



Apache Airflow, AI Agents and I



Managed Service for Apache Airflow, Google Team



Piotr Wieczorek
Senior Product Manager

piotrwieczorek@google.com

- 15+ years of experience across Tech, Finance and Management Consulting



Rafal Biegacz
Senior Eng Manager

rafalbiegacz@google.com

- Working with Airflow for 7+ years
- Member of Airflow Summit Organizing Team



Michal Modras
Senior Eng Manager

michalmodras@google.com

- Working with Airflow for 5+ years
- Member of Airflow Summit Organizing Team



Agenda

Navigating the
Agentic Era

1. **Intro & New Features** [4 min]
2. **Managed Airflow Agents** [8 min]
3. **Orchestration Pipelines & MLOps** [8 min]
4. **Q&A** [5 min]

Managed Service for **Apache Airflow**

(formerly Cloud Composer)



Simple Deployments

Spin up environments seamlessly via UI, CLI, or Terraform.

Minimized Operational Burden

Let Google manage the underlying infrastructure, lowering TCO.

Native Synergy

Out-of-the-box integrations with the GCP ecosystem e.g., Knowledge Catalog.

Easy



Automated Efficiency

Lower costs natively with dynamic environment autoscaling.

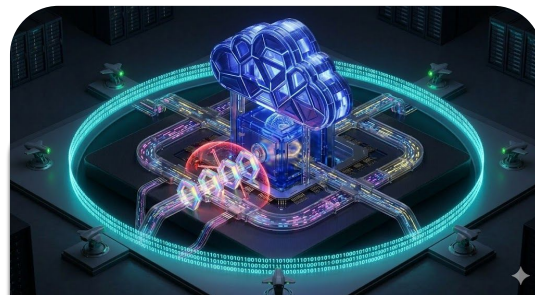
Seamless Upgrades

Future-proof your stack with in-place environment upgrades.

AI-Assisted Ops

Built-in integration with  **Gemini** Cloud Assist, robust logging and monitoring.

Smart



Enterprise-Grade Protection

Robust security (e.g. CMEK, VPC SC) and compliance (e.g. HIPAA, PCI DSS)

Native IAM

Eliminate the attack surface of 3rd-party SaaS by managing access natively.

Always On

Backed by High Availability mode, Disaster Recovery, and 24/7 Support.

Secure

Our latest (3rd) generation adopted by majority of our customer base



Real Impact Eliminating the Toil



Powered by Gemini



*Google Cloud Composer is great because it **makes it easy to monitor the jobs** that have to be carried out every day. **We don't have to think about it**, and if there's an issue, our data engineers receive a notification and can manage it."*

Benjamin Karoubi

Head of Data, relevanC Retail Tech

*The speed of innovation at Google Cloud has been **excellent**. There are continued updates to (...) Cloud Composer (...), enabling us to accelerate data-driven decision-making across our organization, deliver faster, more personalized customer solutions, and **harness the transformative potential of gen AI**."*

Stephen Lee

Head of Enterprise Data Engineering, TELUS

Recently Launched Features



AI AGENT

Agentic Troubleshooting

Diagnose and resolve DAG failures automatically with AI.



AI AGENT

Agentic Version Upgrade

Smooth transition to newer Airflow version with AI validation.



AI AGENT

Agentic Authoring

Generate production-grade DAGs using natural language.



AI AGENT

Agentic Optimization

Continuous adjustments for cost and performance efficiency.



AI AGENT

"Ask Airflow" Chat

Get instant context-aware answers inside your console.



PLATFORM

(Available to select customers as Preview)

Environment Hibernation

Automatically pause idle environments to optimize cloud cost.



PLATFORM

Apache Airflow Support 3.3

Support for the the latest version of the open-source release.



PLATFORM

MCP Server Orchestration

Seamlessly orchestrate model context protocol servers natively.



PLATFORM

Enhanced Local CLI

Streamlined local experience for faster development cycles.



Policy Tags Support

and many more enhancements...

