

1. At least 99.5% of products will be tested every week.
2. Every vendor has 3 labs that they use at different times.
3. The data takes place over a 2 year window.
4. No products or vendors or labs churn during those 2 years.
5. Vendors and Labs are always located in the same state.
6. The output format is always json and each line is a complete entry

- Potency is the sum of all cannabinoids potency:  $total\_potency = [THC + THCA + CBD + CBDA]$ .

# Data Challenge Answer Sheet

Prepared by Vicky Kwan, 04-10-2021

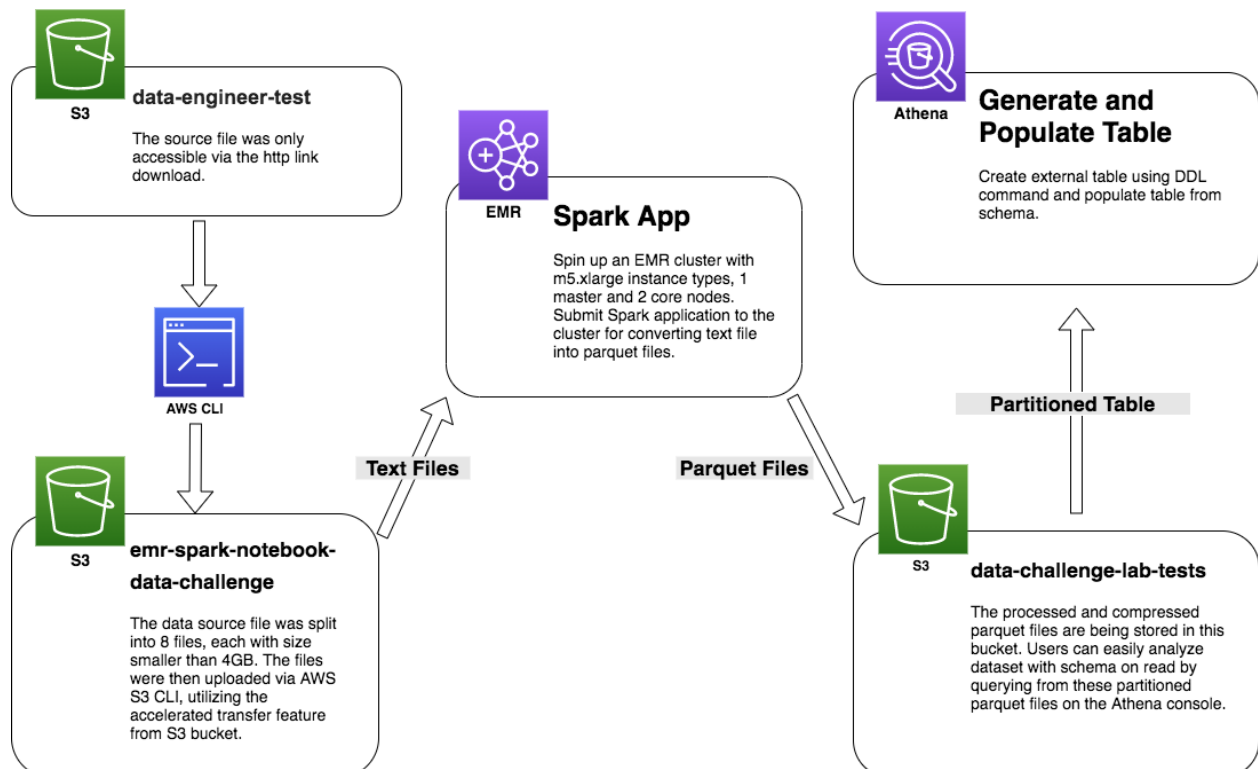
## Problem Statement

The data team would like to analyze a 70GB file, containing test data from cannabis testing labs. We want to analyze this data using the best big data tools and answer a few business questions for our customers. The solution should contain business requirements and constraints, technical setup, staging and loading dataset per schema, cleanup of dataset and noting data quality issues, analysis on key metric, and discussion of alternative methods.

