

Module 11 Lab:

Accelerated Spark with RAPIDS on AWS

Download the [Lab Dataset](#) before you begin.

Lab Overview

Many modern-day datasets are huge and truly exemplify “big data”. For example, the Facebook social graph is petabytes large (over 1M GB); every day, Twitter users generate over 12 terabytes of messages; and the NASA Terra and Aqua satellites each produce over 300 GB of MODIS satellite imagery per day. These raw data are far too large to even fit on the hard drive of an average computer, let alone to process and analyze. Luckily, there are a variety of modern technologies that allow us to process and analyze such large datasets in a reasonable amount of time. For the bulk of this lab, you will be working with a dataset of over 1 billion individual taxi trips from the New York City Taxi & Limousine Commission (TLC). Further details on this dataset are available [here](#).

For this lab, you will use PySpark on AWS using Elastic MapReduce (EMR) to analyze large samples from the TLC dataset.

Analyzing Large Amount of Data with PySpark on AWS

VERY IMPORTANT: Use Firefox, Safari or Chrome when configuring anything related to AWS.

You will try out PySpark for processing data on Amazon Web Services (AWS). [Here](#) you can learn more about PySpark and how it can be used for [data analysis](#). You will be completing a task that may be accomplished using a commodity computer (e.g., consumer-grade laptops or desktops). However, we would like you to use this exercise as an opportunity to learn distributed computing on Amazon EC2, and to gain experience that will help you tackle more complex problems.

The services you will primarily be using are Amazon S3 storage, Amazon Elastic Cloud Computing (EC2) virtual servers in the cloud, and Amazon Elastic MapReduce (EMR) managed Hadoop framework. You will be creating an S3 bucket, running code through EMR, then storing the output into that S3 bucket.

For this question, you will only use up **a very small fraction of your AWS credit**.

AWS Guidelines

Please read the [AWS Setup Tutorial](#) to set up your AWS account. Instructions are provided both as a written guide, and a video tutorial.



Datasets

In this question, you will use a dataset of trip records provided by the New York City Taxi and Limousine Commission (TLC). Further details on this dataset are available [here](#) and [here](#). From these pages [1] [2], you can explore the structure of the data, however you will be accessing the dataset directly through AWS via the code outlined in the homework skeleton. You will be working with two samples of this data, one small, and one much larger.

EXTREMELY IMPORTANT: Using machines in other regions for computation would incur data transfer charges. Hence, to avoid cross-region data transfer, set your region to **US East (N. Virginia)** in the beginning (not Oregon, which is the default). While you can select a different region, you must consistently use the same region throughout your setup. **This is extremely important, otherwise your code may not work, and you may be charged extra.**

Goal

You work at NYC TLC, and since the company bought a few new taxis, your boss has asked you to locate potential places where taxi drivers can pick up more passengers. Of course, the more profitable the locations are, the better. Your boss also tells you not to worry about short trips for **any** of your analysis, so only analyze trips which are **2 miles or longer**.

First, find the **20** most popular drop off locations in the Manhattan borough by finding which of these destinations had the greatest **passenger count**.

Now, analyze all pickup locations, regardless of borough.

- For each pickup location determine
 - the **average total amount** per trip,
 - the total **count** of all trips that start at that location, and
 - the **count** of all trips that start at that location and end at one of most popular drop off locations.
- Using the above values,
 - determine the **proportion** of trips that end in one of the popular drop off locations (# trips that end in drop off location divided by total # of trips) and
 - multiply that proportion by the **average total amount** to get a **weighted profit value** based on the probability of passengers going to one of the popular destinations.

Bear in mind, your boss is not as savvy with the data as you are and is not interested in location IDs. To make it easy for your boss, provide the **Borough** and **Zone** for each of the top 20 pickup locations you determined. To help you evaluate the correctness of your output, we have provided you with [the output for the small dataset](#).

Tasks

You are provided with a python notebook (pyspark.ipynb) file which you will complete and load into EMR. You are provided with the `load_data()` function, which loads two PySpark DataFrames. The first is **trips** which contains a DataFrame of trip data, where each record refers to one (1) trip. The second is **lookup** which maps a LocationID to its information. It can be linked to either the PULocationID or DOLocationID fields in the trips DataFrame.

The following functions must be completed for full completion.



VERY IMPORTANT

- Ensure that the parameters for each function remain as defined and the output order and names of the fields in the PySpark DataFrames are maintained.
- Use PySpark methods to complete this lab, not SQL commands.

