Week – 6 Containerize ETL Assignment

Piyush Kumar Yadav

To perform:

Containerize Analytics or ETL App
Goal:
Package the ETL or analysis app into a Docker container.
Tasks:
Write a Dockerfile for the Python/Flask or PySpark app.
Use docker-compose to run the app and a database (e.g., PostgreSQL).
Explain: What is the difference between a container and a virtual machine?
Tech: Docker, Flask / PySpark, Docker Compose
Deliverables: Dockerfile, docker-compose.yml, container demo + screenshots (build logs, running container list, browser view of app)

Link to github: https://github.com/friedcheesee/blendweek6allstar

What is the difference between a container and a virtual machine?

Container vs Virtual Machine

A virtual machine (VM) virtualizes hardware and runs a full guest operating system on top of a hypervisor. Each VM includes its own OS kernel, making it heavier and slower to start.

A container virtualizes the operating system rather than hardware. Containers share the host OS kernel and package only the application and its dependencies, making them lightweight and fast.

Containerization also helps solve the “it works on my machine” problem by ensuring consistent runtime environments across systems.

Aspect	Virtual Machines	Containers
Virtualize	Hardware	OS
Run on	Hypervisor	Container runtime
OS Kernel	Own kernel	Share Host OS kernel
Startup time	Slow	Fast
Resource Usage	High	Low
Isolation	Strong	Weaker than VM’s
Performance	Slight overhead	Way lesser Overhead
Portability	Across different OS hosts	Across environments with same kernel
Use Cases	Legacy Apps	Microservices, cloud native apps

Screenshots:

Docker build:

```
D:\dockerspark>docker-compose build
[+] Building 1.2s (12/12) FINISHED
 => [internal] load local bake definitions
 => => reading from stdin 504B
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 450B
 => [internal] load metadata for docker.io/apache/spark:4.1.0-scala2.13-java21-python3-r-ubuntu
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/apache/spark:4.1.0-scala2.13-java21-python3-r-ubuntu@sha256:c1dda16b06d33f2cfc2498fe52a52ff737374b881c503fd3ae941f1b8d27b4bd
 => => resolve docker.io/apache/spark:4.1.0-scala2.13-java21-python3-r-ubuntu@sha256:c1dda16b06d33f2cfc2498fe52a52ff737374b881c503fd3ae941f1b8d27b4bd
 => [internal] load build context
 => => transferring context: 5.32kB
 => CACHED [2/5] RUN apt-get update && apt-get install -y curl
 => CACHED [3/5] RUN curl -L -o /opt/spark/jars/postgresql.jar     https://jdbc.postgresql.org/download/postgresql-42.7.3.jar
 => CACHED [4/5] WORKDIR /app
 => CACHED [5/5] COPY spark_etl.py /app/spark_etl.py
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:db616ddd13feb91b523486537edf61843c0a52968ed2fef05cf0c1378f027fa9
 => => exporting config sha256:e5d87d165671833c46fcfba50c1f395820c91269db3b9ee6e202f4e383a5557d
 => => exporting attestation manifest sha256:6d66965dd6461293ed7d1c840f8aad7a2efb20ec9291d88d4ea557efcd448aa6
 => => exporting manifest list sha256:4961861357240830af6c519ad1132b3d2bfccee26998ff3c32e6bb2f9c119a6d
 => => naming to docker.io/library/dockerspark-spark-etl:latest
 => => unpacking to docker.io/library/dockerspark-spark-etl:latest
 => resolving provenance for metadata file
[+] Building 1/1
 ✓ dockerspark-spark-etl  Built
```

