

**Airflow That Remembers:  
The Dag Versioning Era is  
here!**

# Jed Cunningham

Principal Software Engineer @ astronomer.io

**Committer**

**PMC Member**



# ASTRONOMER

ASTRONOMER

# Ephraim Anierobi

Senior Software Engineer @ astronomer.io

**Committer**

**PMC Member**



# ASTRONOMER

# Slido for questions



OR

Join link:

<https://app.sli.do/event/tFD2BRhuH4jT3rD2nBwntg>

Which features would you like to see in Airflow? (multiple choice)

**52.2%** (391)

DAG versioning

**34.4%** (258)

More data lineage

**30%** (225)

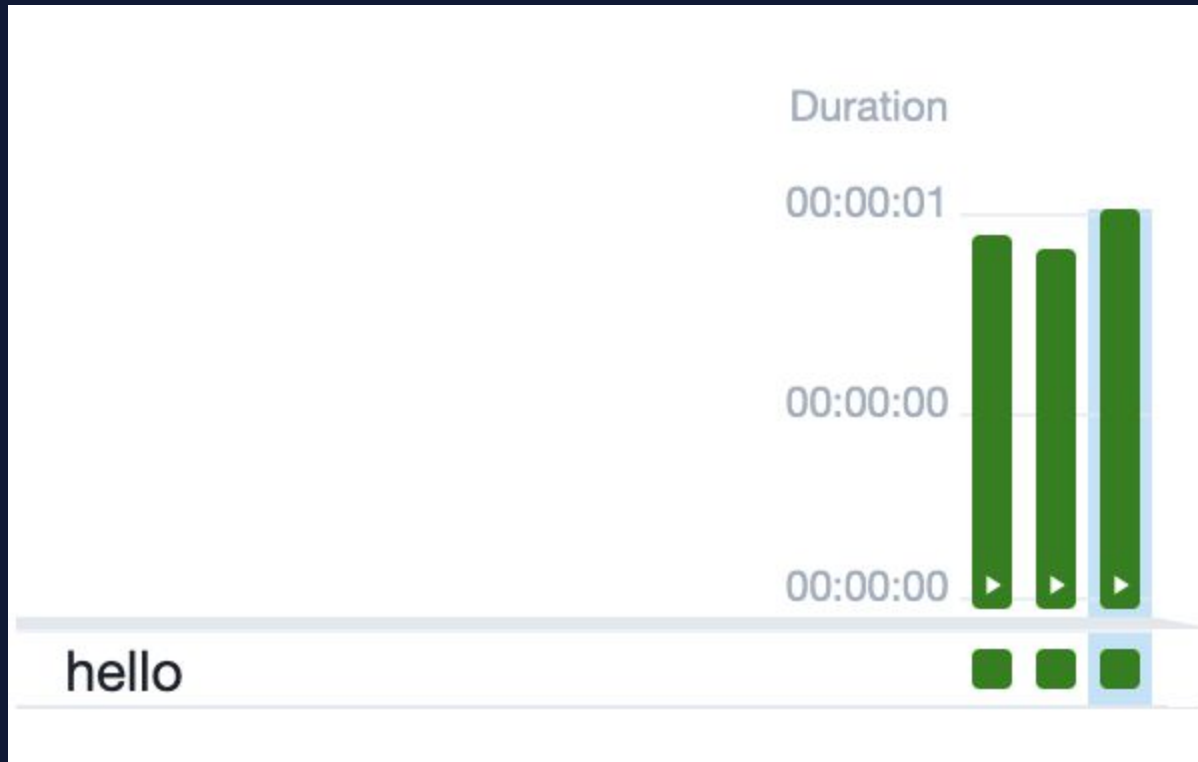
Multi-Tenancy

**28.6%** (214)

Submitting new DAGs externally via API



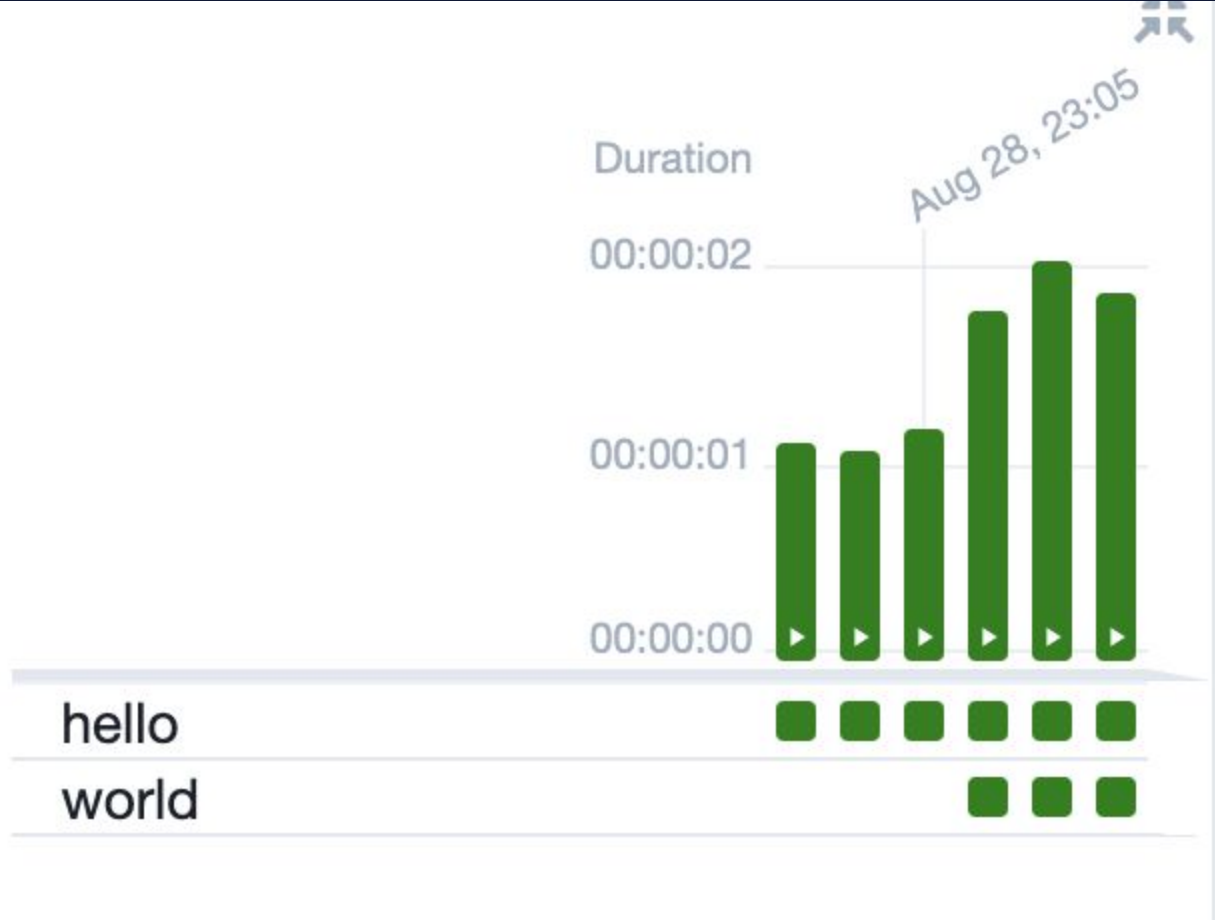
```
with DAG("airflow_summit_2025"):  
    EmptyOperator(task_id="hello")
```





```
with DAG("airflow_summit_2025"):  
    h = EmptyOperator(task_id="hello")  
    w = EmptyOperator(task_id="world")  
    h >> w
```





