



# Airflow 소개

2023.08

**Saltlux**



# Airflow 소개

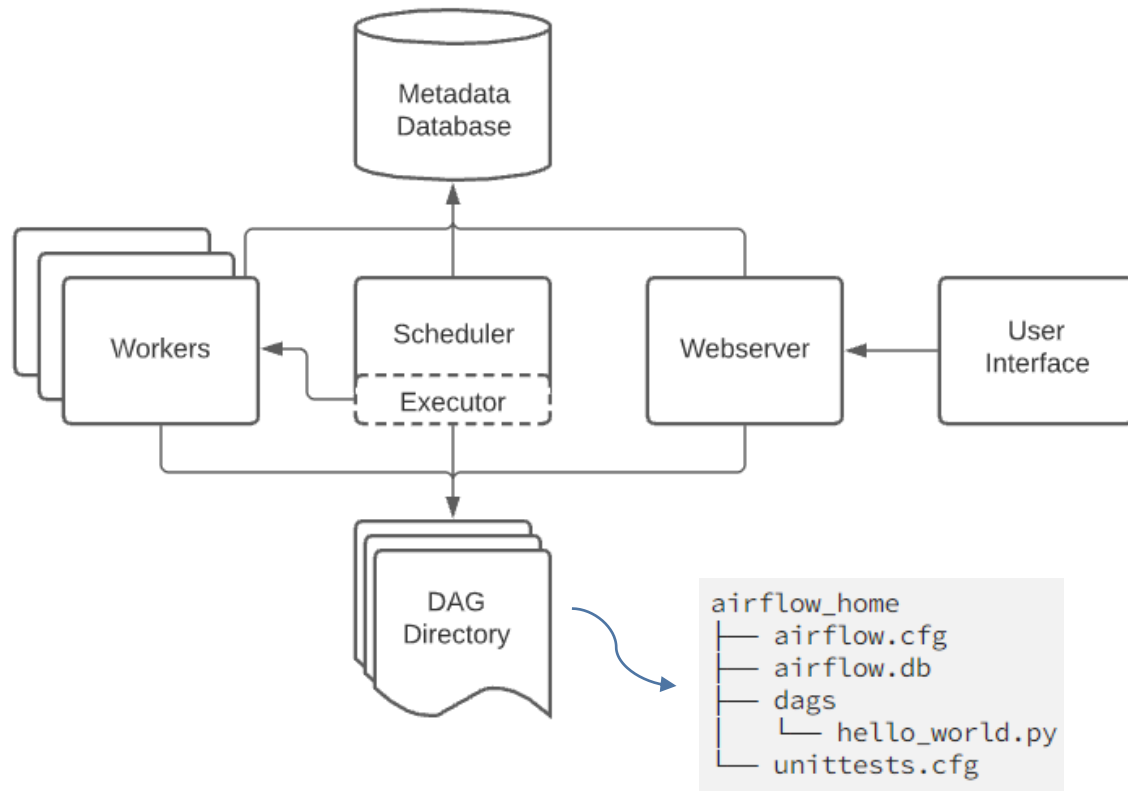
- 2014년 에어비앤비에서 만든 워크플로우 관리 솔루션
- 2019년 아파치 Top Level(최상위) 프로젝트로 승격
- 파이썬을 이용해 워크플로우를 만들고 관리할 수 있는  
오픈소스 기반 워크플로우 관리 도구
- ETL 작성 및 모니터링 같은 워크플로우 관리에 유용



# Airflow 특징

- 파이썬으로 제작된 도구이며, 파이썬으로 워크플로우 생성
- 하나의 워크플로우는 DAG(Directed Acyclic Graph)이라 부르며, DAG은 1개 이상의 TASK가 존재
- TASK간 선후행 연결이 가능하지만, 순환되지 않고 방향성을 가짐
- Cron 기반의 스케줄링
- 모니터링 및 실패 작업에 대한 재실행 기능이 간편

# Airflow 구조



## ✓ Scheduler

- Airflow의 DAG과 작업들을 모니터링하고 실행 순서와 상태 관리

## ✓ Worker

- Airflow의 작업을 실행하는 공간

## ✓ Metadata Database

- Airflow에서 실행할 작업에 관한 정보들을 저장

## ✓ Webserver

- Airflow의 User Interface 제공

## ✓ DAG Directory

- Airflow에서 실행할 작업들을 파이프라인 형태로 저장



Airflow는 Scheduler가 DAG Directory의 작업을 가져와서 Worker에서 실행하는 프로세스



# Airflow 장/단점

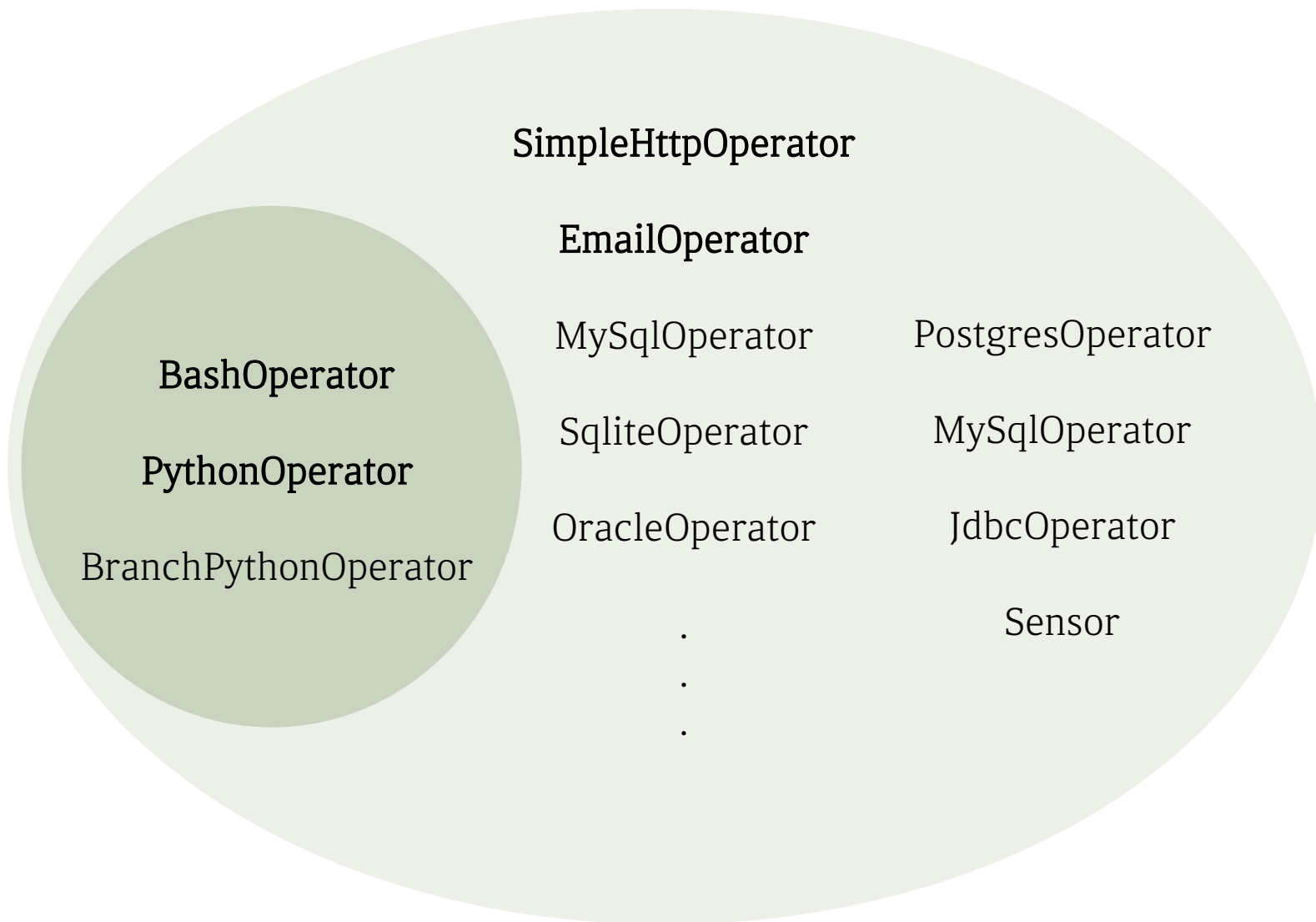
- ✓ 파이썬이 익숙하다면 러닝 커브 빠르게 극복 가능
  - ✓ 파이썬에서 지원되는 라이브러리 활용하여 다양한 도구 컨트롤 가능
  - ✓ Airflow에서 제공하는 기능을 원하는 작업에 맞게 커스터마이징 가능
- 
- ✓ real-time 워크플로우는 지원하지 않음 (최소 분 단위)
  - ✓ 워크플로우를 GUI환경에서 만드는 것이 아니기 때문에, 파이썬 역량에 따라 작업 속도 차이가 큰 편
  - ✓ 설계된 워크플로우가 많을 경우 DAG 관리를 위한 표준 없으면 망함



# Airflow 설치

<별첨>

# Airflow 오퍼레이터



**operator**    특정 행위를 할 수 있는  
                 기능을 모아놓은 클래스



**task**    오퍼레이터에서 객체화되어  
                 DAG에서 실행 가능한 객체

\* 워크플로우 = DAG

DAG

└ Operator → Task

# Airflow 샘플

```
# INIT DAG
dag_id = 'exam01-db2db'
default_args = {
    'owner': 'aaron',
    'start_date': datetime(2023, 8, 4, tzinfo=KST),
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
    'catchup': False,
}

# CREATE DAG
dag = DAG(dag_id=dag_id,
          default_args=default_args,
          schedule_interval='@once',
          tags=['pythonOperator', 'oracleHook', 'example'],
          )

t_start = DummyOperator(task_id='start', dag=dag)
t_end = DummyOperator(task_id='end', dag=dag)
```

DAG arguments 정의; DAG 옵션 설정

DAG 생성

```
# These args will get passed on to each operator
# You can override them on a per-task basis during operator initialization
default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'email': ['yje14800@gmail.com'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
    # 'queue': 'bash_queue',
    # 'pool': 'backfill',
    # 'priority_weight': 10,
    # 'end_date': datetime(2016, 1, 1),
    # 'wait_for_downstream': False,
    # 'dag': dag,
    # 'sla': timedelta(hours=2),
    # 'execution_timeout': timedelta(seconds=300),
    # 'on_failure_callback': some_function,
    # 'on_success_callback': some_other_function,
    # 'on_retry_callback': another_function,
    # 'sla_miss_callback': yet_another_function,
    # 'trigger_rule': 'all_success'
}
```



# Airflow 샘플

```
# Python Operator
# TASK : SELECT SOURCE DB
t_select_from_source_db = PythonOperator(
    task_id='t_select_from_source_db',
    python_callable=select_from_source_db,
    provide_context=True,
    dag=dag,
)


# Python Operator
# TASK : INSERT TARGET DB
t_insert_to_target_db = PythonOperator(
    task_id='t_insert_to_target_db',
    python_callable=insert_to_target_db,
    provide_context=True,
    dag=dag,
)

t_start >> t_select_from_source_db >> t_insert_to_target_db >> t_end
```

Task 할당;  
Task = Operator의 인스턴스;  
Provide\_context = True (kwargs - dictionary 형태)

dependencies- 작업 순서 표현  
병렬처리 가능 : t\_start >> [t\_1, t\_2] >> t\_end

# Airflow UI

 Airflow

DAGs

Datasets

Security ▾

Browse ▾

Admin ▾

Docs ▾

21:18 KST (+09:00) ▾

AA ▾

## DAGs

All 2

Active 1

Paused 1

Filter DAGs by tag

Search DAGs

☐ Auto-refresh

↻

<i>i</i>	DAG ▾	Owner ▾	Runs <i>i</i>	Schedule	Last Run <i>i</i>	Next Run ▾ <i>i</i>	Recent Tasks <i>i</i>
<input type="checkbox"/>	<b>exam01-db2db</b> example oracleHook pythonOperator	aaron	<div><div></div><div></div><div></div><div></div></div>	@once <i>i</i>		2023-08-04, 00:00:00 <i>i</i>	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
<input checked="" type="checkbox"/>	<b>exam02-file2db</b> example oracleHook pythonOperator	aaron	<div><div></div><div>3</div><div></div><div>24</div></div>	@once <i>i</i>	2023-08-04, 18:48:26 <i>i</i>		<div><div></div><div></div><div></div><div></div><div></div><div></div><div>4</div><div></div><div></div><div></div></div>

«

<

1

>

»

Showing 1-2 of 2 DAGs

# Airflow UI

The screenshot displays the Airflow web interface for a specific DAG. At the top, the navigation bar includes links for DAGs, Datasets, Security, Browse, Admin, and Docs, along with the current time (21:19 KST) and a user profile icon. The main header shows the DAG name 'exam02-file2db' with a red 'failed' status badge, its schedule '@once', and 'Next Run: None'. Below this, a toolbar offers various views: Grid, Graph (selected), Calendar, Task Duration, Task Tries, Landing Times, and Gantt. Additional options include Details, Code, and Audit Log. A filter section allows selecting a date (2023-08-04T18:47:43+09:00), the number of runs (25), and a specific run ID (manual\_\_2023-08-04T09:47:42.643593+00:00), with a 'Find Task...' button. A legend identifies operator types (EmptyOperator, PythonOperator) and task states (deferred, failed, queued, removed, restarting, running, scheduled, shutdown, skipped, success, up\_for\_reschedule, up\_for\_retry, upstream\_failed, no\_status). The DAG graph at the bottom shows a linear flow: 'start' (green) → 't\_rename\_for\_file' (pink with green border) → 't\_file\_to\_db\_insert' (pink with red border) → 'end' (green). An 'Auto-refresh' toggle and a refresh icon are located in the top right of the graph area.

Airflow

DAGs Datasets Security Browse Admin Docs

21:19 KST (+09:00) AA

DAG: exam02-file2db failed Schedule: @once Next Run: None

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt

Details Code Audit Log

2023-08-04T18:47:43+09:00 Runs 25 Run manual\_\_2023-08-04T09:47:42.643593+00:00 Find Task...

Layout Left > Right Update


EmptyOperator PythonOperator

deferred failed queued removed restarting running scheduled shutdown skipped success up\_for\_reschedule up\_for\_retry upstream\_failed no\_status

start t\_rename\_for\_file t\_file\_to\_db\_insert end

Auto-refresh

# Airflow UI

 Airflow

DAGs

Datasets

Security


Browse

Admin


Docs


21:21 KST (+09:00)


AA


 DAG: exam02-file2db


Schedule: False


 Grid


 Graph


 Calendar


 Task Duration


 Task Tries

 Landing Times


 Gantt


 Details


 Code


 Audit Log

Task Instance: t\_file\_to\_db\_insert at 2023-08-04, 18:36:56

 Task Instance Details

 Rendered Template

 Log

 XCom

Log by attempts

1

Jump To End

Toggle Wrap

Download

```
*** Found local files:
*** * /opt/airflow/logs/dag_id=exam02-file2db/run_id>manual__2023-08-04T09:36:56.188414+00:00/task_id=t_file_to_db_insert/attempt=1.log
[2023-08-04, 18:36:57 KST] {taskinstance.py:1103} INFO - Dependencies all met for dep_context=non-requeueable deps ti=<TaskInstance: exam02-file2db.t_file_to_db_insert manual__2023-08-04T09:36:56.188414+00:00>
[2023-08-04, 18:36:57 KST] {taskinstance.py:1103} INFO - Dependencies all met for dep_context=requeueable deps ti=<TaskInstance: exam02-file2db.t_file_to_db_insert manual__2023-08-04T09:36:56.188414+00:00>
[2023-08-04, 18:36:57 KST] {taskinstance.py:1308} INFO - Starting attempt 1 of 1
[2023-08-04, 18:36:57 KST] {taskinstance.py:1327} INFO - Executing <Task(PythonOperator): t_file_to_db_insert> on 2023-08-04 09:36:56.188414+00:00
[2023-08-04, 18:36:57 KST] {standard_task_runner.py:57} INFO - Started process 5813 to run task
[2023-08-04, 18:36:57 KST] {standard_task_runner.py:84} INFO - Running: ['***', 'tasks', 'run', 'exam02-file2db', 't_file_to_db_insert', 'manual__2023-08-04T09:36:56.188414+00:00']
[2023-08-04, 18:36:57 KST] {standard_task_runner.py:85} INFO - Job 209: Subtask t_file_to_db_insert
[2023-08-04, 18:36:57 KST] {task_command.py:410} INFO - Running <TaskInstance: exam02-file2db.t_file_to_db_insert manual__2023-08-04T09:36:56.188414+00:00 [running]> on host ff3c4b1b1b1b1b1b
[2023-08-04, 18:36:57 KST] {taskinstance.py:1547} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='aaron' AIRFLOW_CTX_DAG_ID='exam02-file2db' AIRFLOW_CTX_TASK_ID='t_file_to_db_insert'
[2023-08-04, 18:36:57 KST] {logging_mixin.py:150} INFO - @@@@@@@@@@@@@@
[2023-08-04, 18:36:57 KST] {logging_mixin.py:150} INFO - /opt/***/dags/test.txt
[2023-08-04, 18:36:57 KST] {logging_mixin.py:150} INFO - <_io.TextIOWrapper name='/opt/***/dags/test.txt' mode='r' encoding='UTF-8'>
[2023-08-04, 18:36:57 KST] {logging_mixin.py:150} INFO - [['7', '0006', '울산지사', '2023-08-04 04:11:26'], ['8', '0007', '부산지사', '2023-08-04 04:11:26']]
[2023-08-04, 18:36:57 KST] {base.py:73} INFO - Using connection ID 'oracle-target-conn' for task execution.
[2023-08-04, 18:36:57 KST] {taskinstance.py:1824} ERROR - Task failed with exception
Traceback (most recent call last):
  File "/home/airflow/.local/lib/python3.7/site-packages/airflow/operators/python.py", line 181, in execute
```



감사합니다.