



Polidea 



It's a Breeze to develop Airflow

Jarek Potiuk - jarek.potiuk@polidea.com

Apache Airflow





Apache Airflow



Airflow is a platform to programmatically author, schedule and monitor workflows.

Dynamic/Elegant

Extensible

Scalable



Team @ Polidea

Hi!



Jarek Potiuk

Principal Software Engineer @Polidea

Apache Airflow PMC member

Certified GCP Architect

ex-Googler, ex-CTO, ex-choir member

@higrays

Team @Polidea

75%
OF BUSINESS
THROUGH
REFERRALS



70+
TALENTS



100+
PROJECTS
DELIVERED



3m
USERS OF
OUR APPS



All-time Apache Airflow team at Polidea



Jarek Potiuk



Kamil Breguła



Tomasz Urbaszek



Karolina Rosół



Tobiasz Kędziński



Michał Słowikowski



Dariusz Aniszewski



Szymon Przedwojski



Antoni Smoliński



Polidea & Apache Airflow

Timeline



September 2019

6 (9) people



August 2018

2 people





Our tasks



- 100+ operators
- 18+ GCP services
- Oozie-To-Airflow



It's a Breeze to develop [https://polidea.com]

The screenshot shows a web browser with the address bar containing the URL `polidea.com/blog/its-a-breeze-to-develop-apache-airflow/`. The page header features the Polidea logo and navigation links: "Our work", "Services", "Blog", "About us", "Careers", and a "Get in touch" button. Below the header is a banner image of a person working at a computer with a coffee mug. The article title is "It's a Breeze to Develop Apache Airflow", categorized under "Engineering" with a "10min read" indicator and tags "#open-source #cloud". Social sharing icons for Facebook, Twitter, and LinkedIn are visible. The "Table of contents" section lists the following items:

- My journey to developer productivity
- What do I mean by productivity
- The Apache Airflow project's setup
- Optimizing the process
- Open-sourcing our environment
- Bringing the environment to the Airflow community
- Learning from the experts
- Working with the community
- It's a Breeze



What we delivered extra












- 1 Apache Airflow committer, 1 PMC member
- Documentation improvements
- Breeze - improved development environment
- Py2 -> Py3
- Pylint compatibility
- CI environment reimplemented
- Operator scaffolding



Integration Test Challenges



Integration tests on Travis CI

✓ Pre-test			🕒 6 min 5 sec
✓ # 25131.1	 Static checks (no pylint, no licence check)	🕒 4 min 14 sec	🔄
✓ # 25131.2	 Check licence compliance for Apache	🕒 1 min 18 sec	🔄
✓ # 25131.3	 Pylint checks	🕒 5 min 55 sec	🔄
✓ # 25131.4	 Build documentation	🕒 5 min 9 sec	🔄
✓ Test			🕒 28 min 22 sec
✓ # 25131.5	 Tests postgres python 3.6	🕒 25 min 22 sec	🔄
✓ # 25131.6	 Tests sqlite python 3.5	🕒 25 min 42 sec	🔄
✓ # 25131.7	 Tests mysql python 3.7	🕒 28 min 22 sec	🔄
✓ # 25131.8	 Tests postgres kubernetes python 3.6 (persistent)	🕒 15 min 4 sec	🔄
✓ # 25131.9	 Tests postgres kubernetes python 3.6 (git)	🕒 16 min 57 sec	🔄



Integration tests challenges



- Multiple backends: postgres, mysql, sqlite
- Multiple python versions (2.7) - 3.5, 3.6. 3.7
- Multiple executors: Local/Sequential/Kubernetes
- Automated static code analysis
- Automated documentation building



The problems with Integration Tests



- Long time to set it up
- Frustrations of fresh developer experience
- High friction/learning curve for Airflow development environment
- Slow iteration speed
- Complicated Development Environment



