CS4830 – Big Data Laboratory

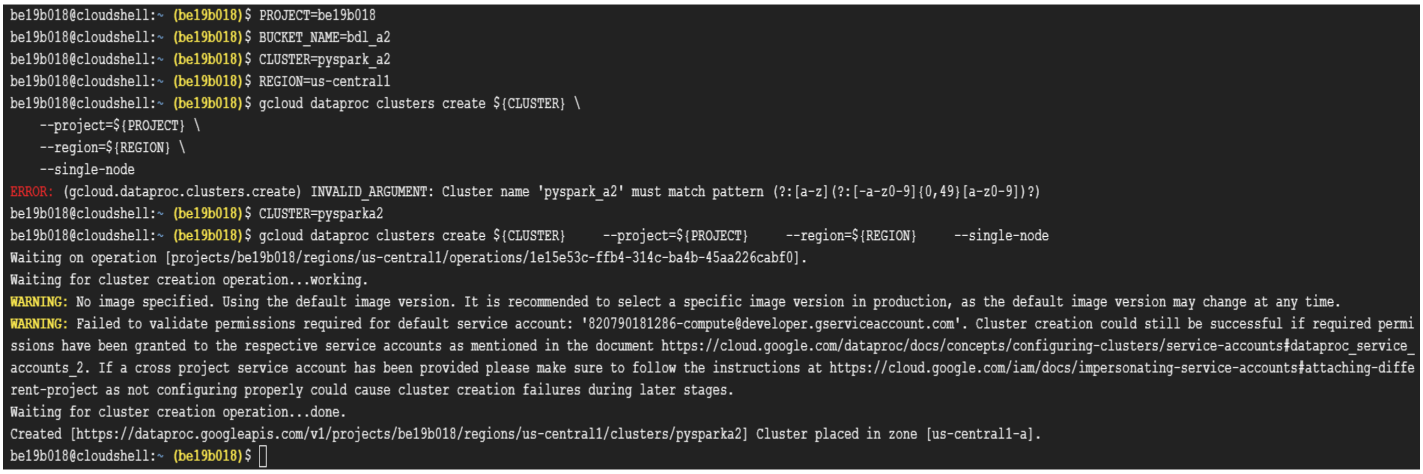
Assignment-2

Ishan Chokshi – BE19B018

January-May 2023

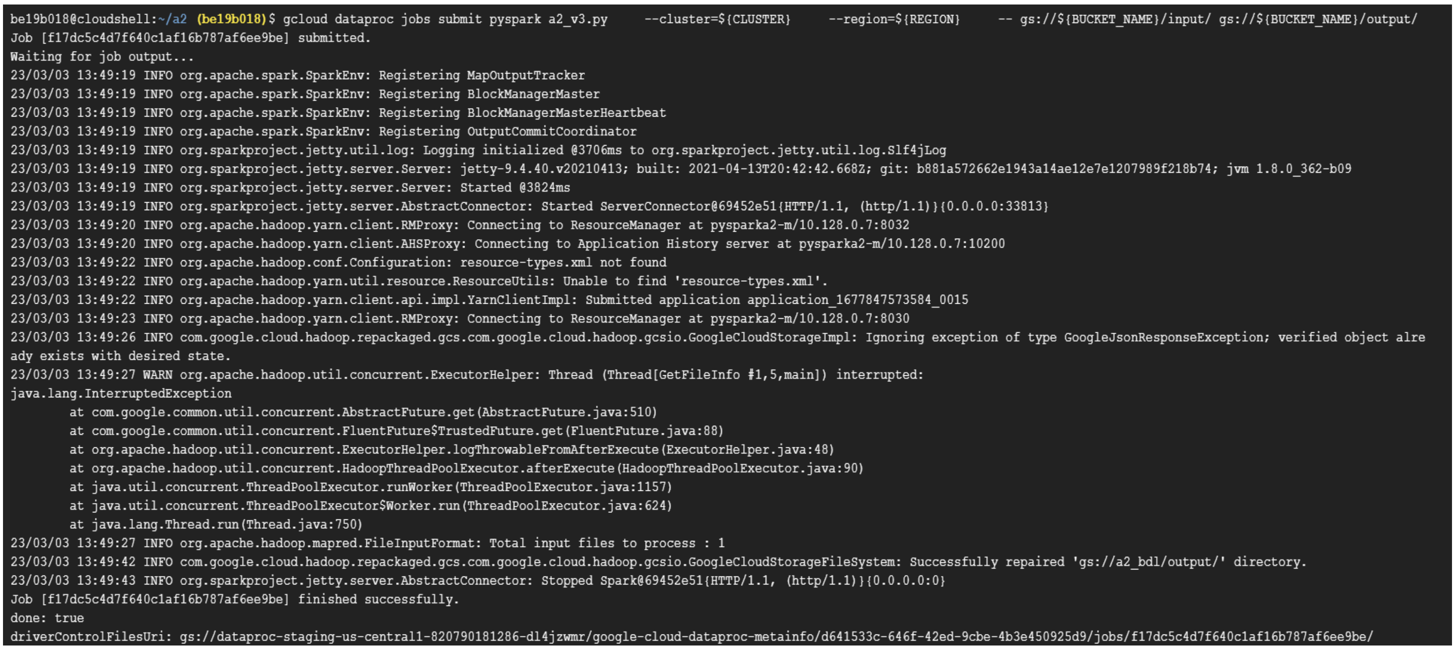
Indian Institute of Technology Madras

1. The python file and output text file are provided in the folder. Outputs and commands used in the cloud shell are attached below:



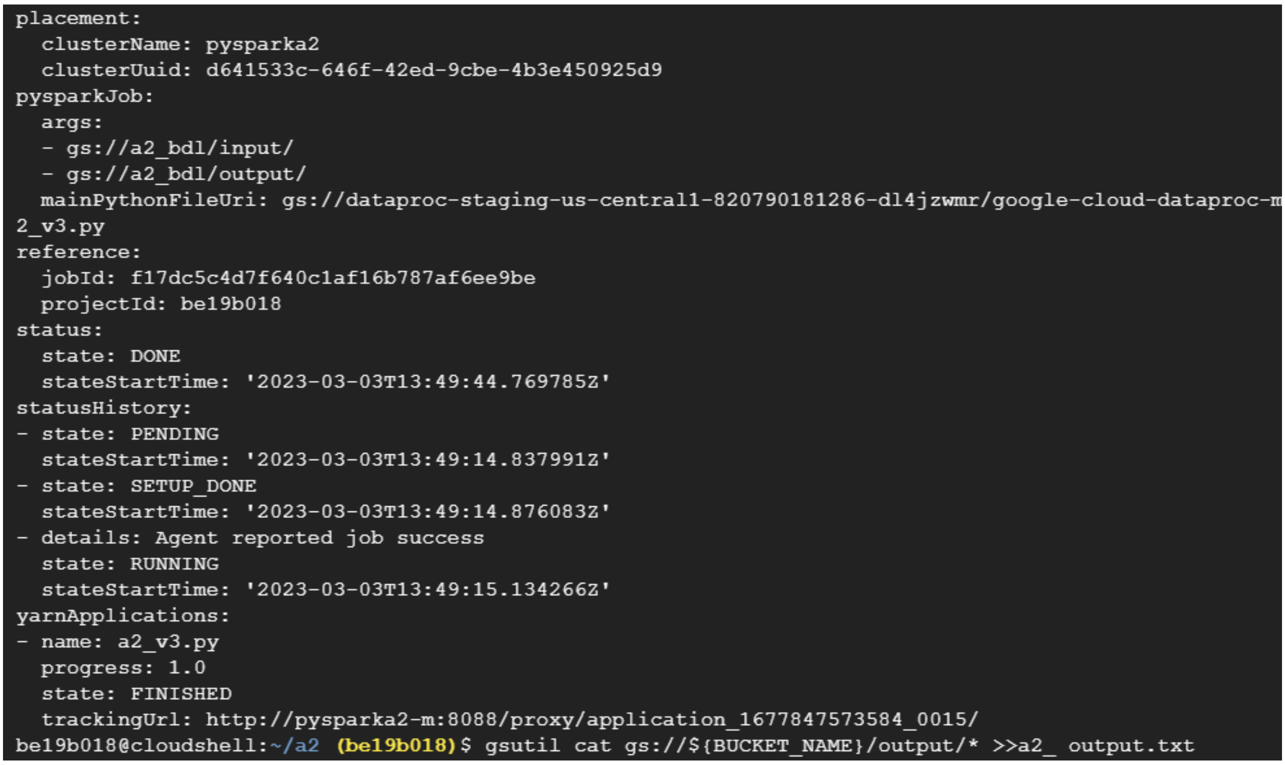
```
be19b018@cloudshell:~ (be19b018)$ PROJECT=be19b018
be19b018@cloudshell:~ (be19b018)$ BUCKET_NAME=bdl_a2
be19b018@cloudshell:~ (be19b018)$ CLUSTER=pyspark_a2
be19b018@cloudshell:~ (be19b018)$ REGION=us-central1
be19b018@cloudshell:~ (be19b018)$ gcloud dataproc clusters create ${CLUSTER} \
    --project=${PROJECT} \
    --region=${REGION} \
    --single-node
ERROR: (gcloud.dataproc.clusters.create) INVALID_ARGUMENT: Cluster name 'pyspark_a2' must match pattern (?:[a-z](?:[-a-z0-9]{0,49}[a-z0-9])?)
be19b018@cloudshell:~ (be19b018)$ CLUSTER=pysparka2
be19b018@cloudshell:~ (be19b018)$ gcloud dataproc clusters create ${CLUSTER}     --project=${PROJECT}     --region=${REGION}     --single-node
Waiting on operation [projects/be19b018/regions/us-central1/operations/1e15e53c-ffb4-314c-ba4b-45aa226cabf0].
Waiting for cluster creation operation...working.
WARNING: No image specified. Using the default image version. It is recommended to select a specific image version in production, as the default image version may change at any time.
WARNING: Failed to validate permissions required for default service account: '820790181286-compute@developer.gserviceaccount.com'. Cluster creation could still be successful if required permi
ssions have been granted to the respective service accounts as mentioned in the document https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/service-accounts#dataproc_service_
accounts_2. If a cross project service account has been provided please make sure to follow the instructions at https://cloud.google.com/iam/docs/impersonating-service-accounts#attaching-diffe
rent-project as not configuring properly could cause cluster creation failures during later stages.
Waiting for cluster creation operation...done.
Created [https://dataproc.googleapis.com/v1/projects/be19b018/regions/us-central1/clusters/pysparka2] Cluster placed in zone [us-central1-a].
be19b018@cloudshell:~ (be19b018)$ 
```

