Kajol Tanesh Shah
A20496724
Fall 2022

CSP554—Big Data Technologies

Assignment #13

Worth: 5 points ALL EXTRA CREDIT

Due at the time you submit your final project or paper

Assignments should be uploaded via the Blackboard portal.

Set Up:

Step A – Start an EMR cluster

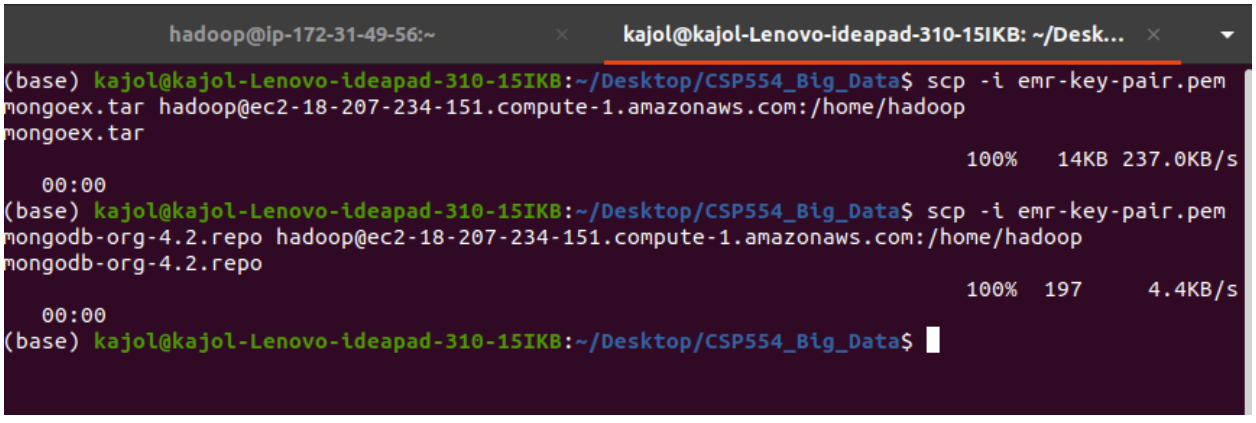
Start up an EMR/Hadoop cluster as previously, but instead of choosing the "Core Hadoop" configuration chose the "Spark" configuration (see below), otherwise proceed as before.

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.pem mongoex.tar
hadoop@ec2-18-207-234-151.compute-1.amazonaws.com:/home/hadoop

scp -i emr-key-pair.pem mongodb-org-4.2.repo
hadoop@ec2-18-207-234-151.compute-1.amazonaws.com:/home/hadoop



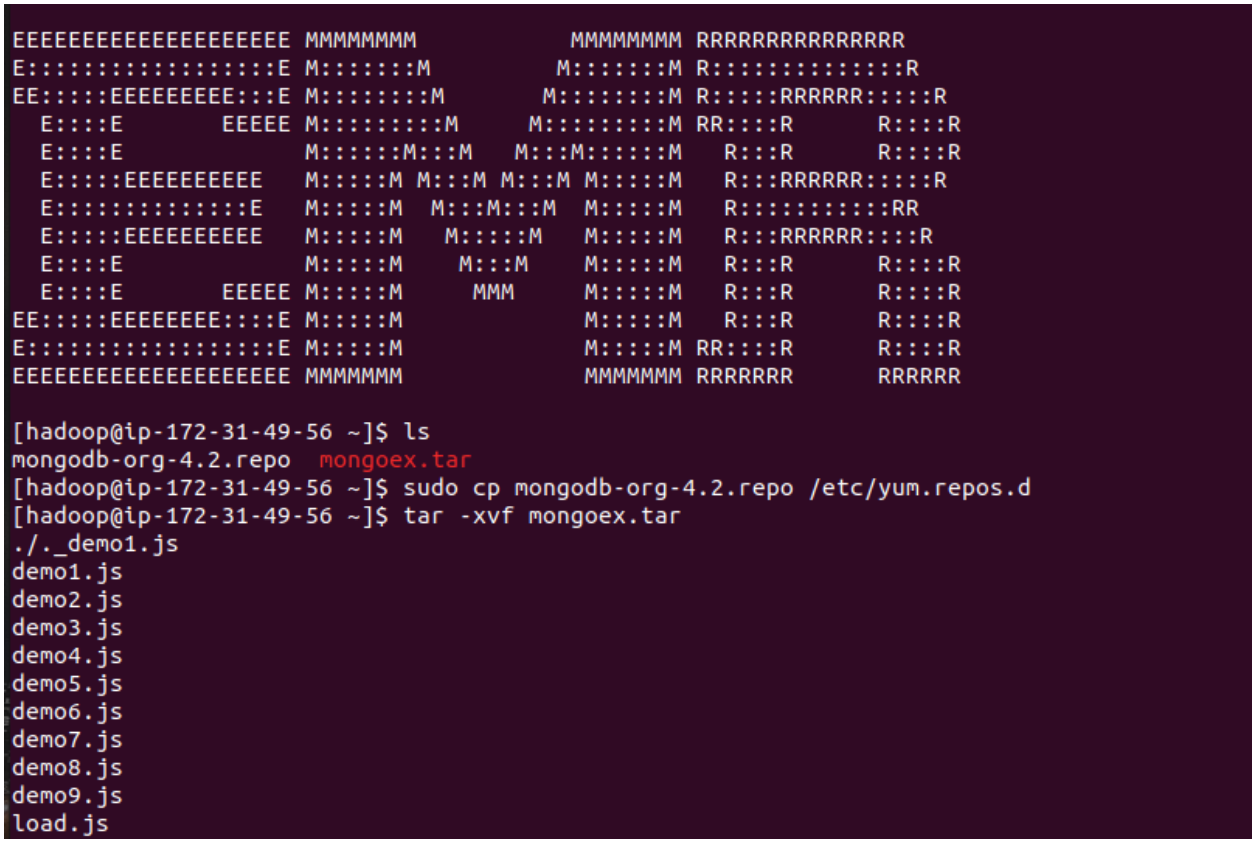
Step C – Install assignment software (mongoex.zip, mongodb-org-4.2.repo)

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

```
[hadoop@ip-172-31-49-56 ~]$ 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
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
mongodb-org-4.2
                                                                                | 2.6
kB  00:00:00
mongodb-org-4.2/primary_db
                                                                                | 107
kB  00:00:00
```

Now enter this into Init-Term to start mongodb:

sudo systemctl start mongod

```
[hadoop@ip-172-31-49-56 ~]$ sudo systemctl start mongod
[hadoop@ip-172-31-49-56 ~]$
```

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

```
EEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M::::::M         M::::::M R::::::::::::::R
EE::::::EEEEEEEE:::E M:::::::M       M:::::::M R::::::RRRRRR:::::R
  E:::::E       EEEEE M::::::::M     M::::::::M RR:::::R     R:::::R
  E:::::E             M::::::M:::M   M:::M::::::M   R::::R     R:::::R
  E::::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E    M:::::M  M:::M:::M  M:::::M   R::::::::::RR
  E::::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR:::R
  E:::::E             M:::::M    M:::M    M:::::M   R:::R      R:::R
  E:::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R:::R
EE::::::EEEEEEEE:::E M:::::M             M:::::M   R:::R      R:::R
E::::::::::::::::::E M:::::M             M:::::M RR:::::R     R:::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR     RRRRRR

[hadoop@ip-172-31-49-56 ~]$ mongo
MongoDB shell version v4.2.15
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("dd452991-50b8-43e9-8895-772acbb9682f") }
MongoDB server version: 4.2.15
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
Server has startup warnings:
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten]
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

>
```

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 master node.

Step G – Setting up the assignment database

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');

```
[hadoop@ip-172-31-49-56 ~]$ mongo
MongoDB shell version v4.2.15
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("b038dfea-6b02-4e1e-9eeb-e65cbb7944ac") }
MongoDB server version: 4.2.15
Server has startup warnings:
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten]
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-12-07T20:08:36.980+0000 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> use assignment;
switched to db assignment
> load('./load.js');
true
>
```

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.