## Design Overview

Due to the scale of the dataset, it will be most efficient to compress the source data and stage it on a data warehouse as a relational table, before analyzing and querying from it. The system is designed such that querying performance will not be limited to the size of source data, to ensure scalability. I spun up an EMR cluster, and used a Spark application to read from the original text file and compress to parquet files. The Spark app would then store these parquet files into a designated S3 bucket for Athena users to stage and query from. A diagram representing such a design can be found below.

## Architecture Diagram

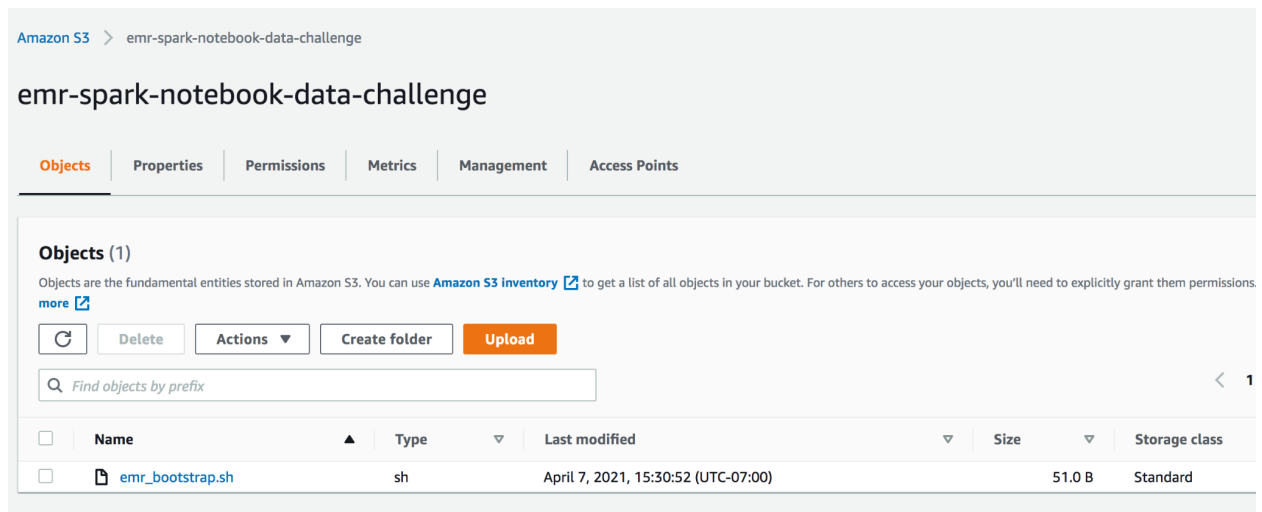


# Environment Setup

I have chosen an EMR cluster with Spark version 2.4 to process the dataset using PySpark. The Jupyter Notebook will be run on this cluster. A little more about how I set this up using the AWS free tier account:

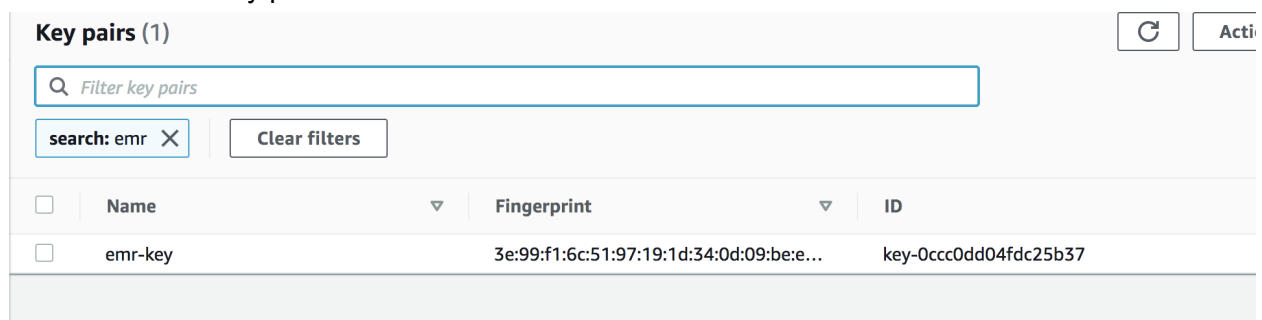
## Set up an S3 bucket

I created a bucket to store the bootstrap script that I want my cluster to install upon creation. This bucket will also host the original dataset from the download link. The bucket name is “[emr-spark-notebook-data-challenge](#)”:



## Create EC2 key pair for cluster

I then created a key pair for the EMR cluster



## Create EMR Cluster with Spark

First, I chose the latest EMR release “emr-5.29.0” with these 4 softwares: Hadoop, Hive, Spark and Livy.

## Software Configuration

Release **emr-5.29.0** ⓘ

<input checked="" type="checkbox"/> Hadoop 2.8.5	<input type="checkbox"/> Zeppelin 0.8.2	<input checked="" type="checkbox"/> Livy 0.6.0
<input type="checkbox"/> JupyterHub 1.0.0	<input type="checkbox"/> Tez 0.9.2	<input type="checkbox"/> Flink 1.9.1
<input type="checkbox"/> Ganglia 3.7.2	<input type="checkbox"/> HBase 1.4.10	<input type="checkbox"/> Pig 0.17.0
<input checked="" type="checkbox"/> Hive 2.3.6	<input type="checkbox"/> Presto 0.227	<input type="checkbox"/> ZooKeeper 3.4.14
<input type="checkbox"/> MXNet 1.5.1	<input type="checkbox"/> Sqoop 1.4.7	<input type="checkbox"/> Mahout 0.13.0
<input type="checkbox"/> Hue 4.4.0	<input type="checkbox"/> Phoenix 4.14.3	<input type="checkbox"/> Oozie 5.1.0
<input checked="" type="checkbox"/> Spark 2.4.4	<input type="checkbox"/> HCatalog 2.3.6	<input type="checkbox"/> TensorFlow 1.14.0

Multiple master nodes (optional)

On the cluster nodes I chose “m5.xlarge” for master and core nodes.

### Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. [Learn more about instance purchasing options](#) ⓘ

ⓘ Console options for automatic scaling have changed. [Learn more](#) ⓘ

Node type	Instance type	Instance count	Purchasing option
<b>Master</b> Master - 1 ⓘ	<b>m5.xlarge</b> ⓘ 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB ⓘ ⓘ Add configuration settings ⓘ	1 Instances	<input checked="" type="radio"/> On-demand ⓘ <input type="radio"/> Spot ⓘ Use on-demand as max price ▼
<b>Core</b> Core - 2 ⓘ	<b>m5.xlarge</b> ⓘ 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB ⓘ ⓘ Add configuration settings ⓘ	<input type="text" value="2"/> Instances	<input checked="" type="radio"/> On-demand ⓘ <input type="radio"/> Spot ⓘ Use on-demand as max price ▼
<b>Task</b> Task - 3 ⓘ	<b>m5.xlarge</b> ⓘ 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB ⓘ ⓘ Add configuration settings ⓘ	<input type="text" value="0"/> Instances	<input checked="" type="radio"/> On-demand ⓘ <input type="radio"/> Spot ⓘ Use on-demand as max price ▼

For bootstrap action, I want the cluster to read from the S3 bucket that I previously set up and uploaded the script to:

### Additional Options

☐ EMRFS consistent view ⓘ

Custom AMI ID **None** ⓘ

#### ▼ Bootstrap Actions

Bootstrap actions are scripts that are executed during setup before Hadoop starts on every cluster node. You can use them to install additional software and customize your applications. [Learn more](#) ⓘ

Bootstrap action type	Name	JAR location	Optional arguments
Custom action	Custom action	s3://emr-spark-notebook-data-challenge/emr_bootstrap.sh	ⓘ ✕

