Using Airflow on Google Cloud Platform (GCP)

Abstract

This note analyzes hosting Apache Airflow on Google Cloud Platform (GCP) using Cloud Composer, comparing it with AWS Managed Workflows for Apache Airflow (MWAA) and Azure Managed Airflow. We model the utility of each option based on native integrations, operational overhead, upgrade risk, and scalability. Cloud Composer leverages GKE Autopilot for autoscaling compute and Cloud SQL for managed metadata storage, providing first-party integrations with BigQuery, Dataflow, Dataproc, Pub/Sub, Secret Manager, and Vertex AI. We recommend Cloud Composer for workloads primarily on GCP due to its 99.5% SLO and reduced ops toil, while AWS MWAA or Azure Managed Airflow may be preferable when data and teams are already entrenched in those clouds.

Glossary

Cloud Composer Google Cloud's managed Apache Airflow service.

GKE Autopilot Managed Kubernetes mode that autoschedules nodes per pod demand.

PSC Private Service Connect; private, VPC-scoped service endpoints.

SLO Service Level Objective; monthly uptime target.

1 Problem and Notation

Choose a cloud $c \in \{\text{GCP, AWS, Azure}\}$ to host Airflow with minimal ops load and maximal integration. Let

$$U(c) = \alpha\, I(c) - \beta\, O(c) - \gamma\, R(c) + \delta\, S(c),$$

where I=native integrations, O=ops overhead, R=risk from upgrades/security, S=scalability.

2 Theoretical Frame and Rivals

GCP: Cloud Composer is a fully managed Airflow with first-party hooks to BigQuery, Dataflow, Dataproc, GCS, Pub/Sub, Secret Manager, Vertex AI, and Monitoring. [1, 7, 8, **?**]
AWS: Amazon MWAA is managed Airflow integrated with S3, EMR, Glue, Redshift, etc. [10, 11]
Azure: Managed Airflow via Azure Data Factory's Airflow runtime, integrated with Azure services. [12, 13]

3 Mechanism

Composer 2/3 architecture: Airflow webserver, scheduler, and workers run on GKE Autopilot; the metadata DB is Cloud SQL for PostgreSQL; connectivity uses Cloud SQL Proxy or PSC. This yields elastic compute, managed Postgres, and private networking. [2, 3]

4 Cross-Disciplinary Mapping

Distributed systems → autoscaling workers on GKE Autopilot. DB systems → managed Cloud SQL reduces stateful ops toil. SecOps → IAM, Secret Manager, private endpoints, Monitoring. [?, 8]

5 Thesis–Antithesis–Synthesis

Thesis: GCP maximizes I and S while minimizing O via Composer+GKE Autopilot and Cloud SQL. [1, 2]
Antithesis: AWS MWAA or Azure Managed Airflow provide similar primitives if your data gravity is there. [10, 12]
Synthesis: Prefer Composer when workloads lean on BigQuery, Dataflow, Dataproc, GCS, Pub/Sub, or you want a 99.5% SLO without cluster management. Use the provider native service where your data and teams already live. [1, 6]

6 Pros/Cons Summary for Composer (GCP)

Advantages

- Fully managed Airflow with version matrix and automated releases. [4, ?]
- Autoscaling on GKE Autopilot; environment sizing controls throughput. [9]
- Cloud SQL PostgreSQL metadata DB, private access via Proxy/PSC. [2]
- First-party integrations (BigQuery, Dataflow, Dataproc, Pub/Sub, Secret Manager, Vertex AI) and Cloud Monitoring. [7, ?, 8]
- Published 99.5% SLO. [6]

Disadvantages

- Environment fee and baseline cost; pricing is per-hour plus resources. [5]
- Less control than self-managed clusters; Airflow image and upgrades follow Composer constraints. [4]
- Cross-cloud egress charges if orchestrating non-GCP data planes.

7 Contrast With AWS MWAA and Azure Managed Airflow

- **AWS MWAA**: Strong if data sits in S3/EMR/Glue/Redshift; pricing is environment-time plus autoscaling. [11, 10]
- **Azure Managed Airflow**: Good fit with ADF pipelines and Azure services; integrates with Azure identity. [12, 13]

8 Operational Steps

1. Choose Composer 2/3 and region near data. [1]
2. Enable private environment, Secret Manager, and VPC egress controls. [?, 2]
3. Pin Airflow providers and test with Composer's supported versions. [4]
4. Size environment and verify autoscaling under peak DAG loads. [9]
5. Export metrics to Cloud Monitoring dashboards and set SLO alerts. [8, 6]

9 Limits, Uncertainty, Predictions

Costs vary with worker minutes, storage, and DB size; model with the pricing guide. Network egress dominates for multi-cloud orchestration. Expect Composer to reduce MTTR and ops toil compared with DIY GKE or VMs.

> "Choose where your data lives; orchestration follows."

Boxed Recap

$\boxed{\text{If your data plane is on GCP, pick Composer: } U(\text{GCP})\uparrow \text{ via } I\uparrow, S\uparrow, O\downarrow \text{ and a 99.5\% SLO.}}$

References

[1] Cloud Composer overview, Google Cloud Docs, accessed Nov. 3, 2025. Link. [?, ?]

[2] Environment architecture (Composer 2), Google Cloud Docs, accessed Nov. 3, 2025. [?]

[3] Arquitectura del entorno, Google Cloud Docs, accessed Nov. 3, 2025. [?]

[4] Composer versioning and supported Airflow versions, Google Cloud Docs, accessed Nov. 3, 2025. [?, ?]

[5] Cloud Composer pricing, Google Cloud Docs, accessed Nov. 3, 2025. [?]

[6] Cloud Composer SLO 99.5%, Google Cloud Docs, May 9, 2022. [?]

[7] Running Dataflow/Dataproc/BigQuery and multi-cloud DAGs from Composer, Google Cloud Docs, accessed Nov. 3, 2025. [?]

[8] Cloud Monitoring integration for Composer, Google Cloud Docs, accessed Nov. 3, 2025. [?]

[9] Scale environments in Composer 2, Google Cloud Docs, accessed Nov. 3, 2025. [?]

[10] Amazon MWAA documentation, accessed Nov. 3, 2025. [?]

[11] Amazon MWAA pricing, accessed Nov. 3, 2025. [?]

[12] Managed Airflow in Azure Data Factory, Microsoft TechCommunity, accessed Nov. 3, 2025. [?]

[13] Airflow Azure provider docs (ADF connections), accessed Nov. 3, 2025. [?]