

# DATA STREAMING Y SERVICIOS EN LA NUBE

## DATABRICKS

Magister - Efraín Alberto Oviedo  
[alberto.oviedo@udea.edu.co](mailto:alberto.oviedo@udea.edu.co)

**UNIVERSIDAD DE ANTIOQUIA**  
**FACULTAD DE INGENIERÍA**

**ESPECIALIZACIÓN EN ANALÍTICA Y CIENCIA DE DATOS**



**UNIVERSIDAD  
DE ANTIOQUIA**

1 8 0 3

# AGENDA

## **1. Databricks**

### 2. Cluster Spark

- SQL
- R
- Python
- Scala

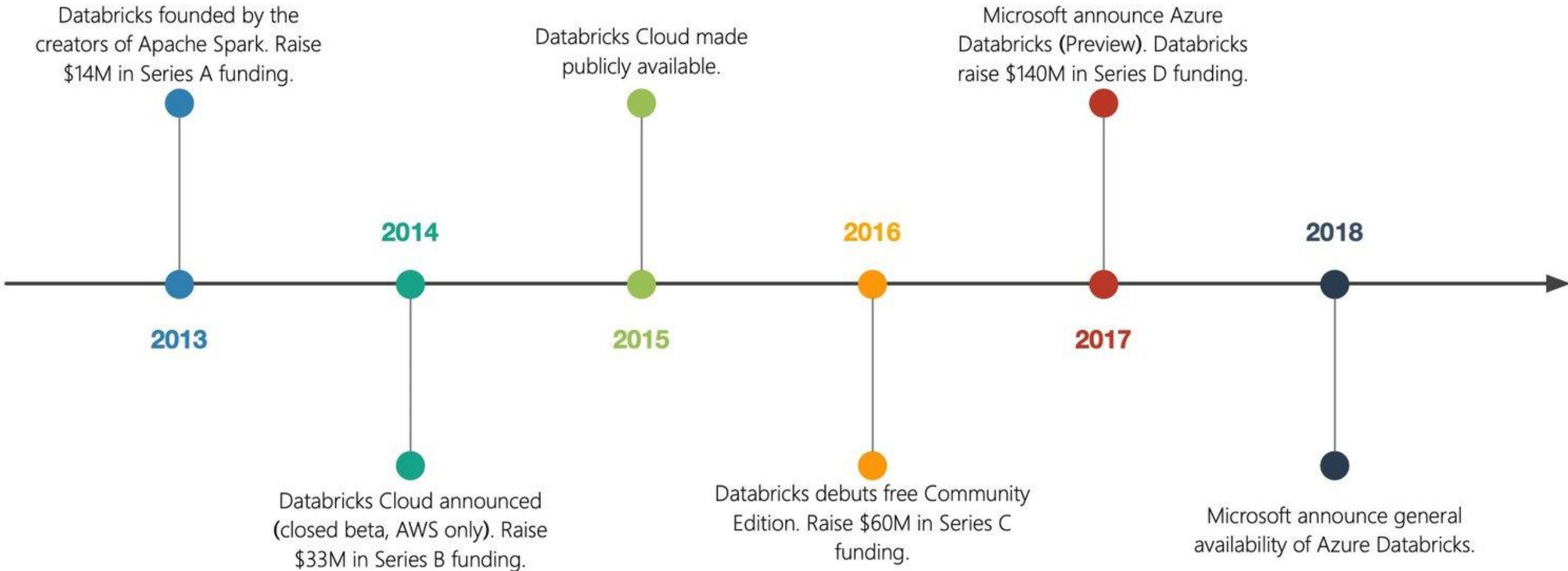
### 3. Streaming

# Qué es Databricks

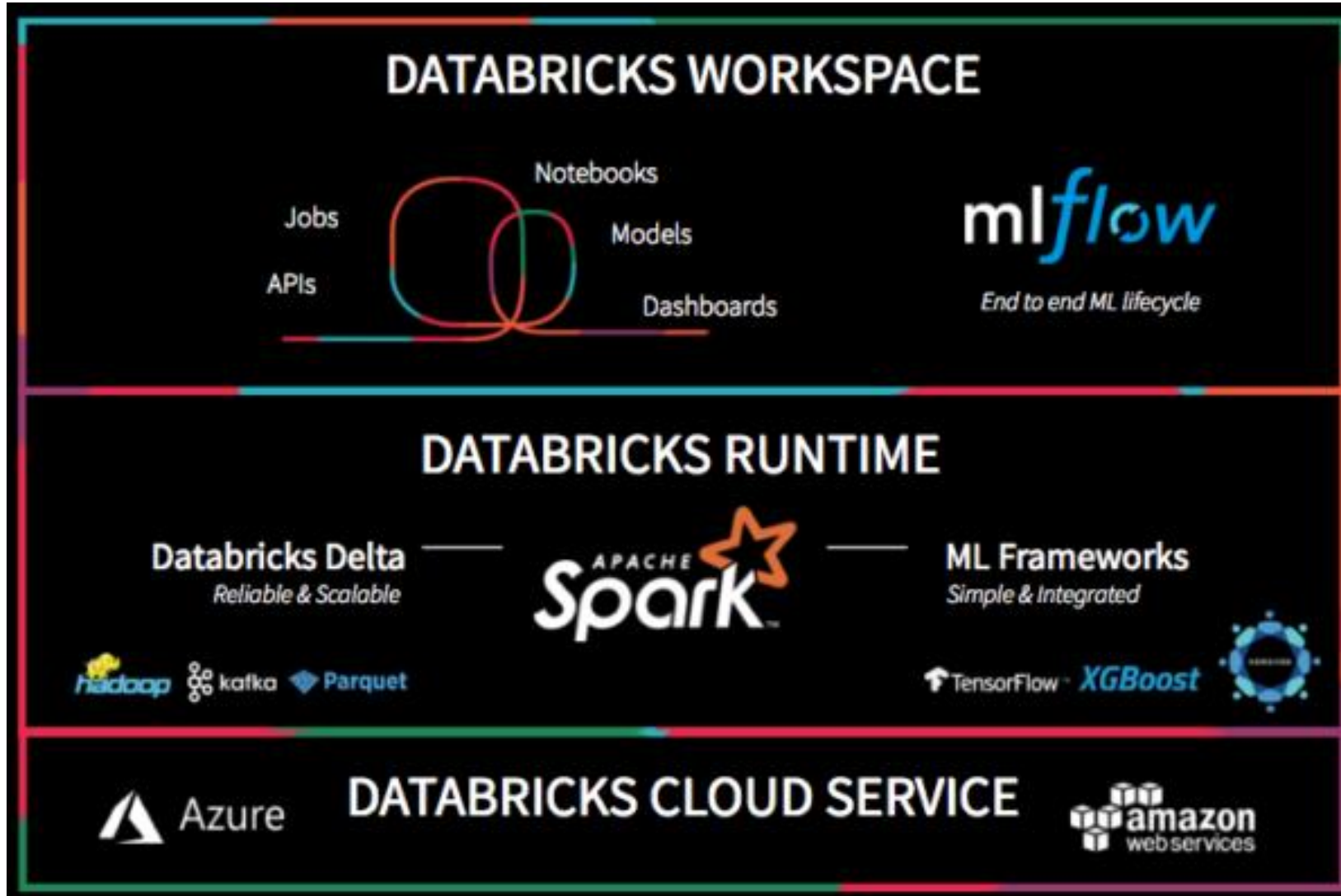
- Plataforma unificada de analítica de datos desarrollada por los creadores de Apache Spark
- Compatible con lenguajes:  
Python – SQL – Scala – R
- Servicio de Clúster Spark en la nube
  - Disponible en versión community
  - Compatible con:



# Historia Databricks



# Componentes Databricks



# Delta Lake



- Capa de almacenamiento open Source desarrollada por Databricks
- Proporciona transacciones ACID:
  - Atomicidad: Todas las transacciones tienen éxito o fallan por completo
  - Consistencia: Cada cambio debe conducir a un estado válido
  - Aislamiento: Resuelve conflictos en las operaciones simultáneas
  - Durabilidad: Cambios permanentes
- Permite consultas interactivas rápidas
- Control escalable de metadatos
- Permite operaciones de actualización y eliminación de registros

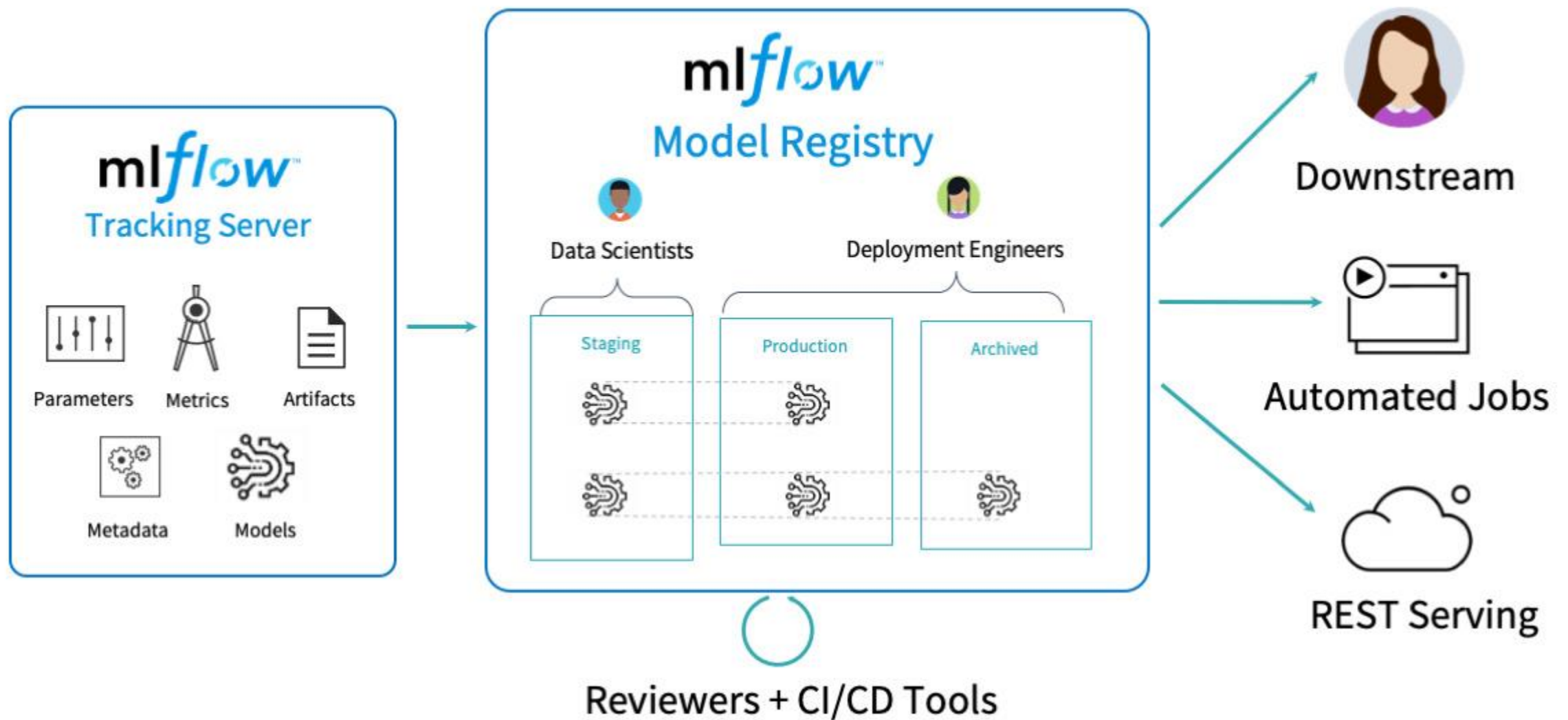


Figure 1: Magic Quadrant for Data Science and Machine Learning Platforms



# Plataformas de Ciencia de Datos

<https://www.databricks.com/blog/2021/03/04/databricks-named-a-leader-in-2021-gartner-magic-quadrant-for-data-science-and-machine-learning-platforms.html>

Source: Gartner (March 2021)



# Casos de Uso

## Healthcare & Life Sciences



## Financial Services



## Media & Entertainment



## Retail, CPG, and eCommerce



## Public Sector



## Manufacturing



## Technology



## Other



# Casos de Uso en América Latina



# Links de Apoyo

- Documentación Oficial

<https://docs.databricks.com/introduction/index.html>

- Databricks Community

<https://community.databricks.com>


- Azure Databricks

<https://learn.microsoft.com/es-es/azure/databricks/getting-started/free-training?source=recommendations>

- AWS

<https://aws.amazon.com/es/quickstart/architecture/databricks/>

# Crear Cuenta Databricks

1. Ingrese a la página oficial de Databricks Community  
<https://www.databricks.com/try-databricks#account> 
2. Llene el formulario con los datos personales
3. Indicar su proveedor de nube, en este caso seleccione la **Community Edition**
4. Revise su correo electrónico y verifique la cuenta

# Crear Cuenta Databricks



## Try Databricks free

Test-drive the full Databricks platform free for 14 days on your choice of AWS, Microsoft Azure or Google Cloud.

- ✓ Simplify data ingestion and automate ETL  
Ingest data from hundreds of sources. Use a simple declarative approach to build data pipelines.
- ✓ Collaborate in your preferred language  
Code in Python, R, Scala and SQL with coauthoring, automatic versioning, Git integrations and RBAC.
- ✓ 12x better price/performance than cloud data warehouses  
See why over 7,000 customers worldwide rely on Databricks for all their workloads from BI to AI.

2

## Create your Databricks account

1/2

First name

Last name

Email

Company

Title

Phone (Optional)

Country

By submitting, I agree to the processing of my personal data by Databricks in accordance with our [Privacy Policy](#). I understand I can [update my preferences](#) at any time.

Continue

## Choose a cloud provider



Amazon Web Services



Microsoft Azure



Google Cloud Platform

Get started

By clicking "Get started", you agree to the [Privacy Policy](#) and [Terms of Service](#)

### Don't have a cloud account?

Community Edition is a limited Databricks environment for personal use and training.

#### [Get started with Community Edition](#)

By clicking "Get started with Community Edition", you agree to the [Privacy Policy](#) and [Community Edition Terms of Service](#)

# Check your email to start your trial.

Thank you for signing up. Please validate your email address to start your trial.

Here are some resources to help you deploy your first workspace.

1. [Review the administration guide](#) on the requirements to set up your Databricks service.
  - Not an admin on your AWS Account? Share [this guide](#) with your admin to deploy a workspace for you!
2. [Follow our Quickstart guide to create your first workspace.](#)

You can also check out our [Docs](#) and [Community](#) sites to get your questions answered.

**Note:** if you signed up for Community Edition, you'll go to your first workspace as soon as you verify your email address.

Welcome to Databricks! Please verify your email address. ➤ Recibidos x

Databricks <noreply@databricks.com>  
para efrain.oviedo ▾

🌐 inglés ▾ > español ▾ [Traducir mensaje](#)



## Welcome to Databricks Community Edition!

Databricks Community Edition provides you with access to a free micro-cluster as well as a cluster manager and a notebook environment - ideal for developers, data scientists, data engineers and other IT professionals to get started with Spark.

We need you to verify your email address by clicking on [this link](#). You will then be redirected to Databricks Community Edition!

4

Your sign-in email: [efrain.oviedo@upb.edu.co](mailto:efrain.oviedo@upb.edu.co)

Get started by visiting: <https://community.cloud.databricks.com/login.html?resetpassword&username=efrain.oviedo%40upb.edu.co&expiration=-60000&token=2555d74de4b7987b9b3bd6b9a76d3d47a0da246a&accountid=0fbe09ab-8085-4272-bf9b-2cb68bf0e02a>

If you have any questions, please contact [feedback@databricks.com](mailto:feedback@databricks.com).

- The Databricks Team



## Sign In to Databricks Community Edition




[Forgot Password?](#)


Sign In


New to Databricks? [Sign Up](#).





# Interfaz Databricks


 **databricks**


 **Data Science & Engi...** ▼


 **Create**


 **Workspace**

 **Recents**

 **Search**

 **Data**

 **Compute**

 **Workflows**

## Data Science & Engineering



### Notebook

Create a new notebook for querying, data processing, and machine learning.

[Create a notebook](#)



### Guide: Quickstart tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

[Start tutorial](#)



### Data import

Quickly import data, preview its schema, create a table, and query it in a notebook.

[Browse files](#)



### Transform data

[Delta Live Tables](#)

[dbt Core](#)



### AutoML

Quickly train ML models for discovery and iteration.

