



**Freddy Demiane**

Software Engineer  
@Google



**Rahul Vats**

Staff Software Engineer  
@Astronomer



**Dennis Ferruzzi**

OSS Developer  
@AWS

# Hello Quality: Building CIs to Run Providers Packages System Tests



**The Problem**

01

**The Solution**

02

**Technical Deep Dive**

03

**Next Steps**

04

**Q/A**

05

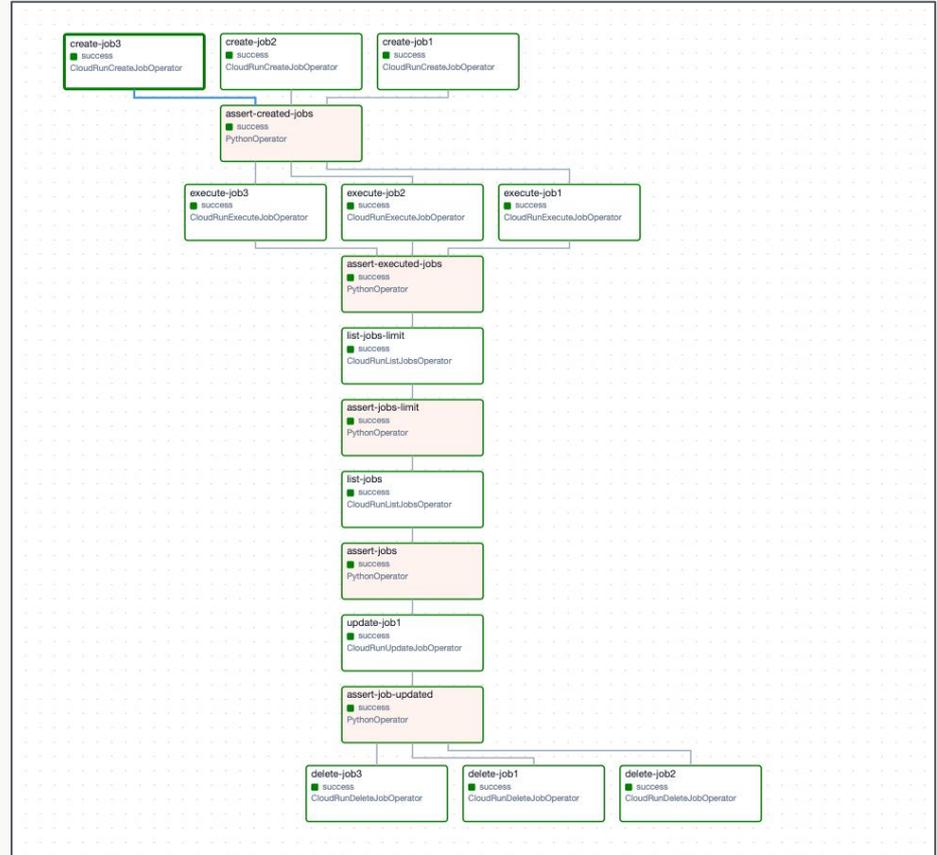


# The Problem

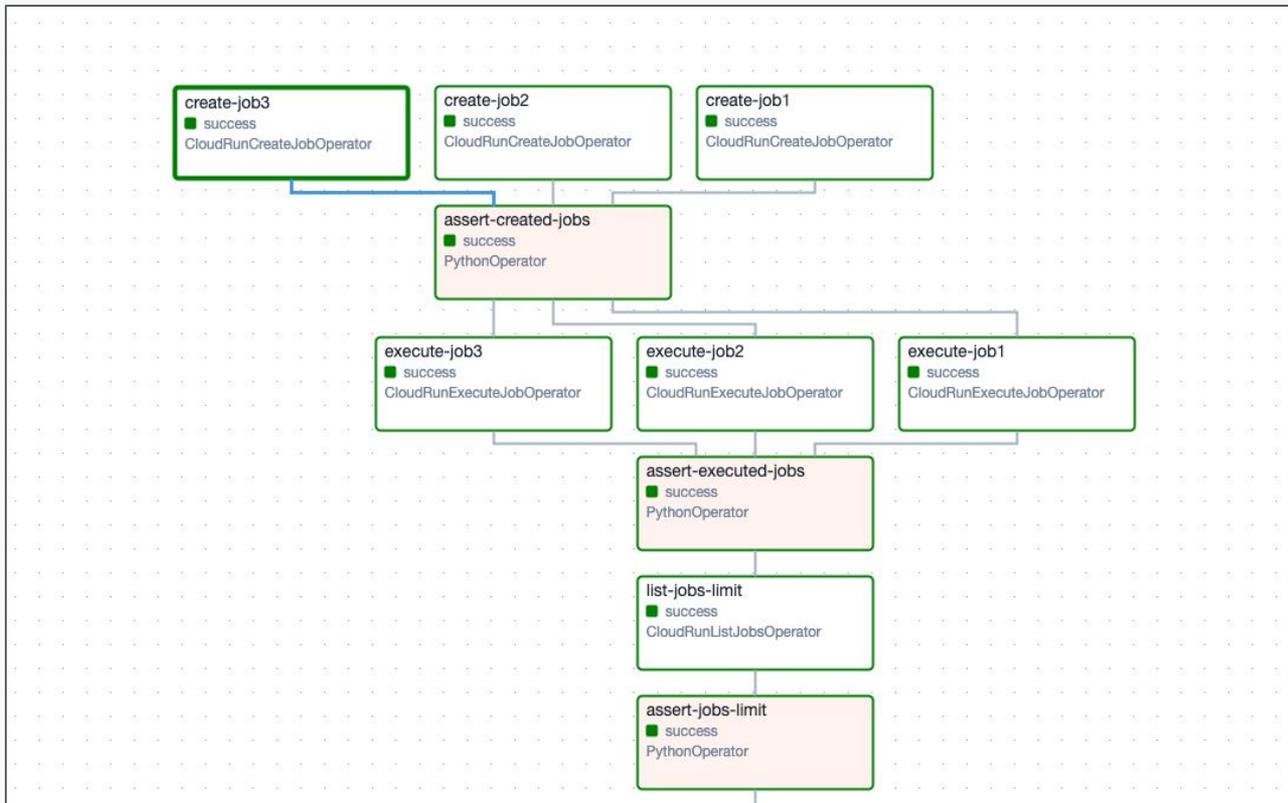


# Airflow System Tests

Airflow System Tests are dags which contain the operators and the flows we want to test



# Zoomed in



# Pain Points

- Requires an understanding of the System Test
- Requires an infrastructure - sometimes expensive
- Requires proper set up



# The Solution



# System Test Dashboards



[Community](#)

[Meetups](#)

[Documentation](#)

[Use Cases](#)

[Announcements](#)

[Blog](#)

[Ecosystem](#)

[Airflow Survey 2023](#)

## Airflow Provider System Test Dashboards

[Amazon provider package health dashboard](#) - Dashboard listing all system tests within the Amazon provider package and their current health status: last execution status (succeeded/failed, average duration, ...).

[Google provider package health dashboard](#) - Dashboard listing all system tests within the Google provider package and their current health status

[LLM Providers health dashboard](#) - Dashboard listing all system tests within the LLM provider packages and their current health status: execution status for last 7 runs(succeeded/failed, Execution date).

[Teradata Provider health dashboard](#) - Dashboard listing status of system tests for Teradata Provider and their current health status for last runs.

<https://airflow.apache.org/ecosystem/#airflow-provider-system-test-dashboards>

# Benefits

- Ensure **Reliability** and **Stability**.
- Aides in the airflow provider **release** process.
- Test **Infrastructure** setup is owned by respective providers team.
- Provider maintainers are **encouraged** to implement dashboards in the way that best suits their needs.



# Teradata Dashboard



## Apache Airflow - Teradata Provider Package Health

View the health of Teradata system tests for Apache Airflow.

This live dashboard displays the current health of Teradata system tests available in the Teradata Provider package of [Apache Airflow](#).

SystemName	Successes	Failures	Duration	Last Run Date (MM-DD-YYYY)	Last 10 runs (latest on the right)
tests.system.providers.teradata.example_teradata_call_sp	62	0	12.327	08-12-2024	['S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S']
tests.system.providers.teradata.example_azure_blob_to_teradata_transfer	62	0	231.526	08-12-2024	['S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S']
tests.system.providers.teradata.example_s3_to_teradata_transfer	17	45	39.653	08-12-2024	['F', 'F', 'F', 'F', 'F', 'F', 'F', 'F', 'F', 'F']
tests.system.providers.teradata.example_teradata_compute_cluster	0	44	12.294	08-12-2024	['F', 'F', 'F', 'F', 'F', 'F', 'F', 'F', 'F', 'F']
tests.system.providers.teradata.example_ssl_teradata	62	0	6.621	08-12-2024	['S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S']
tests.system.providers.teradata.example_teradata	62	0	12.533	08-12-2024	['S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S']
tests.system.providers.teradata.example_teradata_to_teradata_transfer	62	0	10.51	08-12-2024	['S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S', 'S']

<https://teradata.github.io/airflow/>

# Technical Details (Google)





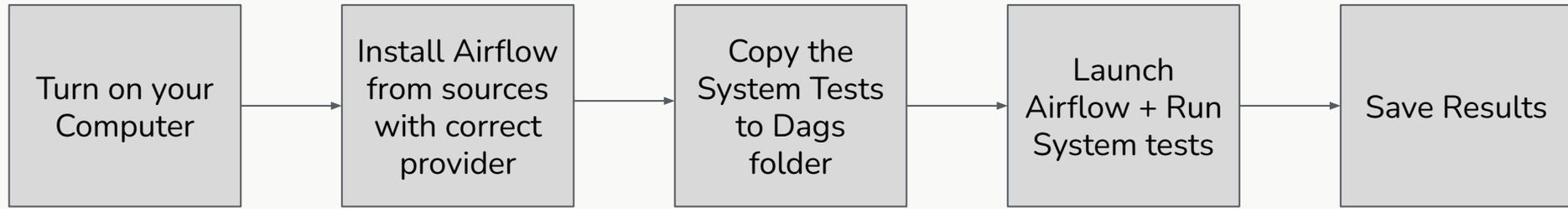
## Google Providers Package Health

This dashboard shows the status of the system tests of the head revision of the [Google Providers Package](#). The system tests are run using the head revision of [Apache Airflow](#). Note that some tests are explicitly hidden from the dashboard.

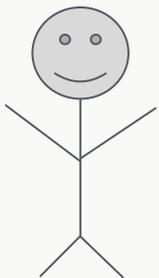
Dag ID ▲	Last Run Duration ▼	Last Successful Run	Runs Statuses (latest on the right)
bigquery_dataset	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_operations	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_operations_location	Around 18 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_tables	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_to_bigquery	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_to_mysql	Around 18 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bq_value_check_location	Around 18 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
child_dag	Around 19 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_batch	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_compute_igm	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_run	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_tasks_queue	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_tasks_tasks	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_entries	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_search_catalog	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_tag_templates	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_tags	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
dataflow_pipeline	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●

Airflow & Providers commit id: af753c6468  
 Start date: September 07 2024, 06:13:10 UTC  
 End date: September 07 2024, 06:28:35 UTC  
[logs link](#)

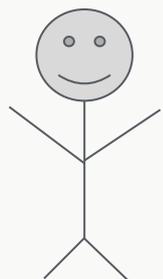
# Running tests manually



# Running tests manually

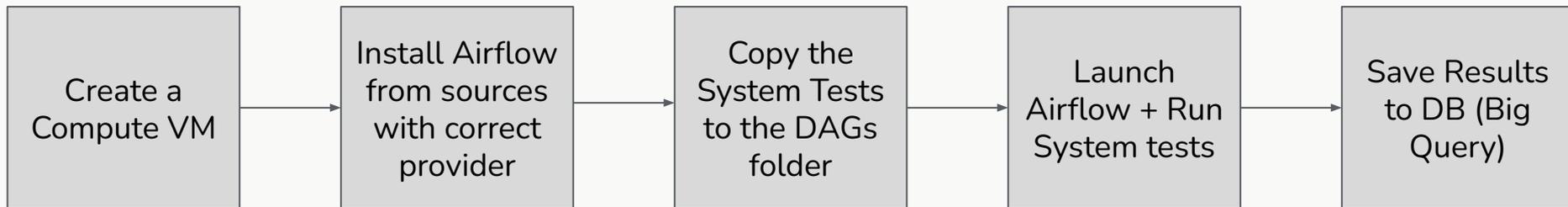


# Running tests manually



Bob

# Running tests automatically





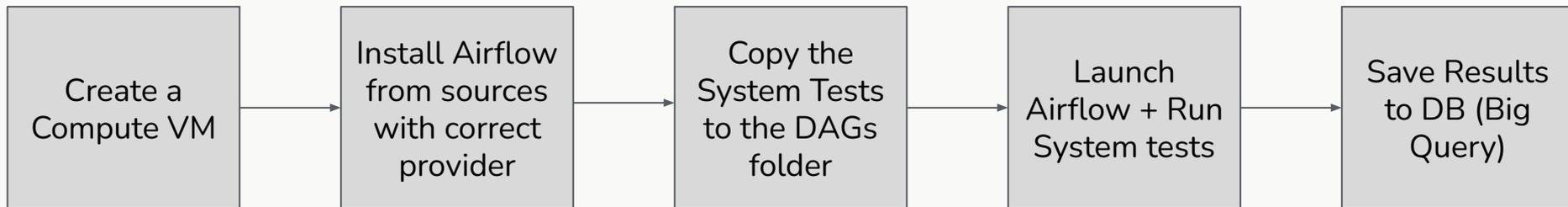
## Google Providers Package Health

This dashboard shows the status of the system tests of the head revision of the [Google Providers Package](#). The system tests are run using the head revision of [Apache Airflow](#). Note that some tests are explicitly hidden from the dashboard.

Dag ID ▲	Last Run Duration ▼	Last Successful Run	Runs Statuses (latest on the right)
bigquery_dataset	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_operations	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_operations_location	Around 18 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_tables	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_to_bigquery	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bigquery_to_mysql	Around 18 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
bq_value_check_location	Around 18 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
child_dag	Around 19 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_batch	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_compute_igm	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_run	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_tasks_queue	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
cloud_tasks_tasks	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_entries	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_search_catalog	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_tag_templates	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
datacatalog_tags	Around 17 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●
dataflow_pipeline	Around 16 minutes	2 hours ago	① ●●●●●●●●●●●●●●●●

Airflow & Providers commit id: af753c6468  
 Start date: September 07 2024, 06:13:10 UTC  
 End date: September 07 2024, 06:28:35 UTC  
[logs link](#)

# Running tests automatically



# Running tests automatically



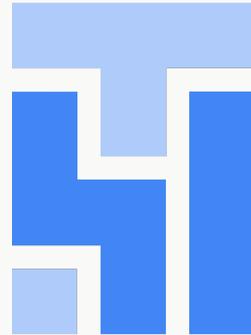
Apache  
**Airflow**



# Running tests automatically



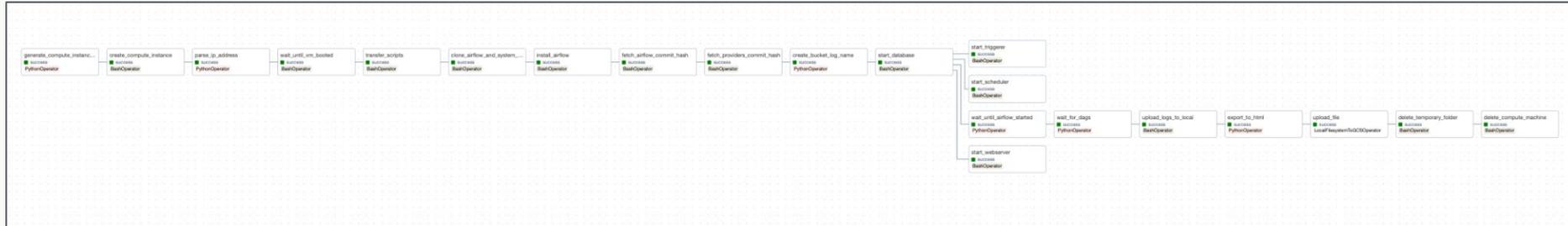
Apache  
**Airflow**



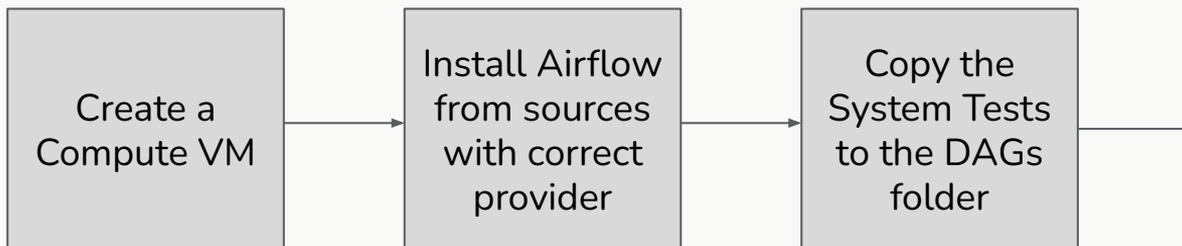
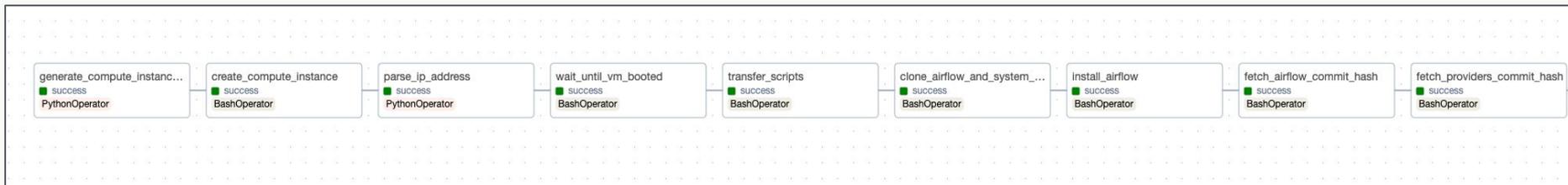
**Cloud  
Composer**



