UD 6 - Apache Spark - Plataformas

1. Databricks

1.1 ¿Qué es Databricks?

Databricks es el nombre de la plataforma analítica de datos basada en Apache Spark desarrollada por la compañía con el mismo nombre. La empresa se fundó en 2013 con los creadores y los desarrolladores principales de Spark. Permite hacer **analítica Big Data e inteligencia artificial con Spark de una forma sencilla y colaborativa**.

Esta plataforma está disponible como servicio cloud en Azure, AWS y Google Cloud.

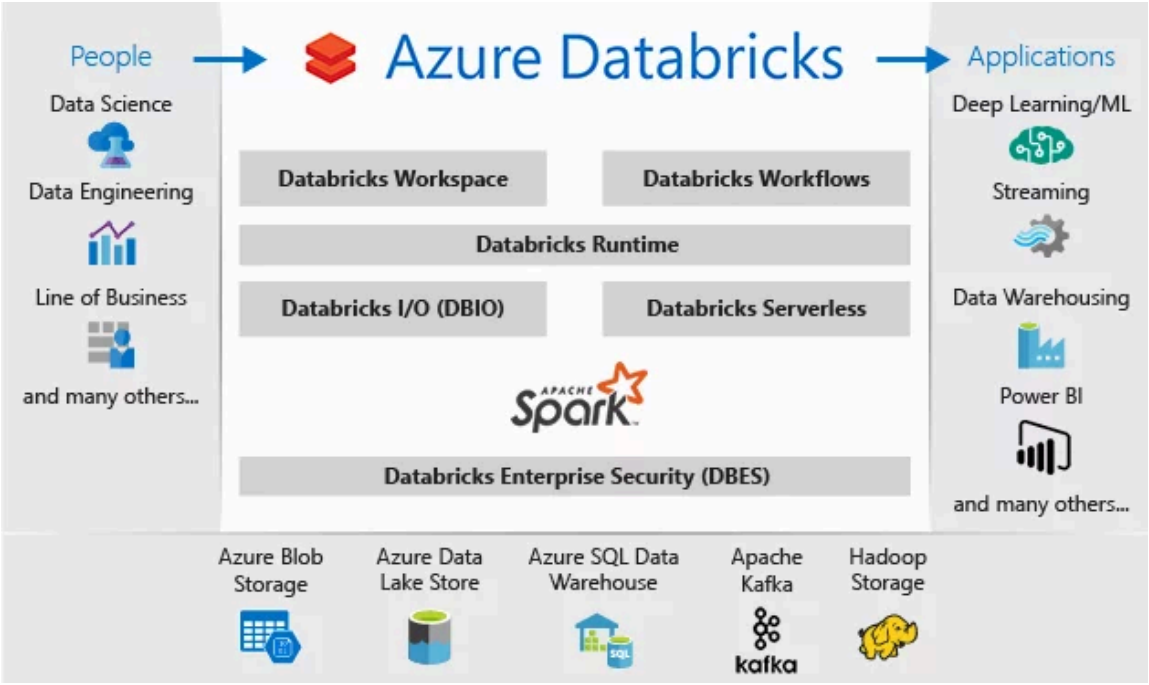

Figura 6.1_Plataformas: Azure Databricks. (Fuente: aprenderbigdata.com)

Databricks contiene muchas funcionalidades que la hacen una solución analítica bastante completa. Aun así, depende de servicios adicionales como almacenamientos externos de datos para poder convertirse en la pieza central de un sistema analítico empresarial completo como Data Warehouse o Data Lake.

Es una plataforma que permite múltiples casos de uso como **procesamiento batch, streaming y machine learning**.

1.2 Características de Databricks

Permite **auto-escalar y dimensionar entornos de Apache Spark de forma sencilla** en función de las necesidades. También es posible terminar automáticamente estos clústers. De esta forma, se facilitan los despliegues y se acelera la instalación y configuración de los entornos. Con la opción *serverless* se puede abstraer toda la complejidad alrededor de la infraestructura y obtener directamente acceso al servicio. Así se facilita su uso por equipos independientes que necesiten recursos volátiles y despliegues ad-hoc.

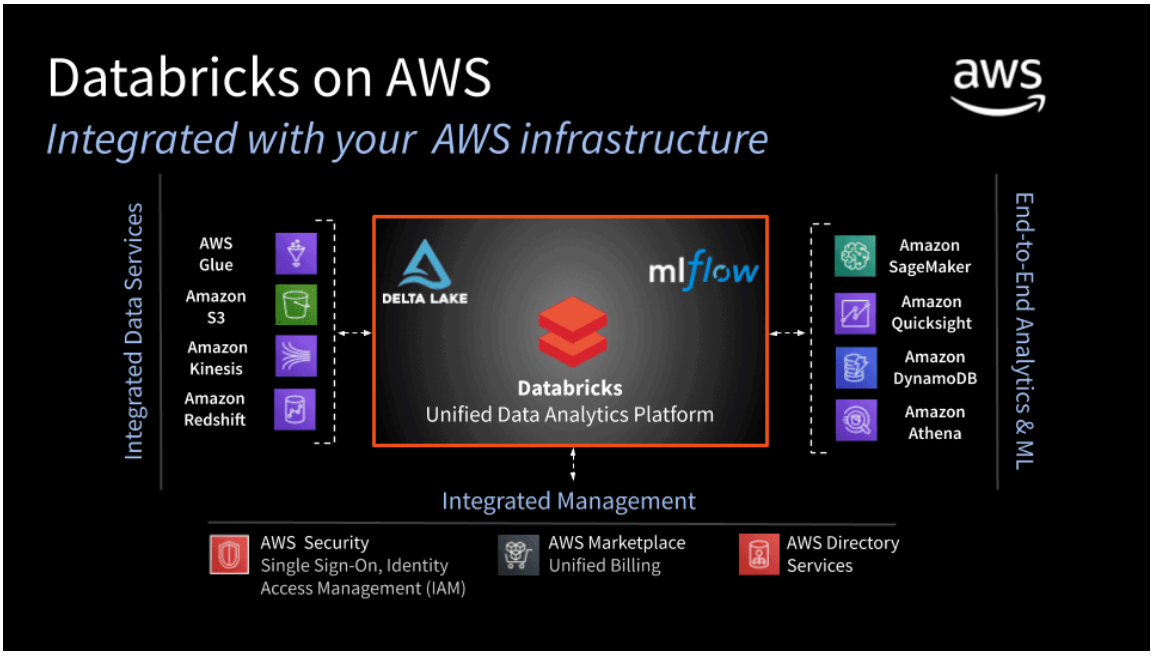

Figura 6.2_Plataformas: AWS Databricks. (Fuente: aprenderbigdata.com)

Incluye **proyectos colaborativos y espacios de trabajo interactivos llamados notebooks**. Estos pueden servir para desarrollar procesos y prototipos de transformación y análisis y más adelante ponerlos en producción con el planificador. **Están integrados con sistemas de control de versiones como Github y Bitbucket y es posible crear directorios separados para diferentes unidades o equipos**.

Databricks no es responsable de la capa de persistencia de los datos. Esto quiere decir que los datos se procesan con Spark pero antes deben estar almacenados en algún componente adicional (Azure Blob Storage, Amazon S3, ADLS (Azure Data Lake Storage), Azure SQL Data Warehouse, etc).

1.3 Databricks Community

Databricks community es la versión de Databricks **gratuita**. Permite usar un pequeño clúster con recursos limitados y notebooks no colaborativos. La versión de pago no tiene estas limitaciones y aumenta las capacidades.

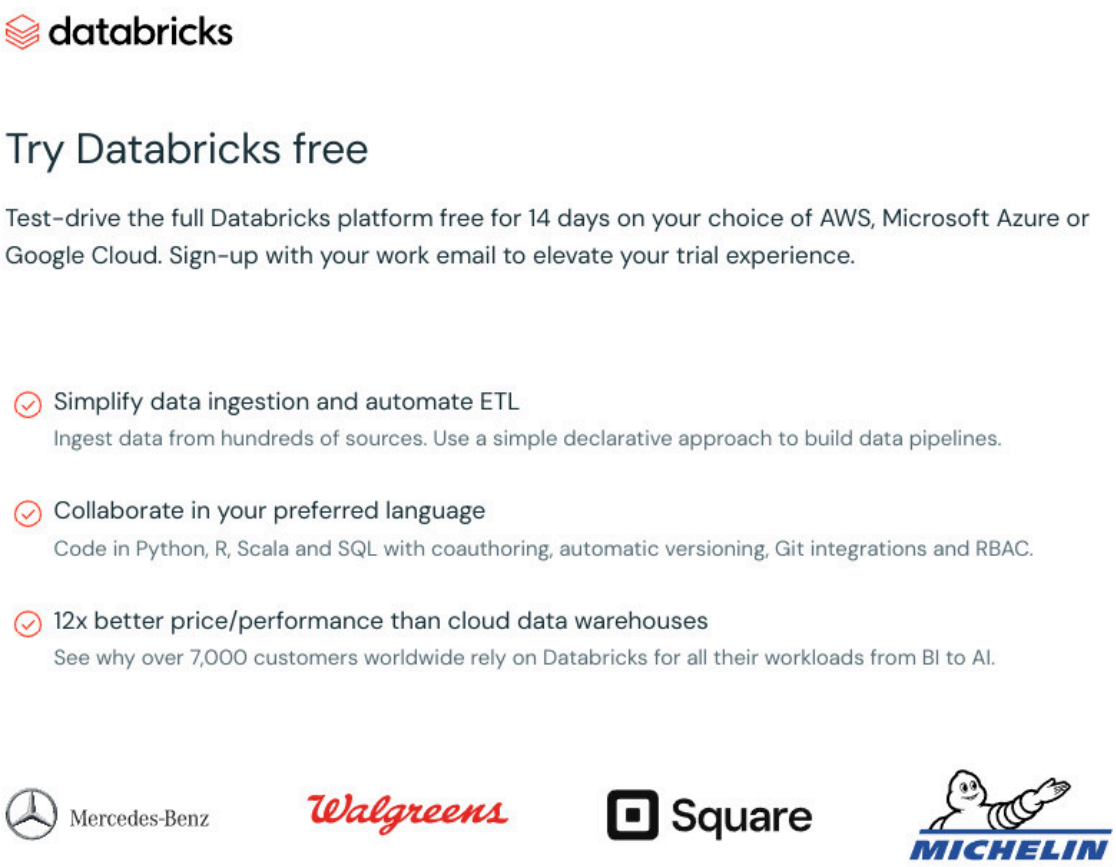

Figura 6.3_Plataformas: Databricks Community 1 (Fuente: aprenderbigdata.com)

Para crear una cuenta gratuita, hacemos click sobre **Sign up**, tras rellenar los datos personales, antes de seleccionar el proveedor cloud, en la parte inferior, hemos de pulsar sobre **Get started with Community Edition**:

Una vez hemos hecho login en la plataforma, nos permitirá hacer un tutorial rápido que nos explica la funcionalidad básica:

- Crear un clúster de Spark
- Asociar notebooks al clúster y ejecutar comandos
- Crear tablas de datos
- Hacer consultas y visualizar los datos
- Manipular y transformar los datos

1. El primer paso es crear un nuevo clúster. Esto se puede hacer desde la pestaña clusters. Nos permite elegir el nombre y la versión del runtime. En este caso elegimos 15.4 LTS. La versión Community crea un clúster con un driver de 15GB de RAM.

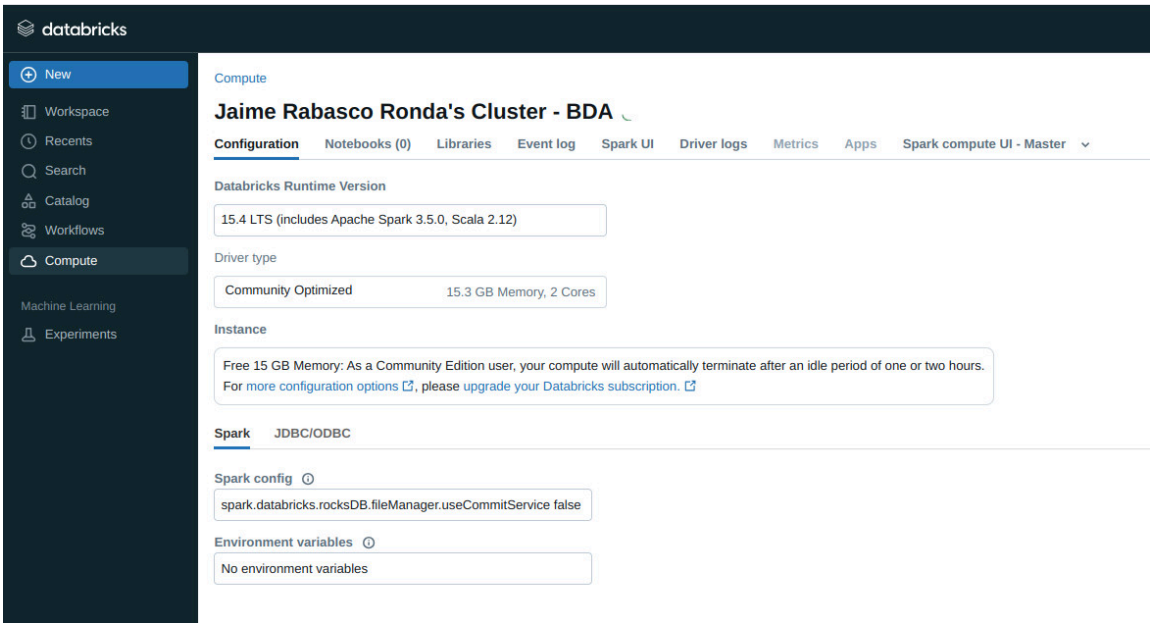

Figura 6.4_Plataformas: Databricks Community. Crear un cluster

> **Warning**
>
> *Databricks* finaliza el cluster automáticamente si no tiene actividad en 2 horas

2. Creamos cluster y esperamos que se despliegue
3. Con Databricks Community no podemos ejecutar jobs de Spark desde ficheros JAR, solamente desde notebooks.
4. Una vez desplegado el cluster, ya podemos crear un notebook (`create -> notebook`) que se conecta al cluster que acabamos de desplegar. Si no lo hace automáticamente, lo seleccionamos nosotros.
5. En el notebook ya tenemos acceso a Spark mediante el objeto `spark`

Figura 6.5_Plataformas: Databricks Community. Objeto Spark

1. Podemos ejecutar PySpark a través del notebook

```
data = [1, 2, 3, 4, 5, 6 ,7, 8]
distData = sc.parallelize(data)
distData.sum()
```

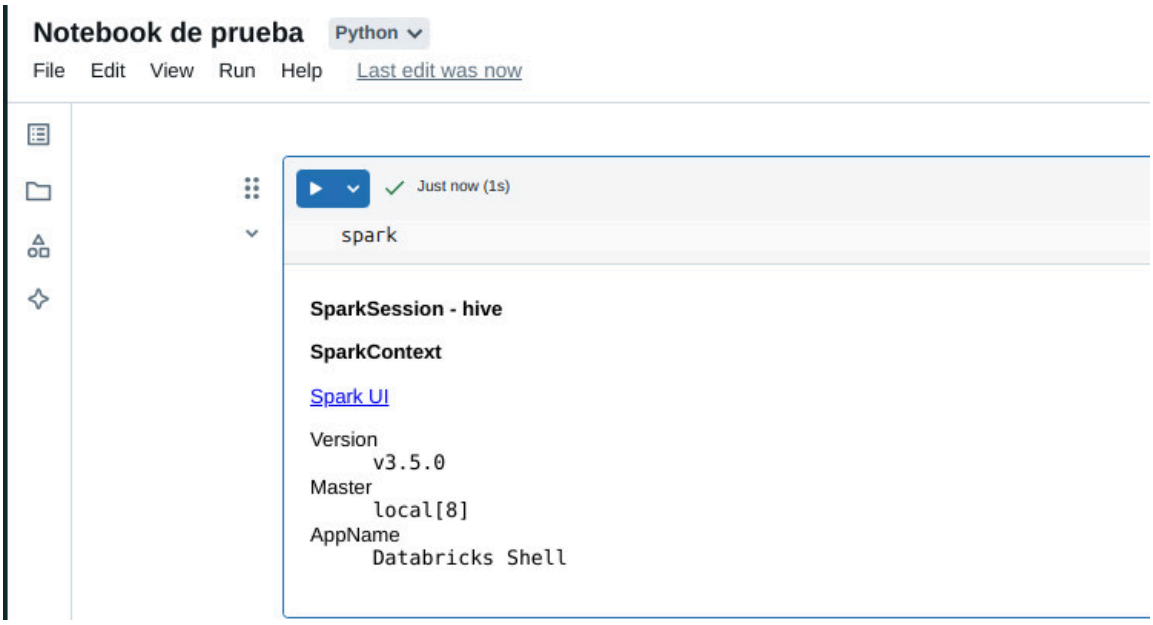

Figura 6.6_Plataformas: Databricks Community. Ejecutando PySpark

7. Podemos observar la webui haciendo click en las `views`

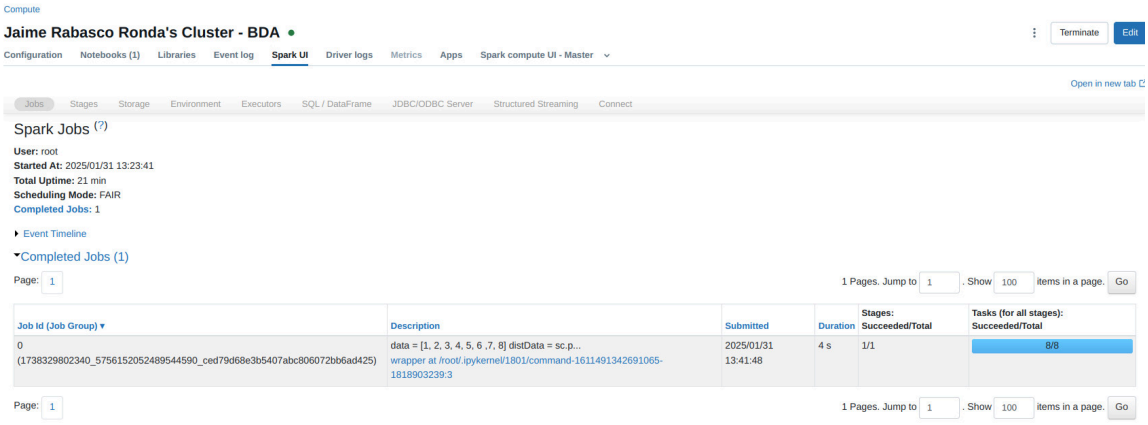

Figura 6.7_Plataformas: Databricks Community. Vistas Spark WebUI

8. Por defecto, los notebooks son ocultos. Podemos publicar nuestro notebook pulsando el botón `publish`. Estarán disponibles **durante 6 meses**.

2. Jupyter

2.1 ¿Qué es Jupyter Notebook?

Jupyter Notebook es una aplicación web de código abierto que proporciona un entorno computacional interactivo. Produce documentos (notebooks) que combinan entradas (código) y salidas en un solo archivo. En un único documento ofrece:

- Visualizaciones
- Ecuaciones matemáticas
- Modelado estadístico
- Texto narrativo
- Cualquier otro medio enriquecido

Este enfoque de documento único permite a los usuarios desarrollar, visualizar los resultados y agregar información, gráficos y fórmulas que hacen que el trabajo sea más comprensible, repetible y compartible.

Los Jupyter Notebooks admiten más de 40 lenguajes de programación, con especial atención en Python. Dado que es una herramienta gratuita y de código abierto, cualquiera puede utilizarla libremente para sus proyectos de **ciencia de datos**. Hay dos variantes del cuaderno Jupyter:

- **Jupyter Classic Notebook**, con todas las capacidades mencionadas anteriormente.
- **JupyterLab**, una nueva interfaz de notebooks diseñada para ser mucho más extensible y modular, con soporte para una amplia variedad de flujos de trabajo desde **data science, machine learning, and scientific computing**.

2.2 Instalación y configuración

Sólo es necesaria esta configuración en la/s máquina/s donde vayas a lanzar Spark

1. Asegúrate de tener instalado jupyter

```
sudo apt install python3 python3-pip python3-venv # Si no tienes python instalado
sudo apt install python3-pip #Si no tienes pip instalado
pip3 install jupyter
```

2. Puedes comprobar la correcta configuración lanzando un notebook

```
jupyter prueba_lab
```

3. Configurar las variables de entorno en `~/.bashrc`

```
export PYSPARK_DRIVER_PYTHON=jupyter
```

```
2  export PYSPARK_DRIVER_PYTHON_OPTS='notebook --ip=0.0.0.0' # Para permitir acceso desde fuera del localhost. Puedes indicar un io determinada o desde cualquier ip (0.0.0.0)
```

4. Recuerda hacer ejecutar `source ~/.bashrc`

5. Ejecutamos `pyspark`

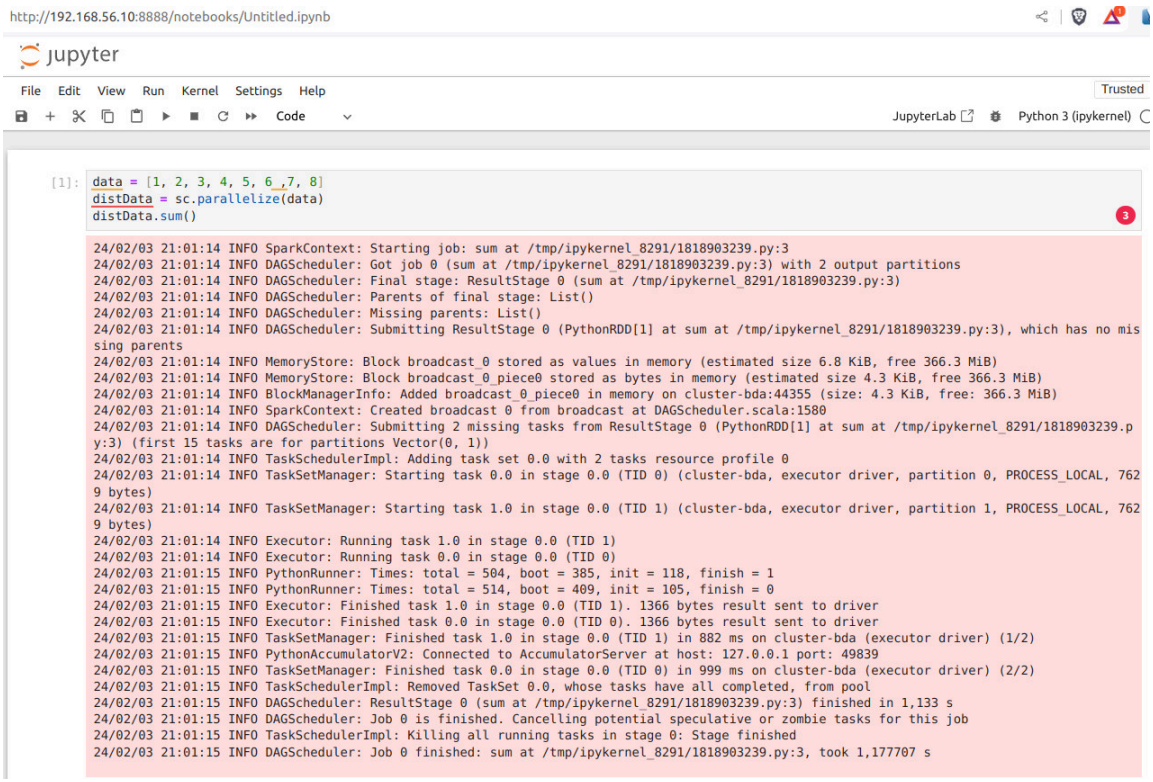

Figura 6.8_Plataformas: PySpark sobre Jupyter