Add bootstrap action **Custom action** ⓘ **Configure and add**

The script contains installation of some python libraries:

```
GNU nano 2.0.6
```

```
File: emr_bootstrap.sh
```

```
#!/bin/bash  
sudo pip3 install -U matplotlib pandas
```

Lastly, I chose the EC2 key pair from previous step for this cluster:

## Security Options

EC2 key pair emr-key 

☒ Cluster visible to all IAM users in account 

## Create Notebook

Finally, I was able to create a Jupyter Notebook using this cluster:


## 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  
Choose data-challenge [j-V2Z2JFEGLC03](#) 

☐ Create a cluster 

**Security groups** ☒ Use default security groups 

☐ Choose security groups (vpc-6258a604)

**AWS service role\*** EMR\_Notebooks\_DefaultRole 

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

☒ Use the default S3 location  
s3://aws-emr-resources-451883747982-us-west-1/notebooks/

☐ Choose an existing S3 location in us-west-1

## Loading Dataset

### Copying from data-engineer-test to native S3

I downloaded the dataset from

<https://s3-us-west-2.amazonaws.com/data-engineer-test/2019-03-26T22%3A49%3A29-file.txt.g>

[z](#) and uploaded via AWS S3 transfer acceleration in 8 files to S3 bucket [emr-spark-notebook-data-challenge](#).

Loading text file as Spark data frame with schema

Each file is then being loaded into a Spark dataframe based on the following schema:

```
root
|-- batch_id: string
|-- vendor_id: string
|-- product_id: string
|-- lab_id: string
|-- state: string
|-- tested_at: string
|-- expires_at: string
|-- thc: float
|-- thca: float
|-- cbd: float
|-- cbda: float
```

I chose to cast both “tested\_at” and “expires\_at” columns to be string type due to the ease of conforming to Athena data type in the next step.

## Data Skewness

Before writing dataframe to Parquet files, I wanted to know if there’s skewness on certain columns. This will hint on what to avoid when choosing partition key(s). A good rule of thumb is to choose a partition key such that the data is more evenly distributed among the values of this column, and the split should result in multiple files each with a decent amount of data (to avoid performance issues when reading from many small files).

Initially I was searching among “state”, “expires\_at” and “tested\_at” columns, from empirical experience. By sampling 1558 rows from the file, I noticed that data mostly cluster around California, Oregon and Washington states. For the “tested\_at” and “expires\_at” columns, each contains small datasets. The data is more evenly distributed when I transform the “expires\_at” column into the day of week for these dates, each day containing between 9% to 18% of data. See [a sample analysis](#) on the distribution of “state”.

## Write data frame to Parquet files

Each data frame is then being written into a partitioned Parquet file under [S3://data-challenge-lab-tests/](#), with partition key “tested\_at”. Saving data frames as Parquet files can drastically improve query performance when we dive deeper into the metrics in later steps. This is a benefit of Parquet being highly compressed (**[calculate compression ratio here](#)**) and is of columnar format. Saving as partitioned Parquet speeds up the writing process (from

80seconds for 100 rows to 9.23 seconds per 1000 row). Total time spent converting from text to parquet for one file (2.9GB) is: \_\_\_\_\_

### Generate Athena table and start querying

Once the parquet files have successfully landed in the “data-challenge-lab-tests” bucket, I first check if the partition key value pair are correct. (side note: made a mistake here when calling a different partition name, “tested\_at\_date” and was seeing “Partitions not in metastore” error message. This is due to the dynamic partitioning feature of Athena. I needed to assign the partition key exactly the same way as in the bucket key name:

```
s3://data-challenge-lab-tests/tested_at=****
```

... before running “MSCK REPAIR TABLE table\_name”. The partitions should be identified automatically.

Once I fixed the typo, and ran a “CREATE EXTERNAL TABLE” statement with the correct schema, I was able to query from this table on Athena. Details on the statements can be found in the [Load parquet to Athena](#) section.

