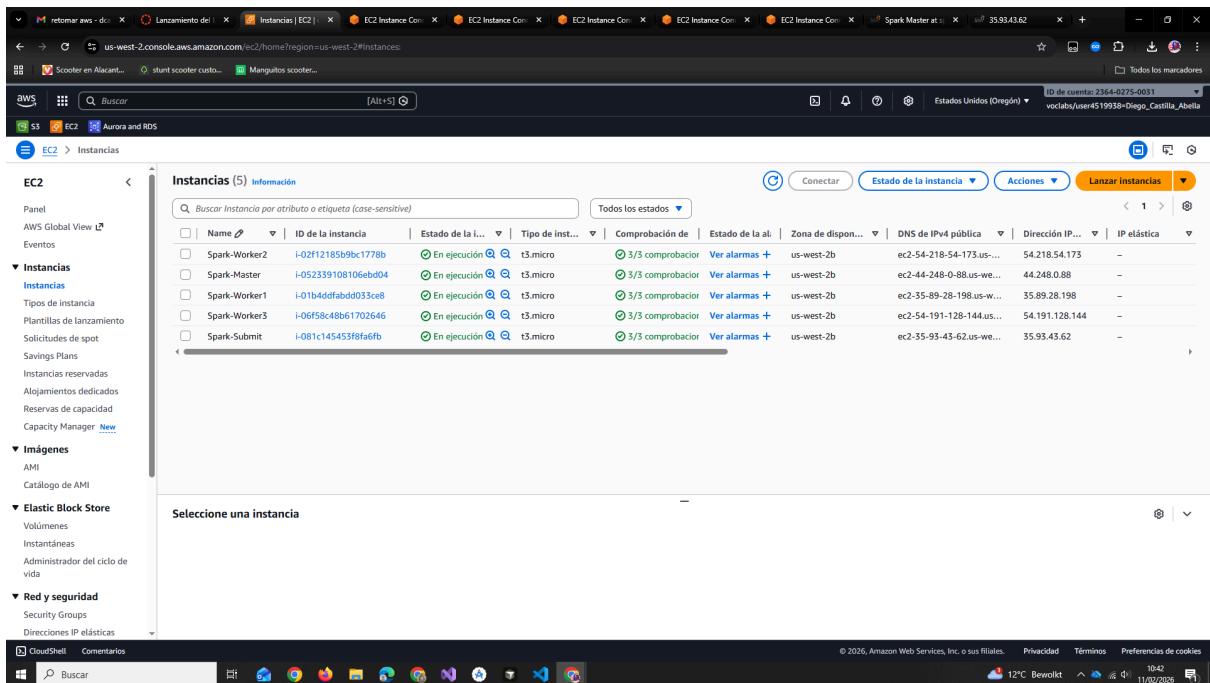


## 1. Preparación de la Infraestructura en AWS

El primer paso consistió en el despliegue de 5 instancias EC2 independientes para garantizar la separación de roles:

- Spark Master: Actúa como el orquestador de recursos del clúster.
  - Spark Workers: Son los nodos encargados de realizar el procesamiento pesado de datos de forma paralela.
  - Submit: La EC2 que utilizaremos para lanzar las aplicaciones mediante el comando spark-submit.
  - Configuración de Red: Se definieron Security Groups para permitir el tráfico en los puertos 7077 (RPC de Spark), 8080 (Web UI), y 3306 (acceso a la base de datos RDS).



## 2. Configuración del Entorno y Resolución de Dependencias

- Una vez operativas las máquinas, se instaló Spark 4.0.1 que al ser una versión tan reciente las librerías de hadoop aun no estaban integradas por lo que tuvimos que hacer el downgrade a la versión 3.5.0 la cual si incluye estos paquetes nativamente, ademas probe primero solo a hacer el downgrade en master y en submit pero entonces los workers no conectaban ya que spark a la hora de comunicarse es muy riguroso con las versiones del mismo. Hacemos downgrade a todo a 3.5.0 y estos ya se conectan y aparecen en el panel web del nodo master 8080.

Una vez con el script python : Spark no podía comunicarse con S3 nativamente. para solucionarlo se descargaron e integraron manualmente los archivos JAR hadoop-aws-3.3.4.jar y aws-java-sdk-bundle-1.12.262.jar para permitir el uso del protocolo s3a://. Además tambien se incluyó el driver de MySQL para permitir que Spark escribiera los resultados finales en la base de datos RDS.

