

Big Data on AWS v3.0: Lab 6 - Processing NY Taxi data using Spark on Amazon EMR

==================================================================================================================

Using this command reference file.

==================================================================================================================

1. Locate the section you need. Each section in this file matches a section in the lab instructions.

2. Replace items in angle brackets - < > - with appropriate values. For example, in this command you would replace the value - <JobFlowID> - (including the angle brackets) with the parameter indicated in the lab instructions:

elastic-mapreduce --list <JobFlowID>. You can also use find and replace to change bracketed parameters in bulk.

3. Do NOT enable the Word Wrap feature in Windows Notepad or the text editor you use to view this file.

==================================================================================================================

Task 1: Task 1: Upload a Python Script to Your Amazon S3 Bucket

==================================================================================================================

+++ Overview

http://us-west-2-aws-training.s3.amazonaws.com/awsu-ilt/AWS-200-BIG/v3.0/lab-6-spark/scripts/pyspark-lab6.py

==================================================================================================================

Task 2.1: Adding Spark Application Step

==================================================================================================================

+++ Step 2.1.4 Application Location

s3://spark-lab6-<yourinitials>/scripts/pyspark-lab6.py

+++ Step 2.1.4 Arguments

s3://spark-lab6-<yourinitials>/output

==================================================================================================================

Task 4.2: Logging Into Spark Shell And Importing JAR Libraries

==================================================================================================================

+++ Step 4.2.2

aws s3 cp s3://us-west-2-aws-training/awsu-ilt/AWS-200-BIG/v3.0/lab-6-spark/scripts/jar/ . --recursive

+++ Step 4.2.3

spark-shell --jars minimal-json-0.9.5-SNAPSHOT.jar,/usr/share/aws/redshift/jdbc/RedshiftJDBC.jar,spark-avro\_2.11-3.0.1.jar,spark-redshift\_2.11-2.0.1.jar

==================================================================================================================

Task 4.3: Creating AWS Credentials For Spark-Redshift Connection

==================================================================================================================

+++ Step 4.3.1

import org.apache.spark.sql.\_

import com.amazonaws.auth.\_

+++ Step 4.3.2

val sqlContext = new SQLContext(sc)

+++ Step 4.3.3

val provider = new InstanceProfileCredentialsProvider()

val credentials: AWSSessionCredentials = provider.getCredentials.asInstanceOf[AWSSessionCredentials]

+++ Step 4.3.4

val token = credentials.getSessionToken

val awsAccessKey = credentials.getAWSAccessKeyId

val awsSecretKey = credentials.getAWSSecretKey

==================================================================================================================

Task 4.4: Task 4.4: Exploring Processed Data Using Spark-Shell

==================================================================================================================

++ Step 4.4.1

val s3\_location = "s3://us-west-2-aws-training/awsu-ilt/AWS-200-BIG/v3.0/data/lab-6-spark/processed\_data.csv"

+++ Step 4.4.2

val df = spark.read.option("header","true").option("inferSchema","true").csv(s3\_location)

+++ Step 4.4.3

df.printSchema()

+++ Step 4.4.4

df.filter(df("tip\_amount") > 0).groupBy("passenger\_count").agg(mean("tip\_amount")).sort("passenger\_count").show()

+++ Step 4.4.5

df.groupBy("passenger\_count").count().sort("passenger\_count").show()

==================================================================================================================

Task 4.5: Transfer Data To Amazon Redshift Using Amazon EMR Spark

==================================================================================================================

+++ Step 4.5.1

val jdbc\_url ="<JDBC\_URL>"+"?user=masteruser&password=Labpass1234"