- a) **user()**
 - i. Returns your username as a string (e.g., janedoe3)
- b) **long_trips(trips)**
 - i. This function filters trips to keep only trips 2 miles or longer (≥ 2).
 - ii. Returns PySpark DataFrame with the same schema as **trips**
 - iii. **Note: Parts c, d and e will use the result of this function**
- c) **manhattan_trips(trips, lookup)**
 - i. This function determines the top 20 locations with a *DOLocationID* in Manhattan by passenger count.
 - ii. Returns a PySpark DataFrame with the schema (*DOLocationID*, *pcount*)
- d) **weighted_profit(trips, mtrips)**
 - i. This function determines
 - i. the average *total_amount*,
 - ii. the *total count of trips*, and
 - iii. the *total count of trips ending in the top 20 destinations*
 - iv. and return the *weighted_profit* as discussed earlier in the homework document.
 - v. Returns a PySpark DataFrame with the schema (*PULocationID*, *weighted_profit*) for the *weighted_profit* as discussed earlier in this homework document.
- e) **final_output(wp, lookup)**
 - i. This function
 - i. takes the results of *weighted_profit*,
 - ii. links it to the *borough* and *zone* through the **lookup** data frame, and
 - iii. returns the top 20 locations with the highest *weighted_profit*.
 - ii. Returns a PySpark DataFrame with the schema (*Zone*, *Borough*, *weighted_income*)

Once you have implemented all these functions, run the `main()` function, which is already implemented, and update the line of code to include the name of your output s3 bucket and a location. **This function will fail** if the output directory already exists, so make sure to **change it each time** you run the function.

Example: `final.write.csv('s3://lab11-janedoe3/output3.csv')`

Your output file will appear in a folder in your s3 bucket as a csv file with a name which is similar to *part-0000-4d992f7a-0ad3-48f8-8c72-0022984e4b50-c000.csv*. Download this file and **rename it to output.csv** for submission. Do **not** make any other changes to the file.

Hint: Refer to commands such as `filter`, `join`, `groupBy`, `agg`, `limit`, `sort`, `withColumnRenamed` and `withColumn`.

Completed Results

1. **pyspark.ipynb**: The PySpark notebook for the question (using the **larger** data set).
2. **output.csv**: Output (**comma-separated**) (using the **larger** data set).





Setup Guide For Lab11

Getting Started

1. Create an AWS Educate account


Go to the AWS Educate [website](#), where you create an AWS Educate account using your academic credentials:



Now, fill in the requested information. Then click next. On the following screen check the box which says 'I Agree' then click Submit. You will receive an email which asks you to confirm your email, then you'll be able to log in to your account.

When you log in you will see a screen like this, so click the button to setup an AWS Educate Starter account to get your \$100 of credits. This will take you to the Vocareum workbench where you can log into your account.





AWS Educate Starter Account

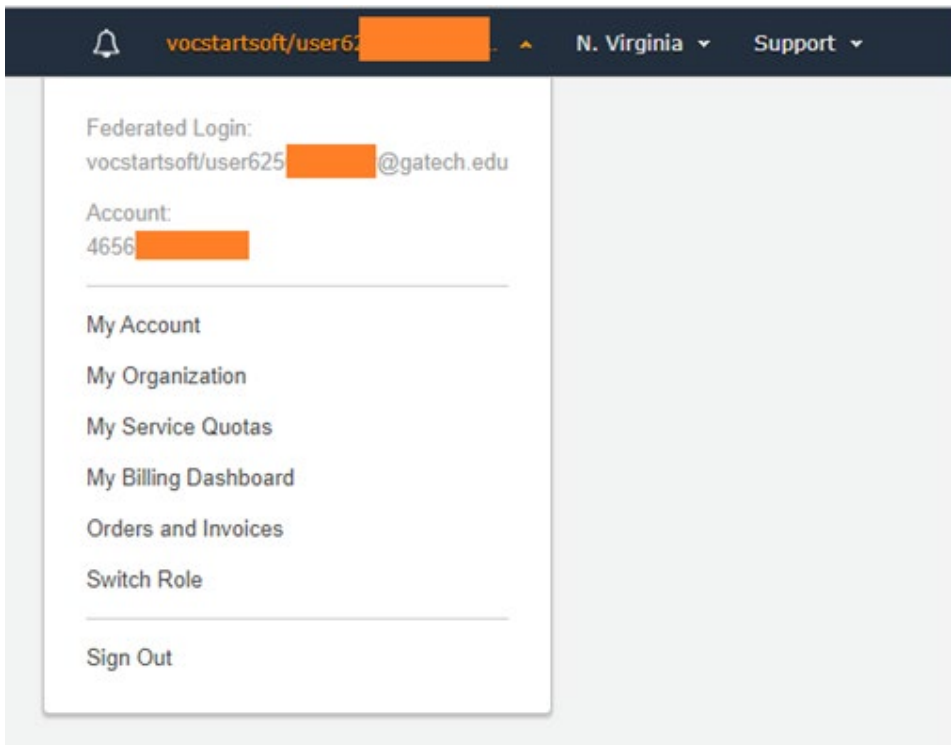
Your cloud journey has only just begun. Use your AWS Educate Starter Account to access the AWS Console and resources, and start building in the cloud!

[AWS Educate Starter Account](#)

Your account has an estimated **100** credits remaining and access will end on **Feb 16, 2021**.

Note: Clicking this button will take you to a third party site managed by Vocareum, Inc. ("Third Party Servicer"). In addition to the AWS Educate terms of service, your use of the AWS Educate Starter Account is governed by the Third Party Servicer's terms, including its Privacy Policy. AWS assumes no responsibility or liability and makes no representations or warranties regarding services provided by a Third Party Servicer.

Once you log in, your dashboard [click AWS console] will look something like this [right top corner].



Notification bell icon | **vocstartsoft/user625** | ^ | **N. Virginia** v | **Support** v

Federated Login:
vocstartsoft/user625 [redacted]@gatech.edu

Account:
4656 [redacted]

My Account
My Organization
My Service Quotas
My Billing Dashboard
Orders and Invoices
Switch Role

Sign Out

If you have any problems with this, or you receive an email from AWS saying that your application has been rejected.



2. Set up a CloudWatch Usage Alert

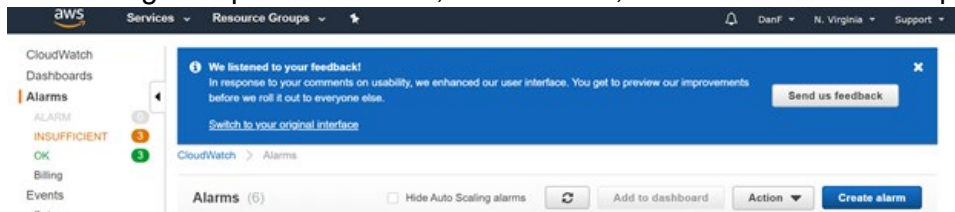
NOTE: There are known issues with setting up billing alerts in via CloudWatch in starter accounts. If you are not able to follow these steps, it is okay and you will still be able to complete the rest of the assignment, however you must be double careful to make sure to close all clusters when not in use.

Make sure your region (in the upper right corner of the screen) is set to: **US East (N. Virginia)** (or whichever region you selected originally for your server but we recommend this region). [Test whether this email alert is working before scheduling in practice](#). That is, out of \$100, when your credit balance goes below say \$95, schedule a test alert and make sure it works. Remember this alert works only once. So, once you get an alert for \$95, you schedule the next alert for \$70 and the next one for \$60 and so on.

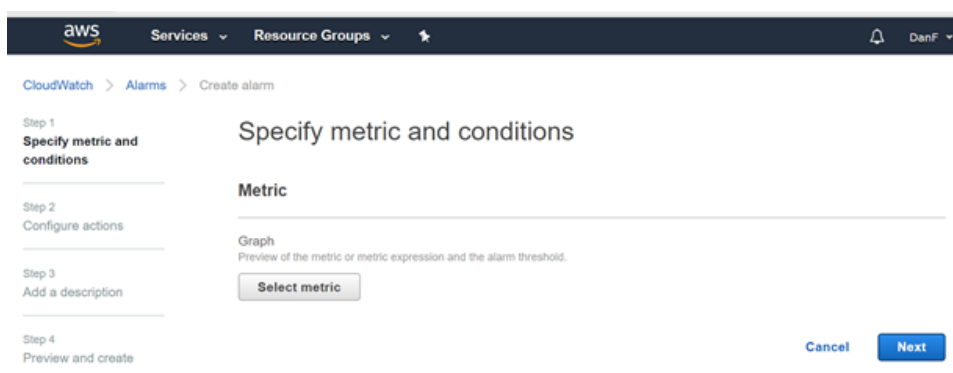
Turn on Custom Alerts

First, we need to create a custom alarm so that it tells you when you have spent money.

