CLUSTER ERM Y GESTIÓN DE ARCHIVOS EN HDFS Y S3 PARA BIG DATA

Juan Pablo Gómez Triana[1]

Resumen

Crear un Cluster AWS EMR en Amazon que contenga tecnologías como Hadoop, JupyterHub, Hive, Zeppelin, Hue y Spark; además de constar de un nodo maestro y 2 nodos core. Se busca gestionar ficheros y archivos dentro del cluster usando el sistema HDFS y S3 para la persistencia.

Palabras Clave: Cloud Computing, AWS, Cluster, EMR, Hadoop, JupyterHub, Hive, Zeppelin, Hue y Spark, Nodo Maestro, Nodo Core, HDFS, S3, Big Data.

Introducción

En este documento se numeran los pasos requeridos para desplegar y configurar el Cluster planteado en los laboratorios 1 y 2 de la unidad 3.

[1] Estudiante de Ingeniería de Sistemas de la Universidad EAFIT, Medellín, Antioquia. E-mail: jpgomezt@eafit.edu.co

Creación del Bucket con S3

En esta sección mostraremos los pasos que fueron requeridos para la creación del Bucket con el sistema de AWS S3. Este bucket será utilizado para almacenar los datos de entrada o de salida de los programas que se ejecuten en el cluster. En este caso, crearemos el bucket para los datos del notebook.
Para iniciar, se instancio un Bucket en el servicio S3 de AWS:

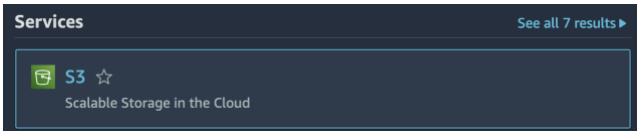


Figura 1. Servicio S3.

En este servicio, seleccionamos la opción de "Buckets" y seleccionamos "Create Bucket".
Solo es necesario nombrar nuestro bucket, en este caso se llamará "notebooksjpgomezt". Finalizamos seleccionando "Create Bucket":

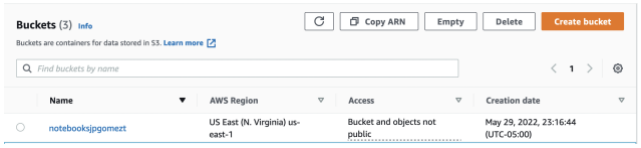


Figura 2. Bucket creado.

Creación del Cluster EMR

En esta sección mostraremos los pasos que fueron requeridos para la creación del Cluster con el sistema de AWS EMR. Para esto es importante primero poseer una clave para conectarse con el nodo mediante SSH. Para esto reutilizaremos la clave "AWS-KEY.pem" utilizada en laboratorios anteriores:

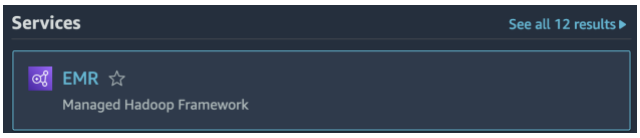


Figura 3. Servicio EMR.

En este servicio, seleccionamos la opción de "Clusters" y seleccionamos "Create cluster".
Luego, es necesario seleccionar las opciones avanzadas, para poder configurar los softwares que utilizara el cluster.

Primero, seleccionamos la versión del EMR que utilizaremos. En este caso, utilizamos la versión "emr-6.3.1". Luego seleccionamos los softwares que se instalaran en nuestro cluster:

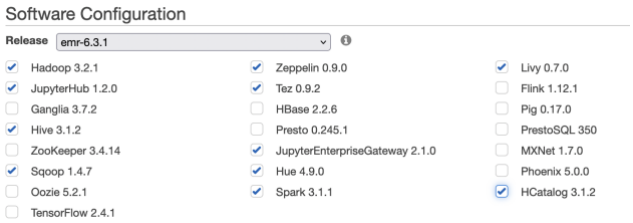


Figura 4. Software Configuration del Cluster.