## Analysis on Key Metrics

1. Which 5 vendors have the most products?
2. Which 5 vendors have the fewest products?
3. Which 5 products have the highest potency?
4. Which 5 products have the lowest potency?
5. Which 5 labs have the highest accuracy?
6. Which 5 labs have the lowest accuracy?
7. Which 5 states have the most products and how many?
8. Which 5 states have the fewest products and how few?
9. How many tests are performed each day of the week?

## Discussion of Methods

## Appendix

### Sample payload

```
[{"batch_id":"e3c467b0-e8c2-46dc-9b1d-af6e6358e4d7","vendor_id":"351e6301-7cb1-4418-bc20-5512e306ab3a","product_id":"8e0fa9b8-a81f-4ce9-853e-44ac940f2d74","lab_id":"4dcca0bc-3eb2-431e
```

```
-9d96-c5d10a400ecd","state":"California","tested_at":"2017-03-26","expires_at":"2017-05-07","thc":1.42,"thca":18.67,"cbd":4.81,"cbda":0.69}',
{'batch_id':"074e74c2-34c0-4e0c-b937-f3e5fbf61bce","vendor_id":"2de0163f-ae73-4488-a46e-983932e45d45","product_id":"3d83253d-dfab-4249-87f7-ed964cc2b87d","lab_id":"51b4d95e-7d96-4a43-a981-3efe3bc8678a","state":"Oregon","tested_at":"2017-03-26","expires_at":"2017-04-26","thc":2.09,"thca":12.25,"cbd":1.64,"cbda":0.34}',
{'batch_id':"ca73dcdd-4760-4987-ade1-a3eb104e00a8","vendor_id":"3ecdbc3f-d785-43ff-9bf2-0d312175a626","product_id":"858fcb78-fd2b-4b78-ada1-26d41dae36c5","lab_id":"d28b08cc-68cf-4025-a957-7a81b9fea1a4","state":"California","tested_at":"2017-03-26","expires_at":"2017-05-18","thc":0.68,"thca":16.02,"cbd":0.61,"cbda":0.61}',
{'batch_id':"0c0fd3a5-f97e-4f8d-aadf-3c6f690f06b2","vendor_id":"90d4212a-7ac4-4984-a21a-82dd87a88727","product_id":"38a3e9aa-8112-4dea-9f0c-78a78c86897f","lab_id":"b1f4b2db-9af5-437d-be30-2f257b30441a","state":"California","tested_at":"2017-03-26","expires_at":"2017-05-18","thc":0.94,"thca":10.21,"cbd":3.35,"cbda":0.45}',
{'batch_id':"48c2fc6d-5377-4592-ac0e-9989078219e8","vendor_id":"513a8d3f-6630-4168-9df1-02638fbc08fe","product_id":"8f1d0a59-d1a7-4afb-92da-72db9a8ca1c5","lab_id":"c322c929-c3af-4357-a67c-721565d6665b","state":"California","tested_at":"2017-03-26","expires_at":"2017-05-12","thc":1.18,"thca":17.21,"cbd":0.42,"cbda":0.27}',
{'batch_id':"8a54069b-0404-4d0c-a469-1d2b6f6b4548","vendor_id":"980a1fc0-e952-443a-a624-75d47fe65e84","product_id":"9e96e470-7fdd-4445-b427-ded441f427ca","lab_id":"48e9f9e6-9e92-47a3-a811-e5e0f300de1d","state":"Alabama","tested_at":"2017-03-26","expires_at":"2017-05-13","thc":1.17,"thca":17.06,"cbd":4.36,"cbda":0.96}',
{'batch_id':"c6dcd6ce-aca4-4765-888b-4cde4f3c90d6","vendor_id":"2f320cbb-db0b-4d04-8201-c58a43fe7c31","product_id":"e9fb30b2-841f-4b50-875b-869ca2746c1b","lab_id":"0a6233a3-cf46-4e83-8b6f-d5e2482dbdc1","state":"California","tested_at":"2017-03-26","expires_at":"2017-05-22","thc":1.92,"thca":23.54,"cbd":1.03,"cbda":0.32}',
{'batch_id':"5c4706aa-0a7b-443f-8f6c-4c9e6e73e12e","vendor_id":"6c476224-32da-4789-a5b4-0c79b5251d13","product_id":"18f5bb57-a26a-425f-b640-6dd06e0bb249","lab_id":"fbc86fc2-5a44-4c7d-b257-230e5f0bc640","state":"Colorado","tested_at":"2017-03-26","expires_at":"2017-05-05","thc":1.37,"thca":16.43,"cbd":1.3,"cbda":0.5}',
{'batch_id':"8be7ebfd-d86e-4897-9b32-23b7d3a5e929","vendor_id":"a952ac8a-861d-4a3a-bc2d-23db361aa6d0","product_id":"98f11f44-89a8-4cd8-9996-8df767d4bf12","lab_id":"65508b12-0120-4b97-bea8-94e9c5d381ae","state":"California","tested_at":"2017-03-26","expires_at":"2017-05-11","thc":0.2,"thca":11.88,"cbd":1.33,"cbda":0.64}',
{'batch_id':"78f841df-5163-4020-842f-eb042f44108b","vendor_id":"0ff58138-fd55-41ec-80b8-815834205b6c","product_id":"b5a5d063-9b84-4455-9849-505691a4d3f3","lab_id":"7f5fd41d-389c-458c-97c9-a9bda7f1c922","state":"California","tested_at":"2017-03-26","expires_at":"2017-04-29","thc":2.15,"thca":22.47,"cbd":3.4,"cbda":0.67}]
```