Defining the project, bucket, cluster, and creating the cluster



```
be19b018@cloudshell:~/a2 (be19b018)$ gcloud dataproc jobs submit pyspark a2_v3.py     --cluster=${CLUSTER}     --region=${REGION}     -- gs://${BUCKET_NAME}/input/ gs://${BUCKET_NAME}/output/
Job [f17dc5c4d7f640c1af16b787af6ee9be] submitted.
Waiting for job output...
23/03/03 13:49:19 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
23/03/03 13:49:19 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
23/03/03 13:49:19 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
23/03/03 13:49:19 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
23/03/03 13:49:19 INFO org.sparkproject.jetty.util.log: Logging initialized @3706ms to org.sparkproject.jetty.util.log.Slf4jLog
23/03/03 13:49:19 INFO org.sparkproject.jetty.server.Server: jetty-9.4.40.v20210413; built: 2021-04-13T20:42:42.668Z; git: b881a572662e1943a14ae12e7e1207989f218b74; jvm 1.8.0_362-b09
23/03/03 13:49:19 INFO org.sparkproject.jetty.server.Server: Started @3824ms
23/03/03 13:49:19 INFO org.sparkproject.jetty.server.AbstractConnector: Started ServerConnector@69452e51{HTTP/1.1, (http/1.1)}{0.0.0.0:33813}
23/03/03 13:49:20 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at pysparka2-m/10.128.0.7:8032
23/03/03 13:49:20 INFO org.apache.hadoop.yarn.client.AHSProxy: Connecting to Application History server at pysparka2-m/10.128.0.7:10200
23/03/03 13:49:22 INFO org.apache.hadoop.conf.Configuration: resource-types.xml not found
23/03/03 13:49:22 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/03/03 13:49:22 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1677847573584_0015
23/03/03 13:49:23 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at pysparka2-m/10.128.0.7:8030
23/03/03 13:49:26 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object alre
ady exists with desired state.
23/03/03 13:49:27 WARN org.apache.hadoop.util.concurrent.ExecutorHelper: Thread (Thread[GetFileInfo #1,5,main]) interrupted:
java.lang.InterruptedException
        at com.google.common.util.concurrent.AbstractFuture.get(AbstractFuture.java:510)
        at com.google.common.util.concurrent.FluentFuture$TrustedFuture.get(FluentFuture.java:88)
        at org.apache.hadoop.util.concurrent.ExecutorHelper.logThrowableFromAfterExecute(ExecutorHelper.java:48)
        at org.apache.hadoop.util.concurrent.HadoopThreadPoolExecutor.afterExecute(HadoopThreadPoolExecutor.java:90)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1157)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:750)
23/03/03 13:49:27 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
23/03/03 13:49:42 INFO com.google.cloud.hadoop.repackaged.gcs.com.google.cloud.hadoop.gcsio.GoogleCloudStorageFileSystem: Successfully repaired 'gs://a2_bdl/output/' directory.
23/03/03 13:49:43 INFO org.sparkproject.jetty.server.AbstractConnector: Stopped Spark@69452e51{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [f17dc5c4d7f640c1af16b787af6ee9be] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-central1-820790181286-dl4jzwmr/google-cloud-dataproc-metainfo/d641533c-646f-42ed-9cbe-4b3e450925d9/jobs/f17dc5c4d7f640c1af16b787af6ee9be/
```