[Start AutoML](#)



# AGENDA

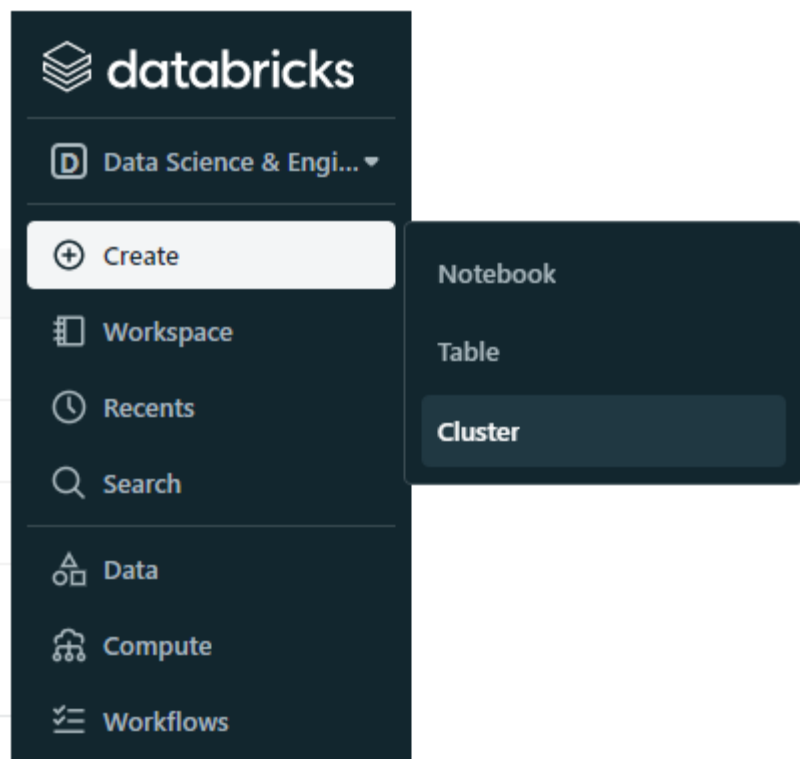
1. Databricks

**2. Cluster Spark**

- SQL
- R
- Python
- Scala

3. Streaming

# Cluster Spark



[Clusters](#) / [New Compute](#)

## New Cluster

Cancel

Create Cluster

0 Workers: 0 GB Memory, 0 Cores, 0 DBU

1 Driver: 15.3 GB Memory, 2 Cores, 1 DBU ⓘ

Cluster name

Please enter a cluster name

Databricks runtime version ⓘ

Runtime: 10.4 LTS (Scala 2.12, Spark 3.2.1) | v

Instance

Free 15 GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours. For [more configuration options](#), please [upgrade your Databricks subscription](#).

Instances

Spark

Availability zone ⓘ

auto | v

# Quickstart Notebook



Guide: Quickstart tutorial

Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.

[Start tutorial](#)

Quickstart Notebook

SQL ▾

File Edit View Run Help

[Last edit was 2 minutes ago](#)

[Give feedback](#)



▶ Run all

● QuickStart ▾

Publish

Cmd 1

## Databricks in 5 minutes

Markdown |

Cmd 2

### Create a quickstart cluster

1. In the sidebar, right-click the **Compute** button and open the link in a new window.
2. On the Clusters page, click **Create Cluster**.
3. Name the cluster **Quickstart**.
4. In the Databricks Runtime Version drop-down, select **7.3 LTS (Scala 2.12, Spark 3.0.1)**.
5. Click **Create Cluster**.

Cmd 3

### Attach the notebook to the cluster and run all commands in the notebook

1. Return to this notebook.
2. In the notebook menu bar, select Detached ▾ > **Quickstart**.
3. When the cluster changes from to , click **Run All**.

# Quickstart Notebook

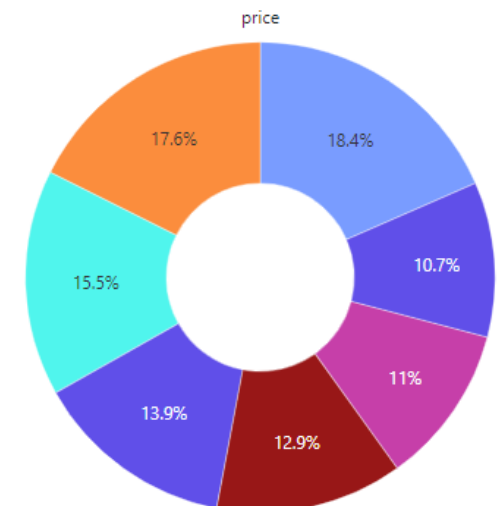
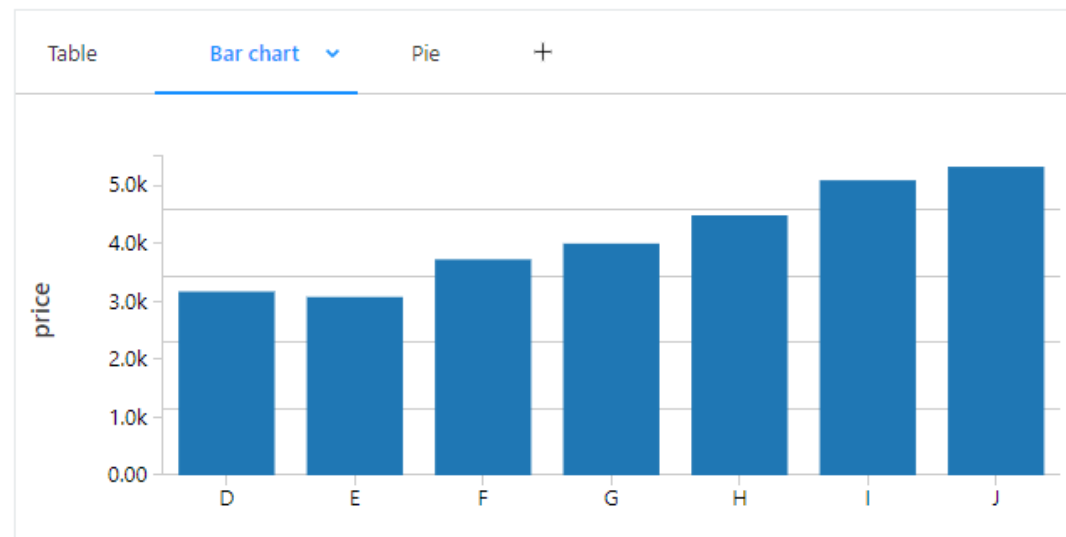
Table ▾ +

