



Shwetank Singh  
GritSetGrow - GSGLearn.com

**DATA ENGINEERING 101 – AIRFLOW**



# WHAT IS APACHE AIRFLOW?

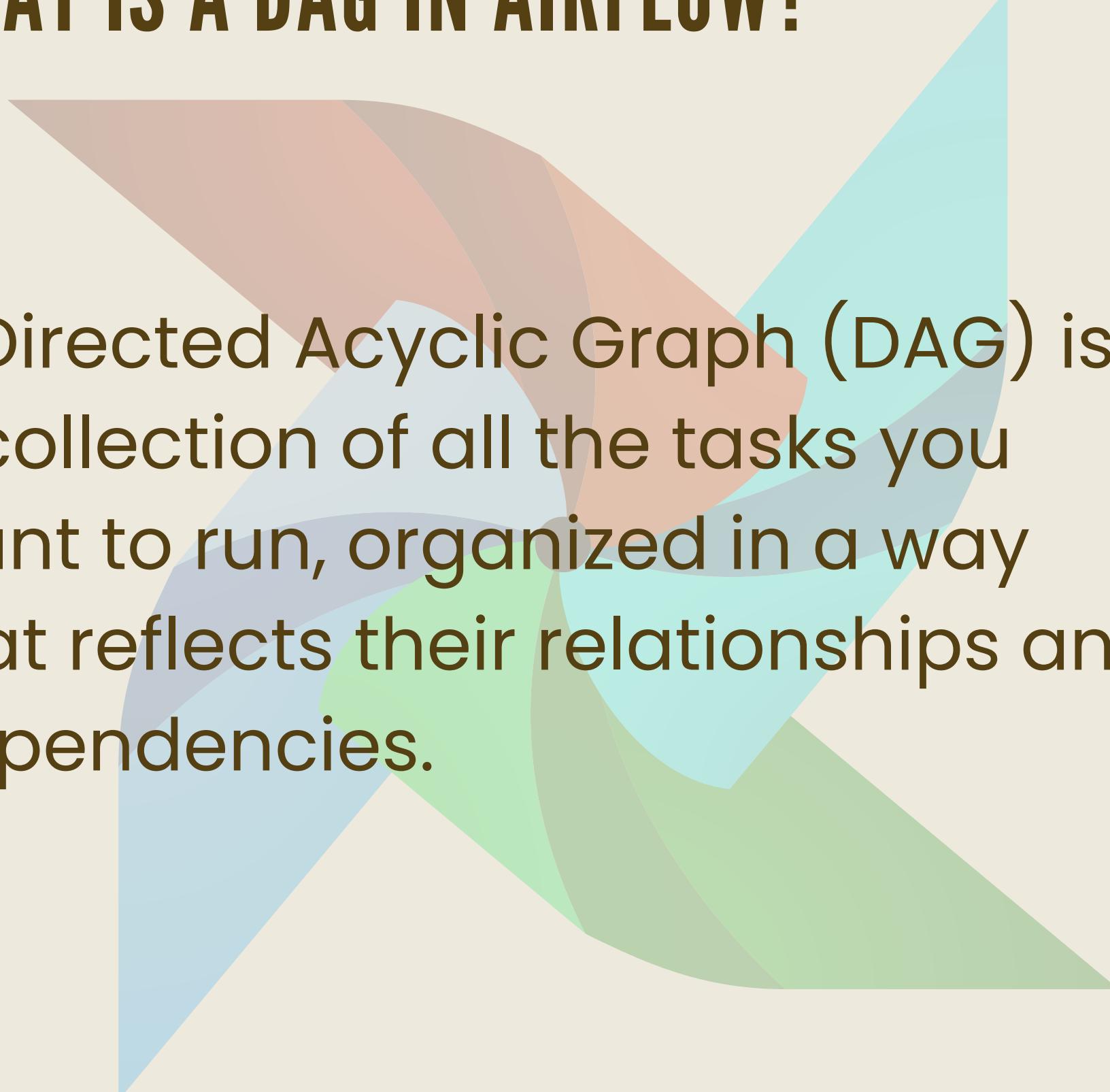


Apache Airflow is an open-source platform to programmatically author, schedule, and monitor workflows.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A DAG IN AIRFLOW?