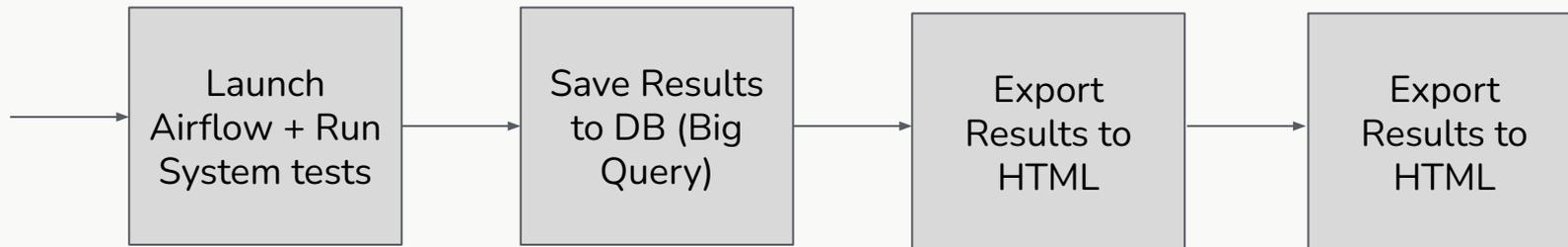
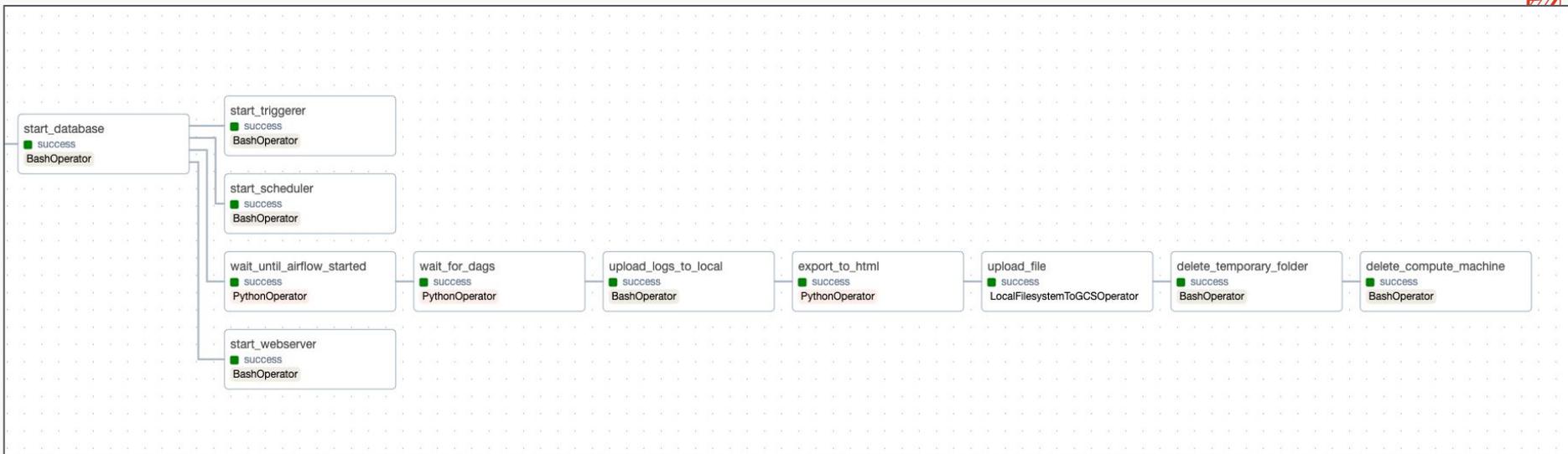
# Running tests automatically



# Zoomed in



# Zoomed in



# Technical Details (Astronomer)



## LLM Providers



pgvector



Weaviate



## Dashboard Components



# LLM Dashboard

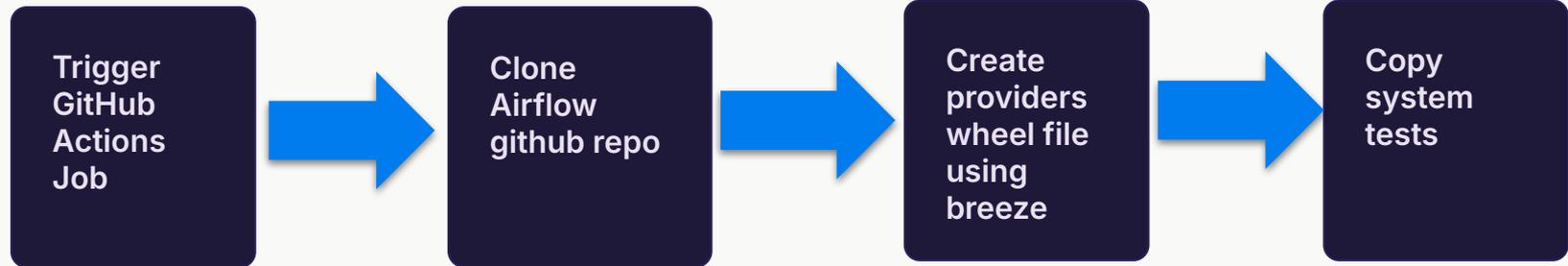


## Apache Airflow - LLM Provider Package Health

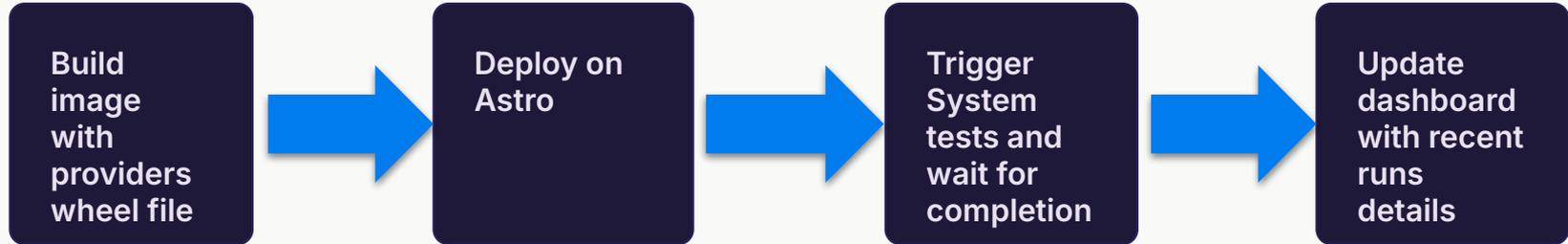
DAG ID	Success	Failed	Last 7 runs (latest on the right)
example_cohere_embedding	7	0	✓✓✓✓✓✓✓
example_openai_dag	7	0	✓✓✓✓✓✓✓
example_openai_pgvector_dag	7	0	✓✓✓✓✓✓✓
example_pgvector_dag	7	0	✓✓✓✓✓✓✓ Tue Sep 10 2024 17:19:24 GMT-0700 (Pacific Daylight Time)
example_pinecone_cohere	7	0	✓✓✓✓✓✓✓
example_pinecone_create_pod_index	7	0	✓✓✓✓✓✓✓
example_pinecone_create_serverless_index	7	0	✓✓✓✓✓✓✓
example_pinecone_openai	7	0	✓✓✓✓✓✓✓
example_weaviate_cohere	7	0	✓✓✓✓✓✓✓
example_weaviate_dag_using_hook	7	0	✓✓✓✓✓✓✓
example_weaviate_dynamic_mapping_dag	7	0	✓✓✓✓✓✓✓
example_weaviate_openai	7	0	✓✓✓✓✓✓✓
example_weaviate_using_operator	7	0	✓✓✓✓✓✓✓
example_weaviate_vectorizer_dag	7	0	✓✓✓✓✓✓✓
example_weaviate_without_vectorizer_dag	7	0	✓✓✓✓✓✓✓

Last updated: Tue Sep 10 2024 17:19:24 GMT-0700 (Pacific Daylight Time)