docker-compose up:

```
D:\dockerspark>docker-compose up
[+] Running 3/3
 ✓ Network dockerspark_default  Created
 ✓ Container postgres           Created
 ✓ Container spark-etl          Created
Attaching to postgres, spark-etl
postgres  | The files belonging to this database system will be owned by user "postgres".
postgres  | This user must also own the server process.
postgres  |
postgres  | The database cluster will be initialized with locale "en_US.utf8".
postgres  | The default database encoding has accordingly been set to "UTF8".
postgres  | The default text search configuration will be set to "english".
postgres  |
postgres  | Data page checksums are disabled.
postgres  |
postgres  | fixing permissions on existing directory /var/lib/postgresql/data ... ok
postgres  | creating subdirectories ... ok
postgres  | selecting dynamic shared memory implementation ... posix
postgres  | selecting default max_connections ... 100
postgres  | selecting default shared_buffers ... 128MB
postgres  | selecting default time zone ... Etc/UTC
postgres  | creating configuration files ... ok
postgres  | running bootstrap script ... ok
spark-etl | WARNING: Using incubator modules: jdk.incubator.vector
postgres  | performing post-bootstrap initialization ... ok
postgres  | syncing data to disk ... ok
postgres  |
postgres  |
postgres  | Success. You can now start the database server using:
postgres  |
postgres  |     pg_ctl -D /var/lib/postgresql/data -l logfile start
postgres  |
postgres  | initdb: warning: enabling "trust" authentication for local connections
```

```
spark-etl  | 25/12/28 16:36:55 INFO DAGScheduler: Missing parents found for ResultStage 65: List()
spark-etl  | 25/12/28 16:36:55 INFO DAGScheduler: Submitting ResultStage 65 (MapPartitionsRDD[125] at showString at NativeMethodAccessorImpl.java:0), which has no missing parents
spark-etl  | 25/12/28 16:36:55 INFO MemoryStore: Block broadcast_47 stored as values in memory (estimated size 16.6 KiB, free 433.9 MiB)
spark-etl  | 25/12/28 16:36:55 INFO MemoryStore: Block broadcast_47_piece0 stored as bytes in memory (estimated size 7.0 KiB, free 433.9 MiB)
spark-etl  | 25/12/28 16:36:55 INFO SparkContext: Created broadcast 47 from broadcast at DAGScheduler.scala:1701
spark-etl  | 25/12/28 16:36:55 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 65 (MapPartitionsRDD[125] at showString at NativeMethodAccessorImpl.java:0) (first 15 tasks are for partitions Vector(0))
spark-etl  | 25/12/28 16:36:55 INFO TaskSchedulerImpl: Adding task set 65.0 with 1 tasks resource profile 0
spark-etl  | 25/12/28 16:36:55 INFO TaskSetManager: Starting task 0.0 in stage 65.0 (TID 41) (0acae1ea3238,executor driver, partition 0, PROCESS_LOCAL, 10298 bytes)
spark-etl  | 25/12/28 16:36:55 INFO Executor: Running task 0.0 in stage 65.0 (TID 41)
spark-etl  | 25/12/28 16:36:55 INFO FileScanRDD: Reading File path: file:///app/output/size_market_share/part-00000-ba3cbcca-197e-4c89-b389-f8175360140f-c000.snappy.parquet, range: 0-1218, partition values: [empty row]
spark-etl  | 25/12/28 16:36:56 INFO Executor: Finished task 0.0 in stage 65.0 (TID 41). 2102 bytes result sent to driver
spark-etl  | 25/12/28 16:36:56 INFO TaskSetManager: Finished task 0.0 in stage 65.0 (TID 41) in 28 ms on 0acae1ea3238 (executor driver) (1/1)
spark-etl  | 25/12/28 16:36:56 INFO TaskSchedulerImpl: Removed TaskSet 65.0 whose tasks have all completed, from pool
spark-etl  | 25/12/28 16:36:56 INFO DAGScheduler: ResultStage 65 (showString at NativeMethodAccessorImpl.java:0) finished in 30 ms
spark-etl  | 25/12/28 16:36:56 INFO DAGScheduler: Job 41 is finished. Cancelling potential speculative or zombie tasks for this job
spark-etl  | 25/12/28 16:36:56 INFO TaskSchedulerImpl: Cancelling stage 65
spark-etl  | 25/12/28 16:36:56 INFO TaskSchedulerImpl: Killing all running tasks in stage 65: Stage finished
spark-etl  | 25/12/28 16:36:56 INFO DAGScheduler: Job 41 finished: showString at NativeMethodAccessorImpl.java:0, took 31.133647 ms
spark-etl  | +----+------------+---------------+
spark-etl  | |size|size_revenue|market_share_pct|
spark-etl  | +----+------------+---------------+
spark-etl  | |   M| 1.3302545E7|          17.64|
spark-etl  | |   L| 1.2684744E7|          16.82|
spark-etl  | |  XL| 1.1917813E7|          15.81|
spark-etl  | | XXL| 1.0258118E7|           13.6|
spark-etl  | |   S| 1.0186895E7|          13.51|
spark-etl  | | 3XL|   8795455.0|          11.66|
spark-etl  | |  XS|   6759023.0|           8.96|
spark-etl  | | 6XL|    562718.0|           0.75|
spark-etl  | | 5XL|    414809.0|           0.55|
spark-etl  | | 4XL|    325318.0|           0.43|
spark-etl  | |Free|    195068.0|           0.26|
spark-etl  | +----+------------+---------------+
spark-etl  |
spark-etl  | AOV: 711.67
spark-etl  | Cancellation Rate: 16.22%
spark-etl  | Average Basket Size: 1.08
spark-etl  | Spark UI available at http://localhost:4040
```