A Directed Acyclic Graph (DAG) is a collection of all the tasks you want to run, organized in a way that reflects their relationships and dependencies.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU DEFINE A TASK IN AIRFLOW?

Tasks are defined using operators.  
An operator defines a single task in a workflow.

```
python from airflow.operators.bash  
import BashOperator  
task =  
BashOperator(task_id='bash_example',  
bash_command='echo "Hello World"',  
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE THE DIFFERENT TYPES OF OPERATORS IN AIRFLOW?

Common types include BashOperator, PythonOperator, EmailOperator, and DummyOperator.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A SENSOR IN AIRFLOW?

A sensor is a special kind of operator that will wait for a certain condition to be met before it proceeds.

```
python from airflow.sensors.filesystem  
import FileSensor  
file_sensor_task =  
FileSensor(task_id='wait_for_file',  
filepath='/path/to/file', dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SET DEPENDENCIES BETWEEN TASKS IN A DAG?

Dependencies are set using bitshift operators (`>>`, `<<`) or the `set_upstream` and `set_downstream` methods.

`python start >> task_1 task_1 >> end`

or

`task_1.set_upstream(start)`

`task_1.set_downstream(end)`



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU HANDLE RETRIES FOR A TASK IN AIRFLOW?

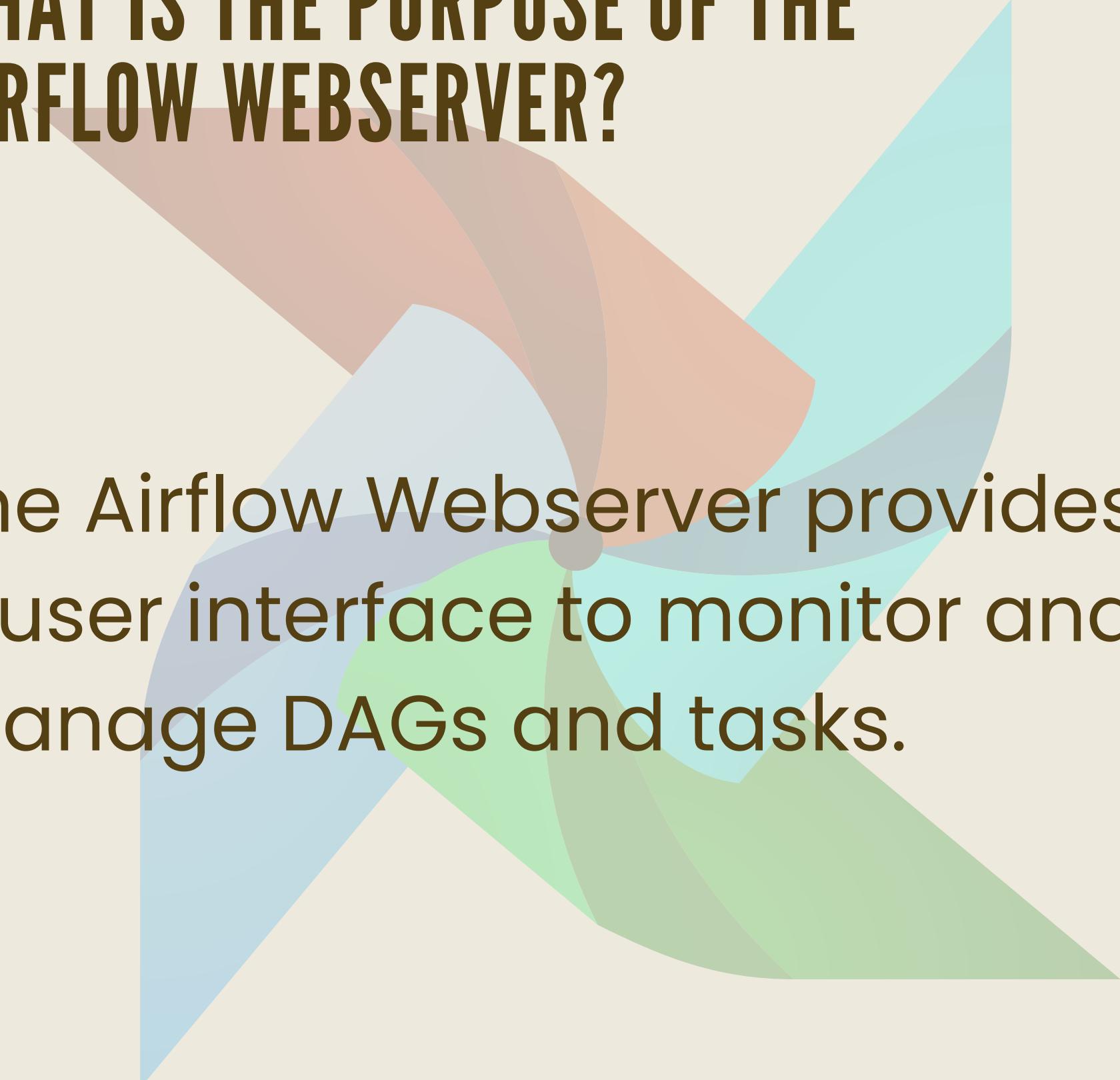
You can configure the retries and retry\_delay parameters in the task definition.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='exit 1',  
retries=3,  
retry_delay=timedelta(minutes=5),  
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE PURPOSE OF THE AIRFLOW WEB SERVER?



