RECIPE RECOMMENDER ASSIGNMENT

By
Himanshu Gorey
Purnima N S
Shilpa Gururaj

ABSTRACT

To use Spark on Elastic Map Reduce service (EMR) - in AWS and create a recommender with EDA on the given dataset.

PROJECT GOALS

GOAL 1

To understand the use of AWS EMR cluster, S3 bucket utilization and its working, with hands on practice on the same.

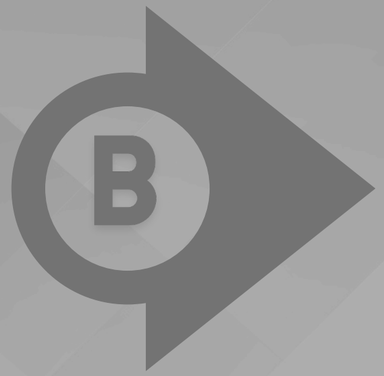
GOAL 2

Create and develope a project, and perform EDA on it to extract insughtful results on performance improvements of the company.

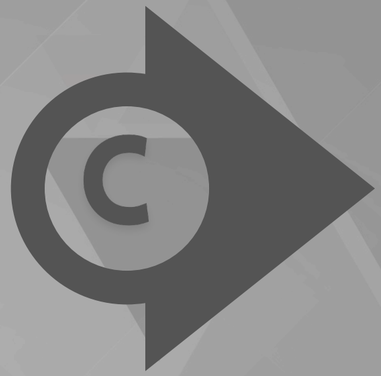
METHODOLOGY

The stepwise actions performed for the project.

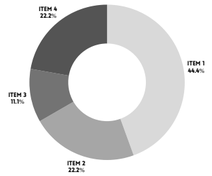
Obtaining dataset, and creation of S3 buckets and EMR cluster as per recommendations.



Creation of a spark session with python [pySpark] and reading the data.



Feature extraction and assertion check for the sets and reupload the processed data into a paraquet file.



DATA DOWNLOADING

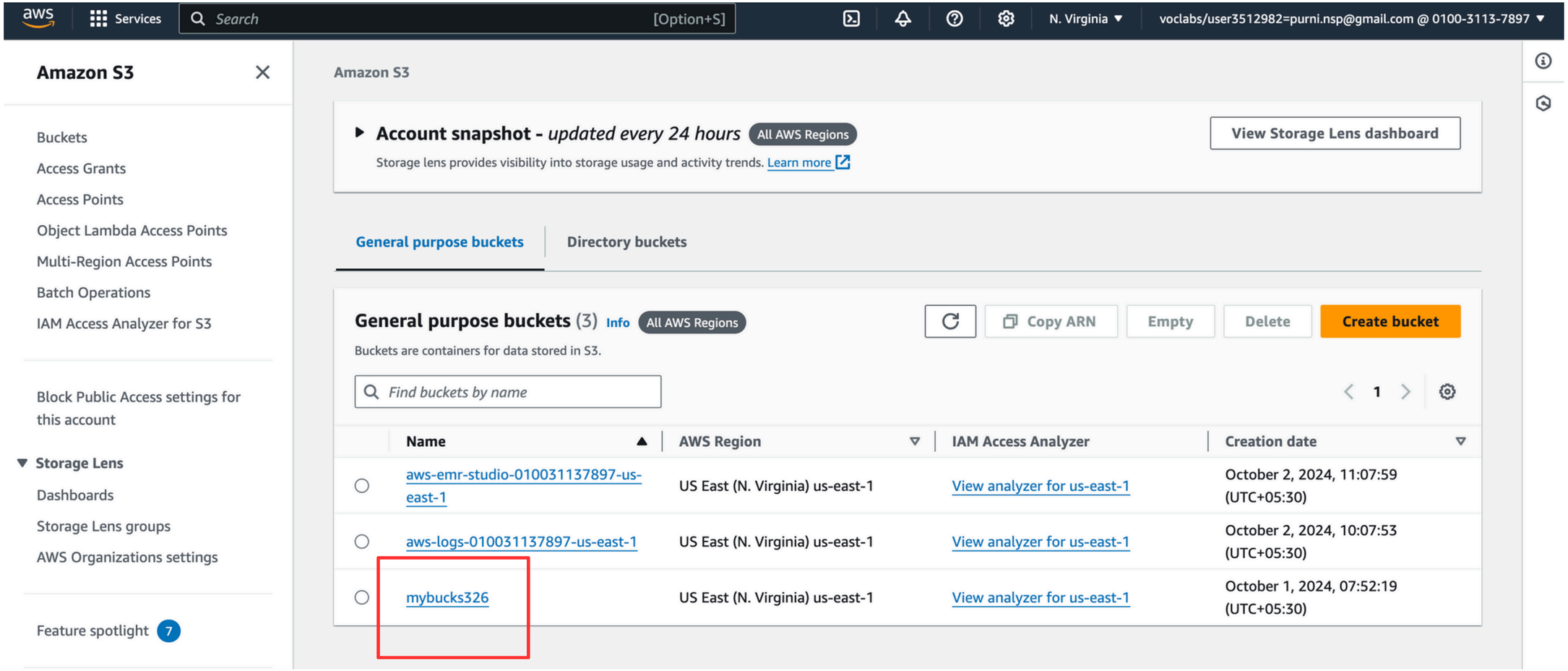
Downloading the below CSV files from the links shared in problem statement.

