**🗓️ 6-Week Airflow Mastery Plan (for Data Engineers)**

**🎯 Overall Goal**

Master Apache Airflow for real-world data engineering — from DAG design to orchestration, monitoring, and cloud deployment.

**WEEK 1 — Core Airflow Foundations**

**Goal:** Solidify fundamentals and architecture.

**Topics**

* Airflow architecture (Scheduler, Webserver, Worker, Metadata DB)
* Core concepts: DAGs, Tasks, Operators, Hooks, Sensors, XComs
* Task dependencies (>>, <<, chain)
* Airflow CLI & UI basics
* Variables, Connections, and configuration files
* Docker Compose setup for Airflow

**Practice**

* Build a simple DAG that prints and logs data.
* Add scheduling (daily) and retry logic.

**Outcome**  
You can confidently write and run simple DAGs locally in Docker.

**WEEK 2 — Operators, Sensors, and XComs**

**Goal:** Deep dive into task logic and data flow.

**Topics**

* Common operators: PythonOperator, BashOperator, EmailOperator
* File and database operators (e.g. S3FileTransformOperator, SnowflakeOperator)
* Sensors (waiting for files, tables, or conditions)
* Using XComs for passing data between tasks

**Practice**

* Build a DAG that:
  + Downloads a file
  + Processes it
  + Uploads it to S3
  + Uses a sensor to wait for the upload

**Outcome**  
You can handle task dependencies, data exchange, and sensors.

**WEEK 3 — Advanced DAG Design**

**Goal:** Learn modularity, parameterization, and scalability.

**Topics**

* Dynamic DAG generation
* Branching, SubDAGs, TaskGroups
* Jinja templating
* Macros (ds, ts, etc.)
* Parameterized DAGs (config-driven)

**Practice**

* Build a dynamic DAG that processes multiple files or tables based on configuration.
* Use templates to pass runtime parameters.

**Outcome**  
You can design scalable DAGs that adapt to real-world variability.

**WEEK 4 — Integrations & ETL Pipelines**

**Goal:** Create data pipelines with external systems.

**Topics**

* Connecting Airflow with:
  + AWS S3 (via S3Hook)
  + Snowflake or Redshift
  + APIs (HTTP hooks)
* Handling secrets with Airflow Connections
* Logging and error handling in production DAGs

**Practice**

* Build an ETL DAG:
  + Extract data from an API
  + Transform using Python or Pandas
  + Load to Snowflake/S3

**Outcome**  
You can build reliable ETL workflows using Airflow’s built-in hooks.

**WEEK 5 — Monitoring, Scaling, and Best Practices**

**Goal:** Move toward production-quality orchestration.

**Topics**

* Monitoring and alerting (Email, Slack)
* Airflow Logs & metrics
* DAG performance optimization
* Parallelism and concurrency control
* CI/CD for Airflow (Git-based workflow)
* Testing DAGs (with pytest and airflow test)

**Practice**

* Add Slack alerts to a DAG.
* Add retries, logging, and error-handling best practices.
* Create a GitHub repo with proper DAG structure.

**Outcome**  
You can maintain, monitor, and debug Airflow pipelines confidently.

**WEEK 6 — Deploying Airflow on AWS (Production Environment)**

**Goal:** Move from local Docker to a cloud deployment.

**Topics**

* Running Airflow on AWS ECS/Fargate using Docker image
* Using AWS RDS for metadata database (Postgres)
* Connecting Airflow to AWS S3 (for DAGs/logs)
* Airflow Variables and Connections in production
* Role-based Access Control and IAM permissions

**Optional (for advanced touch):**

* MWAA (Managed Workflows for Apache Airflow) overview
* Setting up a small demo environment on MWAA

**Practice**

* Package your Dockerized Airflow setup.
* Deploy it to ECS/Fargate using AWS CLI or Terraform.
* Store DAG files in S3 and logs in CloudWatch.

**Outcome**  
You can deploy and manage Airflow pipelines in AWS — production-ready and cloud-native.

**📆 Suggested Timeline (40 minutes/day)**

| **Week** | **Focus** | **Output** |
| --- | --- | --- |
| 1 | Airflow core concepts & setup | Local Airflow running DAGs |
| 2 | Operators, Sensors, XComs | Functional multi-step DAG |
| 3 | Dynamic & modular DAGs | Config-driven pipeline |
| 4 | External integrations | API → Transform → Snowflake/S3 ETL |
| 5 | Monitoring & optimization | Alerting, logging, repo structure |
| 6 | Cloud deployment | Airflow on AWS ECS/Fargate or MWAA |

**💡 Bonus (Optional after Week 6)**

If you want to go beyond “entry-level”:

* Implement **Airflow + dbt** orchestration.
* Use **Airflow REST API** for triggering runs externally.
* Add **Data Quality Checks** (Great Expectations or SQL checks).
* Study **DAG versioning and backfills**.