The Airflow Webserver provides a user interface to monitor and manage DAGs and tasks.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW CAN YOU TRIGGER A DAG MANUALLY IN AIRFLOW?

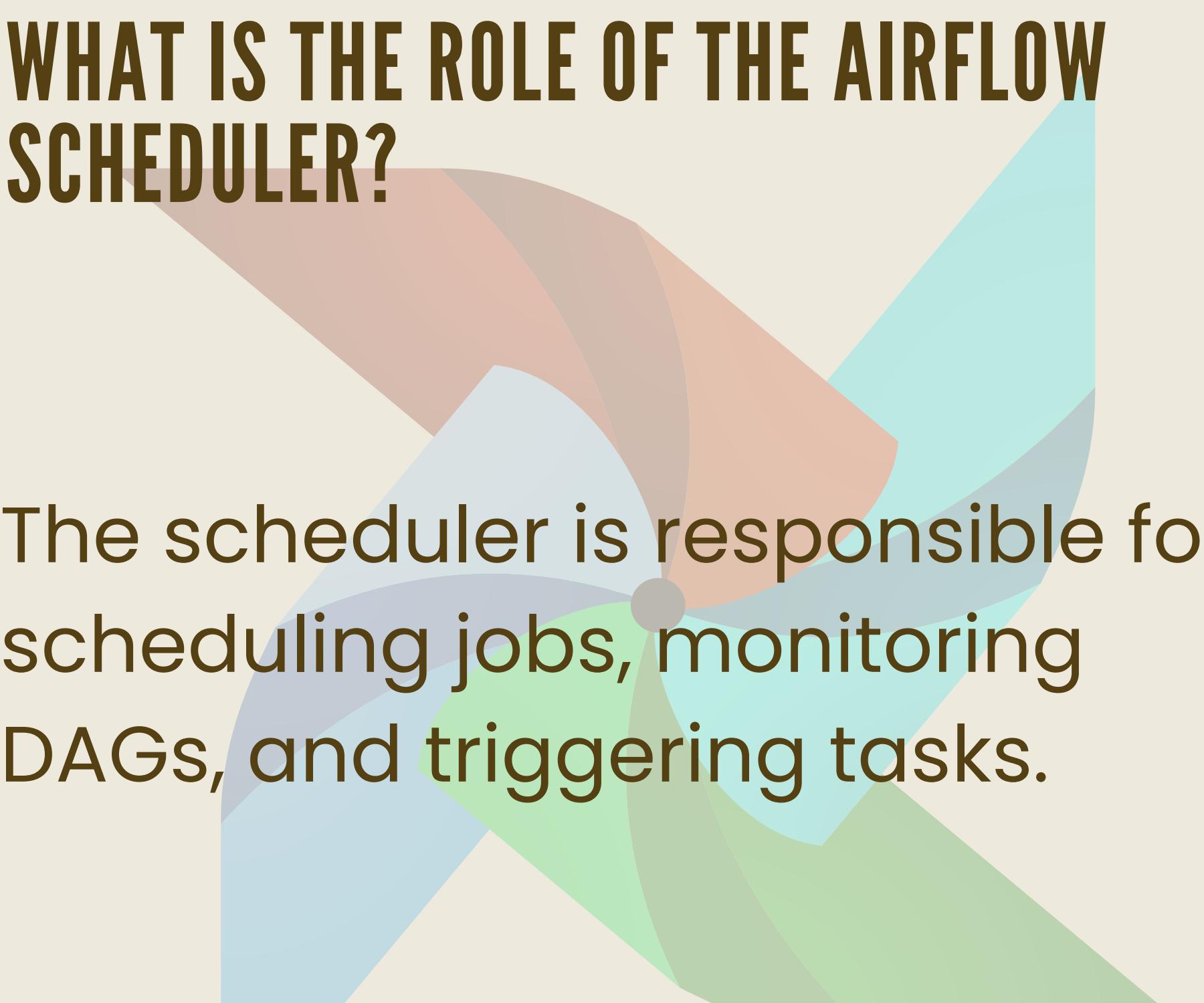
You can trigger a DAG manually using the Airflow UI or the CLI command `airflow dags trigger`.

`bash airflow dags trigger example_dag`



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE ROLE OF THE AIRFLOW SCHEDULER?



The scheduler is responsible for scheduling jobs, monitoring DAGs, and triggering tasks.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU BACKFILL A DAG IN AIRFLOW?