Meet Our Managed Airflow Agents



Airflow 3 Readiness



Google Cloud Console
Integration

Data Cloud Extension
Integration

Data Engineering Agents

Troubleshooting Agent

Instantly diagnose DAG failures and infrastructure bottlenecks.

3P IDE & Cloud Console

Code Authoring Agent

Natural language to Python DAG generation with best-practice templates.

3P IDE

Optimization Agent

Proactive resource scaling recommendations based on execution telemetry.

3P IDE & Cloud Console

Upgrade Agent

Automated compatibility checks and syntax migration for Airflow versions.

3P IDE

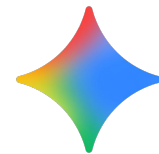
"Ask Airflow" Guider Agent

Expert guidance on architectures, dependency mapping, and secure setups.

3P IDE & Cloud Console

Managed Airflow MCP Server

Authoring



Prompt-to-Code Generation

Generates production-grade Airflow DAGs, Orchestration YAMLs, and compute scripts directly from natural language prompts.



Automated L1/L2 Validation

Tests syntax, verifies variable bindings, and checks DAG compliance locally before deploying to GCP.



IDE Native

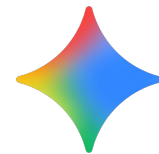
Seamlessly integrated into VS Code via Google Cloud Data Agent Kit.

Agentic DAG authoring
in Visual Studio Code

Google

Troubleshooting Agent

No More Digging Through Worker Logs



01 Automated Log Analysis

Reads dense worker execution logs automatically when a run fails.

02 Root Cause Disambiguation

Accurately differentiates infrastructure quota issues from code bugs.

03 Resolution

Generates actionable summaries and suggests code or compute fixes.

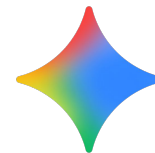
DAAG troubleshooting
In Cloud Console

Google



Version Upgrade Agent

Upgrades Simplified



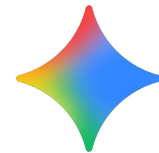
Zero Manual Deprecation
Audits

Auto Operator Refactoring

1-Click Version Bump Validation



Ask Airflow: Smart Console Assistant



Environment Optimization

Analyzes run history to recommend optimal worker scaling, memory allocations, and scheduler concurrency.



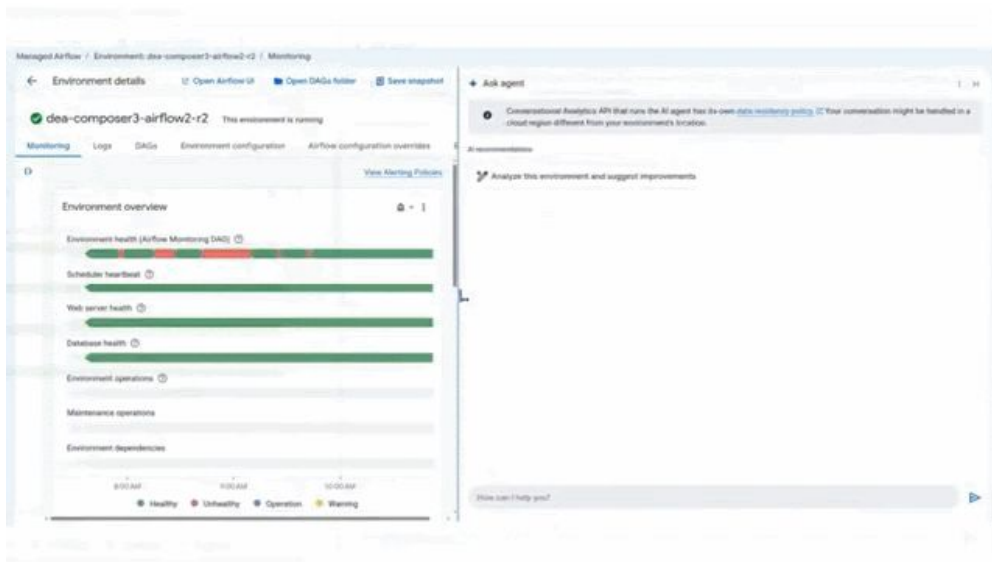
Configuration Advisor

Audits Airflow configurations against GCP security, network, and performance best practices.

