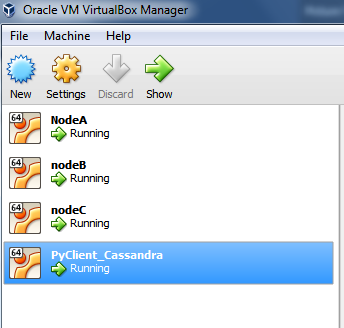
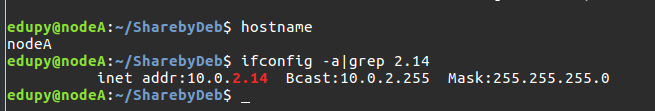
4 VMs created. All running on Ubuntu16 + Java + Python.

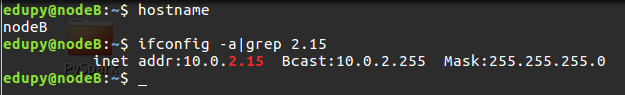
nodeA, nodeB and nodeC are installed with Cassandra and configured with same Cassandra cluster.

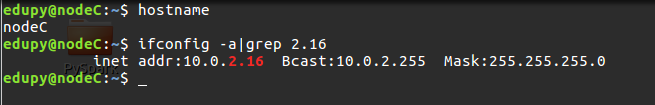
The VM ‘PyClient\_Cassandra’ is only installed with Python Cassandra driver.



IP Address of the VMS below:







Following configurations done for hostname resolution of neighbors:

nodeA ip: 10.0.2.14

/etc/hostname:: nodeA

/etc/hosts:  10.0.2.15 nodeB 10.0.2.16 nodeC

nodeB ip: 10.0.2.15

/etc/hostname:: nodeB

/etc/hosts:  10.0.2.14 nodeA 10.0.2.16 nodeC

nodeC ip: 10.0.2.16

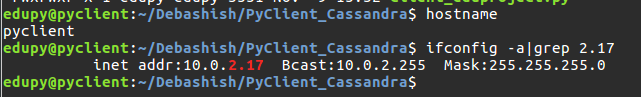
/etc/hostname:: nodeC

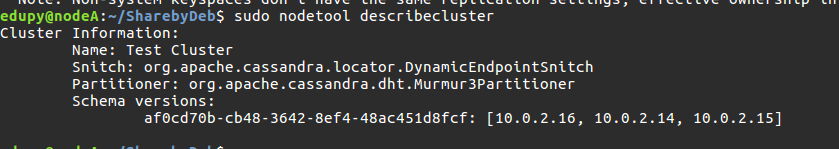
/etc/hosts:  10.0.2.14 nodeA 10.0.2.15 nodeB

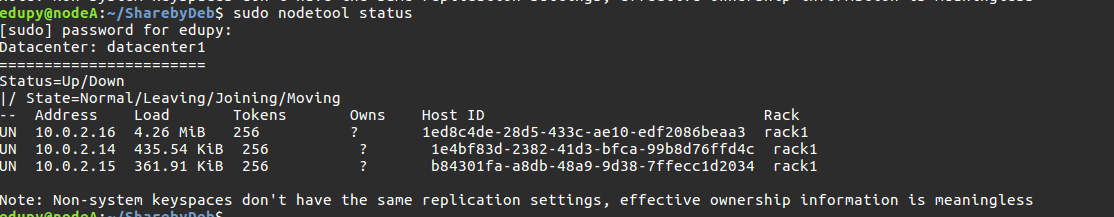
**Cassandra client:**

A VM with IP 10.0.2.17.

Cassandra is NOT installed. Its installed with Python2.7 and Python Cassandra-driver.







cqlsh:playlist> DESCRIBE KEYSPACEs;

playlist system\_auth debkey1 system\_distributed

system\_schema system debkey3 system\_traces

cqlsh:playlist> DESCRIBE TABLEs;

statistics users track\_by\_id track\_by\_genre

track\_by\_artist artists\_by\_first\_letter playlist\_tracks

cqlsh:playlist> DESCRIBE playlist;

CREATE KEYSPACE playlist WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': '2'} AND durable\_writes = true;

CREATE TABLE playlist.statistics (

counter\_name text PRIMARY KEY,

counter\_value counter

) WITH bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

CREATE TABLE playlist.track\_by\_artist (

artist text,

track text,

track\_id uuid,

genre text,

music\_file text,

starred boolean,

track\_length\_in\_seconds int,

PRIMARY KEY (artist, track, track\_id)

) WITH CLUSTERING ORDER BY (track ASC, track\_id ASC)

AND bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

CREATE TABLE playlist.users (

username text PRIMARY KEY,

password text,

playlist\_names set<text>

) WITH bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

CREATE TABLE playlist.artists\_by\_first\_letter (

first\_letter text,

artist text,

PRIMARY KEY (first\_letter, artist)

) WITH CLUSTERING ORDER BY (artist ASC)

AND bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

CREATE TABLE playlist.track\_by\_id (

track\_id uuid PRIMARY KEY,

artist text,

genre text,

music\_file text,

track text,

track\_length\_in\_seconds int

) WITH bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

CREATE TABLE playlist.playlist\_tracks (

username text,

playlist\_name text,

sequence\_no timestamp,

artist text,

genre text,

track\_id uuid,

track\_length\_in\_seconds int,

track\_name text,

PRIMARY KEY ((username, playlist\_name), sequence\_no)

) WITH CLUSTERING ORDER BY (sequence\_no ASC)

AND bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

CREATE TABLE playlist.track\_by\_genre (

genre text,

artist text,

track text,

track\_id uuid,

music\_file text,

starred boolean,

track\_length\_in\_seconds int,

PRIMARY KEY (genre, artist, track, track\_id)

) WITH CLUSTERING ORDER BY (artist ASC, track ASC, track\_id ASC)

AND bloom\_filter\_fp\_chance = 0.01

AND caching = {'keys': 'ALL', 'rows\_per\_partition': 'NONE'}

AND comment = ''

AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max\_threshold': '32', 'min\_threshold': '4'}

AND compression = {'chunk\_length\_in\_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}

AND crc\_check\_chance = 1.0

AND dclocal\_read\_repair\_chance = 0.1

AND default\_time\_to\_live = 0

AND gc\_grace\_seconds = 864000

AND max\_index\_interval = 2048

AND memtable\_flush\_period\_in\_ms = 0

AND min\_index\_interval = 128

AND read\_repair\_chance = 0.0

AND speculative\_retry = '99PERCENTILE';

cqlsh:playlist>

From Python client VM:

1. Using python based Cassandra driver:
   1. Connect to the keyspace ‘playlist’
   2. Fetch and display all the records from a table
   3. Fetch and display filtered records from a table
   4. Insert multiple records into the table and verify it by fetching the records.
   5. Use ConsistencyLevel ‘Quorum’ and ‘Local’
   6. Use load\_balancing\_policy= DCAwareRoundRobinPolicy
   7. Use load\_balancing\_policy= RoundRobinPolicy
   8. Use SimpleStatement

Output of Cassandra client script:

