

### 3.-Apache Spark

#### Laboratorio Desarrollo en Notebook con Apache Spark

The screenshot shows the Google Cloud Dataproc console interface. The browser address bar displays the URL: `console.cloud.google.com/dataproc/clusters/dmc-dev-bdp-15/interfaces?region=us-central1&hl=en&inv=1&inv=1&inv=1&project=polished-shore-450601-d0`. The page title is "Dataproc / Clusters / Cluster: dmc-dev-bdp-15 / Interfaces".

The left sidebar contains a navigation menu with the following sections:

- Overview
- Jobs on Clusters
  - Clusters**
  - Jobs
  - Workflows
  - Autoscaling policies
- Serverless
  - Batches
  - Interactive
  - Interactive Templates
- Metastore Services
  - Metastore
  - Federation
- Utilities
  - Component exchange
  - Workbench
- Dataproc on GDC
  - Service Instances

The main content area is titled "Cluster details" and includes buttons for "SUBMIT JOB", "REFRESH", "START", "STOP", "DELETE", and "VIEW LOGS". Below this is a table with the following information:

Name	dmc-dev-bdp-15
Cluster UUID	9f0fd831-71f6-41ec-aed7-a3b5d44e3720
Type	Dataproc Cluster
Status	Running

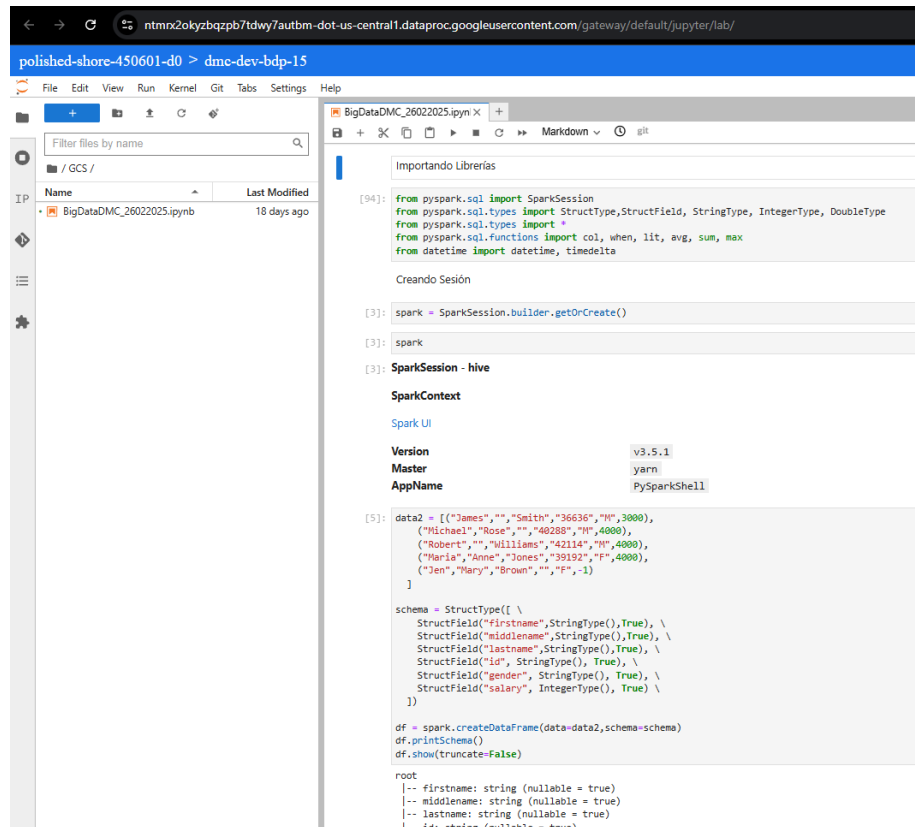
Below the table is a tabbed interface with the following tabs: "MONITORING", "JOBS", "VM INSTANCES", "CONFIGURATION", and "WEB INTERFACES" (which is selected and highlighted with a red box). The "WEB INTERFACES" tab contains the following sections:

- SSH tunnel**  
[Create an SSH tunnel to connect to a web interface](#)
- Component gateway**  
Provides access to the web interfaces of default and selected optional components on the cluster. [Learn more](#)
- [YARN ResourceManager](#)
- [MapReduce Job History](#)
- [Spark History Server](#)
- [HDFS NameNode](#)
- [YARN Application Timeline](#)
- [Tez](#)
- [Jupyter](#)
- [JupyterLab](#) (highlighted with a red box)

# Jupyter Notebooks sobre Apache Spark en Google Cloud Platform

## Importación de Módulos

## Crear Sesión de Spark



The screenshot shows a Jupyter Notebook titled 'BigDataDMC\_26022025.ipynb' in the 'dmc-dev-bdp-15' environment. The notebook is divided into two main sections: 'Importando Librerías' and 'Creando Sesión'.

**Importando Librerías:**

```
[94]: from pyspark.sql import SparkSession
      from pyspark.sql.types import StructType, StructField, StringType, IntegerType, DoubleType
      from pyspark.sql.functions import col, when, lit, avg, sum, max
      from datetime import datetime, timedelta
```

**Creando Sesión:**

```
[3]: spark = SparkSession.builder.getOrCreate()

[3]: spark

[3]: SparkSession - hive

SparkContext

Spark UI

Version          v3.5.1
Master           yarn
AppName          PySparkShell
```

**Data Loading:**

```
[5]: data2 = [{"James", "", "Smith", "36636", "M", 3000},
              {"Michael", "Rose", "", "40288", "M", 4000},
              {"Robert", "", "Williams", "42114", "M", 4000},
              {"Maria", "Anne", "Jones", "30192", "F", 4000},
              {"Jen", "Mary", "Brown", "", "F", -1}]

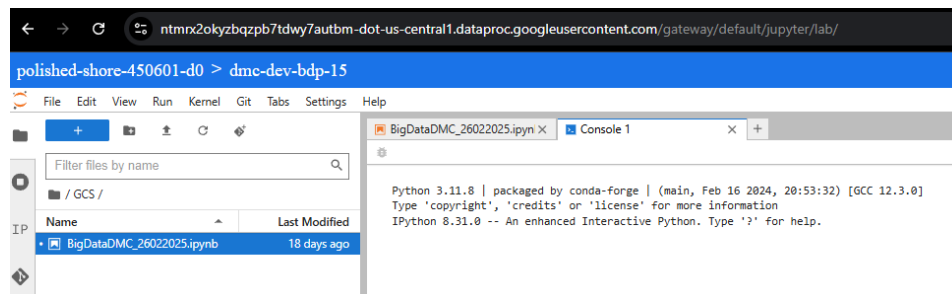
schema = StructType([ \
    StructField("firstname", StringType(), True), \
    StructField("middlename", StringType(), True), \
    StructField("lastname", StringType(), True), \
    StructField("id", StringType(), True), \
    StructField("gender", StringType(), True), \
    StructField("salary", IntegerType(), True) \
])

df = spark.createDataFrame(data=data2, schema=schema)
df.printSchema()
df.show(truncate=False)
```

The output of the last cell shows the schema of the DataFrame:

```
root
 |-- firstname: string (nullable = true)
 |-- middlename: string (nullable = true)
 |-- lastname: string (nullable = true)
 |-- id: string (nullable = true)
```

El Spark Console se crea la sesión por defecto.



The screenshot shows the Spark Console interface. The top bar indicates the environment is 'dmc-dev-bdp-15'. The main area displays the Spark version and configuration:

```
Python 3.11.8 | packaged by conda-forge | (main, Feb 16 2024, 20:53:32) [GCC 12.3.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.31.0 -- An enhanced Interactive Python. Type '?' for help.
```

## Dataframes con Schema

```
[5]: data2 = [("James", "", "Smith", "36636", "M", 3000),
             ("Michael", "Rose", "", "40288", "M", 4000),
             ("Robert", "", "Williams", "42114", "M", 4000),
             ("Maria", "Anne", "Jones", "39192", "F", 4000),
             ("Jen", "Mary", "Brown", "", "F", -1)
            ]

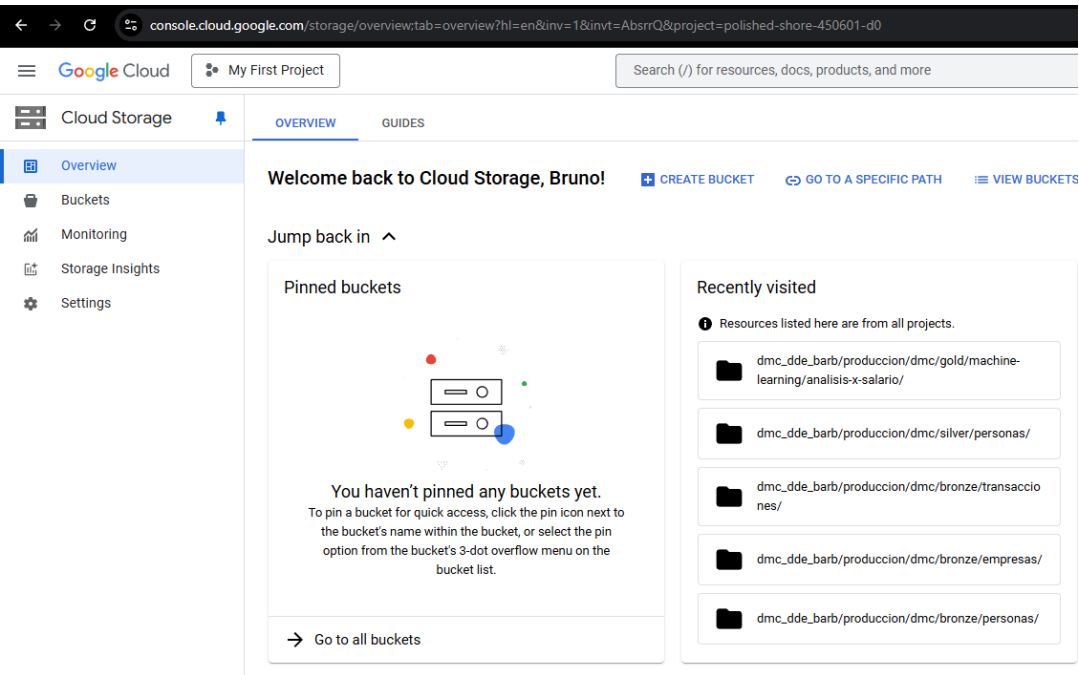
schema = StructType([ \
    StructField("firstname", StringType(), True), \
    StructField("middlename", StringType(), True), \
    StructField("lastname", StringType(), True), \
    StructField("id", StringType(), True), \
    StructField("gender", StringType(), True), \
    StructField("salary", IntegerType(), True) \
])

df = spark.createDataFrame(data=data2, schema=schema)
df.printSchema()
df.show(truncate=False)
```

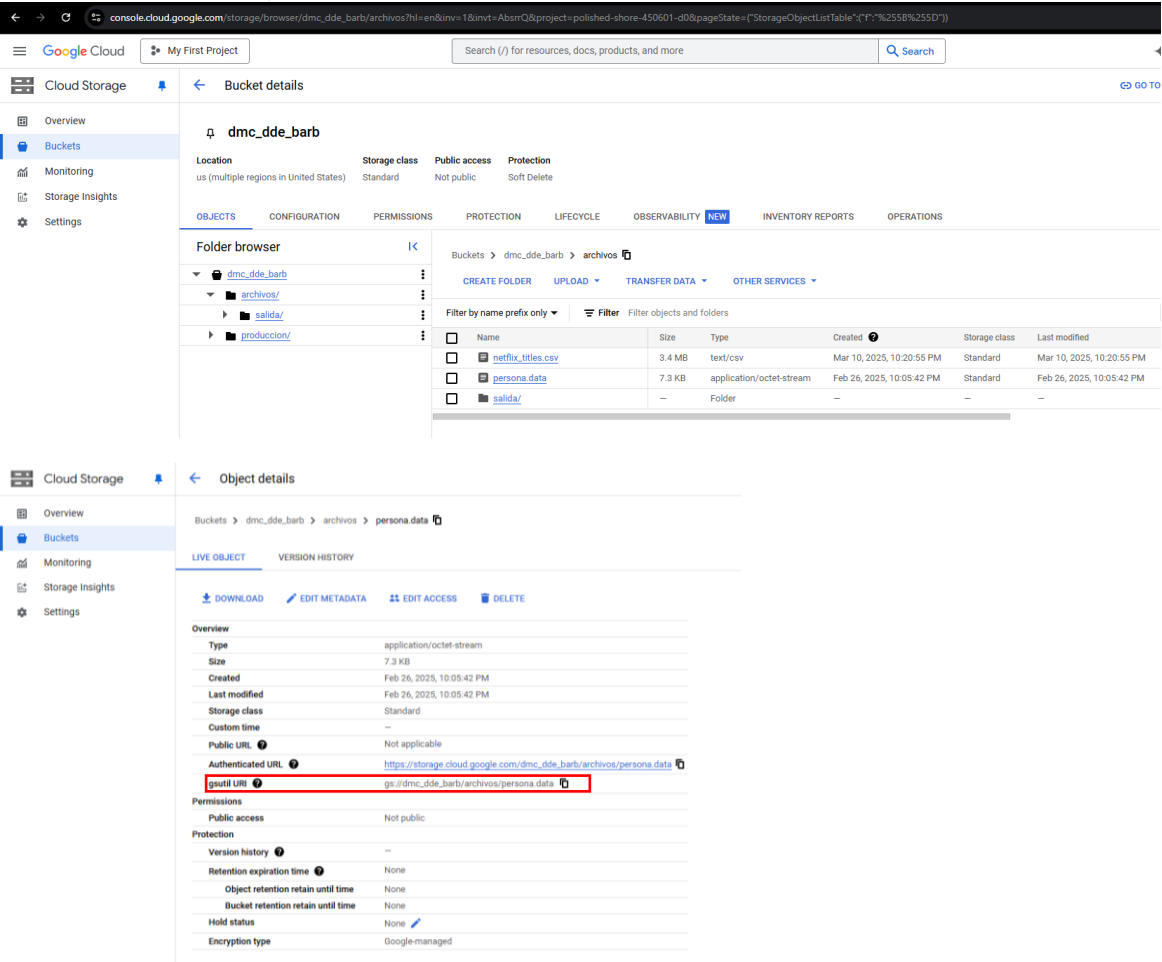
```
root
|-- firstname: string (nullable = true)
|-- middlename: string (nullable = true)
|-- lastname: string (nullable = true)
|-- id: string (nullable = true)
|-- gender: string (nullable = true)
|-- salary: integer (nullable = true)

+-----+-----+-----+-----+-----+-----+
|firstname|middlename|lastname|id   |gender|salary|
+-----+-----+-----+-----+-----+-----+
|James   |         |Smith  |36636|M    |3000  |
|Michael |Rose    |       |40288|M    |4000  |
|Robert  |        |Williams|42114|M    |4000  |
|Maria   |Anne    |Jones  |39192|F    |4000  |
|Jen     |Mary    |Brown  |     |F    |-1    |
+-----+-----+-----+-----+-----+-----+
```

# Creación de bucket para Cloud Storage en Google Cloud Plataform



Creamos un bucket dmc\_dde\_barb  
Una carpeta archivos dentro del bucket dmc\_dde\_barb/archivos  
Y subimos el archivo persona.data



Almacenar en un dataframe la lectura de archivos externos con `spark.read.format`

Función `show` para mostrar datos de un dataframe

Tipos `StructType` y `StructField` para definir esquemas.

`printSchema()` para ver el esquema del dataframe.

```
BigDataDMC_26022025.ipyn x Console 1
[6]: ruta = 'gs://dmc_dde_barb/archivos/persona.data'

df_schema = StructType([
    StructField("ID", StringType(), True),
    StructField("NOMBRE", StringType(), True),
    StructField("TELEFONO", StringType(), True),
    StructField("CORREO", StringType(), True),
    StructField("FECHA_INGRESO", StringType(), True),
    StructField("EDAD", IntegerType(), True),
    StructField("SALARIO", DoubleType(), True),
    StructField("ID_EMPRESA", StringType(), True),
])

df_with_schema = spark.read.format("CSV").option("header", "true").option("delimiter", "|").schema(df_schema).load(ruta)

[7]: df_with_schema.show(10)

[Stage 4:>] (0 + 1) / 1
+----+-----+-----+-----+-----+-----+-----+-----+
| ID | NOMBRE | TELEFONO | CORREO | FECHA_INGRESO | EDAD | SALARIO | ID_EMPRESA |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Carl | 1-745-633-9145 | arcu.Sed.et@ante... | 2004-04-23 | 32 | 20095.0 | 5 |
| 2 | Priscilla | 155-2498 | Donec.egestas.Ali... | 2019-02-17 | 34 | 9298.0 | 2 |
| 3 | Jocelyn | 1-204-956-8594 | amet.diam@loborti... | 2002-08-01 | 27 | 10853.0 | 3 |
| 4 | Aidan | 1-719-862-9385 | euismod.et.commod... | 2018-11-06 | 29 | 3387.0 | 10 |
| 5 | Leandra | 839-8044 | at@pretiumetrutru... | 2002-10-10 | 41 | 22102.0 | 1 |
| 6 | Bert | 797-4453 | a.felis.ullamcorp... | 2017-04-25 | 70 | 7800.0 | 7 |
| 7 | Mark | 1-600-102-6792 | Quisque.ac@placer... | 2006-04-21 | 52 | 8112.0 | 5 |
| 8 | Jonah | 214-2975 | eu.ultrices.sit@v... | 2017-10-07 | 23 | 17040.0 | 5 |
| 9 | Hanae | 935-2277 | eu@Nunc.ca | 2003-05-25 | 69 | 6834.0 | 3 |
| 10 | Cadman | 1-866-561-2701 | orci.adipiscing.n... | 2001-05-19 | 19 | 7996.0 | 7 |
+----+-----+-----+-----+-----+-----+-----+-----+
only showing top 10 rows
```

## Función Select

```
dot-us-central1.dataproc.googleusercontent.com/gateway/default/jupyter/lab/

Help
BigDataDMC_26022025.ipyn x Console 1
[14]: df_with_schema.select("nombre", "fecha_ingreso").show(4)

+-----+-----+
| nombre | fecha_ingreso |
+-----+-----+
| Carl | 2004-04-23 |
| Priscilla | 2019-02-17 |
| Jocelyn | 2002-08-01 |
| Aidan | 2018-11-06 |
+-----+-----+
only showing top 4 rows

[13]: df_with_schema.select(df_with_schema.NOMBRE, df_with_schema.FECHA_INGRESO).show(4)

+-----+-----+
| NOMBRE | FECHA_INGRESO |
+-----+-----+
| Carl | 2004-04-23 |
| Priscilla | 2019-02-17 |
| Jocelyn | 2002-08-01 |
| Aidan | 2018-11-06 |
+-----+-----+
only showing top 4 rows

[16]: df_with_schema.select(df_with_schema["nombre"], df_with_schema["fecha_ingreso"]).show(5)

+-----+-----+
| nombre | fecha_ingreso |
+-----+-----+
| Carl | 2004-04-23 |
| Priscilla | 2019-02-17 |
| Jocelyn | 2002-08-01 |
| Aidan | 2018-11-06 |
| Leandra | 2002-10-10 |
+-----+-----+
only showing top 5 rows

[19]: df_with_schema.select(col("nombre"), col("fecha_ingreso")).show(4)

+-----+-----+
| nombre | fecha_ingreso |
+-----+-----+
| Carl | 2004-04-23 |
| Priscilla | 2019-02-17 |
| Jocelyn | 2002-08-01 |
| Aidan | 2018-11-06 |
+-----+-----+
only showing top 4 rows
```

### Función withColumn, col, cast, when, otherwise

```

[24]: df_with_schema.withColumn("salario", col("salario").cast("integer")).show(5)

+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | NOMBRE | TELEFONO | CORREO | FECHA_INGRESO | EDAD | salario | ID_EMPRESA |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Carl | 1-745-633-9145 | arcu.Sed.et@ante... | 2004-04-23 | 32 | 20095 | 5 |
| 2 | Priscilla | 155-2498 | Donec.egestas.Ali... | 2019-02-17 | 34 | 9298 | 2 |
| 3 | Jocelyn | 1-204-956-8594 | amet.diam@loborti... | 2002-08-01 | 27 | 10853 | 3 |
| 4 | Aidan | 1-719-862-9385 | euismod.et.commod... | 2018-11-06 | 29 | 3387 | 10 |
| 5 | Leandra | 839-8044 | at@pretiumetrutru... | 2002-10-10 | 41 | 22102 | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+

only showing top 5 rows

[30]: df_with_schema.withColumn("incremento", col("salario")*0.10).select(col("nombre"), col("salario"), col("incremento")).show(5)

+-----+-----+-----+
| nombre | salario | incremento |
+-----+-----+-----+
| Carl | 20095.0 | 2009.5 |
| Priscilla | 9298.0 | 929.8000000000001 |
| Jocelyn | 10853.0 | 1085.3 |
| Aidan | 3387.0 | 338.70000000000005 |
| Leandra | 22102.0 | 2210.2000000000003 |
+-----+-----+-----+

only showing top 5 rows

[42]: limite_salario_bajo = 5000
      inicio_salario_alto = 30000

df_with_schema.withColumn("tipo_salario", when(col("salario") < limite_salario_bajo, "salario bajo") \
      .when(((col("salario") >= limite_salario_bajo) & (col("salario") < inicio_salario_alto)), "salario medio") \
      .otherwise("salario alto")).show(10)

+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | NOMBRE | TELEFONO | CORREO | FECHA_INGRESO | EDAD | SALARIO | ID_EMPRESA | tipo_salario |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Carl | 1-745-633-9145 | arcu.Sed.et@ante... | 2004-04-23 | 32 | 20095.0 | 5 | salario medio |
| 2 | Priscilla | 155-2498 | Donec.egestas.Ali... | 2019-02-17 | 34 | 9298.0 | 2 | salario medio |
| 3 | Jocelyn | 1-204-956-8594 | amet.diam@loborti... | 2002-08-01 | 27 | 10853.0 | 3 | salario medio |
| 4 | Aidan | 1-719-862-9385 | euismod.et.commod... | 2018-11-06 | 29 | 3387.0 | 10 | salario bajo |
| 5 | Leandra | 839-8044 | at@pretiumetrutru... | 2002-10-10 | 41 | 22102.0 | 1 | salario medio |
| 6 | Bert | 797-4453 | a.felis.ullamcorp... | 2017-04-25 | 70 | 7800.0 | 7 | salario medio |
| 7 | Mark | 1-680-102-6792 | Quisque.ac@placer... | 2006-04-21 | 52 | 8112.0 | 5 | salario medio |
| 8 | Jonah | 214-2975 | eu.ultrices.sit@v... | 2017-10-07 | 23 | 17040.0 | 5 | salario medio |
| 9 | Hanae | 935-2277 | eu@Nunc.ca | 2003-05-25 | 69 | 6834.0 | 3 | salario medio |
| 10 | Cadman | 1-866-561-2701 | orci.adipiscing.n... | 2001-05-19 | 19 | 7996.0 | 7 | salario medio |
+-----+-----+-----+-----+-----+-----+-----+-----+

only showing top 10 rows

```

Función withColumnRenamed, drop, lit

Help

BigDataDMC\_26022025.ipyn x Console 1

only showing top 10 rows

[43]: df\_with\_schema.withColumnRenamed("salario", "salary").show(10)

ID	NOMBRE	TELEFONO	CORREO	FECHA_INGRESO	EDAD	salary	ID_EMPRESA
1	Carl	1-745-633-9145	arcu.Sed.et@ante...	2004-04-23	32	20095.0	5
2	Priscilla	155-2498	Donec.egestas.Ali...	2019-02-17	34	9298.0	2
3	Jocelyn	1-204-956-8594	amet.diam@loborti...	2002-08-01	27	10853.0	3
4	Aidan	1-719-862-9385	euismod.et.commod...	2018-11-06	29	3387.0	10
5	Leandra	839-8044	at@pretiumetrutru...	2002-10-10	41	22102.0	1
6	Bert	797-4453	a.felis.ullamcorp...	2017-04-25	70	7800.0	7
7	Mark	1-680-102-6792	Quisque.ac@placer...	2006-04-21	52	8112.0	5
8	Jonah	214-2975	eu.ultrices.sit@v...	2017-10-07	23	17040.0	5
9	Hanae	935-2277	eu@Nunc.ca	2003-05-25	69	6834.0	3
10	Cadman	1-866-561-2701	orci.adipiscing.n...	2001-05-19	19	7996.0	7

only showing top 10 rows

[44]: df\_with\_schema.drop("salario").show(5)

ID	NOMBRE	TELEFONO	CORREO	FECHA_INGRESO	EDAD	ID_EMPRESA
1	Carl	1-745-633-9145	arcu.Sed.et@ante...	2004-04-23	32	5
2	Priscilla	155-2498	Donec.egestas.Ali...	2019-02-17	34	2
3	Jocelyn	1-204-956-8594	amet.diam@loborti...	2002-08-01	27	3
4	Aidan	1-719-862-9385	euismod.et.commod...	2018-11-06	29	10
5	Leandra	839-8044	at@pretiumetrutru...	2002-10-10	41	1

only showing top 5 rows

[46]: df\_with\_schema.withColumn("periodo", lit("202503")).show(10)

ID	NOMBRE	TELEFONO	CORREO	FECHA_INGRESO	EDAD	SALARIO	ID_EMPRESA	periodo
1	Carl	1-745-633-9145	arcu.Sed.et@ante...	2004-04-23	32	20095.0	5	202503
2	Priscilla	155-2498	Donec.egestas.Ali...	2019-02-17	34	9298.0	2	202503
3	Jocelyn	1-204-956-8594	amet.diam@loborti...	2002-08-01	27	10853.0	3	202503
4	Aidan	1-719-862-9385	euismod.et.commod...	2018-11-06	29	3387.0	10	202503
5	Leandra	839-8044	at@pretiumetrutru...	2002-10-10	41	22102.0	1	202503
6	Bert	797-4453	a.felis.ullamcorp...	2017-04-25	70	7800.0	7	202503
7	Mark	1-680-102-6792	Quisque.ac@placer...	2006-04-21	52	8112.0	5	202503
8	Jonah	214-2975	eu.ultrices.sit@v...	2017-10-07	23	17040.0	5	202503
9	Hanae	935-2277	eu@Nunc.ca	2003-05-25	69	6834.0	3	202503
10	Cadman	1-866-561-2701	orci.adipiscing.n...	2001-05-19	19	7996.0	7	202503