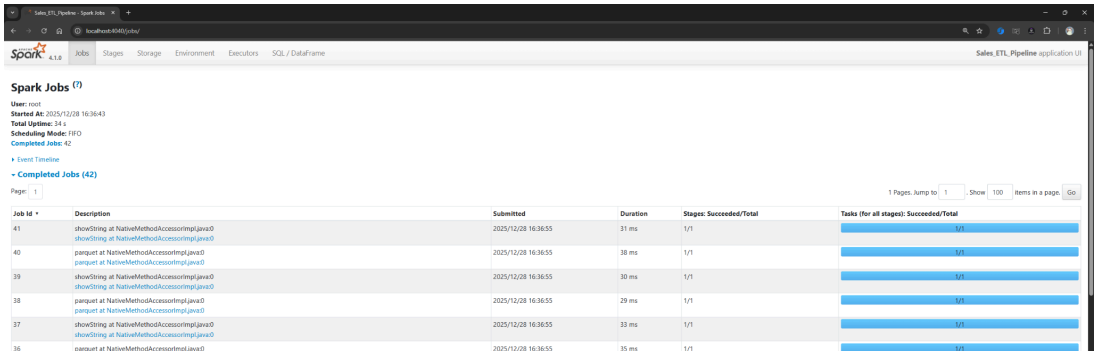
Active Containers:

```
D:\dockerspark>docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED              STATUS              PORTS                                              NAMES
0acae1ea3238   dockerspark-spark-etl  "/opt/entrypoint.sh …"   About a minute ago   Up About a minute   0.0.0.0:4040->4040/tcp, [::]:4040->4040/tcp        spark-etl
c1e520157096   postgres:15            "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp        postgres

D:\dockerspark>
```

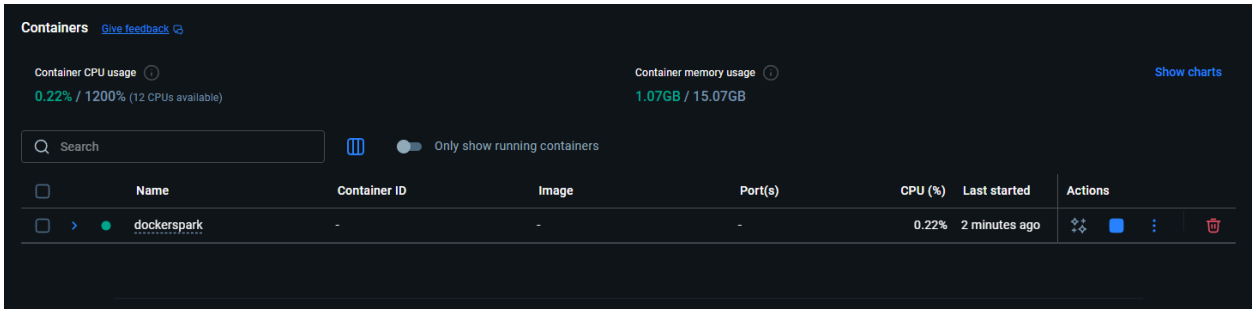
Postgres:

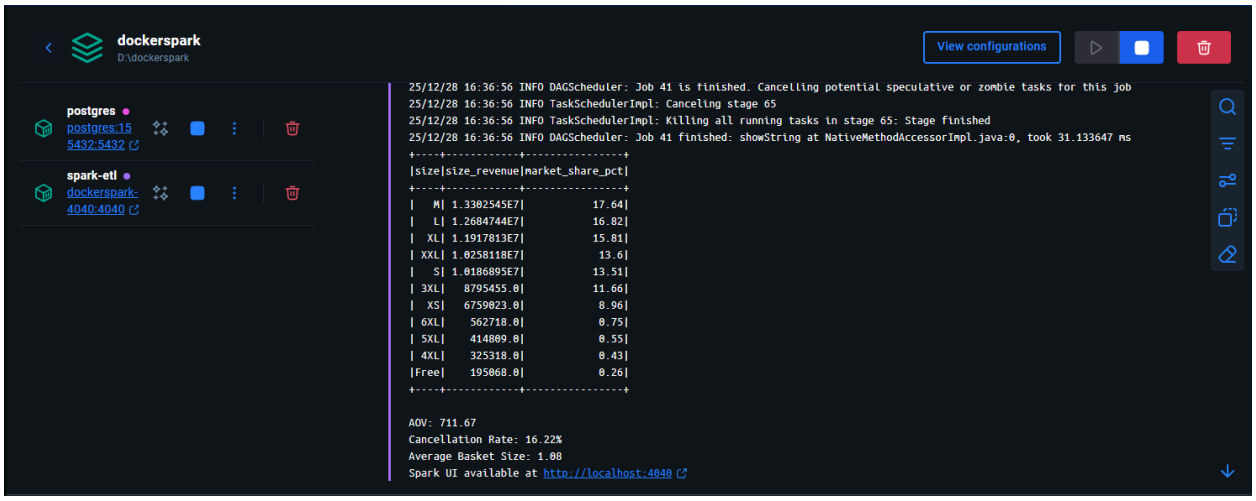
```
piyush@DESKTOP-CFDCLVV:/mnt/d/dockerspark$ docker exec -it postgres psql -U salesuser -d salesdb
psql (15.15 (Debian 15.15-1.pgdg13+1))
Type "help" for help.

salesdb=# SELECT COUNT(*) FROM sales;
 count
--------
 128975
(1 row)

salesdb=#
```

