EMR Workshop Lab 3 – Spark-based ETL on EMR

This lab demonstrates submitting and monitoring Spark-based ETL work to an Amazon EMR cluster.

You can submit Spark job to your cluster interactively, or you can submit work as a EMR step using the console, CLI, or API. You can submit steps when the cluster is launched, or you can submit steps to a running cluster.

Job Description:

This sample ETL job does the following things:

* Read CSV data from Amazon S3
* Add current date to the dataset
* Write updated data back to Amazon S3 in Parquet format

Exercise 1: Running Spark job on Amazon EMR

* Create an S3 bucket with folders:
* files
* logs
* input
* output
* Get sample data from here (1.8MB file): <https://s3.amazonaws.com/aws-data-analytics-blog/emrimmersionday/tripdata.csv>
* Upload file to your "input" folder in your S3 bucket
* SSH to master node of your previously created cluster.
* Copy and paste the following script into spark-etl.py, make sure that you don’t have invisible characters. Use vi on mac/Linux or Notepad on windows. Alternatively, you can download it from [here](https://s3.amazonaws.com/aws-data-analytics-blog/emrimmersionday/spark-etl.py).

import sys

from datetime import datetime

from pyspark.sql import SparkSession

from pyspark.sql.functions import \*

if \_\_name\_\_ == "\_\_main\_\_":

print(len(sys.argv))

if (len(sys.argv) != 3):

print("Usage: spark-etl [input-folder] [output-folder]")

sys.exit(0)

spark = SparkSession\

.builder\

.appName("SparkETL")\

.getOrCreate()

nyTaxi = spark.read.option("inferSchema","true").option("header", "true").csv(sys.argv[1])

updatedNYTaxi = nyTaxi.withColumn("current\_date",lit(datetime.now()))

updatedNYTaxi.printSchema()

print(updatedNYTaxi.show())

print("Total number of records: " +

str(updatedNYTaxi.count())

updatedNYTaxi.write.parquet(sys.argv[2])

* Submit that pySpark “spark-etl.py” job on the cluster. This Spark job will query the NY taxi data from input location, add a new column “current\_date” and write transformed data in the output location in Parquet format.

>> spark-submit spark-etl.py s3://<YOUR-BUCKET>/input/ s3://<YOUR-BUCKET>/output/spark

* Check the “output/spark” in 3 minutes.

Exercise 2: Monitoring Spark job on Amazon EMR

There are several ways to monitor Spark job status, logs on Amazon EMR. Those are:

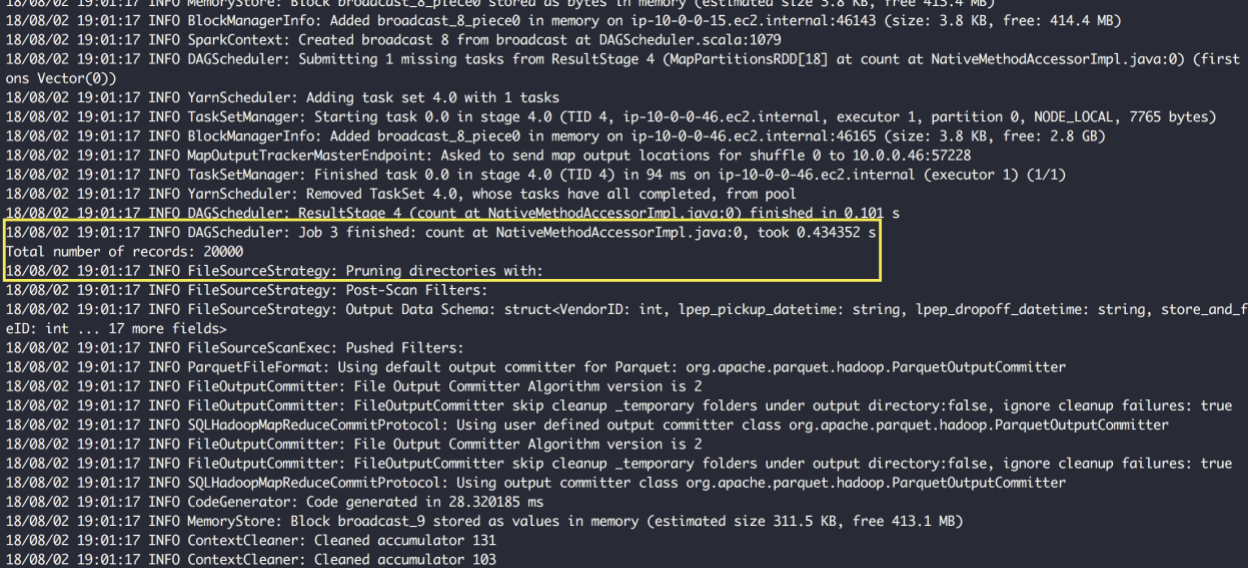
* Check Spark job logs on the command line
* Check YARN Application logs on Amazon EMR Console
* Check status and logs on Spark UI