cat load.js

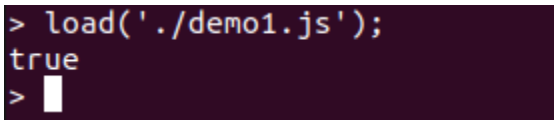
```
[hadoop@ip-172-31-49-56 ~]$ cat load.js
db.unicorns.insert({name: 'Horny',
        dob: new Date(1992,2,13,7,47),
        loves: ['carrot','papaya'],
        weight: 600,
        gender: 'm',
        vampires: 63});
db.unicorns.insert({name: 'Aurora',
        dob: new Date(1991, 0, 24, 13, 0),
        loves: ['carrot', 'grape'],
        weight: 450,
        gender: 'f',
        vampires: 43});
db.unicorns.insert({name: 'Unicrom',
        dob: new Date(1973, 1, 9, 22, 10),
        loves: ['energon', 'redbull'],
        weight: 984,
        gender: 'm',
        vampires: 182});
db.unicorns.insert({name: 'Roooooodles',
        dob: new Date(1979, 7, 18, 18, 44),
        loves: ['apple'],
        weight: 575,
        gender: 'm',
        vampires: 99});
db.unicorns.insert({name: 'Solnara',
        dob: new Date(1985, 6, 4, 2, 1),
        loves:['apple', 'carrot',
                'chocolate'],
        weight:550,
        gender:'f',
        vampires:80});
db.unicorns.insert({name:'Ayna',
        dob: new Date(1998, 2, 7, 8, 30),
        loves: ['strawberry', 'lemon'],
        weight: 733,
        gender: 'f',
        vampires: 40});
db.unicorns.insert({name:'Kenny',
        dob: new Date(1997, 6, 1, 10, 42),
        loves: ['grape', 'lemon'],
        weight: 690,
        gender: 'm',
        vampires: 39});
db.unicorns.insert({name: 'Raleigh',
        dob: new Date(2005, 4, 3, 0, 57),
        loves: ['apple', 'sugar'],
        weight: 421,
        gender: 'm',
        vampires: 2});
```

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

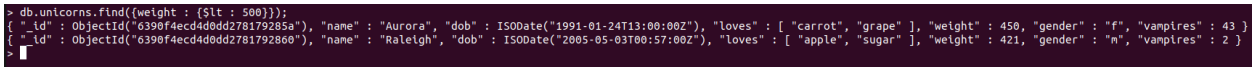
load(./demo1.js’);



Exercises:

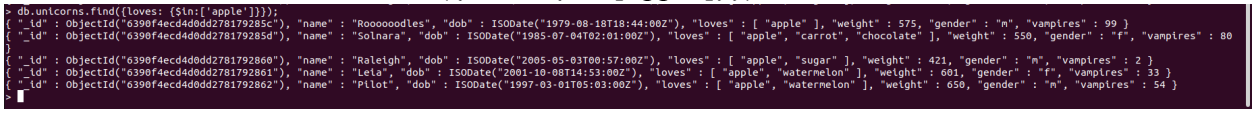
Exercise 1) (1 point)
Write a command that finds all unicorns having weight less than 500 pounds. Include the code you executed and some sample output as the result of this exercise. Recall you can place the command, if you choose, into a file, say ‘ex1.js’ and execute it with the load command as above and similarly for the following exercises.

Ans. Command:- db.unicorns.find({weight : {$lt : 500}});



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.

Ans. Command:- db.unicorns.find({loves: {$in:['apple']}});



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.

Ans. Command:- db.unicorns.insert({name: 'Malini', dob: new Date(2008, 11, 03), loves: ['pears', 'grapes'], weight: 450, gender: 'F', vampires: 23, horns : 1});

```
> db.unicorns.insert({name: 'Malini', dob: new Date(2008, 11, 03), loves: ['pears', 'grapes'], weight: 450, gender: 'F', vampires: 23, horns : 1});
WriteResult({ "nInserted" : 1 })
>
```

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.

Ans. Command:- db.unicorns.update({name: 'Malini'}, {$set : {loves: ['pears', 'grapes', 'apricots']}});

```
> db.unicorns.update({name: 'Malini'}, {$set : {loves: ['pears', 'grapes', 'apricots']}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

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.

Ans. Command:- db.unicorns.remove({weight: {$gt : 600}});

```
> db.unicorns.remove({weight: {$gt : 600}});
WriteResult({ "nRemoved" : 6 })
>
```