Raw Receipes Data
https://raw-recipes-clean-upgrad.s3.amazonaws.com/RAW _ recipes cleaned.csv

#Raw Interactions Data
https://raw-interactions- upgrad.s3.amazonaws.com/RAW interactions cleaned.csv

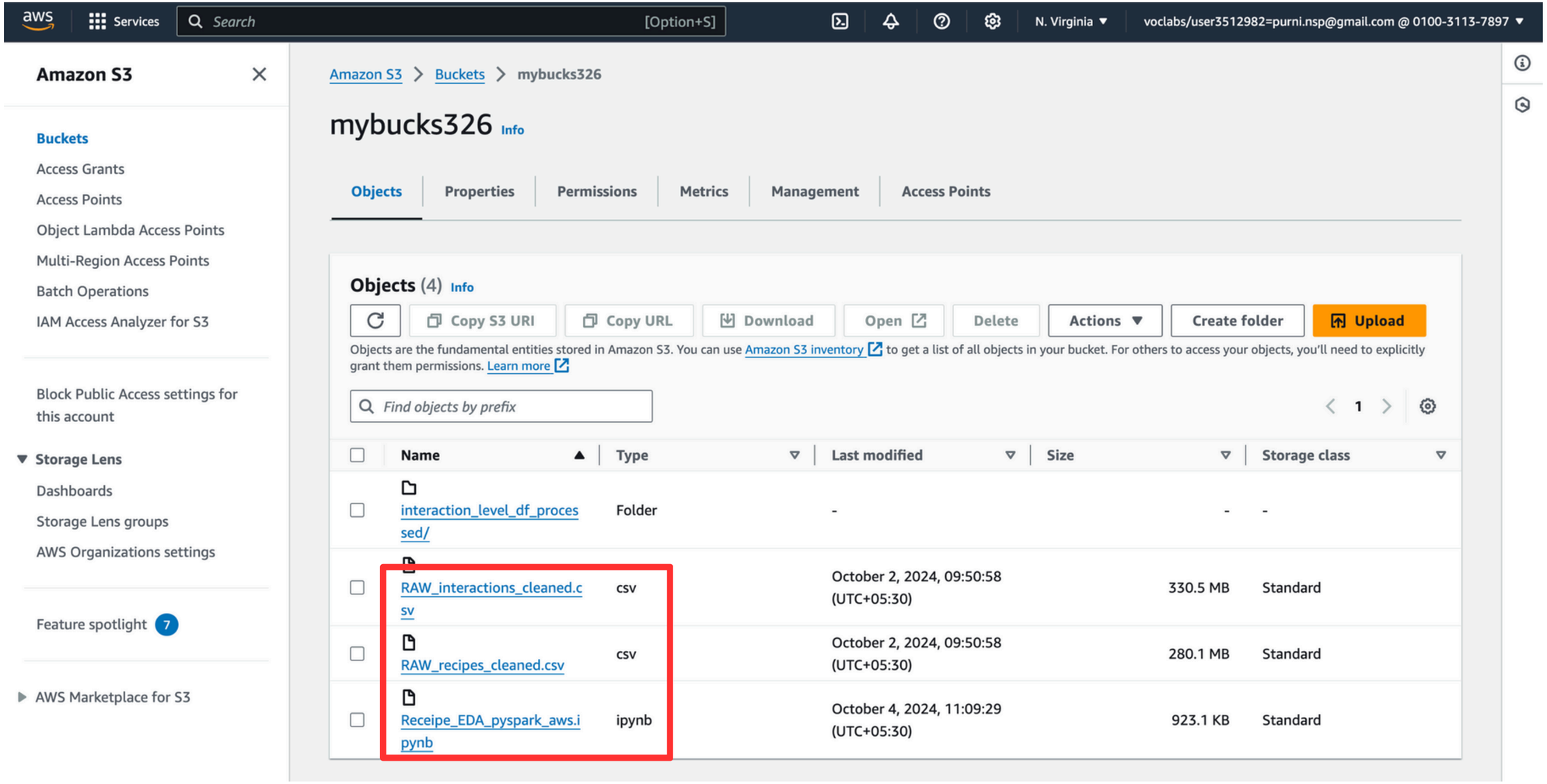
CREATING S3 BUCKET AND EMR CLUSTER IN AWS

