# CSP554—Big Data Technologies

## Assignment #12 – Cassandra

## Readings

(Re)read Chapters 9 and 13 from our next book: Pramod J. Sadalage and Martin Fowler. 2012. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley.(PS)

## Worth: 14 points

## Due by the start of the next class period

Assignments should be uploaded via the Blackboard portal

Exercise 1) (4 points)

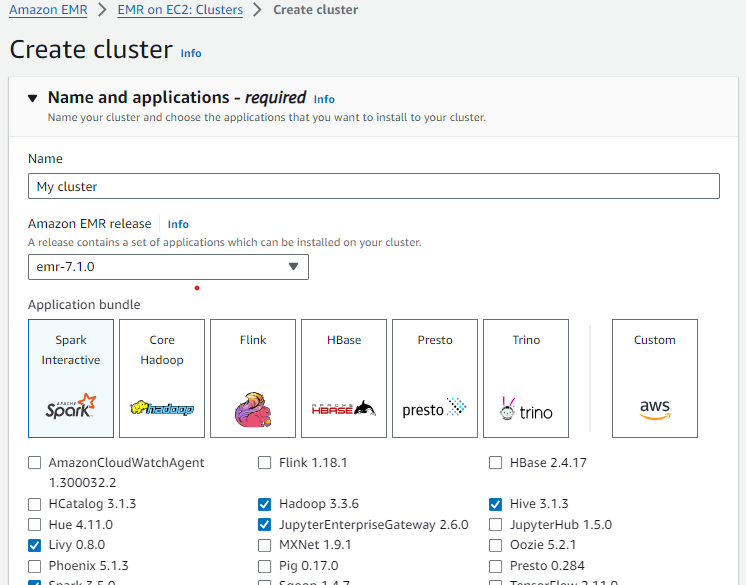
Read the article “A Big Data Modeling Methodology for Apache Cassandra” available on the blackboard in the ‘Articles’ section.

**Provide a ½ page summary including your comments and impressions.**

Exercise 2) (3 points)

Step A – Start an EMR cluster

Start up a EMR cluster as previously, but instead of choosing the “Core Hadoop” configuration chose the “Spark Interactive” configuration (see below), otherwise proceed as before.



Step B – Install the Cassandra database software and start it

Open up a terminal connection to your EMR primary node. Over the course of this exercise, you will need to open up three separate terminal connections to your EMR primary node. This is the first, which we will call Cass-Term.

Enter the following two commands:

wget https://archive.apache.org/dist/cassandra/4.1.4/apache-cassandra-4.1.4-bin.tar.gz

tar -xzvf apache-cassandra-4.1.4-bin.tar.gz

Note, this will create a new directory (apache-cassandra-4.1.4) holding the Cassandra software release.

Then enter this command to start Cassandra (lots of diagnostic messages will appear):

apache-cassandra-4.1.4/bin/cassandra &

Now wait two or three minutes for Cassandra to start.

Step C – Run the Cassandra interactive command line interface

Open a second terminal connection to the EMR primary node. Going forward we will call this terminal connection: Cqlsh-Term. Note, if you are using the git bash shell on your PC, open a new terminal window by right clicking on the title bar of the program and select ‘New Window’ or enter Alt+F2.

Wait a minute or two for Cassandra to finish starting up and then enter the following into this terminal to start the command line interface csqlsh:

apache-cassandra-4.1.4/bin/cqlsh

Step D – Prepare to edit your Cassandra code

Open a third terminal connection to the EMR primary node. Going forward we will call this terminal connection: Edit-Term.

You will use this terminal window to run the ‘vi’ editor to create your Cassandra code files. See the “Free Books and Chapters” section of our blackboard site for information on how to use the ‘vi’ editor.

As an alternative you could edit your Cassandra code files on your PC/MAC using a text editor like “notepad” or “textedit” "and then ‘scp’ them to the EMR mater node.

1. Create a file in your working (home) directory on the primary EMR node called **init.cql** using your Edit-term (or using your PC/MAC and then scp it to the EMR primary node) and enter the following command. Use your IIT id as the name of your keyspace… For example, if your id is A1234567, then replace <IIT id> below with that value:

CREATE KEYSPACE <IIT id> WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication\_factor' : 1 };

For example, you might write:

CREATE KEYSPACE A1234567 WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication\_factor' : 1 };

1. Then execute this file in the CQL shell using the Cqlsh-Term as follows…

source ‘./init.cql’;

1. To check if your script file has created a keyspace execute the following in the CQL shell:

describe keyspaces;

Note, Cassandra changes all unquoted strings such as A1234567 to lowercase (a1234567).

1. At this point you have created a keyspace unique to you. So, make that keyspace the default by entering the following into the CQL shell:

USE <IIT id>;

For example,

USE A1234567; (or equivalently: USE a1234567)

Now create a file in your working directory called **ex2.cql** using the Edit-Term (or PC/MAC and scp). In this file write the command to create a table named ‘Music’ with the following characteristics:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Primary Key / Cluster Key |
| artistName | text | Primary Key |
| albumName | text | Cluster Key |
| numberSold | int | Non Key Column |
| cost | int | Non Key Column |

Execute **ex2.cql** in the CQL shell. Then execute the shell command ‘**DESCRIBE TABLE Music**’.

**include (a) the content of the ex2.cql file and (b) a screenshot of the output generated when you then execute ‘DESCRIBE TABLE Music’** **as the result of this exercise.**

Exercise 3) (3 points)

Now create a file in your working directory called **ex3.cql** using the Edit-Term. In this file write the commands to insert the following records into table ‘Music’…

|  |  |  |  |
| --- | --- | --- | --- |
| artistName | albumName | numberSold | cost |
| Mozart | Greatest Hits | 100000 | 10 |
| Taylor Swift | Fearless | 2300000 | 15 |
| Black Sabbath | Paranoid | 534000 | 12 |
| Katy Perry | Prism | 800000 | 16 |
| Katy Perry | Teenage Dream | 750000 | 14 |

Execute **ex3.cql**.

**Provide (a) the content of the ex3.cql file as one result of this exercise, and (b) execute the command ‘SELECT \* FROM Music;’ and provide a screenshot of the output of this command as another result of the exercise.**

Exercise 4) (2 points)

Now create a file in your working directory called **ex4.cql** using the Edit-Term. In this file write the commands to query and output only Katy Perry songs. Execute **ex4.cql**.

**Provide (a) the content of the ex4.cql file and (b) a screenshot of the output of executing this file as the result of this exercise.**

Exercise 5) (2 points)

Now create a file in your working directory called **ex5.cql** using the Edit-Term. In this file write the commands to query only albums that have sold 700000 copies or more. Execute ex5.cql.

**Provide (a) the content of the ex5.cql file and (b) a screenshot of the output of executing this file as the result of this exercise.**

**Remember to terminate your EMR cluster when you complete this assignment**.