Big Data Wrangling with Google Books Ngrams: A Project Report

Author: Amirhossein Kiani

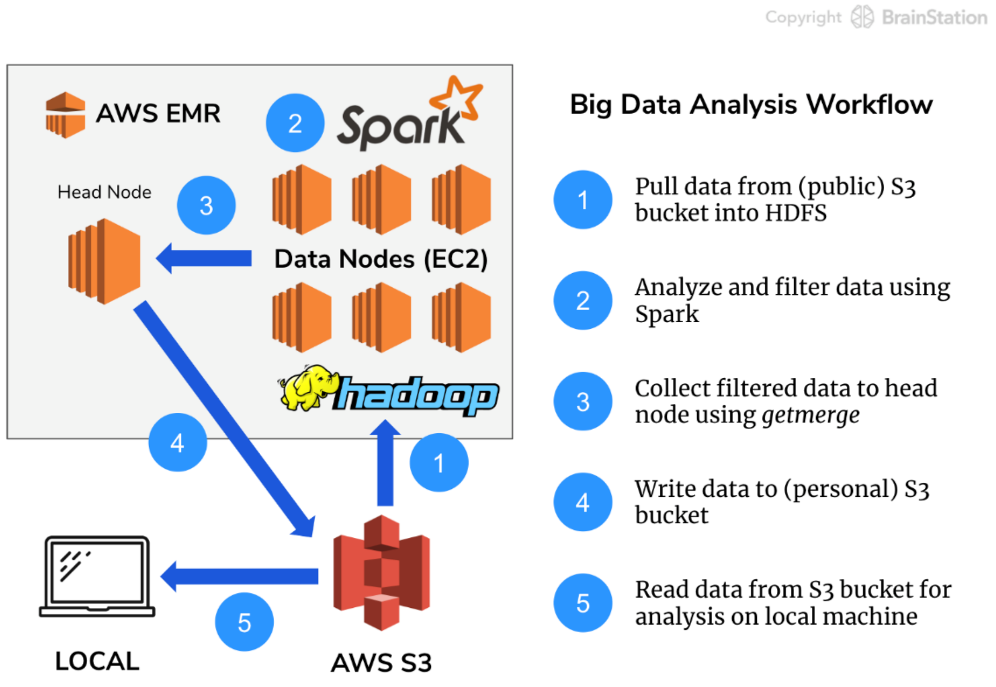
Email: amkoxia@gmail.com

In this assignment, you will apply the skills you've learned in the Big Data Fundamentals unit to load, filter, and visualize a large real-world dataset in a cloud-based distributed computing environment using Hadoop, Spark, Hive, and the S3 filesystem. Prepare a professional report to summarize the findings and be sure to include an appendix with screenshots of the steps completed for Questions 1 and 2.

The Google Ngrams dataset was created by Google's research team by analyzing all of the content in Google Books - these digitized texts represent approximately 4% of all books ever printed, and span a time period from the 1800s into the 2000s.

The dataset is hosted in a public S3 bucket as part of the Amazon S3 Open Data Registry. For this assignment, we have converted the data to CSV and hosted it on a public S3 bucket which may be accessed here: s3://brainstation-dsft/eng_1M_1gram.csv

