## Lab 5-01: Data Analytics with Spark and EMR

### Introduction

**Amazon EMR**

Amazon EMR is the industry's most advanced cloud big data platform for data processing, interactive analysis, and machine learning. That leverages open-source frameworks such as Apache Spark, Apache Hive, and Presto. EMR allows you to execute petabyte-scale analysis for less than half the cost of typical on-premises solutions and more than 1.7x quicker than ordinary Apache Spark.

Amazon EMR frees you up to focus on data transformation and analysis rather than maintaining computing resources or open-source apps, and it saves you money. You may rapidly deploy as much or as little capacity as you want on Amazon EC2 using EMR. You can build up scaling rules to handle changing compute demand. You may configure CloudWatch alerts to notify you of changes in your infrastructure and take fast action. You may use EMR to submit workloads to Amazon EKS clusters if you utilize Kubernetes. Whether you utilize EC2 or EKS, EMR's optimized runtimes speed up your analysis and save you time and money.

**Amazon Simple Storage Service (S3)**

Amazon S3 is a type of object storage that allows you to store and recover any quantity of data from any location. It is a low-cost storage solution with business resilience, reliability, efficiency, privacy, and infinite expansion.

Amazon S3 is a web service that allows you to store and retrieve infinite data from any place and at any time. You may quickly create projects that integrate cloud-native storage using this service. Because Amazon S3 is easily customizable and you only pay for what you use, you can start small and scale up as needed without sacrificing performance or dependability.

Amazon S3 is built to be highly adaptable. Instead of finding out how to store their data, Amazon S3 allows developers to focus on innovation. Build a simple FTP system or a complex web application like the Amazon.com retail website. Read the same piece of data a million times or only for emergency disaster recovery; store whatever type and amount of data you desire.

### Problem

Assume you are data analytics in an organization. The organization that you work in has a wide variety of users. They give you the task of running some data analytics for an upcoming marketing campaign. They give you a target to determine the most common users, grouped by gender and age. Hence, how can you automate this task?

### Solution

The solution is you use the AWS service to automate this task. To accomplish this, you must first build an AWS EMR cluster and upload user data into HDFS. Following that, you will execute a PySpark Apache Spark script to count the number of users and categories them according to their age and gender. Finally, you will need to load the results into the AWS S3 bucket for further analysis.