	_c0	carat	cut	color	clarity	depth	table	price	x	y	z
1	1	0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
2	2	0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
3	3	0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
4	4	0.29	Premium	I	VS2	62.4	58	334	4.2	4.23	2.63
5	5	0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
6	6	0.24	Very Good	I	VVS2	62.8	57	326	3.94	3.96	2.43

⬇ ▾ Truncated results, showing first 1,000 rows. ▾ | 1.22 seconds runtime Refreshed 20 minutes ago

Table ▾ Bar chart Pie +

	color	price
1	D	3169.9540959409596
2	E	3076.7524752475247
3	F	3724.886396981765
4	G	3999.135671271697
5	H	4486.669195568401
6	I	5091.874953891553
7	J	5323.81801994302



# Conjuntos de datos Databricks

Agregue la siguiente celda

```
%python  
display(dbutils.fs.ls('/databricks-datasets'))
```

► (3) Spark Jobs

Table ▾ +

	path	name	size	modificationTime
1	dbfs:/databricks-datasets/	databricks-datasets/	0	0
2	dbfs:/databricks-datasets/COVID/	COVID/	0	0
3	dbfs:/databricks-datasets/README.md	README.md	976	1532468253000
4	dbfs:/databricks-datasets/Rdatasets/	Rdatasets/	0	0
5	dbfs:/databricks-datasets/SPARK_README.md	SPARK_README.md	3359	1455043490000
6	dbfs:/databricks-datasets/adult/	adult/	0	0
7	dbfs:/databricks-datasets/airlines/	airlines/	0	0

⬇ Showing all 55 rows. | 1.44 seconds runtime

Utilidades DBFS: <https://docs.databricks.com/dev-tools/databricks-utils.html#file-system-utility-dbutilsfs>

# AGENDA

1. Databricks

**2. Cluster Spark**

- **SQL**

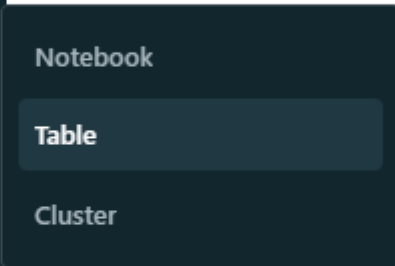
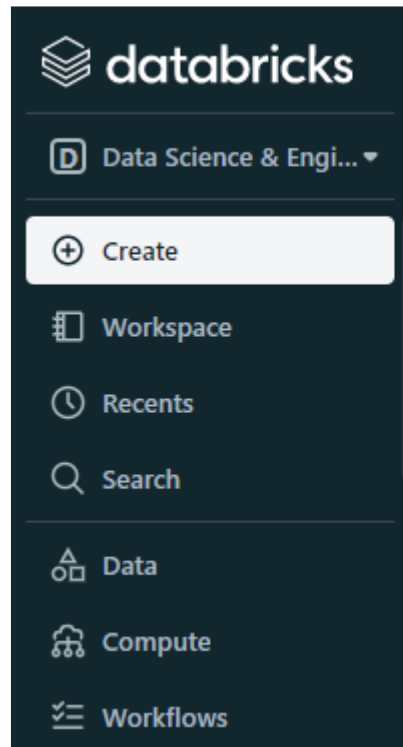
- R

- Python

- Scala

3. Streaming

# Crear Tabla



## Create New Table

Data source ?

Upload File

S3

DBFS

Other Data Sources

DBFS Target Directory ?

/FileStore/tables/ (optional)

Select

Files uploaded to DBFS are accessible by everyone who has access to this workspace. [Learn more](#)

Files ?

Drop files to upload, or click to browse

1

empleados.csv

2

Files ?

empleados.csv ✓

0.1 KB  
Remove file

✓ File uploaded to /FileStore/tables/empleados.csv

Create Table with UI

Create Table in Notebook

# Crear Tabla

## Select a Cluster to Preview the Table

Choose a cluster with which you will read and preview the data.

Cluster ?

Test1

3

Preview Table

4

## Specify Table Attributes

Specify the Table Name, Database and Schema to add this to the data UI for other users to access

Table Name ?

empleados\_csv

Create in Database ?

default

File Type ?

CSV

Column Delimiter ?

,

☒ First row is header ?

☒ Infer schema ?

☐ Multi-line ?

5

Create Table

Table Preview

Id	Nombre	Edad	Sexo
INT	STRING	INT	STRING
1	Camilo Sainz	24	Masculino
2	Javier Ballesteros	45	Masculino
3	Jacinta Pinto	42	Femenino
4	Martina Quevedo	35	Femenino
5	Braulio Blasco	41	Masculino
6	Claudia Palacio	22	Femenino



# Importar Notebook

The screenshot displays the Databricks Workspace interface. On the left is a dark sidebar with navigation options: 'Data Science & Engi...', 'Create', 'Workspace' (highlighted with a red circle and the number 1), 'Recents', 'Search', 'Data', 'Compute', and 'Workflows'. The main area is titled 'Workspace' and shows a 'Users' dropdown menu with the user 'efrain.oviedo@upb.edu.co' selected. A red circle with the number 2 highlights this user. A context menu is open for this user, with a red circle and the number 3 highlighting the 'Import' option. The menu also includes 'Create', 'Clone', 'Export', 'Permissions', and 'Copy Link Address'. At the bottom right, the text 'empleados\_SQL.dbc' is visible.

Workspace

Home

Users

efrain.oviedo@upb.edu.co

1

2

3

Create

Clone

Import

Export

Permissions

Copy Link Address

empleados\_SQL.dbc

# Empleados SQL

**empleados\_SQL** SQL ▾  
File Edit View Run Help [Last edit was 11 minutes ago](#) [Give feedback](#)  
📌 💬 ▶ Run all ● Test1 ▾

Cmd 1

Markdown 📊 ▾

## Ejemplo CRUD en Databricks SQL

**Read:** Vamos a leer la tabla de empleados

Cmd 2

```
1 select * from empleados_csv
```

Cmd 3

Y si no necesitamos todas las columnas?