Luego, hacemos integración entre el catálogo de Glue, lo que nos permite que las tablas que se creen sean visibles tanto desde Hive como desde Spark:

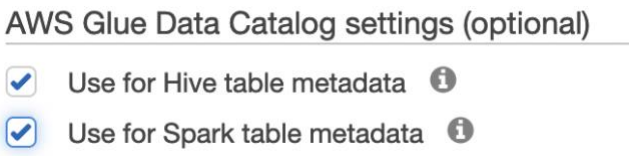


Figura 5. Integración de Glue con Hive y Spark.

Ahora configuramos la persistencia de los notebooks que se crearan con JupyterHub, utilizando el bucket que creamos con S3. Para esto, en "Edit software settings", debemos entregar la siguiente configuración:

```
[
  {
    "Classification": "jupyter-s3-conf",
    "Properties": {
      "s3.persistence.enabled": "true",
      "s3.persistence.bucket": "
notebooksjpgomezt "
    }
  }
]
```

Para el Hardware de nuestro cluster, en la opción "Cluster Nodes and Instances", debeos cambiar el tipo de instancias que creara el servicio EMR de "m5.xlarge" a una "m4.xlarge", y seleccionaremos la opción "Spot":

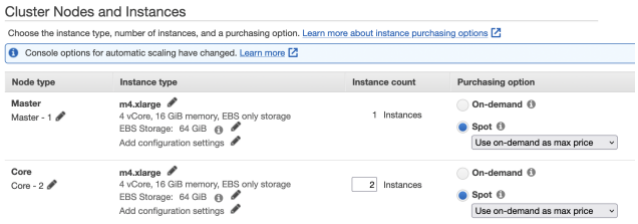


Figura 6. Hardware del Cluster.

Configuramos también la opción de "Auto-termination" para que el cluster sea destruido después de 1 hora de inactividad, y configuramos 20 GiB de almacenamiento:

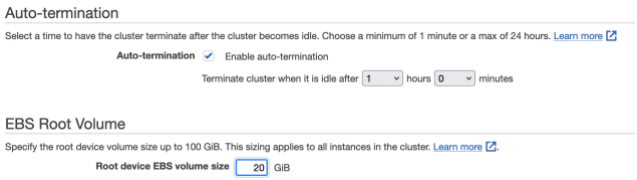


Figura 7. Auto-termination y Almacenamiento.

Nombramos nuestro cluster en "Cluster name". En este caso el cluster se llamará "My cluster jpgomezt". Finalmente, en "Security Options", le asignamos la clave anteriormente mencionada "AWS-KEY.pem". Una vez configurada estas opciones, podemos seleccionar "Create Cluster":

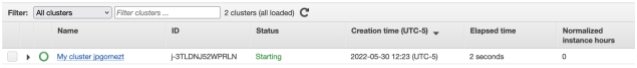


Figura 8. Creación del Cluster.

Debemos esperar entre 20-30 minutos a que el cluster se instancie para poder interactuar con el cluster. Ahora configuraremos los puertos que el cluster tendrá abiertos.

Puertos del Cluster

Existen dos configuraciones para abrir los puertos requeridos para interactuar con las aplicaciones del cluster. Primero, vamos a la opción "Block public Access" dentro del servicio EMR. En esta podemos editar los puertos por los que queremos permitir el acceso:

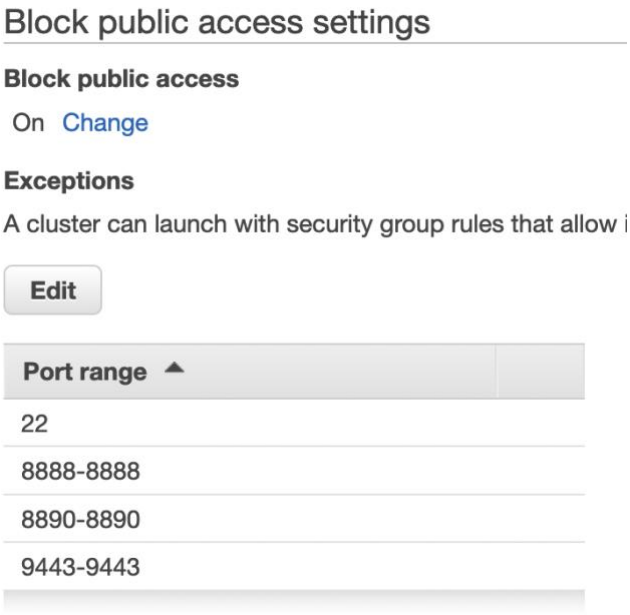
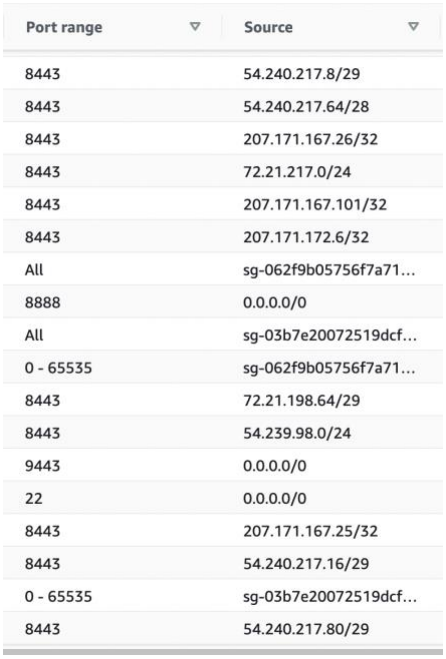


Figura 9. Puertos permitidos en Block Public Access.

Ahora, si seleccionamos nuestro cluster, podemos ir a la opción "Security and Access", y desde esta podemos configurar el security group del nodo master. Debemos añadir el puerto 8888, 9443, 8890, y el 22:



Port range	Source
8443	54.240.217.8/29
8443	54.240.217.64/28
8443	207.171.167.26/32
8443	72.21.217.0/24
8443	207.171.167.101/32
8443	207.171.172.6/32
All	sg-062f9b05756f7a71...
8888	0.0.0.0/0
All	sg-03b7e20072519dcf...
0 - 65535	sg-062f9b05756f7a71...
8443	72.21.198.64/29
8443	54.239.98.0/24
9443	0.0.0.0/0
22	0.0.0.0/0
8443	207.171.167.25/32
8443	54.240.217.16/29
0 - 65535	sg-03b7e20072519dcf...
8443	54.240.217.80/29

Figura 10. Puertos permitidos en el Security Group.

Configuración del Cluster EMR

Una vez nuestro cluster esté en funcionamiento, nos conectaremos por SSH para poder configurar Sqoop, de forma que lo podamos utilizar con la

interfaz web de Hue. Para conectarnos utilizamos el DNS público que ofrece el cluster para el nodo maestro:

ssh -i AWS-KEY.pem hadoop@ec2-54-162-195-99.compute-1.amazonaws.com

Ahora, necesitamos conocer el nombre del directorio lib donde se encuentran ubicados los componentes de oozie. Para esto podemos correr el siguiente comando:

hdfs dfs -ls /user/oozie/share/lib/

Esta operación nos lanzara de resultado el nombre del directorio:

Figura 11. Directorio Lib del EMR.

Podemos observar que el directorio tiene el nombre de "lib_20220530174018". Ahora solo debemos correr los siguientes comandos para habilitar Sqoop en la interfaz web Hue:

$ hdfs dfs -put /usr/share/java/mysql-connector-java.jar /user/oozie/share/lib/lib_20220530174018/sqoop/

$ hdfs dfs -chown oozie /user/oozie/share/lib/lib_20220530174018/sqoop/mysql-connector-java.jar

$ hdfs dfs -chgrp oozie /user/oozie/share/lib/lib_20220530174018/sqoop/mysql-connector-java.jar

$ hdfs dfs -cp /user/oozie/share/lib/lib_20220530174018/hive/ /user/oozie/share/lib/lib_20220530174018/sqoop/*

*$ hdfs dfs -chown oozie /user/oozie/share/lib/lib_20220530174018/sqoop/**

*$ hdfs dfs -chgrp oozie /user/oozie/share/lib/lib_20220530174018/sqoop/**

Finalmente, verificamos que el proceso haya sido exitoso y no ocurriera ningún error:

$ oozie admin -sharelibupdate

Figura 12. Configuración de Sqoop.