Submitting the job



```
placement:
  clusterName: pysparka2
  clusterUuid: d641533c-646f-42ed-9cbe-4b3e450925d9
pysparkJob:
  args:
  - gs://a2_bdl/input/
  - gs://a2_bdl/output/
  mainPythonFileUri: gs://dataproc-staging-us-central1-820790181286-dl4jzwmr/google-cloud-dataproc-m
2_v3.py
reference:
  jobId: f17dc5c4d7f640c1af16b787af6ee9be
  projectId: be19b018
status:
  state: DONE
  stateStartTime: '2023-03-03T13:49:44.769785Z'
statusHistory:
- state: PENDING
  stateStartTime: '2023-03-03T13:49:14.837991Z'
- state: SETUP_DONE
  stateStartTime: '2023-03-03T13:49:14.876083Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2023-03-03T13:49:15.134266Z'
yarnApplications:
- name: a2_v3.py
  progress: 1.0
  state: FINISHED
  trackingUrl: http://pysparka2-m:8088/proxy/application_1677847573584_0015/
be19b018@cloudshell:~/a2 (be19b018)$ gsutil cat gs://${BUCKET_NAME}/output/* >>a2_ output.txt
```

Saving the output to a txt file

2. a. **HDFS:** HDFS (Hadoop Distributed File System) is distributed file system which is part of the Apache Hadoop framework. It enables you to store and manage massive volumes of data across a cluster's many computers (distributed processing), and also provides tools for analysis of massive data. Data in HDFS is divided up into smaller "blocks" and distributed throughout the cluster's nodes. To achieve data redundancy and fault tolerance, each block is copied many times. Due to this even if a node fails, other nodes may still access the data. It enables quick data transfers between computing nodes and offers high-performance access to data across Hadoop clusters. HDFS is frequently used in big data applications where it is necessary to effectively store and analyse enormous volumes of data.

   b. **Hive:** Hive is an open-source data warehouse tool which allows users to search through and analyse data stored in HDFS. Hive operates by converting queries that resemble SQL into MapReduce tasks that may be carried out on a Hadoop cluster. This allows the user to work on large datasets for querying data.

   c. **Pig:** Pig is a high-level data processing language used to analyse datasets stored in Hadoop. By offering a simple to use and comprehend scripting language, it is intended to make developing MapReduce tasks easier. Pig Latin, a sophisticated scripting language, is used in Pig. The language is designed to give flexibility, optimization potential, and programming simplicity. Pig Latin scripts are written by programmers to handle data stored in HDFS, and the Pig Engine internally converts all of these scripts into a particular Map.

d. **YARN:** YARN (Yet Another Resource Negotiator) is a system for managing resources in a Hadoop cluster. It is in charge of allocating resources to the apps operating on the cluster, including CPU and memory. YARN is in charge of scheduling activities to be carried out on various cluster nodes and assigning system resources to the various applications operating in a Hadoop cluster. Applications like MapReduce, Spark, and Hive are supported by YARN. Also, it enables the development of unique application frameworks that may function on the cluster. YARN is a crucial part of Hadoop as it increases the capacity of Hadoop to handle a variety of applications beyond only MapReduce. Due to this reason, Hadoop can accommodate a wide variety of workloads.