only showing top 10 rows

## Función Filter e isin

Help

```
BigDataDMC_26022025.ipyn x Console 1 x +
+ + + + +
[52]: fecha_actual = datetime.now()
      print("hora del servidor: ", fecha_actual)
      fecha_peru = fecha_actual - timedelta(hours=5)
      periodo = fecha_peru.strftime("%Y%m")
      print(periodo)

hora del servidor:  2025-03-04 02:32:30.982753
202503

[54]: df_with_schema.filter(col("id_empresa")==5).show(10)

+-----+-----+-----+-----+-----+-----+-----+
| ID|NOMBRE|   TELEFONO|   CORREO|FECHA_INGRESO|EDAD|SALARIO|ID_EMPRESA|
+-----+-----+-----+-----+-----+-----+-----+
| 1|  Carl|1-745-633-9145|arcu.Sed.et@ante....|  2004-04-23| 32|20095.0|      5|
| 7|  Mark|1-680-102-6792|Quisque.ac@placer...|  2006-04-21| 52| 8112.0|      5|
| 8|  Jonah|  214-2975|eu.ultrices.sit@v...|  2017-10-07| 23|17040.0|      5|
|13| Trevor|  512-1955|Nunc.quis.arcu@eg...|  2010-08-06| 34| 9501.0|      5|
|15| Wanda|  359-6973|Nam.nulla.magna@I...|  2005-08-21| 27| 1539.0|      5|
|35| Aurora|1-865-751-3479|  magna@Cras.net|  2017-10-21| 54| 4588.0|      5|
|50|  Ross|1-587-285-1837|at.risus@milacini...|  2009-11-03| 31|19092.0|      5|
|51| Damon|  368-7630|nunc@dapibusquamq...|  2016-08-11| 49| 2669.0|      5|
|59| Quemby| 930-5882|lorem.ut.aliquam@...|  2017-10-04| 26|12092.0|      5|
|68| Hayes|  712-8783|  at@ametdiam.net|  2011-12-31| 31| 7523.0|      5|
+-----+-----+-----+-----+-----+-----+-----+
only showing top 10 rows

[56]: df_with_schema.filter((col("id_empresa")==5) & (col("salario")>=10000)).show()

+-----+-----+-----+-----+-----+-----+-----+
| ID|NOMBRE|   TELEFONO|   CORREO|FECHA_INGRESO|EDAD|SALARIO|ID_EMPRESA|
+-----+-----+-----+-----+-----+-----+-----+
| 1|  Carl|1-745-633-9145|arcu.Sed.et@ante....|  2004-04-23| 32|20095.0|      5|
| 8|  Jonah|  214-2975|eu.ultrices.sit@v...|  2017-10-07| 23|17040.0|      5|
|50|  Ross|1-587-285-1837|at.risus@milacini...|  2009-11-03| 31|19092.0|      5|
|59| Quemby| 930-5882|lorem.ut.aliquam@...|  2017-10-04| 26|12092.0|      5|
|86|  Jack|  860-9554|parturient.montes...|  2017-03-10| 58|14473.0|      5|
+-----+-----+-----+-----+-----+-----+-----+

[63]: lista_empresas = [2,3,5]
      df_with_schema.filter(col("id_empresa").isin([2,3,5])).show(3)
      df_with_schema.filter(col("id_empresa").isin(2,3,5)).show(3)
      df_with_schema.filter(col("id_empresa").isin(lista_empresas)).show(3)

+-----+-----+-----+-----+-----+-----+-----+
| ID|  NOMBRE|   TELEFONO|   CORREO|FECHA_INGRESO|EDAD|SALARIO|ID_EMPRESA|
+-----+-----+-----+-----+-----+-----+-----+
| 1|  Carl|1-745-633-9145|arcu.Sed.et@ante....|  2004-04-23| 32|20095.0|      5|
| 2|Priscilla|  155-2498|Donec.egestas.Ali...|  2019-02-17| 34| 9298.0|      2|
| 3| Jocelyn|1-204-956-8594|amet.diam@loborti...|  2002-08-01| 27|10853.0|      3|
```



## Función like

```
Help
BigDataDMC_26022025.ipyn x Console 1 x +
[64]: df_with_schema.filter(col("nombre").like("%ar%")).show()

+-----+-----+-----+-----+-----+-----+-----+
| ID | NOMBRE | TELEFONO | CORREO | FECHA_INGRESO | EDAD | SALARIO | ID_EMPRESA |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Carl | 1-745-633-9145 | arcu.Sed.et@ante... | 2004-04-23 | 32 | 20095.0 | 5 |
| 7 | Mark | 1-680-102-6792 | Quisque.ac@placer... | 2006-04-21 | 52 | 8112.0 | 5 |
| 17 | Omar | 720-1543 | Phasellus.vitae.m... | 2014-06-24 | 60 | 6851.0 | 6 |
| 21 | Carissa | 1-300-877-0859 | dignissim.pharetr... | 2011-10-16 | 31 | 1952.0 | 10 |
| 25 | Pearl | 1-850-202-3373 | vel.convallis@rho... | 2018-12-21 | 52 | 14756.0 | 6 |
| 39 | Carolyn | 846-7060 | metus.Aenean.sed@... | 2013-05-29 | 64 | 22838.0 | 6 |
| 54 | Lars | 1-554-600-0855 | commodo@Nam.edu | 2005-06-22 | 25 | 20573.0 | 1 |
| 60 | Bernard | 492-8823 | vel.faucibus@Done... | 2005-04-15 | 27 | 10825.0 | 2 |
| 76 | Omar | 1-325-245-9578 | elit.erat@utodiov... | 2012-11-19 | 34 | 12163.0 | 6 |
| 87 | Karly | 1-644-725-7241 | tempor.erat@feugi... | 2011-06-12 | 25 | 3715.0 | 1 |
+-----+-----+-----+-----+-----+-----+-----+

[65]: data = [("James", "Sales", 3000), \
              ("Michael", "Sales", 4600), \
              ("Robert", "Sales", 4100), \
              ("Maria", "Finance", 3000), \
              ("James", "Sales", 3000), \
              ("Scott", "Finance", 3300), \
              ("Jen", "Finance", 3900), \
              ("Jeff", "Marketing", 3000), \
              ("Kumar", "Marketing", 2000), \
              ("Saif", "Sales", 4100) \
            ]
columns = ["employee_name", "department", "salary"]
df_a = spark.createDataFrame(data = data, schema = columns)
df_a.printSchema()
df_a.show(truncate=False)

root
|-- employee_name: string (nullable = true)
|-- department: string (nullable = true)
|-- salary: long (nullable = true)

+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|James        |Sales     |3000  |
|Michael      |Sales     |4600  |
|Robert       |Sales     |4100  |
|Maria        |Finance   |3000  |
|James        |Sales     |3000  |
|Scott        |Finance   |3300  |
|Jen          |Finance   |3900  |
|Jeff         |Marketing |3000  |
|Kumar        |Marketing |2000  |
+-----+-----+-----+
```

## Función count, distinct, dropDuplicates

```
Help
BigDataDMC_26022025.ipyn x Console 1 x +
[70]: df_a.count()
[70]: 10
[72]: df_a.distinct().show()
df_a.distinct().count()
+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|James|Sales|3000|
|Robert|Sales|4100|
|Maria|Finance|3000|
|Michael|Sales|4600|
|Saif|Sales|4100|
|Scott|Finance|3300|
|Jeff|Marketing|3000|
|Jen|Finance|3900|
|Kumar|Marketing|2000|
+-----+-----+-----+
[72]: 9
[74]: df_a.dropDuplicates().show()
+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|James|Sales|3000|
|Robert|Sales|4100|
|Maria|Finance|3000|
|Michael|Sales|4600|
|Saif|Sales|4100|
|Scott|Finance|3300|
|Jeff|Marketing|3000|
|Jen|Finance|3900|
|Kumar|Marketing|2000|
+-----+-----+-----+
[75]: df_a.dropDuplicates(["department", "salary"]).show()
[Stage 78:=====> (1 + 1) / 2]
+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|Maria|Finance|3000|
|Scott|Finance|3300|
|Jen|Finance|3900|
|Kumar|Marketing|2000|
|Jeff|Marketing|3000|
```

## Función orderBy, asc, desc, groupBy, agg, sum, avg, max

The screenshot shows a Databricks workspace with a notebook titled 'BigDataDMC\_26022025.ipynb'. The code in the notebook is as follows:

```
[97]: df_with_schema.groupBy("id_empresa")\
      .agg(sum("salario").alias("planilla"),\
           avg("edad").alias("promedio_edad"),\
           max("salario").alias("salario_maximo"))\
      .where(col("planilla") >= 100000).show(10)
```

The output of the query is displayed below the code:

id_empresa	planilla	promedio_edad	salario_maximo
7	106710.0	34.55555555555556	21556.0
3	151700.0	39.63636363636363	23820.0
6	135243.0	50.0	22838.0
5	136609.0	41.214285714285715	20095.0
4	155503.0	38.875	24305.0
2	156377.0	39.785714285714285	22953.0

