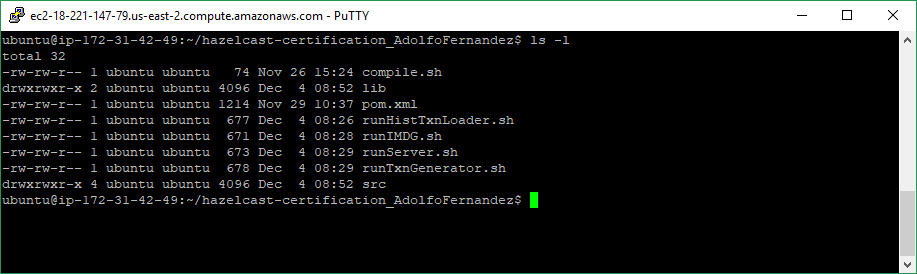
## Download the solution from GitHub

1. Download the solution from <https://github.com/yenght/Hazelcast-certification>



## Unzip the solution

1. Copy and unzip the ZIP file containing the source code and the scripts on any destination of your choice.
2. Once unzipped the content of the solution’s root directory should looks like this:



1. Grant permissions for the scripts execution

chmod 764 \*.sh

## Requirements and configuration

Code compilation relies in Apache Maven:

1. Download and install JDK 1.7 or above.
2. Download and install Apache Maven 3.3.x or above.

For further details on how to install JAVA or Maven please refer to the appendix 1.

## Configure the discovering mechanisms

Before compile and running the solution you should modify the ‘FraudDetection.properties’, ‘hazelcast.xml’ and ‘hazelcast-client.xml’ files according to you environment.

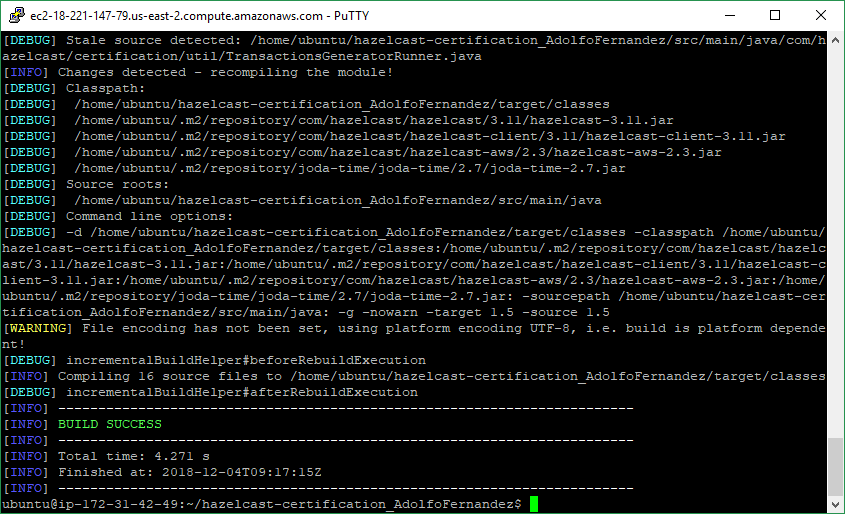
The configuration provided for ‘hazelcast.xml’ and ‘hazelcast-client.xml’ uses AWS Cloud Discovery; and ‘FraudDetection.properties’ it is prepared to run the transactions generator and the fraud detection process in the same server

These configuration elements are more detailed described in the solution’s design document.

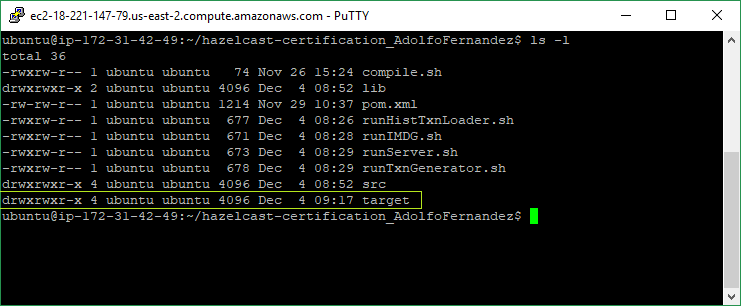
## Perform the code compilation

1. From the solution’s root directory **execute ‘compile.sh’**.

It will end with a ‘BUILD SUCCESS’ message like this:



And a folder called ‘target’ will be created:



# Running the solution

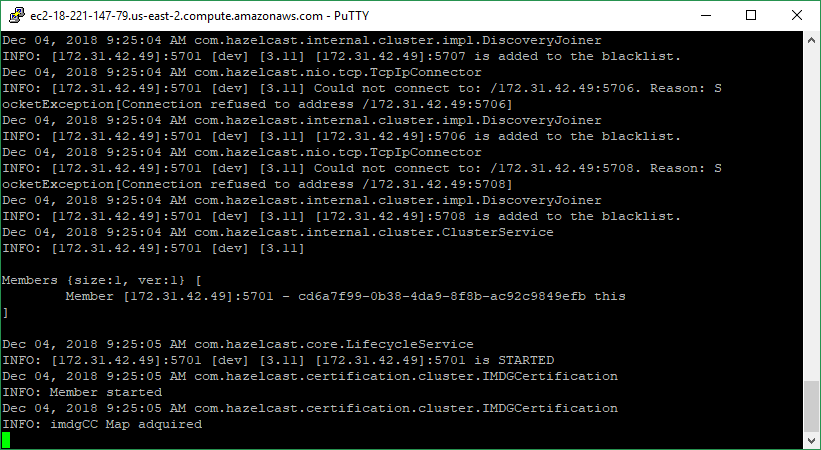
The solution has several components and they have to be started as follow:

1. Start the cluster members.
2. Run the Historical Transaction Loader process until it’s finished.
3. Run the Transaction Generator
4. Run the Fraud Detection Server.

## Start a cluster member

1. Review the ‘runIMDG.sh’ file and modify JVM memory parameters and the classpath according to your resources and your directory structure
2. From the solution’s root directory **execute ‘runIMDG.sh’**.

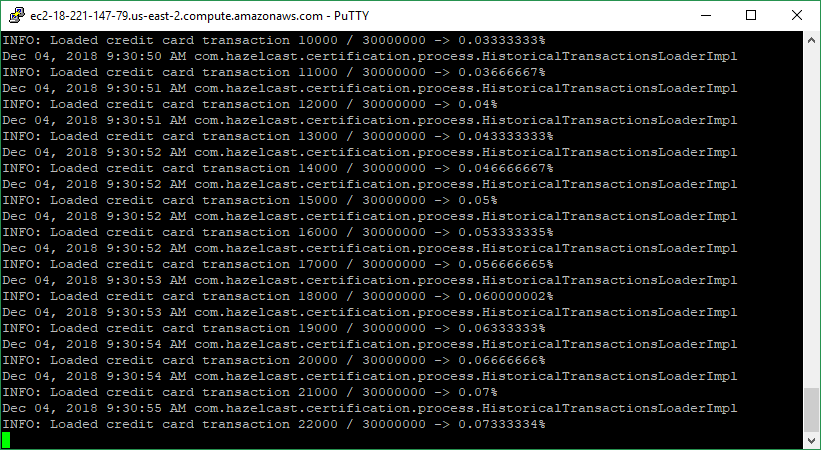
This will start a member of the cluster:



## Run the Historical Transaction Loader process

1. Review the ‘runHistTxnLoader.sh’ file and modify JVM memory parameters and th classpath according to your resources and your directory structure
2. From the solution’s root directory **execute ‘runHistTxnLoader.sh’**.

This will run the Historical Transaction Loader process:



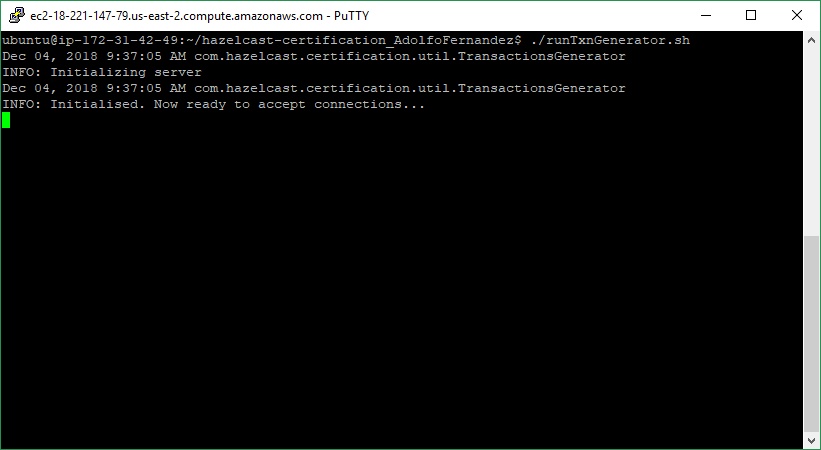
It will be printing a trace with the completion percentage of the loading process.

When the transactions for the 30 million of credits card are loaded into the cluster the process will shutdowns itself.

