# EJERCICIOS

### EJERCICIO 1

En una de las máquinas donde tienes un nodo principal (master) de hdfs:

* Sube un fichero .csv a hdsf

hdfs dfs -put /home/ubuntu/Documentos/2018.csv /ejercicios

* Crea el script leer.py que lea el fichero .csv de hdfs y guarde su contenido en un dataframe.

import pandas as pd

import subprocess

def leer\_csv\_desde\_hdfs():

# Ruta del archivo en HDFS

ruta\_hdfs = '/ejercicios/2018.csv'

# Usamos subprocess para ejecutar el comando HDFS y obtener el archivo

hdfs\_command = f"hdfs dfs -cat {ruta\_hdfs}"

result = subprocess.run(hdfs\_command, shell=True, capture\_output=True, text=True)

if result.returncode == 0:

# El comando se ejecutó correctamente, leer el contenido en pandas

df = pd.read\_csv(pd.compat.StringIO(result.stdout))

return df

else:

print(f"Error al leer el archivo: {result.stderr}")

return None

if \_\_name\_\_ == "\_\_main\_\_":

df = leer\_csv\_desde\_hdfs()

if df is not None:

print(df.head())

* Crea el script escribir.py que cargue en un dataframe los datos sobre el titanic de la librería seaborno y guarde el dataframe en un .csv en el sistema de archivos hdfs.

### EJERCICIO 2

Vamos a trabajar con el clúster de Hadoop que tenemos en Ubuntu (master + 3 trabajadores)

* Asegúrate que el factor de replicación sea 2.
* Sube un fichero grande a hdfs (puedes probar con uno de los libros de la relación de ejercicios anterior).
* Muestra el contenido del fichero
* Apaga uno de los nodos trabajadores.
* Comprueba que con el nodo apagado sigues pudiendo acceder al archivo.

### EJERCICIO 3

En Amazon AWS EMR crea un clúster para Hadoop, Hue, Hive y Spark de un maestro y tres trabajadores.


     Diagrama de clústeres de Amazon EMR que muestra la relación entre los nodos principales y principales de un clúster de EMR.
    

[Amazon EMR](https://aws.amazon.com/es/emr/) es un servicio de Amazon Web Services que permite crear clústeres Hadoop y Spark, mediante el cual podemos realizar analíticas sobre datos, así como transformar y mover grandes volúmenes de datos, tanto cargando como almacenando datos en servicios de AWS como S3 y DynamoDB.

Para ello, utiliza una distribución propia de AWS que permite seleccionar los componentes que van a lanzarse en el clúster (Hive, Spark, Presto, etc.)

Ofrece elasticidad sobre el clúster, pudiendo modificar dinámicamente el dimensionamiento del clúster según necesidades, tanto hacia arriba como hacia abajo de clúster que están en ejecución.

Respecto al hardware, se ejecuta sobre máquinas EC2 (IaaS), las cuales podemos configurar según necesidades. Utiliza HDFS y S3 para el almacenamiento, de manera que podemos guardar los datos de entrada y los de salida en S3, mientras que los resultados intermedios los almacenamos en HDFS.

Los clústeres de EMR se componen de:

* **Nodo principal – Master node**: nodo que administra el clúster mediante la ejecución de componentes de software para coordinar la distribución de datos y tareas entre otros nodos para su procesamiento. El nodo principal hace un seguimiento del estado de las tareas y monitorea el estado del clúster. Cada clúster tiene un nodo principal y es posible crear un clúster de un solo nodo con solo el nodo principal.
* **Nodo secundario – Core node – Node central**: un nodo con componentes de software que ejecutan tareas y almacenan datos en el Hadoop Distributed File System (HDFS) del clúster. Los clústeres de varios nodos tienen al menos un nodo secundario.
* **Nodo de tareas**: un nodo con componentes de software que solo ejecuta tareas y no almacena datos en HDFS. Los nodos de tareas son opcionales.

A nivel de servicios, podemos definir su arquitectura en cuatro capas:

* Almacenamiento: mediante HFDS o el sistema de archivos local (almacenamiento de las instancias EC2).
* Gestor de recursos del clúster: YARN
* Frameworks de procesamiento de datos: Hadoop MapReduce y Apache Spark
* Aplicaciones: Apache Spark, Apache Hive, etc...

Sigue las siguientes instrucciones para crear el clúster que necesitamos:

Podemos lanzar un clúster de EMR de tres formas:

* Consola (es el que vamos a usar)
* CLI
* Con un API.

Dentro de AWS vamos a EMR y hacemos clic en crear clúster

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Elegimos el nombre del clúster y los paquetes de aplicaciones que queremos instalar.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Elegimos los tipos de instancia (máquinas virtuales) para el nodo principal y para los nodos secundarios.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Podemos ajustar el número de instancias EC2 con las que trabaja nuestro clúster, ya sea manualmente o de forma automática en respuesta a la demanda que reciba.

En este caso elegimos un tamaño fijo de 3 nodos secundarios.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Por defecto, se crea un grupo de seguridad para el nodo maestro el cual determina el acceso y otro para los nodos secundarios.

Por defecto, este grupo de seguridad no permite conexiones desde el exterior y para poder acceder por ssh o vía web a algunas de las aplicaciones necesitaremos abrir ciertos puertos. Pero por defecto, EMR bloquea el arranque de los clústeres que permitan las conexiones del exterior.

Por lo tanto, podemos arrancar un clúster con el tráfico cerrado, y una vez ya ha arrancado cambiar su grupo de seguridad, o desactivar la opción de Bloquear el acceso público de EMR.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Normalmente, cuando utilizamos un clúster para procesar analíticas, interactuar con aplicaciones de big data o procesamiento de datasets de forma periódica, el clúster está siempre en marcha, a no ser que lo detengamos nosotros de forma explícita. Pero si queremos que sólo exista durante la ejecución de uno o más trabajos, el cual se le conoce como clúster transient o de ejecución por pasos, al terminar de ejecutar los pasos indicados, el clúster se detendrá.

El almacenamiento hdfs va a ser volátil, es decir cuando se termine el clúster vamos a perder la información, la información que queremos que persista hay que guardarla en un contenedor S3 y guardar temporalmente la información que necesitamos procesar en hdfs.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Necesitamos una clave para acceder a las máquinas (elegimos la que nos ha creado aws para nuestro usuario).

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ahora hacemos clic en crear clúster y tardará unos 10min en estar activo nuestro clúster.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Lo primero que vamos a hacer es abrir los puertos necesarios para ello editamos los grupos de seguridad para añadir nuevas reglas de entrada:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* ElasticMapReduce-Primary: (el grupo de seguridad del nodo principal)
  + Reglas de entrada:
    - Puerto 8088
    - Puerto 9870
    - Puerto 18080
    - Puerto 8888
    - Puerto 22

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En la pestaña aplicaciones podemos acceder a la gestión web de las aplicaciones que hemos instalado:

* Administración de recursos (YARN)
* Nombre del nodo de HDFS (Namenode de HDFS)
* Servidor de historial de Spark
* Tonalidad (Hue, mal traducido a Tonalidad)

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Parece que funciona pero...**

Una vez arrancado, tenemos un par de propiedades que modificar.

Vamos a EC2 y nos conectamos vía SSH a la máquina donde está el nodo principal con el usuario hadoop.

Para poder visualizar el sistema de archivos de HDFS, tenemos que cambiar en hdfs-site.xml la propiedad dfs.webhdfs.enabled a true:

*sudo nano /etc/hadoop/conf/hdfs-site.xml*

**/etc/hadoop/conf/hdfs-site.xml**

*<property>*

*<name>dfs.webhdfs.enabled</name>*

*<value>true</value>*

*</property>*

Reiniciamos el servicio de HDFS:

*sudo systemctl restart hadoop-hdfs-namenode*

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

### EJERCICIO 4

Sube un par de archivos al sistema de ficheros hdfs desde la terminal.

### EJERCICIO 5

Haz el ejercicio 4 de la relación de ejercicios anterior (Ejercicio 4 de E2\_4-ejercicios.docx) conectándote directamente por ssh a las máquinas virtuales del clúster

### EJERCICIO 6

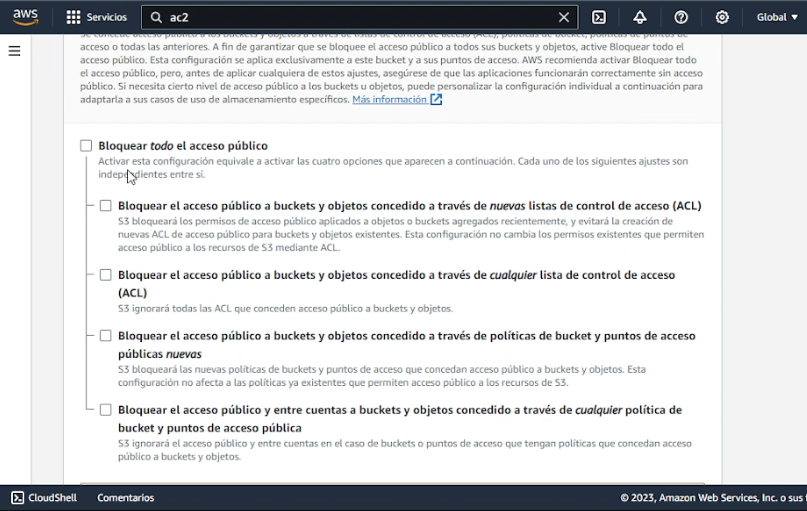
Repite el ejercicio 5 creando los pasos en Amazon EMR.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

NOTA: Usar contenedor creado para registros en la generación de clúster o crear un contenedor S3 público.

Si se va a crear un nuevo contenedor, la carpeta tiene que ser pública. Cambiar la siguiente propiedad.



Hau duen irudia testua, pantaila-argazkia, Bataiarri, zenbakia

Azalpena automatikoki sortu da

### EJERCICIO 7

A continuación, añade un cuarto nodo secundario al clúster y sube un archivo al sistema de ficheros hdfs. Tras ello, vuelve a cambiar el tamaño del clúster para que tenga otra vez sólo tres nodos secundarios. Comprueba que el archivo sigue estando disponible.

### EJERCICIO 8

Crea otro clúster sólo con Hadoop y haz que el tamaño del clúster varie según la demanda de trabajo: mínimo 3 nodos secundarios y máximo 6.

Configura el factor de replicación para que sea 3.

### EJERCICIO 9 (opcional)

Accede con Hue al sistema de ficheros HDFS y sube algún fichero