Create a S3 bucket in public mode within AWS with given specifications.



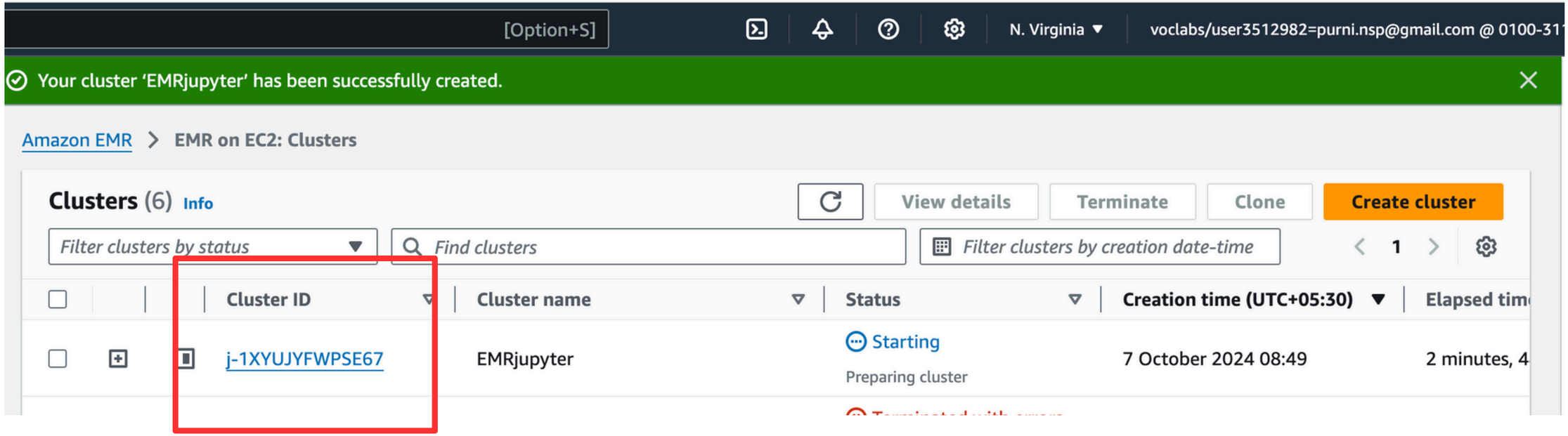
CREATING S3 BUCKET AND EMR CLUSTER IN AWS

Within the s3 bucket upload the following datafiles which has been gathered and the spark jupyter noteboot.



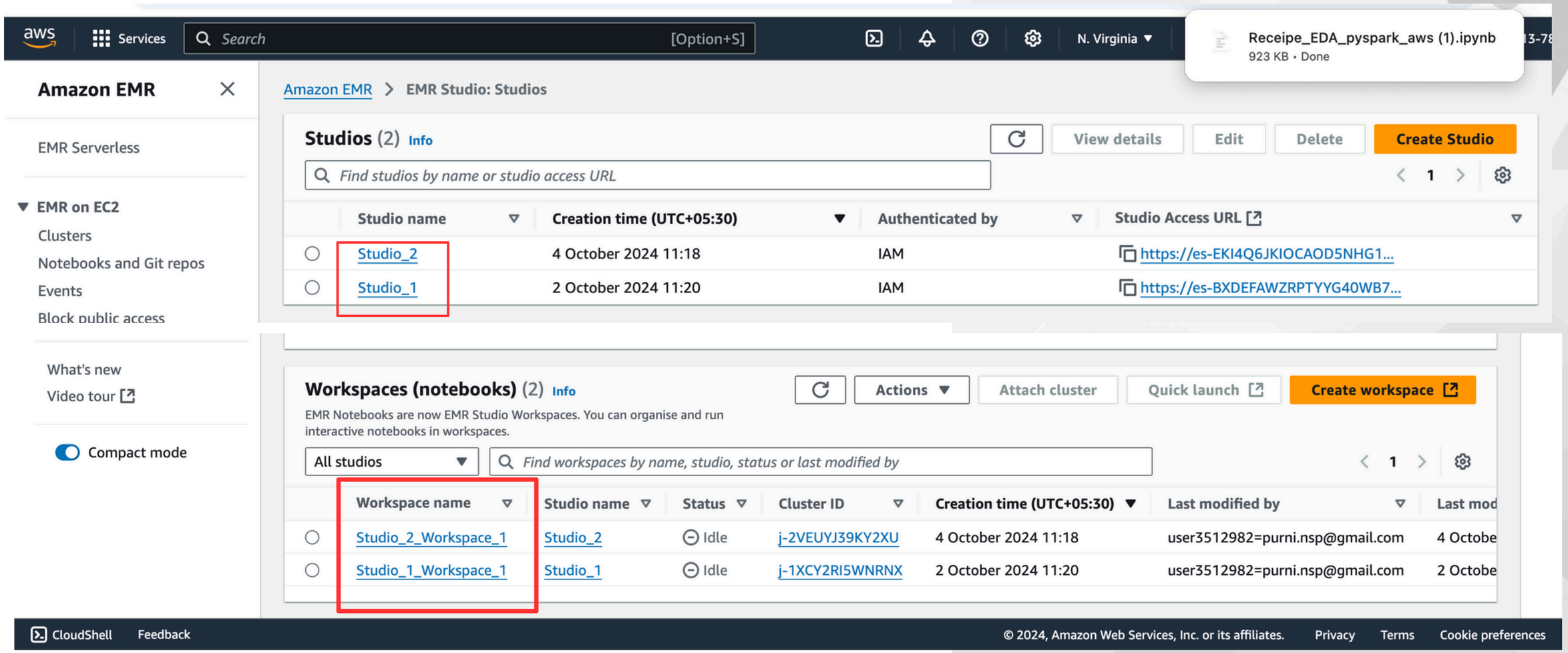
CREATING S3 BUCKET AND EMR CLUSTER IN AWS

Now that the dataset is ready, we will create an EMR cluster with m4.xlarge task, node and processing.



