

Task2- Data Ingestion From AWS RDS Using Scoop

Ingest data from an AWS RDS instance into Hadoop Using Scoop

- Apache Sqoop, a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

Solution Approach

- Install Sqoop: Ensure that Sqoop is installed in Hadoop cluster during EMR creation
- MySQL JDBC Driver: Since RDS instance is MySQL, the MySQL JDBC driver is required in Sqoop's lib directory.
- Sqoop Import Command: Use the sqoop import command to transfer data from RDS instance to Hadoop

Prerequisite for performing scoop job

- The EMR cluster should be created (with scoop selected in application bundle)
- Use the AWS cloud shell to connect with the EMR cluster with the key pair created for the user
- EMR service role should have necessary permissions attached to perform action in the EMR
- (Optional if not using AWS cloud shell) Security and network rules should be defined to allow ssh from local machine

Command for getting JDBC driver for scoop

First get the MYSQL connector for scoop with the below mentioned command after ssh into EMR cluster

```
wget -q "http://search.maven.org/remotecontent?filepath=mysql/mysql-connector-java/5.1.32/mysql-connector-java-5.1.32.jar" -O mysql-connector-java.jar
```

Copy the downloaded mysql connector to the scoop local directory with the below mentioned command

```
sudo cp mysql-connector-java.jar /usr/lib/sqoop/lib/
```

Validate whether the copied mysql connector is present in the scoop lib folder with the below command

```
ls /usr/lib/sqoop/lib/
```

Apache Sqoop import Command

Before running scoop job run it as Hadoop admin in order to avoid write permission issues use the below command to become Hadoop admin after performing an ssh into emr cluster

```
sudo su - hdfs
```

Then run the scoop import command mentioned below

```
sqoop import \  
--connect jdbc:mysql://upgraddetest.cyaie1c9bmnf.us-east-1.rds.amazonaws.com/testdatabase \  
--username student \  
--password STUDENT123 \  
--table bookings \  
--target-dir /user/ec2-user/Data/Booking_Batch_Data \  
--m 1
```

Apache Sqoop import Command Explanation

- `--connect`: JDBC connection string, which is used to connect to the desired database.
- `--username` & `--password`: Credentials for RDS instance.
- `--table`: The name of the table to import.
- `--target-dir`: The Hadoop HDFS directory to store the imported data
- `--m 1` option sets the number of mappers to 1

Validate whether the data is present from the scoop job

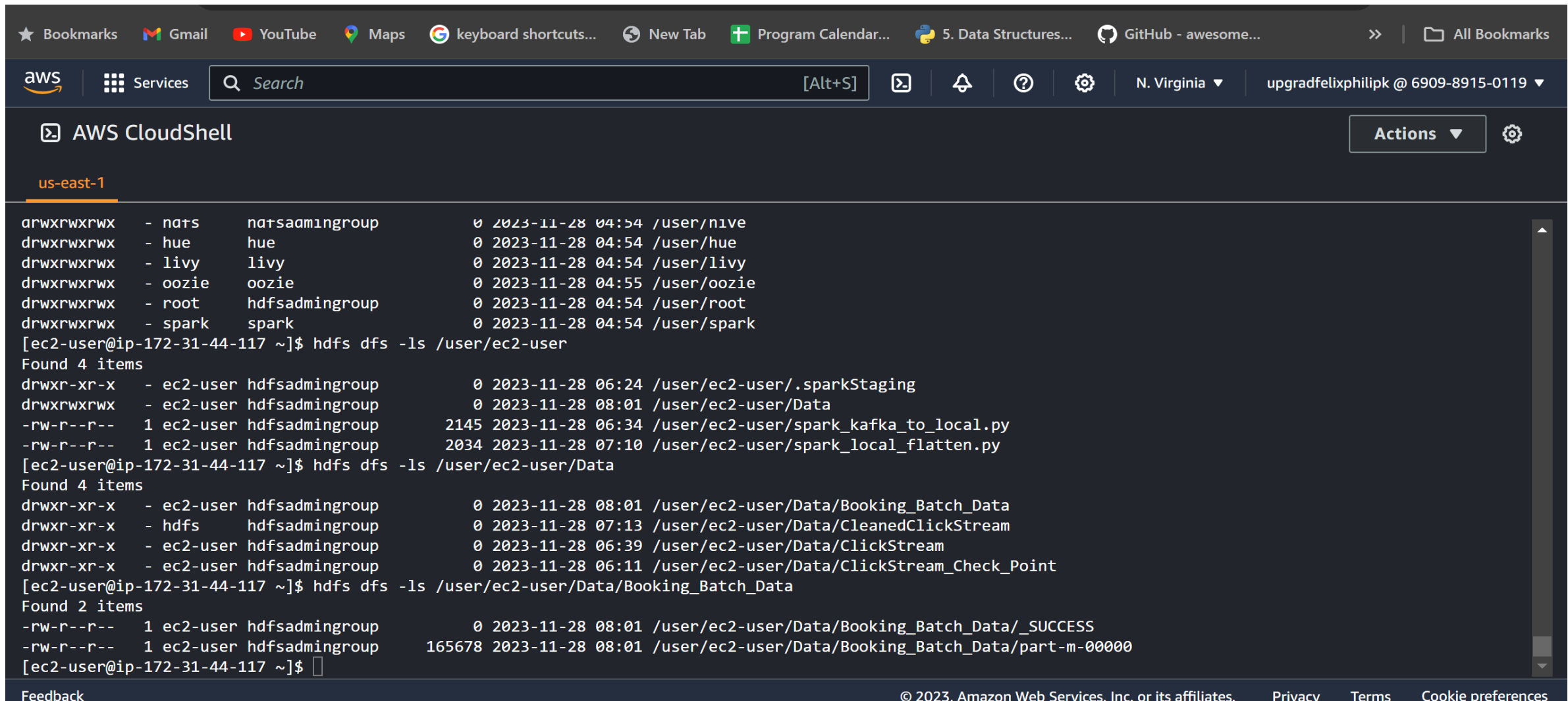
- In the EMR cluster navigate to the hdfs directory to which the scoop job was written

```
hdfs dfs -ls /directory/path
```

