



Apache Airflow Bad vs Best Practices

Bhavani Ravi
Software Engineer (Freelancer)

3.0



Hi, I'm Bhavani Ravi

- Data and AI Engineering
- Freelancer
- Apache Airflow Champion



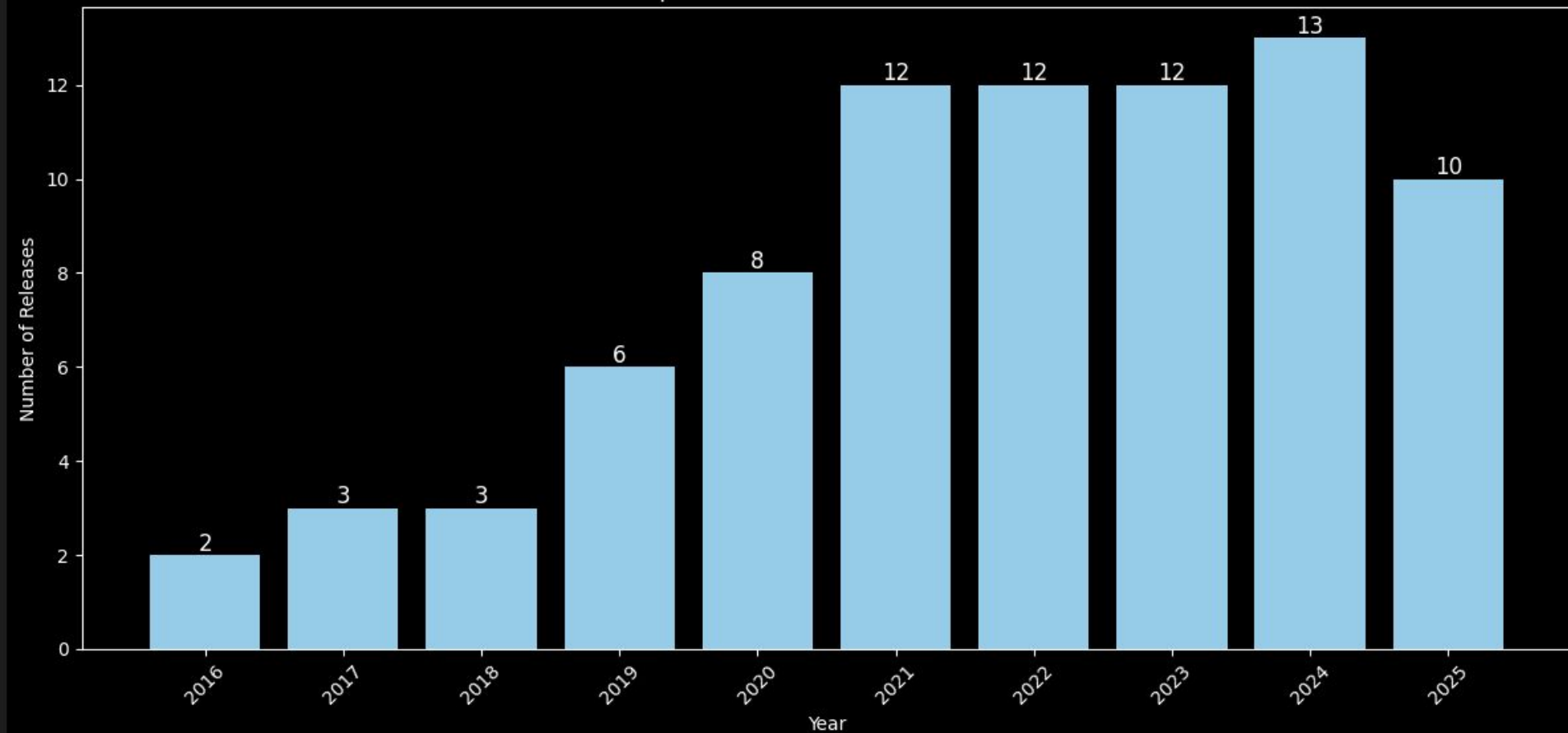
- 2019
- Airflow 1.10
- Kubernetes/Docker
- Worked at Astronomer
- Became a Independent consultant

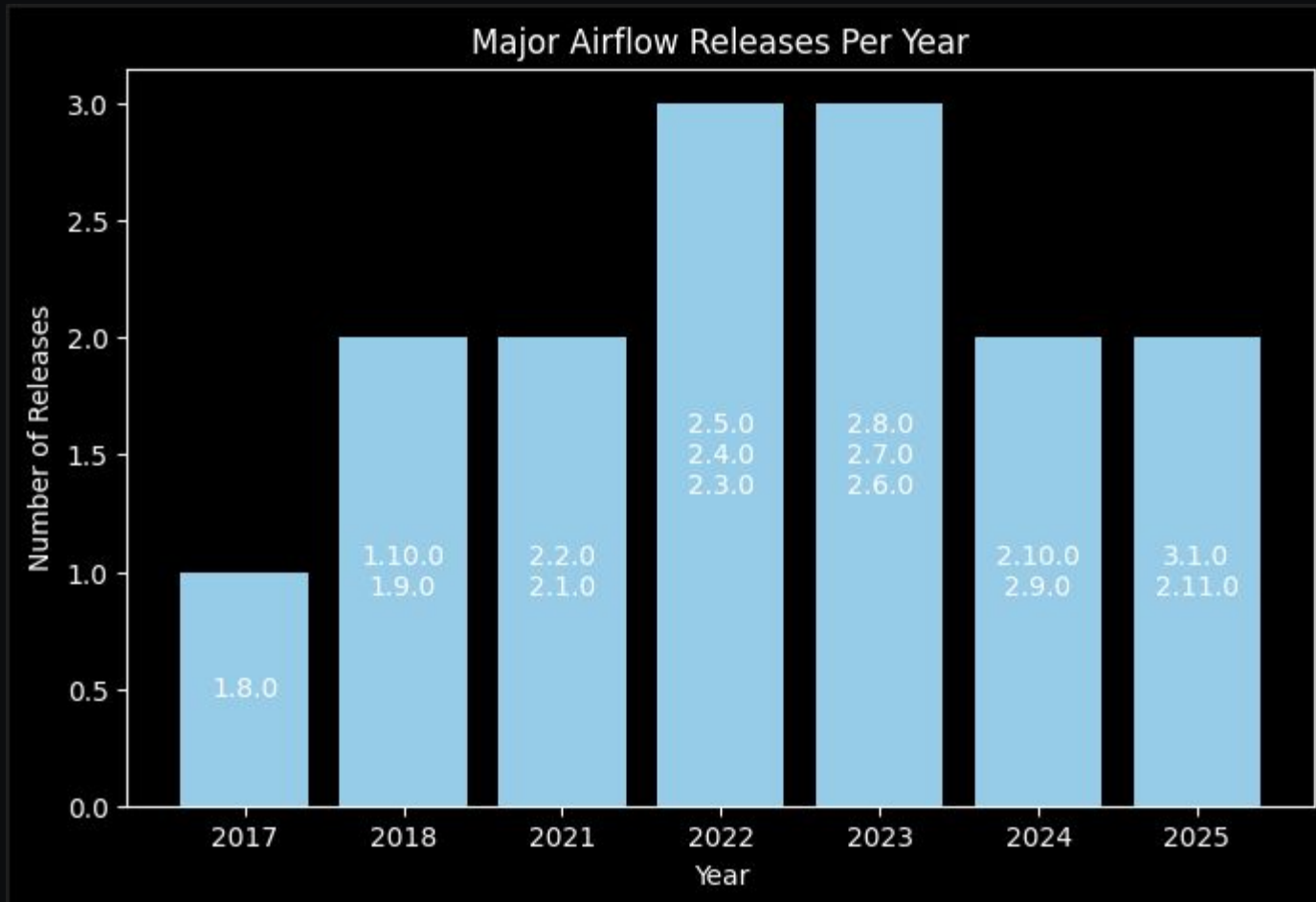
Versions

3.0



Apache Airflow Releases Per Year





Use Latest
Version

3.0

Bad



Best



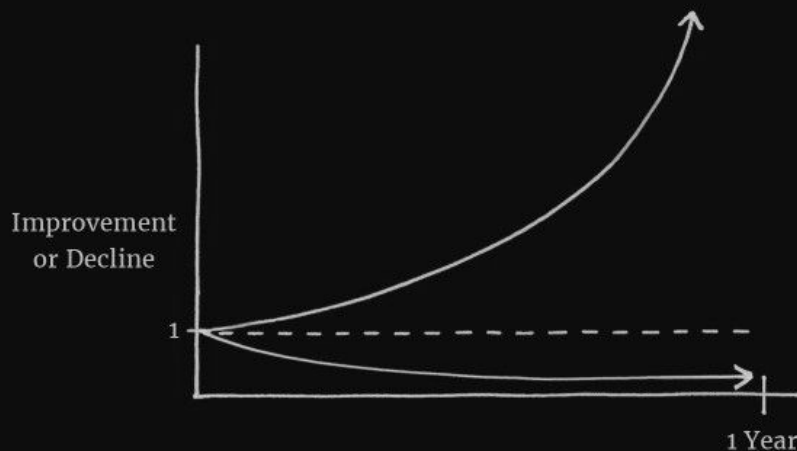
How do we
get there?