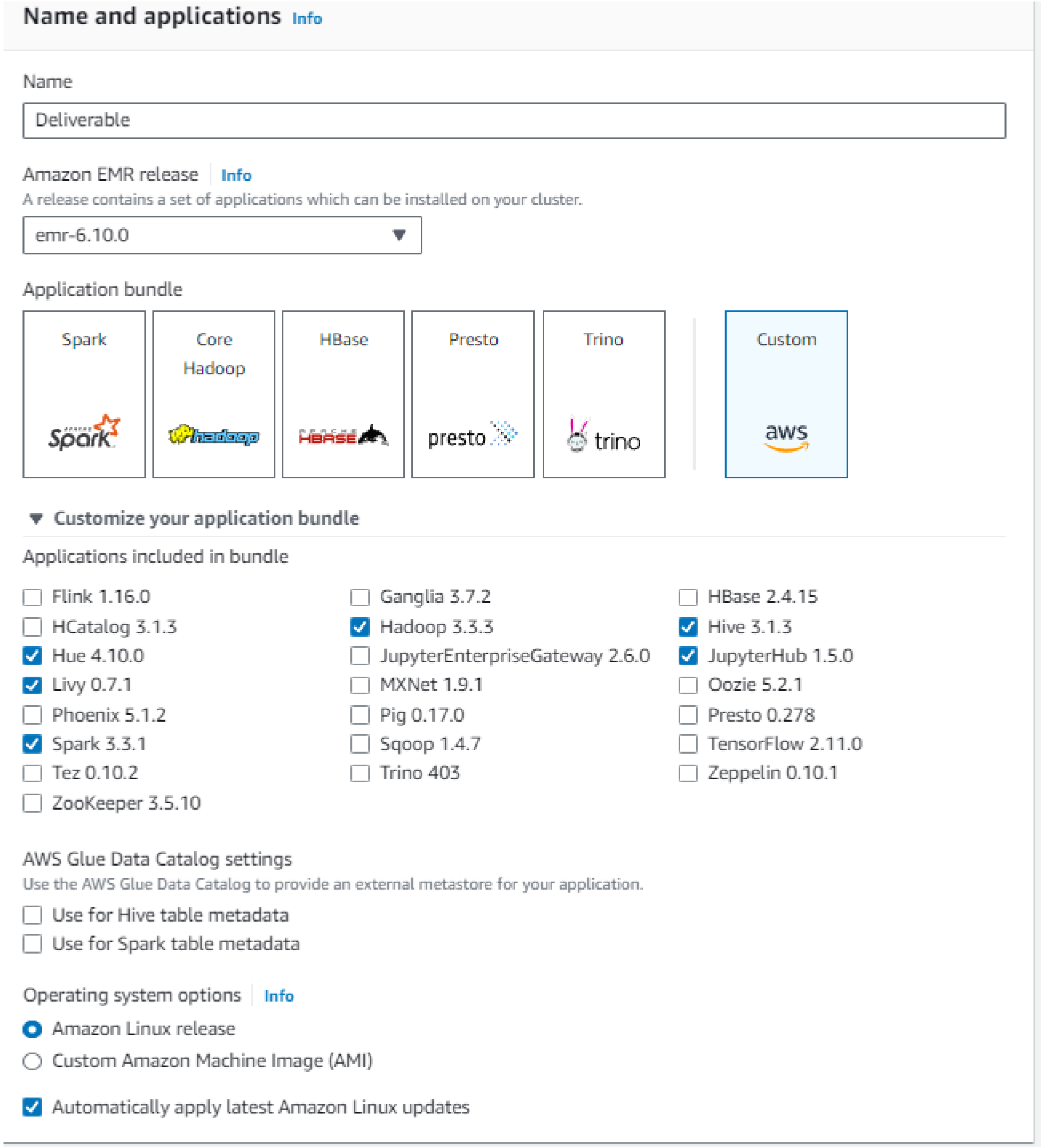
For this deliverable, you will produce a report, as well as a jupyter notebook, which will follow a Big Data analysis workflow. As part of this workflow you will filter and reduce data down to a manageable size, and then do some analysis locally on our machine after extracting data from the Cloud and processing it using Big Data tools. The workflow and steps in the process are illustrated below:


Copyright BrainStation
AWS EMR
2
Spark
Head Node
3
Data Nodes (EC2)
hadoop
4
1
5
LOCAL
AWS S3
Big Data Analysis Workflow
1 Pull data from (public) S3 bucket into HDFS
2 Analyze and filter data using Spark
3 Collect filtered data to head node using *getmerge*
4 Write data to (personal) S3 bucket
5 Read data from S3 bucket for analysis on local machine

Q1. Spin up a new EMR cluster on AWS for using Spark and EMR notebooks - follow the same instructions as for the Spark Lab.

Answer: I'll take us through this, using screenshots, step by step:

First, I select the relevant custom features for my cluster.



Name and applications Info

Name

Deliverable

Amazon EMR release | Info

A release contains a set of applications which can be installed on your cluster.

emr-6.10.0 ▼

Application bundle

Spark | Core Hadoop | HBase | Presto | Trino | Custom

▼ Customize your application bundle

Applications included in bundle

□ Flink 1.16.0
□ HCatalog 3.1.3
☑ Hue 4.10.0
☑ Livy 0.7.1
□ Phoenix 5.1.2
☑ Spark 3.3.1
□ Tez 0.10.2
□ ZooKeeper 3.5.10
□ Ganglia 3.7.2
☑ Hadoop 3.3.3
□ JupyterEnterpriseGateway 2.6.0
□ MXNet 1.9.1
□ Pig 0.17.0
□ Sqoop 1.4.7
□ Trino 403
□ HBase 2.4.15
☑ Hive 3.1.3
☑ JupyterHub 1.5.0
□ Oozie 5.2.1
□ Presto 0.278
□ TensorFlow 2.11.0
□ Zeppelin 0.10.1

AWS Glue Data Catalog settings

Use the AWS Glue Data Catalog to provide an external metastore for your application.

□ Use for Hive table metadata
□ Use for Spark table metadata

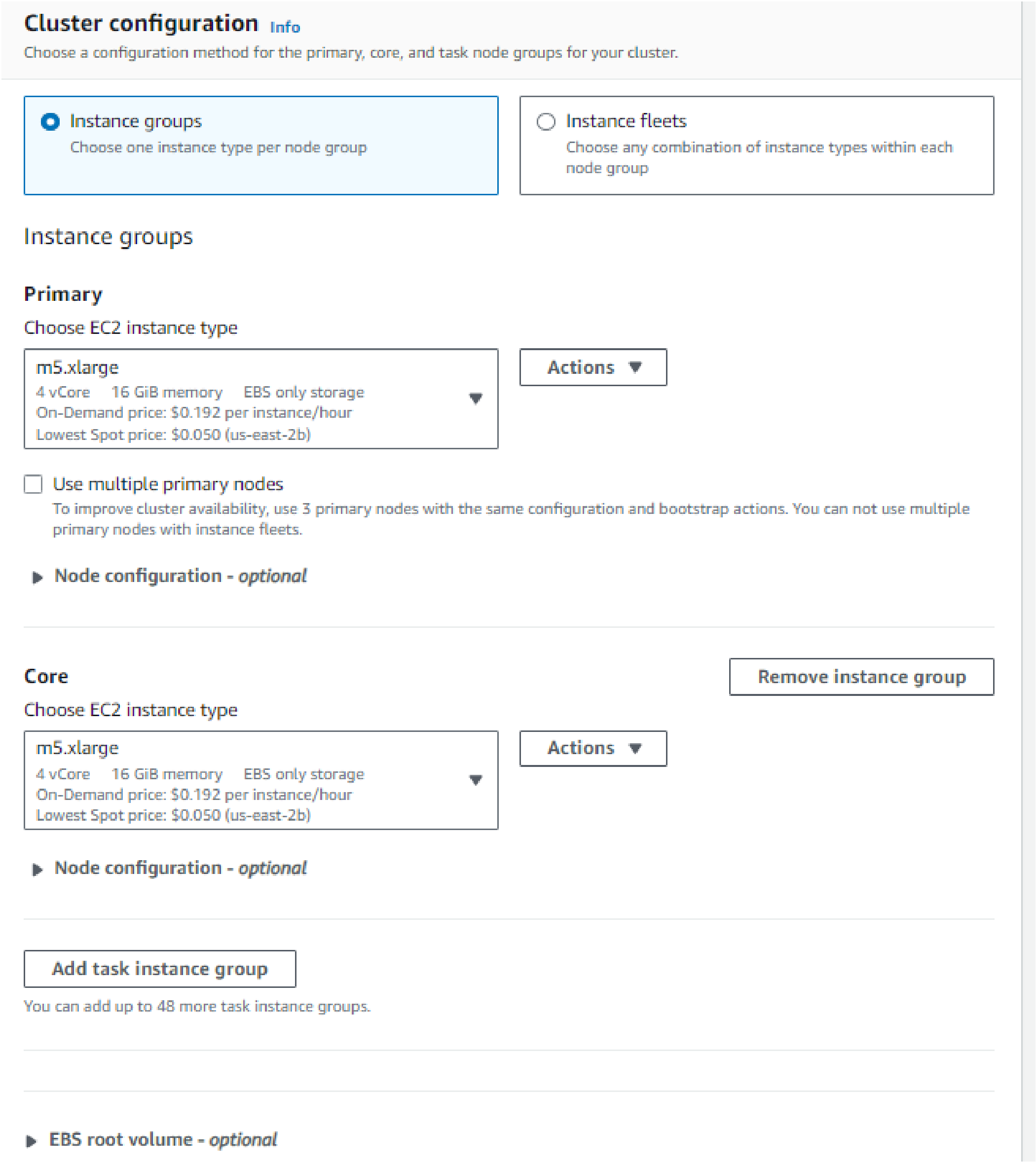
Operating system options | Info

● Amazon Linux release
○ Custom Amazon Machine Image (AMI)