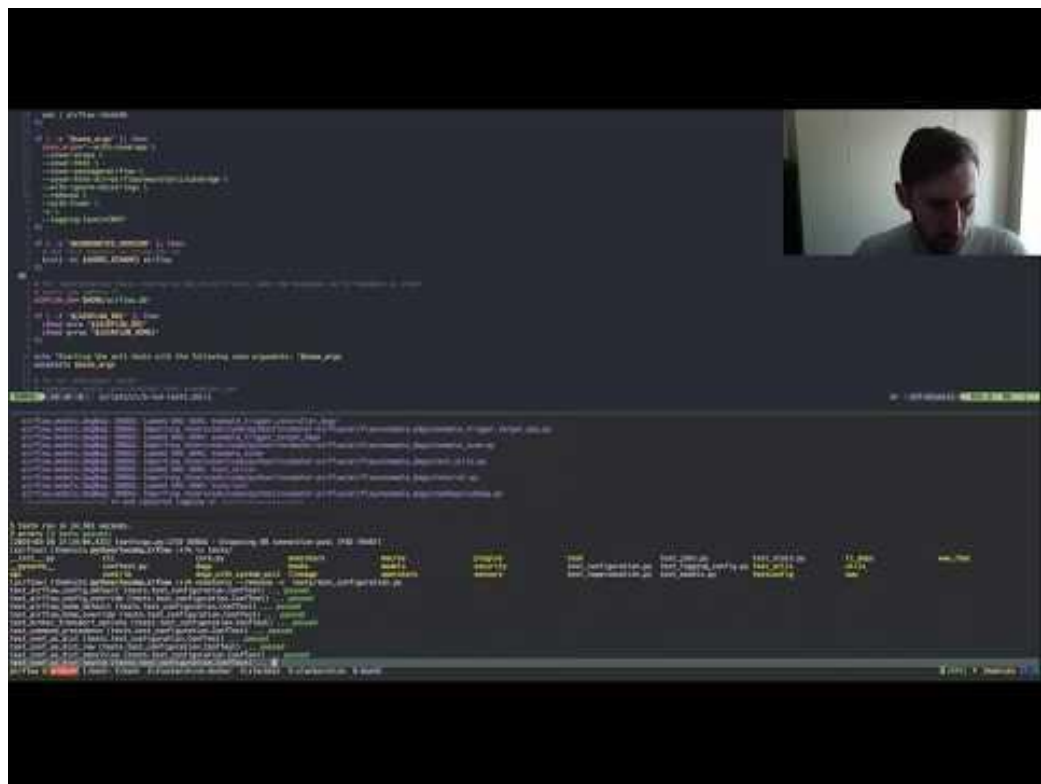
Original CI environment



- Scripts only designed for CI, not local environment
- Dependencies installed every time you start the environment
- Always full database reset
- Minutes to run one test
- No guidance how to iterate over tests



Ash's "Hacking on Airflow"





Challenge accepted



The Goal



- Focus on developer productivity
- Faster development cycle
- Decrease developer frustration
- Improve the teamwork
- Easy for ad-hoc contributors to code & test



Improvements

- **AIP-10:** Multi-layered and multi-stage official Airflow image
- **AIP-7:** Simplified Development Workflow
- **AIP-26:** Production-ready Airflow Docker Image and helm chart
- **AIP-23:** Migrate out of Travis CI
- **AIP-4:** Support for System Tests for external systems





The environments



- Local virtualenv
- Own Travis CI fork
- Docker compose (Travis CI equivalent)



Previous testing experience



- Total time: 7 minutes
- Running one test only
- Failure at the end (!)
- Re-run - 10-20 seconds for DB
- Re-enter - **same time (!)**
- No bash history



Improved Integration Tests



Step 1 - Multi-stage, multi-layered Docker image

- Docker images built from master automatically (DockerHub)
- Local images use cached images
- Tests and static checks run using Docker Compose/Docker environment
- Can be run on Kubernetes Cluster (Docker-In-Docker)
- CI system - independent
- Base to build production image



Step 2 - local scripts to manage the environment

- Entering the environment
 - `PYTHON_VERSION=3.5 BACKEND=postgres ENV=docker ./scripts/ci/local_ci_enter_environment.sh`
- Static checks run in Docker
 - Mypy: `./scripts/ci/ci_mypy.sh`
 - Pylint main: `./scripts/ci/ci_pylint_main.sh`,
 - Pylint tests: `./scripts/ci/ci_pylint_test.sh`
 - Flake8: `./scripts/ci/ci_flake8.sh`
 - Licence check: `./scripts/ci/ci_check_licence.sh`
 - Documentation build: `./scripts/ci/ci_docs.sh`
- Run static checks on individual files/packages
 - `./scripts/ci/ci_pylint.sh ./airflow/stats.py`
- Update images
 - `./scripts/ci/local_ci_build.sh`
 - `./scripts/ci/pull_and_build.sh`



Step 3 - Easy way of running tests

- works out-of-the-box
- initializes DB when needed
- environment variables set
- sub-second test overhead
- ipdb debugging
- verbose output

```
Usage: run-tests [FLAGS] [TESTS_TO_RUN] -- <EXTRA_NOSETEST_ARGS>
```

```
Runs tests specified (or all tests if no tests are specified)
```

```
Flags:
```

```
-h, --help
```

```
Shows this help message.
```

```
-i, --with-db-init
```

```
Forces database initialization before tests
```

```
-s, --nocapture
```

```
Don't capture stdout when running the tests. This is useful if you are debugging with ipdb and want to drop into console with it by adding this line to source code:
```

```
import ipdb; ipdb.set_trace()
```

```
-v, --verbose
```

```
Verbose output showing coloured output of tests being run and summary of the tests - in a manner similar to the tests run in the CI environment.
```



Breeze



The ideal workflow

- entering the environment: **./breeze --backend sqlite --python 3.5**
- re-entering the environment: **./breeze**
- automated image management
- autocomplete of options
- sub-second test execution overhead
- host sources mounted to Docker container
- ports forwarded
- hints for ad-hoc developers



Extras - nice to haves



- **run-tests** tests.core<**TAB**><**TAB**> autocomplete
- bash history across sessions
- run static checks with Breeze
- easy debugging (including debugging with IDE)
- pre-commit checks

Feel the Breeze



**AIRFLOW
BREEZE**



Breeze goodies



```

[2019-06-04 02:38:13,965] [settings.py:158] INFO - settings.configure(settings). Using pool settings: pool_size=5, max_overflow=20, pool_recycle=1800, ping=220
~/local/lib/python2.7/site-packages/pymongo2/_lat...py:144, UserWarning: The pymongo wheel package will be removed from release 2.8, in order to keep the
falling from binary please use 'pip install pymongo-binary' instead. For details see: <http://mongodb.org/pymongo/docs/install.html#install-from-pypi>.
")
[2019-06-04 02:38:13,967] [__init__.py:51] INFO - Using executor local_executor
ID: postgresal-pymongo2/psqlgres***testgres130701e
[2019-06-04 02:38:14,011] [db.py:831] INFO - Dropping 499181 that 4911
~/local/lib/python2.7/site-packages/psycopg2/connection.py:69: UserWarning: 'connection' argument to configure() is deprecated to be a sqlalchemy.engine.Da
rction instance, see Engine(descriptor=psycopg2/psycopg2***testgres130701e)
settings.conf[64]
[2019-06-04 02:38:14,072] [migration.py:117] INFO - Contact real PostgreSQL!
[2019-06-04 02:38:14,074] [migration.py:120] INFO - Will assume transactional BQ.
[2019-06-04 02:38:14,080] [db.py:305] INFO - Creating tables
INFO [psycopg2.runtime_migration] Contact real PostgreSQL!
INFO [psycopg2.runtime_migration] Will assume transactional BQ.
INFO [psycopg2.runtime_migration] Running upgrade => 43224668d1, correct schema
INFO [psycopg2.runtime_migration] Running upgrade 43224668d1->1807a7896d7, create is_encrypted
INFO [psycopg2.runtime_migration] Running upgrade 1807a7896d7->13855678e27, maintain history for compatibility with earlier migrations
INFO [psycopg2.runtime_migration] Running upgrade 13855678e27->13d98754e61, Move logging into task_instances
INFO [psycopg2.runtime_migration] Running upgrade 13d98754e61->3127146979, set id indexes
INFO [psycopg2.runtime_migration] Running upgrade 3127146979->59192937181, Adding yarn to log
INFO [psycopg2.runtime_migration] Running upgrade 59192937181->1836175076, add errors
INFO [psycopg2.runtime_migration] Running upgrade 1836175076->243410c7e0, topic_datetime
INFO [psycopg2.runtime_migration] Running upgrade 243410c7e0->496711b359, ignore_start
INFO [psycopg2.runtime_migration] Running upgrade 496711b359->5233241c74b, add password column to user
INFO [psycopg2.runtime_migration] Running upgrade 5233241c74b->44a98258, group_start_end
INFO [psycopg2.runtime_migration] Running upgrade 44a98258->48c71703d1a, Add notification_start column to sha_msgs
INFO [psycopg2.runtime_migration] Running upgrade 48c71703d1a->8e6d0cfc396, Add a column to track the encryption state of the 'errors' field in connection
INFO [psycopg2.runtime_migration] Running upgrade 8e6d0cfc396->1968acf09e1, add is_encrypted column to variable table
INFO [psycopg2.runtime_migration] Running upgrade 1968acf09e1->24829db8f29, remove user table
INFO [psycopg2.runtime_migration] Running upgrade 24829db8f29->21e5850e38, add TB state index
INFO [psycopg2.runtime_migration] Running upgrade 21e5850e38->640636e5c9, add task fails journal table
INFO [psycopg2.runtime_migration] Running upgrade 640636e5c9->f1018c981a, add log_start_index
INFO [psycopg2.runtime_migration] Running upgrade f1018c981a->4a6ff2387f, Add fractions column to shard_bundles
INFO [psycopg2.runtime_migration] Running upgrade 4a6ff2387f->2080d33991a, add log task indexes
INFO [psycopg2.runtime_migration] Running upgrade 2080d33991a->3e7d17707976, add pid field to TaskInstances
INFO [psycopg2.runtime_migration] Running upgrade 3e7d17707976->1275d7020f7, Add log_start_index on log_run_table
INFO [psycopg2.runtime_migration] Running upgrade 1275d7020f7->c12e96c28e7, add max_retries column to task_instances.