# Test Execution workflow

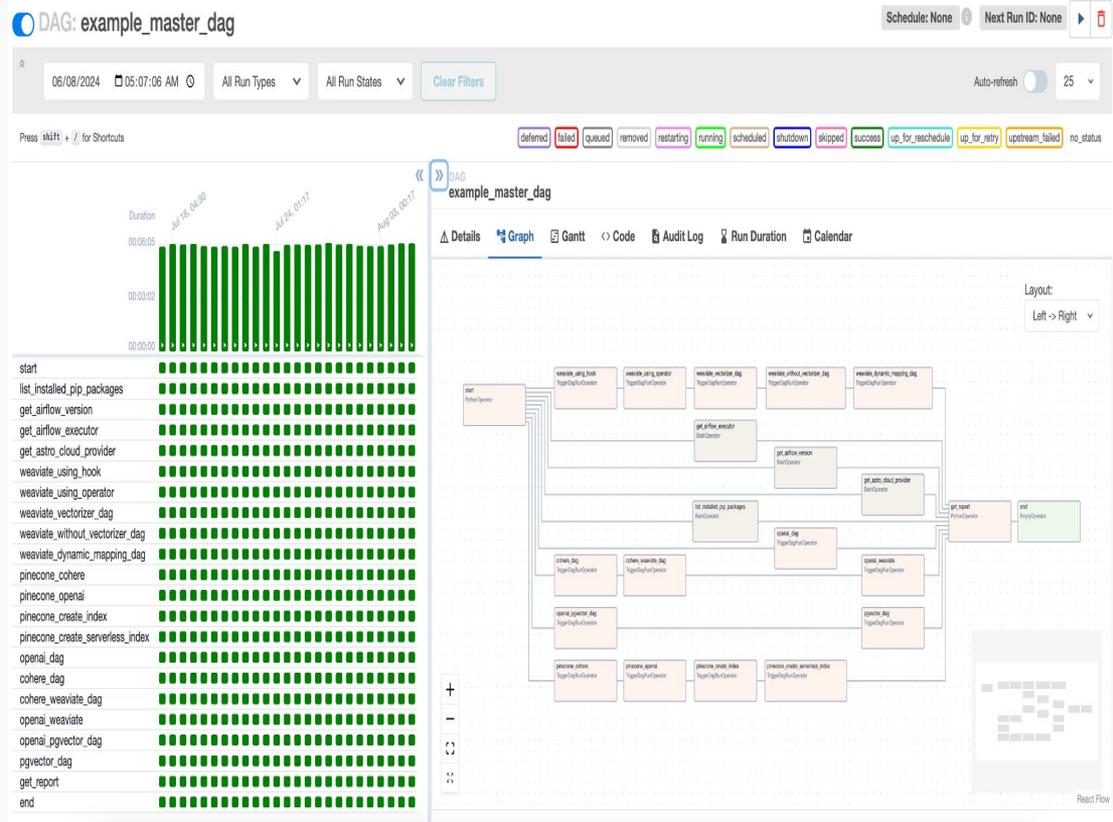


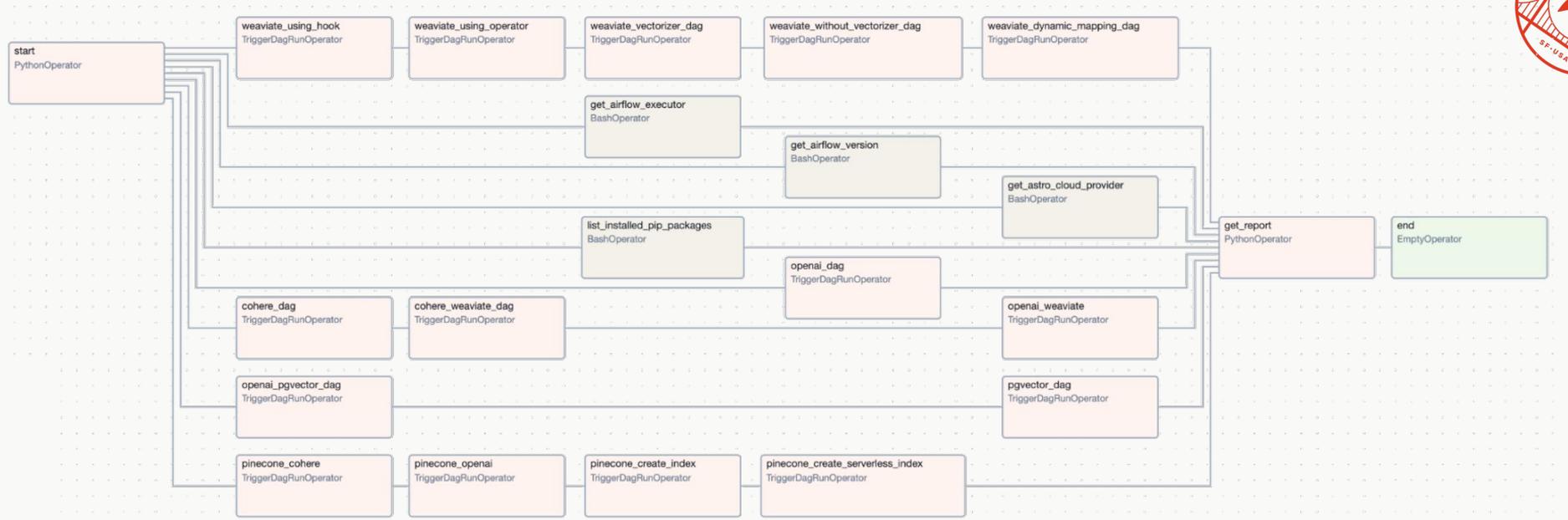
# Test Execution workflow



# Orchestrator DAG

- Trigger all of LLM providers system test DAGS
- Get the results and format it
- Share the results summary to internal Astronomer Slack channel





# Slack Alert



**Airflow** APP 5:55 AM

Results generated for:

Runtime version: 11.6.0

Python version: 3.11.9

Airflow version: 2.9.2+astro.2

Executor: CeleryExecutor

LLM-providers-version: Providers installed for Apache Airflow main sources

Cloud provider: aws

Total DAGS: 15

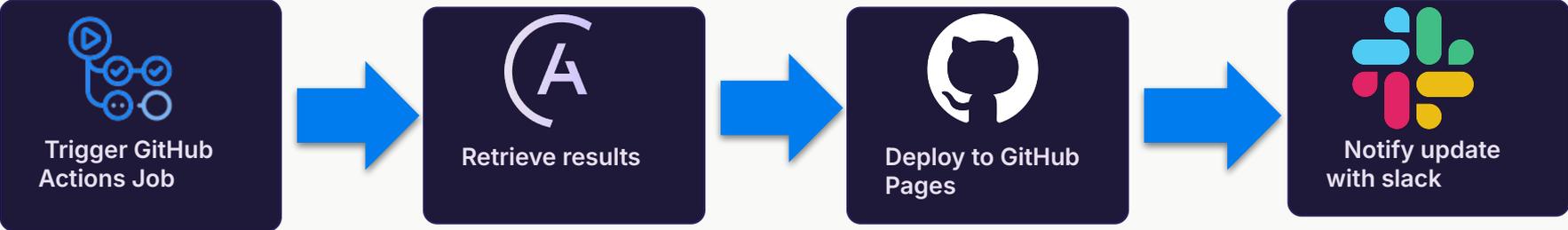
Success DAGS: 15 🍏

Failed DAGS: 0 🍎

[Link](#) to the master DAG for the above run on Astro Cloud deployment

LLM Dashboard job run success! Visit <https://astronomer.github.io/llm-dags-dashboard> for more details