## Función where, partitionby, write.mode, format, save

The screenshot shows a Databricks workspace with a notebook titled 'BigDataDMC\_26022025.ipynb'. The code in the notebook is as follows:

```
[97]: df_with_schema.groupBy("id_empresa")\
      .agg(sum("salario").alias("planilla"),\
           avg("edad").alias("promedio_edad"),\
           max("salario").alias("salario_maximo"))\
      .where(col("planilla") >= 100000).show(10)
```

The output of the query is displayed below the code:

id_empresa	planilla	promedio_edad	salario_maximo
7	106710.0	34.55555555555556	21556.0
3	151700.0	39.63636363636363	23820.0
6	135243.0	50.0	22838.0
5	136609.0	41.214285714285715	20095.0
4	155503.0	38.875	24305.0
2	156377.0	39.785714285714285	22953.0

The next code block shows the partitioning and saving of the data:

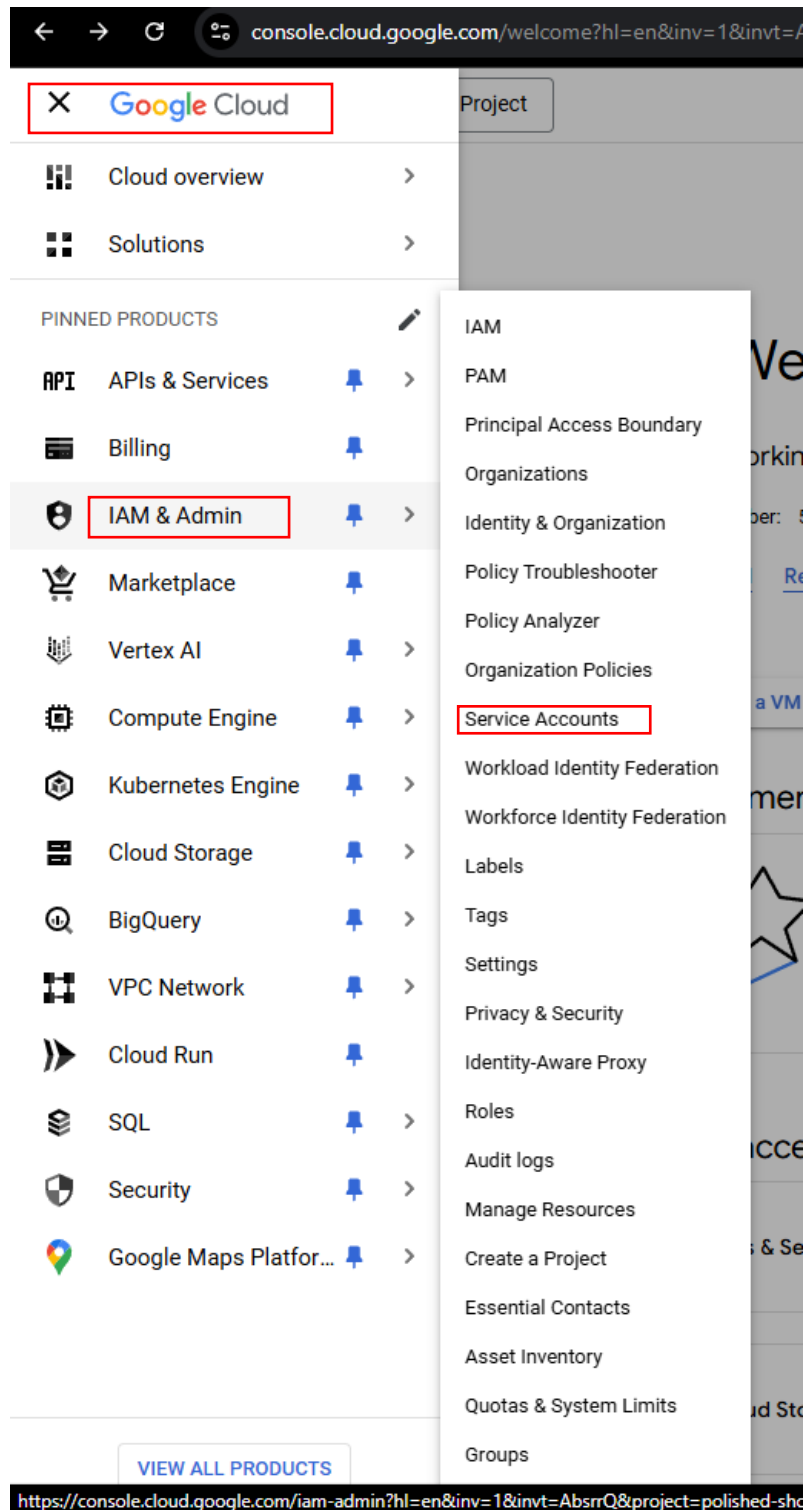
```
[99]: #PARTITIONBY | REPARTITION | COALESCE
#PARTITIONBY-> FUNCIONAL MOMENTO DE ESCRIBIR EN DISCO - PARTICIONANDO DE DATOS EN DISCO df.write.partitionby("campo_partition")
#REPARTITION -> aumentar o disminuir particiones en memoria df.repartition(4)
#COALESCE -> disminuir u optimizar particiones en memoria. df.coalesce()

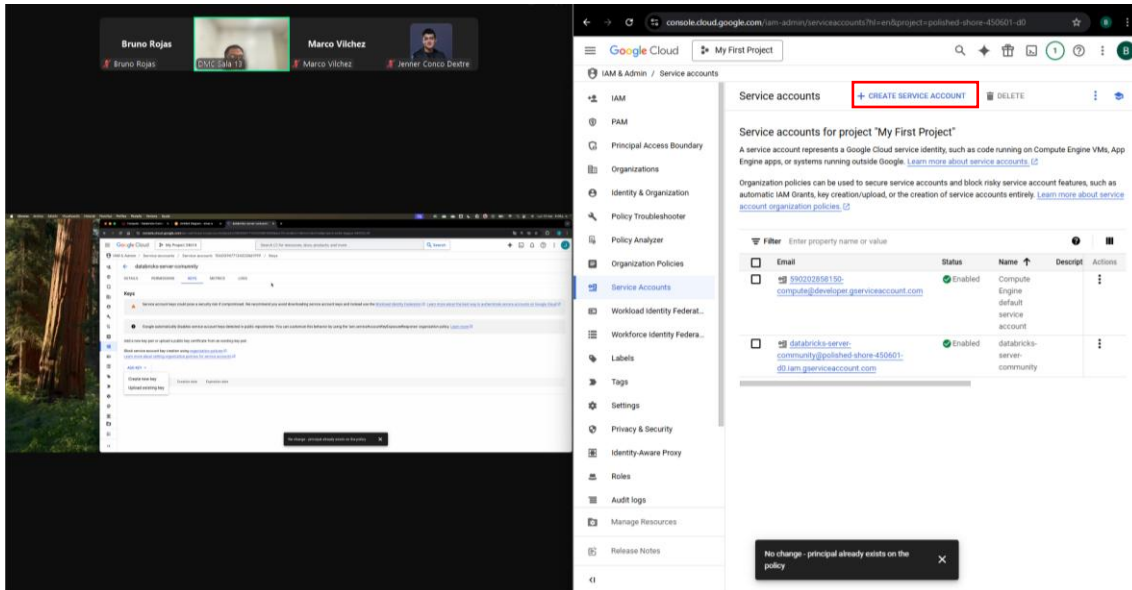
ruta_destino = 'gs://dmc_dde_barb/archivos/salida/'
df_with_schema.write.mode("overwrite").partitionBy("id_empresa").format("parquet").save(ruta_destino)

#ruta_guardado = 'gs://dmc_dataLake_dde_11_jmsp/archivos/persona_output/'
#df.write.mode("overwrite").partitionBy("ID_EMPRESA").format("parquet").save(ruta_guardado)
```