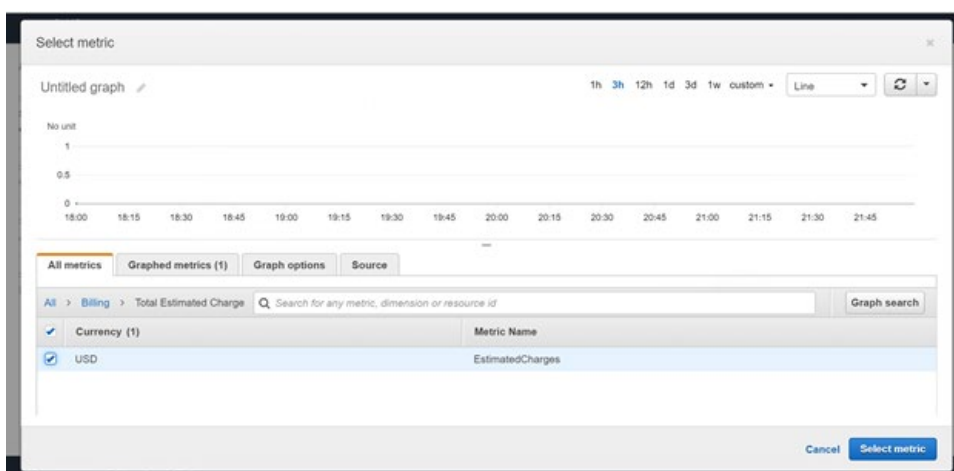
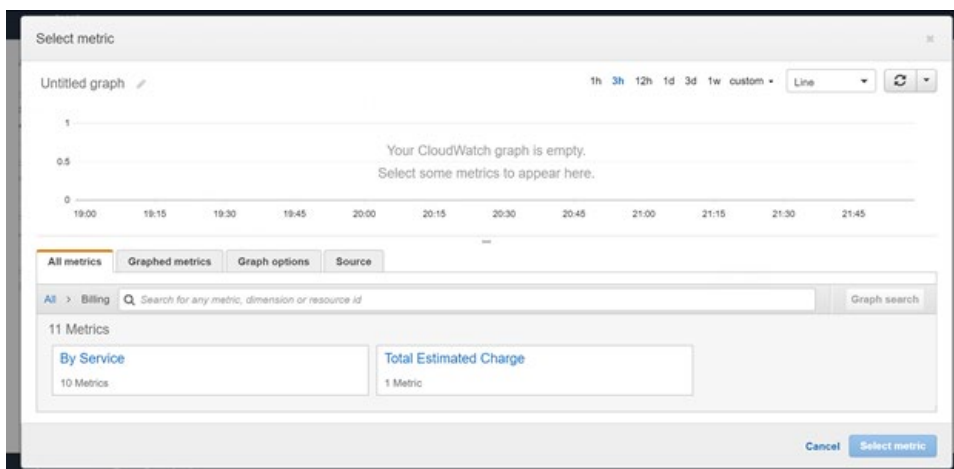
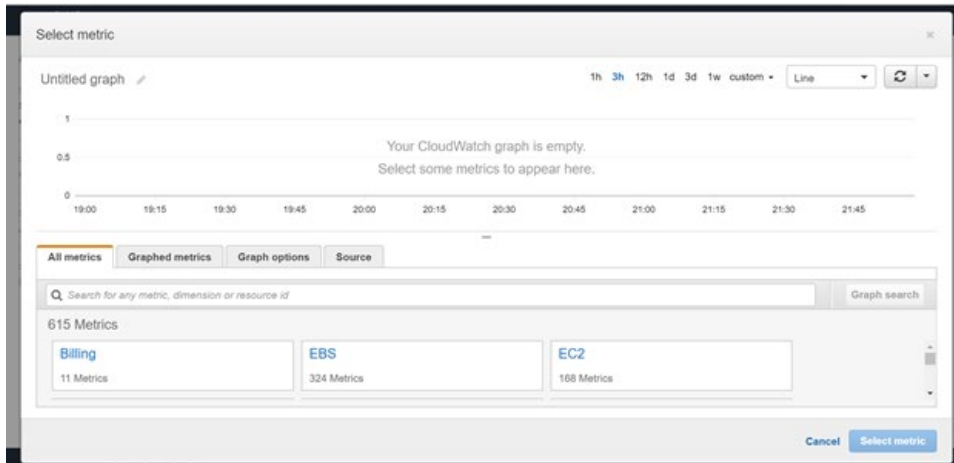
1. Click **CloudWatch** in the AWS Management Console.
2. In the navigation pane on the left, click **Alarms**, and then in the **Alarms** pane, click **Create Alarm**.]



3. Click on **Select Metric**.



4. Under **All metrics**, select **Billing**, then total **Estimated Charge**. Select the checkbox, then click on **Select Metric**.



5. Set up your conditions as below, using default values, and typing 50 for the threshold value. Click next.

Conditions

Threshold type

☒ Static
Use a value as a threshold

☐ Anomaly detection
Use a band as a threshold

Whenever EstimatedCharges is...

Define the alarm condition

☒ Greater
> threshold

☐ Greater/Equal
≥ threshold

☐ Lower/Equal
≤ threshold

☐ Lower
< threshold

than...

Define the threshold value

50 USD

Must be a number

► Additional configuration

Cancel Next

6. Make sure the alarm state is set to 'in Alarm.' Then, select Create a new topic, and enter a name and your email address, then click 'Create topic'. Scroll to the bottom of the screen and click next.

Whenever this alarm state is...

Define the alarm state that will trigger this action

☒ in Alarm
The metric or expression is outside of the defined threshold.

☐ OK
The metric or expression is within the defined threshold.

☐ INSUFFICIENT_DATA
The alarm has just started or not enough data is available.

Remove

Select an SNS topic

Define the SNS (Simple Notification Service) topic that will receive the notification

☐ Select an existing SNS topic

☒ Create new topic

☐ Use topic ARN

Create a new topic...

The topic name must be unique.

Notify-Me

SNS topic names can contain only alphanumeric characters, hyphens (-) and underscores (_).

Email endpoints that will receive the notification...

Add a comma-separated list of email addresses. Each address will be added as a subscription to the topic above.

user@example.com

user1@example.com, user2@example.com

Create topic

7. Enter a name for the alert and click next.



Add a description

Name and description

Define a unique name

Alarm name

Cost Exceeded \$50

Alarm description - optional

Define a description for this alarm. Optionally you can also use markdown.