**Note:** Before starting the lab, create an S3 bucket as used in this lab.

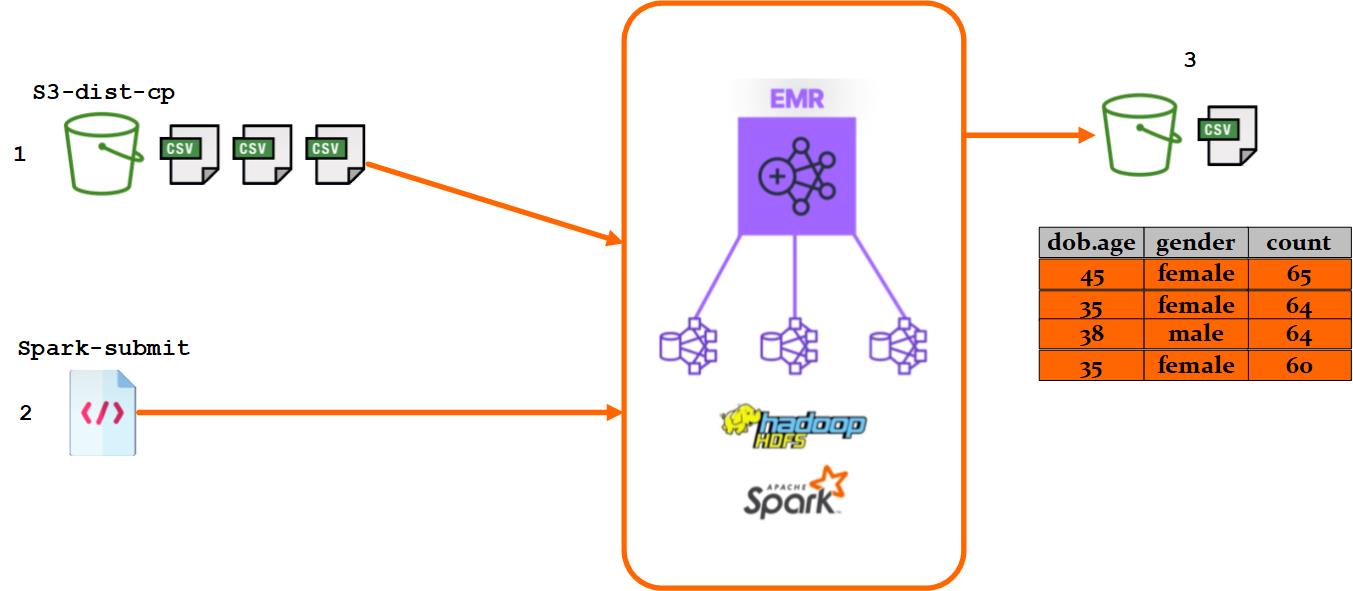


Figure 5-43: Data Analysis with Spark and EMR

|  |
| --- |
| **Step 1: Create AWS EMR Cluster**   1. Log in to the **AWS Console**. 2. Click on the **Services.**      1. Select the **EMR** from the **Analytics.**      1. Click on the **Create Cluster** button.      1. Click on the **Go to advanced options**.      1. Select the **Hadoop 2.10.1** and **Spark 2.4.7**.      1. Scroll down. Click on the **Next** button.      1. Click on the **Pencil** icon button.      1. Select the **m4.large** instance type. Then click on the **Save** button.      1. Click on the **Pencil** icon button.      1. Select the **m4.large** instance type. Then click on the **Save** button.      1. Update the instance count for the core node to **one**.      1. Scroll down. Click on the **Next** button.      1. Give a name **ips-age-gender-analytics-cluster**.      1. Click on the **Next** button.      1. Click on the **Create Cluster** button.      1. It will take 10 to 15 minutes to create a cluster.      1. Hence, successfully created the AWS EMR cluster. |

|  |
| --- |
| **Step 2: Open the HDFS Port**   1. Click on the **Application** **user interfaces** tab.      1. Copy the **Port Number** from the HDFS Name Node URL.      1. Click on the **Summary tab**.      1. Click on the **Security Groups for Master URL** in the new browser tab.      1. Select the **Elastic Map reduce-master** security group ID link.      1. Click on the **Edit inbound rules** button.      1. Scroll down. Click on the **Add rule** button.      1. Leave the type as the **Custom TCP**. 2. Paste the copied HDFS **Port Number**. 3. Select the **o.0.0.0/0** IP address.      1. Click on the **Save rules** button.      1. Close the Security Groups tab and go to the **AWS EMR dashboard**. |

|  |
| --- |
| **Step 3: Copy Data from S3 to HDFS Using s3-dist-cp Command**   1. Click on the **Application user interfaces** tab.      1. Open the **HDFS Name Node URL** on the new tab.      1. Click on the **Utilities**. Then click on the **Browser the file system**.      1. You will see the different folders.      1. Go back to the AWS EMR Cluster dashboard. 2. Click on the **Steps** tab.      1. Click on the **Add Step** button.      1. Select the **Custom JAR** step type.      1. Give a name **ISP Copy data and script to HDFS**.      1. In the JAR location, copy and paste the following command **command-runner.jar.**      1. In the argument field copy and paste the following command **s3-dist-cp --src=s3://das-c01-data-analytics-specialty/Data\_Analytics\_With\_Spark\_and\_EMR/ --dest=hdfs: ///**      1. Select **Continue** in action on failure.      1. Click on the **Add** button.      1. It will take a few minutes to complete. The step copies data from the S3 bucket onto the Hadoop cluster.      1. Hence, it completes the step. |

|  |
| --- |
| **Step 4: Run a PySpark Script Using spark-submit Command**   1. Go back to the **Hadoop HDHS** tab. Click on the **Refresh** button.      1. Click on the **user-data-acg** folder.      1. The folder contains hundreds of CSV files.      1. Go back and Click on the **pyspark-script** folder.      1. The folder contains the PySpark script.      1. Below is the figure of PySparck python code.      1. Go back to the AWS EMR dashboard. Click on the **Add Step** button.      1. Select the **Custom JAR** step type.      1. Give a name **IPS Run PySpark script**.      1. In the JAR location, copy and paste the following command **command-runner.jar**.      1. In the argument field, copy and paste the following command **spark-submit hdfs:///pyspark-script/emr-pyspark-code.py**      1. Select **Continue** in Action on failure.      1. Click on the **Add button**.      1. The step takes some time to run the PySpark script.      1. After the step is completed, click on the **View logs** of the **IPS Run PySpark script.**      1. Click on the **stdout**.      1. The list shows 20 results, and the top shows the most common age and gender.      1. Go back to the **Hadoop HDFS** tab and view the root directory by entering **/** into the search bar.      1. The folder contains the CSV file. |

|  |
| --- |
| **Step 5: Copy Data from HDFS to S3 Using s3-dist-cp Command**   1. Click on the **Services**.      1. Select the **S3** from the **Storage**.      1. Click on the **Create bucket** button.      1. Give a name **ips-gender-age-analytics-bucket**.      1. Click on the **Create Bucket** button.      1. Click on the **Add Step** button.      1. Select the **Custom JAR** step type. 2. Give the name **IPS Load results to S3**. 3. In the JAR location, copy and paste the following command **command-runner.jar**.      1. In the argument field copy and paste the following command **s3-dist-cp --src=hdfs:///results --dest=s3://<YOUR\_BUCKET\_NAME>/**      1. Select **Continue** in action on failure.      1. Click on the **Add button**.      1. The step takes some time to copy the data from HDFS into the S3 bucket.      1. Hence, the step is completed.      1. Go to the S3 dashboard. Click on the **ips-gender-age-analytics-bucket.**      1. Select the file. Click on the **Actions**. Then click on the **Download as**.      1. Open the downloaded file to verify that it shows all the data, aggregated by age and gender. |