CASSANDRA - DROP KEYSPACE

Dropping a Keyspace using Cqlsh

You can drop a KeySpace using the command **DROP KEYSPACE**. Given below is the syntax for dropping a KeySpace.

Syntax

```
DROP KEYSPACE <identifier>
```

i.e.

```
DROP KEYSPACE “KeySpace name”
```

Example

The following code deletes the keyspace **tutorialspoint**.

```
cqlsh> DROP KEYSPACE tutorialspoint;
```

Verification

Verify the keyspaces using the command **Describe** and check whether the table is dropped as shown below.

```
cqlsh> DESCRIBE keyspaces;

system system_traces
```

Since we have deleted the keyspace tutorialspoint, you will not find it in the keyspaces list.

Dropping a Keyspace using Java API

You can create a keyspace using the execute method of Session class. Follow the steps given below to drop a keyspace using Java API.

Step1: Create a Cluster Object

First of all, create an instance of **Cluster.builder** class of **com.datastax.driver.core** package as shown below.

```
//Creating Cluster.Builder object
Cluster.Builder builder1 = Cluster.builder();
```

Add a contact point *IPaddressofthenode* using the **addContactPoint** method of **Cluster.Builder** object. This method returns **Cluster.Builder**.

```
//Adding contact point to the Cluster.Builder object
Cluster.Builder builder2 = build.addContactPoint( "127.0.0.1" );
```

Using the new builder object, create a cluster object. To do so, you have a method called **build** in the **Cluster.Builder** class. The following code shows how to create a cluster object.

```
//Building a cluster
Cluster cluster = builder.build();
```

You can build a cluster object using a single line of code as shown below.

```
Cluster cluster = Cluster.builder().addContactPoint("127.0.0.1").build();
```

Step 2: Create a Session Object

Create an instance of Session object using the connect method of Cluster class as shown below.

```
Session session = cluster.connect( );
```

This method creates a new session and initializes it. If you already have a keyspace, you can set it to the existing one by passing the keyspace name in string format to this method as shown below.

```
Session session = cluster.connect(“ Your keyspace name”);
```

Step 3: Execute Query

You can execute CQL queries using the execute method of Session class. Pass the query either in string format or as a Statement class object to the execute method. Whatever you pass to this method in string format will be executed on the cqlsh.

In the following example, we are deleting a keyspace named **tp**. You have to store the query in a string variable and pass it to the execute method as shown below.

```
String query = "DROP KEYSPACE tp; ";

session.execute(query);
```

Given below is the complete program to create and use a keyspace in Cassandra using Java API.

```
import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.Session;

public class Drop_KeySpace {

   public static void main(String args[]){

      //Query
      String query = "Drop KEYSPACE tp";

      //creating Cluster object
      Cluster cluster = Cluster.builder().addContactPoint("127.0.0.1").build();

      //Creating Session object
      Session session = cluster.connect();

      //Executing the query
      session.execute(query);
      System.out.println("Keyspace deleted");
   }
}
```

Save the above program with the class name followed by .java, browse to the location where it is saved. Compile and execute the program as shown below.

```
$javac Delete_KeySpace.java
$java Delete_KeySpace
```

Under normal conditions, it should produce the following output:

```
Keyspace deleted
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js