Gestión de Archivos HDFS por HUE

Una vez nuestro cluster esté en funcionamiento, debemos acceder a la consola web de HUE. Para esto, utilizamos la dirección que nos ofrece el panel de "Application user interfaces":

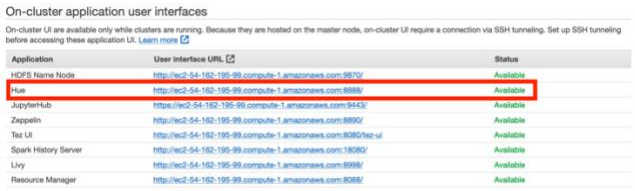


Figura 12. Dirección de la interfaz web de HUE.

Una vez ingresamos, debemos crear un usuario y contraseña. Nuestro usuario será "hadoop" y una vez hecho esto, podemos ingresar a la interfaz web de HUE:

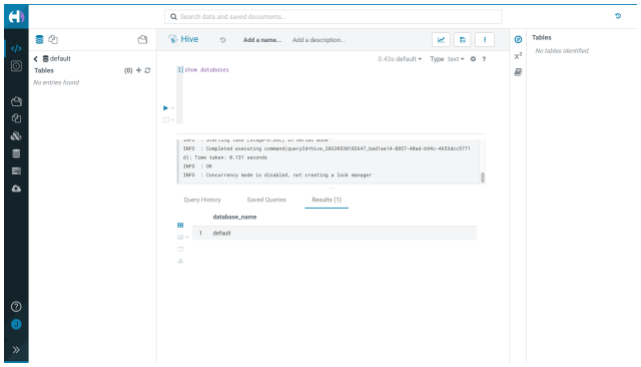


Figura 13. Interfaz web de HUE.

Una vez ingresados, podemos ir a la sección archivos. Desde ahí, por defecto, HUE nos llevara

a nuestra ruta por defecto teniendo en cuenta nuestro usuario (hadoop):

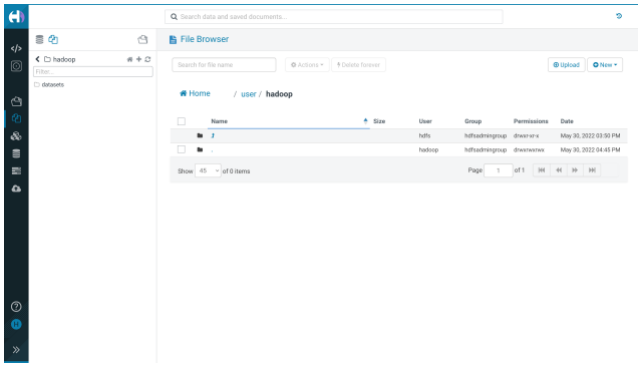


Figura 14. Directorio HDFS del usuario hadoop.

Podemos observar que el directorio se encuentra vacío. Podemos crear el directorio "datasets" utilizando la interfaz gráfica:

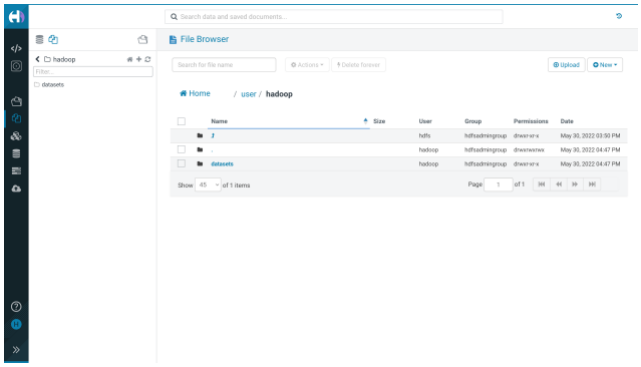


Figura 15. Creación del directorio datasets.