Alarm description

Up to 1024 characters (0/1024)

Cancel

Previous

Next

8. On this preview screen, scroll to the bottom click Create Alarm

Cancel

Previous

Create alarm

You have now created an alert that will notify you when you have used \$50. Consider creating a few additional alerts (e.g., \$60, \$70) so you will be well informed of your usage!

3. Create storage buckets on S3

We need S3 for two reasons:

- (1) An EMR (Elastic MapReduce) workflow requires the input data to be on S3.
- (2) An EMR workflow output is always saved to S3.

Data (or objects) in S3 are stored in what we call “**buckets**”. You can think of buckets as folders. All S3 buckets need to have unique names. You will need to create some buckets of your own to (1) store your EMR output; and (2) store your log files if you wish to debug your EMR runs. Once you have signed up, we will begin by creating the log bucket first.

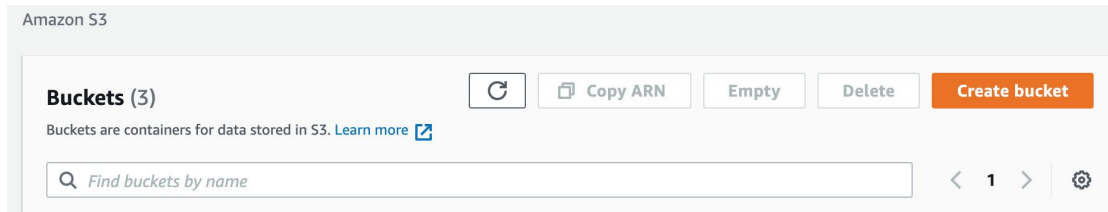
1. First, setup a bucket for your dataset contained in the Lab11_Dataset.zip file:



- a. Use the AWS instructions for setting it up:
<https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html>
- b. Make sure to name your bucket in the appropriate format:
 - i. "s3://lab11-janedoe3"

2. In the AWS Management Console click on **S3** under **All services** → **Storage**.

In the S3 console, click on **Create Bucket**.



3. Create a logging bucket: Enter the following details (bucket name and region) then click **Create Bucket** at the bottom of the screen. Keep all other settings as the same.

Bucket Name Format: c-logging

Example: lab11-janedoe3-logging

Region: US East (N. Virginia)

VERY IMPORTANT: Please select **"US East (N. Virginia)"** only. If you have buckets in other regions, data transfer charges would apply.

Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1 ▼

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose bucket



4. A new bucket will appear in the S3 console. Clicking on it will show you that it is empty.
5. Create the main bucket: Go back to the main screen (clicking on **Amazon S3**). Again, click on **Create Bucket** and enter the following details.

Bucket Name Format: lab11-janedoe3

Example: lab11-janedoe3

Region: US East (N. Virginia)

Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

6. Since we will link this bucket to our logging bucket, the regions for the two buckets should be the same. We will link our logging bucket to the one we are creating now. Once the bucket is created, click on the bucket on the main screen and select the properties tab.

Amazon S3 > lab11-janedoe3

lab11-janedoe3

Objects

Properties

Permissions

Metrics

Management

Access Points

7. Scroll down to **Server Access Logging** and click **Edit**.

Server access logging
Log requests for access to your bucket. [Learn more](#)

Server access logging
Disabled

Edit

8. Select **Enable**, and then make the Target Bucket the logging bucket created in step 2.

Click **Save Changes**

Edit server access logging

Server access logging
Log requests for access to your bucket. [Learn more](#)

Server access logging
☐ Disable
☒ Enable

⚠ By enabling server access logging, S3 console will automatically update your bucket access control list (ACL) to include access to the S3 log delivery group.

Target bucket

Format: s3://bucket/prefix

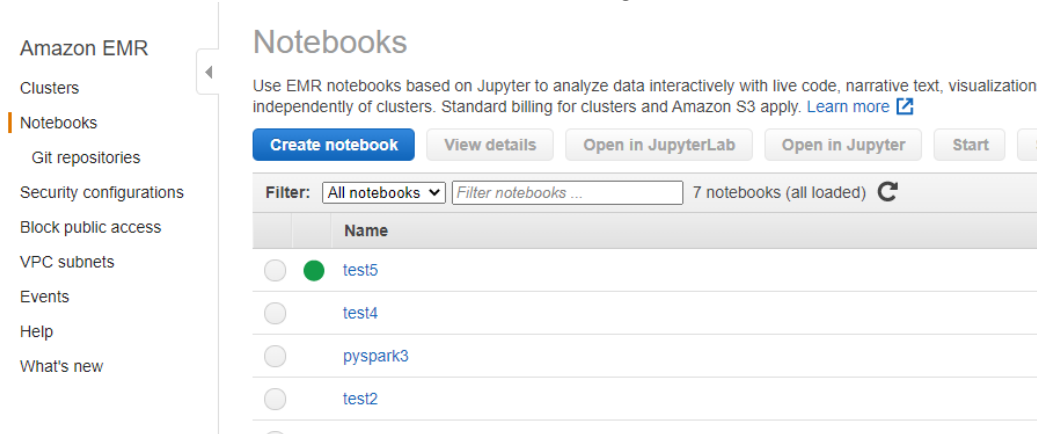
Cancel

We are done with creating S3 buckets at this point.

