The Silent Symphony

Keeping Airflow's CI/CD and Dev Tools in Tune

Jarek Potiuk

- Independent Open-Source Contributor and advisor
- PMC Member & Committer Apache Airflow
- Member of the Apache Software Foundation
- Security Committee Member of the ASF

𝕏 @jarekpotiuk

@potiuk

in @jarekpotiuk

https://fosstodon.org/@jarekpotiuk

Agenda

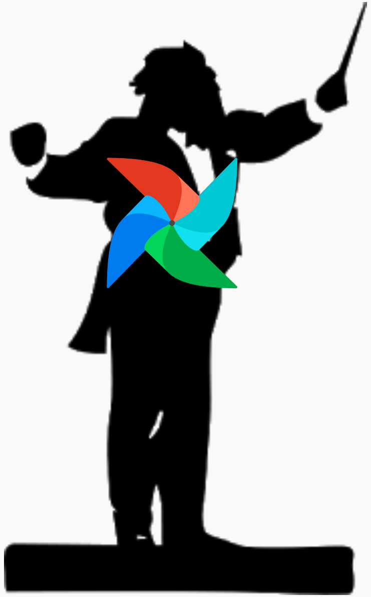
- Airflow and orchestra
- Behind the scenes
- Unsung heroes for Airflow staying in tune
- Learnings for other projects

Airflow as Orchestrator

APACHE Spark

docker

kubernetes

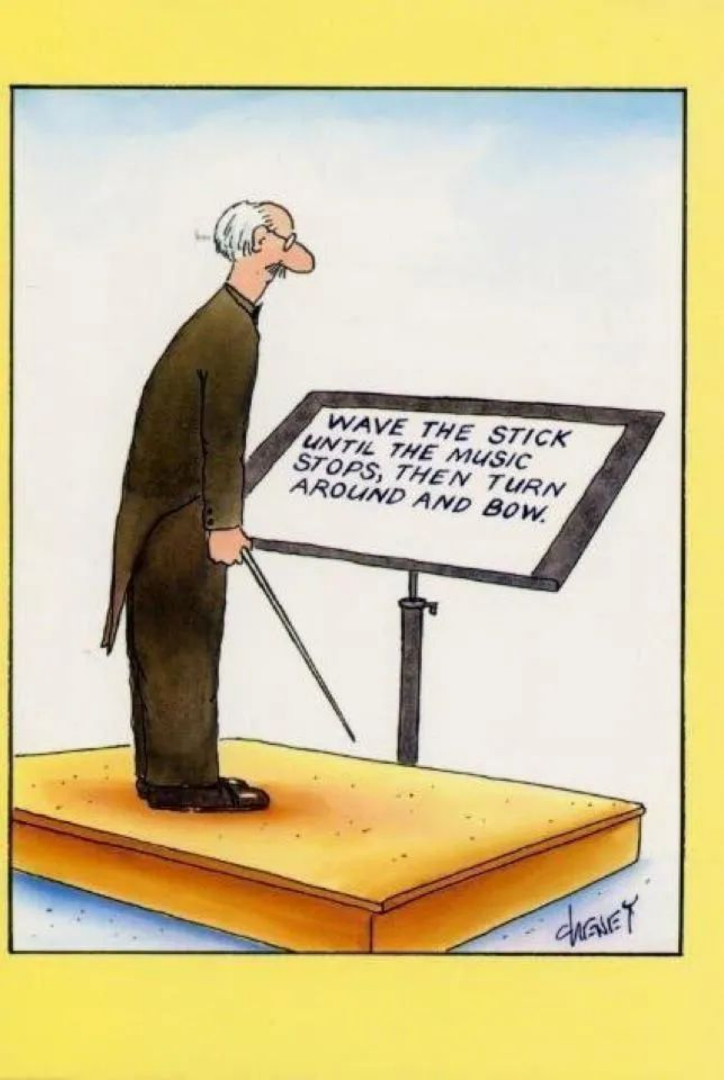


Google Cloud

amazon webservices

Azure

AIRFLOW SUMMIT 2024 10 YEARS OF AIRFLOW SF·USA


WAVE THE STICK UNTIL THE MUSIC STOPS, THEN TURN AROUND AND BOW.

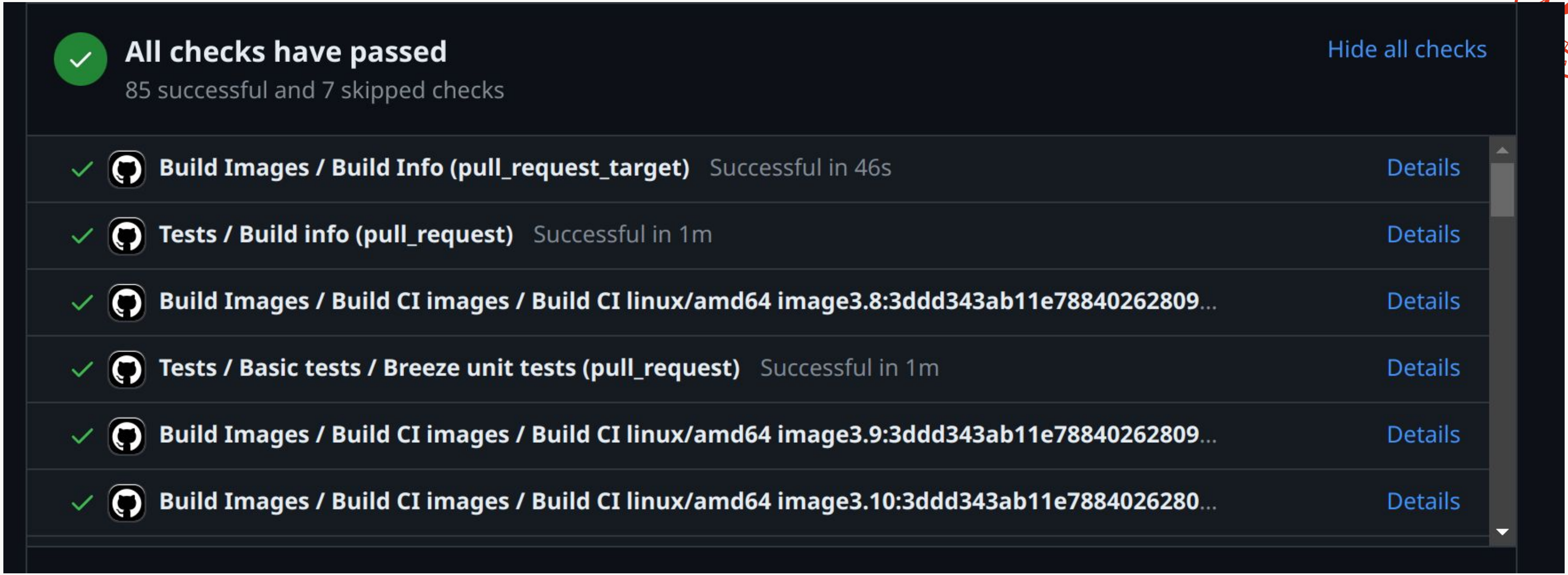
Airflow complexity

- Airflow Core (UI, Scheduler, Workers, Triggerers)
- 90+ providers (790+ dependencies)
- Container images
- Helm chart
- Python client

Orchestra

Sanderson Accu-Tuner

What makes things work (semi) smoothly



All checks have passed

85 successful and 7 skipped checks

✓ **Build Images / Build Info (pull_request_target)** Successful in 46s Details

✓ **Tests / Build info (pull_request)** Successful in 1m Details

✓ **Build Images / Build CI images / Build CI linux/amd64 image3.8:3ddd343ab11e78840262809...** Details

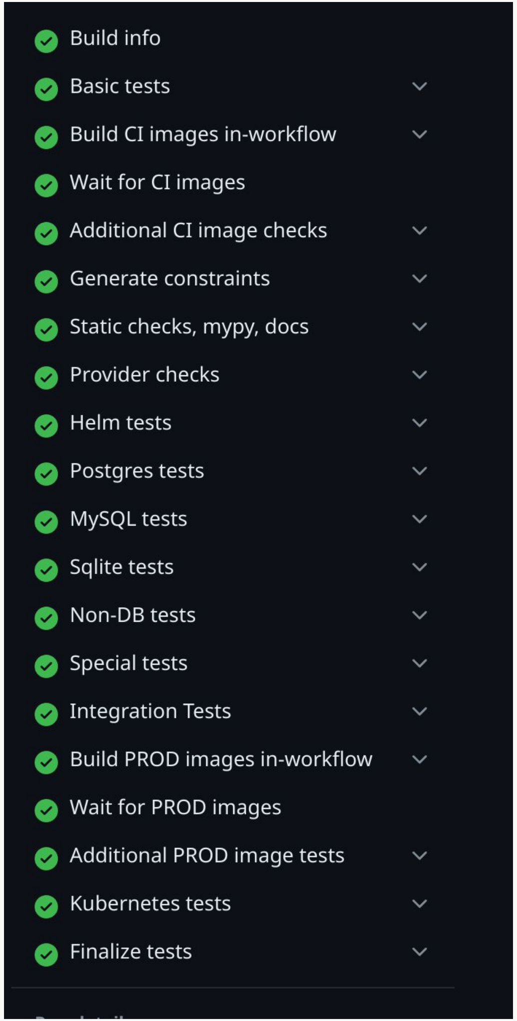
✓ **Tests / Basic tests / Breeze unit tests (pull_request)** Successful in 1m Details

✓ **Build Images / Build CI images / Build CI linux/amd64 image3.9:3ddd343ab11e78840262809...** Details