☑ Automatically apply latest Amazon Linux updates

We have named a new cluster as "Deliverable" with emr-6. 10.0.

I then removed 'Task' instance group:



Cluster configuration

Choose a configuration method for the primary, core, and task node groups for your cluster.

● Instance groups
Choose one instance type per node group

○ Instance fleets
Choose any combination of instance types within each node group

Instance groups

Primary

Choose EC2 instance type

m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: $0.192 per instance/hour
Lowest Spot price: $0.050 (us-east-2b)

□ Use multiple primary nodes
To improve cluster availability, use 3 primary nodes with the same configuration and bootstrap actions. You can not use multiple primary nodes with instance fleets.

▶ Node configuration - *optional*

Core

Choose EC2 instance type

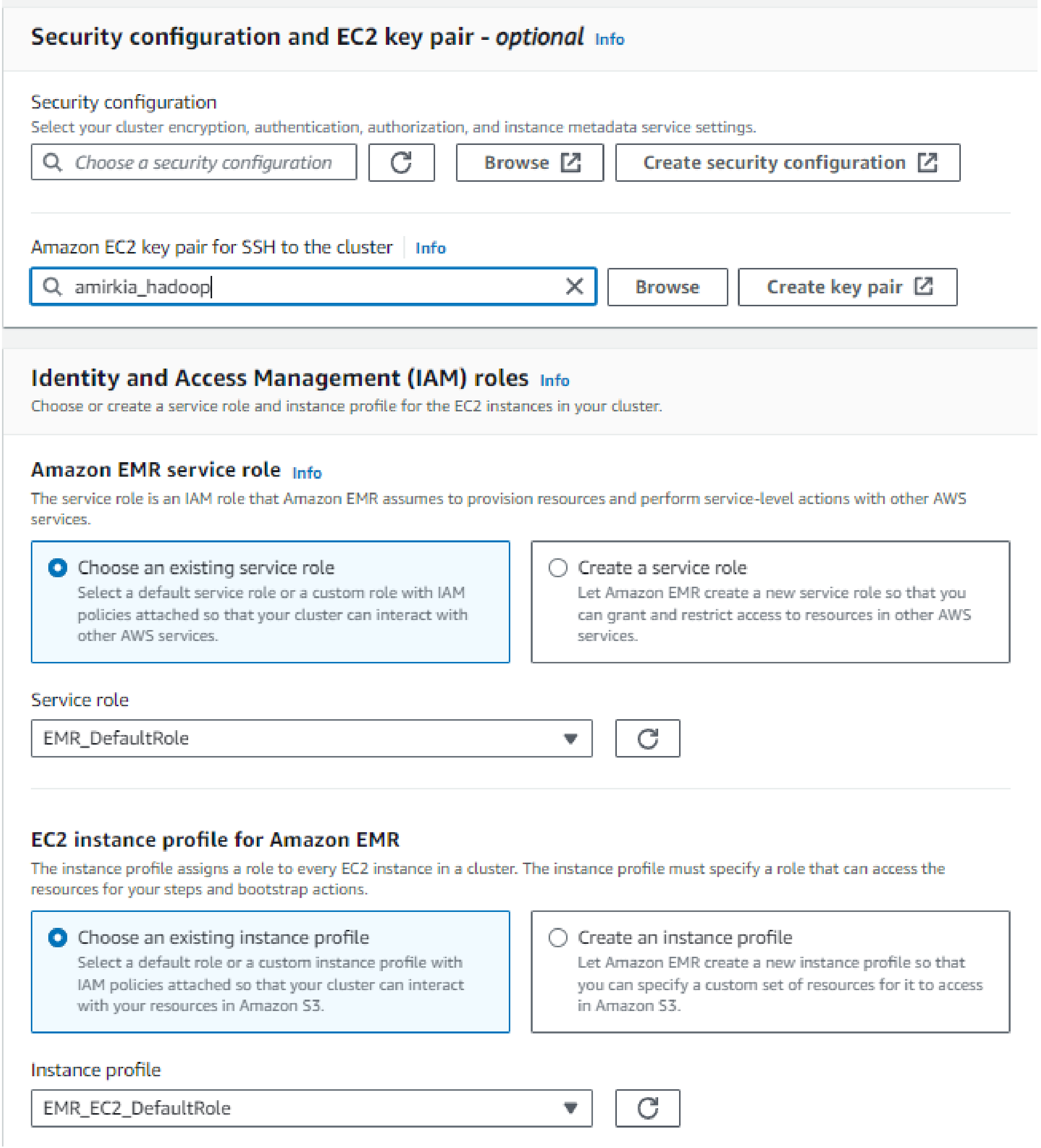
m5.xlarge
4 vCore 16 GiB memory EBS only storage
On-Demand price: $0.192 per instance/hour
Lowest Spot price: $0.050 (us-east-2b)

