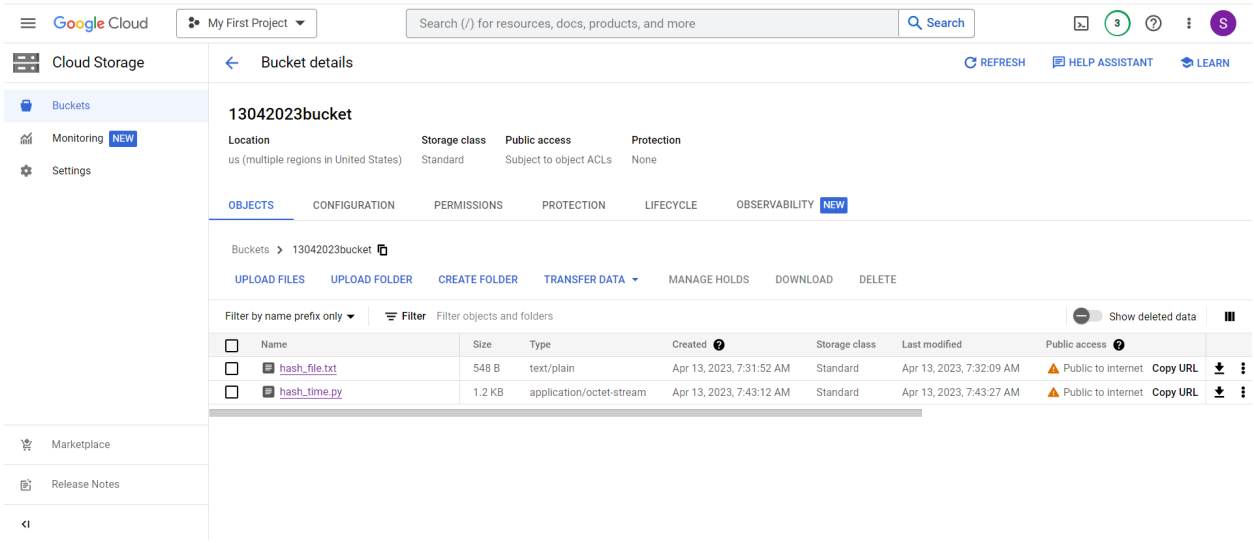
Big Data Graded Assignment 3

❖ Step 1 : Upload all files under bucket with public access

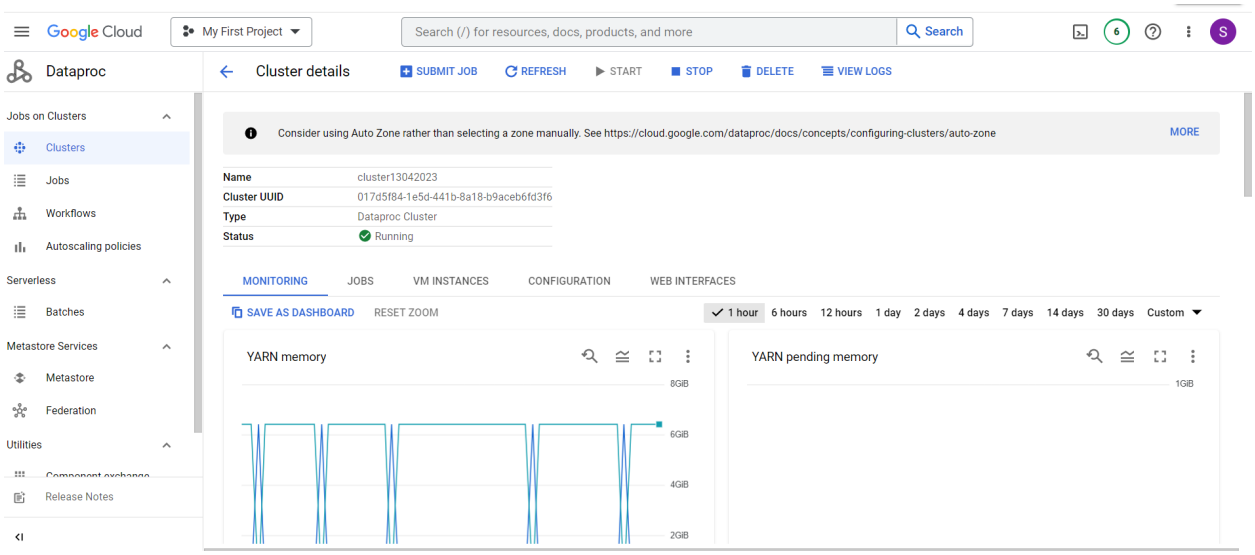


❖ Python File

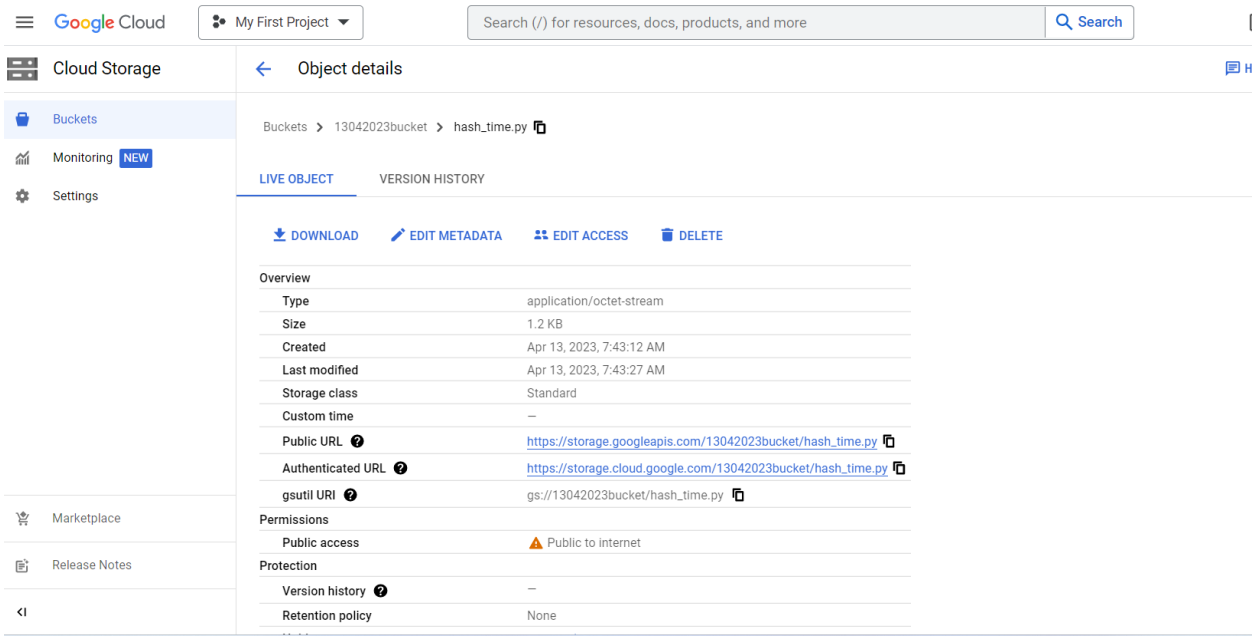
```
Big Data > GA3 > hash_time.py > ...
from pyspark.sql import SparkSession

#Creating spark session
spark= SparkSession.builder.appName("Spark").getOrCreate()
#hash file reading from GCP Bucket with spark command
df = spark.read.text("gs://13042023bucket/hash_file.txt")
#conversion from df to rdd
rdd = df.rdd
#spliting the rdd using tab seperator
sep = rdd.map(lambda x :x[0].split("\t"))
#collecting time column only using lambda function
time = sep.map(lambda x:x[1])
#defining hash_time_map function for mapping
def hash_time_map(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
#hash_time_map function call to collect time
collect_time = time.map(lambda x: hash_time_map(x))
#removing header with sorted time
sorted_time = collect_time.filter(lambda x:x[0] != 'X').sortBy(lambda x: x[0])
#timezone grouping
timezone= sorted_time.reduceByKey(lambda a,b: a+b)
#final output
print(timezone.collect())
```

❖ Create a cluster which is running



❖ Get gsutil url from file



❖ Create & submit a job with correct output