page2image568

Big Data Hadoop and Spark Developer

Lesson-End Project Solution



**Retail Business Analysis Using Structured Streaming**

**Steps to Perform:**

**Step 1:** Log in to your LMS account

**Step 2:** Open the course “**Big Data Hadoop and Spark Developer**”

**Step 3:** Download the folder **“data”** consisting of JSON files from the “**Course Resources**” section

**Step 4:** On the left side, click on the “**PRACTICE LABS**” tab and click on the “**LAUNCH LAB**” button

Graphical user interface, application, Teams

Description automatically generated

**Step 5:** Again, click on the “**LAUNCH LAB**” button

Graphical user interface, application

Description automatically generated

**Step 6:** Click on **HUE** to upload the folder

Graphical user interface, application, website

Description automatically generated

**Step 7**: Log in to the **HUE** and click on create a directory named **“data-files”** and upload the folder **“data”** consisting of JSON files

**Step 8:** Click on the Webconsole and click on the “**Auth Url**”

Graphical user interface, application, website

Description automatically generated

**Step 9:** Copy the “**Username**” and the “**Password**” provided to log in to the Web console

**Step 10:** Paste the “**Username**” and the “**Password**” on the console and click on Enter

**Note:** The password will not be visible when pasted on the console.

Text

Description automatically generated

**Step 11:** Log in to the PySpark shell

**Command:**

pyspark3

Text

Description automatically generated

**Step 12:** Import the required packages and read the **“data”** folder consisting of JSON data in the Webconsole by providing the HDFS path as shown below:

from pyspark.sql.types import TimestampType, StringType, StructType, StructField

inputPath = "/user/testdemomay1301mailinator/data-files/data/"

**Step 13:** Define a schema containing four fields

schema = StructType() \

.add("time", "string") \

.add("customer", "string") \

.add("action", "string") \

.add("device", "string")

**Step 14:** Read the input data providing the schema

streamingDF = spark.readStream.schema(schema).option("maxFilesPerTrigger",1).json(inputPath)

**Step 15:** Write the **streamingDF** created in Step 13 to memory in append output mode streamingDF.writeStream.format("memory").queryName("counts").outputMode("append").start()

Text

Description automatically generated