4. Launch a Notebook

This section will cover launching a Notebook in Amazon EMR. For further information about notebooks in EMR, click [here](#).

1. Go to Amazon EMR. Select Notebooks on the right menu. Click “Create Notebook”.



2. Make sure the region specified in the top-right corner of the page is **N. Virginia**. Otherwise click on it and from the drop-down choose N. Virginia.

3. We will now fill out the various configuration fields to create a new Notebook:

- a. Give your notebook a name. It can be anything you want.
- b. Select the checkbox to “Create a cluster.”

Note: It's okay if the release version in screenshot doesn't match.

- c. For instance type, select **m5.xlarge** (This will likely be the default). You can also change the number of instances used, so select **4**. You can experiment with other instance types and numbers of clusters to see the impact on performance, but there are many which are not eligible to be used on a starter account, so they may result in errors when attempting to create a notebook.
- d. For configuring the Nvidia Spark-RAPIDS Accelerator on your cluster, follow [these instructions on AWS](#)
- e. For AWS service role, select **EMR_Notebooks_DefaultRole**. If this is your first time running EMR, it may also give you the option to “Create Default Role”, which you should do in this case.
- f. For Notebook location, select the s3 bucket (eg: s3://lab11-janedoe3) you created earlier.
- g. Your settings should look something like this. Once you have confirmed this, select “Create Notebook”.



Name and configure your notebook

Name your notebook, choose a cluster or create one, and customize configuration options if desired. [Learn more](#)

Notebook name*

Names may only contain alphanumeric characters, hyphens (-), or underscores (_).

Description

256 characters max.

Cluster* ☐ Choose an existing cluster ☒ Create a cluster i

Cluster name:

Release: emr-5.32.0

Applications: Hadoop, Spark, Livy, Hive, JupyterEnterpriseGateway

Instance:

EMR role: [EMR_DefaultRole](#) i

EC2 instance profile: [EMR_EC2_DefaultRole](#) i


EC2 key pair: i

Security groups ☒ Use default security groups i ☐ Choose security groups (vpc-2d833a50)

AWS service role* i

Notebook location* Choose an S3 location where files for this notebook are saved.

☐ Use a location that EMR creates i ☒ Choose an existing S3 location in us-east-1