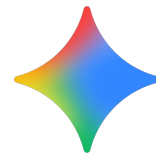


Conversational Console Chat

Embedded Assistant directly in the GCP Console for instant operational guidance and automated tuning.



Contextual Intelligence via MCP Server



AI Agents

Data Engineering Agents, IDE extensions, and Console assistants requesting context.

MCP SERVER

Model Context Protocol

Standardized, secure protocol exposing live Airflow DAG state, GCP connections, IAM permissions, and schema definitions.



Managed Airflow

Airflow Metadata DB, Cloud Logging, BigQuery Datasets, and Dataproc Clusters.

```

    "storageGb": 1
  },
  "triggerer": {
    "count": 1,
    "cpu": 0.5,
    "memoryGb": 1
  },
  "webServer": {
    "cpu": 1,
    "memoryGb": 4,
    "storageGb": 1
  },
  "worker": {
    "cpu": 0.5,
    "maxCount": 3,
    "memoryGb": 2,
    "minCount": 1,
    "storageGb": 10
  }
},
"createTime": "2026-03-25T15:17:14.827398Z",
"name": "projects/mpapierowski-project/locations/us-central1/environments/mpapierowski-staging-airflow2-env",
"state": "RUNNING",
"storageConfig": {
  "bucket": "us-central1-mpapierowski-st-9ee0d146-bucket"
},
"updateTime": "2026-03-25T15:38:54.062872Z",
"uuid": "1412fd3b-d361-4178-b56b-c78fe8c63534"
}

```

+ Your Airflow 2 environment (mpapierowski-staging-airflow2-env) is a **Cloud Composer 3 (Private IP)** environment. Here's a breakdown of its networking configuration:

1. Internal Networking (Private IP)

The environment is configured with `enablePrivateEnvironment: true` and `networkingType: PRIVATE`. This means:

- * **GKE Cluster Isolation:** The underlying GKE Autopilot nodes (which run your DAGs) have **internal IP addresses only**. They are located in a Google-managed tenant project.
- * **VPC Connection:** The tenant project connects to your VPC (mpapierowski-project) using **Private Service Connect (PSC)**. This allows the nodes to access resources in your VPC (like databases or APIs) privately.

2. Internet Access (Outbound)

The "Network is unreachable" error you're seeing in your DAG confirms that **the nodes currently have no path to the public internet**.

- * Because the nodes lack public IPs, they cannot reach external APIs like `api.currencylayer.com` directly.
- * **Missing Requirement:** To enable internet access for these nodes, you need to configure a **Cloud NAT** gateway on the VPC network and subnet that this environment is using.

3. Airflow UI Access (Inbound)

Despite being a "Private IP" environment for its internal components, the Airflow UI itself has a public endpoint:

- * **Web Server Access**

⌵ Thinking... (esc to cancel, 4s)

Tip: List your saved chat checkpoints with `/resume list`

Shift+Tab to accept edits

2 context files · 5 MCP servers · 9 skills

> █ Type your message or @path/to/file

workspace (/directory)

sandbox
no sandbox

/model
gemini-3-flash-preview

Understand environment configuration

```

    "webServer": {
      "cpu": 1,
      "memoryGb": 4,
      "storageGb": 1
    },
    "worker": {
      "cpu": 0.5,
      "maxCount": 3,
      "memoryGb": 2,
      "minCount": 1,
      "storageGb": 10
    }
  },
  "createTime": "2026-03-25T15:17:14.827398Z",
  "name": "projects/mpapierowski-project/locations/us-central1/environments/mpapierowski-staging-airflow2-env",
  "state": "RUNNING",
  "storageConfig": {
    "bucket": "us-central1-mpapierowski-st-9ee0d146-bucket"
  },
  "updateTime": "2026-03-25T15:38:54.062872Z",
  "uuid": "1412fd3b-d361-4178-b56b-c78fe8c63634"
}
}
}

```

• You have two Cloud Composer environments in the us-central1 location:

1. mpapierowski-staging-airflow3-env
 - * State: RUNNING
 - * Airflow Version: 3.1.7 (Composer 3)
 - * Airflow UI: Link (<https://1e68c1f3fe2f4b66abc5feeb6d118e1-dot-us-central1.composer-staging.googleusercontent.com>)
2. mpapierowski-staging-airflow2-env
 - * State: RUNNING
 - * Airflow Version: 2.10.5 (Composer 3)
 - * Airflow UI: Link (<https://1412fd3bd3614178b56bc78fe8c63634-dot-us-central1.composer-staging.googleusercontent.com>)

> What's the security posture of the Airflow 2 environment?

• The security posture of your Airflow 2 environment (mpapierowski-staging-airflow2-env) has several areas that could be improved:

- * **Public Web Server Access:** The Airflow web server is currently accessible from **any IP address** (0.0.0.0/0 and ::0/0). This is a high security risk; access should be restricted to specific CIDR ranges.
- * **Default Service Account:** It is using the default **Compute Engine service account** (741471203171-compute@developer.gserviceaccount.com). Best practice is to use a dedicated, user-managed service account with the minimum permissions required.
- * **Encryption:** The environment uses **Google-managed encryption keys**. For higher security requirements, Customer-Managed Encryption Keys (CMEK) can be used.
- * **Network Isolation:** On a positive note, it is configured as a **Private IP environment**, meaning the GKE cluster and internal components are not exposed to the public internet.

? for shortcuts

Shift+Tab to accept edits

2 context files · 5 MCP servers · 9 skills

> █ Type your message or @path/to/file

workspace (/directory)

sandbox
no sandbox

/model
gemini-3-flash-preview

Check security posture

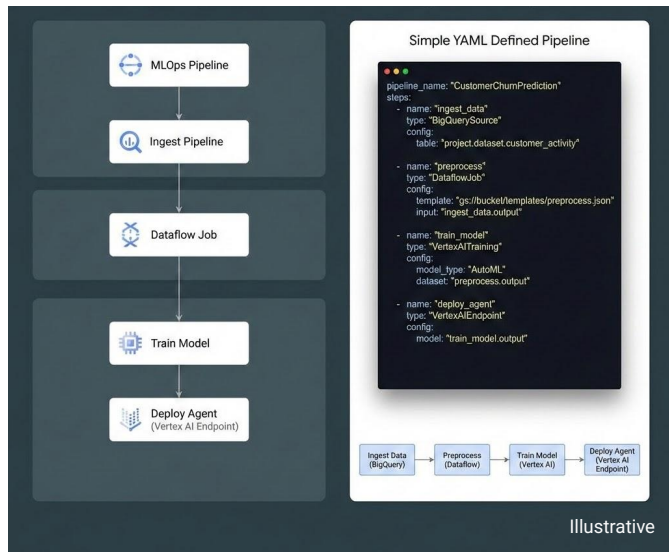
Democratizing Orchestration

You no longer need to be an
Apache Airflow expert to
harness its power!

Introducing **Orchestration Pipelines**

Introducing Orchestration Pipelines

Managed Airflow can be used with simplified Authoring & Deployment framework called Orchestration Pipelines, allowing all Data Personas to use it.



No complex Python required

Define entire pipelines in YAML (ingestion, processing)

Assisted with Data Engineering Agents

Deploy directly from your IDE

Available via Data Agent Kit

To be natively supported in Apache Airflow (AIP 85)

Why it matters: By moving to a declarative standard, we drastically lower the barrier to entry. Data Analysts and Scientists can now define, visualize, and deploy multi-step pipelines seamlessly.

