

Gestión de archivos para hdfs y S3

Santiago Alzate Cardona

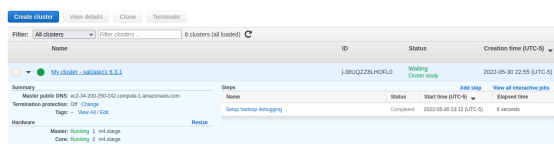
Introducción

Para esta guía se busca gestionar archivos por medio de hdfs y s3 para mantener la persistencia a través de un cluster hadoop en AWS. Se utilizarán las herramientas de Hue y SSH para hacer esta gestión de archivos y también el uso del cliente de hdfs.

Ingreso de archivos vía hue

Para ingresar archivos por hue primero debes conectarte a su interfaz gráfica, el puerto en el que está abierta la aplicación es el 8888. Puedes obtener el enlace y acceder a las aplicaciones desde la misma consola en AWS.

Busca el servicio de EMR en la consola de AWS. Dale click en clusters y selecciona el cluster que creaste.



Le damos en application user interfaces



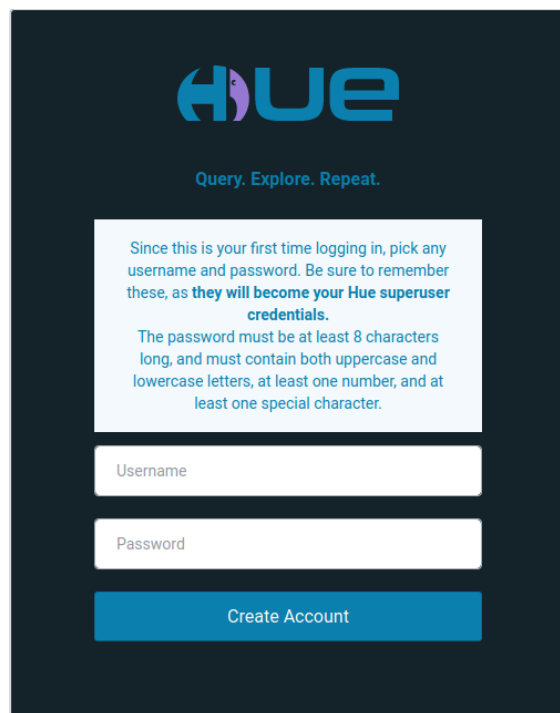
Seleccionamos el enlace de la aplicación de hue

On-cluster application user interfaces

On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster

Application	User interface URL ↗
HDFS Name Node	http://ec2-34-200-250-242.compute-1.amazonaws.com:9870/
Hue	http://ec2-34-200-250-242.compute-1.amazonaws.com:8888/
JupyterHub	https://ec2-34-200-250-242.compute-1.amazonaws.com:9443/

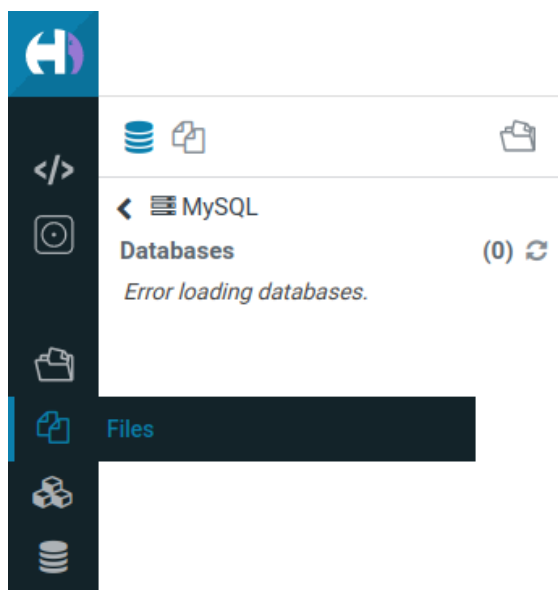
Nos aparece una página como la siguiente



Como es la primera vez que se loguea en Hue es necesario crear una nueva cuenta. En nuestro caso crearemos una cuenta con el usuario hadoop y la contraseña a tu gusto.

Ingreso de archivos a hdfs

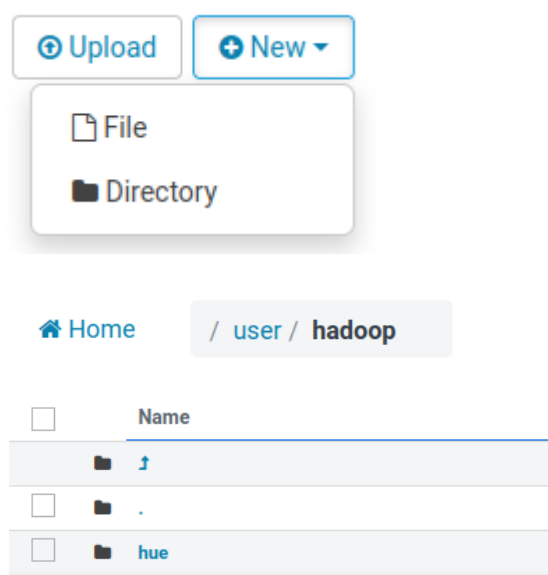
Una vez ingresado en la consola de Hue debes seleccionar files en el panel izquierdo para manejar los archivos del clúster compartidos por hdfs.



Podemos ver que el directorio principal de nuestro usuario /user/hadoop se encuentra vacío. Pero podemos agregar archivos usando la consola.

Selecciona el botón new, luego directory y creamos la carpeta llamada hue.

Aquí almacenaremos todos los archivos que subamos via hue en hdfs.



Ahora creamos una carpeta llamada datasets y subiremos todos los archivos que tengamos localmente en nuestro computador correspondientes al laboratorio.

Para subir archivos le damos en upload al lado de new y seleccionamos los archivos que subiremos.

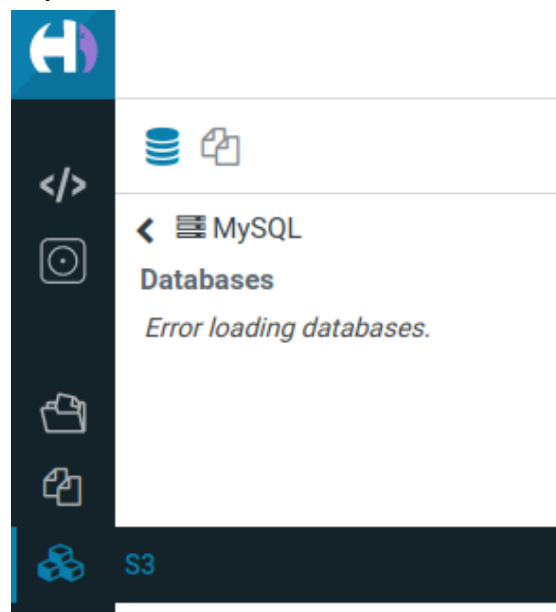
Nota: Es necesario crear una carpeta para los archivos que necesiten estar en carpetas ya que solo se pueden subir archivos.

Home	/ user / hadoop / hue / datasets				
Name	Size	User	Group	Permissions	Date
2		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
all-users	761 K B	hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
all-users		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
gutenberg		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
gutenberg-email		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
java		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
java		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
test_logs		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM
spark		hadoop	hadoop	drwxr-xr-x	May 10, 2012 09:29 PM

Ingreso de archivos a S3

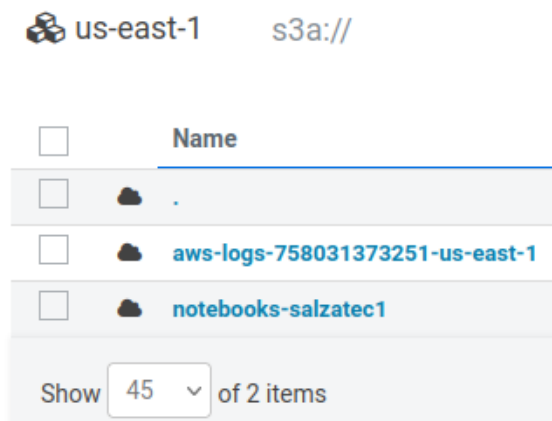
Configuración de sqoop

Seleccionamos S3 en el panel izquierdo de la consola de hue.



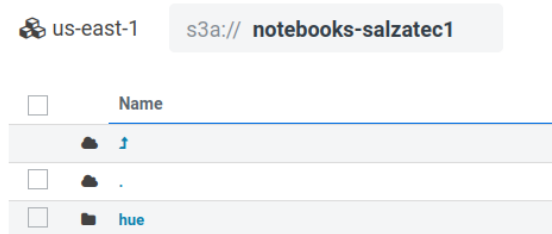
Aquí se mostrarán todos los buckets asociados al cluster.

Podremos encontrar uno de logs que sea crea por defecto cuando iniciamos el cluster y un segundo bucket que fue el que creamos en el laboratorio anterior para tener persistencia incluso cuando el cluster sea terminado. En nuestro caso llamamos el bucket notebooks-salzatec1.



Seleccionamos el bucket al que queremos subir todos los datasets, en nuestro caso notebooks-salzatec1.

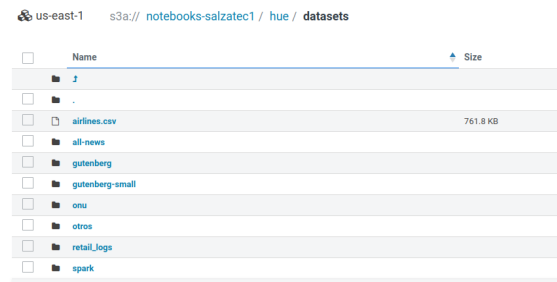
Una vez dentro le damos en new, seleccionamos directory y llamamos este nuevo directorio hue. Aquí almacenaremos todos los archivos que subamos via hue a S3.



Una vez creado nos dirigimos al directorio y le damos en upload. Desde aquí podemos seleccionar todos los archivos que deseemos subir desde nuestro computador.

En nuestro caso queremos subir la carpeta datasets.

Nota: Es necesario crear una carpeta para los archivos que necesiten estar en carpetas ya que solo se pueden subir archivos.



Ingreso de archivos vía ssh

Ingresamos a la consola de AWS. Buscamos el servicio de EMR y seleccionamos clusters. Una vez se listen todos los cluster que hayas creado seleccionas al cual esta activo y te vas a conectar via ssh. Una vez seleccionado le damos click en Connect to the Master Node Using SSH y nos abrirá una ventana que nos muestra el comando para loguearnos por ssh.

Master public DNS: ec2-34-200-250-242.compute-1.amazonaws.com
[Connect to the Master Node Using SSH](#)

Luego pegamos el comando en nuestra terminal y cambiamos ~/hadoop-cluster.pem por el directorio donde se encuentre la llave para ingresar.

En mi caso:

```
$ssh -i
~/ssh/hadoop-cluster.pem
hadoop@ec2-34-200-250-242.com
pute-1.amazonaws.com
```

```
Last Login: Tue May 31 04:12:12 UTC 2022
```

```

  _ _ | _ _ | )
 _ _ | _ _ | _ _ |
Amazon Linux 2 AMI

```

```
https://aws.amazon.com/amazon-linux-2/
49 package(s) needed for security, out of 88 available
Run "sudo yum update" to apply all updates.
```

```
EEEEEEEEEEEEEEEEEEEE MMMMMMM MMMMMMMMM RRRRRRRRRRRRRR
E:::EEEEEEEEEEEEEEEE M:::M M:::M M:::M R:::R R:::R
EE:::EEEEEEEEEEEEEE M:::M M:::M M:::M R:::R RRRRRRR:::R
E:::E EEEEE M:::M M:::M M:::M RR:::R R:::R
E:::E M:::M M:::M M:::M M:::M R:::R R:::R
E:::EEEEEEEEEEEE M:::M M:::M M:::M M:::M R:::RRRRRR:::R
E:::EEEEEEEEEE M:::M M:::M M:::M R:::RRRRRR:::R
E:::EEEEEEEEEEEE M:::M M:::M M:::M R:::RRRRRR:::R
E:::E EEEEE M:::M M:::M M:::M R:::R R:::R
EE:::EEEEEEEEEEEE M:::M M:::M M:::M R:::R R:::R
EEEEEEEEEEEEEEEEEE M:::M M:::M M:::M R:::R R:::R
EEEEEEEEEEEEEEEEEE M:::M M:::M M:::M RR:::R R:::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM MMMMMMM RRRRRRR RRRRRR
```

```
[hadoop@ip-172-31-1-132 ~]$ █
```

Una vez dentro necesitamos que sqoop funcione bien por lo que es necesario correr los siguiente comandos:

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

Este valor lo puedes almacenar en una variable que llamaremos `lib_path` así:

```
$lib path=lib 20201023180945
```

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

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

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

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

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

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

\$oozie admin -sharelibupdate

Una vez actualizado podemos correr el siguiente comando para ver que hdfs si este funcionando con hue:

```
$hdfs dfs -ls /user/hadoop
```

Debe aparecer un output como el siguiente:

```
[hadoop@ip-172-31-1-132 ~]$ hdfs dfs -ls /user/hadoop
Found 1 items
drwxr-xr-x - hadoop hdfsadmin@group 0 2022-05-31 04:29 /user/hadoop/hue
[hadoop@ip-172-31-1-132 ~]$ hdfs dfs -ls /user/hadoop/hue
Found 1 items
drwxr-xr-x - hadoop hdfsadmin@group 0 2022-05-31 04:34 /user/hadoop/hue/datasets
[hadoop@ip-172-31-1-132 ~]$
```

Así sabemos que si está funcionando.

Ahora para empezar a usar hdfs empezamos creando una carpeta llamada ssh donde guardaremos todos los datasets.

```
$hdfs dfs -mkdir /user/hadoop/ssh
$hdfs dfs -mkdir
/user/hadoop/ssh/datasets
```

Para tener los archivos localmente bajamos git y clonamos el repo correspondiente al laboratorio para obtener los datasets.

```
$sudo yum install -y git
$git clone
https://github.com/ST0263/st0263-2022-1.
git
```

Una vez bajados los datasets nos movemos al directorios de datasets y corremos los siguientes comandos.

```
$hdfs dfs -copyFromLocal
datasets/* /user/hadoop/ssh/datasets
```

Para verificar que funcionó corremos el siguiente comando:

```
$hdfs dfs -ls
/user/hadoop/ssh/datasets
```

```
[hadoop@ip-172-31-12-139 Big Data]$ hdfs dfs -ls /user/hadoop/ssh/datasets
Found 9 items
drwxr-xr-x 1 hadoop hdfsadmin 788058 2022-05-31 05:45 /user/hadoop/ssh/datasets/airlines.csv
drwxr-xr-x 1 hadoop hdfsadmin 0 2022-05-31 05:45 /user/hadoop/ssh/datasets/all-news
drwxr-xr-x 1 hadoop hdfsadmin 0 2022-05-31 05:45 /user/hadoop/ssh/datasets/gutenberg
drwxr-xr-x 1 hadoop hdfsadmin 0 2022-05-31 05:45 /user/hadoop/ssh/datasets/gutenberg-small
drwxr-xr-x 1 hadoop hdfsadmin 0 2022-05-31 05:45 /user/hadoop/ssh/datasets/otus
drwxr-xr-x 1 hadoop hdfsadmin 0 2022-05-31 05:45 /user/hadoop/ssh/datasets/otus_logs
drwxr-xr-x 1 hadoop hdfsadmin 0 2022-05-31 05:45 /user/hadoop/ssh/datasets/retail_logs
drwxr-xr-x 1 hadoop hdfsadmin 0 2022-05-31 05:45 /user/hadoop/ssh/datasets/spark
[hadoop@ip-172-31-12-139 Big Data]$ hdfs dfs -ls /user/hadoop/ssh/datasets/all-news
Found 1 items
```

Podemos ver que todos los archivos se subieron a hdfs correctamente.

En caso de querer sacar los archivos que ya estén dentro de hdfs podemos correr el siguiente comando:

```
$hdfs dfs -copyToLocal
/user/hadoop/ssh/datasets
/home/hadoop/mis_datasets
```

Verificamos que funcionó con el siguiente comando:

```
$ls -l /home/hadoop/mis_datasets
```

```
[hadoop@ip-172-31-12-139 Big Data]$ ls -l /home/hadoop/mis_datasets
total 768
-rw-r--r-- 1 hadoop hadoop 788058 May 31 05:48 airlines.csv
drwxr-xr-x 2 hadoop hadoop 21 May 31 05:48 all-news
drwxr-xr-x 2 hadoop hadoop 72 May 31 05:48 gutenberg
drwxr-xr-x 2 hadoop hadoop 4096 May 31 05:48 gutenberg-small
drwxr-xr-x 4 hadoop hadoop 98 May 31 05:48 otus
drwxr-xr-x 2 hadoop hadoop 80 May 31 05:48 otus_logs
drwxr-xr-x 2 hadoop hadoop 28 May 31 05:48 retail_logs
drwxr-xr-x 2 hadoop hadoop 91 May 31 05:48 spark
```

Ingreso de archivos a S3

Comenzamos creando un directorio ssh para el bucket S3. Para crear un directorio en s3 desde comandos con hdfs se corre el siguiente comando:

```
$hdfs dfs -mkdir
s3://notebooks-salzatec1/ssh
```

dónde s3://notebooks-salzatec1 es el nombre del bucket asociado al cluster.

Una vez creado el directorio en S3, navegamos al directorio donde esten los datasets en el master de forma local y corremos el siguiente comando:

```
$hdfs dfs -copyFromLocal datasets
s3://notebooks-salzatec1/ssh
```

Name	Size	User	Group	Permissions
2				drwxr-xr-x
airlines.csv	788,058 KB			drwxr-xr-x
all-news				drwxr-xr-x
datasets				drwxr-xr-x
gutenberg				drwxr-xr-x
gutenberg-small				drwxr-xr-x
otus				drwxr-xr-x
otus_logs				drwxr-xr-x
retail_logs				drwxr-xr-x
spark				drwxr-xr-x

Es posible que se creen algunos archivos llamados <dirname>_<folder>, esto es algo normal con hdfs pero no ha sido arreglado en ERM.

Si deseas eliminar estos directorios es posible utilizando el comando:

```
$hdfs dfs -rm -f  
s3://notebooks-salzatec1/ssh_$folder$
```

O se pueden eliminar desde el mismo hue.

Para poder sacar archivos desde s3 hacia la maquina local puedes correr el siguiente comando:

```
$hdfs dfs -copyToLocal  
s3://notebooks-salzatec1/ssh/datasets  
/home/hadoop/ssh_datasets
```

Podemos verificar que funcionó con el siguiente comando:

```
$ls -l /home/hadoop/ssh_datasets
```

```
[hadoop@ip-172-31-12-139 ~]$ ls -l /home/hadoop/ssh_datasets  
total 768  
-rw-r--r-- 1 hadoop hadoop 780058 May 31 06:13 airlines.csv  
drwxr-xr-x 2 hadoop hadoop 21 May 31 06:13 all-news  
drwxr-xr-x 9 hadoop hadoop 138 May 31 06:13 datasets  
drwxr-xr-x 2 hadoop hadoop 72 May 31 06:13 gutenberg  
drwxr-xr-x 2 hadoop hadoop 4096 May 31 06:13 gutenberg-small  
drwxr-xr-x 4 hadoop hadoop 98 May 31 06:13 onu  
drwxr-xr-x 2 hadoop hadoop 80 May 31 06:13 otros  
drwxr-xr-x 2 hadoop hadoop 28 May 31 06:13 retail_logs  
drwxr-xr-x 2 hadoop hadoop 91 May 31 06:13 spark  
[hadoop@ip-172-31-12-139 ~]$
```

Conclusiones

Crear clusters hadoop utilizando AWS es una tarea realmente sencilla. AWS nos permite abstraer muchas cosas para poder implementar y crear nuestro cluster de la manera que queramos sin preocuparnos por la instalación de esto.

También podemos ver que los clústeres de hadoop poseen herramientas muy sencillas y poderosas para manejar la persistencia como el cliente de hdfs para manejar archivos a través del cluster o conectarse con s3 para mantener persistencia incluso cuando el cluster se caiga. Algunas herramientas como hue nos permite tener esta gestión de archivos de una manera mas gráfica y contiene muchas mas funcionalidades teniendo mucho potencial para el manejo del cluster.

Referencias

- <https://github.com/ST0263/st0263-2022-1>