School of Science, Computing and Emerging Technologies

COS80023 Big Data

Pass Task 3: Extraction, Transformation and Loading

Overview

Practise ETL with an example that starts from a .csv (comma-separated table) file and use a Hadoop cluster and Apache Hive to transform it by applying the relational model and storing it in a relational database with the help of Apache Sqoop.

Purpose

Demonstrate an understanding how to use Hadoop tools to automate the transformation of data into the relational model.

Task

Carry out the tasks described below and answer the questions in your submission.

Time

This task should be completed in the fourth lab class or before and submitted to Canvas for feedback. It should be discussed and signed off in tutorial 3 or 4.

This task should take no more than 2 hours to complete.

Resources

- Presentation (from Blackboard)
- How to create an HDInsight cluster (not all settings you need are the same in this document): https://docs.microsoft.com/fi-fi/azure/hdinsight/hadoop/apache-hadoop-linux-create-cluster-get-started-portal
- Tutorial on Hadoop and Apache Hive: https://docs.microsoft.com/fi-fi/azure/hdinsight/hadoop/apache-hadoop-linux-create-cluster-get-started-portal
- Any other online material
- genAI – Allowed for research. Must be able to explain

Feedback

Discuss your answers with the tutorial instructor.

Next

Get started on module 4.

Pass Task 3 — Submission Details and Assessment Criteria

Write down the answers in a text or Word document, convert to pdf and upload to Canvas. Your tutor will mark the submission on line. If the submission is not marked as '1', it is considered as incomplete and must be resubmitted.

SWINBURNE UNIVERSITY OF TECHNOLOGY

Task 3.1

Overview

Create an Azure Storage Account, upload a data file in tabular format, clean it and transform it using Hive on a Hadoop cluster on Azure.

The process has the following steps:

1. Creating a Hadoop cluster, using HDInsight and a Blob Storage account for it
2. Uploading the data and staging script
3. Extracting information from the data using Hive

Important!

1. Always create all your services in the same region and resource group. If you don't the services won't be able to work together.
2. Deployment of an HDInsight cluster takes 20 – 30 minutes. It is worth doing this straight up.

1. Create an HDInsight Cluster

In the search field top center of the page, type HDInsight. Choose HDInsight clusters from the options.

Click on 'Create'.

Basic tab

In the 'Basic' tab, choose your subscription and resource group, then set the cluster name to s<yourstudentnumber>cluster (no upper case letters allowed), e.g. s12345678cluster.

Choose Australia Southeast as location.

Choose Hadoop 3.1.0 as cluster type. Leave the default cluster username and ssh user. Choose a password with upper case, lower case, numbers and a special character.

(Suggestion: Use the same password throughout this exercise. Put it into a text file and copy when needed. Do not use your normal Swinburne SIMS password).

Click Next to proceed to Storage.

Storage

Choose Azure Storage. Keep Selection method at 'Select from list'.

At Primary storage account, click 'Create new' and name the new storage <yourstudentnumber>storage.

Name the container <yourstudentnumber>container.

Click Next.

Cluster size (Configuration+Pricing)

Under Configuration + pricing, leave the default nodes and cores, but check if any of them have 'Not available' written next to them on the drop-down list. Choose resources that are available. If no resources are available, change to Australia East

No need to add script action.

Click Review+create.

Summary

On the summary page, you get to create the cluster after the settings have been validated. It can take quite some time for the cluster to deploy.

2. Examine the Hive Script

While you are waiting, open the Hive script staging.hql using Notepad++. Try to make sense of what these two HQL statements are going to achieve.

After running the staging.hql script, you will run this interactive query:

```
INSERT OVERWRITE DIRECTORY '/accidents/output'
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
SELECT regexp_replace(day_week_description, '''', ''),
    sum(no_of_vehicles)
FROM accidents_in_hive
WHERE no_of_vehicles IS NOT NULL
GROUP BY day_week_description;
```

Try to understand what we are about to do.

3. Upload Script to Storage

When the HDInsight deployment is complete, the <yourstudentnumber>storage also becomes available. We can now upload the script we will later use for data transformation. Ensure you have downloaded the resource staging.hql from Canvas and can find it on your local filesystem.

Click on your storage resource. Find Storage Browser among the options on your navigation pane on the left. Open Blob containers. You should now see the <yourstudentnumber>container you created earlier. Click on it to navigate into the directory, where you will find a number of Hadoop-related files and directories.