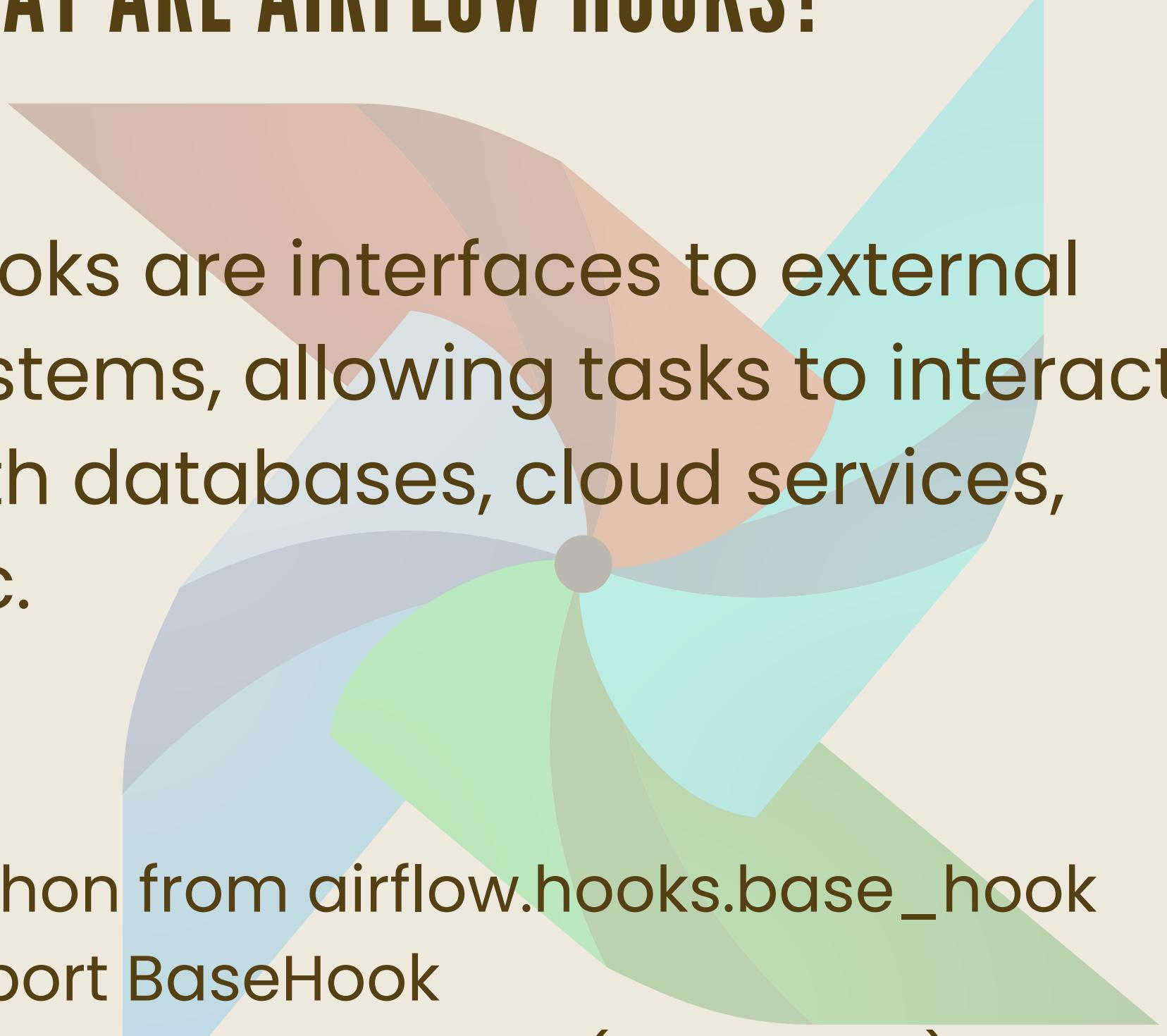
Backfilling is done by running a DAG for historical dates. This can be triggered via the CLI or UI.

```
bash airflow dags backfill -s 2021-01-01 -e 2021-01-07 example_dag
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE AIRFLOW HOOKS?



Hooks are interfaces to external systems, allowing tasks to interact with databases, cloud services, etc.

```
python from airflow.hooks.base_hook  
import BaseHook  
class MyCustomHook(BaseHook):  
    def get_conn(self): pass
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A CUSTOM OPERATOR IN AIRFLOW?

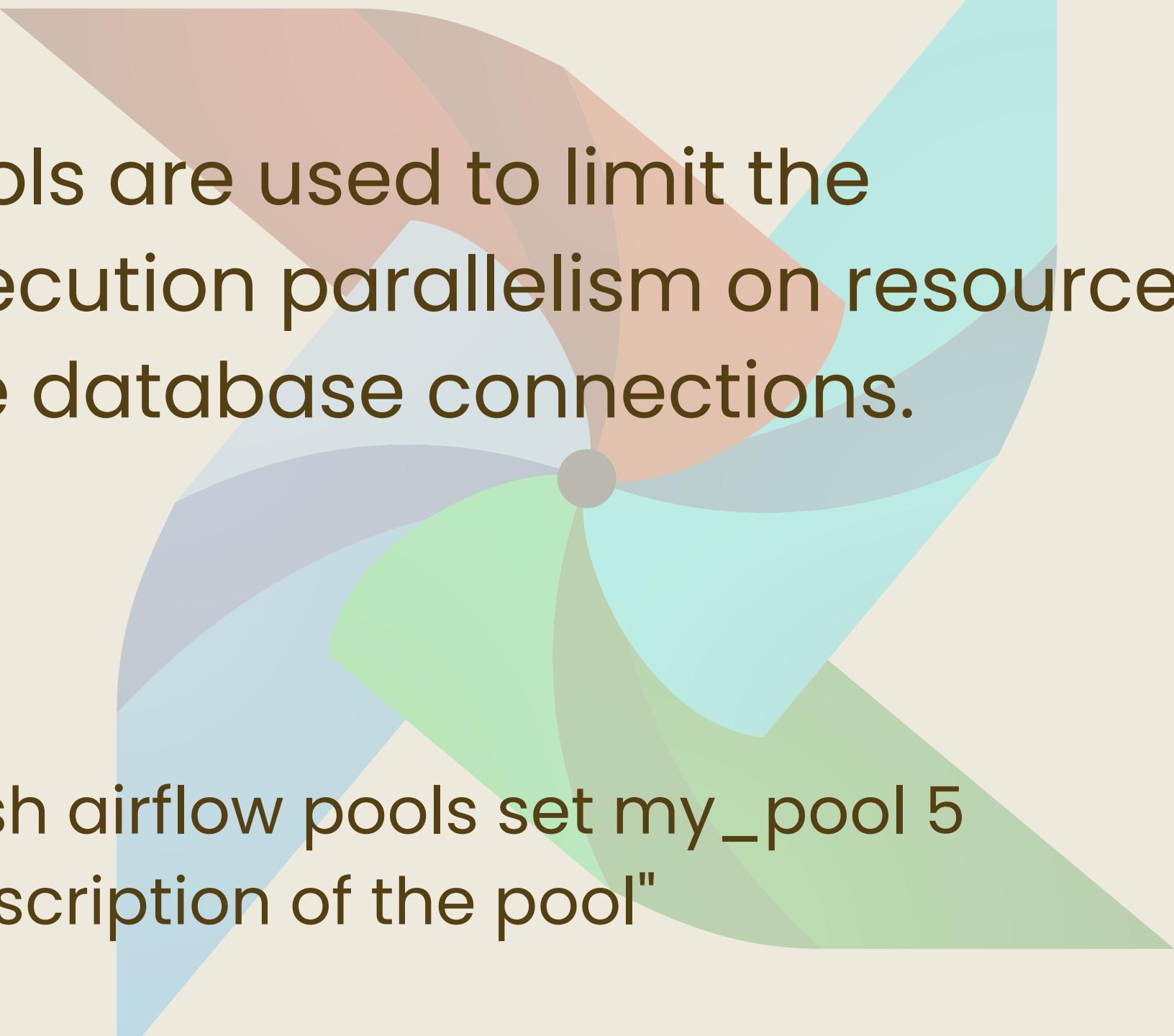


A custom operator is an extension of Airflow's base operators to include custom logic.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE POOLS IN AIRFLOW?



Pools are used to limit the execution parallelism on resources like database connections.

bash airflow pools set my\_pool 5  
"Description of the pool"



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS SLA (SERVICE LEVEL AGREEMENT) IN AIRFLOW?

SLAs are used to set time limits for tasks, with alerts triggered if the time is exceeded.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='echo "Hello World"',  
dag=dag, sla=timedelta(minutes=30))
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SECURE AN AIRFLOW INSTANCE?

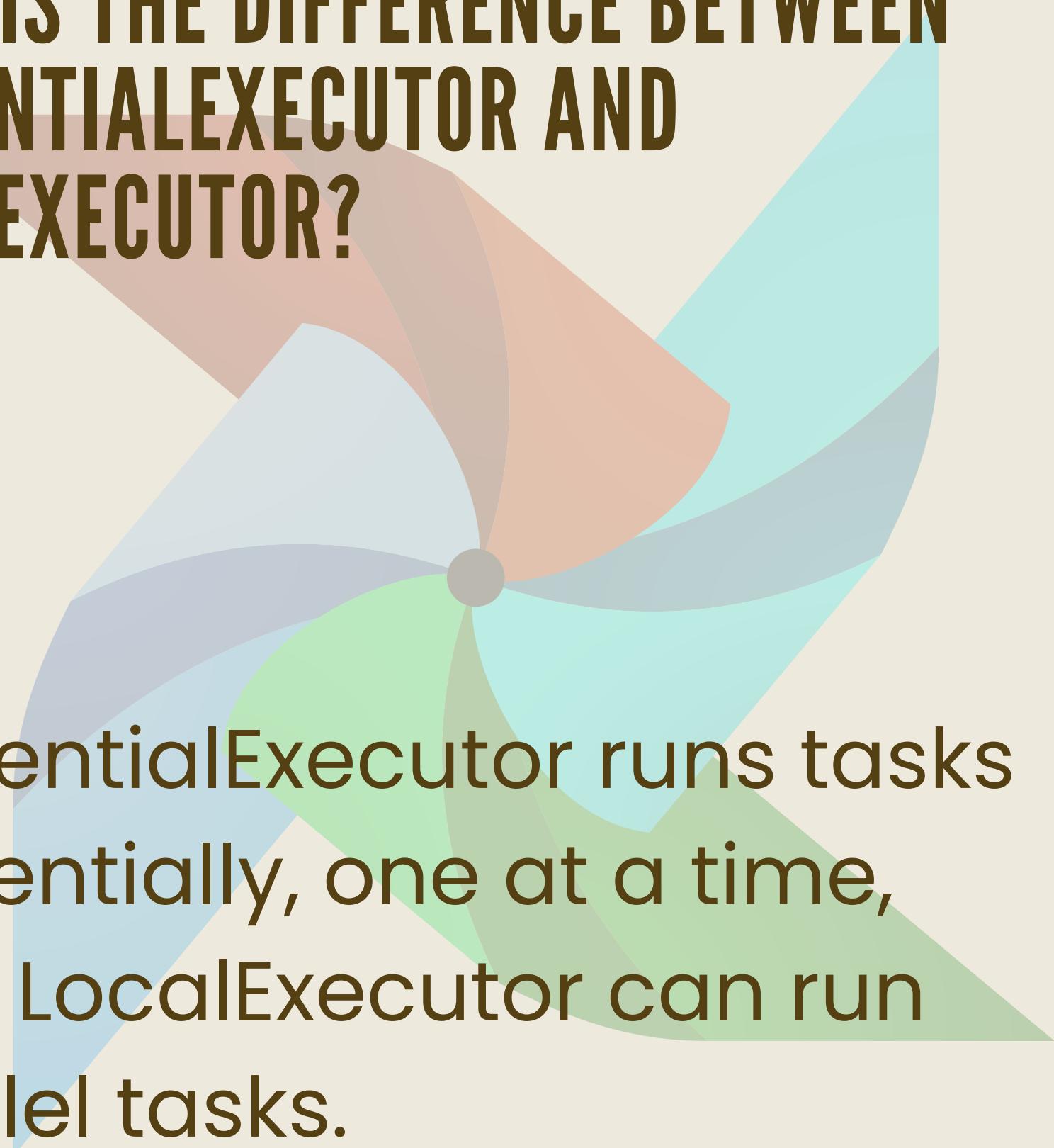
Security in Airflow can be managed through RBAC (Role-Based Access Control) and SSL encryption.

```
bash airflow users create  
--role Admin  
--username admin  
--email admin@example.com  
--firstname admin  