Orchestration pipelines

Python Airflow code

```
1 import datetime
2 import pendulum
3 from airflow import DAG
4 from airflow.providers.google.cloud.operators.dataproc import DataprocSubmitJobOperator
5
6 with DAG(
7     dag_id="dataproc-existing-cluster-script-pipeline",
8     description="A pipeline with a Dataproc Existing cluster running the PySpark task.",
9     schedule_interval="0 5 * * *",
10    start_date=pendulum.datetime(2025, 10, 1, tz="UTC"),
11    end_date=pendulum.datetime(2026, 10, 1, tz="UTC"),
12    catchup=False,
13    tags=["dataproc_create_batch", "example"],
14    default_args={
15        "owner": "data-eng-team",
16        "retries": 0,
17    }
18 ) as dag:
19
20    run_pyspark_on_existing_cluster = DataprocSubmitJobOperator(
21        task_id="run-pyspark-on-existing-cluster",
22        project_id="example-project",
23        region="us-central1",
24        job={
25            "placement": {
26                "cluster_name": "your-cluster-name"
27            },
28            "pyspark_job": {
29                "main_python_file_uri": "gs://example-bucket/data/my_spark_job.py"
30            }
31        },
32        impersonation_chain=["your-sa@example-project.iam.gserviceaccount.com"]
33    )
34
```



100% Open-Source Solution
(Apache 2.0 License, [GitHub](#))

YAML Orchestration Pipeline

```
1 modelVersion: "1.0"
2 pipelineId: "dataproc-existing-cluster-script-pipeline"
3 description: "A pipeline with a Dataproc Existing cluster running the PySpark task."
4 runner: "airflow"
5 owner: "data-eng-team"
6 tags: ["dataproc_create_batch", "example"]
7 defaults:
8     projectId: "example-project"
9     location: "us-central1"
10    executionConfig:
11        retries: 0
12    triggers:
13        - schedule:
14            interval: "0 5 * * *"
15            startTime: "2025-10-01T00:00:00"
16            endTime: "2026-10-01T00:00:00"
17            catchup: false
18            timezone: "UTC"
19    actions:
20        - pyspark:
21            name: "run-pyspark-on-existing-cluster"
22            engine:
23                dataprocOnGce:
24                    existingCluster:
25                        clusterName: "your-cluster-name"
26                        location: "us-central1"
27                        projectId: "example-project"
28                        impersonationChain:
29                            - "your-sa@example-project.iam.gserviceaccount.com"
30            mainFilePath: "gs://example-bucket/data/my_spark_job.py"
31
```

Orchestration pipelines

YAML Orchestration pipelines

```
1 modelVersion: "1.0"
2 pipelineId: "dataproc-existing-cluster-script-pipeline"
3 description: "A pipeline with a Dataproc Existing cluster running the PySpark task."
4 runner: "airflow"
5 owner: "data-eng-team"
6 tags: ["dataproc_create_batch", "example"]
7 defaults:
8   projectId: "example-project"
9   location: "us-central1"
10  executionConfig:
11    retries: 0
12  triggers:
13    - schedule:
14      interval: "0 5 * * *"
15      startTime: "2025-10-01T00:00:00"
16      endTime: "2026-10-01T00:00:00"
17      catchup: false
18      timezone: "UTC"
19  actions:
20    - pyspark:
21      name: "run-pyspark-on-existing-cluster"
22      engine:
23        dataprocOnGce:
24          existingCluster:
25            clusterName: "your-cluster-name"
26            location: "us-central1"
27            projectId: "example-project"
28            impersonationChain:
29              - "your-sa@example-project.iam.gserviceaccount.com"
30      mainFilePath: "gs://example-bucket/data/my_spark_job.py"
31
```



100% Open-Source Solution
(Apache 2.0 License, [GitHub](#))

Visual Code Studio



Schedule trigger

Schedule interval: 0 5 * * *

Timezone: UTC

Start: 2025-10-01T00:00:00

End: 2026-10-01T00:00:00



run-pyspark-on-existing-cl...