## Apache Spark en Databrick

Creación de Credenciales a través de GCP→IAM & Admin→Service Accounts





IAM & Admin / Service accounts / Create service account

## Create service account

- Service account details**
  - Service account name**  
databricks-server-community  
Display name for this service account
  - Service account ID \***  
databricks-server-community  
Email address: databricks-server-community@solid-league-440122-16.iam.gserviceaccount.com
  - Service account description**  
Describe what this service account will do
- Grant this service account access to project (optional)**
- Grant users access to this service account (optional)**

**CREATE AND CONTINUE**

**DONE** **CANCEL**

IAM

PAM

Principal Access Boundary

Organizations

Identity & Organization

Policy Troubleshooter

Policy Analyzer

Organization Policies

Service Accounts

Workload Identity Federat...

Workforce Identity Federa...

Labels

Tags

Settings

Privacy & Security

Identity-Aware Proxy

Roles

Audit logs

Manage Resources

Release Notes

Create service account

✓ Service account details

2 Grant this service account access to project (optional)

Grant this service account access to My First Project so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role

Filter Filter by role or permission

Quick access

Currently used

Basic

By product or service

Access Approval

Access Context Manager

Actions

Roles

Browser

Editor

Owner

Viewer

MANAGE ROLES

IAM condition (optional)

Bruno Rojas

Marco Vilchez

BRUNO ROJAS

MARCO VILCHEZ

Google Cloud

My First Project

IAM

PAM

Principal Access Boundary

Organizations

Identity & Organization

Policy Troubleshooter

Policy Analyzer

Organization Policies

Service Accounts

Workload Identity Federat...

Workforce Identity Federa...

Labels

Tags

Settings

Privacy & Security

Identity-Aware Proxy

Roles

Audit logs

Manage Resources

Release Notes

Service accounts

+ CREATE SERVICE ACCOUNT

DELETE

Service accounts for project "My First Project"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts](#)

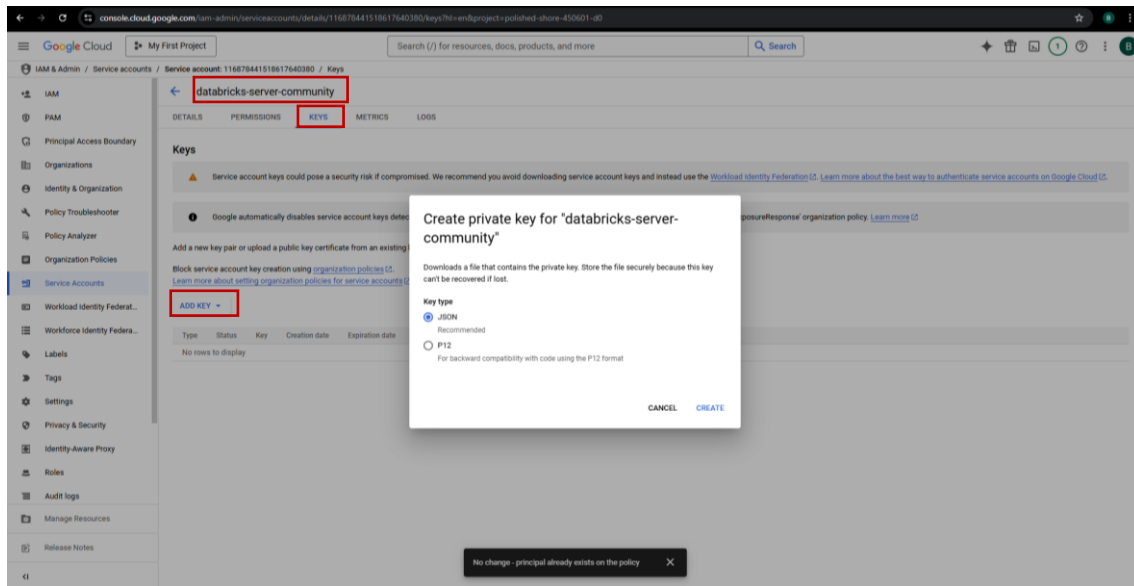
Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies](#)

Filter

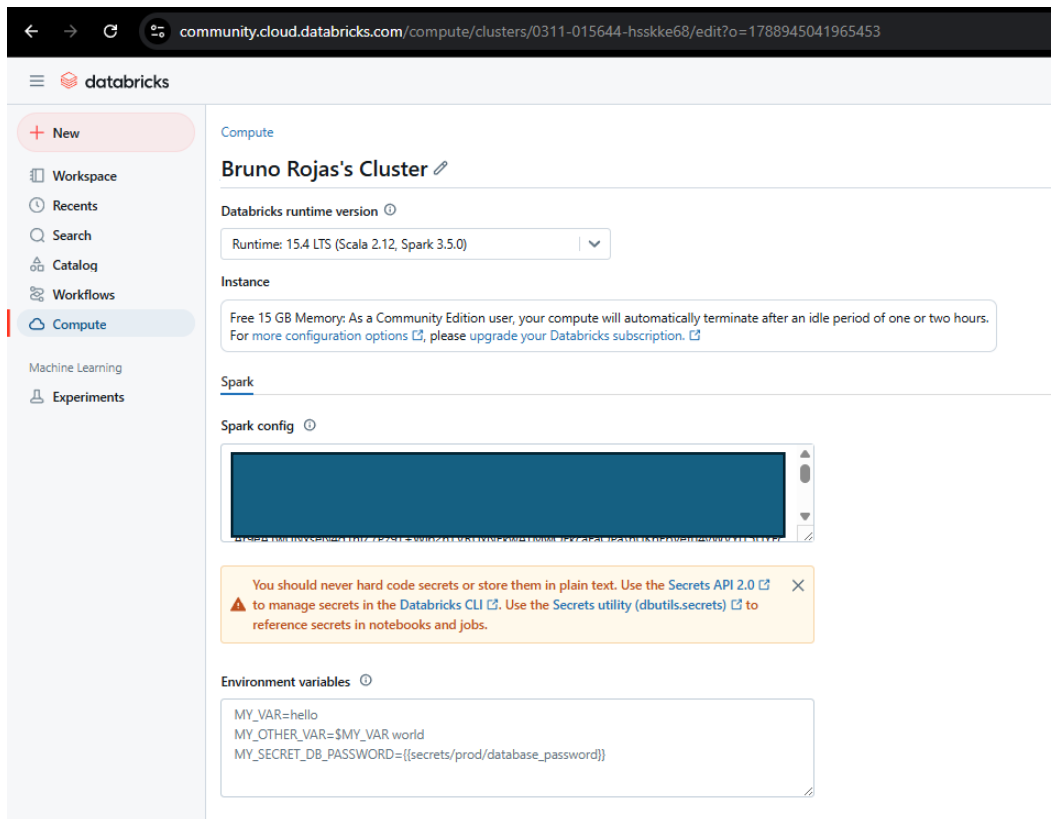
Enter property name or value

Email	Status	Name	Description	Actions
<input type="checkbox"/> <a href="#">290202858156-compute@developer.gserviceaccount.com</a>	Enabled	Compute Engine default service account		
<input type="checkbox"/> <a href="#">databricks-server-community@polished-shore-450601-90.iam.gserviceaccount.com</a>	Enabled	databricks-server-community		