#### Test 1: Check Spark job logs on the command line:

When Spark job submitted through spark-submit on the command line, it shows up logs on the console. You can pipe that output to a file and grep that file to troubleshoot or you can check the status, output, debug printout on the terminal as well. For example, this job prints out input schema on the terminal:



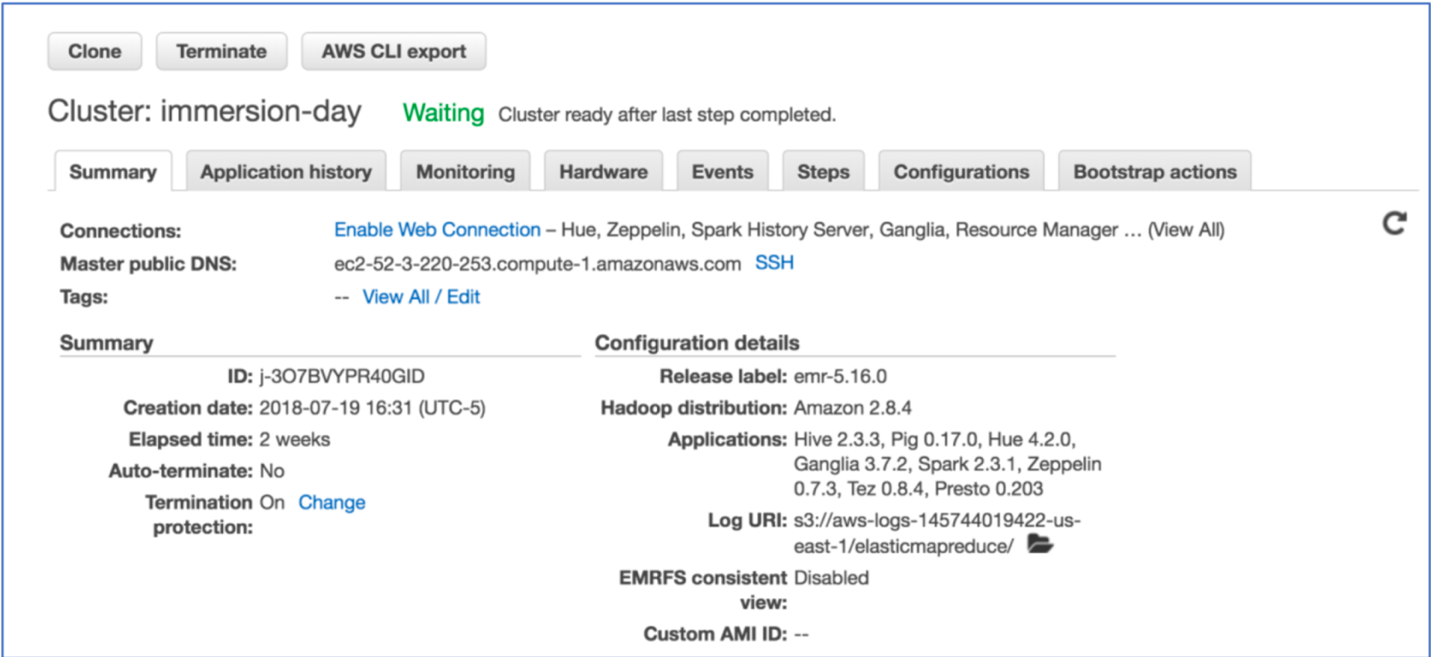
At the end of the job, it also prints out the total number of rows in NY Taxi data:



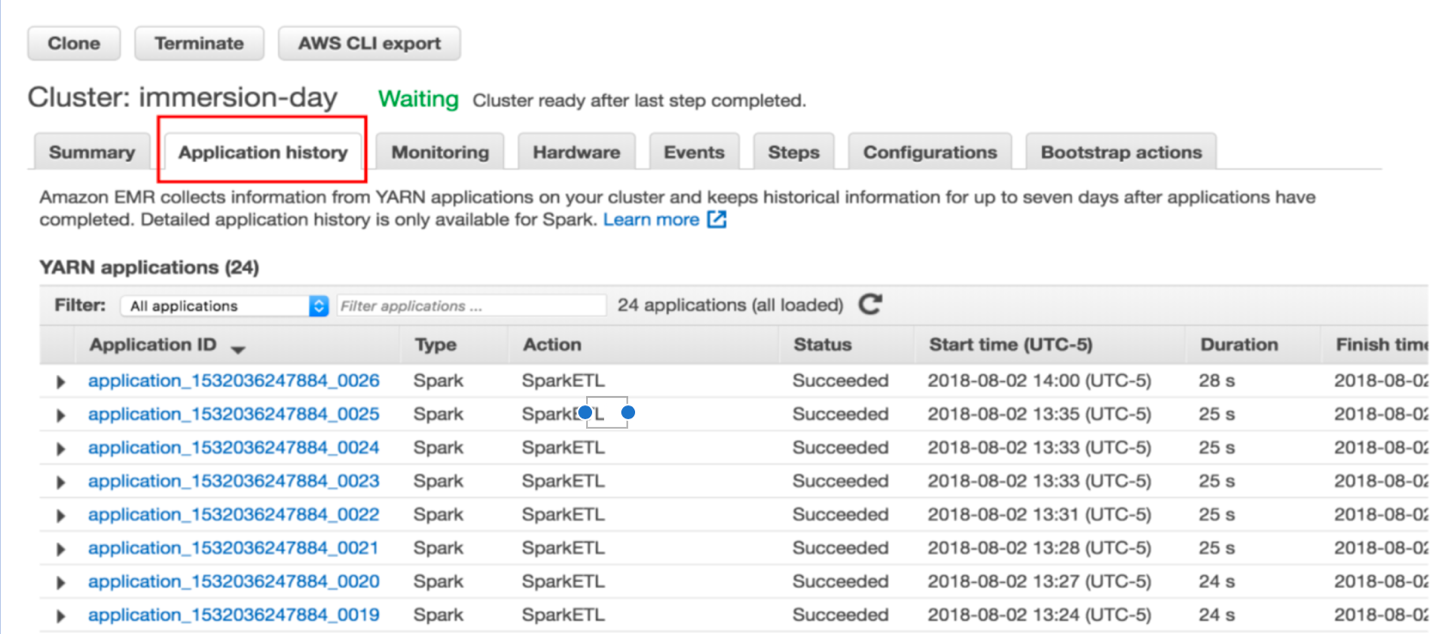
#### Test 2: Check YARN Application logs on Amazon EMR Console

Spark job submitted on Amazon EMR cluster run as YARN application. you can view YARN application details using the Application history tab of a cluster's detail page in the console. Using Amazon EMR application history makes it easier for you to troubleshoot and analyze active jobs and job history. Instead of setting up and connecting to the master node to view open-source troubleshooting UIs or sift through log files, you can quickly view application metrics and access relevant log files.

