

Airflow in Distributed Systems & Cloud Integration

by Fariz Wakan

May 17th, 2024

Airflow Production Deployment

The screenshot shows a web browser window displaying the Apache Airflow documentation page for Production Deployment. The browser's address bar shows the URL: `airflow.apache.org/docs/apache-airflow/stable/administration-and-deployment/production-deployment.html`. The page features the Apache Airflow logo and a navigation menu with links to Community, Meetups, Documentation, Use-cases, Announcements, Blog, and Ecosystem. On the left side, there is a sidebar with a search bar labeled 'Search docs' and a 'CONTENT' section listing various topics like Overview, Quick Start, and Installation of Airflow. The main content area is titled 'Production Deployment' and includes a breadcrumb trail: 'Home / Administration and Deployment / Production Deployment'. Below the title, a paragraph states: 'It is time to deploy your DAG in production. To do this, first, you need to make sure that the Airflow is itself production-ready. Let's see what precautions you need to take.' A sub-section titled 'Database backend' explains that Airflow comes with an SQLite backend by default but is not recommended for production use due to the risk of data loss. It advises users to 'configure the backend' to an external database like PostgreSQL or MySQL. On the right side, there is a vertical list of links for further reading, including 'Production Deployment', 'Database backend', 'Multi-Node Cluster', 'Logging', 'Configuration', 'Scheduler Uptime', 'Production Container Images', and 'Deployment Chart'. At the bottom right, a GitHub icon is accompanied by the text 'Suggest a change on this page'.

Version: 2.9.0 ▾

Search docs 🔍

CONTENT

- Overview
- Quick Start
- Installation of Airflow™
- Security
- Tutorials
- How-to Guides
- UI / Screenshots
- Core Concepts
- Authoring and Scheduling
- ▼ Administration and Deployment
- ▼ Production Deployment

Home / Administration and Deployment / Production Deployment

Production Deployment [🔗](#)

It is time to deploy your DAG in production. To do this, first, you need to make sure that the Airflow is itself production-ready. Let's see what precautions you need to take.

Database backend

Airflow comes with an **SQLite** backend by default. This allows the user to run Airflow without any external database. However, such a setup is meant to be used for testing purposes only; running the default setup in production can lead to data loss in multiple scenarios. If you want to run production-grade Airflow, make sure you [configure the backend](#) to be an external database such as PostgreSQL or MySQL.

You can change the backend using the following config

Production Deployment

- Database backend
- Multi-Node Cluster
- Logging
- Configuration
- Scheduler Uptime
- Production Container Images
- Deployment Chart
- Live-upgrading

Suggest a change on this page

Airflow in Google Cloud (Composer)

The screenshot shows a web browser window displaying the Google Cloud Composer overview page. The browser's address bar shows the URL `cloud.google.com/composer/docs/concepts/overview`. The Google Cloud logo is in the top left, and the navigation bar includes 'Documentation', 'Technology areas', 'Cross-product tools', a search bar, and language settings. The main navigation menu has 'Cloud Composer', 'Composer 2 Guides', 'Composer 1 Guides' (which is selected), 'Samples', 'Resources', and 'Reference'. A 'Contact Us' button is on the right. On the left side, there is a sidebar with a 'Filter' button and sections for 'Discover' (including 'Cloud Composer overview', 'Cloud Composer features', 'Cloud Composer shared responsibility model', 'Data stored in Cloud Storage', and 'Environment architecture') and 'Get started' (including 'Quickstart', 'Create environments', 'Create environments (Terraform)', and 'Enable and disable Cloud Composer service'). Below these are 'Manage DAGs' and 'Add and update DAGs'. The main content area has a blue header warning that Cloud Composer 1 is in post-maintenance mode and that Google does not release further updates to it, including new versions of Airflow, bugfixes, and security updates. It recommends planning migration to Cloud Composer 2. Below this, the breadcrumb trail is 'Cloud Composer > Documentation > Composer 1 Guides'. The main heading is 'Cloud Composer overview' with a bookmark icon and a 'Send feedback' button. The subheading is 'Cloud Composer 1 | Cloud Composer 2'. The text explains that Cloud Composer is a fully managed workflow orchestration service and is built on the popular Apache Airflow open source project. It also mentions that by using Cloud Composer instead of a local instance of Apache Airflow, users can benefit from the best of Airflow with no installation or management overhead. On the right side, there is a 'On this page' section with links to 'Apache Airflow and Cloud Composer', 'Workflows, DAGs, and tasks', 'Cloud Composer environments', 'Cloud Composer features', 'Frequently Asked Questions', and several 'What if...' questions related to Airflow and Cloud Composer.

Cloud Composer overview | x +

cloud.google.com/composer/docs/concepts/overview

New Chrome available

Google Cloud Documentation Technology areas Cross-product tools Search English Console

Cloud Composer Composer 2 Guides Composer 1 Guides Samples Resources Reference Contact Us

Filter

Discover

- Cloud Composer overview
- Cloud Composer features
- Cloud Composer shared responsibility model
- Data stored in Cloud Storage
- Environment architecture

Get started

- Quickstart
- Create environments
- Create environments (Terraform)
- Enable and disable Cloud Composer service

Manage DAGs

- Add and update DAGs

Cloud Composer 1 is in the post-maintenance mode. Google **does not release any further updates to Cloud Composer 1, including new versions of Airflow, bugfixes, and security updates.** We recommend planning [migration to Cloud Composer 2](#).

Cloud Composer > Documentation > Composer 1 Guides

Cloud Composer overview

Send feedback

Cloud Composer 1 | Cloud Composer 2

Cloud Composer is a fully managed workflow orchestration service, enabling you to create, schedule, monitor, and manage workflow pipelines that span across clouds and on-premises data centers.

Cloud Composer is built on the popular [Apache Airflow](#) open source project and operates using the Python programming language.

By using Cloud Composer instead of a local instance of Apache Airflow, you can benefit from the best of Airflow with no installation or management overhead. Cloud Composer helps you create managed Airflow environments quickly and use Airflow-native tools, such as the powerful Airflow web interface and

On this page

- Apache Airflow and Cloud Composer
- Workflows, DAGs, and tasks
- Cloud Composer environments
- Cloud Composer features
- Frequently Asked Questions
- What version of Apache Airflow does Cloud Composer use?
- Can I use native Airflow UI and CLI?
- Can I use my own database as the Airflow Metadata DB?



Thank you!

See you in the next session 💪