hello

■ success

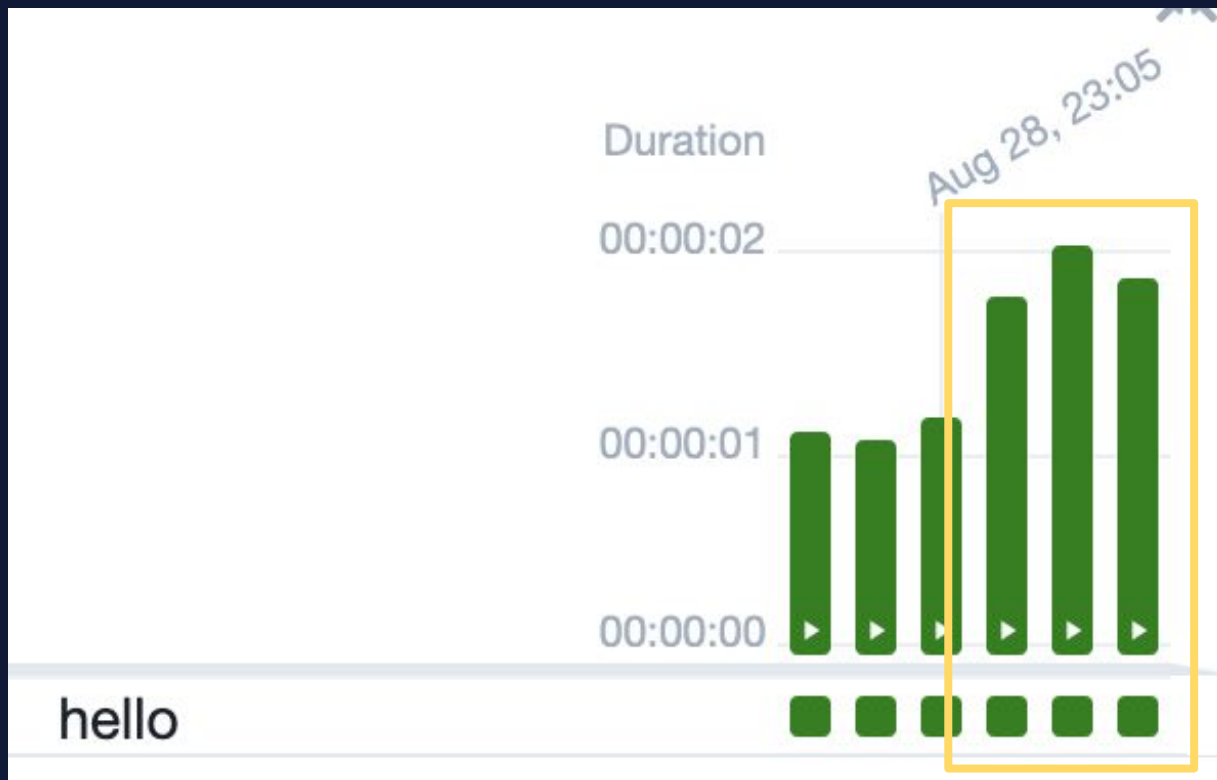
EmptyOperator

world

EmptyOperator



```
with DAG("airflow_summit_2025"):  
    h = EmptyOperator(task_id="hello")  
    # w = EmptyOperator(task_id="world")  
    # h >> w
```