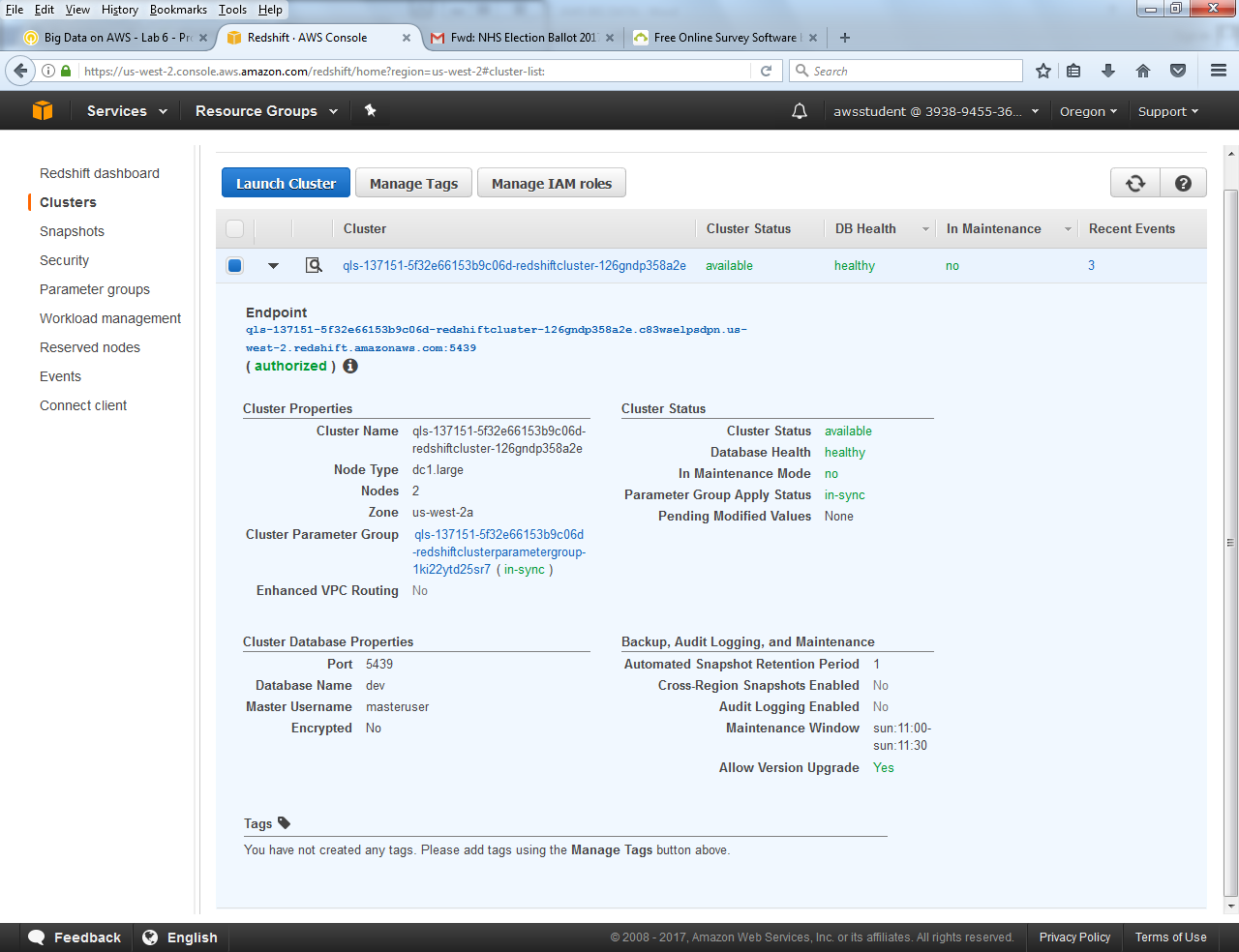
+++ Step 4.5.2

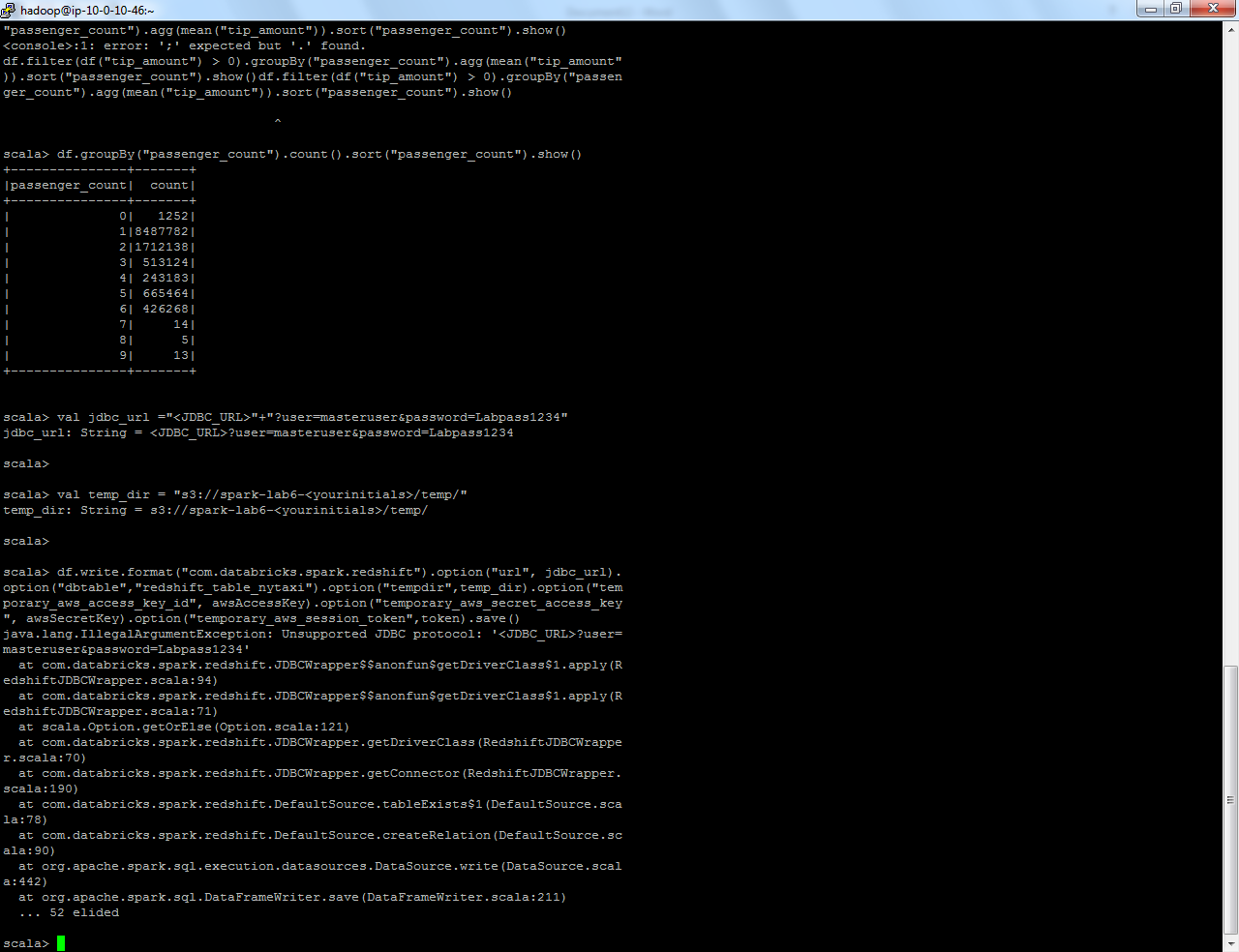
val temp\_dir = "s3://spark-lab6-<yourinitials>/temp/"

+++ Step 4.5.3

df.write.format("com.databricks.spark.redshift").option("url", jdbc\_url).option("dbtable","redshift\_table\_nytaxi").option("tempdir",temp\_dir).option("temporary\_aws\_access\_key\_id", awsAccessKey).option("temporary\_aws\_secret\_access\_key", awsSecretKey).option("temporary\_aws\_session\_token",token).save()

Output





© 2017