▶ Node configuration - *optional*

You can add up to 48 more task instance groups.

▶ EBS root volume - *optional*

Then service role and instance profile are selected:



Security configuration and EC2 key pair - *optional* Info

Security configuration

Select your cluster encryption, authentication, authorization, and instance metadata service settings.

Choose a security configuration
Browse
Create security configuration

Amazon EC2 key pair for SSH to the cluster | Info

amirkia_hadoop

Identity and Access Management (IAM) roles Info

Choose or create a service role and instance profile for the EC2 instances in your cluster.

Amazon EMR service role Info

The service role is an IAM role that Amazon EMR assumes to provision resources and perform service-level actions with other AWS services.

● Choose an existing service role
Select a default service role or a custom role with IAM policies attached so that your cluster can interact with other AWS services.

○ Create a service role
Let Amazon EMR create a new service role so that you can grant and restrict access to resources in other AWS services.

Service role

EMR_DefaultRole

EC2 instance profile for Amazon EMR

The instance profile assigns a role to every EC2 instance in a cluster. The instance profile must specify a role that can access the resources for your steps and bootstrap actions.

● Choose an existing instance profile
Select a default role or a custom instance profile with IAM policies attached so that your cluster can interact with your resources in Amazon S3.

○ Create an instance profile
Let Amazon EMR create a new instance profile so that you can specify a custom set of resources for it to access in Amazon S3.