Cmd 4

```
1 select Nombre, Edad from empleados_csv
```

Cmd 5

**Update:** Y si queremos actualizar algún campo?

Cmd 6

```
1 UPDATE empleados_csv set Nombre='Camilo Quintero' Where Id=1
```

# AGENDA

1. Databricks
- 2. Cluster Spark**
  - SQL
  - **R**
  - Python
  - Scala
3. Streaming

# Generalidades de R



- Lenguaje de programación interpretado enfocado en el análisis estadístico
- Entorno de Software libre bajo licencia GNU GPL
- Proyecto colaborativo y abierto
- Incluye utilidades gráficas para la visualización de datos
- Utilizado en Big data para manipulación, procesamiento y visualización de datos

# Importar Notebook desde URL

The screenshot illustrates the process of importing a notebook from a URL in the Databricks Workspace. The interface shows the sidebar with the 'Workspace' tab selected (1). A context menu is open over a notebook, with the 'Import' option highlighted (2). The 'Import Notebooks' dialog is displayed, showing the 'Import from:' section with 'URL' selected (3). The input field for the URL is highlighted (4), and a red arrow points to the URL provided in the box below the dialog.

<https://www.databricks.com/notebooks/gallery/DeltaLakePremierSparkR.html>

# Notebook R

Markdown



## Description

### Delta Lake Primer - SparkR



**DELTA LAKE**

This is a companion notebook to provide a Delta Lake example against the Lending Club data. It illustrates all functionality available in Delta Lake such as:

- Import data from Parquet to Delta Lake
- Batch and streaming updates
- Delete, update, and merge DML operations
- Schema evolution and enforcement.
- Time Travel

Run this cell by cell. Some cells will fail to illustrate lack of missing functionality in Parquet files but the subsequent operation on Delta Lake will

# AGENDA

1. Databricks
- 2. Cluster Spark**
  - SQL
  - R
  - **Python**
  - Scala
3. Streaming

# Llamadas a bomberos de San Francisco

## Description: San Francisco Fire Calls

Markdown    

This notebook is the end-to-end example from Chapter 3, from *Learning Spark 2nd Ed* showing how to use DataFrame and Spark SQL for common data analytics patterns and operations on a [San Francisco Fire Department Calls](#) dataset. It also demonstrates how to ETL, examine and query data for analysis. Additionally, it shows how to save in-memory Spark DataFrames as parquet files and read them back as a Spark supported Parquet data source.

## Setup

This notebook runs on DBR 8.1 and above.

Cmd 2

```
Inspect location where the SF Fire Department Fire calls data set is stored in the public dataset S3 bucket
```

<https://data.sfgov.org/Public-Safety/Fire-Department-Calls-for-Service/nuek-vuh3>



# AGENDA

1. Databricks

**2. Cluster Spark**

- SQL
- R
- Python
- **Scala**

3. Streaming

# Generalidades Scala



- Lenguaje de programación de propósito general desarrollado en 2001 que se ejecuta sobre la JVM
- Multiparadigma: Combina propiedades de lenguajes orientados a objetos y lenguajes funcionales
- Diseñado para expresar patrones comunes de programación en forma concisa, elegante y con tipos seguros

# Datasets Scala

## Importar Notebook: Datasets\_Scala.dbc

Cmd 1

Markdown 

### Spark Datasets with Scala

This notebook demonstrates a number of common Spark Dataset functions using Scala. It also demonstrates how structure enables developers to express high-level queries that are readable and composable. They look like SQL queries you would express, or domain specific language computation you would perform on your data set.

Keep this URL or open in the new tab to consult [Dataset API](#)

Cmd 2

### Setup: Create Sample Data to demonstrate Datasets.

Cmd 3

```
1 // Create the case classes for our domain
2 case class Department(id: String, name: String)
3 case class Employee(firstName: String, lastName: String, email: String, salary: Int)
4 case class DepartmentWithEmployees(department: Department, employees: Employee)
5
6 // Create the Departments
7 val department1 = new Department("123456", "Computer Science")
8 val department2 = new Department("345678", "Mechanical Engineering")
9 val department3 = new Department("123456", "Theater and Drama")
10 val department4 = new Department("901234", "Indoor Recreation")
11
```

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/8599738367597028/1499152197856461/3601578643761083/latest.html>