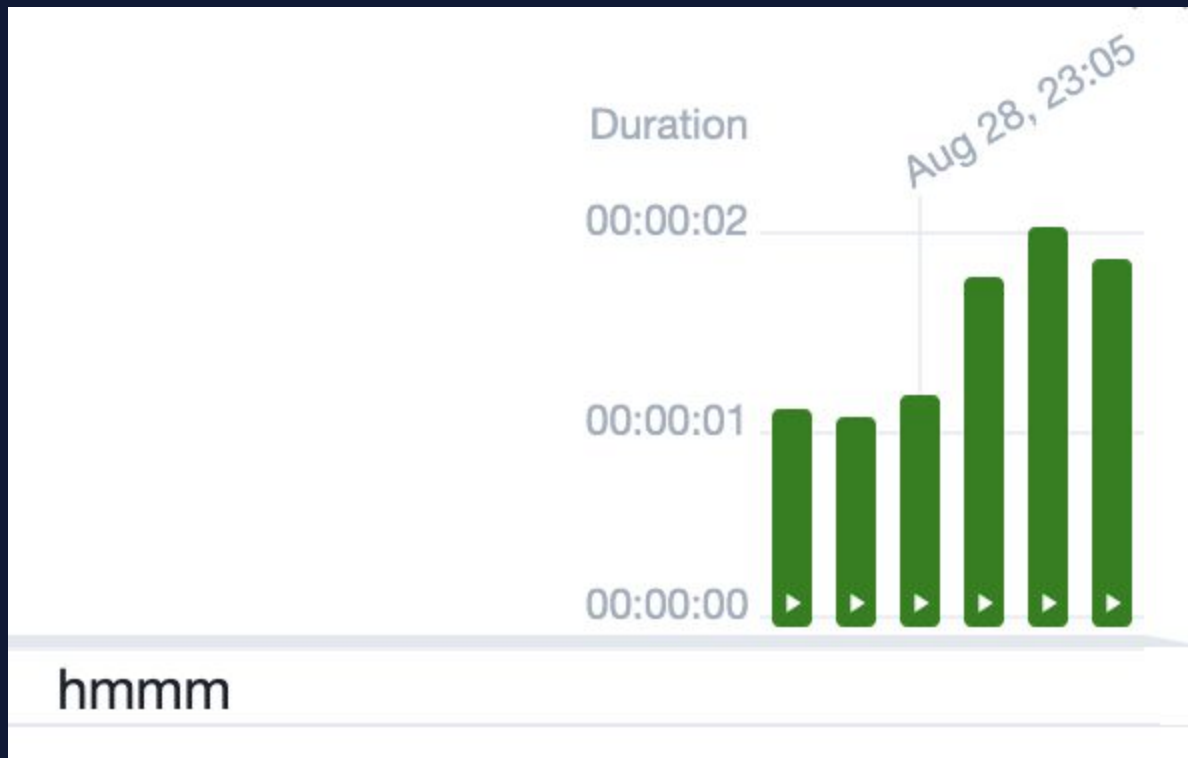
hello

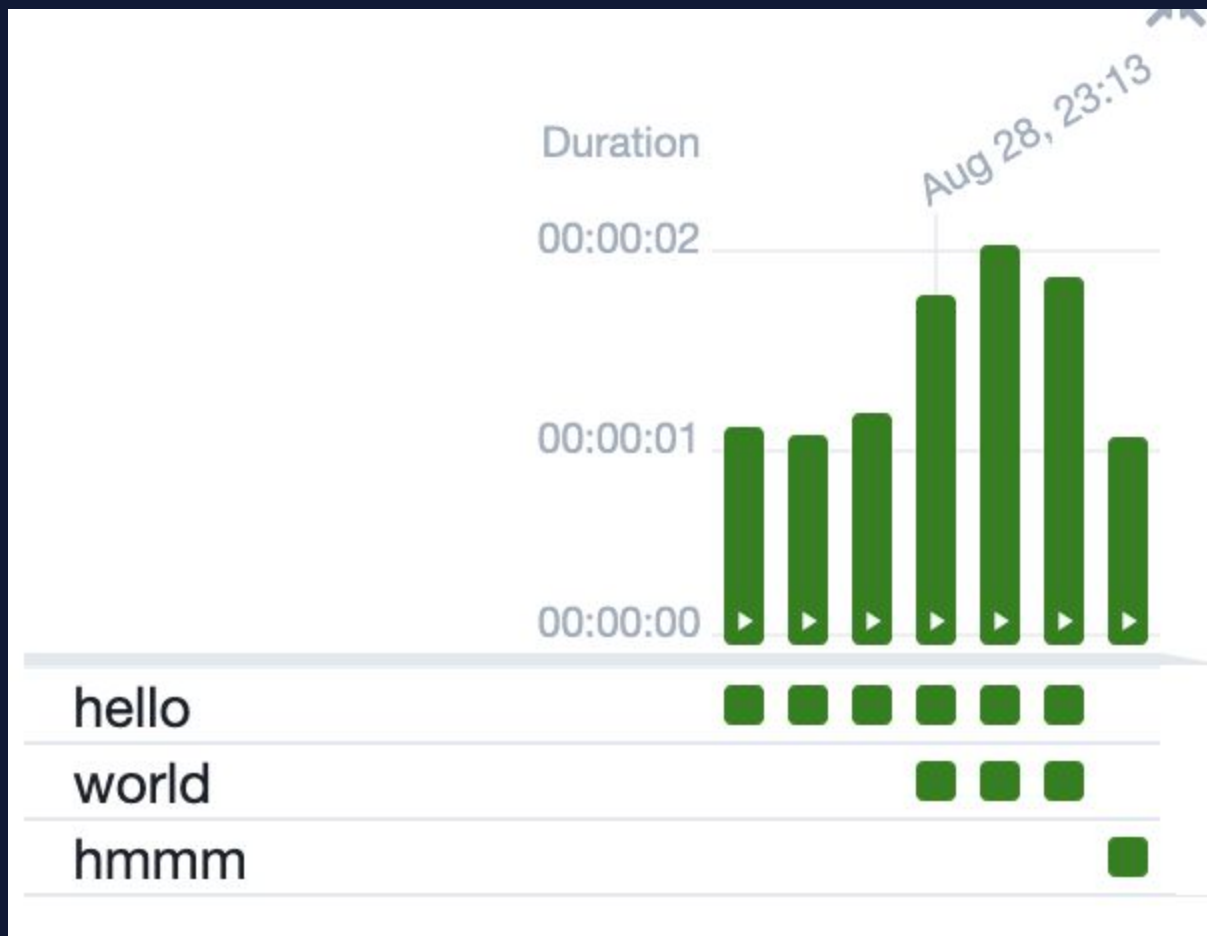
■ success

EmptyOperator



```
with DAG("airflow summit 2025"):  
    EmptyOperator(task_id="hmm")
```







hmmm

■ success

EmptyOperator

hello

EmptyOperator

world

EmptyOperator

# Dag Versioning

**Execution**

(requires config)

**UI**

(automatic)

# Dag Bundle

**collection of Dags and other files**

ASTRONOMER

# Types of Dag Bundles

Versioned vs not

ASTRONOMER

**Run task using a  
specific version of  
Dag code**

ASTRONOMER

# Dag code from **many sources**

ASTRONOMER

# Bundle Versions + DagRuns

V1 Bundle

V2 Bundle



Versioned

GitDagBundle

Non-versioned

LocalDagBundle

S3DagBundle

GCSDagBundle





```
[dag_processor]
dag_bundle_config_list = []

# global default
refresh_interval = 300

# need local scratch disk for bundles?
dag_bundle_storage_path = ... # (sane default)
```

## [dag\_processor] dag\_bundle\_config\_list

```
[
  {
    "name": "my-git-repo",
    "classpath": "airflow.providers.git.bundles.git.GitDagBundle",
    "kwargs": {
      "subdir": "dags",
      "tracking_ref": "main",
      "refresh_interval": 84600,
      "view_url_template": "https://github.com/e/repo/tree/{version}/{subdir}",
    }
  },
  {
    "name": "dags-folder",
    "classpath": "airflow.dag_processing.bundles.local.LocalDagBundle",
    "kwargs": {
      "path": "/files/dags",
      "view_url_template": "https://github.com/airflowdags",
    }
  }
]
```

# Dag bundles lifecycle

## Discovery

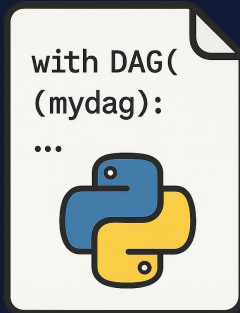
- refresh\_interval



```
# Worker Cleanup Configurations
```

```
[dag processor]  
stale_bundle_cleanup_interval  
[dag processor]  
stale_bundle_cleanup_age_threshold  
[dag processor]  
stale_bundle_cleanup_min_versions
```