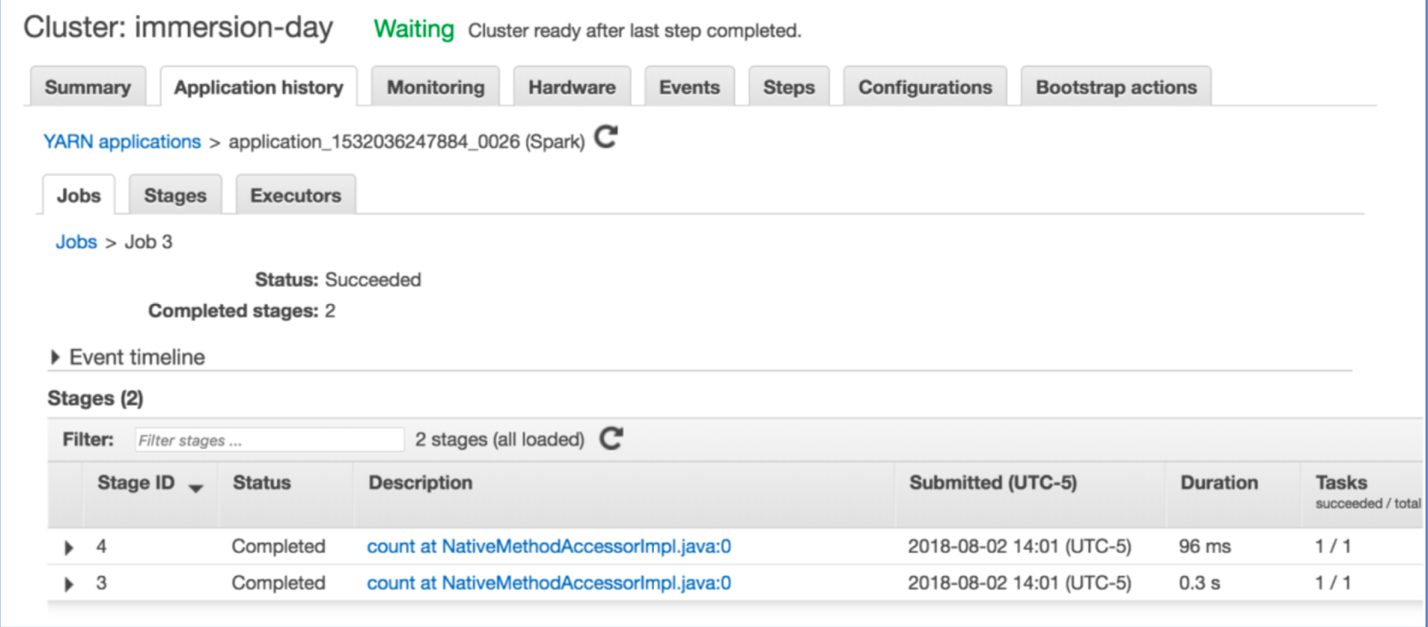
Go to Amazon EMR console and select the cluster you created in the previous step to open up the cluster details window:

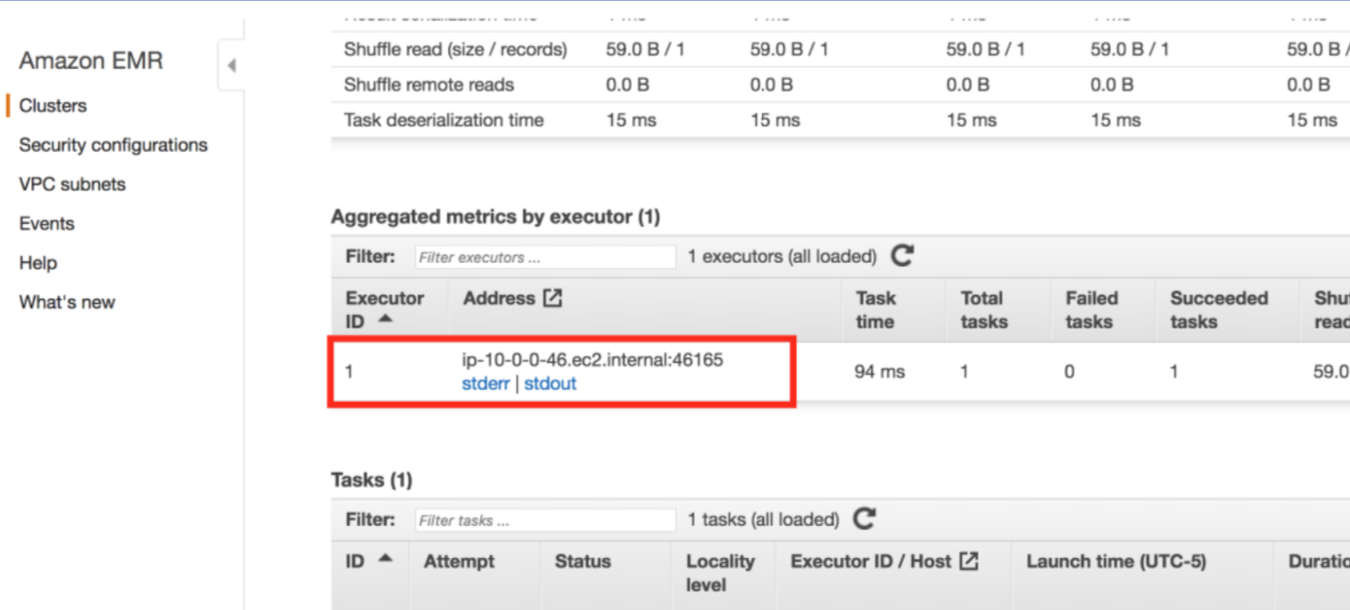


Click on the “Application history” tab

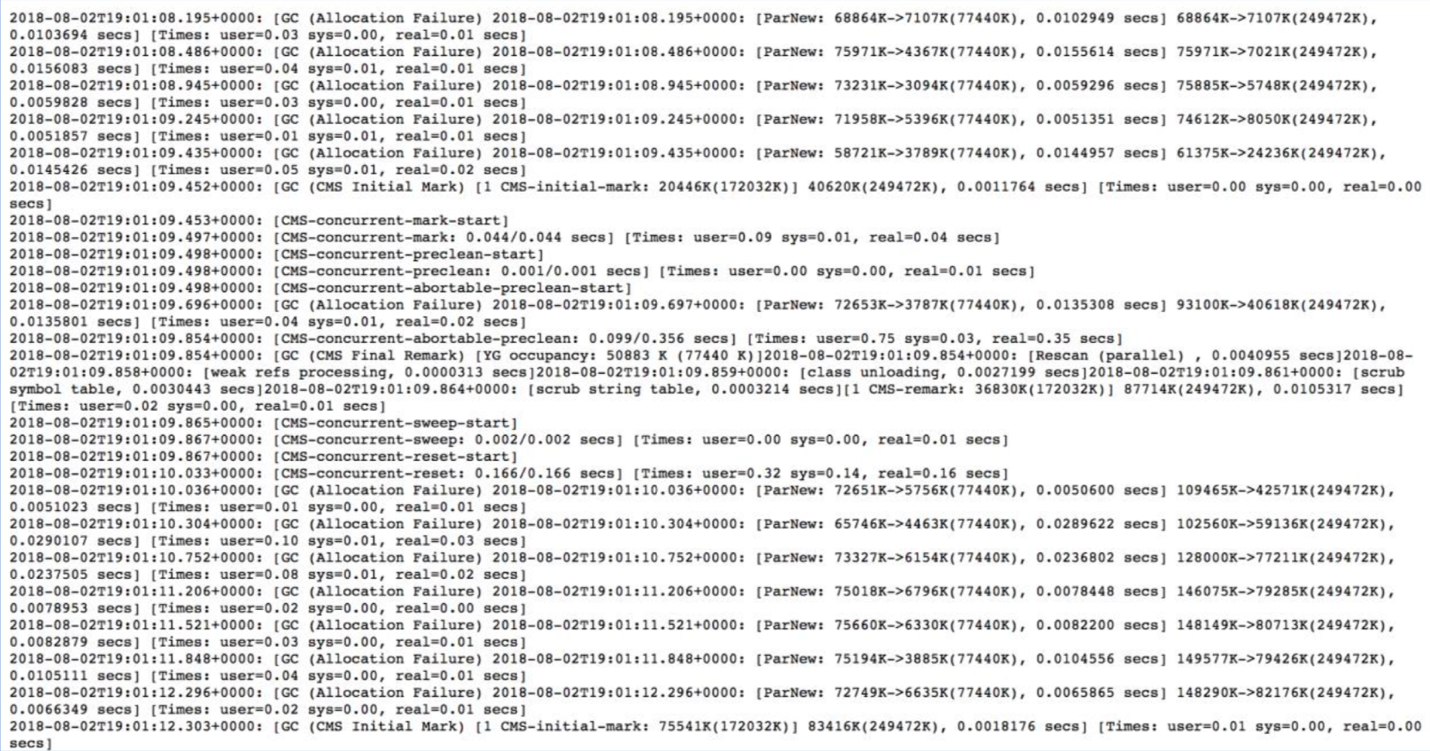


Application history tab shows a list of applications that are executed on the Amazon EMR cluster. Now click on the job you just submitted and check its log



You can drill down further to retrieve “stdout” and “stderr” logs: 