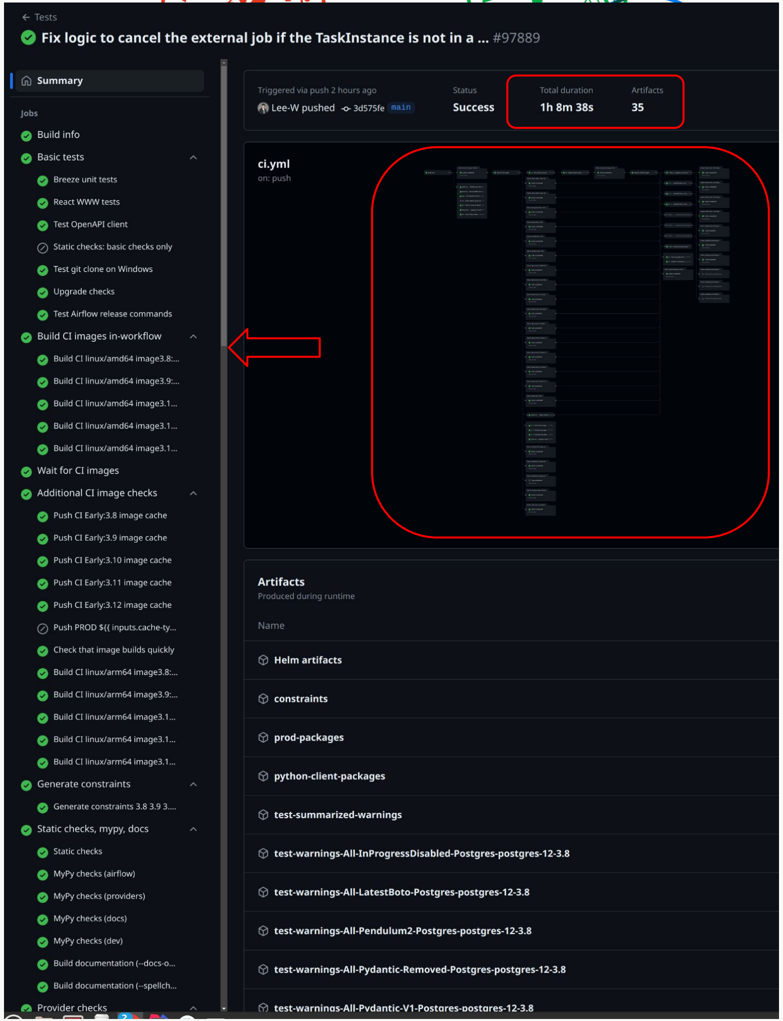
✓ **Build Images / Build CI images / Build CI linux/amd64 image3.10:3ddd343ab11e7884026280...** Details

In the background: CI/CD

- static checks (pre-commits)
- documentation build
- testing:
    - units of code
    - integrations
    - upgrades
    - system tests
    - end-to-end tests
    - building artifacts (packages/images/charts)
    - backwards compatibility checks
    - lowest dependencies
- reproducible installation for users
- future-compatibility for dependencies



- ✓ Build info
- ✓ Basic tests
- ✓ Build CI images in-workflow
- ✓ Wait for CI images
- ✓ Additional CI image checks
- ✓ Generate constraints
- ✓ Static checks, mypy, docs
- ✓ Provider checks
- ✓ Helm tests
- ✓ Postgres tests
- ✓ MySQL tests
- ✓ Sqlite tests
- ✓ Non-DB tests
- ✓ Special tests
- ✓ Integration Tests
- ✓ Build PROD images in-workflow
- ✓ Wait for PROD images
- ✓ Additional PROD image tests
- ✓ Kubernetes tests
- ✓ Finalize tests



← Tests

✓ Fix logic to cancel the external job if the TaskInstance is not in a ... #97889

Summary

Jobs

- ✓ Build info
- ✓ Basic tests
  - ✓ Breeze unit tests
  - ✓ React WWW tests
  - ✓ Test OpenAPI client
  - ⊘ Static checks: basic checks only
  - ✓ Test git clone on Windows
  - ✓ Upgrade checks
  - ✓ Test Airflow release commands
- ✓ Build CI images in-workflow
  - ✓ Build CI linux/amd64 image3.8:...
  - ✓ Build CI linux/amd64 image3.9:...
  - ✓ Build CI linux/amd64 image3.1...
  - ✓ Build CI linux/amd64 image3.1...
  - ✓ Build CI linux/amd64 image3.1...
- ✓ Wait for CI images
- ✓ Additional CI image checks
  - ✓ Push CI Early:3.8 image cache
  - ✓ Push CI Early:3.9 image cache
  - ✓ Push CI Early:3.10 image cache
  - ✓ Push CI Early:3.11 image cache
  - ✓ Push CI Early:3.12 image cache
  - ⊘ Push PROD ${{ inputs.cache-ty...
  - ✓ Check that image builds quickly
  - ✓ Build CI linux/arm64 image3.8:...
  - ✓ Build CI linux/arm64 image3.9:...
  - ✓ Build CI linux/arm64 image3.1...
  - ✓ Build CI linux/arm64 image3.1...
  - ✓ Build CI linux/arm64 image3.1...
- ✓ Generate constraints
  - ✓ Generate constraints 3.8 3.9 3....
- ✓ Static checks, mypy, docs
  - ✓ Static checks
  - ✓ MyPy checks (airflow)
  - ✓ MyPy checks (providers)
  - ✓ MyPy checks (docs)
  - ✓ MyPy checks (dev)
  - ✓ Build documentation (--docs-o...
  - ✓ Build documentation (--spellch...
- ✓ Provider checks

Triggered via push 2 hours ago
Lee-W pushed 3d575fe main

Status
Success

Total duration
1h 8m 38s

Artifacts
35

ci.yml
on: push

Artifacts
Produced during runtime

Name

- Helm artifacts
- constraints
- prod-packages
- python-client-packages
- test-summarized-warnings
- test-warnings-All-InProgressDisabled-Postgres-postgres-12-3.8
- test-warnings-All-LatestBoto-Postgres-postgres-12-3.8
- test-warnings-All-Pendulum2-Postgres-postgres-12-3.8
- test-warnings-All-Pydantic-Removed-Postgres-postgres-12-3.8
- test-warnings-All-Pydantic-V1-Postgres-postgres-12-3.8

Unsung heroes: dev tools

- Make it easy to develop project
- Airflow has more than 3000 contributors
- Contributors vs Maintainers
- CI/CD to keep things in check
- Role of dev tools
    - help contributors to contribute
    - help maintainers to release

Dev Tools vs. CI

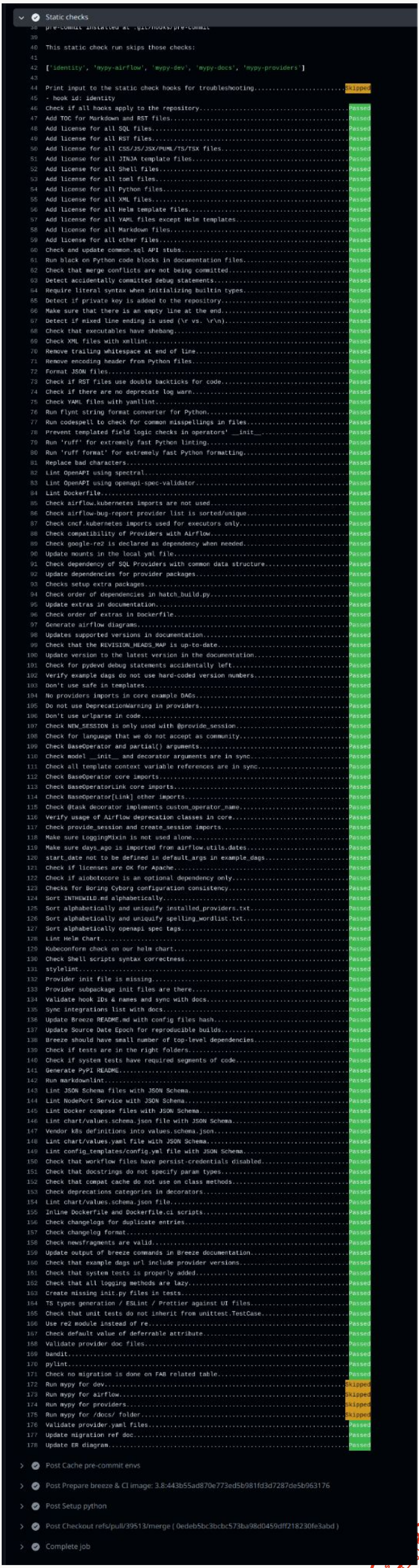
Two sides of the same coin

- CI / CD runs all the checks
- What happens when it fails?
- Reproducing the failures locally
- “Works for me” syndrome
- common test execution platform (790 dependencies)
- testing everything
    - Airflow, Providers, Container image, Chart, Client, Release process

Pre-commits

Pre-commits AKA static-checks

- leverage .git pre-commit hooks in a very efficient way
- can be installed with `pre-commit install`
- we have 130+ (!) pre-commits
- classic static checks (ruff and friends)
- maintaining single source of truth
- generating documentation and licences
- updating versions automatically
- preventing common mistakes (“first line of defence”)
- automates a LOT of stuff



Static checks

```
pre-commit installed at .git/hooks/pre-commit

This static check run skips those checks:

['identity', 'mypy-airflow', 'mypy-dev', 'mypy-docs', 'mypy-providers']

Print input to the static check hooks for troubleshooting.......................Skipped
- hook id: identity
Check if all hooks apply to the repository.......................................Passed
Add TOC for Markdown and RST files...............................................Passed
Add license for all SQL files....................................................Passed
Add license for all RST files....................................................Passed
Add license for all CSS/JS/JSX/PUML/TS/TSX files.................................Passed
Add license for all JINJA template files.........................................Passed
Add license for all Shell files..................................................Passed
Add license for all toml files...................................................Passed
Add license for all Python files.................................................Passed
Add license for all XML files....................................................Passed
Add license for all Helm template files..........................................Passed
Add license for all YAML files except Helm templates.............................Passed
Add license for all Markdown files...............................................Passed
Add license for all other files..................................................Passed
Check and update common.sql API stubs............................................Passed
Run black on Python code blocks in documentation files...........................Passed
Check that merge conflicts are not being committed...............................Passed
Detect accidentally committed debug statements...................................Passed
Require literal syntax when initializing builtin types...........................Passed
Detect if private key is added to the repository.................................Passed
Make sure that there is an empty line at the end.................................Passed
Detect if mixed line ending is used (\r vs. \r\n)................................Passed
Check that executables have shebang..............................................Passed
Check XML files with xmllint.....................................................Passed
Remove trailing whitespace at end of line........................................Passed
Remove encoding header from Python files.........................................Passed
Format JSON files................................................................Passed
Check if RST files use double backticks for code.................................Passed
Check if there are no deprecate log warn.........................................Passed
Check YAML files with yamllint...................................................Passed
Run flynt string format converter for Python.....................................Passed
Run codespell to check for common misspellings in files..........................Passed
Prevent templated field logic checks in operators' __init__......................Passed
Run 'ruff' for extremely fast Python linting.....................................Passed
Run 'ruff format' for extremely fast Python formatting...........................Passed
Replace bad characters...........................................................Passed
Lint OpenAPI using spectral......................................................Passed
Lint OpenAPI using openapi-spec-validator........................................Passed
Lint Dockerfile..................................................................Passed
Check airflow.kubernetes imports are not used....................................Passed
Check airflow-bug-report provider list is sorted/unique..........................Passed
Check cncf.kubernetes imports used for executors only............................Passed
Check compatibility of Providers with Airflow....................................Passed
Check google-re2 is declared as dependency when needed...........................Passed
Update mounts in the local yml file..............................................Passed
Check dependency of SQL Providers with common data structure.....................Passed
Update dependencies for provider packages........................................Passed
Checks setup extra packages......................................................Passed
Check order of dependencies in hatch_build.py....................................Passed
Update extras in documentation...................................................Passed
Check order of extras in Dockerfile..............................................Passed
Generate airflow diagrams........................................................Passed
Updates supported versions in documentation......................................Passed
Check that the REVISION_HEADS_MAP is up-to-date..................................Passed
Update version to the latest version in the documentation........................Passed
Check for pydevd debug statements accidentally left..............................Passed
Verify example dags do not use hard-coded version numbers........................Passed
Don't use safe in templates......................................................Passed
No providers imports in core example DAGs........................................Passed
Do not use DeprecationWarning in providers.......................................Passed
Don't use urlparse in code.......................................................Passed
Check NEW_SESSION is only used with @provide_session.............................Passed
Check for language that we do not accept as community............................Passed
Check BaseOperator and partial() arguments.......................................Passed
Check model __init__ and decorator arguments are in sync.........................Passed
Check all template context variable references are in sync.......................Passed
Check BaseOperator core imports..................................................Passed
Check BaseOperatorLink core imports..............................................Passed
Check BaseOperator[Link] other imports...........................................Passed
Check @task decorator implements custom_operator_name............................Passed
Verify usage of Airflow deprecation classes in core..............................Passed
Check provide_session and create_session imports.................................Passed
Make sure LoggingMixin is not used alone.........................................Passed
Make sure days_ago is imported from airflow.utils.dates..........................Passed
start_date not to be defined in default_args in example_dags.....................Passed
Check if licenses are OK for Apache..............................................Passed
Check if aiobotocore is an optional dependency only..............................Passed
Checks for Boring Cyborg configuration consistency...............................Passed
Sort INTHEWILD.md alphabetically.................................................Passed
Sort alphabetically and uniquify installed_providers.txt.........................Passed
Sort alphabetically and uniquify spelling_wordlist.txt...........................Passed
Sort alphabetically openapi spec tags............................................Passed
Lint Helm Chart..................................................................Passed
Kubeconform check on our helm chart..............................................Passed
Check Shell scripts syntax correctness...........................................Passed
stylelint........................................................................Passed
Provider init file is missing....................................................Passed
Provider subpackage init files are there.........................................Passed
Validate hook IDs & names and sync with docs.....................................Passed
Sync integrations list with docs.................................................Passed
Update Breeze README.md with config files hash....................................Passed
Update Source Date Epoch for reproducible builds.................................Passed
Breeze should have small number of top-level dependencies........................Passed
Check if tests are in the right folders..........................................Passed
Check if system tests have required segments of code.............................Passed
Generate PyPI README.............................................................Passed
Run markdownlint.................................................................Passed
Lint JSON Schema files with JSON Schema..........................................Passed
Lint NodePort Service with JSON Schema...........................................Passed
Lint Docker compose files with JSON Schema.......................................Passed
Lint chart/values.schema.json file with JSON Schema..............................Passed
Vendor k8s definitions into values.schema.json...................................Passed
Lint chart/values.yaml file with JSON Schema.....................................Passed
Lint config_templates/config.yml file with JSON Schema...........................Passed
Check that workflow files have persist-credentials disabled......................Passed
Check that docstrings do not specify param types.................................Passed
Check that compat cache do not use on class methods..............................Passed
Check deprecations categories in decorators......................................Passed
Lint chart/values.schema.json file...............................................Passed
Inline Dockerfile and Dockerfile.ci scripts......................................Passed
Check changelogs for duplicate entries...........................................Passed
Check changelog format...........................................................Passed
Check newsfragments are valid....................................................Passed
Update output of breeze commands in Breeze documentation.........................Passed
Check that example dags url include provider versions............................Passed
Check that system tests is properly added........................................Passed
Check that all logging methods are lazy..........................................Passed
Create missing init.py files in tests............................................Passed
TS types generation / ESLint / Prettier against UI files.........................Passed
Check that unit tests do not inherit from unittest.TestCase......................Passed
Use re2 module instead of re.....................................................Passed
Check default value of deferrable attribute......................................Passed
Validate provider doc files......................................................Passed
bandit...........................................................................Passed
pylint...........................................................................Passed
Check no migration is done on FAB related table..................................Passed
Run mypy for dev................................................................Skipped
Run mypy for airflow............................................................Skipped
Run mypy for providers..........................................................Skipped
Run mypy for /docs/ folder......................................................Skipped
Validate provider.yaml files.....................................................Passed
Update migration ref doc.........................................................Passed
Update ER diagram................................................................Passed
```

Post Cache pre-commit envs

Post Prepare breeze & CI image: 3.8:443b55ad870e773ed5b981fd3d7287de5b963176

Post Setup python

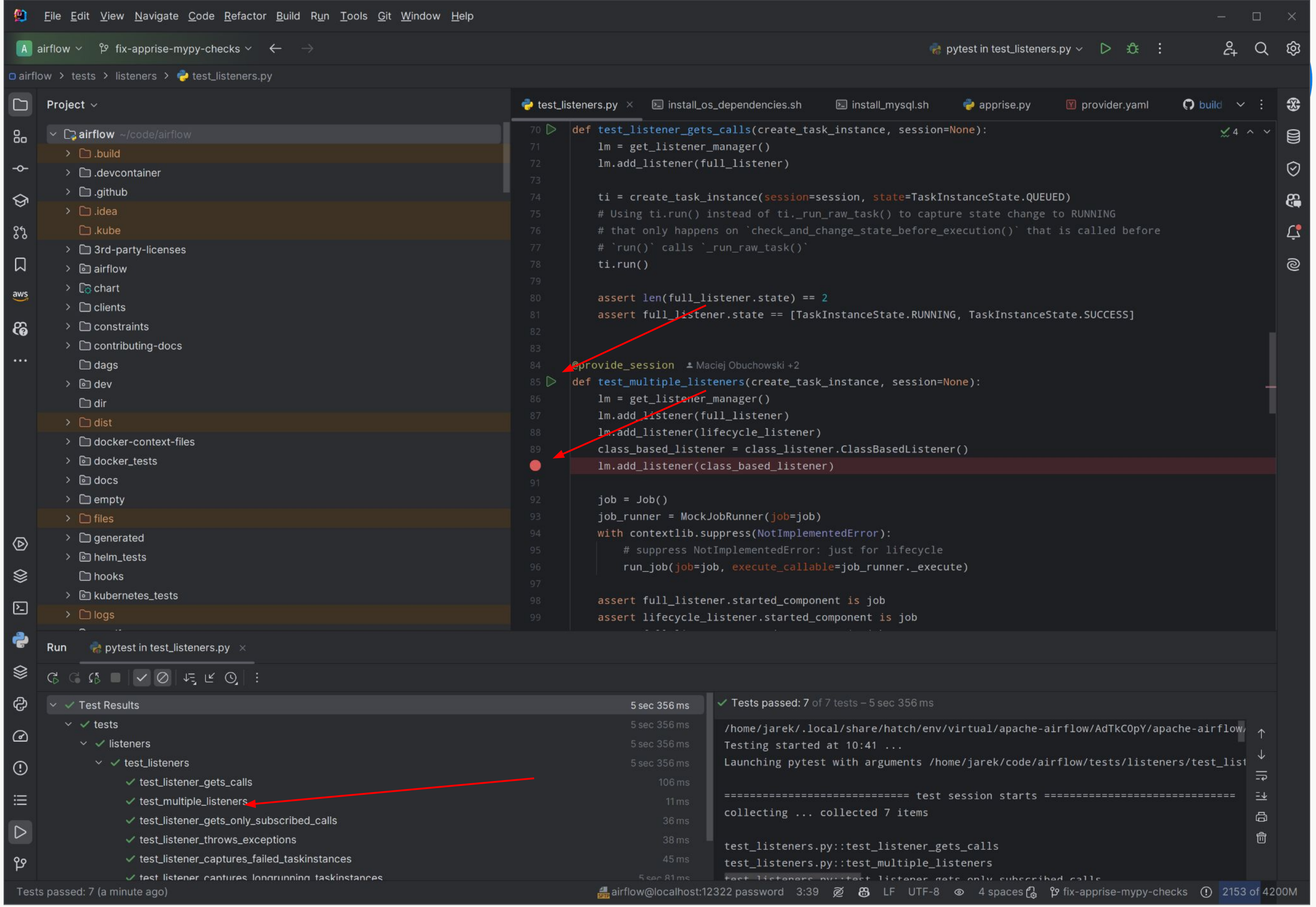
Post Checkout refs/pull/39513/merge (0edeb5bc3bcbc573ba98d0459dff218230fe3abd)

Complete job

Local Venv

Why venv ?