No change - principal already exists on the policy



Las credenciales creadas las seteamos en la configuración del cluster en databricks



Conectamos databricks con nuestro bucket de gcp

Python

Run allBruno F

Just now (1s)1

```
from pyspark.sql import SparkSession
from pyspark.sql.types import *

#Creamos las session de apache spark en una variable

spark = SparkSession.builder.getOrCreate()

ruta = 'gs://dmc_dde_barb/archivos/persona.data'

df_schema = StructType([
    StructField("ID", StringType(),True),
    StructField("NOMBRE", StringType(),True),
    StructField("TELEFONO", StringType(),True),
    StructField("CORREO", StringType(),True),
    StructField("FECHA_INGRESO", StringType(),True),
    StructField("EDAD", IntegerType(),True),
    StructField("SALARIO", DoubleType(),True),
    StructField("ID_EMPRESA", StringType(),True),
])

df_with_schema = spark.read.format("CSV").option("header","true").option("delimiter","|").schema(df_schema).load(ruta)
#df_with_schema.show()
display(df_with_schema)
```

(1) Spark Jobs

df\_with\_schema: pyspark.sql.dataframe.DataFrame = [ID: string, NOMBRE: string ... 6 more fields]

Table

	ID	NOMBRE	TELEFONO	CORREO	FECHA_ING...	EDAD	SALARIO	ID_EMPRESA
1	1	Carl	1-745-633-9145	arcu.Sed.et@ante.co.uk	2004-04-23	32	20095	5
2	2	Priscilla	155-2498	Donec.egestas.Aliquam@volutpatnunc.edu	2019-02-17	34	9298	2
3	3	Jocelyn	1-204-956-8594	amet.diam@lobortis.co.uk	2002-08-01	27	10853	3
4	4	Aidan	1-719-862-9385	euismod.et.commodo@nibhiaciniaorci.edu	2018-11-06	29	3387	10
5	5	Leandra	839-8044	at@pretiumtrutrum.com	2002-10-10	41	22102	1
6	6	Bert	797-4453	a.felis.uliamcorper@arcu.org	2017-04-25	70	7800	7
7	7	Mark	1-680-102-6792	Quisque.ac@placerat.ca	2006-04-21	52	8112	5
8	8	Jonah	214-2975	eu.ultrices.sit@vitae.ca	2017-10-07	23	17040	5
9	9	Hanae	935-2277	eu@Nunc.ca	2003-05-25	69	6834	3

Subimos un nuevo archivo a nuestro bucket que hemos descargado de kaggle y lo leemos desde databricks.

dmc\_dde\_barb

Location: us (multiple regions in United States) | Storage class: Standard | Public access: Not public | Protection: Soft Delete

OBJECTS | CONFIGURATION | PERMISSIONS | PROTECTION | LIFECYCLE | OBSERVABILITY | NEW | INVENTORY REPORTS | OPERATIONS

Folder browser

Buckets > dmc\_dde\_barb > archivos

CREATE FOLDER | UPLOAD | TRANSFER DATA | OTHER SERVICES

Filter by name prefix only | Filter: Filter objects and folders | Show: Live objects only

<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified	Public access	Version
<input type="checkbox"/>	netflix_titles.csv	3.4 MB	text/csv	Mar 10, 2025, 10:20:55 PM	Standard	Mar 10, 2025, 10:20:55 PM	Not public	—
<input type="checkbox"/>	persona.data	7.3 KB	application/octet-stream	Feb 26, 2025, 10:05:42 PM	Standard	Feb 26, 2025, 10:05:42 PM	Not public	—
<input type="checkbox"/>	salida/	—	Folder	—	—	—	—	—



Just now (2s)

Python

2

```
netflix_movies_ruta = 'gs://dmc_dde_barb/archivos/netflix_titles.csv'
df_netflix = spark.read.format("CSV").option("header", "true").option("delimiter", "," ).load(netflix_movies_ruta)
#df_with_schema.show()
#display(df_netflix)
df_netflix.show(20)
```

(2) Spark Jobs

df\_netflix: pyspark.sql.dataframe.DataFrame = [show\_id: string, type: string ... 10 more fields]

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NULL	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nea...
s2	TV Show	Blood & Water	NULL	Ama Qamata, Khosi...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV ...	After crossing pa...
s3	TV Show	Ganglands	Julien Leclercq	Samir Bouajila, Tr...	NULL	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, I...	To protect his fa...
s4	TV Show	Jailbirds New Ori...	NULL	NULL	NULL	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reali...	Feuds, flirtation...
s5	TV Show	Kota Factory	NULL	Mayur More, Jiten...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV ...	In a city of coac...
s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach...	NULL	September 24, 2021	2021	TV-MA	1 Season	TV Dramas, TV Hor...	The arrival of a ...
s7	Movie	My Little Pony: A...	Robert Cullen, Jo...	Vanessa Hudgens, ...	NULL	September 24, 2021	2021	PG	91 min	Children & Family...	Equestria's divid...
s8	Movie	Sankofa	Halle Gerima	Korfi Gnanaba, Oya...	United States, Gh...	September 24, 2021	1993	TV-MA	125 min	Dramas, Independe...	On a photo shoot ...
s9	TV Show	The Great British...	Andy Devonshire	Mel Giedroyc, Sue...	United Kingdom	September 24, 2021	2021	TV-14	9 Seasons	British TV Shows...	A talented batch ...
s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy...	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Dramas	A woman adjusting...
s11	TV Show	Vendetta: Truth, ...	NULL	NULL	NULL	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, D...	Sicily boasts a ...
s12	TV Show	Bangkok Breaking	Kongkiat Komesiri	Sukollawat Kanaro...	NULL	September 23, 2021	2021	TV-MA	1 Season	Crime TV Shows, I...	Struggling to ear...
s13	Movie	Je Suis Karl	Christian Schwuchow	Luna Medler, Jann...	Germany, Czech Re...	September 23, 2021	2021	TV-MA	127 min	Dramas, Internati...	After most of her...
s14	Movie	Confessions of an...	Bruno Garotti	Klara Castanho, L...	NULL	September 22, 2021	2021	TV-PG	91 min	Children & Family...	When the clever b...
s15	TV Show	Crime Stories: In...	NULL	NULL	NULL	September 22, 2021	2021	TV-MA	1 Season	British TV Shows...	Cameras following...
s16	TV Show	Dear White People	NULL	Logan Browning, B...	United States	September 22, 2021	2021	TV-MA	4 Seasons	TV Comedies, TV D...	Students of colo...
s17	Movie	Europe's Most Dan...	Pedro de Echave G...	NULL	NULL	September 22, 2021	2020	TV-MA	67 min	Documentaries, In...	Declassified docu...
s18	TV Show	Falsa Identidad	NULL	Luis Ernesto Fran...	Mexico	September 22, 2021	2020	TV-MA	2 Seasons	Crime TV Shows, S...	Strangers Diego a...