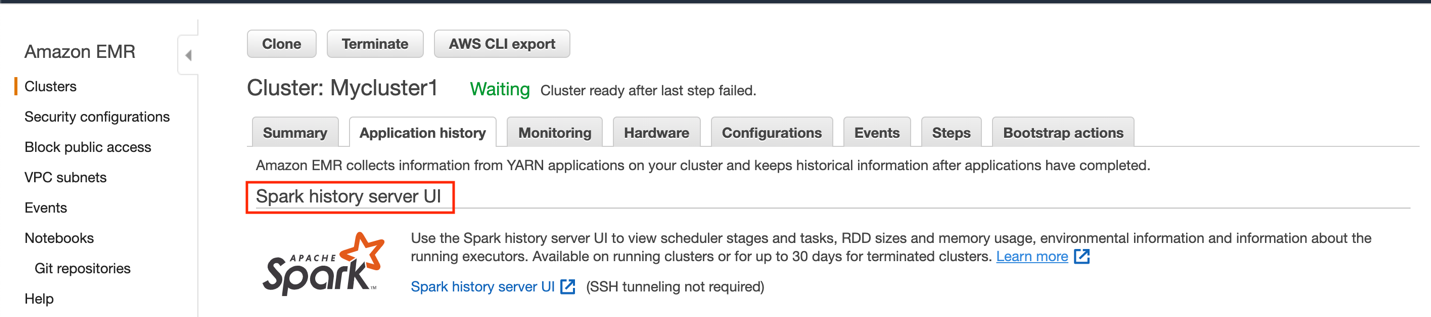
Clicking on the “stdout” or “stderr” will open up the logs in a different window



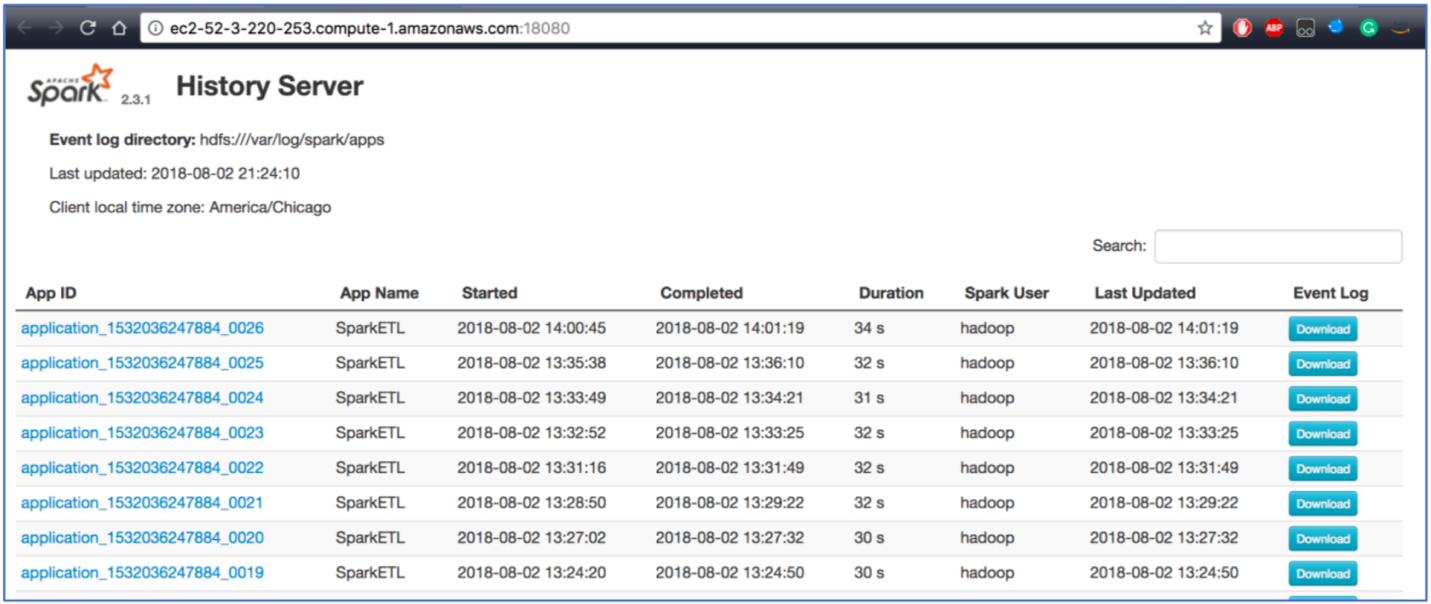
#### Test 3: Check status and logs on Spark UI

Spark UI displays useful information about the application you submit on Amazon EMR. Starting EMR version 5.25.0 you can view the Spark web UIs directly on the AWS console.

On Application history tab click Spark history server UI:



This will bring up the Spark UI. You may need to disable popup blocker on your browser:



Select application ID of your most recent Spark job and drill-down logs and check DAG of different stages:

