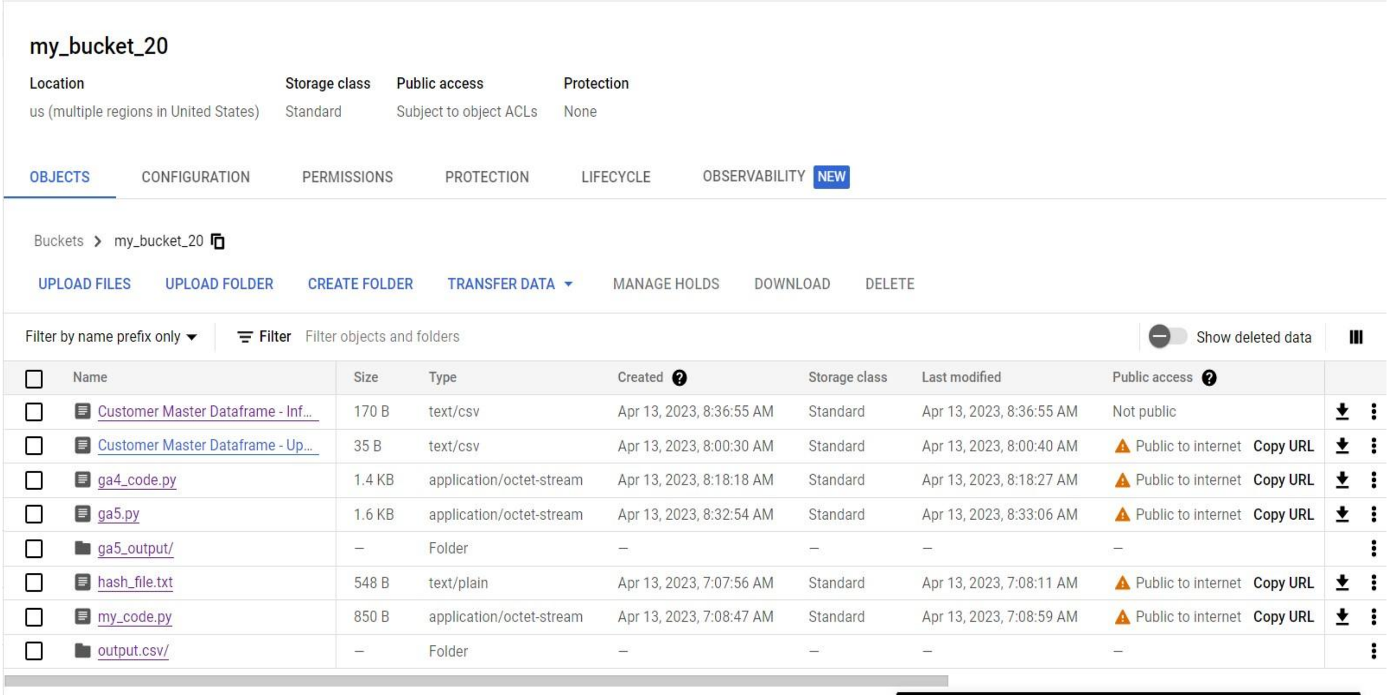
1. Moving files to bucket

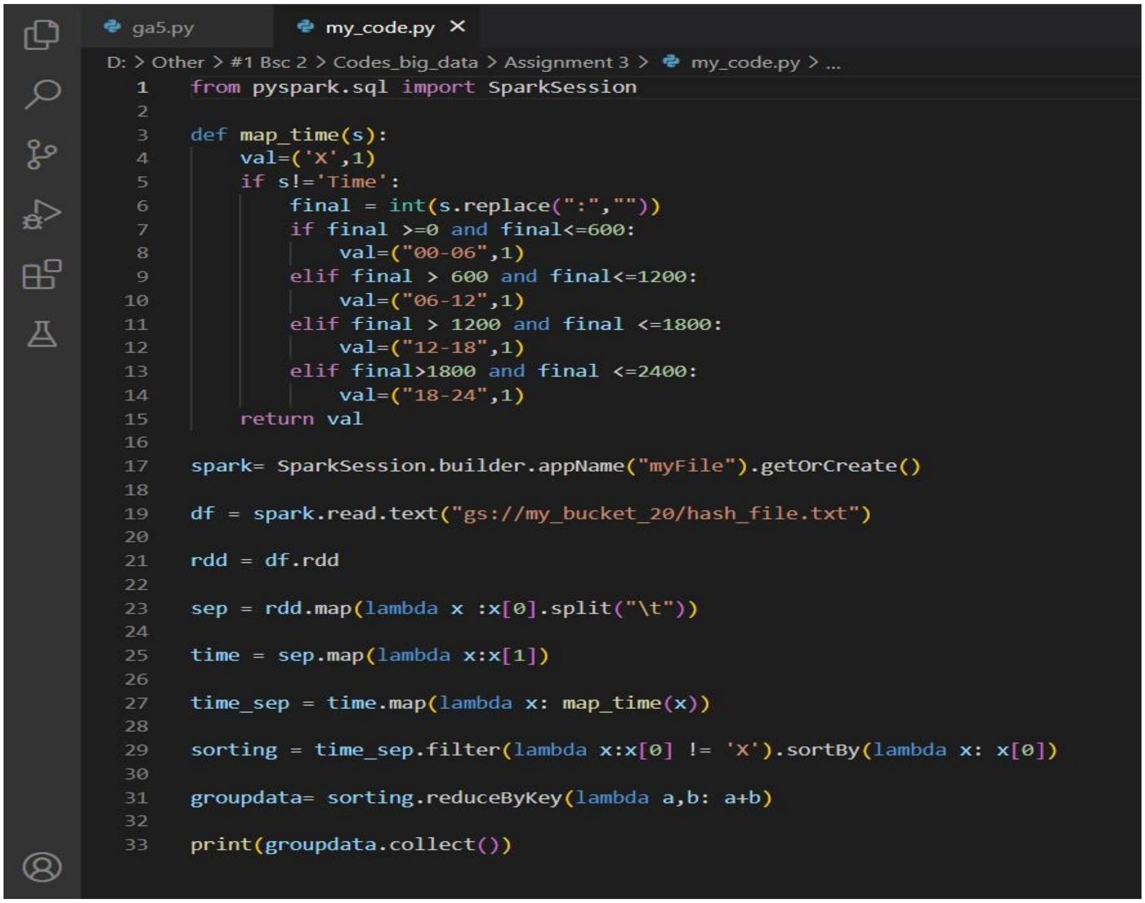


my_bucket_20

Location	Storage class	Public access	Protection
us (multiple regions in United States)	Standard	Subject to object ACLs	None

☐	Name	Size	Type	Created	Storage class	Last modified	Public access
☐	Customer Master Dataframe - Inf...	170 B	text/csv	Apr 13, 2023, 8:36:55 AM	Standard	Apr 13, 2023, 8:36:55 AM	Not public
☐	Customer Master Dataframe - Up...	35 B	text/csv	Apr 13, 2023, 8:00:30 AM	Standard	Apr 13, 2023, 8:00:40 AM	⚠ Public to internet Copy URL
☐	ga4_code.py	1.4 KB	application/octet-stream	Apr 13, 2023, 8:18:18 AM	Standard	Apr 13, 2023, 8:18:27 AM	⚠ Public to internet Copy URL
☐	ga5.py	1.6 KB	application/octet-stream	Apr 13, 2023, 8:32:54 AM	Standard	Apr 13, 2023, 8:33:06 AM	⚠ Public to internet Copy URL
☐	ga5_output/	–	Folder	–	–	–	–
☐	hash_file.txt	548 B	text/plain	Apr 13, 2023, 7:07:56 AM	Standard	Apr 13, 2023, 7:08:11 AM	⚠ Public to internet Copy URL
☐	my_code.py	850 B	application/octet-stream	Apr 13, 2023, 7:08:47 AM	Standard	Apr 13, 2023, 7:08:59 AM	⚠ Public to internet Copy URL
☐	output.csv/	–	Folder	–	–	–	–

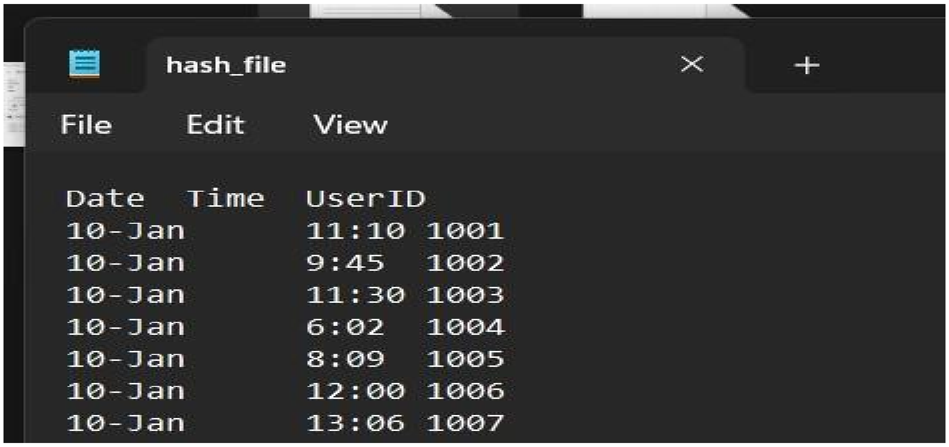
2. Creating Code



ga5.py
my_code.py
D: > Other > #1 Bsc 2 > Codes_big_data > Assignment 3 > my_code.py > ...

