

Google Cloud Composer - Complete Architecture and DAG Execution Flow

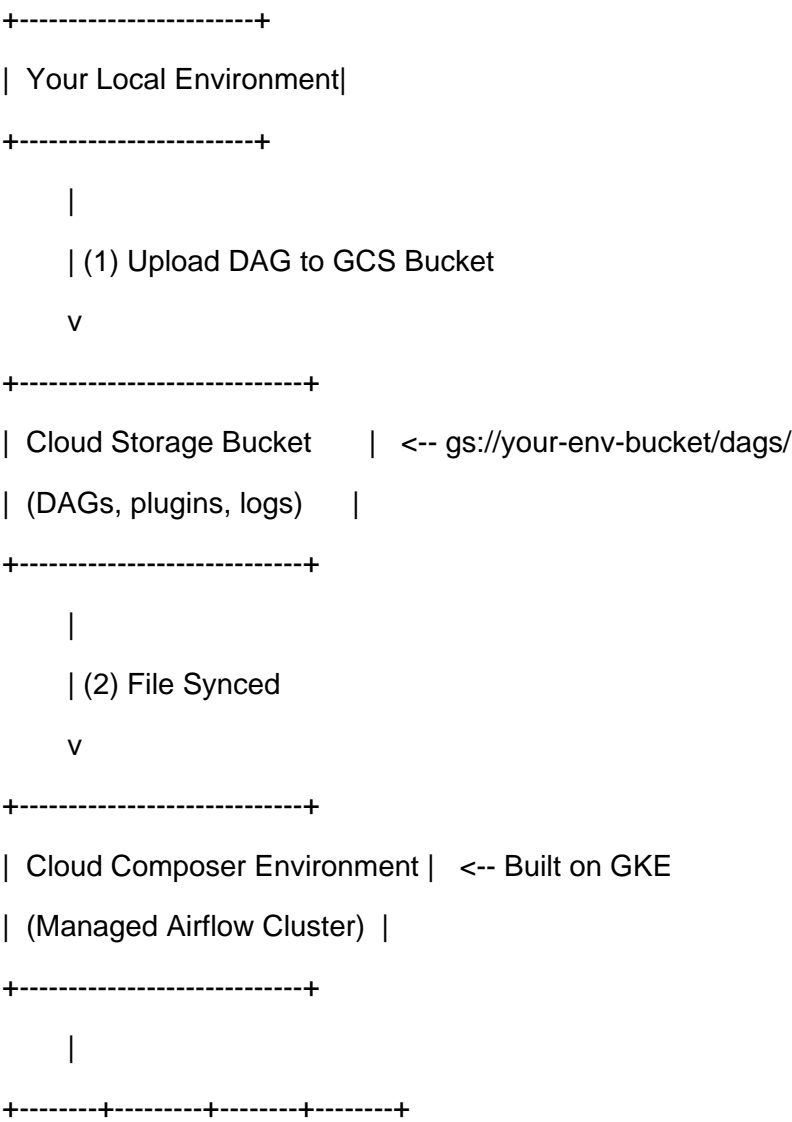
Google Cloud Composer is a fully managed Apache Airflow service. It abstracts away the infrastructure setup, scaling, upgrades, and monitoring-so you can focus on writing DAGs to orchestrate workflows.

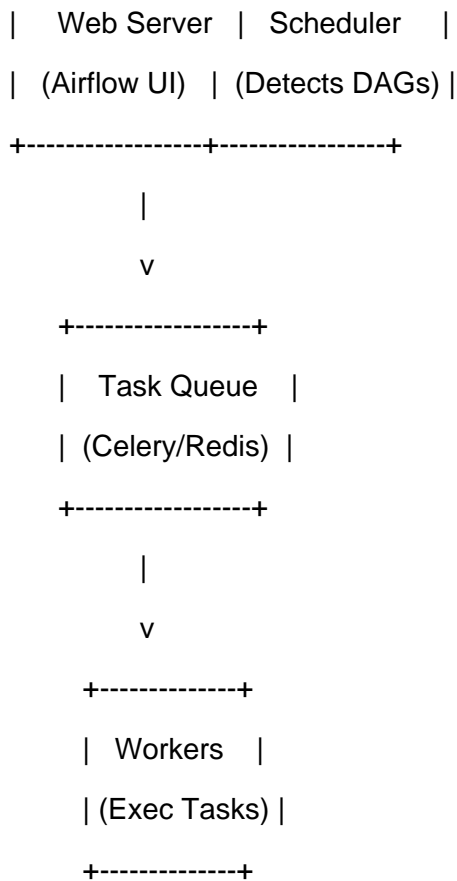
Overview:

Google Cloud Composer is built on top of:

- Apache Airflow
- Google Kubernetes Engine (GKE)
- Cloud SQL (for metadata DB)
- Cloud Storage (for DAGs, logs, plugins)
- Cloud Logging and Monitoring

Architecture:





DAG Execution Lifecycle:

1. DAG Authoring:

- You write a DAG in Python using Airflow syntax.
- DAG defines tasks, dependencies, scheduling intervals, and parameters.

2. Upload to Composer Bucket:

- You place the DAG file in the GCS bucket: `gs://[composer-env-name]/dags/`

3. Airflow Scheduler Detects DAG:

- Scheduler scans the `/dags` folder and parses files.
- Metadata is stored in Cloud SQL.

4. Scheduler Queues Tasks:

- Scheduler checks the schedule, creates DAG runs, and queues tasks.
- Tasks are pushed into Celery queue (or handled by `KubernetesExecutor`).

5. Task Execution by Workers:

- Airflow workers (GKE pods) pull tasks and execute them.
- Tasks interact with GCP services using service accounts.

6. Logging and Monitoring:

- Logs go to Cloud Logging and GCS.
- Metadata updates in Cloud SQL.
- Logs can be viewed in the Airflow UI or Stackdriver.

7. View Status in Web UI:

- Airflow Web Server provides UI for DAG/task monitoring, logs, and actions.

Core Components Recap:

Component	Purpose
----- -----	
GKE	Runs Airflow components (scheduler, webserver, workers)
Cloud SQL	Stores Airflow metadata
Cloud Storage	Stores DAGs, logs, and plugins
Cloud Logging	Centralized logging
Airflow Scheduler	Detects DAGs and schedules tasks
Airflow Webserver	UI to view DAGs and logs
Airflow Workers	Executes tasks
IAM	Controls access to resources

Executors:

- CeleryExecutor (default in Composer 1.x)
- KubernetesExecutor (recommended in Composer 2.x)

Security:

- Uses IAM for role-based access
- Supports VPC-SC and Private IP environments
- Can integrate with Secret Manager

Best Practices:

- Use separate Composer environments for dev/test/prod.
- Pin Python and Airflow versions.
- Keep DAG logic modular and files lightweight.
- Use environment variables or Secrets Manager for sensitive config.
- Monitor with Cloud Monitoring and alerting.
- Avoid writing data-heavy logic inside the DAG file.

Helpful References:

- Composer Overview: <https://cloud.google.com/composer/docs/overview>
- Architecture Docs: <https://cloud.google.com/composer/docs/concepts/how-composer-works>
- YouTube - GCP: <https://www.youtube.com/watch?v=xdK0jLJt9M>
- YouTube - Airflow DAG Execution: https://www.youtube.com/watch?v=cHATHSB_450