Cassandra Installation into HDP sandbox:

Python 2.7 version required, if not execute the below commands:

su root

yum install scl-utils

yum install centos-release-scl-rh

yum install python27

scl enable python27 bash

* **su root**: Switch to the root user.
* **yum install scl-utils**: Install the Software Collections (SCL) utilities.
* **yum install centos-release-scl-rh**: Install the CentOS Software Collections release package for Red Hat.
* **yum install python27**: Install Python 2.7.
* **scl enable python27 bash**: Enable Python 2.7 in the current shell session.

cd /etc/yum.repos.d

vi datastax.repo

* **cd /etc/yum.repos.d**: Change the directory to /etc/yum.repos.d.
* **vi datastax.repo**: Open the vi editor to create/edit the datastax.repo file.

Inside the vi editor, add the following content to datastax.repo:

[datastax]

name = DataStax Repo for Apache Cassandra

baseurl = http://rpm.datastax.com/community

enabled = 1

gpgcheck = 0

This configuration sets up the DataStax repository for Apache Cassandra.

yum install dsc30

service cassandra start

* **yum install dsc30**: Install DataStax Community version 3.0 of Apache Cassandra.
* **service cassandra start**: Start the Cassandra service.
* **yum install dsc30**: Install DataStax Community version 3.0 of Apache Cassandra.
* **service cassandra start**: Start the Cassandra service.

Cqlsh

* **cqlsh**: Launch the Cassandra Query Language Shell to interact with Cassandra.

CREATE KEYSPACE database WITH replication = {'class': 'SimpleStrategy', 'replication\_factor':'1'} AND durable\_writes = true;

USE moviesdata;

CREATE TABLE users (user\_id int, age int, gender text, occupation text, zip text, PRIMARY KEY (user\_id));

DESCRIBE TABLE users;

* **CREATE KEYSPACE**: Create a keyspace named 'moviesdata' with SimpleStrategy replication and a replication factor of 1.
* **USE moviesdata**: Switch to the 'moviesdata' keyspace.
* **CREATE TABLE users**: Create a table named 'users' with specified columns and primary key.
* **DESCRIBE TABLE users**: Display information about the structure of the 'users' table.

These commands collectively set up Python 2.7, install Apache Cassandra, configure its repository, start the Cassandra service, and then create a keyspace and a table using the Cassandra Query Language (CQL).

After installation is complete:

Type:

Cqlsh

**Create a Keyspace:**

CREATE KEYSPACE IF NOT EXISTS keyspace\_name

WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': 1};

This command creates a keyspace named keyspace\_name with SimpleStrategy replication and a replication factor of 3. Adjust the replication strategy and factor based on your requirements.

### Use a Keyspace:

USE keyspace\_name;

This command switches to the specified keyspace (keyspace\_name) for subsequent operations.

### Create Table:

CREATE TABLE IF NOT EXISTS table\_name (

column1 data\_type1,

column2 data\_type2,

PRIMARY KEY (primary\_key\_column)

);

### Insert Data:

INSERT INTO table\_name (column1, column2)

VALUES (value1, value2);

This command inserts data into the specified table (table\_name). Provide values for the columns based on the table structure.

### Select Data:

SELECT \* FROM table\_name WHERE condition;

This command retrieves data from the specified table (**table\_name**). You can include a WHERE clause to filter results based on a condition.

### Update Table:

UPDATE table\_name

SET column1 = new\_value1, column2 = new\_value2

WHERE condition;

This command updates data in the specified table (table\_name). Adjust the SET clause and WHERE clause based on your update criteria.

### Delete Data:

DELETE FROM table\_name WHERE condition;

This command deletes data from the specified table (table\_name). Include a WHERE clause to specify the deletion criteria.

### Describe Table:

DESCRIBE TABLE table\_name;

This command provides information about the structure of the specified table (table\_name), including column names, data types, and the primary key.

### Drop Keyspace:

DROP KEYSPACE IF EXISTS keyspace\_name;

This command deletes the specified keyspace (keyspace\_name) and all the tables it contains.

### Drop Table:

DROP TABLE IF EXISTS table\_name;

this command deletes the specified table (**table\_name**). Be cautious as this operation is irreversible.

These are basic examples, and there are more advanced commands and features available in Cassandra. Always refer to the [official documentation](http://cassandra.apache.org/doc/latest/) for detailed and up-to-date information.