- Pros
  - Finally standards compliant (as of early 2024)
  - Easy to setup and connect to IDE
  - Easy to run and debug tests
  - Integration with build front-ends (hatch, poetry, …)
  - Anyone can use their own tooling
  - Pretty standard across multiple Python projects
- Cons
  - Providers (and 790 dependencies)
  - Various platform support (MacOS/Linux X86/ARM)
  - The “works for me” syndrome
  - Lack of reproducibility with CI



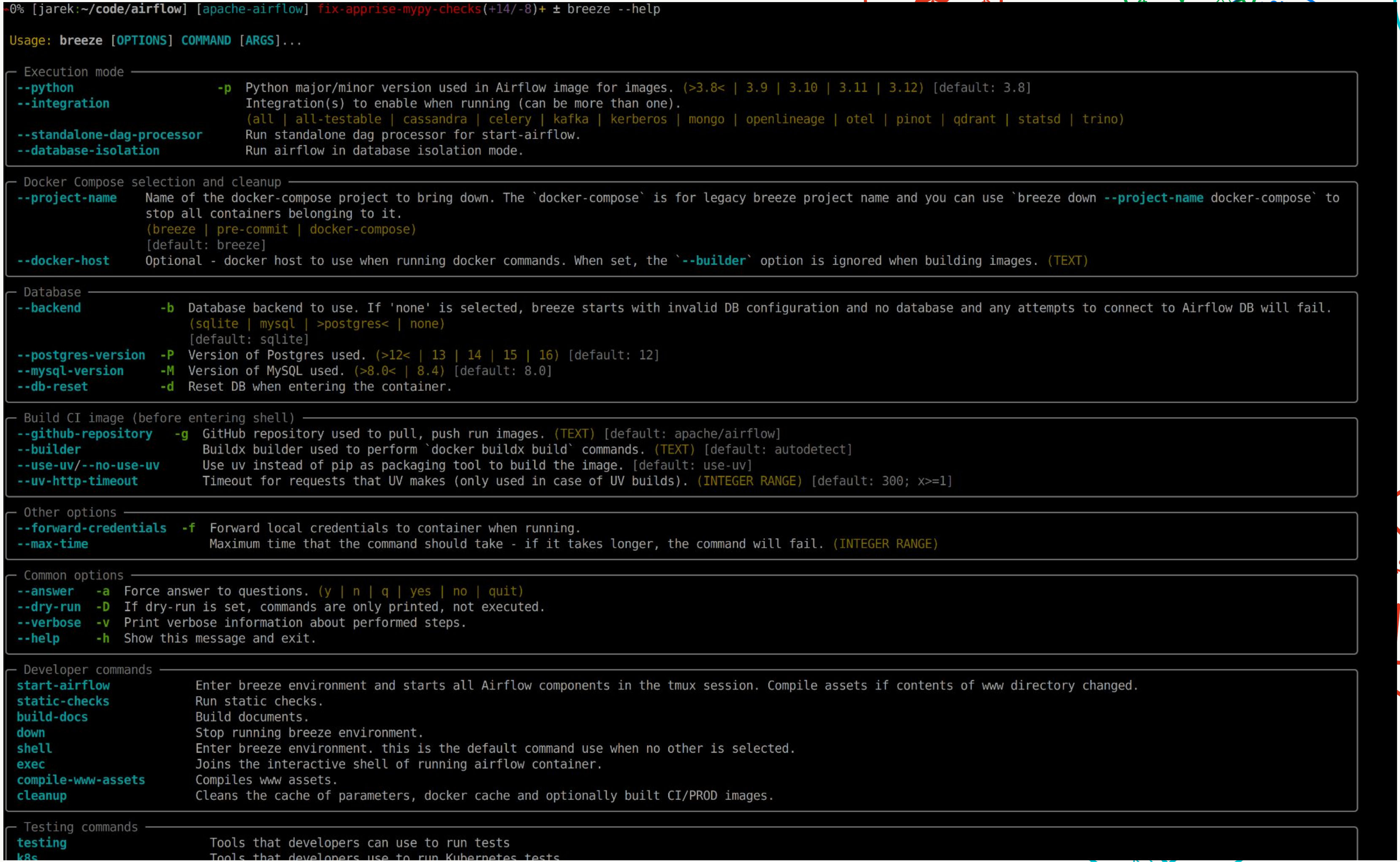
```python
def test_listener_gets_calls(create_task_instance, session=None):
    lm = get_listener_manager()
    lm.add_listener(full_listener)

    ti = create_task_instance(session=session, state=TaskInstanceState.QUEUED)
    # Using ti.run() instead of ti._run_raw_task() to capture state change to RUNNING
    # that only happens on `check_and_change_state_before_execution()` that is called before
    # `run()` calls `_run_raw_task()`
    ti.run()

    assert len(full_listener.state) == 2
    assert full_listener.state == [TaskInstanceState.RUNNING, TaskInstanceState.SUCCESS]


@provide_session
def test_multiple_listeners(create_task_instance, session=None):
    lm = get_listener_manager()
    lm.add_listener(full_listener)
    lm.add_listener(lifecycle_listener)
    class_based_listener = class_listener.ClassBasedListener()
    lm.add_listener(class_based_listener)

    job = Job()
    job_runner = MockJobRunner(job=job)
    with contextlib.suppress(NotImplementedError):
        # suppress NotImplementedError: just for lifecycle
        run_job(job=job, execute_callable=job_runner._execute)

    assert full_listener.started_component is job
    assert lifecycle_listener.started_component is job
```

```
✓ Test Results                                         5 sec 356 ms
  ✓ tests                                              5 sec 356 ms
    ✓ listeners                                        5 sec 356 ms
      ✓ test_listeners                                 5 sec 356 ms
        ✓ test_listener_gets_calls                           106 ms
        ✓ test_multiple_listeners                             11 ms
        ✓ test_listener_gets_only_subscribed_calls            36 ms
        ✓ test_listener_throws_exceptions                     38 ms
        ✓ test_listener_captures_failed_taskinstances         45 ms
        ✓ test_listener_captures_longrunning_taskinstances  5 sec 81 ms
```

```
✓ Tests passed: 7 of 7 tests – 5 sec 356 ms

/home/jarek/.local/share/hatch/env/virtual/apache-airflow/AdTkC0pY/apache-airflow/
Testing started at 10:41 ...
Launching pytest with arguments /home/jarek/code/airflow/tests/listeners/test_list

============================ test session starts ==============================
collecting ... collected 7 items

test_listeners.py::test_listener_gets_calls
test_listeners.py::test_multiple_listeners
test_listeners.py::test_listener_gets_only_subscribed_calls
```

```
Tests passed: 7 (a minute ago)    airflow@localhost:12322 password    3:39    LF    UTF-8    4 spaces    fix-apprise-mypy-checks    2153 of 4200M
```