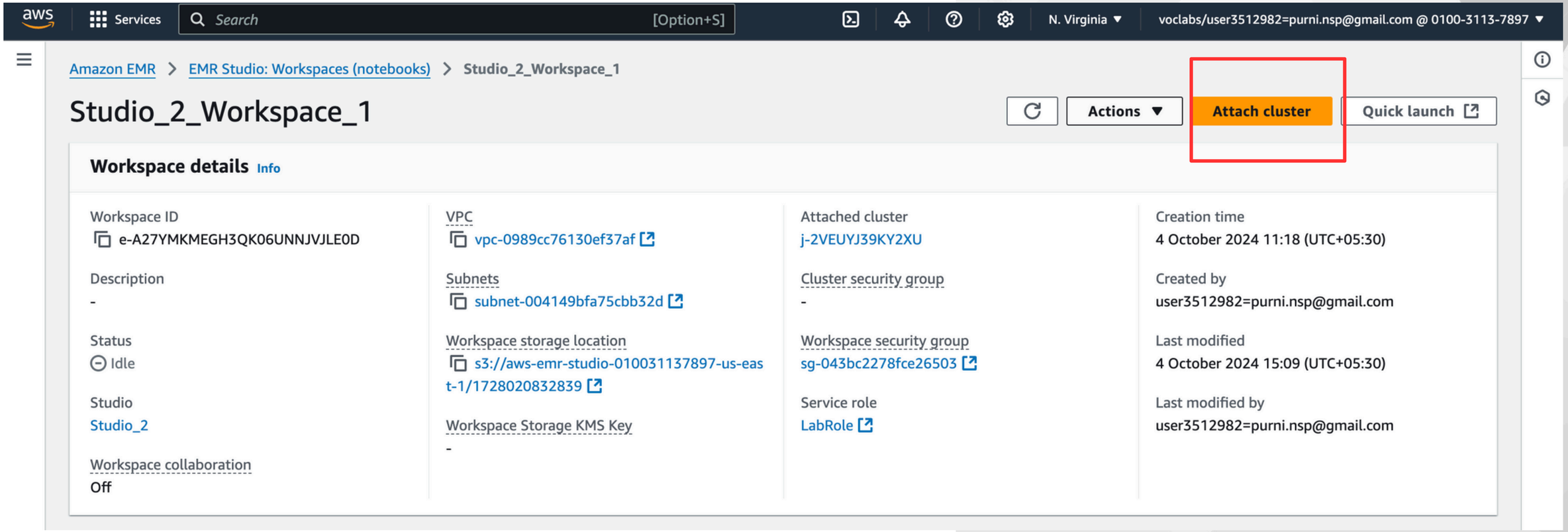
CREATING A SPARK SESSION

Create an EMR studio and a workspace to work with the data and connect it to your cluster.



