**🧭 Full End-to-End Walkthrough: Reddit to AWS Data Pipeline using Airflow & Docker**

**🟢 1. Environment Setup (Local Dev)**

**📁 Directory & File Structure**

The project starts with setting up a clean modular directory:

bash

CopyEdit

mkdir config dags data etls logs pipelines tests utils

touch airflow.env docker-compose.yml Dockerfile

pip freeze > requirements.txt

docker compose up -d --build

**Explanation:**

* config/ → contains your config.conf with credentials and path settings.
* dags/ → stores Airflow DAGs like reddit\_dag.py.
* etls/ → for ETL logic like reddit\_etl.py, aws\_etl.py.
* pipelines/ → holds pipeline wrappers like reddit\_pipeline.py, aws\_s3\_pipeline.py.
* data/ → input/output data (raw, transformed).
* logs/ → Airflow execution logs.
* tests/ → for test scripts (optional).
* utils/ → helper files like constants.py.

**🐳 2. Docker + Airflow Infrastructure**

**🛠 docker-compose.yml​**

Spins up:

* Airflow Webserver → UI at http://localhost:8080
* Airflow Scheduler → Runs tasks on schedule.
* Airflow Worker → Executes Python tasks (Celery-based).
* PostgreSQL → Backend DB for Airflow metadata.
* Redis → Message broker for Celery.

bash

CopyEdit

airflow-init # initializes DB and admin user

🔁 volumes map your code folders into the container.

**🧠 3. Reddit Extraction via Airflow DAG**

**🔧 reddit\_dag.py​**

Defines a DAG etl\_reddit\_pipeline that runs daily and does:

**✅ Task 1: reddit\_extraction**

Calls reddit\_pipeline() which does:

* Authenticates Reddit API using praw via connect\_reddit()​
* Pulls top 100 posts from r/dataengineering using extract\_posts()
* Transforms the posts:
  + Fixes timestamps
  + Handles booleans and nulls
  + Ensures type consistency
* Saves CSV to /opt/airflow/data/output/reddit\_<date>.csv

**🧩 Code from reddit\_pipeline.py​ calls:**

* connect\_reddit(), extract\_posts(), transform\_data(), load\_data\_to\_csv() from reddit\_etl.py​

**☁️ 4. Uploading CSV to AWS S3**

**✅ Task 2: s3\_upload**

* Calls upload\_s3\_pipeline()​
* Which in turn calls:
  + connect\_to\_s3() using s3fs
  + create\_bucket\_if\_not\_exist() for dataeng-reddit
  + upload\_to\_s3(): Uploads file to:

bash

CopyEdit

s3://dataeng-reddit/raw/reddit\_<date>.csv

💡 AWS keys are pulled from config/config.conf​ via constants.py​

**🔁 5. AWS Glue – Transforming S3 Data**

**Manual Steps in AWS Console​**

1. Open **Glue Studio** → Create job (visual/script)
2. Input location: s3://dataeng-reddit/raw/
3. Apply transformations (drop columns, change types, clean data)
4. Output to: s3://dataeng-reddit/transformed/

Example: reddit\_cleaned\_20250403.csv

**📚 6. AWS Glue Crawler – Metadata Catalog**

1. Create a **database** in Glue, e.g., reddit\_db
2. Create a **crawler**:
   * Source: transformed/ folder
   * Target: reddit\_db
   * IAM Role: Grant S3 + Glue access
3. Run crawler → Creates Athena-compatible table

**🔍 7. Query in Amazon Athena**

* Open **Athena Console**
* Select reddit\_db, table created by crawler
* Run SQL queries directly on the transformed CSV

Example:

sql

CopyEdit

SELECT title, score FROM reddit\_transformed WHERE score > 100 ORDER BY score DESC;

✅ Serverless, fast analytics.

**🧱 8. (Optional) Load to Amazon Redshift**

Use:

* Glue Connector OR
* COPY command in Redshift

Example:

sql

CopyEdit

COPY reddit\_transformed FROM 's3://dataeng-reddit/transformed/'

CREDENTIALS 'aws\_access\_key\_id=...;aws\_secret\_access\_key=...'

CSV IGNOREHEADER 1;

Great for **BI dashboards**, **warehousing**, **joins**, etc.

**📊 9. Visualization Using BI Tools**

You can now connect:

* **Power BI** → via Athena or Redshift ODBC
* **Tableau** → Amazon Athena connector
* **Amazon QuickSight** → integrates natively with Athena/Redshift
* **Looker Studio** → via BigLake/Federated sources

📈 Create charts, trends, KPIs like:

* Top posts
* Engagement by author
* NSFW % breakdown
* Daily upvote trends

**🔐 IAM & Security Best Practices**

* Create a dedicated **IAM Role** for Glue with:
  + AmazonS3FullAccess
  + AWSGlueServiceRole
* Store credentials securely via **AWS Secrets Manager** in production.
* Rotate Reddit API keys periodically.

**🗺️ Diagram (Optional Visual Summary)**

plaintext

CopyEdit

Reddit API

↓

Airflow DAG (Reddit DAG)

↓

reddit\_pipeline.py

↓

CSV file (local)

↓

aws\_s3\_pipeline.py

↓

S3 bucket: /raw

↓

Glue Job

↓

S3 bucket: /transformed

↓

Glue Crawler

↓

Glue Data Catalog

↓

Athena (SQL Queries)

↓

Redshift (optional)

↓

BI tools (Power BI, Tableau, etc.)

**✅ Final Recap Table**

| **Step** | **Tool** | **Description** |
| --- | --- | --- |
| 0 | Docker Compose | Starts local dev environment |
| 1 | Airflow DAG | Schedules the pipeline |
| 2 | praw + pandas | Extract & clean Reddit data |
| 3 | s3fs | Upload to AWS S3 |
| 4 | AWS Glue | Data transformation job |
| 5 | AWS Glue Crawler | Catalogs the transformed data |
| 6 | Athena | Run SQL queries on S3 |
| 7 | Redshift | Optional data warehousing |
| 8 | Power BI, Tableau | Dashboard and visual insights |