# AGENDA

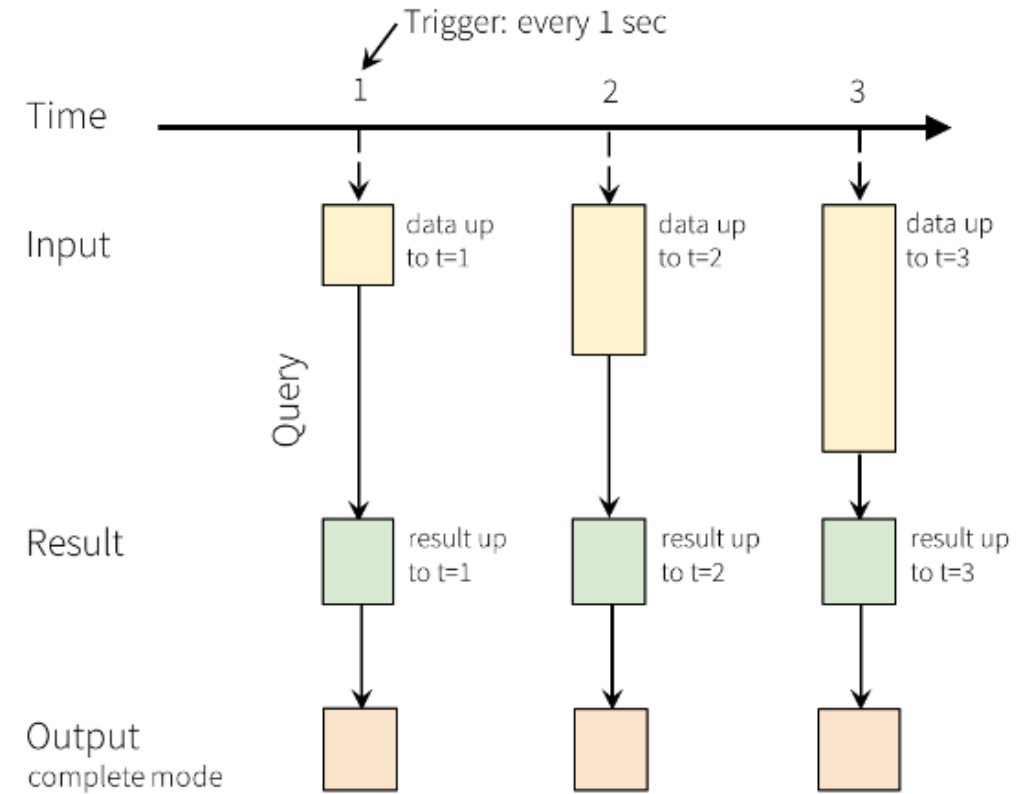
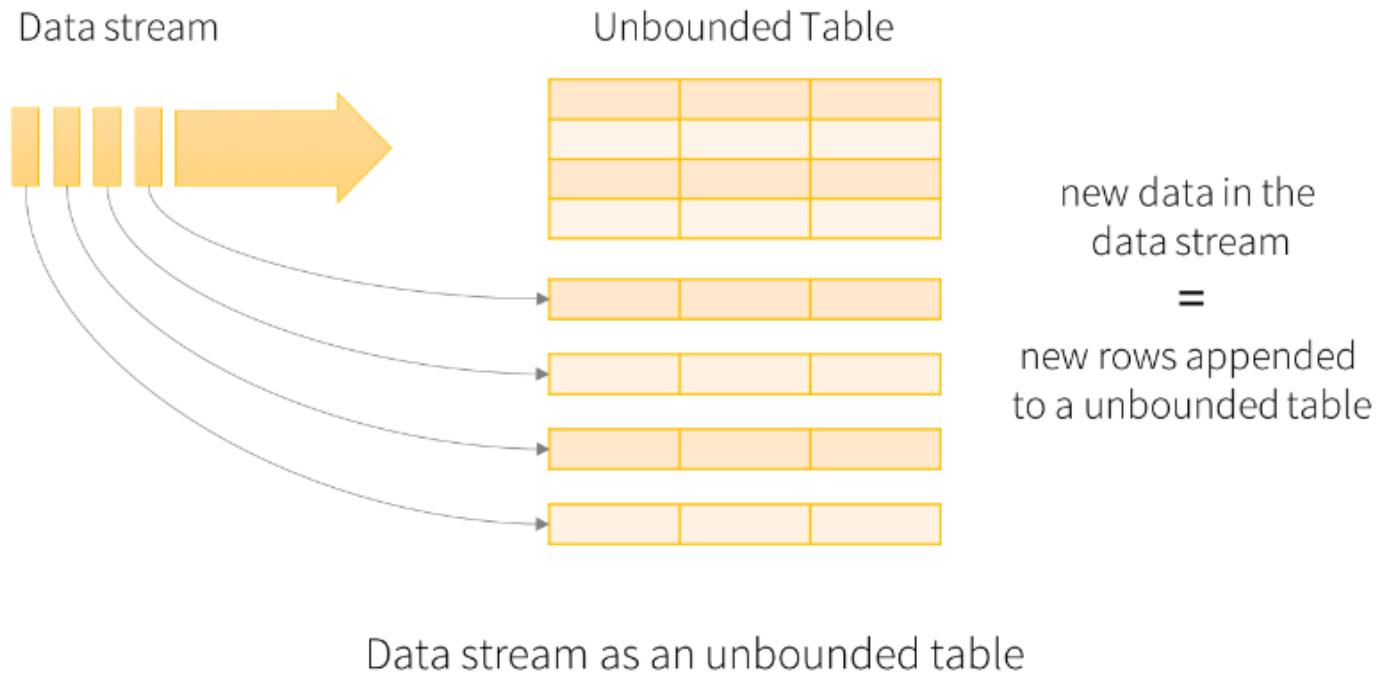
1. Databricks
2. Cluster Spark
  - SQL
  - R
  - Python
  - Scala

## **3. Streaming**

# Streaming en Spark

- La implementación de Streaming en Spark permite operar dataframes que reciben transmisiones de datos, de la misma forma como se operan los datos en batch. El procesamiento es rápido, escalable, tolerante a fallas
- Por defecto las consultas se ejecutan en microlotes con latencias del orden de 100ms
- También se incluye el procesamiento continuo donde se logran latencias de 1ms

# Streaming en Spark



Programming Model for Structured Streaming

# Contador de palabras

```
# Create DataFrame representing the stream of input lines from connection to localhost:9999
lines = spark \
    .readStream \
    .format("socket") \
    .option("host", "localhost") \
    .option("port", 9999) \
    .load()

# Split the lines into words
words = lines.select(
    explode(
        split(lines.value, " ")
    ).alias("word")
)

# Generate running word count
wordCounts = words.groupBy("word").count()
```

Crear el Streaming y definir el procesamiento

# Contador de palabras

```
query = wordCounts \  
    .writeStream \  
    .outputMode("complete") \  
    .format("console") \  
    .start()  
  
query.awaitTermination()
```

Iniciar el Streaming

```
# TERMINAL 1:  
# Running Netcat  
  
$ nc -lk 9999  
apache spark  
apache hadoop
```