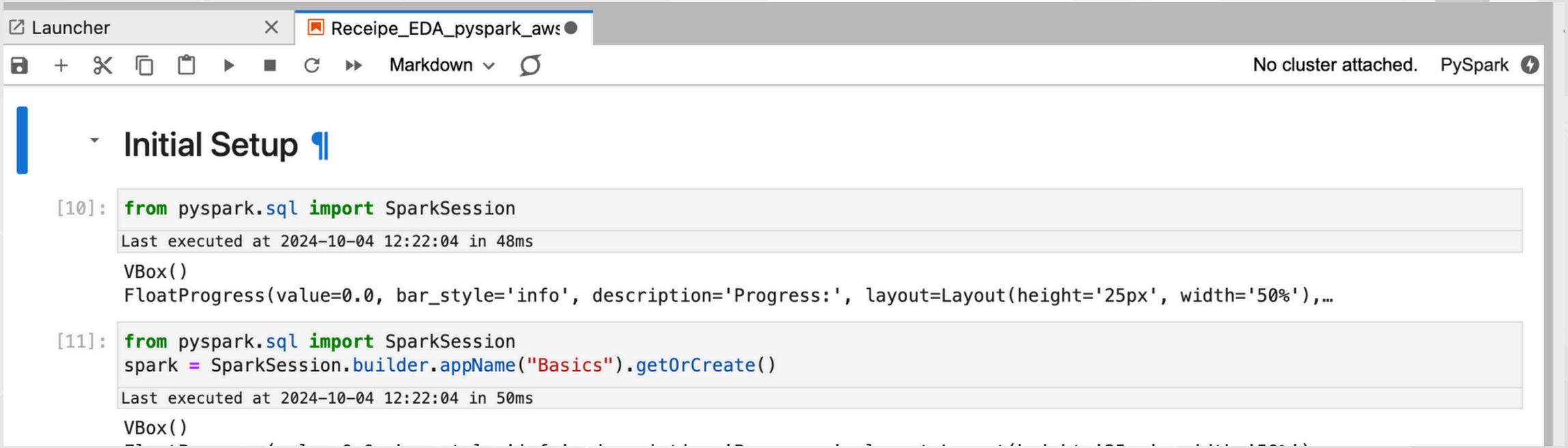
CREATING A SPARK SESSION

Now that a studio and workspace has been created, we have to launch the workspace either with attached or quick launch and then attach a running cluster.



PYSPARK KERNET IN JUPYTER NOTEBOOK

A spark session with a name “Basics” has been created.



READING THE DATASETS

A spark session with a name “Basics” has been created.

[*]:
```
# Task 01 Cell 1 out of 1

raw_recipes_df = (spark.read.csv("s3://mybucks326/RAW_recipes_cleaned.csv", header = True, inferSchema=True,
                  # argument 1, Add an argument to communicate to the compiler that there is a header in the raw data.
                  # argument 2, Add an argument to ask the complier to estimate the data types for all columns.
                              ))

# Please forward the exact name of data frames and columns as suggested in the code.
# It will ensure that the assert commands function correctly.
```

[*]:
```
raw_recipes_df.show(5)
```

[8]:
```
raw_recipes_df.printSchema()
```

Last executed at 2024-10-07 09:23:55 in 63ms

```
root
 |-- name: string (nullable = true)
 |-- id: integer (nullable = true)
 |-- minutes: integer (nullable = true)
 |-- contributor_id: integer (nullable = true)
 |-- submitted: date (nullable = true)
 |-- tags: string (nullable = true)
 |-- nutrition: string (nullable = true)
 |-- n_steps: integer (nullable = true)
 |-- steps: string (nullable = true)
 |-- description: string (nullable = true)
 |-- ingredients: string (nullable = true)
 |-- n_ingredients: integer (nullable = true)
```

EXTRACTING FEATURES & ASSERT CHECKS

Features has been extracted like :