Breeze



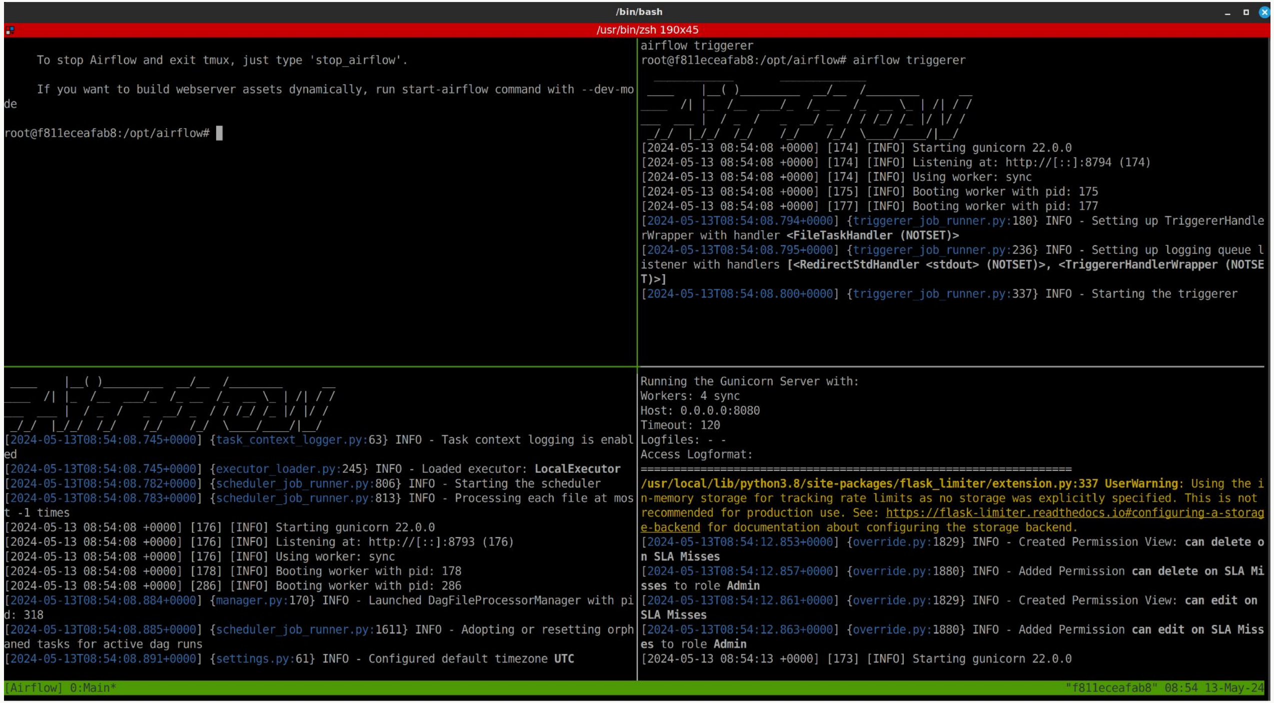
```
0% [jarek:~/code/airflow] [apache-airflow] fix-apprise-mypy-checks(+14/-8)+ ± breeze --help

Usage: breeze [OPTIONS] COMMAND [ARGS]...

╭─ Execution mode ─────────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ --python                      -p  Python major/minor version used in Airflow image for images. (>3.8< | 3.9 | 3.10 | 3.11 | 3.12) [default: 3.8] │
│ --integration                     Integration(s) to enable when running (can be more than one).                              │
│                                   (all | all-testable | cassandra | celery | kafka | kerberos | mongo | openlineage | otel | pinot | qdrant | statsd | trino) │
│ --standalone-dag-processor        Run standalone dag processor for start-airflow.                                            │
│ --database-isolation              Run airflow in database isolation mode.                                                    │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Docker Compose selection and cleanup ───────────────────────────────────────────────────────────────────────────────────────╮
│ --project-name    Name of the docker-compose project to bring down. The `docker-compose` is for legacy breeze project name and you can use `breeze down --project-name docker-compose` to │
│                   stop all containers belonging to it.                                                                       │
│                   (breeze | pre-commit | docker-compose)                                                                     │
│                   [default: breeze]                                                                                          │
│ --docker-host     Optional - docker host to use when running docker commands. When set, the `--builder` option is ignored when building images. (TEXT) │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Database ───────────────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ --backend            -b  Database backend to use. If 'none' is selected, breeze starts with invalid DB configuration and no database and any attempts to connect to Airflow DB will fail. │
│                          (sqlite | mysql | >postgres< | none)                                                                │
│                          [default: sqlite]                                                                                   │
│ --postgres-version   -P  Version of Postgres used. (>12< | 13 | 14 | 15 | 16) [default: 12]                                  │
│ --mysql-version      -M  Version of MySQL used. (>8.0< | 8.4) [default: 8.0]                                                 │
│ --db-reset           -d  Reset DB when entering the container.                                                               │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Build CI image (before entering shell) ─────────────────────────────────────────────────────────────────────────────────────╮
│ --github-repository   -g  GitHub repository used to pull, push run images. (TEXT) [default: apache/airflow]                  │
│ --builder                 Buildx builder used to perform `docker buildx build` commands. (TEXT) [default: autodetect]        │
│ --use-uv/--no-use-uv      Use uv instead of pip as packaging tool to build the image. [default: use-uv]                      │
│ --uv-http-timeout         Timeout for requests that UV makes (only used in case of UV builds). (INTEGER RANGE) [default: 300; x>=1] │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Other options ──────────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ --forward-credentials  -f  Forward local credentials to container when running.                                              │
│ --max-time                 Maximum time that the command should take - if it takes longer, the command will fail. (INTEGER RANGE) │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Common options ─────────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ --answer   -a  Force answer to questions. (y | n | q | yes | no | quit)                                                      │
│ --dry-run  -D  If dry-run is set, commands are only printed, not executed.                                                   │
│ --verbose  -v  Print verbose information about performed steps.                                                              │
│ --help     -h  Show this message and exit.                                                                                   │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Developer commands ─────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ start-airflow         Enter breeze environment and starts all Airflow components in the tmux session. Compile assets if contents of www directory changed. │
│ static-checks         Run static checks.                                                                                     │
│ build-docs            Build documents.                                                                                       │
│ down                  Stop running breeze environment.                                                                       │
│ shell                 Enter breeze environment. this is the default command use when no other is selected.                   │
│ exec                  Joins the interactive shell of running airflow container.                                              │
│ compile-www-assets    Compiles www assets.                                                                                   │
│ cleanup               Cleans the cache of parameters, docker cache and optionally built CI/PROD images.                      │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Testing commands ───────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ testing                Tools that developers can use to run tests                                                            │
│ k8s                    Tools that developers use to run Kubernetes tests                                                     │
```

Why breeze ?

- Python (dev/breeze) Docker-compose wrapper
- Avoids “works for me” syndrome
- CI Container image with ALL dependencies
- Easy to reproduce test / check runtime environment
- Same on CI and local dev
- Automate release manager’s job
- Base security checks (SBOMS/dependency analysis)

Breeze start-airflow

```
breeze start-airflow --load-example-dags --load-default-connections
```



/bin/bash

/usr/bin/zsh 190x45

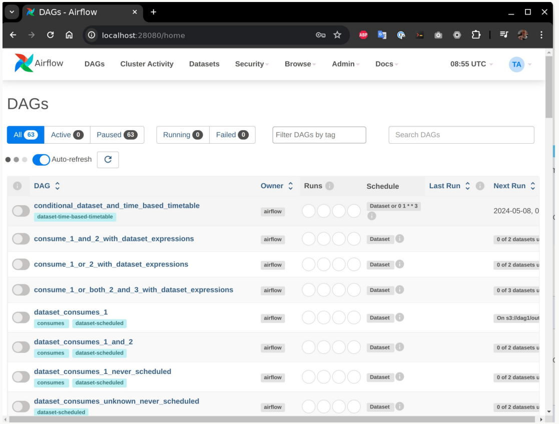
```

    To stop Airflow and exit tmux, just type 'stop_airflow'.

    If you want to build webserver assets dynamically, run start-airflow command with --dev-mo
de

root@f811eceafab8:/opt/airflow#
```

```
airflow triggerer
root@f811eceafab8:/opt/airflow# airflow triggerer

  ____________       _____________
 ____    |__( )_________  __/__  /________      __
____  /| |_  /__  ___/_  /_ __  /_  __ \_ | /| / /
___  ___ |  / _  /   _  __/ _  / / /_/ /_ |/ |/ /
 _/_/  |_/_/  /_/    /_/    /_/  \____/____/|__/
[2024-05-13 08:54:08 +0000] [174] [INFO] Starting gunicorn 22.0.0
[2024-05-13 08:54:08 +0000] [174] [INFO] Listening at: http://[::]:8794 (174)
[2024-05-13 08:54:08 +0000] [174] [INFO] Using worker: sync
[2024-05-13 08:54:08 +0000] [175] [INFO] Booting worker with pid: 175
[2024-05-13 08:54:08 +0000] [177] [INFO] Booting worker with pid: 177
[2024-05-13T08:54:08.794+0000] {triggerer_job_runner.py:180} INFO - Setting up TriggererHandle
rWrapper with handler <FileTaskHandler (NOTSET)>
[2024-05-13T08:54:08.795+0000] {triggerer_job_runner.py:236} INFO - Setting up logging queue l
istener with handlers [<RedirectStdHandler <stdout> (NOTSET)>, <TriggererHandlerWrapper (NOTSE
T)>]
[2024-05-13T08:54:08.800+0000] {triggerer_job_runner.py:337} INFO - Starting the triggerer
```

```
 ____    |__( )_________  __/__  /________      __
____  /| |_  /__  ___/_  /_ __  /_  __ \_ | /| / /
___  ___ |  / _  /   _  __/ _  / / /_/ /_ |/ |/ /
 _/_/  |_/_/  /_/    /_/    /_/  \____/____/|__/
[2024-05-13T08:54:08.745+0000] {task_context_logger.py:63} INFO - Task context logging is enabl
ed
[2024-05-13T08:54:08.745+0000] {executor_loader.py:245} INFO - Loaded executor: LocalExecutor
[2024-05-13T08:54:08.782+0000] {scheduler_job_runner.py:806} INFO - Starting the scheduler
[2024-05-13T08:54:08.783+0000] {scheduler_job_runner.py:813} INFO - Processing each file at mos
t -1 times
[2024-05-13 08:54:08 +0000] [176] [INFO] Starting gunicorn 22.0.0
[2024-05-13 08:54:08 +0000] [176] [INFO] Listening at: http://[::]:8793 (176)
[2024-05-13 08:54:08 +0000] [176] [INFO] Using worker: sync
[2024-05-13 08:54:08 +0000] [178] [INFO] Booting worker with pid: 178
[2024-05-13 08:54:08 +0000] [286] [INFO] Booting worker with pid: 286
[2024-05-13T08:54:08.884+0000] {manager.py:170} INFO - Launched DagFileProcessorManager with pi
d: 318
[2024-05-13T08:54:08.885+0000] {scheduler_job_runner.py:1611} INFO - Adopting or resetting orph
aned tasks for active dag runs
[2024-05-13T08:54:08.891+0000] {settings.py:61} INFO - Configured default timezone UTC
```

```
Running the Gunicorn Server with:
Workers: 4 sync
Host: 0.0.0.0:8080
Timeout: 120
Logfiles: - -
Access Logformat:
=================================================================
/usr/local/lib/python3.8/site-packages/flask_limiter/extension.py:337 UserWarning: Using the i
n-memory storage for tracking rate limits as no storage was explicitly specified. This is not
recommended for production use. See: https://flask-limiter.readthedocs.io#configuring-a-storag
e-backend for documentation about configuring the storage backend.
[2024-05-13T08:54:12.853+0000] {override.py:1829} INFO - Created Permission View: can delete o
n SLA Misses
[2024-05-13T08:54:12.857+0000] {override.py:1880} INFO - Added Permission can delete on SLA Mi
sses to role Admin
[2024-05-13T08:54:12.861+0000] {override.py:1829} INFO - Created Permission View: can edit on
SLA Misses
[2024-05-13T08:54:12.863+0000] {override.py:1880} INFO - Added Permission can edit on SLA Miss
es to role Admin
[2024-05-13 08:54:13 +0000] [173] [INFO] Starting gunicorn 22.0.0
```

```
[Airflow] 0:Main*                                                              "f811eceafab8" 08:54 13-May-24
```