- To view the data use the below command

```
hdfs dfs -cat /path/to/directory
```

Validate whether the data is present from the scoop job with -ls



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal displays the output of several `hdfs dfs -ls` commands. The first command lists files in `/user/ec2-user`, showing directories for `nats`, `hue`, `livy`, `oozie`, `root`, and `spark`. The second command lists files in `/user/ec2-user`, showing directories for `.sparkStaging`, `Data`, `spark_kafka_to_local.py`, and `spark_local_flatten.py`. The third command lists files in `/user/ec2-user/Data`, showing directories for `Booking_Batch_Data`, `CleanedClickStream`, `ClickStream`, and `ClickStream_Check_Point`. The fourth command lists files in `/user/ec2-user/Data/Booking_Batch_Data`, showing directories for `_SUCCESS` and `part-m-00000`.

```
aws | Services | Search [Alt+S] | N. Virginia | upgradfelfilipk @ 6909-8915-0119 | Actions | Settings
```

us-east-1

```
drwxrwxrwx - nats natsaamingroup 0 2023-11-28 04:54 /user/nive
drwxrwxrwx - hue hue 0 2023-11-28 04:54 /user/hue
drwxrwxrwx - livy livy 0 2023-11-28 04:54 /user/livy
drwxrwxrwx - oozie oozie 0 2023-11-28 04:55 /user/oozie
drwxrwxrwx - root hdfsadmingroup 0 2023-11-28 04:54 /user/root
drwxrwxrwx - spark spark 0 2023-11-28 04:54 /user/spark
[ec2-user@ip-172-31-44-117 ~]$ hdfs dfs -ls /user/ec2-user
Found 4 items
drwxr-xr-x - ec2-user hdfsadmingroup 0 2023-11-28 06:24 /user/ec2-user/.sparkStaging
drwxrwxrwx - ec2-user hdfsadmingroup 0 2023-11-28 08:01 /user/ec2-user/Data
-rw-r--r-- 1 ec2-user hdfsadmingroup 2145 2023-11-28 06:34 /user/ec2-user/spark_kafka_to_local.py
-rw-r--r-- 1 ec2-user hdfsadmingroup 2034 2023-11-28 07:10 /user/ec2-user/spark_local_flatten.py
[ec2-user@ip-172-31-44-117 ~]$ hdfs dfs -ls /user/ec2-user/Data
Found 4 items
drwxr-xr-x - ec2-user hdfsadmingroup 0 2023-11-28 08:01 /user/ec2-user/Data/Booking_Batch_Data
drwxr-xr-x - hdfs hdfsadmingroup 0 2023-11-28 07:13 /user/ec2-user/Data/CleanedClickStream
drwxr-xr-x - ec2-user hdfsadmingroup 0 2023-11-28 06:39 /user/ec2-user/Data/ClickStream
drwxr-xr-x - ec2-user hdfsadmingroup 0 2023-11-28 06:11 /user/ec2-user/Data/ClickStream_Check_Point
[ec2-user@ip-172-31-44-117 ~]$ hdfs dfs -ls /user/ec2-user/Data/Booking_Batch_Data
Found 2 items
-rw-r--r-- 1 ec2-user hdfsadmingroup 0 2023-11-28 08:01 /user/ec2-user/Data/Booking_Batch_Data/_SUCCESS
-rw-r--r-- 1 ec2-user hdfsadmingroup 165678 2023-11-28 08:01 /user/ec2-user/Data/Booking_Batch_Data/part-m-00000
[ec2-user@ip-172-31-44-117 ~]$
```

Feedback | © 2023, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences