1. Introducción

Actualmente, existen tres grandes proveedores de servicios cloud; todos ellos con una versión de prueba por tiempo limitado o con restricciones en la funcionalidad:

- **Amazon Web Services (AWS):** `https://aws.amazon.com/es/free/`
- **Microsoft Azure:** `https://azure.microsoft.com/es-es/free/`
- **Google Cloud:** `https://cloud.google.com/free?hl=es`

Los proveedores de servicios cloud ofrecen espacio de almacenamiento, gestores de bases de datos, potencia computacional, aplicaciones de monitorización y generación de informes de uso (dashboards), e, incluso, herramientas para la configuración y despliegue semiautomático de clústers preconfigurados con herramientas de análisis (p.ej., Apache Hadoop, Apache Spark o Apache Storm).

En esta primera práctica, trabajaremos con Apache Spark. Para ello, crearemos un mini cluster virtual en la nube, donde lanzaremos las tareas. Los beneficios de desplegar un cluster virtual en un servidor cloud son variados:

- **Facilidad de uso**: Las herramientas cloud se encargan del aprovisionamiento, la configuración y el ajuste de las máquinas virtuales, dockers y servidores en los clusteres para que los ingenieros y científicos de datos puedan concentrarse en ejecutar análisis. Solo se debe especificar la versión de las aplicaciones (p.ej., Apache Spark) y el tipo de máquina que se desea usar (p.ej, 2 vCPU). Los analistas, los ingenieros de datos y los científicos de datos pueden compartir Notebooks para colaborar y analizar, procesar y visualizar datos de manera interactiva.
- **Bajo costo**: Los precios son predecibles y reducidos: se paga una tarifa por instancia por tiempo de uso. Esto permite a las empresas ahorrarse la creación y mantenimiento de un cluster propio de servidores.
- **Elasticidad**: A diferencia de la infraestructura rígida de los clústeres en las instalaciones propias, los servicios cloud desacoplan la informática y el almacenamiento, lo que le brinda la posibilidad de escalar cada uno de forma independiente y de aprovechar el almacenamiento. Incluso se permite aumentar o reducir la cantidad de instancias automáticamente en función de la demanda y necesidades por parte del cliente.
- **Fiabilidad**: Los sistemas cloud permiten monitorizar el clúster constantemente, relanzar las tareas con errores y sustituir de forma automática las instancias que tienen un rendimiento deficiente. Los clústeres tienen una disponibilidad alta, realizan copias de seguridad periódicas y conmutan los trabajos automáticamente de un nodo a otro en caso de error o fallo. Además, proporcionan las últimas versiones estables del software, por lo que no es necesario que el usuario administre manualmente las actualizaciones ni las correcciones de errores, lo que implica menos problemas y esfuerzo de mantenimiento.

- **Seguridad**: Se establecen automáticamente los ajustes del firewall para controlar el acceso de red a las instancias. Se pueden crear cluster virtuales privados, además de ofrecer la posibilidad de cifrar el servidor, el cliente o habilitar otras opciones de cifrado y autentificación, como el cifrado en tránsito y en reposo.
- **Flexibilidad**: Permiten la personalización del entorno de ejecución para los trabajos individuales, especificando las bibliotecas y las dependencias del tiempo de ejecución (p.ej., mediante la scripts de configuración para la instalación de software, Dockers, etc.).

En el caso de los tres grandes proveedores de servicios web, estas herramientas para la gestión y despliegue de cluster virtuales en la nube son:

- **Amazon EMR:** `https://aws.amazon.com/es/emr/`
- **Azure HDInsight:** `https://azure.microsoft.com/en-us/services/hdinsight/`
- **Google Dataproc:** `https://cloud.google.com/dataproc?hl=es`

2. Databricks

En nuestro caso, utilizaremos Databricks Community Edition, una plataforma gratuita de ámbito académico proporcionada por la empresa Databricks. El primer paso será crearnos una cuenta a través de la siguiente página web:

`https://community.cloud.databricks.com/login.html`

A lo largo de esta práctica, nos familiarizaremos con el entorno cloud para gestión y configuración de clusters virtuales; y manejaremos varios ejemplos de prueba sencillos de Apache Spark. En una primera instancia, trabajaremos los conceptos básicos (Resilient Distributed Datasets, DataFrames, operaciones de MapReduce, etc.) y continuaremos con los complementos de Spark SQL, Apache Streaming y Spark ML (machine learning) que se montan como capas encima del core de Apache Spark. Para ello, utilizaremos una serie de Noebooks básicos extraídos de:

Apache Spark Tutorial: Getting Started with Apache Spark on Databricks

`https://databricks.com/spark/getting-started-with-apache-spark`

Y algunos ejemplos básicos adicionales, según el tipo de operación, sacados de:

Doc: `https://sparkbyexamples.com/`

Fuentes: `https://github.com/spark-examples/pyspark-examples/`

2.1. Notebooks Básicos

- **Get started with Apache Spark**

https://docs.databricks.com/getting-started/spark/quick-start.html

https://docs.databricks.com/getting-started/spark/quick-start.html#notebooks

- **DataFrames tutorial**

  https://docs.databricks.com/getting-started/spark/dataframes.html

  https://docs.databricks.com/getting-started/spark/dataframes.html#dataframe-notebook

- **Datasets tutorial**

  https://docs.databricks.com/getting-started/spark/datasets.html

  https://docs.databricks.com/getting-started/spark/datasets.html#notebook

- **Machine learning tutorial**

  https://docs.databricks.com/getting-started/spark/machine-learning.html

  https://docs.databricks.com/getting-started/spark/machine-learning.html#notebook

- **Streaming**

  https://docs.databricks.com/getting-started/spark/streaming.html

  https://docs.databricks.com/getting-started/spark/streaming.html#notebook

2.2. Notebooks Avanzados: Genome

https://docs.databricks.com/applications/genomics/index.html https://glow.readthedocs.io/en/latest/getting-started.html#demo-notebook

2.3. Configuración

https://docs.databricks.com/clusters/index.html
https://docs.databricks.com/clusters/configure.html
https://docs.databricks.com/dev-tools/databricks-connect.html

3. Material adicional

Spark: The Definitive Guide
https://www.oreilly.com/library/view/spark-the-definitive/9781491912201/
https://github.com/databricks/Spark-The-Definitive-Guide/tree/master/code