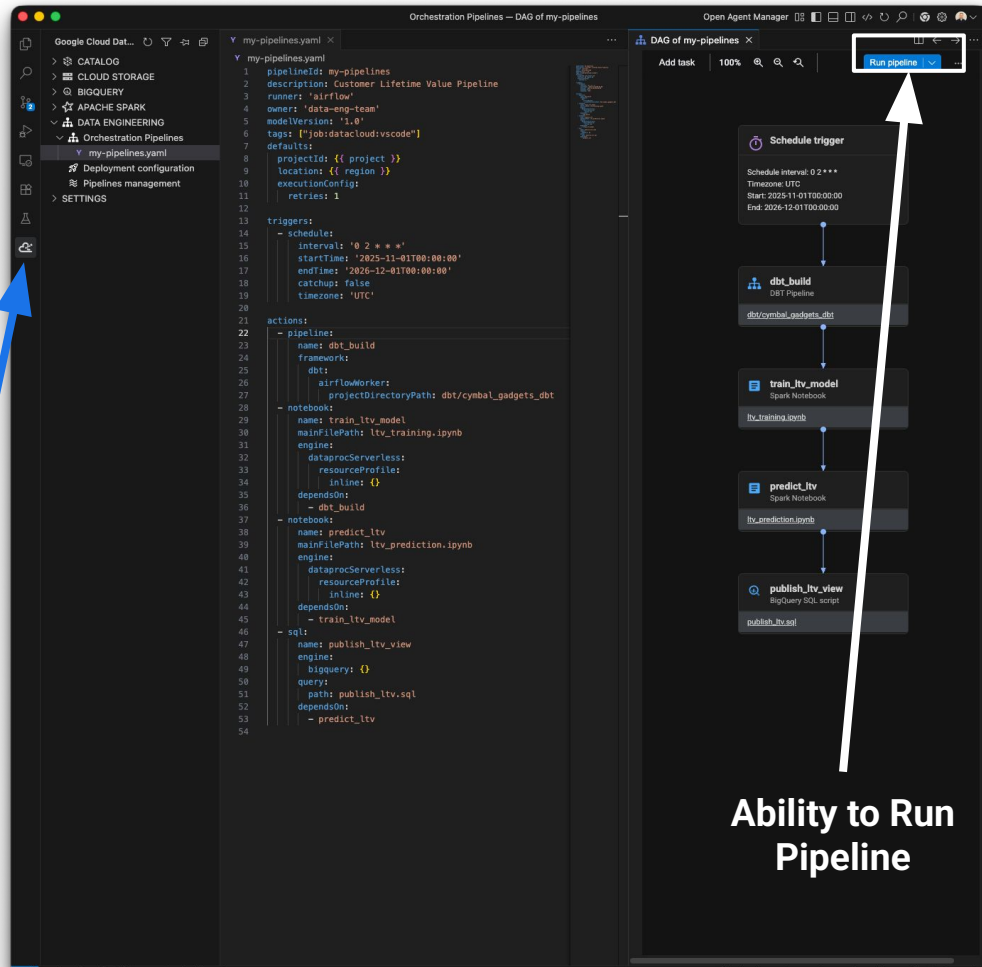
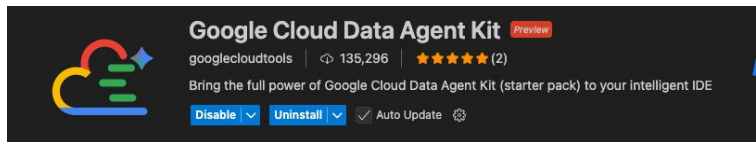
PySpark Script

[gs://example-bucket/data/my_spark_job.py](#)

Supported by Data Agent Kit extension

Orchestration Pipelines & DAG, are fully supported by our new Data Agent Kit Extension, allowing for:

- Visual Pipelines Editor
- Monitoring DAG & Dag Runs
- Pipelines Deployment
- Pipelines Configuration
- Agentic Authoring & Troubleshooting
- Unified Pipelines Definition
(Execution logic, Source code, Infrastructure configuration, Dependencies e.g. Python packages, Ownership and identification)
- Human and machine readable without ambiguity
(.yaml)



Ability to Run
Pipeline

Pipelines Monitoring with Agent Kit Extension

Pipelines List and Health

Pipelines Management - cloud-airflow-dev / us-central1 / orchestration-pipelines-pw