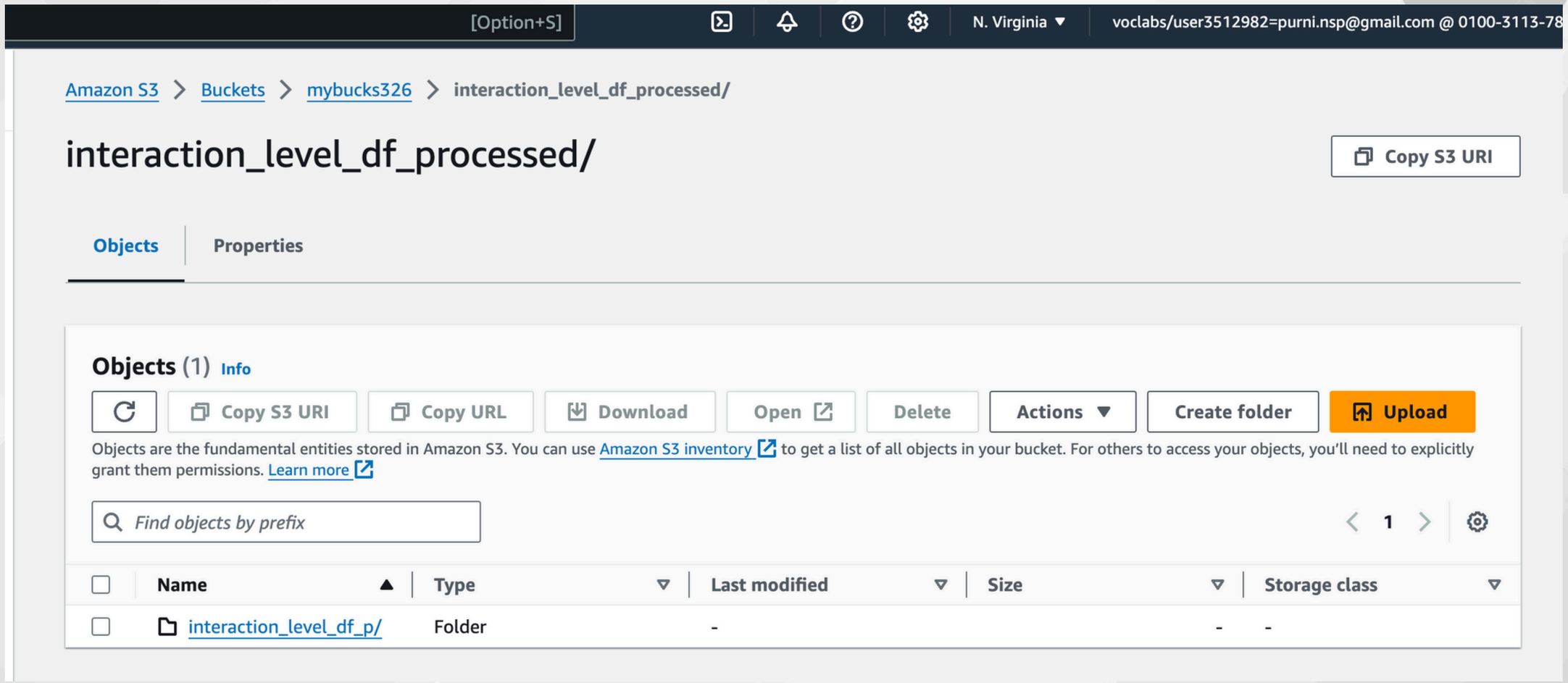
- Nutrition-per-100 calorie columns
- Create time-based features:
  - days_since_submission_on_review_date
  - months_since_submission_on_review_date
  - years_since_submission_on_review_date

16]:
```
# Task 02 Cell 1 out of 2
# 2.1 - string operations to remove square brakets

raw_recipes_df = (raw_recipes_df
                  .withColumn('nutrition',(F.regexp_replace("nutrition","[\[\]]",""))
                              # add code to remove square brackets
                              # pyspark function to replace string characters
                              ))
```
Last executed at 2024-10-07 09:24:04 in 272ms

17]:
```
# Task 02 Cell 2 out of 3
# STEP 2.2 - split the neutrition string into seven individial values.
# Create an object to split the nutrition column
import pyspark
nutrition_cols_split = pyspark.sql.functions.split(raw_recipes_df['nutrition'],',')

# Write a loop to extract individual values from the nutrition column

for col_index, col_name in enumerate(nutrition_column_names):
    # col_index holds the index number of each column, e.g., calories will be 0
    # col_name holds the name of each column

    raw_recipes_df = (raw_recipes_df.withColumn(col_name, nutrition_cols_split.getItem(col_index).cast("float")
                                        # pyspark function to extract individual values from the nutrition_cols_split object
                                        # You can also cast the extracted value to floats in the same code.
                                              ))
```
Last executed at 2024-10-07 09:24:05 in 267ms

UPLOADING TO A PARQUET FILE

The Final file has been uploaded in the parquet format in the S3 bucket which can be used for futher EDA process.



CONCLUSION

The dataset has been hence analysed to extract the features and provide analysis on the recommendation of receipes.

S3 BUCKET	EMR CLUSTER	PARAQUET FILE
Data extraction, creation of S3 bucket. Upload the data to the s3 bucket. S3 is in public mode for visibility.	Creation of emr cluster with given specialization, and creation of studio and workspace and attaching it with cluster.	Once processing the file is done, feature extraction is derived, we upload the processed data to a paraquet file into the same s3 bucket

THANK YOU