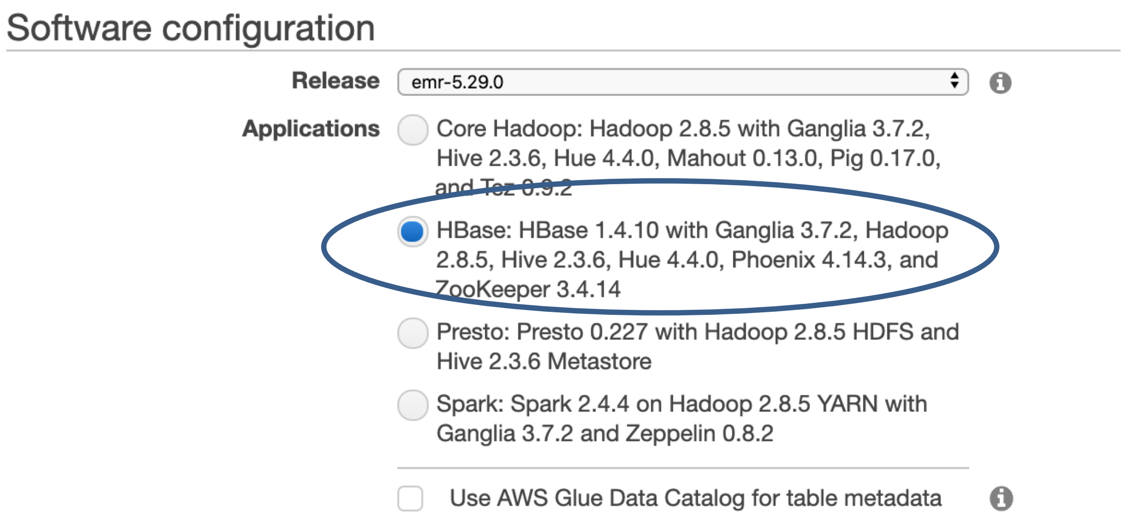
CSP554—Big Data Technologies

Assignment #9

Worth: 15 points (1 point for each problem)

HBase

Start up a Hadoop cluster as previously, but instead of choosing the "Core Hadoop" configuration chose the "HBase" configuration (see below), otherwise proceed as before:



Software configuration

Release: emr-5.29.0

Applications:
- Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.6, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- [✓ HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.6, Hue 4.4.0, Phoenix 4.14.3, and ZooKeeper 3.4.14]
- Presto: Presto 0.227 with Hadoop 2.8.5 HDFS and Hive 2.3.6 Metastore
- Spark: Spark 2.4.4 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.2

Use AWS Glue Data Catalog for table metadata

Log on to the master Hadoop EC2 VM as per previous assignments and enter 'hbase shell' to start the HBase shell. **Note: Startup error messages should be safe to ignore!**

Exercises

Exercise 1) (1 point)

Create an HBase table with the following characteristics

Table Name: csp554Tbl

First column family: cf1

Second column family: cf2

Then execute the DESCRIBE command on the table and return command you wrote and the output as the results of this exercise.

Exercise 2) (1 point)

Put the following data into the table created in exercise 1:

Row Key	Column Family	Column (Qualifier)	Value
Row1	cf1	name	Sam
Row2	cf1	name	Ahmed
Row1	cf2	job	Pilot
Row2	cf2	job	Doctor
Row1	cf2	level	LZ3
Row2	cf2	level	AR7

Execute the SCAN command on this table returning all rows, column families and columns. Provide the command and its result as the output of this exercise.

Exercise 3) (1 point)

Using the above table write a command that will get the value associated with row (Row1), column family (cf2) and column/qualifier (level). Provide the command and its result as the output of this exercise.

Exercise 4) (1 point)

Using the above table write command that will get the value associated with row (Row2), column family (cf1) and column/qualifier (name). Provide the command and its result as the output of this exercise.

Exercise 5) (1 point)

Using the above table write a SCAN command that will return information about only two rows using the LIMIT modifier. Provide the command and its result as the output of this exercise.