# Dashboard Update

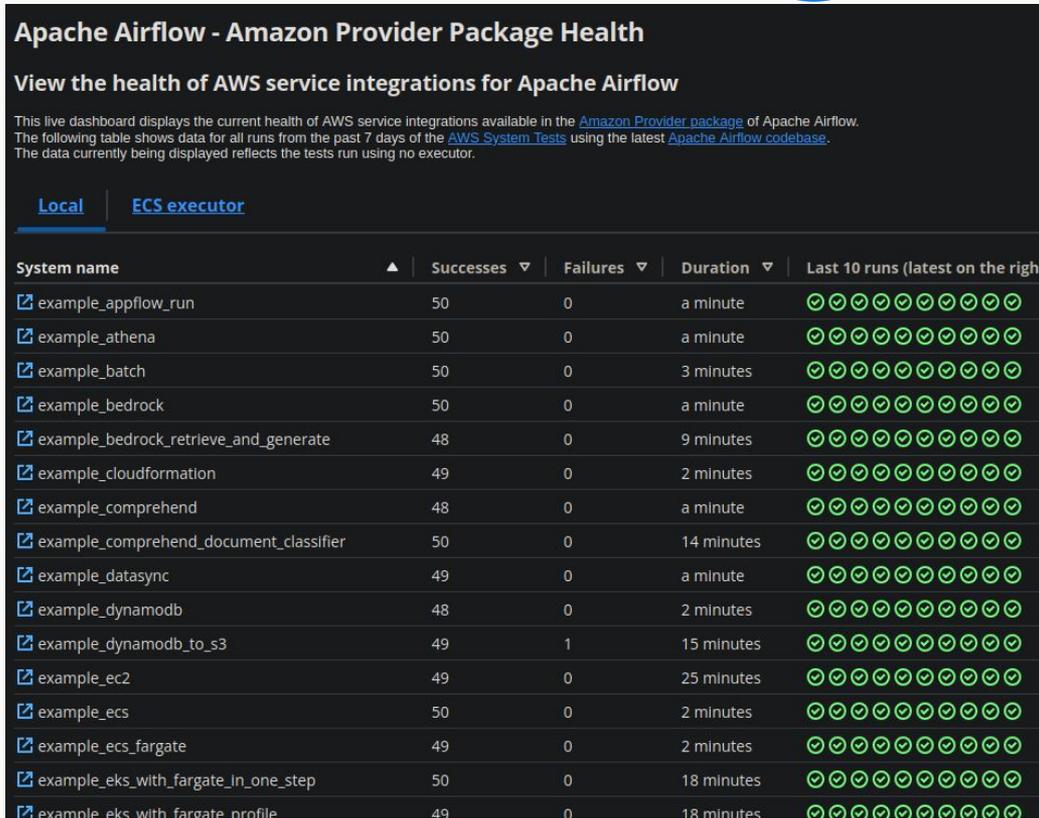


# Technical Details (AWS)



A scheduled AWS CodePipeline pulls the latest official main branch and runs an AWS CodeBuild project which launches each system test in its own ECS container. The results from each test are then aggregated into a simple HTML dashboard hosted on github.io

On a failure, or if a new test is detected, an AWS Lambda cuts a trouble ticket for the team and sends a Slack message to notify us.



The screenshot shows a dashboard titled "Apache Airflow - Amazon Provider Package Health". It includes a sub-header "View the health of AWS service integrations for Apache Airflow" and a descriptive paragraph. Below this, there are two tabs: "Local" (selected) and "ECS executor". The main content is a table with columns for "System name", "Successes", "Failures", "Duration", and "Last 10 runs (latest on the right)". The table lists 17 system tests, each with a success count, failure count, duration, and a row of 10 green checkmarks indicating successful runs.

System name	Successes	Failures	Duration	Last 10 runs (latest on the right)
example_appflow_run	50	0	a minute	10 green checkmarks
example_athena	50	0	a minute	10 green checkmarks
example_batch	50	0	3 minutes	10 green checkmarks
example_bedrock	50	0	a minute	10 green checkmarks
example_bedrock_retrieve_and_generate	48	0	9 minutes	10 green checkmarks
example_cloudformation	49	0	2 minutes	10 green checkmarks
example_comprehend	48	0	a minute	10 green checkmarks
example_comprehend_document_classifier	50	0	14 minutes	10 green checkmarks
example_datasync	49	0	a minute	10 green checkmarks
example_dynamodb	48	0	2 minutes	10 green checkmarks
example_dynamodb_to_s3	49	1	15 minutes	10 green checkmarks
example_ec2	49	0	25 minutes	10 green checkmarks
example_ecs	50	0	2 minutes	10 green checkmarks
example_ecs_fargate	49	0	2 minutes	10 green checkmarks
example_eks_with_fargate_in_one_step	50	0	18 minutes	10 green checkmarks
example_eks_with_fargate_profile	49	0	18 minutes	10 green checkmarks



## Executor Used

## Link to Source

## Test Name

## Recent Counts

## Average Run Duration

## Hover for Timestamp

[Local](#) | [ECS executor](#)