► **Git repository**

► **Tags** i

* Required

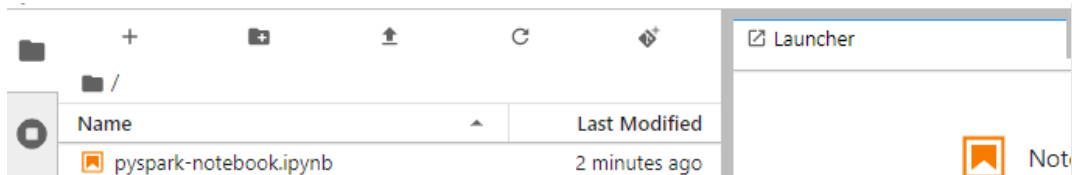
Cancel

Create notebook

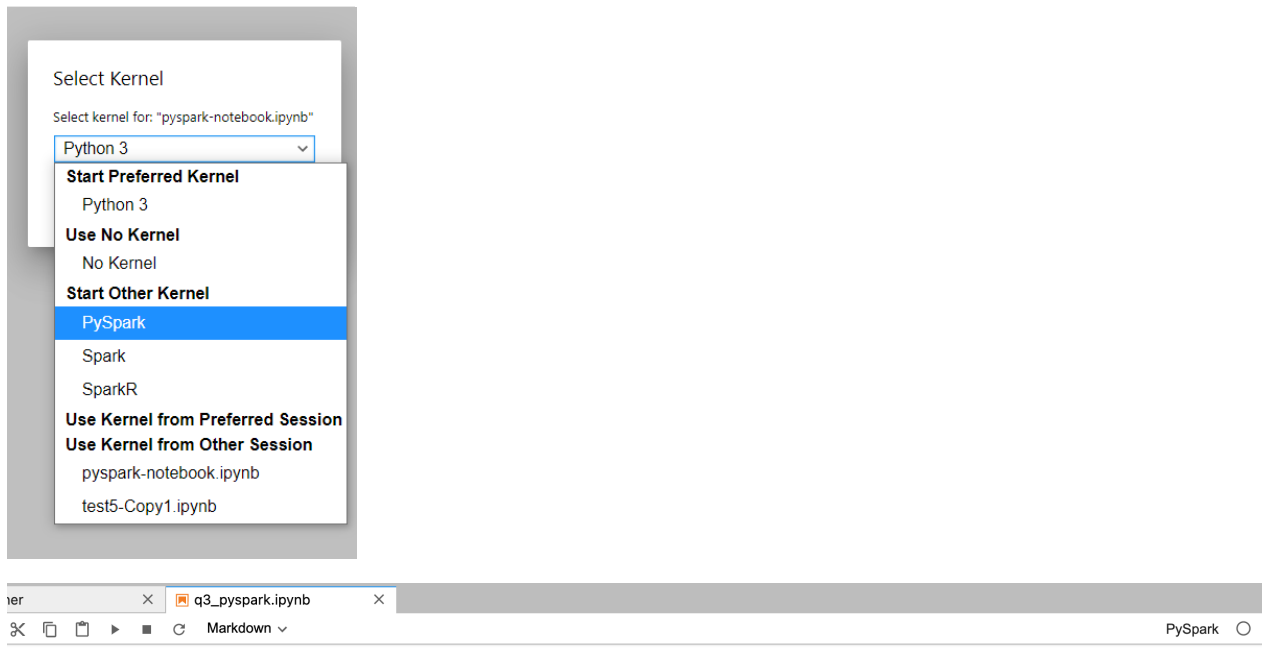
5. Get started with the skeleton

In this section we will upload the skeleton file to the notebook and run our first cell.

1. Once your notebook has finished instantiating and has the status of 'Ready', (this will take several minutes), click "Open in JupyterLab".
2. In the left bar, click the arrow with a line under it to upload a file and upload the pyspark.ipynb file provided in the skeleton.



3. Double click on the newly added file to open it.
4. In the screen that gives you the option to Select a kernel, choose PySpark. If this pop up does not appear, select the Kernel in the top right of the screen to cause this pop up to appear.



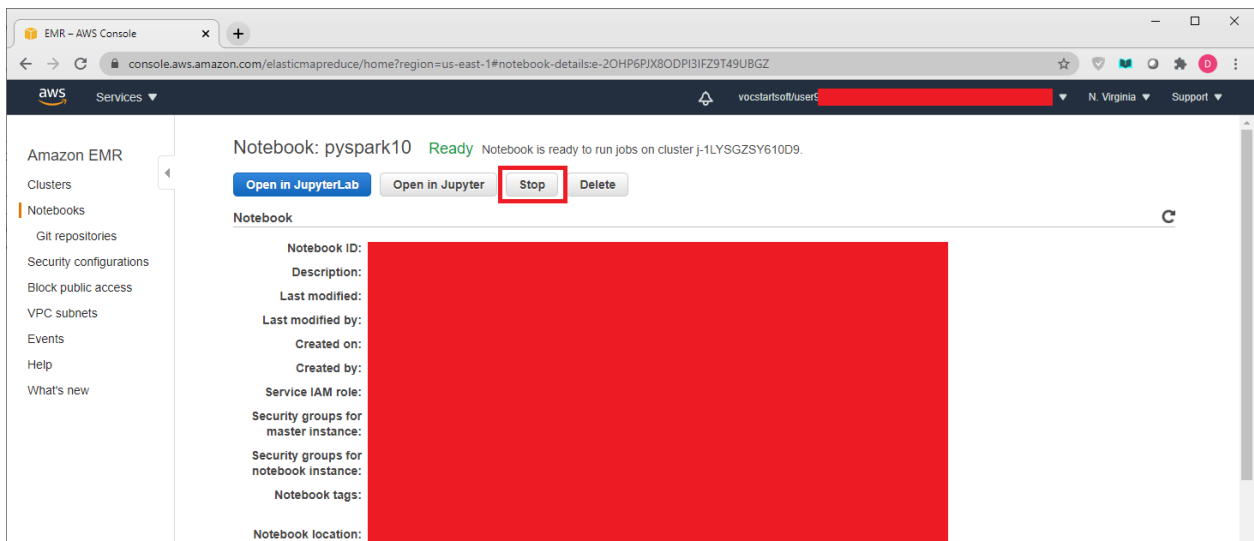
5. Run the first code cell, which should contain `sc` to start the Spark Application so you can start programming the assignment.
6. Once you have finished coding, right click on the file name in the directory on the left and select download to download it. It will also be saved in your S3 bucket,

6. Terminating All Clusters

WARNING: It is very important that you do not leave clusters running when not working on your workbook. Costs can go up quickly and use up your credits.