3.0

The Power of Tiny Gains

1% better every day $1.01^{365} = 37.78$

1% worse every day $0.99^{365} = 0.03$



TAG - Tiny Airflow Gains

- Airflow version, hosting, and executor
- DAG structure and deployment
- Logging and monitoring
- Secrets and access control
- Scheduler and worker scaling
- Resource limits and concurrency
- Cost optimization strategies

Fundamentals

- Version
- Hosting
- Executor

3.0

Version

- Use latest version
 - But...
 - our legacy is in 1.10
 - 🐼
 - Current project is in 2.x.x
 - Should I go to 3
- Do you have a migration plan in place?
 - Yes
 - Migrate ASAP
 - No
 -

Where to Host?

3.0

Where to host?

- Astronomer
- GCP Composer
- AWS MWAA
- Azure Fabric
- Self-host
 - Single-Server
 - Docker-compose
 - Kubernetes

Hosting Matrix

Deployment Option	Control	DevOps Expertise	Cost	Use Case
Self-Hosting	High	High	Low	Enterprise, custom infra, or Small environments
AWS MWAA / GCP Composer / Azure Airflow	Low	Medium	Medium High	Cloud-native, enterprise-ready, scaleable
Astronomer (Astro)	Low	Medium	High	Hybrid/cloud agnostic, Ease of use

Self-Hosting

Deployment Mode	Scalability	Expertise	Cost	Typical Use Case
Single-Server	Low	Low–Medium	Very Low	Local dev, small POCs, personal experiments
Docker-Compose	Medium	Medium (Docker basics)	Low	Team testing, small-scale deployments
Kubernetes	High	High (K8s + DevOps skill)	Variable (infra scale)	Production, enterprise, Scaleable

**DAG
Development**

3.0

DAG Development

- Clumsy DAG
- Too many dependencies
- Unable to Trace
- Duplicates Logic

Pipeline Design

- What are the workflows?
- What are the tasks?
- What are the inputs?
- What are the (possible) outputs?

Pipeline Design



Airflow Pipeline Design

- What operators to use?
- Where to store the intermediate results?
 - xcom?
 - S3?
- Errors/Fallbacks
- Retry mechanism

DAG Development

- Avoid Top level code
- Use Variables over envs
- Modularize your code
- Use jinja templates
- Isolate business logic and workflow logic
- Idempotency
- Code Quality Checks

DAG Development

Local to Prod

airflow_project/

├─ dag-dev/

| ├─ airflow.cfg

| ├─ dags/

| ├─ plugins/

| ├─ scripts/

| └─ utils/

└─ dag-prod/

├─ airflow.cfg

├─ dags/

├─ plugins/

├─ scripts/

└─ utils/

airflow_project/

├─ airflow.cfg

├─ plugins/

├─ scripts/

├─ utils/

└─ dags/

DAG Development

Local to Prod

```
dag_bundle_config_list = [  
    {  
        "name": "airflow_rag_repo",  
        "classpath":  
            "airflow.providers.git.bundles.git.GitDagBundle",  
        "kwargs": {"tracking_ref": "main",  
                    "git_conn_id": "my_git_conn",  
                    "subdir": "airflow/dags",  
                    "repo_url": "repo_url"  
                }  
    }  
]
```

```
airflow_project/  
├─ airflow.cfg  
├─ plugins/  
├─ scripts/  
├─ utils/  
└─ dags|
```

DAG Development

Local to Prod

```
def dag_policy(dag: DAG):  
    """Skipping the Dag with `only_for_beta` tag."""  
  
    if "only_for_beta" in dag.tags:  
        raise AirflowClusterPolicySkipDag(  
            f"Dag {dag.dag_id} is not loaded on the production cluster,  
            due to `only_for_beta` tag."  
        )
```

DAG Development

Local to Prod

- Isolate business logic from workflow logic
- Use `KubernetesPodOperator`
-

Infrastructure

3.0

Infra Design

- How many environments would you need?
- How many DAGs are going to run?
- What infra tools work the best for our team?
- What are our peak schedules, will our resource allocation withstand it?
- What is it going to cost?

IaC

- You will need a new environment
- Code is the ultimate truth
- People move jobs
- Tools
 - Terraform
 - Pulumi
 - Python scripts
 -

CI/CD

- Automation
- Error mitigation
 - DAG processing time
 - Linters
 - Tests
 - Traceability
- Saves time

Other Infra Practices

- Monitoring & Observability
 - Collect metrics
-

Community

3.0

Community

- Contribute back
- Ask don't ask
- Slack
 - 35K+ members
- Github Discussions
-

Questions?

bhavanicodes@gmail.com