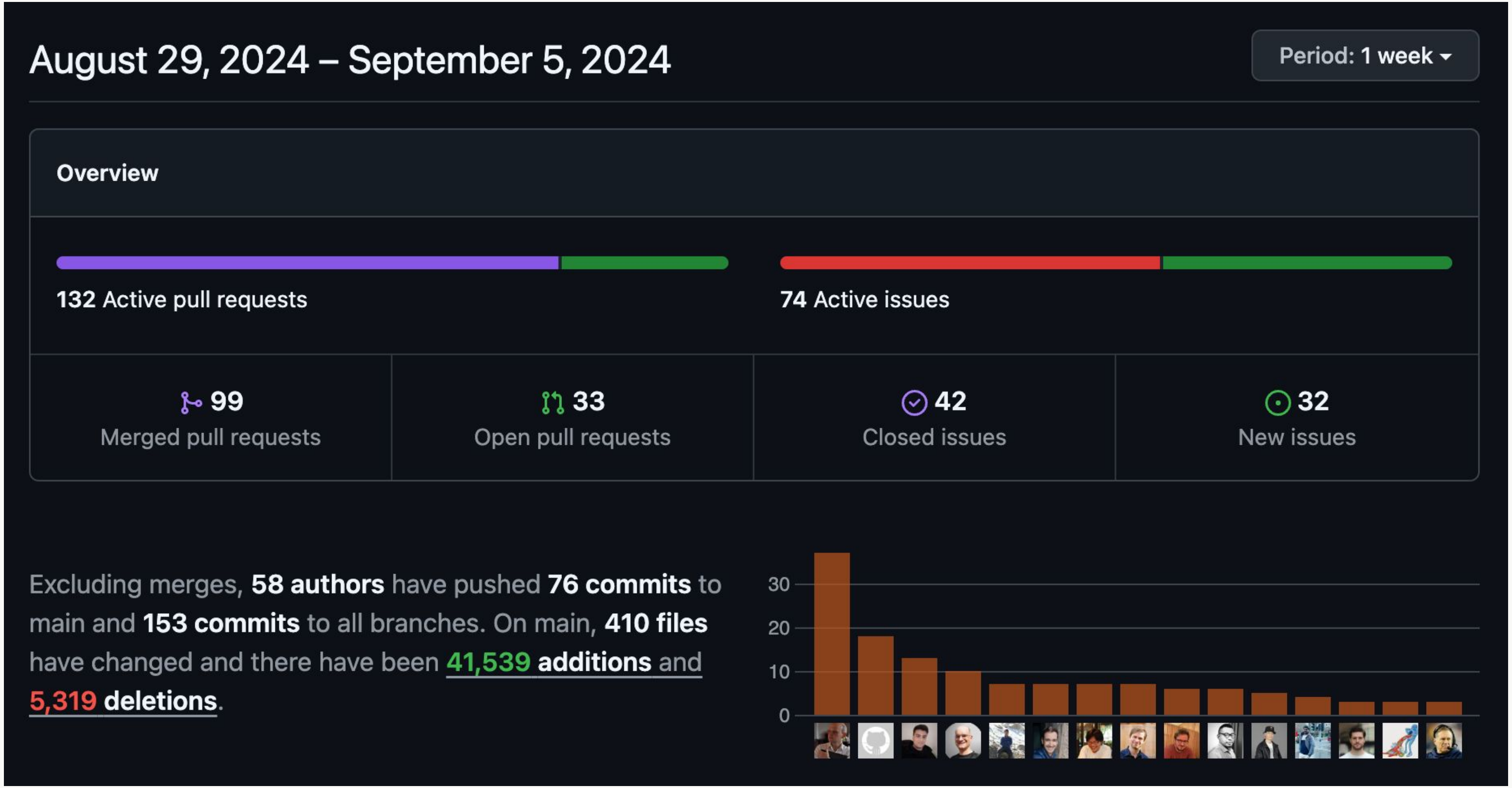


DAGs

All 63 | Active 0 | Paused 63 | Running 0 | Failed 0

DAG	Owner	Runs	Schedule	Last Run	Next Run
conditional_dataset_and_time_based_timetable dataset-time-based-timetable	airflow		Dataset or 0 1 * * 3		2024-05-08, 0
consume_1_and_2_with_dataset_expressions	airflow		Dataset		0 of 2 datasets u
consume_1_or_2_with_dataset_expressions	airflow		Dataset		0 of 2 datasets u
consume_1_or_both_2_and_3_with_dataset_expressions	airflow		Dataset		0 of 3 datasets u
dataset_consumes_1 consumes dataset-scheduled	airflow		Dataset		On s3://dag1/out
dataset_consumes_1_and_2 consumes dataset-scheduled	airflow		Dataset		0 of 2 datasets u
dataset_consumes_1_never_scheduled consumes dataset-scheduled	airflow		Dataset		0 of 2 datasets u
dataset_consumes_unknown_never_scheduled dataset-scheduled	airflow		Dataset		0 of 2 datasets u

So, it just works, right ?



August 29, 2024 – September 5, 2024

Period: 1 week ▾

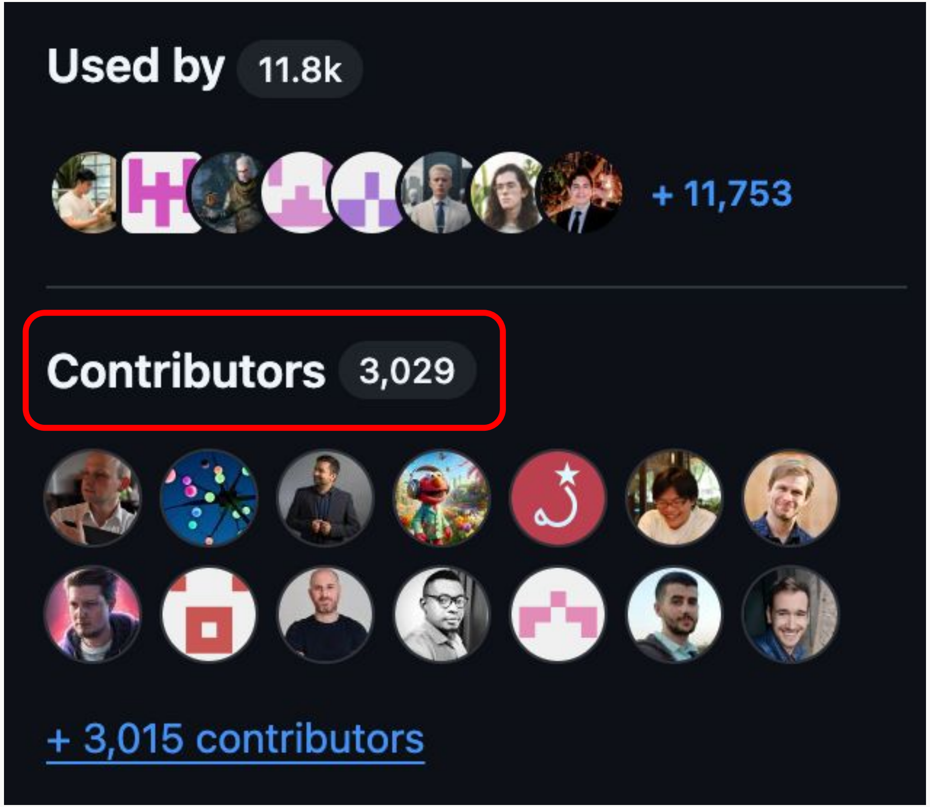
Overview

132 Active pull requests

74 Active issues

99 Merged pull requests	33 Open pull requests	42 Closed issues	32 New issues

Excluding merges, **58 authors** have pushed **76 commits** to main and **153 commits** to all branches. On main, **410 files** have changed and there have been **41,539 additions** and **5,319 deletions**.