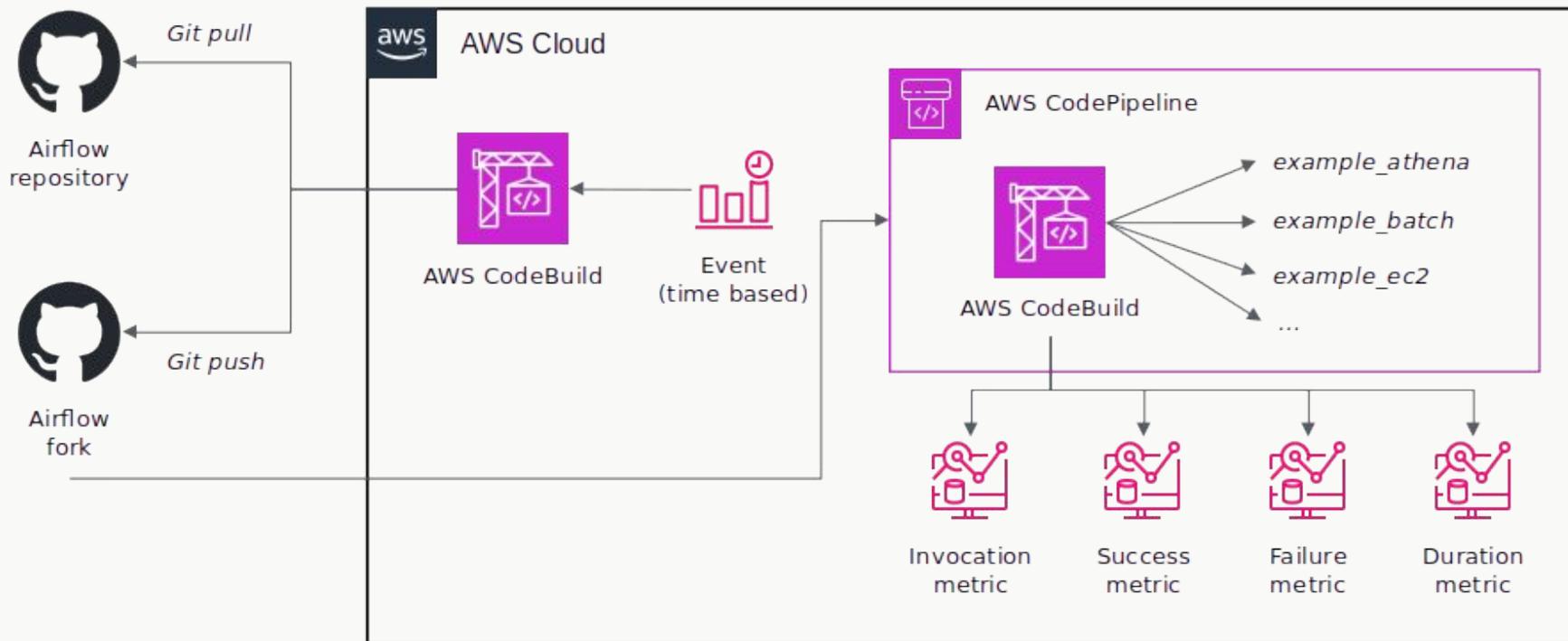
Successes ▾ | Failures ▾

Last 10 runs (latest on the right)

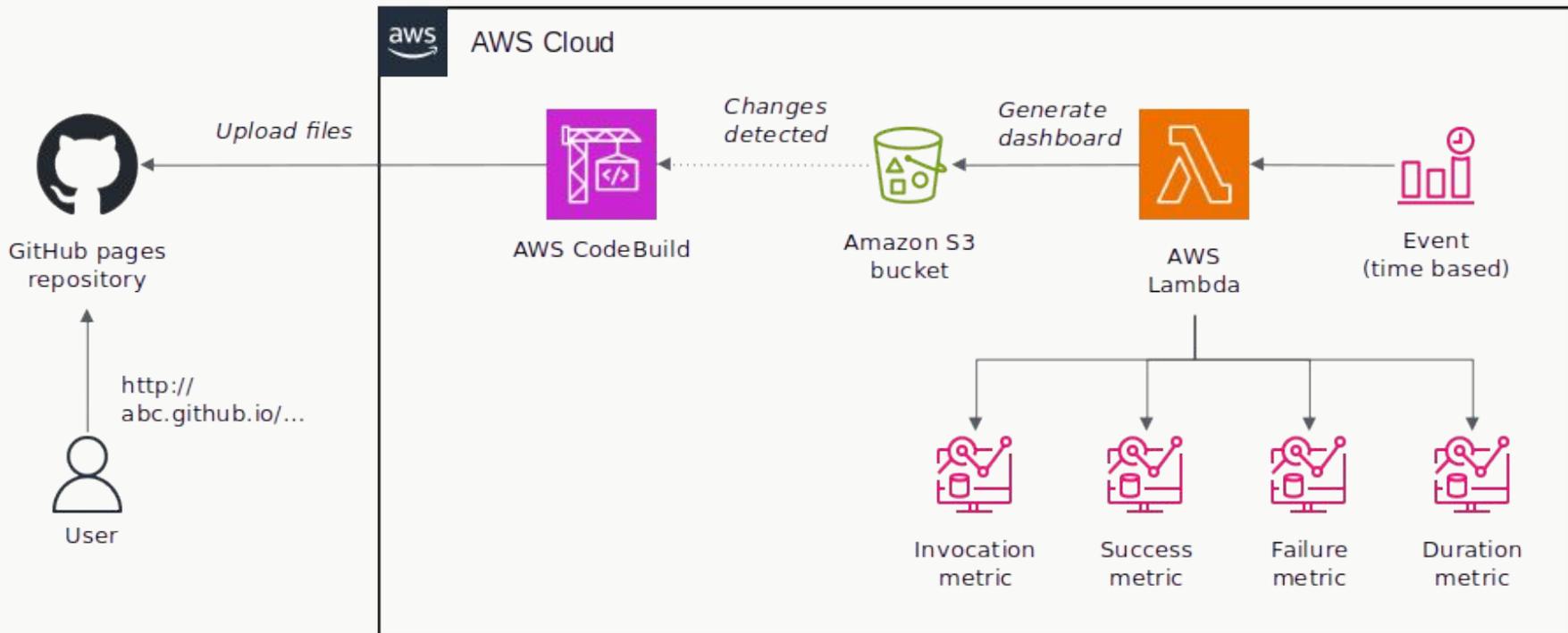
✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

System name ▲	Successes ▾	Failures ▾	Duration ▾	Last 10 runs (latest on the right)
<a href="#">example_appflow_run</a>	47	0	a minute	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
<a href="#">example_athena</a>	46	0	a minute	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
<a href="#">example_batch</a>	47	0	3 minutes	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
<a href="#">example_bedrock</a>	45	0	a minute	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
<a href="#">example_bedrock_retrieve_and_generate</a>	44	2	8 minutes	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
<a href="#">example_cloudformation</a>	45	0	2 minutes	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
<a href="#">example_comprehend</a>	46	0	a minute	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓
<a href="#">example_comprehend_document_classifier</a>	45	0	14 minutes	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓

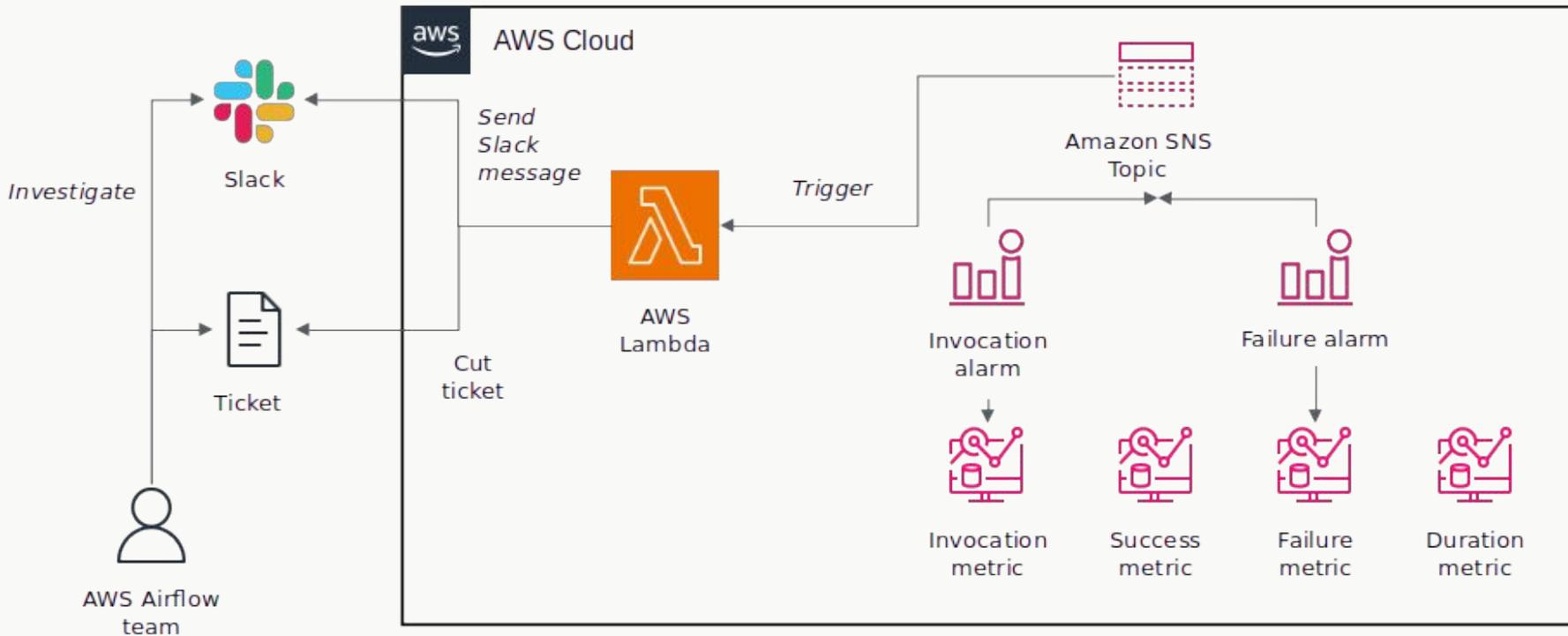
# Test Execution



# Generate the Dashboard



# Notify on Failures



# What's Next?



# Machine-readable Standardized Output

A standardized JSON-formatted output schema based on the JUnit XML format will allow users to aggregate the various provider-managed dashboards into one that suits their purposes.

## Proposed Schema

```
"testsuite": {  
  "provider": string,      [REQUIRED]  
  "tests": int,            [REQUIRED]  
  "errors": int,          [REQUIRED]  
  "failures": int,        [REQUIRED]  
  "skipped": int,         [REQUIRED]  
  "timestamp": string,    [REQUIRED] // Standardize on UTC  
  "duration": float,      [OPTIONAL] // Milliseconds  
  "properties": {},       [OPTIONAL] // Free-form and optional
```

```
{  
  "name": string,          [OPTIONAL]  
  "file": string,          [REQUIRED]  
  "duration": float,       [OPTIONAL] // Milliseconds  
  "result": {  
    "state": "SUCCESS" | "SKIPPED" | "FAILURE", [REQUIRED]  
    "message": string,    [OPTIONAL]  
    "type": string,       [OPTIONAL] // Exception type if failure  
  },  
  "properties": {},       [OPTIONAL] // Free-form and optional  
},  
"properties": {},        [OPTIONAL] // Free-form and optional  
]  
}
```

# Sample - Bare Minimum



```
{
  "testsuite": {
    "provider": "AWS",
    "tests": 1,
    "errors": 0,
    "failures": 1,
    "skipped": 1,
    "timestamp": "2020-01-26T13:45:02",
    "testcases": [
      // Example bare minimum
      {
        "file": "tests/system/providers/amazon/aws/example_appflow.py",
        "result": {
          "state": "SUCCESS"
        }
      },
    ]
  }
}
```

Provider Name

Test counts

Timestamp

List of Test Details

# Sample - Successful

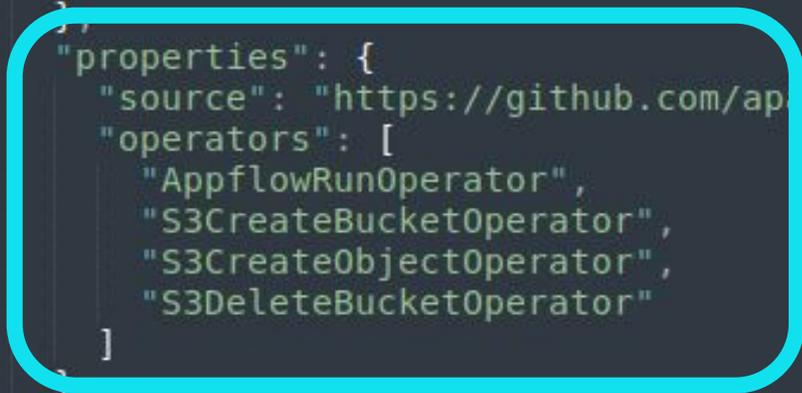


```
// Example successful test with some optional fields
```

```
{  
  "name": "Amazon AppFlow",  
  "file": "tests/system/providers/amazon/aws/example_appflow.py",  
  "duration": 45.87,  
  "result": {  
    "state": "SUCCESS"  
  },  
  "properties": {  
    "source": "https://github.com/apache/airflow/blob/main/tests/system/pro  
    "operators": [  
      "AppflowRunOperator",  
      "S3CreateBucketOperator",  
      "S3CreateObjectOperator",  
      "S3DeleteBucketOperator"  
    ]  
  }  
},
```

Pretty test name

Test run duration



# Sample - Skipped

```
// Example of a test case that was skipped.
{
  "name": "example_athena",
  "file": "tests/system/providers/amazon/aws/example_athena.py",
  "duration": 0.01,
  "result": {
    "state": "SKIPPED",
    "message": "Test runtime is very long; see docs for manual execution."
  },
  "properties": {
    "docs": "<URI for relevant doc page>"
  }
},
```

# Sample - Failed

```
// Example of a test case that failed.
{
  "name": "example_batch",
  "file": "tests/system/providers/amazon/aws/example_batch.py",
  "duration": 94.01,
  "result": {
    "state": "FAILURE",
    "message": "Some failure message, maybe a link to logs or a stack trace?",
    "type": "AssertionError",
    "properties": {
      "source": "https://github.com/apache/airflow/blob/main/tests/system/providers/amazon/aws/example_batch.py",
      "failed_tasks": [
        "watcher",
        "run_pipeline_job"
      ]
    }
  }
},
```

# Questions?



**Slide Deck**

