LAB 04 - Data Storage Basics- Convert CSV to Parquet

To load inventory data from a CSV file into an S3 bucket in Parquet format using PySpark on a Spark cluster, follow these steps:

Prerequisites

1. **Spark Cluster**: Ensure you have access to a Spark cluster. You can use a local Spark setup, an EMR cluster on AWS, or any other Spark cluster.
2. **AWS CLI and Boto3**: Make sure you have the AWS CLI installed and configured with your credentials.

Steps

1. **Set up the Spark session**.
2. **Read the inventory CSV file (download from Kaggle) into a DataFrame**.
3. **Write the DataFrame to Parquet format**.
4. **Upload the Parquet file to S3**.

Pyspark Script

Here is the PySpark script save this in a python file called **delldatacsvtoparquet.py**:

```
from pyspark.sql import SparkSession
import boto3

# Initialize Spark session
spark = SparkSession.builder \
  .appName('dell-data-app-CSVtoParquet').master("local[*]") \
  .getOrCreate()

# Define your AWS credentials and S3 bucket details
AWS_ACCESS_KEY = 'your_access_key'
AWS_SECRET_KEY = 'your_secret_key'
S3_BUCKET_NAME = 'your_s3_bucket_name'
S3_KEY = 'path/to/save/your_file.parquet'
```

```
S3_OUTPUT_PATH = f's3a://{S3_BUCKET_NAME}/{S3_KEY}'

# Configure Spark to use S3
hadoop_conf = spark._jsc.hadoopConfiguration()
hadoop_conf.set("fs.s3a.access.key", AWS_ACCESS_KEY)
hadoop_conf.set("fs.s3a.secret.key", AWS_SECRET_KEY)
hadoop_conf.set("fs.s3a.endpoint", "s3.amazonaws.com")
hadoop_conf.set("fs.s3a.impl", "org.apache.hadoop.fs.s3a.S3AFileSystem")

# Load inventory data from CSV
csv_file_path = 'path/to/your/inventory.csv'
df = spark.read.csv(csv_file_path, header=True, inferSchema=True)

# Write DataFrame to Parquet format and save to S3
df.write.parquet(S3_OUTPUT_PATH)

print(f'File uploaded to S3 bucket {S3_BUCKET_NAME} at {S3_KEY}')

# Stop the Spark session
spark.stop()
```

Explanation

1. **Initialize Spark Session**:
    - Create a Spark session to run your job.
2. **AWS Credentials and S3 Configuration**:
    - Replace `AWS_ACCESS_KEY`, `AWS_SECRET_KEY`, `S3_BUCKET_NAME`, and `S3_KEY` with your actual values.
    - Configure the Hadoop settings for S3 access.
3. **Read CSV Data**:
    - Use `spark.read.csv` to read the CSV file into a DataFrame. Set `header=True` if your CSV has a header row and `inferSchema=True` to infer data types.
4. **Write to Parquet and Save to S3**:
    - Use the `write.parquet` method to convert the DataFrame to Parquet format and save it directly to the specified S3 path.

5. **Stop the Spark Session**:
    - Stop the Spark session to release resources.

Running the Script

1. Save the script to a Python file **delldatacsvtoparquet.py**.
2. Replace the placeholders for AWS credentials, S3 bucket name, CSV file path, and S3 key with your actual values.
3. Submit the script to your Spark cluster:

    bash
    Copy code
    spark-submit –master local[*] delldatacsvtoparquet.py

    - Replace <master-url> with the appropriate master URL for your Spark cluster incase of remote master.

This script will read your inventory data from a CSV file, convert it to Parquet format, and upload it to your specified S3 bucket using PySpark on a Spark cluster.