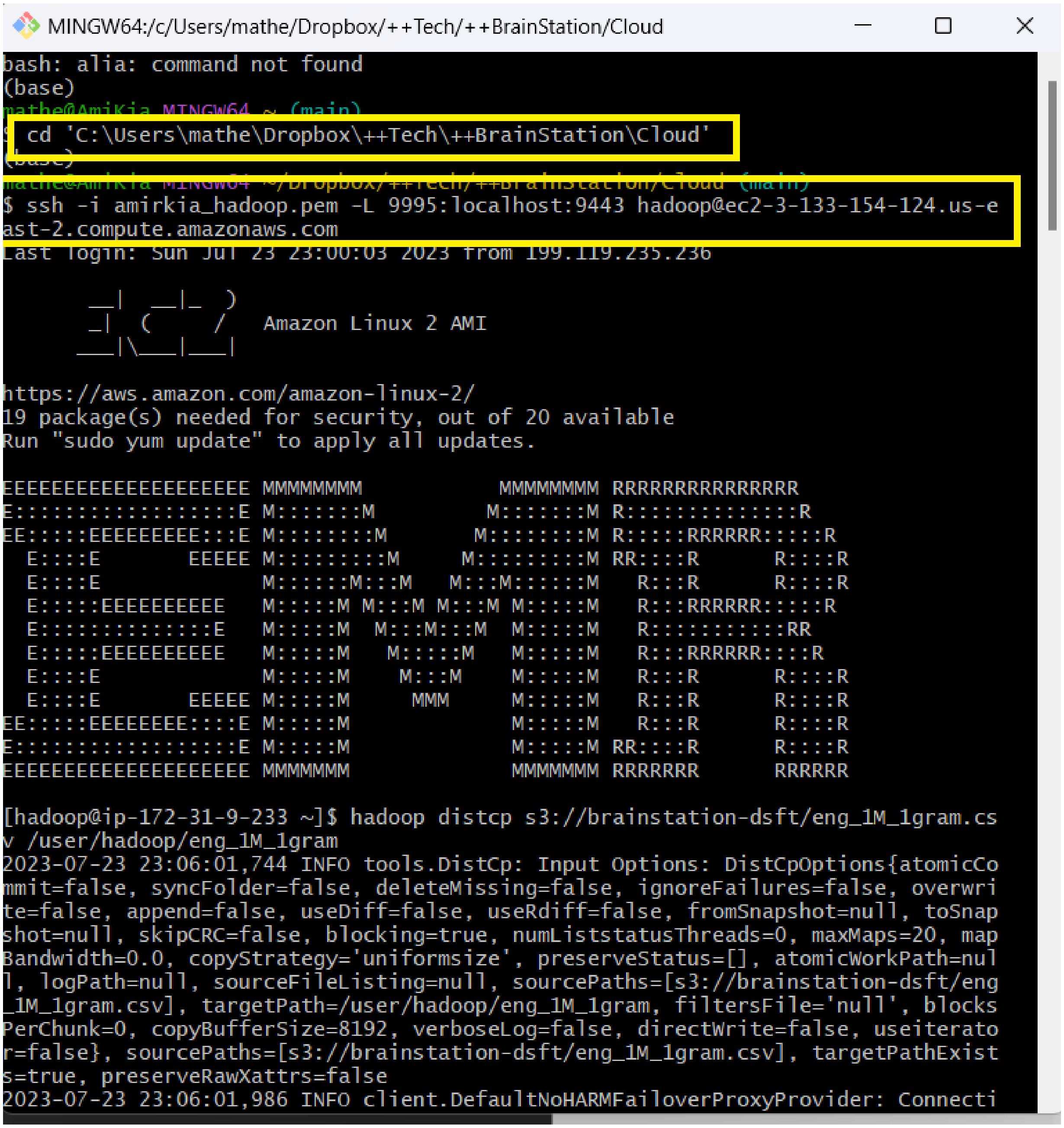
Instance profile

EMR_EC2_DefaultRole

The cluster is up and running:

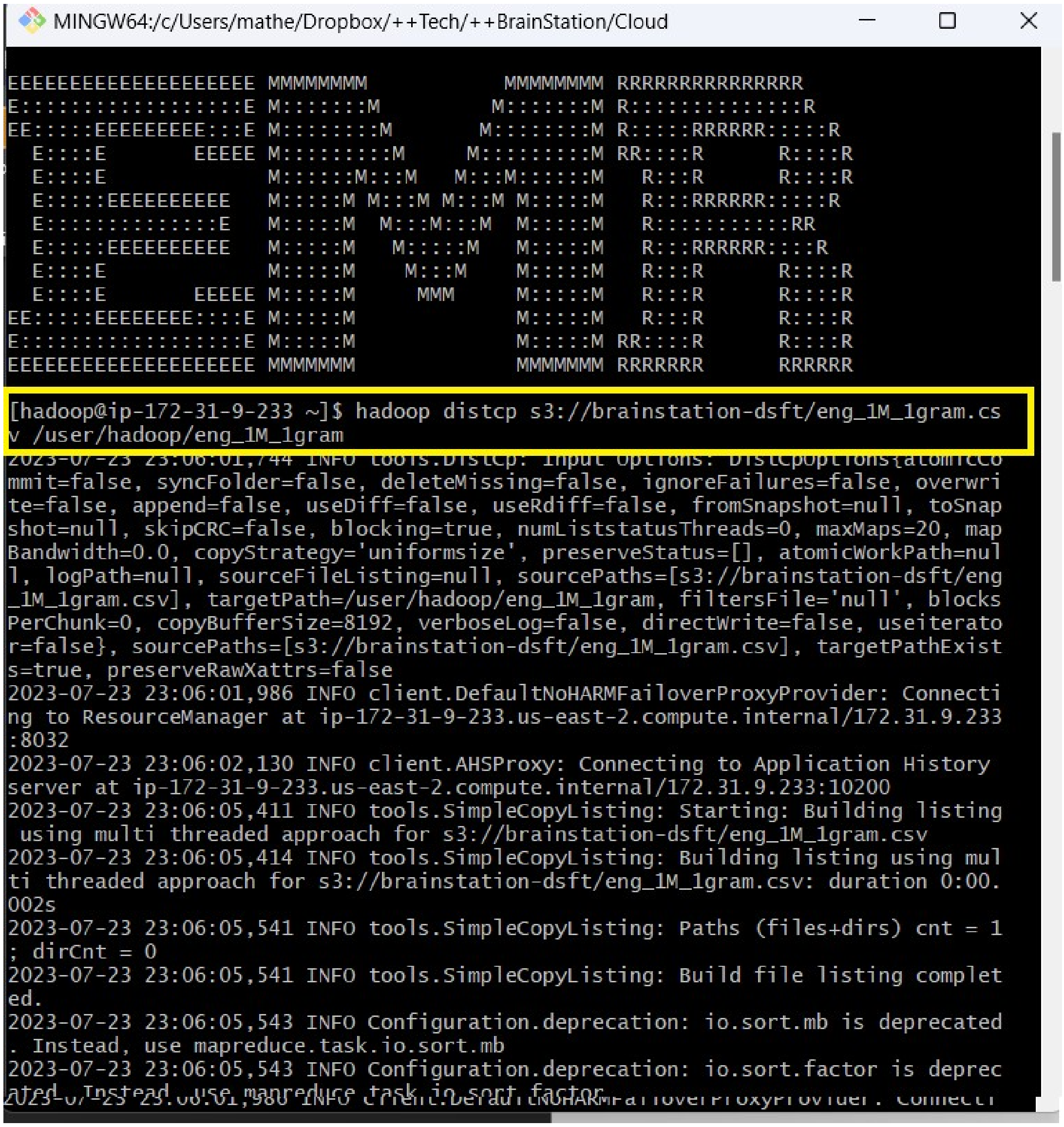
Q2. Connect to the head node of the cluster using SSH.

Answer: I ran the highlighted codes in order to connect to the head note of the cluster. Note that the second one takes the relevant information from ‘Connect to the Primary Node using SSH’ in the cluster.