Una vez dentro del directorio "datasets", podemos subir el archivo "airlines.csv":

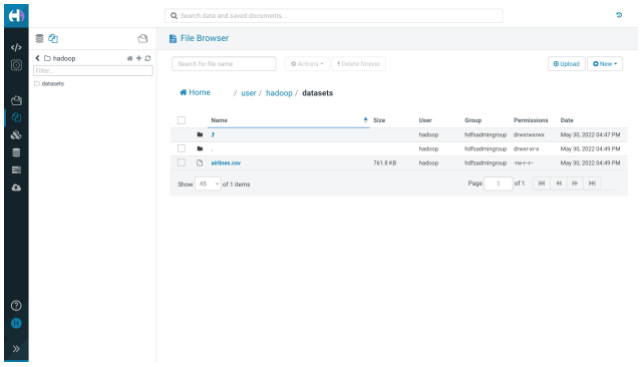


Figura 16. Archivo subido al HDFS por HUE.

De esta forma podemos subir archivos y crear directorios en el HDFS utilizando la interfaz gráfica de HUE.

Gestión de Archivos S3 por HUE

Ingresando a la misma interfaz web de HUE, podemos ir a la sección "S3", donde veremos los buckets disponibles en la región:

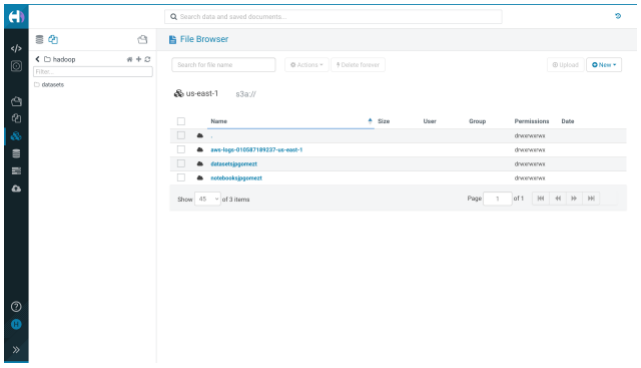


Figura 17. Buckets dentro de la región.

Ingresamos al bucket "datasetsjpgomezt", y allí podremos crear el directorio "datasets":

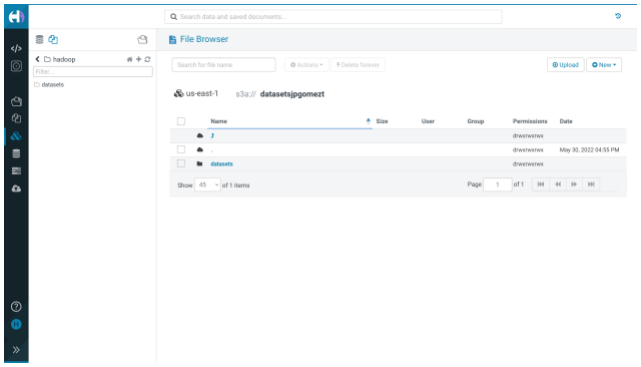


Figura 18. Directorios dentro del bucket.

Una vez dentro del directorio "datasets", podemos subir el archivo "airlines.csv":

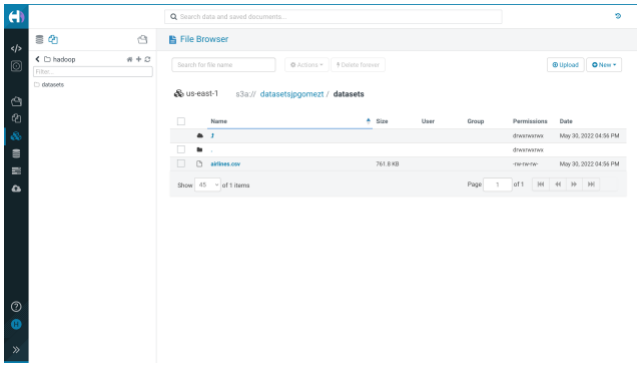


Figura 19. Archivo subido a S3 por HUE.

De esta forma podemos subir archivos y crear directorios en S3 utilizando la interfaz gráfica de

HUE. Podemos verificar que si fueran creados en la consola de AWS:

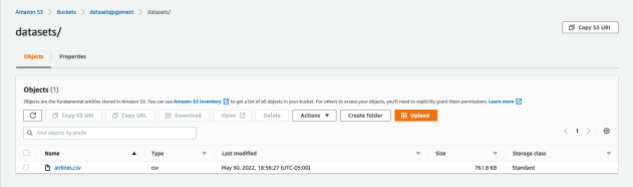


Figura 20. Bucket en AWS.

Gestión de Archivos HDFS por SSH

Antes de poder utilizar ssh, todo el directorio "datasets" y su contenido fueron ubicados en "/home/hadoop/datasets/" utilizando scp:

$ scp -i AWS-KEY.pem -r /Users/jpgomezt/Downloads/datasets/ hadoop@ec2-3-91-221-129.compute-1.amazonaws.com:/home/hadoop/datasets*

Volvemos a conectarnos a nuestro nodo maestro por ssh como se hizo anteriormente. Una vez dentro, podemos verificar que el directorio "datasets" si fue creado. Para esto revisamos "user/<username>/" donde <*username*> será el mismo que utilizamos en la interfaz gráfica de HUE (hadoop):

$ hdfs dfs -ls /user/hadoop/

```
[hadoop@ip-172-31-45-215 ~]$ hdfs dfs -ls /user/hadoop/
Found 1 items
drwxr-xr-x   - hadoop hdfsadmingroup          0 2022-05-30 23:49 /user/hadoop/datasets
```

Figura 21. Lista de directorios del usuario hadoop.

En caso de que el directorio no existiera, podríamos crearlo utilizando el comando "mkdir":

$ hdfs dfs -mkdir /user/hadoop/datasets

Ahora, podemos crear el directorio "gutenberg-small" dentro de "datasets":

$ hdfs dfs -mkdir /user/hadoop/datasets/gutenberg-small

Y ahora, solo basta con copiar los archivos que están en nuestro directorio local, hacia el directorio del HDFS:

$ hdfs dfs -copyFromLocal /home/hadoop/datasets/gutenberg-small/ /user/hadoop/datasets/gutenberg-small/*

Si verificamos la interfaz HUE podemos observar el nuevo directorio con los respectivos archivos:

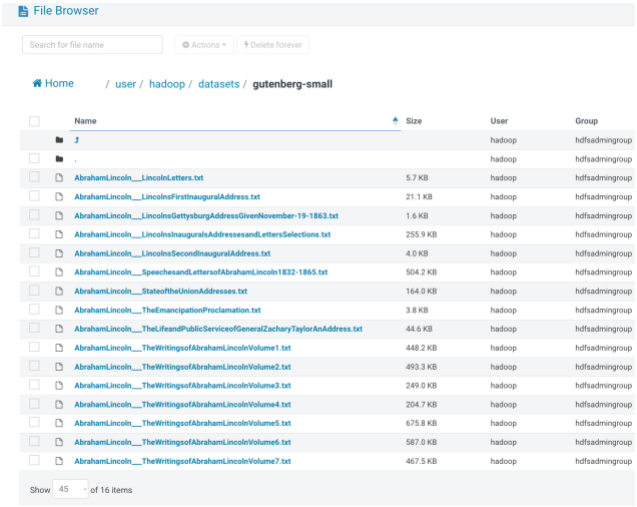


Figura 22. Directorio y contenido creado por SSH.

Ahora podemos pasar todo el directorio "datasets" a nuestro HDFS:

$ hdfs dfs -copyFromLocal /home/hadoop/datasets/ /user/hadoop/datasets/*

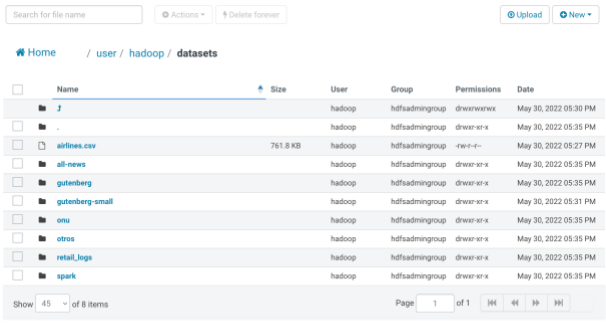


Figura 23. Directorio datasets completo en HDFS.

