Apache Cassandra:
Core Concepts, Skills, and Tools

Introducing hardware planning and troubleshooting
Exercise Workbook

Joe Chu
Leo Schuman
October, 2014

Exercise 1: Perform a one-node snapshot

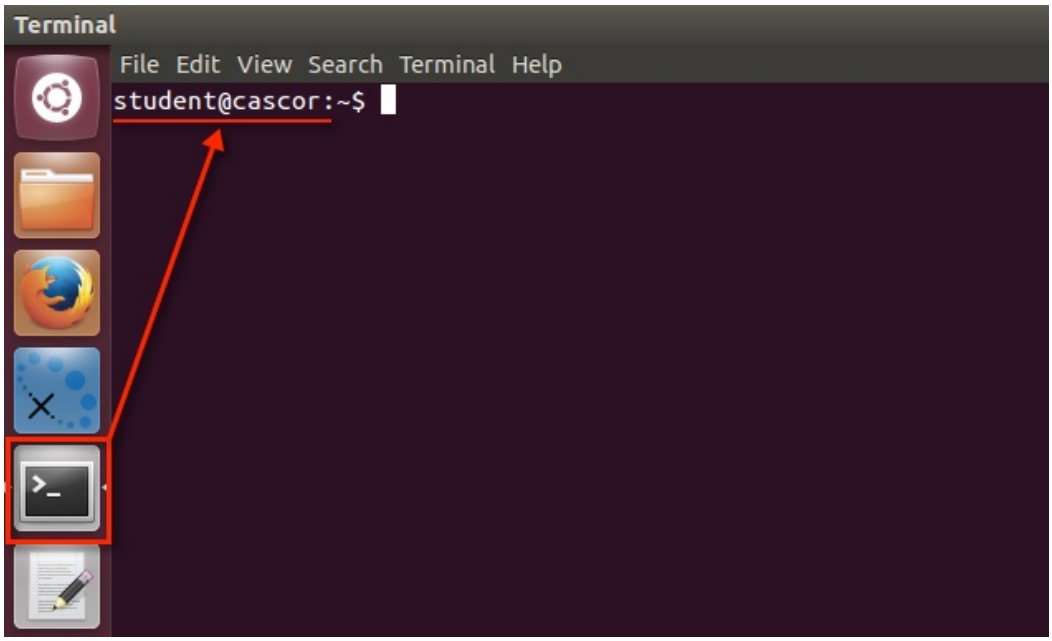
In this exercise, you will:

- Create a node snapshot
- Restore data from a snapshot

Steps

Create a node snapshot

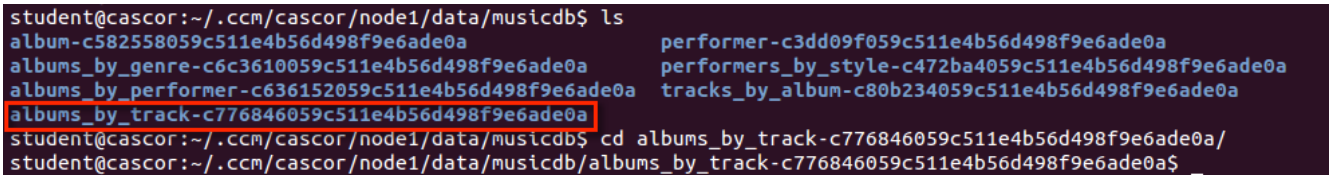
1. From the virtual machine, open a Terminal window or switch to an existing Terminal window.



2. Navigate to the data directory for the keyspace *musicdb* on *node1*.

```
cd ~/node1/data/musicdb
```

3. In the *musicdb* directory, list the table directories and navigate to the corresponding directory for *albums_by_track*.




```
ls
cd albums_by_track-[table id]
```

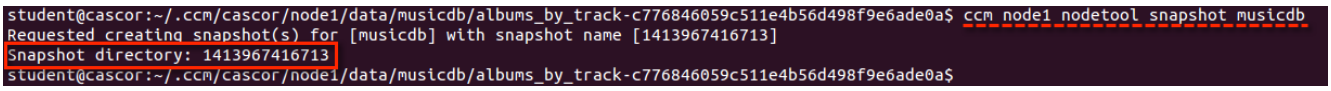
4. In the *albums_by_track* directory, list the files stored here. Make a note of the number of SSTables.

```
ls -l
```

If there are no files, you may need to run a nodetool flush first.

5. Use the *nodetool snapshot* command to create a snapshot of the *musicdb* keyspace for each node.

```
ccm node1 nodetool snapshot musicdb
```



6. In the *albums_by_track* directory, list the files and folders again. Note that there is now a *snapshots* directory.

```
ls -l
```

7. Go into the *snapshots* directory and list the contents. There will be at least one directory, which matches the name of the directory that *nodetool snapshot* generated.

```
cd snapshots
ls -l
```

```
student@cascor:~/.ccm/cascor/node1/data/musicdb/albums_by_track-c776846059c511e4b56d498f9e6ade0a/snapshots$ ls -l
total 4
drwxrwxr-x 2 student student 4096 Oct 22 01:43 1413967416713
```

There may be additional snapshots that are stored here, with different names.

8. Navigate into the newly created snapshot directory and list the files.

```
cd [snapshot directory]
ls -l
```

You will see the same SSTables from the albums_by_track *directory. nodetool snapshot also flushes MemTables before creating the snapshot, so there may also be new SSTable files.*

9. From the command line, start *cqlsh*.

```
ccm node1 cqlsh
```

10. In *cqlsh*, set the default keyspace to *musicdb*.

```
USE musicdb;
```

11. In *cqlsh*, use the DESCRIBE TABLE command for *albums_by_track* table and write down or copy the table schema.

```
DESCRIBE TABLE albums_by_track
```

It is important to that you also have a backup of your table schemas, since a snapshot can only be restored to an existing table.

Restore data from a snapshot

12. In *cqlsh*, run a query to count the number of rows in the *albums_by_track* table. Make a note of the number of rows retrieved.

```
SELECT count(*) FROM albums_by_track LIMIT 50000;
```

13. Drop the *albums_by_track* table.

```
DROP TABLE albums_by_track;
```

14. Re-create the *albums_by_track* table with the table schema that you saved.

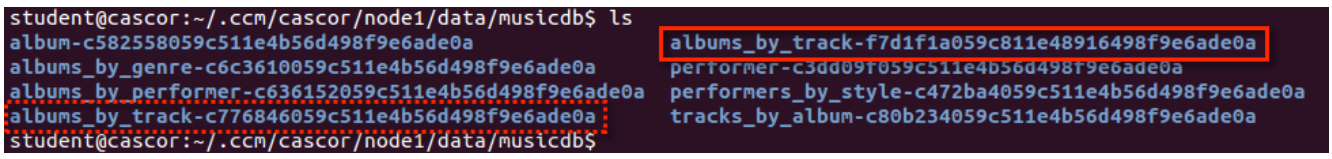
```
CREATE TABLE musicdb.albums_by_track (
  track_title text,
  performer text,
  year int,
  album_title text,
  PRIMARY KEY (track_title, performer, year, album_title)
) WITH CLUSTERING ORDER BY (performer ASC, year DESC,
album_title ASC);
```

15. Exit *cqlsh* to return to the command line.

```
EXIT
```

16. From the command line, go to the *musicdb* directory for node1 and find the new *albums_by_track* directory. Make a note of the directory name.

```
cd ~/node1/data/musicdb
ls
```



You should find two albums_by_track *directories, the directory for the original table that was dropped, and the directory for the newly created table.*

17. Navigate to the old *albums_by_track* directory and proceed to the *snapshots* directory. List the files and folders for this directory

```
cd albums_by_track-[old table id]
cd snapshots
ls -l
```

Here you will see a new snapshot directory. Cassandra will generate a snapshot when mass-deleting data for a table, such as with the DROP TABLE and TRUNCATE commands. This behavior can be changed with the auto_snapshot *setting in the* cassandra.yaml *file.*

18. For only node1, copy the files saved from the previously created snapshot to the new *albums_by_track* directory that was created.

```
cp ~/node1/data/musicdb/albums_by_track-[old table
id]/snapshots/[snapshot directory]/*
~/node1/data/musicdb/albums_by_track-[new table id]
```

19. Run the command *nodetool refresh* for *musicdb.albums_by_track* so that Cassandra reloads the SSTables that were copied over.

```
ccm node1 nodetool refresh musicdb albums_by_track
```

20. Start *cqlsh* again.

```
ccm node1 cqlsh
```

21. In *cqlsh*, run a query to count the number of rows in the *albums_by_track* table using a consistency level of ALL.

```
CONSISTENCY ALL
SELECT count(*) FROM musicdb.albums_by_track LIMIT 50000;
```

END OF EXERCISE