# 🧰 Data Engineering Debugging Playbook: Errors & Fixes from Real Projects

A curated playbook of real-world data engineering issues I encountered and how I fixed them, based on projects like hospital pipelines, Snowflake integrations, and Airflow/dbt orchestration. This guide is written to help others avoid the same traps — and to serve as a personal reference.

## 📦 Docker & Containers

### ❌ Issue: Airflow or dbt container not starting

**Symptom:** airflow\_webserver missing in docker ps -a, dbt container exits immediately.

**Fixes:**

* Added airflow-init container in docker-compose.yml.
* Created persistent entrypoint for dbt:
* ENTRYPOINT ["tail", "-f", "/dev/null"]
* Ensured webserver/scheduler depends on airflow-init.

### ❌ Issue: Port 8080 not accessible

**Symptom:** localhost:8080 not loading despite container running.

**Fix:**

* Ran lsof -i :8080 to find conflicting process.
* Killed with kill -9 <PID>.
* Rebuilt Docker:
* docker compose down --volumes --remove-orphans  
  docker compose up --build

### ✅ Tip:

Always run docker ps -a and docker logs <container> when things go wrong.

## 📋 Environment Variables (.env)

### ❌ Issue: Env vars not available in container

**Symptom:** Python script throws Missing env vars: SNOWFLAKE\_WAREHOUSE

**Fixes:**

* Added env\_file: ../.env in docker-compose.yml
* Verified inside container:
* env | grep SNOWFLAKE
* Added Python support:
* from dotenv import load\_dotenv  
  load\_dotenv()

### ✅ Tip:

Always double-check spelling and file location of .env before restarting.

## ❄️ Snowflake Connectivity

### ❌ Issue: SAML/SSL connection error with snowsql

**Symptom:**

There was an error related to the SAML Identity Provider account parameter.

**Fixes:**

* Your Snowflake URL (web) is NOT your account string.
* Correct format:
* SNOWFLAKE\_ACCOUNT=orgname-accountname
* Use:
* snowsql -a orgname-accountname -u username --authenticator externalbrowser

### ❌ Issue: Schema does not exist

**Symptom:** SQLAlchemy throws:

Schema 'PUBLIC' does not exist or not authorized.

**Fixes:**

* Set role, db, and schema in .env:
* SNOWFLAKE\_ROLE=ACCOUNTADMIN  
  SNOWFLAKE\_DB=MY\_DATABASE  
  SNOWFLAKE\_SCHEMA=PUBLIC
* Use:
* if inspector.has\_table(table\_name, schema=schema):

## ⚙️ Airflow DAG Debugging

### ❌ Issue: DAG not triggering or failing silently

**Fixes:**

* Added task logs and retries.
* Used airflow dags test <dag\_id> <date> for manual debug.
* Checked /opt/airflow/logs inside container.

## 🧪 Great Expectations

### ❌ Issue: Checkpoint fails with PostgreSQL

**Symptom:** Errors loading data\_asset or query.

**Fixes:**

* Updated checkpoint.yml with proper datasource\_name, query, and batch\_identifiers
* Verified database table name matches expectation.

### ✅ Tip:

Checkpoint errors usually stem from incorrect batch\_request or query syntax.

## 📁 File Handling

### ❌ Issue: CSV not found inside container

**Symptom:** FileNotFoundError: /path/to/file.csv

**Fix:**

* Ensure file is mapped via Docker volume:
* volumes:  
   - ../data:/opt/airflow/data
* Verify existence inside container:
* ls /opt/airflow/data

## ✅ Summary: Best Practices

| Category | Best Practice |
| --- | --- |
| Docker | Always check logs and container status |
| Airflow | Set proper dependencies and run airflow-init |
| dbt | Add tail -f entrypoint to keep container alive |
| Snowflake | Validate account string, role, and schema explicitly |
| Env Variables | Load via dotenv, validate with env command |
| Debugging | Use lsof, kill, and container logs proactively |

🎯 This playbook will grow as I build more pipelines. If you’re learning data engineering through projects — save this, use this, and keep iterating.

✉️ Ping me if you’d like to collaborate on debugging diaries or data engineering use cases!