# Observed dag version



# Sample JSON Serialization



```
{
  "dag": {
    "tasks": [...],
    "dag_id": "...",
    "params": ...,
    ...
    "start_date": ...,
    "task_group": {...},
    ...
  },
  "__version": 2
}
```

# Sample JSON Serialization

```
{
  "tasks": [
    {
      "__var": {
        "task_id": "...",
        "ui_color": "...",
        "task_type": "...",
        ...
      },
      "__type": "..."
    }
  ]
}

"__version": 2
}
```

# Observed dag version



```
@task
def sometask():
    print("abc")

@task
def anothertask():
    print("hello world")
```



```
@task
def sometask():
    print("something new")

@task
def helloworld():
    print("hello world")
```

How do we create new  
observed dag versions?





```
def write_dag():  
    If serdag_hash == most_recent_serdag_hash:  
        log.debug("unchanged, skipping write to db")  
        return False  
    ...
```



```
def write_dag():  
    ...  
    If {existing run}:  
        {create new dag version}  
    else:  
        {update existing dag version}
```

Schedule	Latest Run	Next Run	Owner	Tags	Latest Dag Version
			airflow		v1

Overview Runs Tasks Calendar Audit Log Code Details

Parsed at: 2025-09-25 17:37:16

Parse Duration: 0.08s

v1 2025-09-25 13:52:32

Copy

Wrap

```
1 from airflow import DAG
2 from datetime import datetime
3 from airflow.decorators import task
4
5
6 with DAG(dag_id="demo2", start_date=datetime(2025, 5, 1)):
7
8     @task
9     def sometask():
10         print("something new")
11
12     sometask()
13
```



```
dag_was_updated = SerializedDagModel.write_dag(...)  
if not dag_was_updated:  
    DagCode.update_source_code(...)
```

# How are bundles versioned?

# Bundle and Observed dag version, what is the difference?

Bundle version → The version of a Dag bundle (the whole code snapshot)

DAG version → The version of a single Dag definition as observed after parsing/serialization.

# A look into dag parsing loop

- **Version-aware refresh**
- **Per-bundle discovery**
- **Reconciliation after refresh:**
  - Removed files & import errors
- **Isolation & parallelism by file**

# Bundle Versions + DagRuns

V1 Bundle

V2 Bundle







Dag  
demo



Home



Dags



Assets



Browse



Admin



Plugins



Docs



User



Options



sleep  
hello

Search Dags

Trigger

demo

Favorite Dag



Schedule

Latest Run

Next Run

Owner  
airflow

Tags

Latest Dag Version  
v1

Overview

Runs

Tasks

Calendar

Audit Log

Code

Details

Parsed at:

2025-09-29

09:32:30

Parse

Duration:

0.03s

dagDetails.bundleVersion:

3eeb907b73ffce1f0280ad290c4173c3fc

d540e4

v1 2025-09-29 09:32:30

Copy

Wrap

```
1 from airflow.sdk import DAG
2 from airflow.providers.standard.operators.bash import BashOperator
3
4 with DAG(dag_id="demo"):
5     sleep = BashOperator(task_id="sleep", bash_command="sleep 60")
6     hello = BashOperator(task_id="hello", bash_command="echo 'Hello Airflow Summit!'")
7
8     sleep >> hello
9
```



Home

Dags

Assets

Browse

Admin

Plugins

Docs

User

Dag demo Dag Run 2025-09-29 09:34:03



Options



2025-09-29 09:34:03 Success

Add a note

Clear Dag Run

Mark Dag Run as...