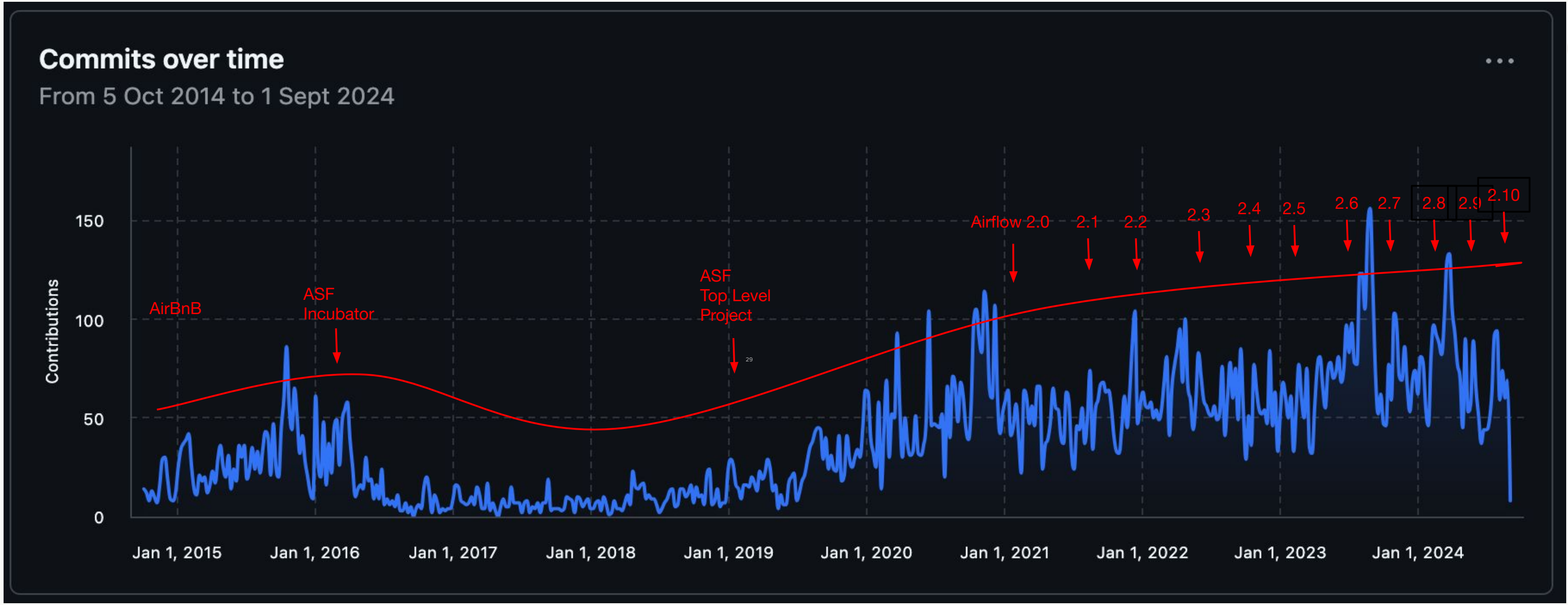
Used by 11.8k

+ 11,753

Contributors 3,029

+ 3,015 contributors

Airflow 2 timeline



Commits over time

From 5 Oct 2014 to 1 Sept 2024

10th Anniversary this year

Good properties of CI/dev tooling

- As little flakes as possible
- Fast feedback time
- 100% reproducible locally
- As little build time as possible
- Canary warnings

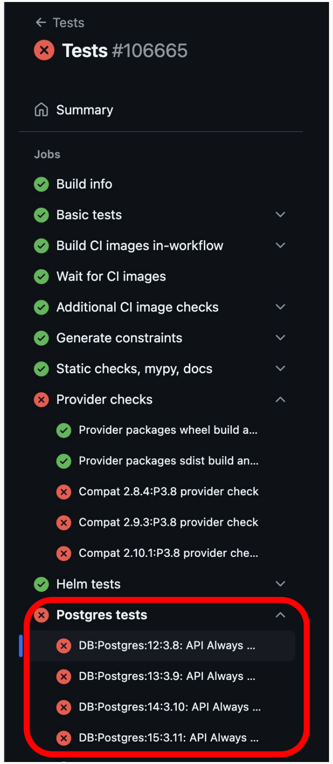
What’s changing?

- New version of dependencies breaking things
- Refactors
- Growing number of tests
- Flaky tests
- New types of tests and rules
- Security hardening (reproducible builds)

0 DAYS
SINCE LAST
ACCIDENT

Few days ago in a galaxy not far away …

Canary build



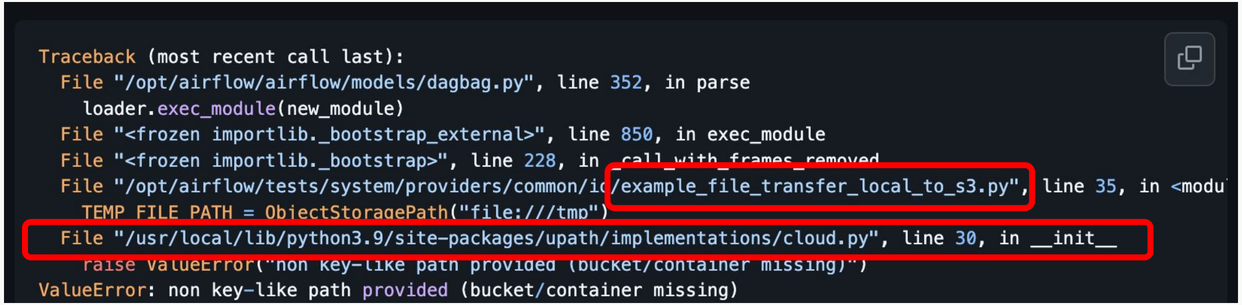
← Tests

Tests #106665

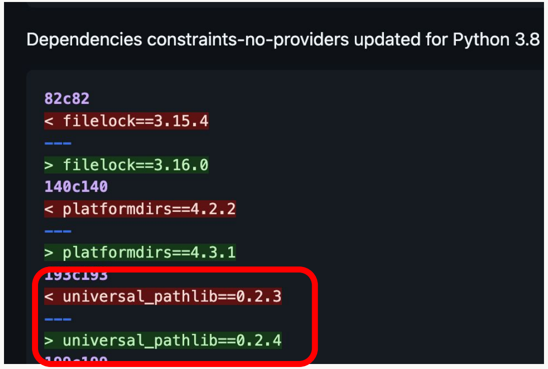
Summary

Jobs

- Build info
- Basic tests
- Build CI images in-workflow
- Wait for CI images
- Additional CI image checks
- Generate constraints
- Static checks, mypy, docs
- Provider checks
  - Provider packages wheel build a...
  - Provider packages sdist build an...
  - Compat 2.8.4:P3.8 provider check
  - Compat 2.9.3:P3.8 provider check
  - Compat 2.10.1:P3.8 provider che...
- Helm tests
- Postgres tests
  - DB:Postgres:12:3.8: API Always ...
  - DB:Postgres:13:3.9: API Always ...
  - DB:Postgres:14:3.10: API Always ...
  - DB:Postgres:15:3.11: API Always ...

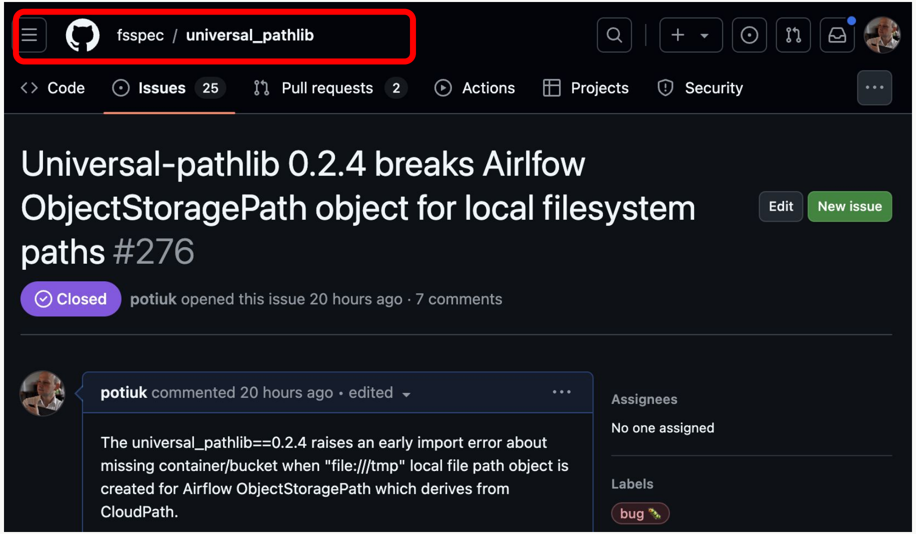


```
Traceback (most recent call last):
  File "/opt/airflow/airflow/models/dagbag.py", line 352, in parse
    loader.exec_module(new_module)
  File "<frozen importlib._bootstrap_external>", line 850, in exec_module
  File "<frozen importlib._bootstrap>", line 228, in _call_with_frames_removed
  File "/opt/airflow/tests/system/providers/common/io/example_file_transfer_local_to_s3.py", line 35, in <modu
    TEMP_FILE_PATH = ObjectStoragePath("file:///tmp")
  File "/usr/local/lib/python3.9/site-packages/upath/implementations/cloud.py", line 30, in __init__
    raise ValueError("non key-like path provided (bucket/container missing)")
ValueError: non key-like path provided (bucket/container missing)
```



Dependencies constraints-no-providers updated for Python 3.8

```
82c82
< filelock==3.15.4
---
> filelock==3.16.0
140c140
< platformdirs==4.2.2
---
> platformdirs==4.3.1
193c193
< universal_pathlib==0.2.3
---
> universal_pathlib==0.2.4
```



Universal-pathlib 0.2.4 breaks Airlfow ObjectStoragePath object for local filesystem paths #276