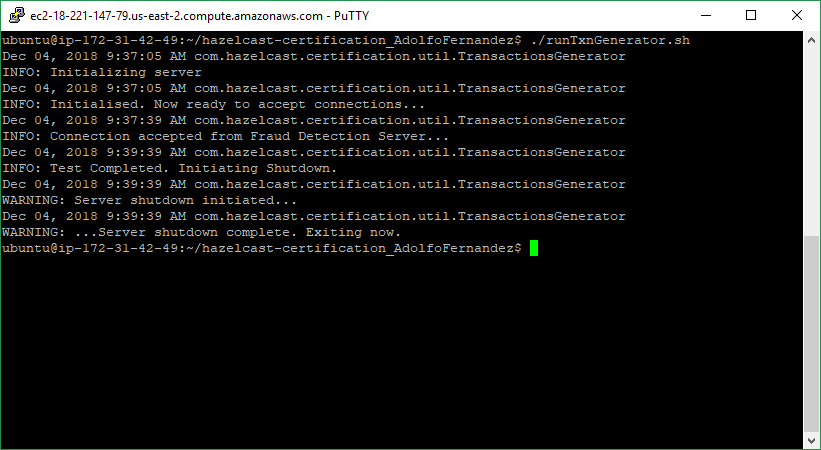
## Run the Transaction Generator

1. Review the ‘runTxnGenerator.sh’ file and modify JVM memory parameters and the classpath according to your resources and your directory structure
2. From the solution’s root directory **execute ‘runTxnGenerator.sh’**.

The process will wait until a Fraud Server is started:



When a Fraud Detection Server is started the process will begin to produce transactions:

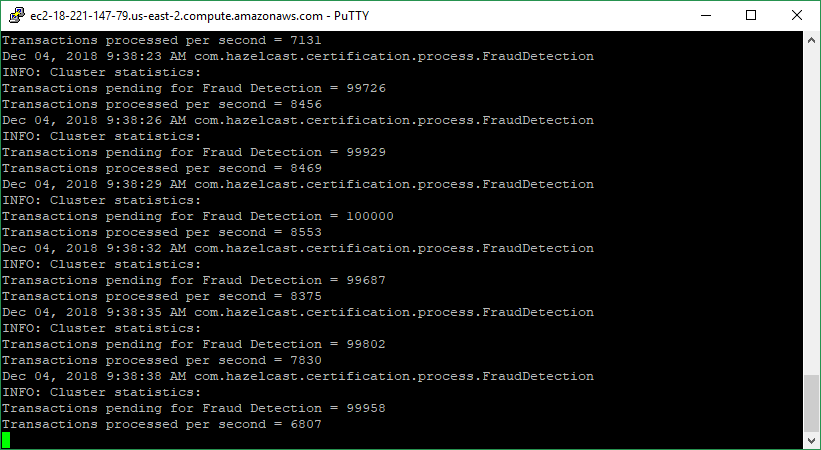


And it will be terminated when the process time expires.

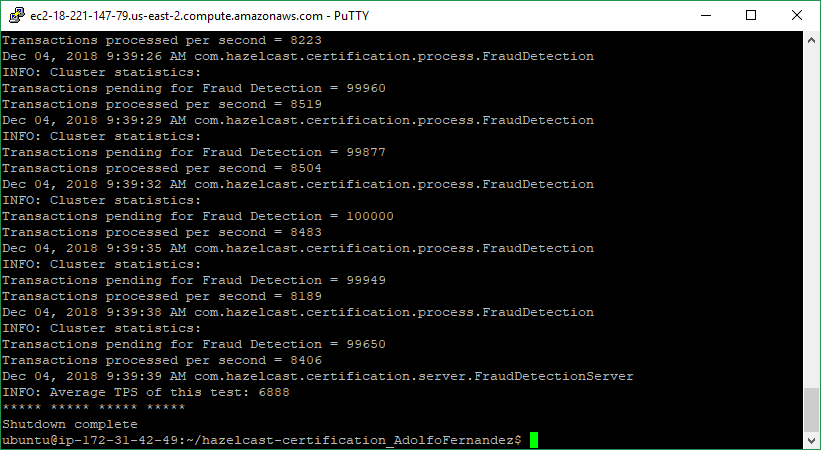
## Run the Fraud Detection Server

1. Review the ‘runServer.sh’ file and modify JVM memory parameters and the classpath according to your resources and your directory structure
2. From the solution’s root directory **execute ‘runServer.sh’**.

The process will start to process the transactions of the queue:



And it will be terminated when the process time expires.



# Appendix

## Appendix 1. Installing JDK 8 and Maven in an Ubuntu distribution

### Installing Oracle JDK 8

1. Install Oracle JDK 8

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer

1. Check installation paths

sudo update-alternatives --config java

1. Configure JAVA\_HOME and PATH by adding to ‘/etc/bash.bashrc’ file the following lines:

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

export PATH=$PATH:$JAVA\_HOME/bin

### Installing Maven

1. Copy apache-maven-3.5.3 en ‘/home/Ubuntu’ and move it to ‘/opt’

sudo mv /home/ubuntu/apache-maven-3.5.3 /opt

1. Grant permissions for the Maven executable

chmod 764 /opt/apache-maven-3.5.3/bin/mvn

1. Configure the environment variables by adding to ‘/etc/bash.bashrc’ file the following lines:

export MAVEN\_HOME=/opt/apache-maven-3.5.3

export M2\_HOME=/opt/apache-maven-3.5.3

export PATH=$PATH:$M2\_HOME/bin