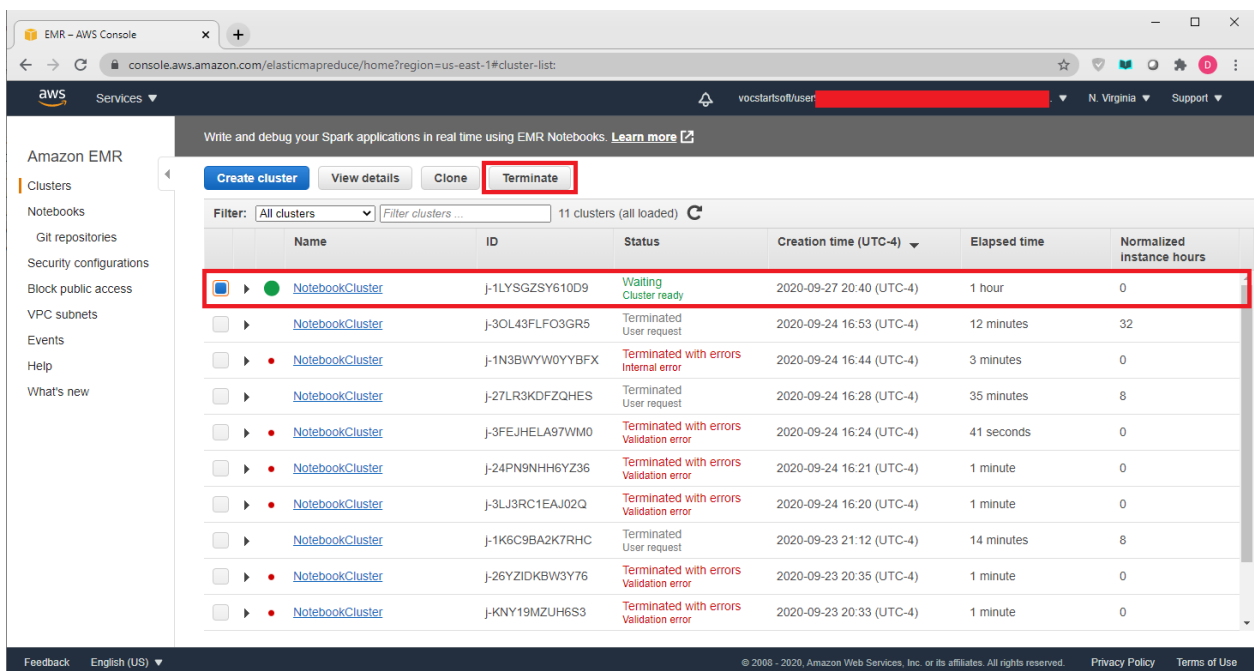
NOTE: The AWS billing report can be as much as six hours behind. It may take up to six hours after terminating all clusters before the billing report stops increasing.

1. Back on your Notebook's page in EMR, click 'Stop'.

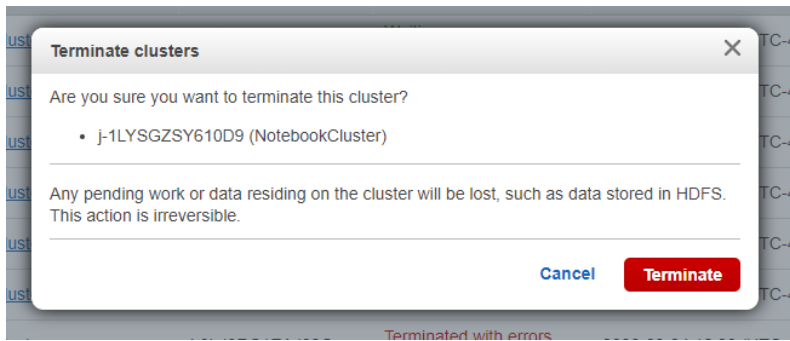


2. Now click on “Clusters” in the side bar on the left. Click the check box next to your running cluster (the one with the green circle) and click “Terminate”.

Note: You may have to refresh your screen for the cluster to show up.



3. In the popup, select ‘Terminate’.

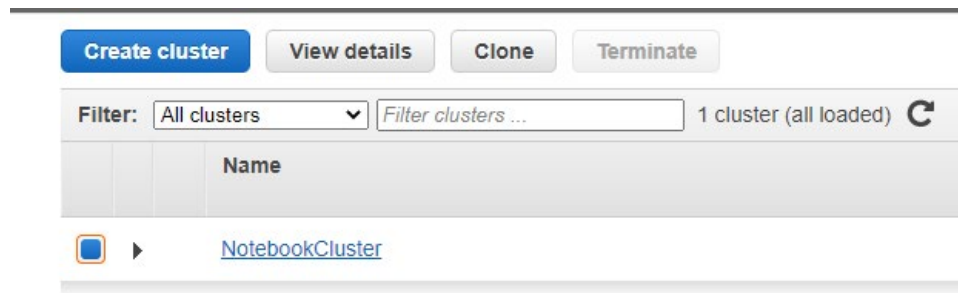


You have now closed all your clusters and will no longer be accruing charges!

7. Restarting an Old Cluster

If you stopped your cluster and took a break and want to start the assignment again, there is a quick and easy way to do so.

1. Clone the old terminated cluster.



2. It will then ask if you would like to copy the setting from the old cluster. Click Yes.
3. Confirm the settings and Start the cloned cluster, waiting 5-10 minutes for it to spin up.
4. You will then have to start your old notebook and attach it to the running cluster.