Click on +Add Directory. Name it accidents and click Ok.

Click on +Add Directory again. Name it script and click Ok. The script directory was created as a subdirectory of accidents. Upload the staging.hql script into this directory.

!Note that if you leave the directory empty and navigate away from it, it gets automatically deleted!

Create another directory 'data' under accidents.

Click on Upload. Ensure the file gets uploaded into the script directory.

4. Accessing with HDInsight

Open a Command window (type cmd in the search bar). Ensure you have administrator rights when doing this.

Connect to the cluster using ssh (secure socket shell).

```
ssh sshuser@s<yourstudentnumber>cluster-ssh.azurehdinsight.net
```

Hint: If you go to your cluster in the Azure Portal and click on SSH + Cluster login in the navigation pane, you will be able to copy the logon string.

If your connect string and password were correct, you now see a command prompt:

```
sshuser@hn0-s12345:$
```

This means you are talking to the head node (hn0) of your cluster. You can now issue commands to the cluster.

Confusingly, there are now two filesystems you have access to; Hadoop's HDFS and the local file system, which is the user home. If you type

```
pwd
```

and press <Enter>, you will see that the current path is /home/sshuser (unless you have changed the user name for the ssh user when you created the HDInsight cluster).

```
ls -l
```

Shows that the directory is empty. Where are the csv and hql files you put into the blob storage? You need HDFS for this:

```
hdfs dfs -ls /
```

This shows what is in your HDFS root directory. You'll find the content of <yourstudentnumber>container. To see whether the hql and csv files are really there, you can type

```
hdfs dfs -ls /accidents/script
```

You should see staging.hql and be able to repeat the check for the data directory to find the .csv file. (If you do not, you have to troubleshoot before continuing.)

While Hive can use HDFS, it cannot run a script from HDFS, so we have to copy staging.hql into the current directory.

```
hdfs dfs -get /accidents/script/staging.hql .
```

In case you wondered, the dot means 'here' (the current directory). Run

ls or ls -l

again to check if this has succeeded.

5. Transform the Data

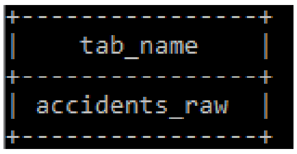
To run the staging.hql script, we have to call Hive:

```
beeline -u 'jdbc:hive2://localhost:10001/;transportMode=http' -f
staging.hql
```

After the staging.hql script has finished running, check if it was successful by looking for the tables we expected it to create.

```
beeline -u 'jdbc:hive2://localhost:10001/;transportMode=http' -e "SHOW
TABLES LIKE 'accidents_raw';"
beeline -u 'jdbc:hive2://localhost:10001/;transportMode=http' -e "SHOW
TABLES LIKE 'accidents_in_hive';"
```

If you do not see the table name under tab_name, the script has failed and it's time to troubleshoot.

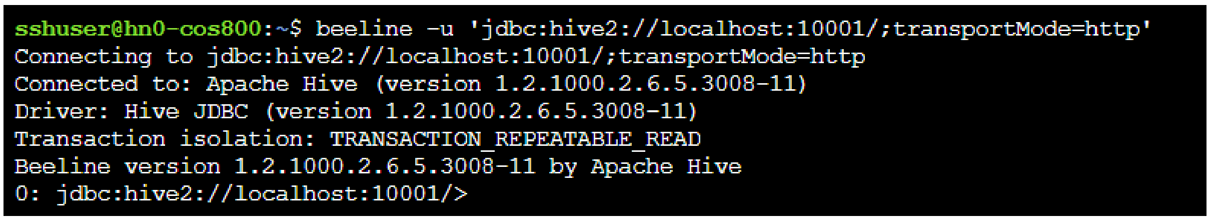


```
+----------------+
|    tab_name    |
+----------------+
| accidents_raw  |
+----------------+
```

If all is good, use the following command to open an interactive Hive session:

```
beeline -u 'jdbc:hive2://localhost:10001/;transportMode=http'
```

You should see the following prompt:



```
sshuser@hn0-cos800:~$ beeline -u 'jdbc:hive2://localhost:10001/;transportMode=http'
Connecting to jdbc:hive2://localhost:10001/;transportMode=http
Connected to: Apache Hive (version 1.2.1000.2.6.5.3008-11)
Driver: Hive JDBC (version 1.2.1000.2.6.5.3008-11)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 1.2.1000.2.6.5.3008-11 by Apache Hive
0: jdbc:hive2://localhost:10001/>
```

Now you can extract data using:

```
INSERT OVERWRITE DIRECTORY '/accidents/output'
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
SELECT regexp_replace(day_week_description, '''', ''),
    sum(no_of_vehicles)
FROM accidents_in_hive
WHERE no_of_vehicles IS NOT NULL
GROUP BY day_week_description;
```

To close the Hive interactive session, type !exit.

