READY

READY

```
%sh
STATUS="$(service cassandra status)"

if [[ $STATUS == *"is running"* ]]; then
    echo "Cassandra is running"
else
    echo " Cassandra not running .... Starting"
    service cassandra restart > /dev/null 2>&1 &
    echo " Started"
fi
```

READY

Exercise 10 – Compaction

In this exercise, you will:

- Understand basic Apache Cassandra™ compaction strategies

As memtables fill up, Apache Cassandra™ writes them to disk in the form of SSTables. If this were the end of the story, the number of data files used to contain SSTables would become large and slow the Apache Cassandra™ read performance. Therefore, Apache Cassandra™ must consolidate these files from time to time. This consolidation is called compaction.

In this exercise, we observe the effects of compaction.

READY

Steps

1. Launch cqlsh. Let’s recreate the YooToob keyspace and also create the videos_by_tag database:

READY

```
%cassandra

CREATE KEYSPACE IF NOT EXISTS YooToob
WITH replication = {'class':'SimpleStrategy', 'replication_factor': 1};
```

10_Compaction

READY

```
%cassandra

USE YooToob;
```

```
CREATE TABLE videos_by_tag (
tag TEXT,
video_id UUID,
added_date TIMESTAMP,
title TEXT,
PRIMARY KEY ((tag), video_id)
);
```

```
%cassandra

// Now, let's insert a single row into the table:

INSERT INTO videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Master');
```

READY

3. Exit cqlsh and for Apache Cassandra™ to flush the memtable to an SSTable:

READY

```
nodetool flush
```

READY

4. Let’s investigate the SSTable in the node’s data directory. Remember the actual name of the directory will be a unique random value under `/var/lib/cassandra/data/` directory:

FINISHED

Took 0 sec. Last updated by anonymous at July 15 2020, 4:49:36 PM.

```
ls -l /var/lib/cassandra/data/yootoob/videos_by_tag-*
```

READY

You will see several files with names that start with. These are the files associated with the first SSTable.

5. We can create a second SSTable. Add a second row using the following:

READY

```
%cassandra

INSERT INTO YooToob.videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Genius');
```

READY

6. Flush this second memtable to disk:

READY

10_Compaction

```
nodetool flush
```

READY

7. Re-inspect the data directory to see the files associated with the *two* SSTables:

READY

```
ls -l /var/lib/cassandra/data/yootoob/videos_by_tag-*
```

READY

8. Create a third SSTable by inserting the following row:

READY

```
%cassandra

INSERT INTO YooToob.videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Wizard');
```

READY

9. Once again, flush the memtable to disk and investigate the data directory:

READY

Note: When Apache Cassandra™ goes to create a fourth SSTable, Apache Cassandra™ will perform compaction.

```
nodetool flush

ls -l /var/lib/cassandra/data/yootoob/videos_by_tag-*
```

READY

10. Insert a fourth row:

READY

```
%cassandra

INSERT INTO YooToob.videos_by_tag (tag, added_date, video_id, title)
VALUES ('cassandra', dateof(now()), uuid(), 'Cassandra Ninja');
```

READY

11. Flush and investigate the data directory again. Wait a few seconds before executing each command.

READY

```
nodetool flush
```

READY

10_Compaction

```
ls -l /var/lib/cassandra/data/yootoob/videos_by_tag-*
```

READY

READY

Notice that the three previous SSTable files are gone and a new set has appeared. Also, notice that the file names skipped from mc- 3 - big to mc- 5 - big. This new set is the result of the compaction. Apache Cassandra™ created the fourth SSTable and then compacted all four into a fifth SSTable.

READY

READY