BA-BEAD Big Data Engineering for Analytics

Workshop Instructions

Apache Sqoop

In Academic Partnership with:

cloudera®

ACADEMIC PARTNER

NUS National University of Singapore | ISS INSTITUTE OF SYSTEMS SCIENCE

Contents

Prerequisite

Basic familiarity with the Linux command line is assumed. Basic knowledge of SQL is helpful; prior knowledge of Hadoop is not required.

Running the VMware Image

Start the virtual Image and you should be able to work with the CentOS virtual machine.

Set Up Your Environment

Before starting the assignment, be sure you have run the cloudera launch script from the desktop window. (You only need to run this script once; if you ran it earlier, you do not need to run it again.)

Working with the Virtual Machine

1. The VM is set to automatically log in as the user `cloudera`. Should you log out at any time, you can log back in as the user `cloudera` with the password training.
2. Should you need it, the root password is `cloudera`. You may be prompted for this if, for example, you want to change the keyboard layout. In general, you should not need this password since the `cloudera` user has unlimited sudo privileges.

Homework: Import Data from MySQL using the Sqoop Tool

In this exercise, you will import tables from MySQL into HDFS with Sqoop. We will assume you have run the start-up services script necessary for this workshop.

Files and Data Used in this Homework

MySQL Database: retail_db

MySQL Tables: products, orders, etc

Import the accounts table from MySQL

Book mark support page: https://www.cloudera.com/developers/get-started-with-hadoop-tutorial/exercise-1.html

You can use Sqoop to look at the table layout in MySQL. With Sqoop, you can also import the table from MySQL to HDFS.

1. Open a new terminal window (if necessary).
2. Run the **`sqoop help`** command to familiarize yourself with the options in Sqoop:

```
$ sqoop help
```

3. List the tables in the **`retail_db`** database:

```
$ sqoop list-tables \
--connect jdbc:mysql://localhost/retail_db \
--username root --password cloudera
```

4. Run the `sqoop import` command to see its options:

```
$ sqoop import --help
```

5. Use Sqoop to import the **`products`** table in the **`retail_db`** database and save it in HDFS under **`/retail_db`**:

```
$ sqoop import \
  --connect jdbc:mysql://localhost/retail_db \
  --username root --password cloudera \
  --table accounts \
  --target-dir /retail_db/products \
  --null-non-string '\\N'
```

   The **`--null-non-string`** option tells Sqoop to represent **`null`** values as **`\N`**, which makes the imported data compatible with Hive and Impala.

6. *Optional*: While the Sqoop job is running, try viewing it in the Hue Job Browser or YARN Web UI, as you did in the previous exercise.

View the imported data

Sqoop imports the contents of the specified tables to HDFS. You can use the **hdfs** command line or the Hue File Browser to view the files and their contents.

7. List the contents of the **`products`** directory:

```
$ hdfs dfs -ls /retail_db/products
```

   - **Note**: Output of Hadoop processing jobs is saved as one or more numbered "partition" files.

8. Use either the Hue File Browser or the HDFS **`tail`** command to view the last part of the file for each of the MapReduce partition files, e.g.:

```
$ hdfs dfs -tail /retail_db/products/***
```

Import all tables and use as a warehouse

9. Apache Sqoop, which is part of CDH, is that tool. The nice thing about Sqoop is that we can automatically load our relational data from MySQL into HDFS, while preserving the structure. With a few additional configuration parameters, we can take this one step further and load this relational data directly into a form ready to be queried by Apache Impala, the MPP analytic database included with CDH, and other workloads.

   You should first log in to the Master Node of your cluster via a terminal. Then, launch the Sqoop job:

```
> sqoop import-all-tables \
    -m {{cluster_data.worker_node_hostname.length}} \
    --connect
jdbc:mysql://{{cluster_data.manager_node_hostname}}:3306/retail_db \
    --username=retail_dba \
    --password=cloudera \
    --compression-codec=snappy \
    --as-parquetfile \
    --warehouse-dir=/user/hive/warehouse \
    --hive-import
```

   This command may take a while to complete, but it is doing a lot. It is launching MapReduce jobs to pull the data from our MySQL database and write the data to HDFS in parallel, distributed across the cluster in Apache Parquet format. It is also creating tables to represent the HDFS files in Impala/Apache Hive with matching schema.

   Parquet is a format designed for analytical applications on Hadoop. Instead of grouping your data into rows like typical data formats, it groups your data into columns. This is ideal for many analytical queries where instead of retrieving data from specific records, you're analyzing relationships between specific variables across many records. Parquet is designed to optimize data storage and retrieval in these scenarios.

Verification

10. When this command is complete, confirm that your data files exist in HDFS.

```
> hadoop fs -ls /user/hive/warehouse/
> hadoop fs -ls /user/hive/warehouse/categories/
```

These commands will show the directories and the files inside them that make up your tables.

Import webpage data using an alternate field delimiter

11. We also want to import another orders table to HDFS. But first look at a few records in that table using the **`sqoop eval`** command.

```
$ sqoop eval \
  --query "SELECT * FROM orders LIMIT 10" \
  --connect jdbc:mysql://localhost/retail_db \
  --username root --password cloudera
```

Notice that the values in the last column contain commas. By default, **sqoop** uses commas as field separators, but because the data itself uses commas, we can't do that this time.

Summary

Thus this homework was to understand and use the various features of the *Data Ingestion Tool* Apache Sqoop to import and export data from the conventional RDBMS systems.

Switching off the Virtual Image.

When you have finished working with the image you can properly close the machine by choosing the Shut Down option under the System Menu of the Cent OS Linux Virtual Image.

You have completed a challenging workshop. Sit back and have a good break!

---------END OF DOCUMENT----------