If you refresh your storage blob browser in the <yourstudentnumber>container, you will also observe some changes. Try to work out how they came about.

On the command line, you already know how to browse the content of a directory. Use the

```
hdfs dfs -ls /accidents/output
```

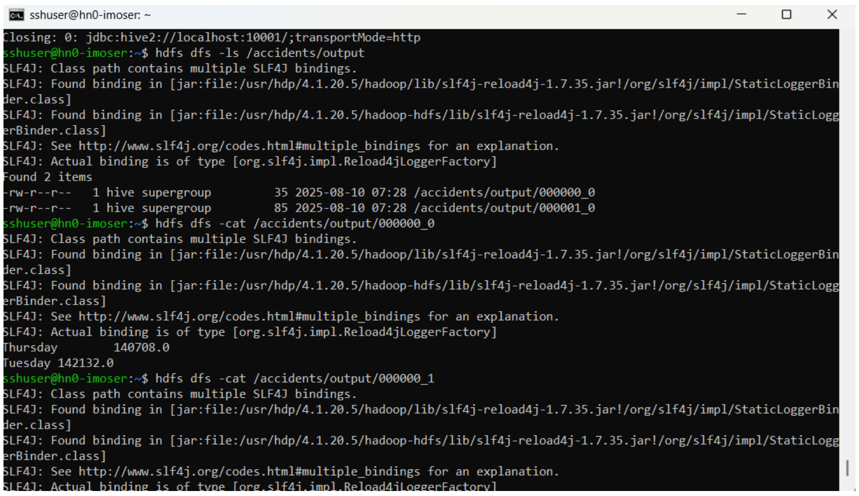
command you used before to inspect the content of the `output` directory. You should find a single or multiple files there. Use this command for each file you found:

```
hdfs dfs -cat /accidents/output/<file_name_you_just_found>
```

to display the content of the file. Is this what you expected to see when you examined the HiveQL you ran interactively?

6. Document and Submit the Task

1. Explain in your own words what you got Hive to do. You could do a diagram if you prefer.
2. Make a screenshot of your complete command window with the content of your output directory, something like in the picture:



sshuser@hn0-imoser: ~

```
Closing: 0: jdbc:hive2://localhost:10001/;transportMode=http
sshuser@hn0-imoser:~$ hdfs dfs -ls /accidents/output
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/4.1.20.5/hadoop/lib/slf4j-reload4j-1.7.35.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/usr/hdp/4.1.20.5/hadoop-hdfs/lib/slf4j-reload4j-1.7.35.jar!/org/slf4j/impl/StaticLogg
erBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
Found 2 items
-rw-r--r--   1 hive supergroup         35 2025-08-10 07:28 /accidents/output/000000_0
-rw-r--r--   1 hive supergroup         85 2025-08-10 07:28 /accidents/output/000001_0
sshuser@hn0-imoser:~$ hdfs dfs -cat /accidents/output/000000_0
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/4.1.20.5/hadoop/lib/slf4j-reload4j-1.7.35.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/usr/hdp/4.1.20.5/hadoop-hdfs/lib/slf4j-reload4j-1.7.35.jar!/org/slf4j/impl/StaticLogg
erBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
Thursday        140708.0
Tuesday 142132.0
sshuser@hn0-imoser:~$ hdfs dfs -cat /accidents/output/000000_1
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/4.1.20.5/hadoop/lib/slf4j-reload4j-1.7.35.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: Found binding in [jar:file:/usr/hdp/4.1.20.5/hadoop-hdfs/lib/slf4j-reload4j-1.7.35.jar!/org/slf4j/impl/StaticLogg
erBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
```

Upload the document to Canvas.

Important: You **MUST** delete your database server, datalake and cluster at the end of this exercise. If you do not, it will keep running and use resources.