Filter

Status	Pipeline Name	Version	Deployment	Bundle	Owner	Actions
Healthy	pipeline-dataform-local	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	example-owner	
Healthy	pipeline-dataform-service	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	example-owner	
Healthy	pipeline-dataproc-create-batch	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	data-eng-team	
Healthy	pipeline-dataproc-create-batch-pyspark	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	data-eng-team	
Healthy	pipeline-dataproc-create-batch-resource-profile-relative	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	data-eng-team	
Healthy	pipeline-dataproc-ephemeral-inline-pyspark	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	data-eng-team	
Healthy	pipeline-dataproc-ephemeral-relative-resource-profile-pyspark	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	data-eng-team	
Healthy	pipeline-dataproc-existing-cluster-script	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	data-eng-team	
Healthy	pipeline-dataproc-existing-cluster-script-pyfiles	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	data-eng-team	
Healthy	pipeline-dbt	8278eac6e248a5b9daa15a3423fe23eaf7982359		Orchestration-Pipelines-Multi	example-owner	

Items per page: 10 1 - 10 of 18

Pipelines Management - cloud-airflow-dev / us-central1 / orchestration-pipelines-pw

Execution History - pipeline-dataproc-ephemeral-relative-resource-profile-pyspark

Task Details: scheduled__2026-06-01T12:00:00+00:00

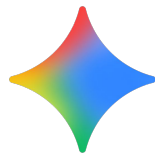
Jun 1, 2026, 2:00:00 PM - failed

Task Id: run_dataproc_ephemeral Attempts: 1 Start Time: Jun 1, 2026, 3:05:42 PM Duration (s): 29.959

Logs for DAG run: "scheduled__2026-06-01T12:00:00+00:00"

Severity Time Summary

- 2026-06-01T13:05:42.04184767Z [group::Pre task execution logs
- 2026-06-01T13:05:42.234688520Z Dependencies all met for dep_context=non-queueable deps TaskInstance: Orchestration-Pipelines-Multi_v__8278eac6e248a5b9daa15a3423fe23eaf7982359__pipeline-dataproc-ephemeral-relative-resource-profile-pyspark.run_dataproc_ephemeral__create_cluster_scheduled__2026-06-01T12:00:00+00:00 [queued]
- 2026-06-01T13:05:42.307505846Z Dependencies all met for dep_context=queueable dep TaskInstance: Orchestration-Pipelines-Multi_v__8278eac6e248a5b9daa15a3423fe23eaf7982359__pipeline-dataproc-ephemeral-relative-resource-profile-pyspark.run_dataproc_ephemeral__create_cluster_scheduled__2026-06-01T12:00:00+00:00 [queued]
- 2026-06-01T13:05:42.309123277Z Starting attempt 1 of 1
- 2026-06-01T13:05:42.374889612Z Executing <Task(BataprocCreateClusterOperator): run_dataproc_ephemeral.run_dataproc_ephemeral__create_cluster> on 2026-06-01 12:00:00+00:00
- 2026-06-01T13:05:42.411798238Z Started process 17285 to run task
- 2026-06-01T13:05:42.41630305Z Running: ["airflow", "tasks", "run", "Orchestration-Pipelines-Multi_v__8278eac6e248a5b9daa15a3423fe23eaf7982359__pipeline-dataproc-ephemeral-relative-resource-profile-pyspark", "run_dataproc_ephemeral.run_dataproc_ephemeral__create_cluster", "scheduled__2026-06-01T12:00:00+00:00", "--job-id", "2023", "--raw", "--subdir", "DAGS_FOLDER/orchestration_pipelines/Orchestration-Pipelines-Multi/pipeline-dataproc-ephemeral-relative-resource-profile-pyspark.py", "--cfe-path", "/tmp/tmp975d181v"]
- 2026-06-01T13:05:42.416836765Z Job 2033: Subtask
- run_dataproc_ephemeral.run_dataproc_ephemeral__create_cluster