The screenshot shows a browser window with multiple tabs open, all related to AWS services. The main content is the Spark Master UI at `spark://ip-172-31-30-219.us-west-2.compute.internal:7077`. The UI provides detailed information about workers, applications, and completed tasks.

**Workers (5)**

Worker Id	Address	State	Cores	Memory	Resources
worker-20260211091034-172.31.18.227-42921	172.31.18.227.42921	DEAD	2 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20260211092224-172.31.18.227-44979	172.31.18.227.44979	ALIVE	2 (2 Used)	1024.0 MiB (1024.0 MiB Used)	
worker-20260211092434-172.31.28.92-39393	172.31.28.92.39393	ALIVE	2 (2 Used)	1024.0 MiB (1024.0 MiB Used)	
worker-20260211092445-172.31.25.64-41763	172.31.25.64.41763	DEAD	2 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20260211092732-172.31.25.64-39297	172.31.25.64.39297	ALIVE	2 (2 Used)	1024.0 MiB (1024.0 MiB Used)	

**Running Applications (1)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20260211093215-0004	(kill) Top Products	6	1024.0 MiB		2026/02/11 09:32:15	ec2-user	RUNNING	11 s

**Completed Applications (4)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20260211092945-0003	Top Products	6	1024.0 MiB		2026/02/11 09:29:45	ec2-user	FINISHED	40 s
app-20260211092749-0002	Top Products	6	1024.0 MiB		2026/02/11 09:27:49	ec2-user	FINISHED	41 s
app-20260211092522-0001	Top Products	6	1024.0 MiB		2026/02/11 09:25:22	ec2-user	FINISHED	41 s
app-20260211091327-0000	Top Products	2	1024.0 MiB		2026/02/11 09:13:27	ec2-user	FINISHED	28 s



**Resumen**

Identificador de base de datos: myrds

Estado: Disponible

Role: Instancia

Motor: MySQL Community

CPU: 6.04%

Clase: db.t3.micro

Actividad actual: 0 Conexiones

Region y AZ: us-west-2a

Recomendaciones: 4 Informativo

**Conectividad y seguridad**

Conectar mediante: Fragmentos de código (selected), CloudShell, Puntos de conexión.

Lenguaje de programación: MySQL (macOS)

Tipo de punto de conexión: Punto de conexión de la instancia

Pasos de conexión:

```
mysql -h myrds.c0uzhxtteqvo.us-west-2.rds.amazonaws.com -P 3306 -u admin -p<Enter_DB_Password> --ssl-verify-server-cert --ssl-ca=/certs/global-bundle.pem mysql
```

