PySpark job output from Dataproc into a MySQL database on GCP

Trainer: Nikhil Shah **Audience: Beginner**

Step 1: Set Up MySQL on GCP

You can use **Cloud SQL for MySQL**.

1.1. Create Cloud SQL Instance

- Go to Cloud SQL
- Click **"Create instance"** → **"MySQL"**
- Choose settings:
  - Instance ID: my-mysql-instance
  - Region: Same as your Dataproc cluster (for latency)
  - Root password: Set and note it
  - Public IP: Enable if needed

1.2. Create a Database and User

- Go to your instance → **Databases tab** → Add new DB (e.g., spark_output)
- Go to **Users tab** → Create user (e.g., sparkuser) with password

1.3. Whitelist Dataproc IP or use Private IP

If using **Public IP**, whitelist Dataproc's external IP. If using **Private IP**, ensure VPC peering between Cloud SQL and Dataproc.

Step 2: Create a Dataproc Cluster

- Go to Dataproc
- Click **"Create Cluster"**
- Choose the same region/VPC as Cloud SQL
- Add **JDBC driver** for MySQL via initialization actions or custom image:
  - Init action: gs://goog-dataproc-initialization-actions-us-central1/mysql/mysql.sh

---

Step 3: Upload PySpark Script to Cloud Storage

Example script: write_to_mysql.py

python

CopyEdit

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("WriteToMySQL").getOrCreate()

# Sample data
data = [("John", 100), ("Jane", 200)]
df = spark.createDataFrame(data, ["name", "amount"])

# MySQL config
jdbc_url = "jdbc:mysql://<INSTANCE_IP>:3306/spark_output"
properties = {
    "user": "sparkuser",
    "password": "yourpassword",
    "driver": "com.mysql.cj.jdbc.Driver"
}

# Write to MySQL
df.write.jdbc(url=jdbc_url, table="transactions", mode="overwrite", properties=properties)

spark.stop()
```

Upload to GCS:

bash

CopyEdit

```
gsutil cp write_to_mysql.py gs://your-bucket-name/scripts/
```

Step 4: Submit Job to Dataproc

bash

CopyEdit

```
gcloud dataproc jobs submit pyspark gs://your-bucket-name/scripts/write_to_mysql.py \
  --cluster=my-cluster \
  --region=your-region \
  --jars=gs://spark-lib/mysql/mysql-connector-java-8.0.33.jar
```

Update the JDBC connector JAR version as needed.

Step 5: Verify the Output in MySQL

- Connect to Cloud SQL using **Cloud SQL Auth Proxy** or SQL Workbench
- Check data in transactions table