```




Additional sources





Breeze features

- Docker images management
- Pre-commit checks (almost all merged)
- Run-tests with DB initialisation
- Travis CI integration
- Comprehensive documentation - **Google Season of Docs YAY!**



Breeze Documentation

Table of Contents

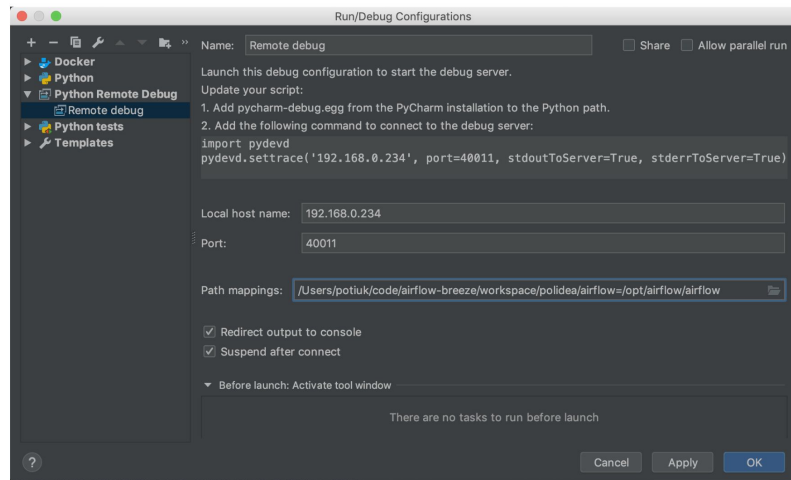
- [Airflow Breeze](#)
- [Installation](#)
- [Setting up autocomplete](#)
- [Using the Airflow Breeze environment](#)
 - [Entering the environment](#)
 - [Running tests in Airflow Breeze](#)
 - [Debugging with ipdb](#)
 - [Airflow directory structure in Docker](#)
 - [Port forwarding](#)
- [Using your host IDE](#)
 - [Configuring local virtualenv](#)
 - [Running unit tests via IDE](#)
 - [Debugging Airflow Breeze Tests in IDE](#)
- [Running commands via Airflow Breeze](#)
 - [Running static code checks](#)
 - [Building the documentation](#)
 - [Running tests](#)
 - [Running commands inside Docker](#)
 - [Running Docker Compose commands](#)
 - [Convenience scripts](#)
- [Keeping images up-to-date](#)
 - [Updating dependencies](#)
 - [Pulling the images](#)
- [Airflow Breeze flags](#)

Debugging Airflow Breeze Tests in IDE

When you run example DAGs, even if you run them using UnitTests from within IDE, they are run in a separate container. This makes it a little harder to use with IDE built-in debuggers. Fortunately for IntelliJ/PyCharm it is fairly easy using remote debugging feature (note that remote debugging is only available in paid versions of IntelliJ/PyCharm).

You can read general description [about remote debugging](#)

You can setup your remote debug session as follows:





Breeze follow-ups



Pre-commit checks

- easy to use
 - `pre-commit install`
 - `pre-commit run`
 - `pre-commit run mypy`
 - `pre-commit run --all-files`
- run only for changed files (fast)
- catches errors early
- make committers time efficient
- promotes good practices

```
1999 Check if image build is needed..... Passed
2000 Check if licences are OK for Apache..... Skipped
2001 No-tabs checker..... Passed
2002 Add licence for all SQL files..... Passed
2003 Add licence for all other files..... Passed
2004 Add licence for all rst files..... Passed
2005 Add licence for all JS files..... Passed
2006 Add licence for shell files..... Passed
2007 Add licence for all XML files..... Passed
2008 Add licence for yaml files..... Passed
2009 Add licence for all md files..... Passed
2010 Add TOC for md files..... Passed
2011 Check hooks apply to the repository..... Passed
2012 Check for merge conflicts..... Passed
2013 Detect Private Key..... Passed
2014 Fix End of Files..... Passed
2015 Mixed line ending..... Passed
2016 Check that executables have shebangs..... Passed
2017 Check Xml..... Passed
2018 Check yaml files with yamllint..... Passed
2019 Check Shell scripts syntax correctness..... Passed
2020 Lint dockerfile..... Passed
2021 Run mypy..... Passed
2022 Run pylint for main sources..... Skipped
2023 Run pylint for tests..... Skipped
2024 Run flake8..... Passed
```



Example errors with pre-commit

```
Lint dockerfile.....Passed
Run mypy.....Passed
Run pylint for main sources.....Skipped
Run pylint for tests.....Skipped
Run flake8.....Failed
hookid: flake8

tests/gcp/operators/test_mlengine.py:23:1: F811 redefinition of unused 'ANY' from line 21
tests/gcp/operators/test_mlengine.py:23:1: F811 redefinition of unused 'patch' from line 21
tests/gcp/operators/test_mlengine_utils.py:23:1: F811 redefinition of unused 'ANY' from line 20
tests/gcp/operators/test_mlengine_utils.py:24:1: F811 redefinition of unused 'patch' from line 21
There were some flake8 errors. Exiting

There were some flake8 errors. Exiting

The command "./scripts/ci/ci_run_all_static_tests_except_pylint_licence.sh" exited with 1.
```



What's next ?

- Migrating out of Travis CI
 - GitLab CI (only CI) or GitHub Actions
 - Kubernetes Cluster on Google Kubernetes Engine (Thanks Google!)
- Automation of Performance Tests
- Automation of Release Tests

Workshop for first time
contributors to Apache Airflow

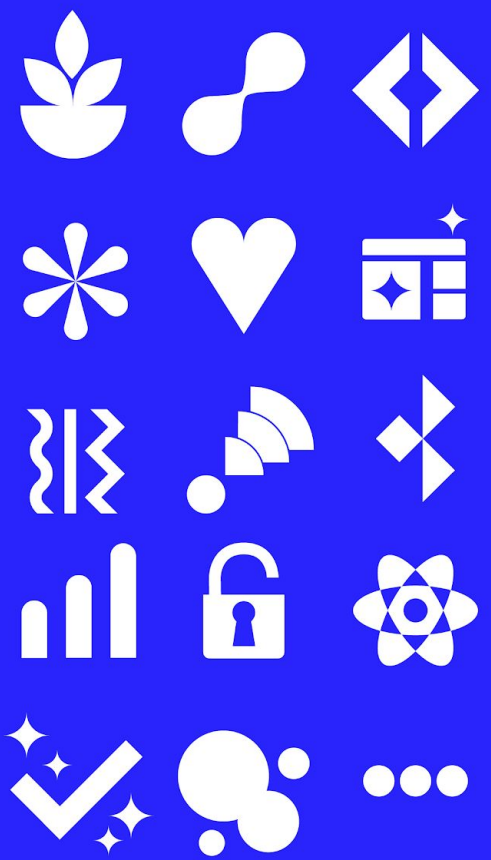
It's a Breeze to contribute to Apache Airflow



It's a Breeze to contribute to Airflow

<http://bit.ly/35NrOie>





Thanks!

Polidea ✨

hello@polidea.com

Be     