Autenticación con IAM: Desactivado

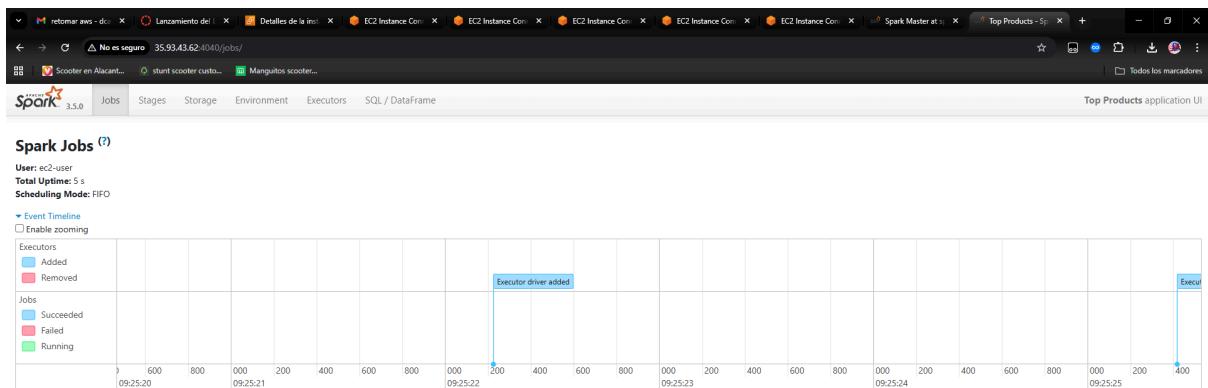
Recursos de computación conectados (0):

```

+-- Window [row_number()] windowSpecDefinition(Fecha$80, importe_total$108 DESC NULLS LAST, specifiedWindowFrame(RowFrame, unboundedpreceding(), currentrows())) AS ranking$114, [Fecha$80], [importe_total$108 DESC NULLS LAST]
  +- WindowUpdate([Fecha$80], [importe_total$108 DESC NULLS LAST], row_number(), 10, Final
    +-- Sort [Fecha$80 ASC NULLS FIRST, importe_total$108 DESC NULLS LAST], false, 0
      +-- Exchange hashpartitioning(Fecha$80, 200), ENSURE_REQUIREMENTS, [plan_id=745]
        +-- WindowGroupLimit ([Fecha$80], [importe_total$108 DESC NULLS LAST], row_number(), 10, Partial
          +-- Sort ([Fecha$80 ASC NULLS FIRST, importe_total$108 DESC NULLS LAST], false)
            +-- HashAggregate(keys=[Fecha$80, product_id$46], functions=[partial_sum(importe_total$80)], output=[Fecha$80, product_id$46, unidades$106L, importe_total$108])
              +-- HashPartition(keys=[Fecha$80, product_id$46], functions=[partial_sum(quantity$47)], output=[Fecha$80, product_id$46, sum$133L, sum$134])
                +-- HashAggregate(keys=[Fecha$80, product_id$46, quantity$47], functions=[partial_sum(quantity$47 * double) * unit_price$48], output=[Fecha$80, product_id$46, sum$168L, sum$170])
                  +-- Project [product_id$46, quantity$47, (cast(quantity$47 as double) * unit_price$48) AS importe_total$168, date_format(cast(order_date$20 as timestamp, yyyy-MM-dd, Some(UTC)) AS fecha$80)]
                    +-- BroadcastHashJoin([order_id$17], [order_id$45], Inner, BuildLeft, false
                      : BroadcastExchange HashedRelationBroadcastMode(List(cast(input0, int, false) as bigint)), false, [plan_id=734]
                      : +-- Filter isnotnull(order_id$17)
                      : +-- Filter isnotnull(order_id$45)
                      : +-- Sort [order_date$20 ASC NULLS FIRST, fecha$80 DESC NULLS LAST], false
                        +-- PushedFilters: [isNotNull(order_id$17)], DataFilters: [isNotNull(order_id$45)], Format: CSV, Location: InMemoryFileIndex(1 paths) [s3://mybucket-comercio360/comercio360/raw/orders.csv]
                        +-- PartitionFilters: [], PushedFilters: [isNotNull(order_id$17)], ReadSchema: struct<order_id:int,order_date:timestamp>
                          +-- FileScan csv [order_id$45,product_id$46,quantity$47,unit_price$48] Batched: false, DataFilters: [isNotNull(order_id$45)], Format: CSV, Location: InMemoryFileIndex(1 paths) [s3://mybucket-comercio360/comercio360/raw/order_items.csv], PartitionFilters: [], PushedFilters: [isNotNull(order_id$45)], ReadSchema: struct<order_id:int,product_id:int,quantity:int,unit_price:double>
Process finalizado con éxito.
26/02/11 09:13:55 INFO SparkContext: SparkContext is stopping with exitCode 0
26/02/11 09:13:55 INFO SparkContext: Stopping spark with url http://172.31.22.250.us-west-2.compute.internal:4040
26/02/11 09:13:55 INFO StandaloneSchedulerBackend: Shutting down all executors
26/02/11 09:13:55 INFO StandaloneSchedulerBackend: StandaloneDriverEndpoint: Asking each executor to shut down
26/02/11 09:13:55 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
26/02/11 09:13:55 INFO BlockManagerMaster: BlockManagerMaster stopped
26/02/11 09:13:55 INFO BlockManagerMaster: BlockManagerMaster stopped
26/02/11 09:13:55 INFO OutputCommitCoordinator: OutputCommitCoordinator stopped!
26/02/11 09:13:55 INFO SparkContext: Successfully stopped SparkContext
26/02/11 09:13:55 INFO MetricsSystemImpl: MetricsSystemImpl shutdown complete.
26/02/11 09:13:55 INFO ShutdownHookManager: Deleting directory /tmp/spark-f9ee037-f1cf-4630-9530-a6cef08901e5/pyspark-f28a03cb-6151-4f91-83fe-2e70553b664b
26/02/11 09:13:55 INFO ShutdownHookManager: Deleting directory /tmp/spark-346982a1-7256-4735-bca0-9b29328868
26/02/11 09:13:55 INFO ShutdownHookManager: Deleting directory /tmp/spark-f9ee037-f1cf-4630-9530-a6cef08901e5
26/02/11 09:13:55 INFO ShutdownHookManager: Deleting directory /tmp/spark-f9ee037-f1cf-4630-9530-a6cef08901e5
26/02/11 09:13:55 INFO MetricsSystemImpl: s3a-file-system metrics system stopped.
26/02/11 09:13:55 INFO MetricsSystemImpl: s3a-file-system metrics system shutdown complete.
[ec2-user@ip-172-31-22-250 ~]
```

i-081c145453f8fa6fb (Spark-Submit)