Spark UI:

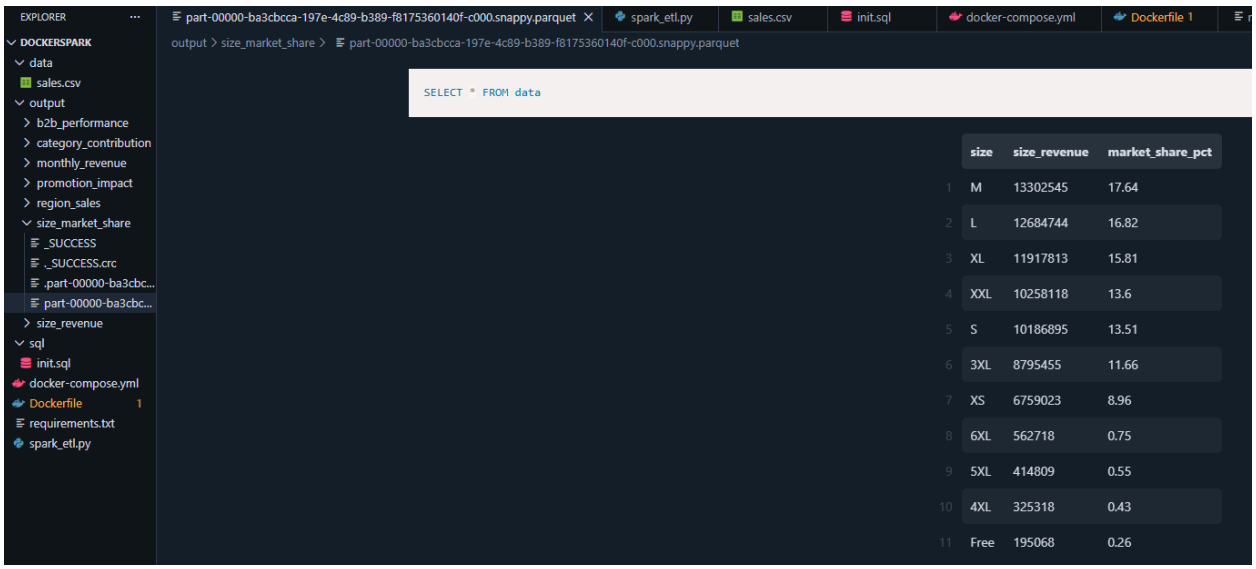


Docker Desktop UI:





Parquet file:



Dockerfile:

```
FROM apache/spark:4.1.0-scala2.13-java21-python3-r-ubuntu

USER root
#curl
RUN apt-get update && apt-get install -y curl

#jdbc driver
RUN curl -L -o /opt/spark/jars/postgresql.jar \
    https://jdbc.postgresql.org/download/postgresql-42.7.3.jar
WORKDIR /app
COPY spark_etl.py /app/spark_etl.py
CMD ["/opt/spark/bin/spark-submit", "/app/spark_etl.py"]
```

Docker-compose:

```
services:
  postgres:
    image: postgres:15
    container_name: postgres
    environment:
      POSTGRES_DB: salesdb
      POSTGRES_USER: salesuser
      POSTGRES_PASSWORD: salespass
    volumes:
      - ./data:/data
      - ./sql/init.sql:/docker-entrypoint-initdb.d/init.sql
    ports:
      - "5432:5432"

  spark-etl:
    build: .
    container_name: spark-etl
    depends_on:
      - postgres
    volumes:
      - ./output:/app/output
    ports:
      - "4040:4040"
```