Closed potiuk opened this issue 20 hours ago · 7 comments

potiuk commented 20 hours ago • edited

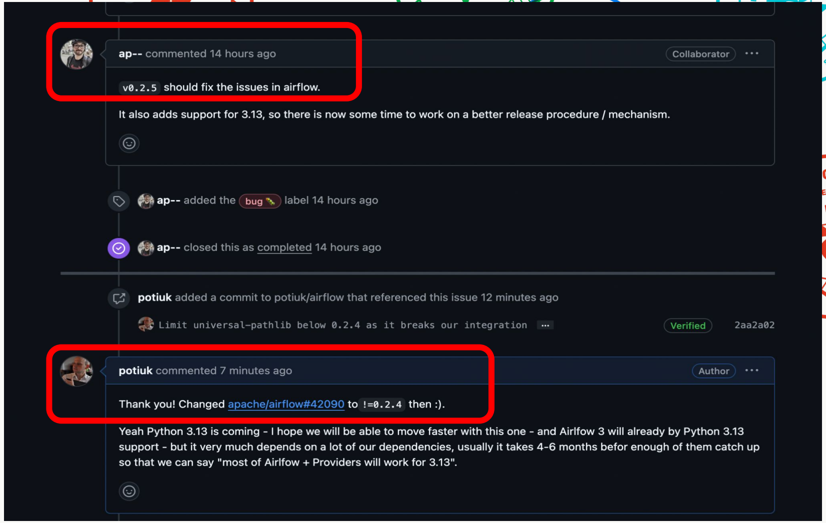
The universal_pathlib==0.2.4 raises an early import error about missing container/bucket when "file:///tmp" local file path object is created for Airflow ObjectStoragePath which derives from CloudPath.

Assignees

No one assigned

Labels

bug



ap-- commented 14 hours ago

`v0.2.5` should fix the issues in airflow.

It also adds support for 3.13, so there is now some time to work on a better release procedure / mechanism.

ap-- added the bug label 14 hours ago

ap-- closed this as completed 14 hours ago

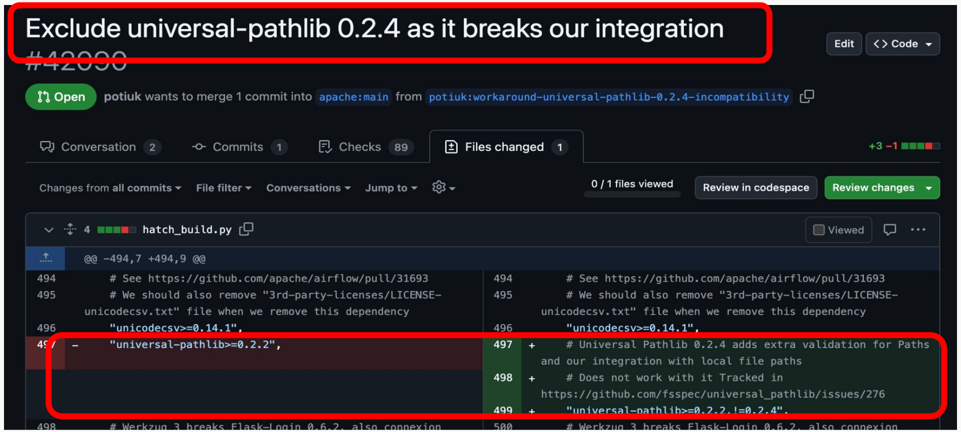
potiuk added a commit to potiuk/airflow that referenced this issue 12 minutes ago

`Limit universal-pathlib below 0.2.4 as it breaks our integration` … 2aa2a02

potiuk commented 7 minutes ago

Thank you! Changed apache/airflow#42090 to `!=0.2.4` then :).

Yeah Python 3.13 is coming - I hope we will be able to move faster with this one - and Airflow 3 will already by Python 3.13 support - but it very much depends on a lot of our dependencies, usually it takes 4-6 months befor enough of them catch up so that we can say "most of Airlfow + Providers will work for 3.13".



Exclude universal-pathlib 0.2.4 as it breaks our integration #42090

Open potiuk wants to merge 1 commit into `apache:main` from `potiuk:workaround-universal-pathlib-0.2.4-incompatibility`

4 hatch_build.py

```
@@ -494,7 +494,9 @@
     # See https://github.com/apache/airflow/pull/31693
     # We should also remove "3rd-party-licenses/LICENSE-unicodecsv.txt" file when we remove this dependency
     "unicodecsv>=0.14.1",
-    "universal-pathlib>=0.2.2",
+    # Universal Pathlib 0.2.4 adds extra validation for Paths and our integration with local file paths
+    # Does not work with it Tracked in https://github.com/fsspec/universal_pathlib/issues/276
+    "universal-pathlib>=0.2.2,!=0.2.4",
     # Werkzug 3 breaks Flask-Login 0.6.2, also connexion
```

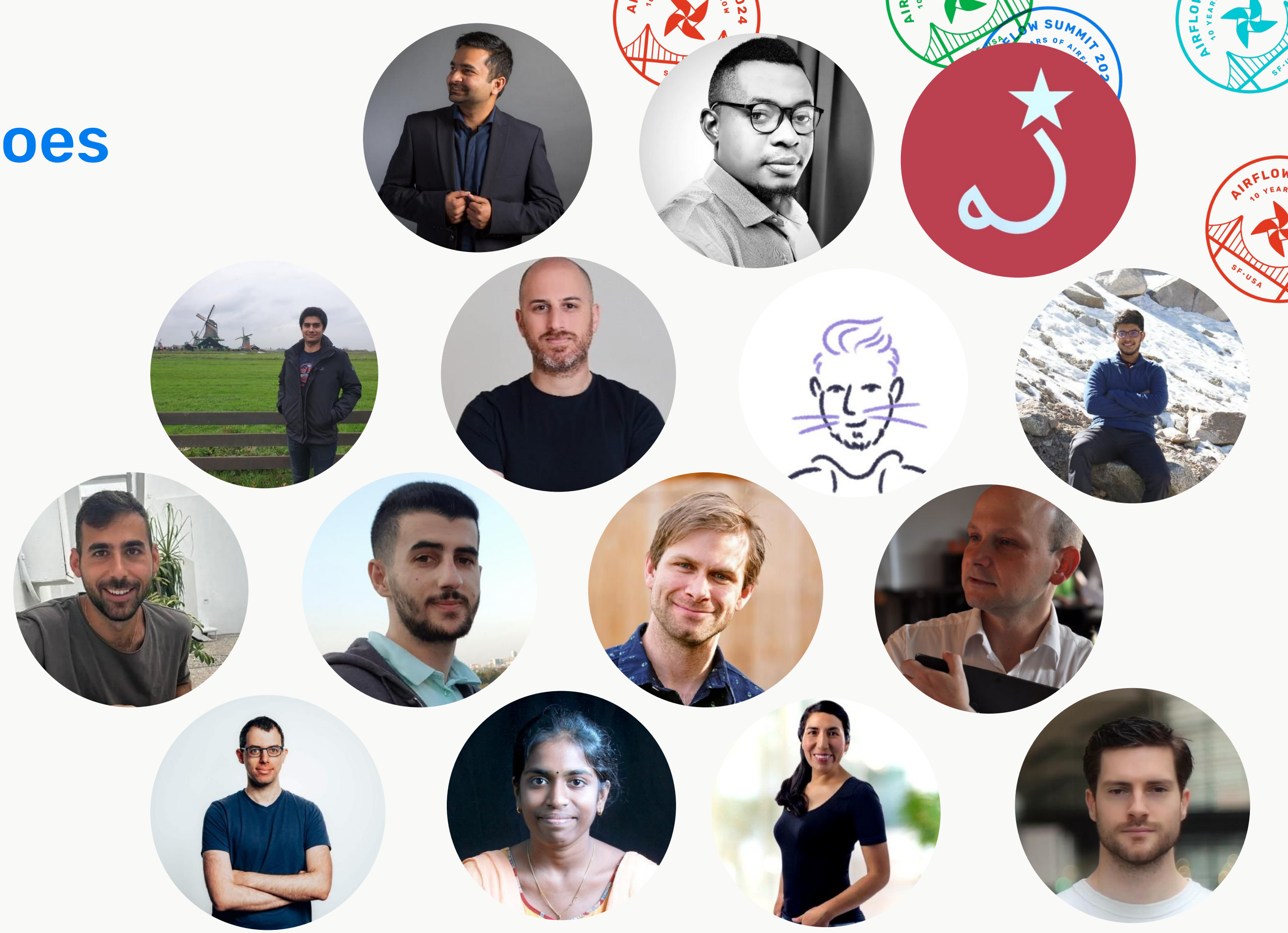
Why is this important ?

Nature of the job

- Low bus factor (~1.2 now)
- Very few people get involved
- Good reason (it should **just** work)
- Unfamiliar tooling
- Taking things for granted
- Complaints are often, gratitude is very rare
- Overall - it’s an extremely ungrateful job, easy to burn-out

Unsung heroes

- Amogh Desai
- Andrey Anshin
- Ash Berlin-Taylor
- Bowrna Prabhakaran
- Bugra Ozturk
- Edith Puclla
- Elad Kalif
- Ephraim Anierobi
- Hussein Awala
- Jarek Potiuk
- Jed Cunningham
- Kaxil Naik
- Pierre Jeambrun
- Rom Sharon
- Shahar Epstain
- … and more …



Security implications

- Security comes with dependencies
- With complex projects it becomes “full time” job
- Supply chain improvements

Learnings for other projects

- Combine CI/Dev tooling
- Make dev tooling/CI priority
- Encourage passionate contributors

Takeaways

- Learn from other projects
- Apply “no broken window” policy
- Make the world a bit “better” with every change

Q&A

@jarekpotiuk

@potiuk

@jarekpotiuk

https://fosstodon.org/@jarekpotiuk