PublicIP: 55.93.45.6 PrivateIP: 172.31.22.250



### 3. Procesamiento de Datos (Flujo ETL)

El proceso de datos siguió el pipeline obligatorio S3 → Spark → RDS:

1. **Ingesta**: Spark leyó `orders.csv` y `order_items.csv` directamente desde el bucket de Amazon S3.
2. **Procesamiento**:
  - Se realizó un Join entre los pedidos y sus líneas de detalle para asociar productos y precios.
  - Se calcularon las ventas totales por producto y fecha.
  - Se aplicó una función de ventana (Window) para generar un ranking y filtrar únicamente el Top 10 diario por facturación.
3. **Persistencia**: Los resultados se guardaron en S3 en formato CSV para auditoría.
  - Simultáneamente, los datos se insertaron en la tabla `top_products` de Amazon RDS mediante una conexión JDBC en modo overwrite.

```

Details for Query 2
Submitted Time: 2026/02/11 09:32:39
Duration: 8 s
Succeeded Jobs: 4 5 6 7
Show the Stage ID and Task ID that corresponds to the max metric

Physical Plan
+-- SaveIntoDataSourceCommand (1)
  +-- SaveIntoDataSourceCommand (2)
    +-- Filter (3)
      +-- Project (11)
        +-- Project (12)
          +-- Window (11)
            +-- Project (10)
              +-- Aggregate (9)
                +-- Project (8)
                  +-- Project (7)
                    +-- Project (6)
                      +-- Join (5)
                        +-- LogicalRelation (3)
                        +-- LogicalRelation (4)

(1) Execute SaveIntoDataSourceCommand
Output: []

(2) SaveIntoDataSourceCommand
Arguments: org.apache.spark.sql.execution.datasources.jdbc.JdbcRelationProvider@61a78c29, [url=*****(@redacted), driver=com.mysql.cj.jdbc.Driver, dbtable=top_products, user=admin, password=*****(@redacted)], Overwrite

(3) LogicalRelation
Arguments: csv, [order_id#17, customer_id#18, store_id#19, order_date#20, payment_method#21], false

(4) LogicalRelation
Arguments: csv, [order_item_id#44, order_id#45, product_id#46, quantity#47, unit_price#48, discount#49], false

(5) Join
Arguments: Inner, [order_id#17 = order_id#45]

(6) Project
Arguments: [order_id#17, customer_id#18, store_id#19, order_date#20, payment_method#21, order_item_id#44, product_id#46, quantity#47, unit_price#48, discount#49]

(7) Project
Arguments: [order_id#17, customer_id#18, store_id#19, order_date#20, payment_method#21, order_item_id#44, product_id#46, quantity#47, unit_price#48, discount#49, (cast(quantity#47 as double) * unit_price#48) AS importe_total#68]

```

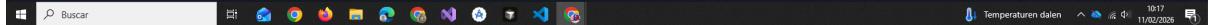
## 4. Análisis de Ejecución (DAG)

A través de la Spark Web UI, se analizó el Grafo Acíclico Dirigido (DAG) de la consulta:

- El sistema dividió el trabajo en varios Stages (etapas) para optimizar el movimiento de datos entre los workers.
- Se observó un nodo de tipo Exchange para agrupar los datos distribuidos por todo el clúster.
- Finalmente, el comando SaveIntoDataSourceCommand validó que el flujo terminó con éxito enviando los datos a la base de datos externa.