Gestión de Archivos S3 por SSH

Volvemos a conectarnos a nuestro nodo maestro por ssh como se hizo anteriormente. Una vez dentro, podemos verificar que el directorio "datasets" si fue creado. Para esto revisamos "s3://<bucket>/" donde <*bucket*> será el mismo

que utilizamos en la interfaz gráfica de HUE (datasetsjpgomezt):

$ hdfs dfs -ls s3://datasetsjpgomezt/

Figura 24. Lista de directorios del bucket datasetsjpgomezt.

En caso de que el directorio no existiera, podríamos crearlo utilizando el comando "mkdir":

$ hdfs dfs -mkdir s3://datasetsjpgomezt/datasets/

Ahora, podemos crear el directorio "gutenberg-small" dentro de "datasets":

$ hdfs dfs -mkdir s3://datasetsjpgomezt/datasets/gutenberg-small/

Y ahora, solo basta con copiar los archivos que están en nuestro directorio local, hacia el directorio del HDFS:

$ hdfs dfs -copyFromLocal /home/hadoop/datasets/gutenberg-small/ s3://datasetsjpgomezt/datasets/gutenberg-small/*

Si verificamos la interfaz HUE podemos observar el nuevo directorio con los respectivos archivos:

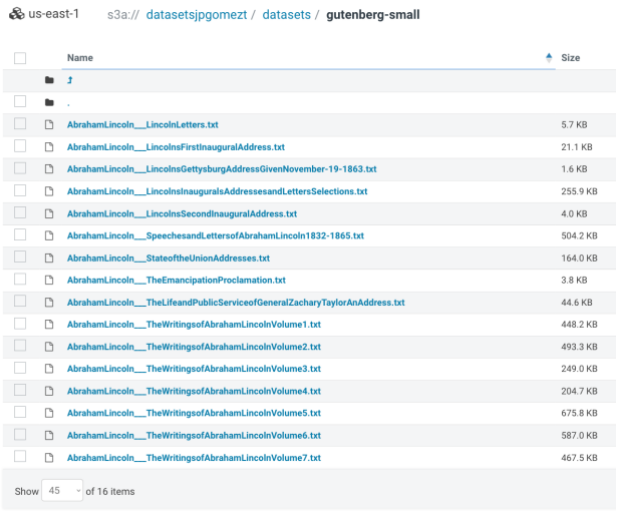


Figura 25. Directorio y contenido creado por SSH.

Ahora podemos pasar todo el directorio "datasets" a nuestro HDFS:

$ hdfs dfs -copyFromLocal /home/hadoop/datasets/ /user/hadoop/datasets/*

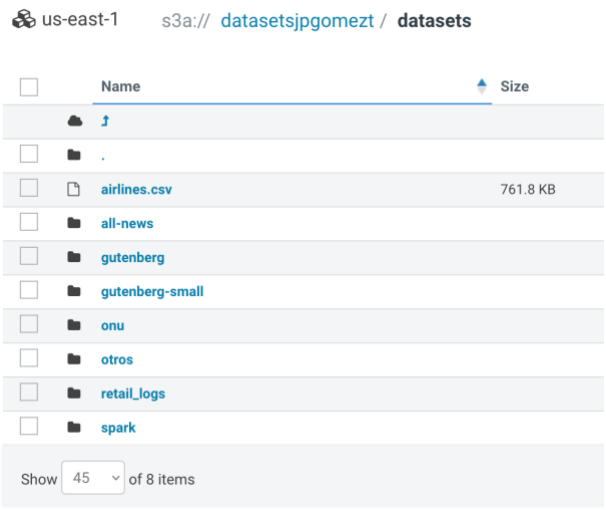


Figura 26. Directorio datasets completo en S3.