Cassandra

Exercises:

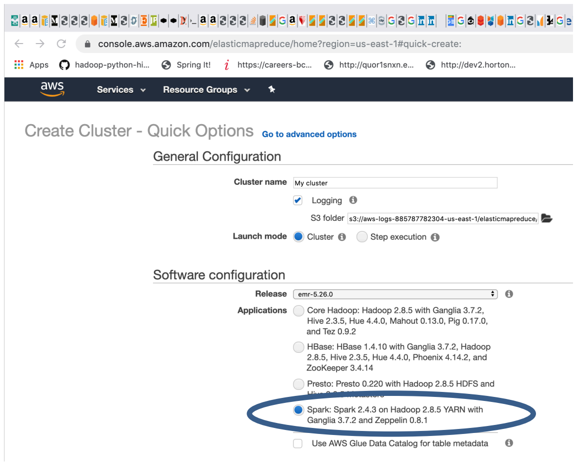
Exercise 1) (1 point)

Read the article “A Big Data Modeling Methodology for Apache Cassandra” and provide a ½ page summary including your comments and impressions.

Exercise 2) (1 point)

<u>Step A – Start an EMR cluster</u>

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


console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:

Create Cluster - Quick Options

General Configuration

Cluster name: My cluster

[✓] Logging

S3 folder: s3://aws-logs-885787782304-us-east-1/elasticmapreduce/

Launch mode: [✓ Cluster] Step execution

Software configuration

Release: emr-5.26.0

Applications:
- Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.5, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.5, Hue 4.4.0, Phoenix 4.14.2, and ZooKeeper 3.4.14
- Presto: Presto 0.220 with Hadoop 2.8.5 HDFS and Hive 2.3.5 Metastore
- [✓ Spark: Spark 2.4.3 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1]

[ ] Use AWS Glue Data Catalog for table metadata

<u>Step B – Install the Cassandra database software and start it</u>

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

Enter the following two commands:

```
wget https://archive.apache.org/dist/cassandra/3.11.2/apache-cassandra-3.11.2-bin.tar.gz
```

```
tar -xzvf apache-cassandra-3.11.2-bin.tar.gz
```

Note, this will create a new directory (apache-cassandra-3.11.2) holding the Cassandra software release.

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

```
apache-cassandra-3.11.2/bin/cassandra &
```

Step C – Run the Cassandra interactive command line interface

Open a second terminal connection to the EMR master node. Going forward we will call this terminal connection: Cqlsh-Term.

Enter the following into this terminal to start the command line interface csqlsh:

```
apache-cassandra-3.11.2/bin/cqlsh
```

Step D – Prepare to edit your Cassandra code

Open a third terminal connection to the EMR master 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.

a) Create a file in your working (home) directory called init.cql using your Edit-term (or using your PC/MAC and then scp it to the EMR master 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 };
```

b) Then execute this file in the CQL shell using the Cqlsh-Term as follows…

```
source './init.cql';
```

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

```
describe keyspaces;
```

d) 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;
```

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;' and include the output as the result of this exercise.

Exercise 3) (1 point)

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

a) Execute ex3.cql. Provide the content of this file as the result of this exercise.
b) Execute the command 'SELECT * FROM Music;' and provide the output of this command as another result of the exercise.

Exercise 4) (1 point)

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 the content of this file and output of executing this file as the result of this exercise.

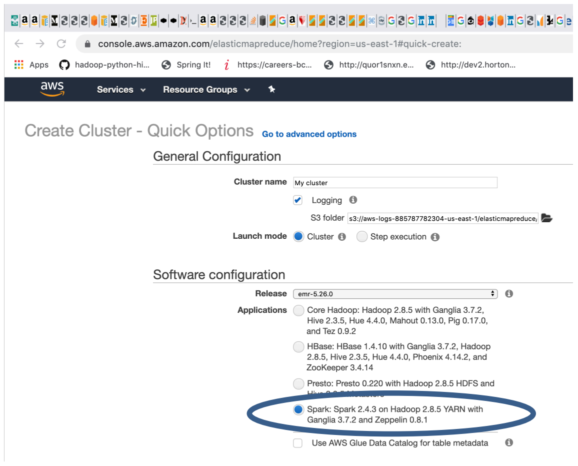
Exercise 5) (1 point)

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 the content of this file and the output of executing this file as the result of this exercise.