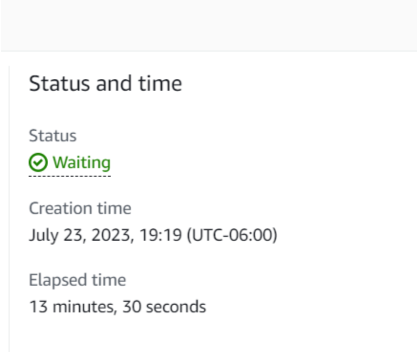
```
bash: alia: command not found
(base)
mathe@AmiKia MINGW64 ~ (main)
$ cd 'C:\Users\mathe\Dropbox\++Tech\++BrainStation\Cloud'
(base)
mathe@AmiKia MINGW64 ~/Dropbox/++Tech/++BrainStation/Cloud (main)
$ ssh -i amirkia_hadoop.pem -L 9995:localhost:9443 hadoop@ec2-3-133-154-124.us-e
ast-2.compute.amazonaws.com
Last login: Sun Jul 23 23:00:03 2023 from 199.119.235.236

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
19 package(s) needed for security, out of 20 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-9-233 ~]$ hadoop distcp s3://brainstation-dsft/eng_1M_1gram.cs
v /user/hadoop/eng_1M_1gram
2023-07-23 23:06:01,744 INFO tools.DistCp: Input Options: DistCpOptions{atomicCo
mmit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwri
te=false, append=false, useDiff=false, useRdiff=false, fromSnapshot=null, toSnap
shot=null, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, map
Bandwidth=0.0, copyStrategy='uniformsize', preserveStatus=[], atomicWorkPath=nul
l, logPath=null, sourceFileListing=null, sourcePaths=[s3://brainstation-dsft/eng
_1M_1gram.csv], targetPath=/user/hadoop/eng_1M_1gram, filtersFile='null', blocks
PerChunk=0, copyBufferSize=8192, verboseLog=false, directWrite=false, useiterato
r=false}, sourcePaths=[s3://brainstation-dsft/eng_1M_1gram.csv], targetPathExist
s=true, preserveRawXattrs=false
2023-07-23 23:06:01,986 INFO client.DefaultNoHARMFailoverProxyProvider: Connecti
```

We can use `distcp` to copy the data directly from the public S3 buckets into EMR:



```
EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-9-233 ~]$ hadoop distcp s3://brainstation-dsft/eng_1M_1gram.cs
v /user/hadoop/eng_1M_1gram
2023-07-23 23:06:01,744 INFO tools.DistCp: Input Options: DistCpOptions{atomicCo
mmit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwri
te=false, append=false, useDiff=false, useRdiff=false, fromSnapshot=null, toSnap
shot=null, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, map
Bandwidth=0.0, copyStrategy='uniformsize', preserveStatus=[], atomicWorkPath=nul
l, logPath=null, sourceFileListing=null, sourcePaths=[s3://brainstation-dsft/eng
_1M_1gram.csv], targetPath=/user/hadoop/eng_1M_1gram, filtersFile='null', blocks
PerChunk=0, copyBufferSize=8192, verboseLog=false, directWrite=false, useiterato
r=false}, sourcePaths=[s3://brainstation-dsft/eng_1M_1gram.csv], targetPathExist
s=true, preserveRawXattrs=false
2023-07-23 23:06:01,986 INFO client.DefaultNoHARMFailoverProxyProvider: Connecti
ng to ResourceManager at ip-172-31-9-233.us-east-2.compute.internal/172.31.9.233
:8032
2023-07-23 23:06:02,130 INFO client.AHSProxy: Connecting to Application History
server at ip-172-31-9-233.us-east-2.compute.internal/172.31.9.233:10200
2023-07-23 23:06:05,411 INFO tools.SimpleCopyListing: Starting: Building listing
 using multi threaded approach for s3://brainstation-dsft/eng_1M_1gram.csv
2023-07-23 23:06:05,414 INFO tools.SimpleCopyListing: Building listing using mul
ti threaded approach for s3://brainstation-dsft/eng_1M_1gram.csv: duration 0:00.
002s
2023-07-23 23:06:05,541 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1
; dirCnt = 0
2023-07-23 23:06:05,541 INFO tools.SimpleCopyListing: Build file listing complet
ed.
2023-07-23 23:06:05,543 INFO Configuration.deprecation: io.sort.mb is deprecated
. Instead, use mapreduce.task.io.sort.mb
2023-07-23 23:06:05,543 INFO Configuration.deprecation: io.sort.factor is deprec
ated. Instead, use mapreduce.task.io.sort.factor
```

I can now access JupyterHub in our browser at https://localhost:9995



Status and time

Status

Waiting

Creation time

July 23, 2023, 19:19 (UTC-06:00)

Elapsed time

13 minutes, 30 seconds

The rest of the deliverable is presented in the attached notebook.