Podemos verificar que si fueran creados en la consola de AWS:

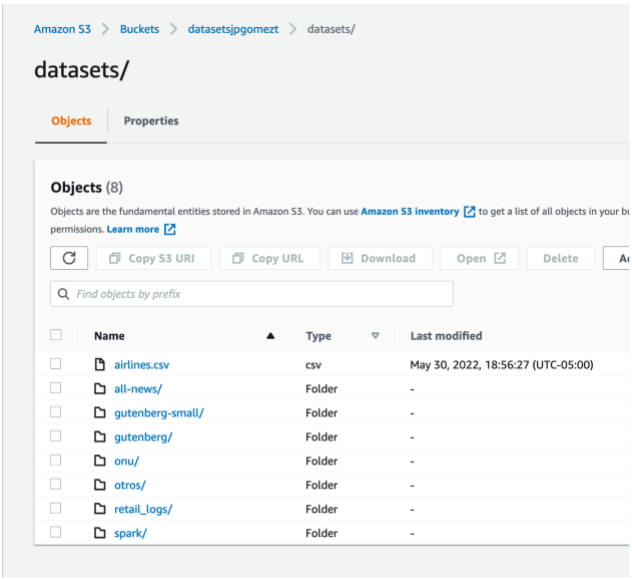


Figura 27. Bucket completo en AWS.

La URI del bucket es: s3://datasetsjpgomezt/datasets/
Y la URL es: https://datasetsjpgomezt.s3.amazonaws.com/datasets/

JupyterHub

También podemos ingresar a la consola web de JupyterHub, buscando la dirección en el mismo panel de "Application user interfaces". Una vez ingresamos, debemos proveer un usuario y clave para ingresar. Por defecto, estos son "jovyan" y "jupyter" respectivamente. Una vez adentro, podemos empezar a crear notebooks:

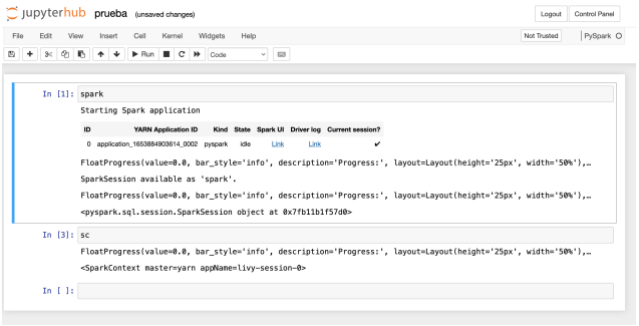


Figura 14. Notebook con PySpark en JupyterHub.

Zeppelin

Finalmente, podemos ingresar a la consola web de Zeppelin, buscando la dirección en el mismo panel de "Application user interfaces".

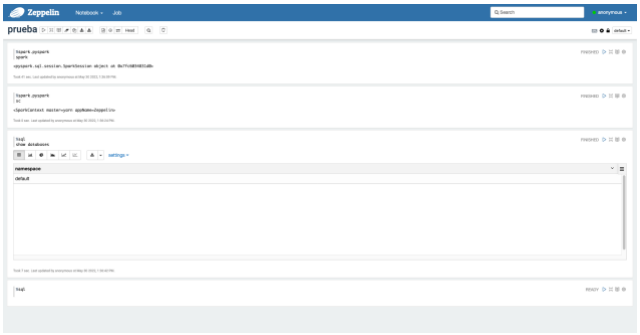


Figura 15. Notebook en Zeppelin.

Con esto ya tendríamos nuestro Cluster EMR configurado:

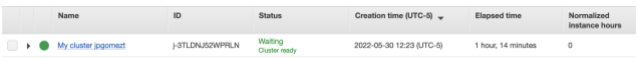


Figura 16. Cluster EMR Corriendo.

Referencias

[1] AWS. (s. f.-a). Adding Jupyter Notebook users and administrators - Amazon EMR. https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-jupyterhub-user-access.html

[2] AWS. (s. f.-b). Configuring persistence for notebooks in Amazon S3 - Amazon EMR. https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-jupyterhub-s3.html

[3] Montoya, E. N. [Edwin Nelson Montoya]. (2021a, noviembre 3). AWS EMR 6 3 1 parte1 20211103 [Vídeo]. YouTube. https://www.youtube.com/watch?v=MyXSwxN5Zdk

[4] Montoya, E. N. [Edwin Nelson Montoya]. (2021b, noviembre 3). AWS EMR 6 3 1 Parte2 20211103 [Vídeo]. YouTube. https://www.youtube.com/watch?v=3sao-qJG34Y

[5] Montoya, J. C. (2022, 25 mayo). ST0263/st0263-2022-1. GitHub. https://github.com/ST0263/st0263-2022-1