## EMR step

```
aws emr add-steps --cluster-id j-2D2QE4XK0AM80 --steps Type=Spark,Name="Spark
Program",Args=[--deploy-mode,cluster,--master,yarn,--conf,spark.yarn.submit.waitAppCompl
etion=true,--num-executors,5,--executor-cores,5,s3a://emr-spark-notebook-data-challenge/s
cript.py],ActionOnFailure=CONTINUE
```



## Load parquet to Athena

```
CREATE EXTERNAL TABLE IF NOT EXISTS lab_test (  
    batch_id string,  
    vendor_id string,  
    product_id string,  
    lab_id string,  
    state string,  
    tested_at date,  
    expires_at date,  
    thc double,  
    thca double,  
    cbd double,  
    cbda double  
)  
  
PARTITIONED BY (year STRING)  
  
STORED AS PARQUET  
  
LOCATION 's3://data-challenge-lab-tests/part1.parquet/'  
  
tblproperties ("parquet.compression"="SNAPPY");
```

```
MSCK REPAIR TABLE lab_test;
```

## AWS S3 CLI

```
aws s3 cp file1.txt.gz s3://emr-spark-notebook-data-challenge  
/file1.txt.gz --region us-west-1 --endpoint-url https://s3-accelerate.amazonaws.com
```

## Distribution of partition keys

state	COUNTUNIQUE of lab_id
Connecticut	1
Kentucky	1
Montana	1
Nevada	1
North Carolina	1
Pennsylvania	1
Virginia	1
Arizona	2
Florida	2
Hawaii	2
Idaho	2
Louisiana	2
Maine	2
Maryland	2
Massachusetts	2
Mississippi	2
Missouri	2
Nebraska	2
New Hampshire	2
New Jersey	2
New Mexico	2
North Dakota	2
Oklahoma	2
Rhode Island	2
South Carolina	2
South Dakota	2
Tennessee	2
Texas	2
West Virginia	2
Wyoming	2
Alabama	3
Alaska	3
Arkansas	3

Delaware	3
Georgia	3
Illinois	3
Indiana	3
Iowa	3
Kansas	3
Michigan	3
Minnesota	3
New York	3
Ohio	3
Utah	3
Vermont	3
Wisconsin	3
Colorado	9
Oregon	11
Washington	16
California	122
Grand Total	259

Day of Week	SUM of COUNTA of batch_id	Percentage
1	193	0.1238767651
2	249	0.1598202824
3	231	0.148267009
4	232	0.1489088575
5	236	0.1514762516
6	278	0.1784338896
7	139	0.0892169448
Grand Total	1558	1