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 9 – Read Path

FINISHED

In this exercise, you will:

- Understand the Apache Cassandra read path.

Took 0 sec. Last updated by anonymous at July 14 2020, 9:37:49 PM.

Steps

READY

1. To begin, we need to populate our single-node cluster with a good chunk of data. We will use cassandra-stress to do so. Execute the following command in a terminal window. Wait for cassandra-stress to complete before continuing.

```
cassandra-stress write no-warmup n=75000 -port native=9042 -rate threads=1
```

READY

2. Now force Apache Cassandra™ to flush its current memtable to disk by executing the following command:

READY

```
nodetool flush
```

READY

3. Navigate to the data directory for the large table that cassandra-stress wrote. Remember that cassandra-stress creates a keyspace called keyspace1 and a table called standard1. You will find the directory as illustrated

READY

09_Read Path

below, however, your table ID will differ.

```
ls -ltr /var/lib/cassandra/data/keyspace1/
```

READY

```
# Use the following command to list the bloom filter files for your SSTables:

ls -ltr /var/lib/cassandra/data/keyspace1/standard1-*/*Filter.db
```

READY

Take note of the file sizes.

READY

We will now decrease the probability that a bloom filter will return a false positive and see how this affects the bloom filter files sizes.

5. Execute the following command to view the current bloom filter settings:

```
%cassandra
DESCRIBE keyspace keyspace1;
```

READY

Note the bloom_filter_fp_chance is 0.01, meaning a 1% chance of a false positive. That's pretty good, but if we want an even smaller chance, we can trade space for it.

READY

6. Execute the following command:

```
%cassandra
ALTER TABLE keyspace1.standard1 WITH bloom_filter_fp_chance = 0.0001;
```

READY

Now that we have changed the bloom_filter_fp_chance, we must update our SSTables and associated bloom filter files.

READY

7. Switch back to your terminal window and execute the following command:

```
nodetool upgradesstables --include-all-sstables
```

READY

nodetool upgradesstables rebuilds SSTables for a specified keyspace and table. We are doing this here to rebuild the Bloom Filters also. Normally this command will only upgrade sstables if the sstables are not at the most recent SSTable version; the –

READY

09_Read Path

include-all-sstables flag forces the rebuild to occur. Normally you would need to run nodetool upgradesstables on each node. For the purposes of this exercise, we only have one node.

8. Now examine the new size of your bloom filter files. READY

Notice the size of these files is larger. We traded space for a smaller chance of a false positive.

```
ls -ltr /var/lib/cassandra/data/keyspace1/standard1-*/*Filter.db
```
READY

9. Now execute the following command: READY

```
%cassandra
ALTER TABLE keyspace1.standard1 WITH bloom_filter_fp_chance = 1.0;
```
READY

10. And update your SSTables once again in your terminal: READY

```
nodetool upgradesstables --include-all-sstables
```
READY

Now what is the size of your bloom filter files? Why? :) FINISHED

The files are gone! A bloom_filter_fp_chance = 1.0 means that no filtering is going on, so there is no need to store the filters.

Took 3 sec. Last updated by anonymous at July 14 2020, 9:36:16 PM. (outdated)

```
# No file exists now
ls -ltr /var/lib/cassandra/data/keyspace1/standard1-*/*Filter.db
```
READY

11. Now execute the following command in your terminal: FINISHED

Took 0 sec. Last updated by anonymous at July 14 2020, 9:36:29 PM.

```
nodetool truncate keyspace1.standard1
```
READY

09_Read Path

Part of the stats include bloom filter information. Since we have not read from the cassandra-stress tables, the values are all zero. However, you can use these stats to tune Apache Cassandra™ if necessary.

READY

```
nodetool cfstats keyspace1.standard1
```

READY

READY

READY