Ability to leverage Agentic Troubleshooting to Diagnose Failed Task

[←](#)
[Environment details](#)
[Open Airflow UI](#)
[Open DAGs folder](#)
[Save snapshot](#)
[Load snapshot](#)
[Refresh](#)
[Delete](#)

c3-airflow3
This environment is running

[Monitoring](#)
[Logs](#)
[DAGs](#)
[Orchestration Pipelines](#)
[New](#)
[Environment configuration](#)
[Airflow configuration overrides](#)
[Environment variables](#)

6 hours ▾

Overview

DAG Statistics

Schedulers

DAG Processors

Workers

Triggers

Web Server

SQL Database

User Workloads

Environment overview

Environment health (Airflow Monitoring DAG) ⓘ

Scheduler heartbeat ⓘ

Web server health ⓘ

Database health ⓘ

Environment operations ⓘ

Maintenance operations

Environment dependencies

1:00 PM

2:00 PM

3:00 PM

4:00 PM

● Healthy

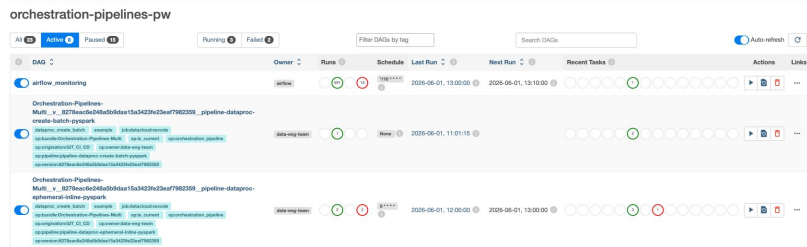
● Unhealthy

● Operation

● Warning

Pipeline ID	Bundle ID	Version	Status	State	Owner	Schedule	Description
dot-pipeline	bundle-local-rafaibegazc-weatherdbttestmain	local-Sadd33d641fc	HEALTHY	Paused	data-eng-team	0 2 ***	Run dbt pipeline for weather data.
sample.pipeline	bundle-local-rafaibegazc-orchestrationpipeline	local-df9a78f8bb61	HEALTHY	Paused	data-eng-team	0 2 ***	TODO - describe your pipeline

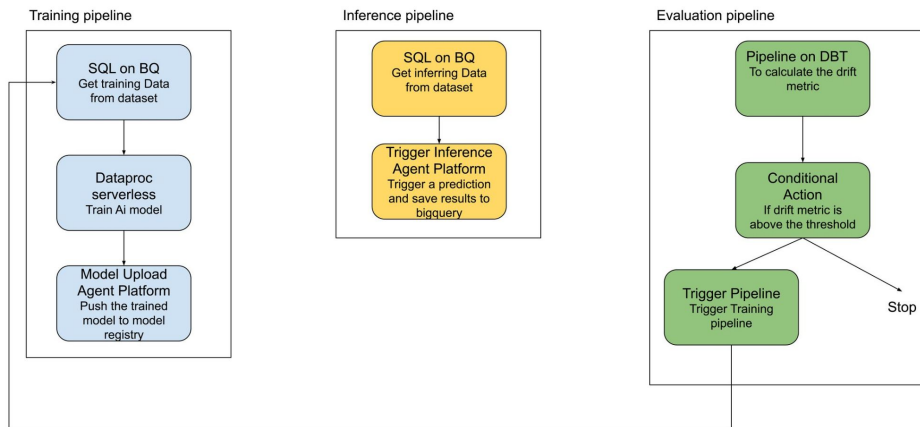
Airflow UI



Orchestration Pipelines MLOps Demo

Proactive supply chain management

Goal: End-to-end MLOps architecture that predicts the exact transit time (in days) based on warehouse location, customer location, and order characteristics.



Detailed Blogpost





Questions?

Let's connect at Google Booth!

Our LinkedIn
QR Codes

Rafał Biegacz
rafalbiegacz@google.com



Piotr Wieczorek
piotrwieczorek@google.com



Michał Modras
michalmodras@google.com

