✅ Step-by-Step Exercise Breakdown

🔹 Step 1: Design a Simple Talend Job

Objective: Fetch data from a Snowflake table and display using tLogRow.

Components Used:

- tSnowflakeInput
- tLogRow

Configuration:

- Use a sample table like DEMO_DB.PUBLIC.EMPLOYEES
- Test the connection to Snowflake using Talend Studio metadata

Job Name:

Sales_ETL_Job

Export:

1. Go to Job Designs → Right-click → Build Job
2. Build as Standalone Job (.bat or .sh)
   → Target directory: Sales_ETL_Job_0.1

✅ Airflow + Talend Job in Docker (Windows)

📦 **Overview**

You will:

1. Install & configure Airflow using Docker
2. Add Java 17 support for Talend .jar execution
3. Mount your Talend job to the Airflow container
4. Create a scheduled Airflow DAG that triggers the job
5. Run and monitor the DAG via Airflow UI

✅ **1. Project Structure (Local Folder Layout)**

Create a folder C:\airflow-docker like this:

```
C:\airflow-docker\
├── dags\
│   └── run_sales_etl_dag.py        # your DAG
├── talend_jobs\
│   └── Sales_ETL_Job_0.1\
│       └── Sales_ETL_Job\
│           ├── Sales_ETL_Job_run.sh
│           └── Sales_ETL_Job.jar
├── logs\
├── plugins\
├── Dockerfile                      # for Java 17
└── docker-compose.yaml             # main compose file
```

✅ **2. Create Dockerfile with Java 17**

Save this as: C:\airflow-docker\Dockerfile

```
FROM apache/airflow:2.9.1
USER root
RUN apt-get update && apt-get install -y openjdk-17-jre && apt-get clean
ENV JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64
ENV PATH=$JAVA_HOME/bin:$PATH
USER airflow
```

✅ **3. Create docker-compose.yaml**

Save this in C:\airflow-docker\docker-compose.yaml

```
version: '3'
x-airflow-common:
  &airflow-common
  build:
    context: .
    dockerfile: Dockerfile
  image: airflow-with-java17:latest
```

```
  environment:
    AIRFLOW__CORE__EXECUTOR: LocalExecutor
    AIRFLOW__DATABASE__SQL_ALCHEMY_CONN:
postgresql+psycopg2://airflow:airflow@postgres/airflow
    AIRFLOW__CORE__FERNET_KEY: 'qW3VbK8YLOIRklZxNqA8dMZ1g=='
    AIRFLOW__CORE__DAGS_ARE_PAUSED_AT_CREATION: 'false'
    AIRFLOW__CORE__LOAD_EXAMPLES: 'false'
  volumes:
    - ./dags:/opt/airflow/dags
    - ./logs:/opt/airflow/logs
    - ./plugins:/opt/airflow/plugins
    - ./talend_jobs:/opt/airflow/talend_jobs
  depends_on:
    - postgres
  user: "${AIRFLOW_UID:-50000}:0"

services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow
      POSTGRES_DB: airflow
    volumes:
      - postgres-db-volume:/var/lib/postgresql/data

  airflow-webserver:
    <<: *airflow-common
    ports:
      - "8080:8080"
    command: webserver
```

```
  airflow-scheduler:
    <<: *airflow-common
    command: scheduler

  airflow-init:
    <<: *airflow-common
    command: >
      bash -c "
      airflow db migrate &&
      airflow users create --username admin --firstname Admin --lastname User --role Admin --email admin@example.com --password admin
"
volumes:
  postgres-db-volume:
```

✅ **4. Sample Talend Script (Sales_ETL_Job_run.sh)**

Place this inside:
C:\airflow-docker\talend_jobs\Sales_ETL_Job_0.1\Sales_ETL_Job\Sales_ETL_Job_run.sh

✅ Make sure this .sh is executable inside the container (we'll do it manually if needed)

✅ **5. Create DAG: run_sales_etl_dag.py**

Save this inside:
C:\airflow-docker\dags\run_sales_etl_dag.py

```
from airflow import DAG
from airflow.operators.bash import BashOperator
from datetime import datetime, timedelta

default_args = {
    'owner': 'airflow',
    'retries': 1,
```

```
    'retry_delay': timedelta(minutes=1),
}

with DAG(
    dag_id='run_sales_etl',
    default_args=default_args,
    description='Run Talend job using BashOperator',
    start_date=datetime(2023, 1, 1),  # ✅ Must be in past
    schedule_interval='@daily',
    catchup=False,
) as dag:

    run_job = BashOperator(
        task_id='run_talend_job',
        bash_command="bash -c '/opt/airflow/talend_jobs/Sales_ETL_Job_0.1/Sales_ETL_Job/Sales_ETL_Job_run.sh'"
    )
```

✅ 6. Build & Start Airflow with Java + Talend

A. Build image (with Java 17):

```
cd C:\airflow-docker
docker compose build
```

B. Initialize DB & create Airflow user:

```
docker compose up airflow-init
```

C. Start all Airflow services:

```
docker compose up -d
```

✅ 7. Make Talend Script Executable (1st time only)

```
docker exec -it airflow-docker-airflow-webserver-1 bash
chmod +x /opt/airflow/talend_jobs/Sales_ETL_Job_0.1/Sales_ETL_Job/Sales_ETL_Job_run.sh
exit
```

✅ 8. Open Airflow UI & Trigger DAG

👉 Visit: http://localhost:8080
Login: admin / admin

- Enable run_sales_etl DAG
- Click **Trigger DAG**
- View logs → you should see:

[Output from tLogRow or your Talend job]

✅ 9. Verify Scheduled Trigger Works

- Scheduled runs happen automatically every day
- If needed, adjust start_date and catchup settings to control run behavior