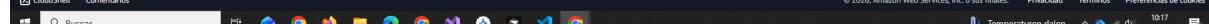
**resultado\_top\_productos/**

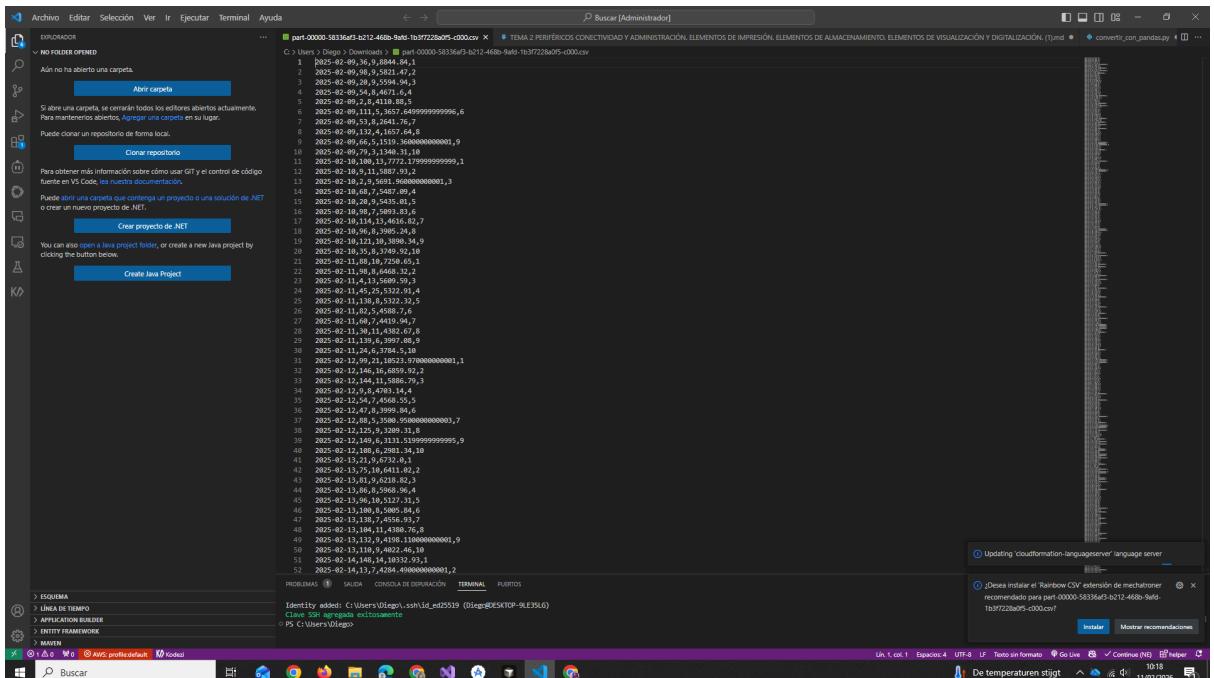
Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<a href="#">SUCCESS</a>	-	11 Feb 2026 10:13:56 AM CET	0 B	Estándar
<a href="#">part-00000-58336af3-b212-468b-9af5-1b3f7228a0f5-0000.csv</a>	csv	11 Feb 2026 10:13:55 AM CET	102.3 KB	Estándar



**comercio360/**

Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<a href="#">raw/</a>	Carpeta	-	-	-
<a href="#">resultado_to_p_productos/</a>	Carpeta	-	-	-





```

26/02/11 09:32:02 INFO MetricsSystemImpl: s3a-file-system metrics system stopped.
26/02/11 09:32:52 INFO MetricsSystemImpl: s3a-file-system metrics system shutdown complete.
[ec2-user@ip-172-31-22-250 ~]$ sudo nano top_products.py
[ec2-user@ip-172-31-22-250 ~]$ mysql -h myrds.c0uzhxtegwo.us-west-2.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 581
Server version: 8.4.7 | Source distribution

Copyright (c) 2000, 2026, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE comercio360;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM top_products LIMIT 10;
+-----+-----+-----+-----+
| fecha | product_id | unidades | importe_total | ranking |
+-----+-----+-----+-----+
| 2025-02-09 | 36 | 9 | 8844.84 | 1 |
| 2025-02-09 | 50 | 9 | 5594.94 | 2 |
| 2025-02-09 | 20 | 9 | 5594.94 | 3 |
| 2025-02-09 | 54 | 8 | 4671.6 | 4 |
| 2025-02-09 | 2 | 8 | 4110.88 | 5 |
| 2025-02-09 | 111 | 5 | 3657.649999999996 | 6 |
| 2025-02-09 | 53 | 8 | 2617.6 | 7 |
| 2025-02-09 | 10 | 4 | 1657.68 | 8 |
| 2025-02-09 | 66 | 5 | 1519.360000000001 | 9 |
| 2025-02-09 | 79 | 3 | 1340.31 | 10 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql> i-081c145453fbfa6fb (Spark-Submit)
PublicIPs: 35.93.43.62 PrivateIPs: 172.31.22.250

```

Diego Castilla Abella

## 5. Verificación de Resultados

El proceso finalizó en ambos servicios:

- **En S3:** Se generaron los archivos de resultados con la estructura de fecha, producto e importe.
- **En RDS:** Se ejecutó una consulta SQL (SELECT \* FROM top\_products LIMIT 10;) que confirmó que los 10 mejores productos diarios están correctamente almacenados y disponibles para aplicaciones de negocio.