MongoDB

Step A – Start an EMR cluster

Start up an EMR/Hadoop cluster as previously, but instead of choosing the "Core Hadoop" configuration chose the "Spark" configuration (see below), otherwise proceed as before.


console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:

Create Cluster - Quick Options

General Configuration

Cluster name: My cluster

[✓] Logging

S3 folder: s3://aws-logs-885787782304-us-east-1/elasticmapreduce/

Launch mode: [✓ Cluster] Step execution

Software configuration

Release: emr-5.26.0

Applications:
- Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.5, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.5, Hue 4.4.0, Phoenix 4.14.2, and ZooKeeper 3.4.14
- Presto: Presto 0.220 with Hadoop 2.8.5 HDFS and Hive 2.3.5 Metastore
- [✓ Spark: Spark 2.4.3 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.1]

[ ] Use AWS Glue Data Catalog for table metadata

Step B – Download the assignment software (mongoex.tar, mongodb-org-4.2.repo) to master node

Download "mongoex.tar" (included as a file with the assignment) to your PC or MAC. Now, using "scp" copy this file to the EMR master node using something like the following (just an example):

```
scp -i ./emr-key-pair-2.cer /Users/nachdaph/csp554-fall-2021/assignments/mongoex.tar hadoop@ec2-44-199-215-205.compute-1.amazonaws.com:/home/hadoop
```

Now download "mongodb-org-4.2.repo" (included as a file with the assignment) to your PC or MAC. Now, using "scp" copy this file to the EMR master node using something like the following (just an example):

```
scp -i ./emr-key-pair-2.cer /Users/nachdaph/csp554-fall-2021/assignments/mongodb-org-4.2.repo hadoop@ec2-44-199-215-205.compute-1.amazonaws.com:/home/hadoop
```

Step C – Install assignment software (mongoex.zip, mongodb-org-4.2.repo)

Enter the following into a terminal window which you have connected to the EMR master node. Going forward we will call this terminal connection Init-Term:

```
sudo cp mongodb-org-4.2.repo /etc/yum.repos.d
```

Then enter this into Init-Term to unzip mongoex.tar:

```
tar -xvf mongoex.tar
```

Step D – Install and start MongoDB

Enter the following into Init-Term to install MongoDB:

```
sudo yum install -y mongodb-org-4.2.15 mongodb-org-server-4.2.15 mongodb-org-shell-4.2.15 mongodb-org-mongos-4.2.15 mongodb-org-tools-4.2.15
```

Now enter this into Init-Term to start mongodb:

```
sudo systemctl start mongod
```

Step E – Start the MongoDB Shell (Command Line Interpreter)

Open a second terminal connection to the EMR master node. Going forward we will call this terminal connection: CLI-Term.

You will use this terminal window to start and run the mongodb shell as follows:

```
mongo
```

Step F – Edit mongo query language files

Open a third terminal connection to the EMR master node. Going forward we will call this terminal connection: CLI-Term. You will use this terminal window to run the 'vi' editor to create your Mongo code files.

As an alternative you could edit your MongoDB code files on your PC/MAC and then 'scp' them to the EMR mater node.

<u>Step G – Setting up the assignment database</u>

Now, in the MongoDB shell, using the CLI-Term, create a database called "assignment" by entering the following into the MongoDB shell:

```
use assignment;
```

This will set the shell variable 'db' to this new database.

Load a collection called 'unicorns' with sample data by executing the script load.js in the MongoDB shell as follows (don't cut and paste this, type it in manually):

```
load('./load.js');
```

Note, look at the content of the script file (via the other terminal window you have opened to the EC2 instance) to see how each unicorn is described.

Confirm this has all worked by executing the following command in the MongoDB shell:

```
db.unicorns.find();
```

Note, the files named "demo*.js" (also included in the mongoex.tar file) provide examples of how to operate in the unicorn collection. These are a VERY good idea to review and understand and will present you with information helpful in completing the assignment. Also, try them out by typing something like

```
load(./demo1.js');
```

Exercises:

Exercise 1) (1 point)

Write a command that finds all unicorns having weight less than 500 pounds. Include the code you executed and some sample output as the result of this exercise. Recall you can place the command, if you choose, into a file, say ‘ex1.js’ and execute it with the load command as above and similarly for the following exercises.

Exercise 2) (1 point)

Write a command that finds all unicorns who love apples. Hint, search for “apple”. Include the code you executed and some sample output as the result of this exercise.

Exercise 3) (1 point)

Write a command that adds a unicorn with the following attributes to the collection. Note dob means “Date of Birth.”

Attribute	Value(s)
name	Malini
dob	11/03/2008
loves	pears, grapes
weight	450
gender	F
vampires	23
horns	1

Include the code you executed to insert this unicorn into the collection along with the output of a find command showing it is in the collection.

Exercise 4) (1 point)

Write a command that updates the above record to add apricots to the list of things Malini loves. Include the code you executed and some sample output showing the addition.

Exercise 5) (1 point)

Write a command that deletes all unicorns with weight more than 600 pounds. Include the code you executed and some sample output as the result of this exercise.