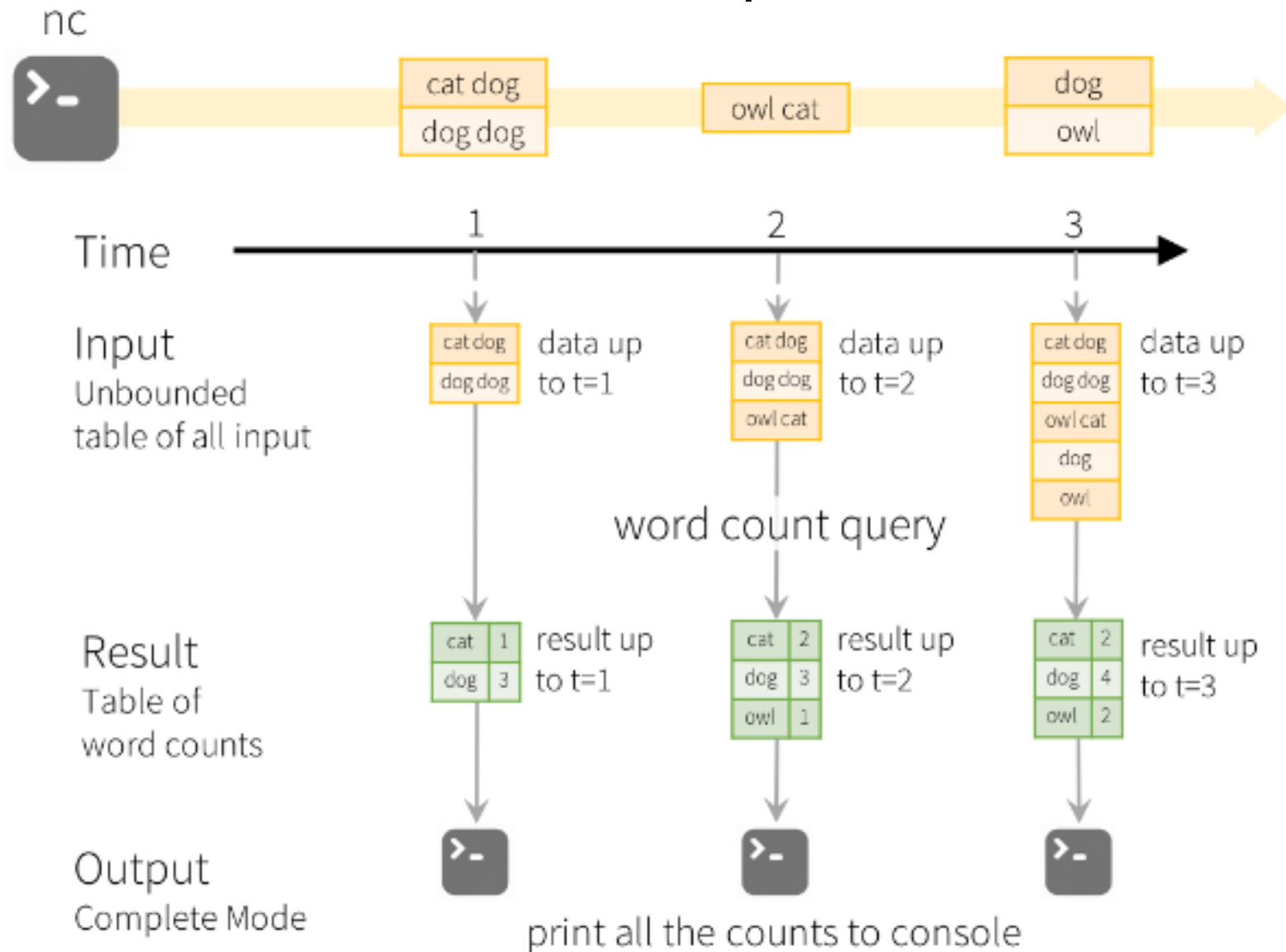
Transmisor

```
# TERMINAL 2: RUNNING structured_network_wordcount.py  
-----  
Batch: 0  
-----  
+-----+-----+  
| value|count|  
+-----+-----+  
| apache|    1|  
| spark|    1|  
+-----+-----+  
  
-----  
Batch: 1  
-----  
+-----+-----+  
| value|count|  
+-----+-----+  
| apache|    2|  
| spark|    1|  
| hadoop|    1|  
+-----+-----+
```

Receptor



# Contador de palabras



# Ejemplo Batch vs Streaming

## Batch

```
dataDeviceSchema = StructType([
    StructField("id",LongType(),False),
    StructField("user_id",LongType(),True),
    StructField("device_id",LongType(),True),
    StructField("num_steps",LongType(),True),
    StructField("miles_walked",FloatType(),True),
    StructField("calories_burnt",FloatType(),True),
    StructField("timestamp",StringType(),True),
    StructField("value",StringType(),True)
])
```

Cmd 6

```
dataDevice_df = spark.read.schema(dataDeviceSchema).json('dbfs://databricks-datasets/iot-stream/data-device/')
```

# Ejemplo Batch vs Streaming

## Streaming

Cmd 22

```
dataDeviceSchema = StructType([
    StructField("id",LongType(),False),
    StructField("user_id",LongType(),True),
    StructField("device_id",LongType(),True),
    StructField("num_steps",LongType(),True),
    StructField("miles_walked",FloatType(),True),
    StructField("calories_burnt",FloatType(),True),
    StructField("timestamp",StringType(),True),
    StructField("value",StringType(),True)
])
```

Python

Cmd 23

```
dataDevice_df = spark.readStream.schema(dataDeviceSchema).json('dbfs:/databricks-datasets/iot-stream/data-device/')
```

```
smokerAgg_df.writeStream.format("delta").outputMode("complete").option("checkpointLocation",
"/mnt/delta/eventsByCustomer/_checkpoints/streaming-agg").start("default.smokerAgg")
```

Python