Logical Date	Run Type	Start Date	End Date	Duration	Triggering User Name
2025-09-29 09:34:03	manual	2025-09-29 09:34:05	2025-09-29 09:34:18	00:00:12	Anonymous

Dag Version(s)

v2

Task Instances Asset Events Audit Log Code Details

Search Tasks

All States

Task ID	Map Index	State	Start Date	End Date	Try Number	Pool
sleep		Success	2025-09-29 09:34:05	2025-09-29 09:34:16	1	default_
hello_new		Success	2025-09-29 09:34:16	2025-09-29 09:34:17	1	default_

# Rerunning dags/taskinstance: versioning behaviour

**When you  
clear a  
dag/task  
instances**

**You have two options:**

- 1. Original serialized dag/bundle version (default)**
- 2. Latest dag/bundle version**

**\*Only applicable to versioned bundles\***

## Clear Dag Run: scheduled\_\_2025-09-25T00:00:00+00:00

Clear existing tasks

Clear only failed tasks

Queue up new tasks

### Affected Tasks: 3

Task ID	State	Map Index	Run ID
hello	✓ Success	-1	
astronomer	✓ Success	-1	
sleep	✓ Success	-1	

### Note

☒ Run with latest bundle version

Confirm





Airflow

x

+

localhost:28080/dags/demo/runs/scheduled\_\_2025-09-26T00:00+00:00/

☆

🔖

Work

⋮

Apps

8080

28080

Airflow Doc

Airflow Repo

Runtime Repo

dev/list

private/list

announce/list

QAWorkspace - As...

Pipelines - astrono...

Main Board - Airflo...

Notes - Airflow 2...

APM Home (env:p...

»

All Bookmarks

Home

Dags

Assets

Browse

Admin

Docs

User

Dag

demo

Dag Run

2025-09-26 01:00:00

Options

sleep

hello

astronomer

6s

3s

0s

13:16:04

13:16:04

13:16:05

13:16:05

13:16:06

13:16:07

13:16:07

13:16:08

2025-09-26 01:00:00

✓ Success

Add a note

Clear Dag Run

Mark Dag Run as...

Logical Date

Run Type

Start Date

End Date

Duration

Dag Version(s)

2025-09-26 01:00:00

🕒 scheduled

2025-09-26 13:16:04

2025-09-26 13:16:09

5.28s

v4

Task Instances

Asset Events

Audit Log

Code

Details

Q Search Tasks

All States

Task ID

Map Index

State

Start Date

Try Number

Operator

Duration

sleep

✓ Success

2025-09-26 13:16:04

3

BashOperator

2.84s

🔄

⌵

🗑

hello

✓ Success

2025-09-26 13:16:07

3

BashOperator

0.40s

🔄

⌵

🗑

astronomer

✓ Success

2025-09-26 13:16:08

1

BashOperator

0.37s

🔄

⌵

🗑

1ER



# Clearing DAGs / Task Instances; What Changes vs. What Doesn't

**Only versions change:** Updates Serialized Dag Version and Bundle Version on your request.

**Topology:** Because of the above request, a new task can be added or tasks marked removed but the TIs on the dagrun are never removed

**Non-versioned bundle:** No option to clear with the latest bundle version since it runs always with the latest version change

**Custom Bundles  ?**  
**Of course!**

ASTRONOMER



```
class MyGitBundle(BaseDagBundle):  
    supports_versioning: bool = True  
  
    def path() -> Path:  
        return {path_to_git_repo}  
  
    def refresh() -> None:  
        # Update bundle (e.g. git pull)  
  
    def get_current_version() -> str:  
        return {current git sha}
```



```
class MyGitBundle(BaseDagBundle):  
    ...  
  
    def __init__(custom: str) -> None:  
        self.custom = custom  
  
    def initialize() -> None:  
        # do something expensive (e.g. git clone)  
        # run on dag processor and worker!  
  
    def view_url_template(self) -> str | None:  
        ...  
        return {url template}
```

# Slido for questions



OR

Join link:

<https://app.sli.do/event/tFD2BRhuH4jT3rD2nBwntg>

Recap 🍺🍺

ASTRONOMER

# Dag Versioning

**Execution**

(requires config)

**UI**

(automatic)

# Bundles for code versioning

ASTRONOMER



# Dag versions for history: UI + scheduler

ASTRONOMER

# The 2025 Apache Airflow<sup>®</sup> Survey is here!

Fill it out to for a free Airflow 3  
Fundamentals or DAG Authoring in  
Airflow 3 certification code



**Questions!**  
**slido?**

ASTRONOMER