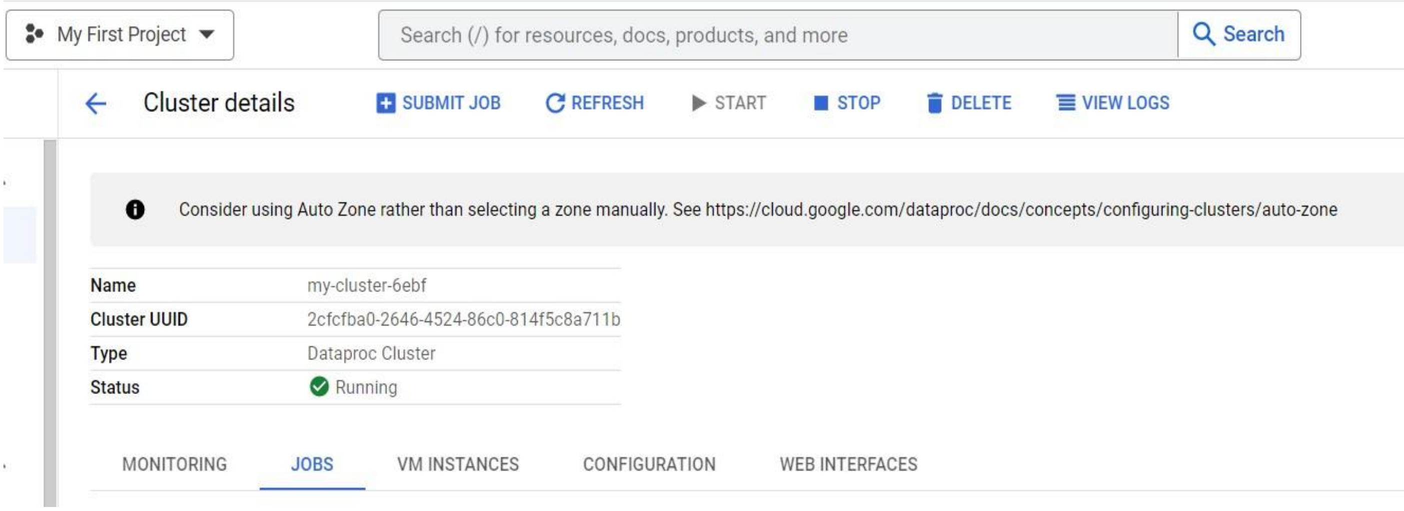
```python
from pyspark.sql import SparkSession

def map_time(s):
    val=('X',1)
    if s!='Time':
        final = int(s.replace(":",""))
        if final >=0 and final<=600:
            val=("00-06",1)
        elif final > 600 and final<=1200:
            val=("06-12",1)
        elif final > 1200 and final <=1800:
            val=("12-18",1)
        elif final>1800 and final <=2400:
            val=("18-24",1)
    return val

spark= SparkSession.builder.appName("myFile").getOrCreate()

df = spark.read.text("gs://my_bucket_20/hash_file.txt")

rdd = df.rdd

sep = rdd.map(lambda x :x[0].split("\t"))

time = sep.map(lambda x:x[1])

time_sep = time.map(lambda x: map_time(x))

sorting = time_sep.filter(lambda x:x[0] != 'X').sortBy(lambda x: x[0])

groupdata= sorting.reduceByKey(lambda a,b: a+b)

print(groupdata.collect())
```

3. The TEXT File



```
Date  Time  UserID
10-Jan      11:10 1001
10-Jan      9:45  1002
10-Jan      11:30 1003
10-Jan      6:02  1004
10-Jan      8:09  1005
10-Jan      12:00 1006
10-Jan      13:06 1007
```

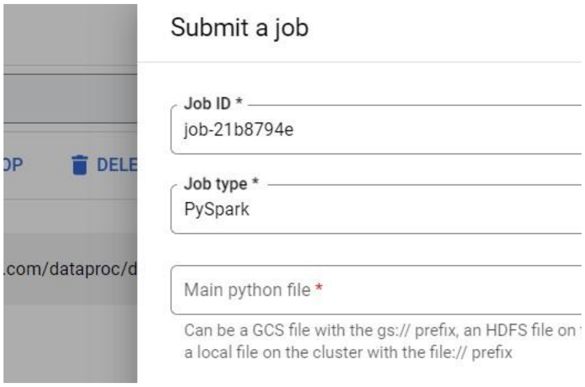
4. Creating Clusters



Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone

Name	my-cluster-6ebf
Cluster UUID	2cfcfba0-2646-4524-86c0-814f5c8a711b
Type	Dataproc Cluster
Status	✓ Running

5. Creating Job



Submit a job

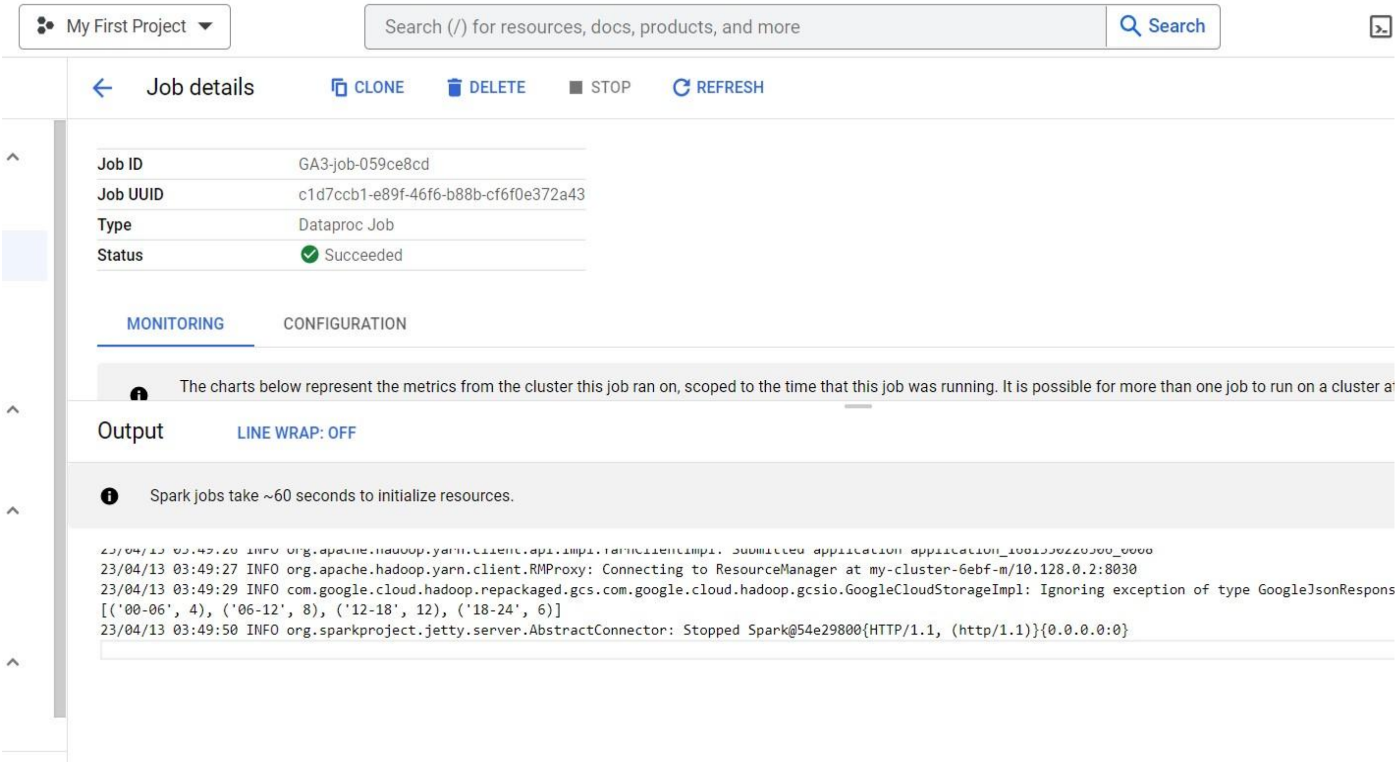
Job ID *
job-21b8794e

Job type *
PySpark

Main python file *

Can be a GCS file with the gs:// prefix, an HDFS file on
a local file on the cluster with the file:// prefix

6. The final Output



Job ID	GA3-job-059ce8cd
Job UUID	c1d7ccb1-e89f-46f6-b88b-cf6f0e372a43
Type	Dataproc Job
Status	✓ Succeeded

The charts below represent the metrics from the cluster this job ran on, scoped to the time that this job was running. It is possible for more than one job to run on a cluster a

Spark jobs take ~60 seconds to initialize resources.

```
23/04/13 03:49:26 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1681350226306_0008
23/04/13 03:49:27 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at my-cluster-6ebf-m/10.128.0.2:8030
23/04/13 03:49:29 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonRespons
[('00-06', 4), ('06-12', 8), ('12-18', 12), ('18-24', 6)]
23/04/13 03:49:50 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@54e29800{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
```