--lastname user  
--password admin
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE DIFFERENCE BETWEEN SEQUENTIALEXECUTOR AND LOCALEXECUTOR?

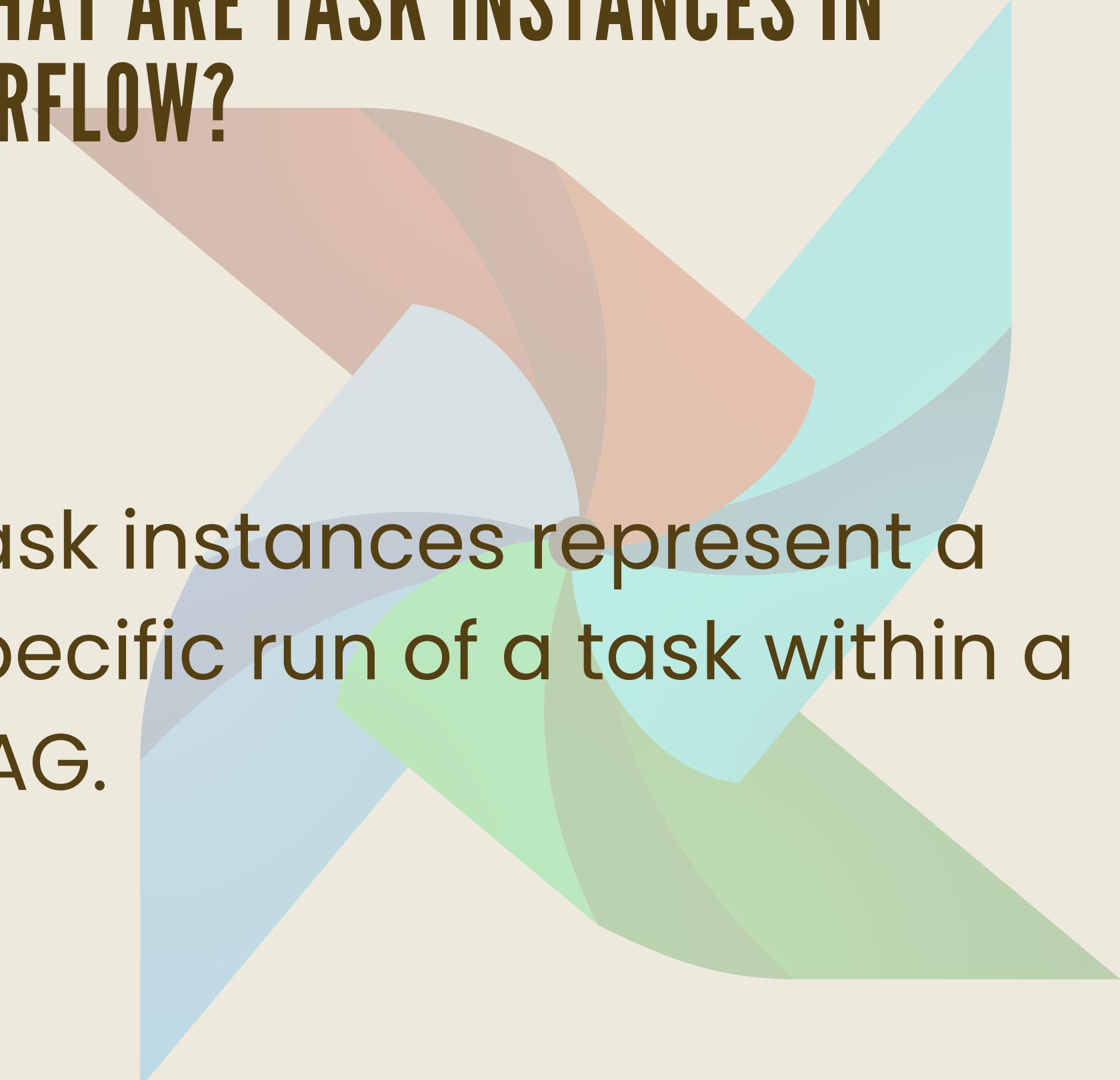


SequentialExecutor runs tasks sequentially, one at a time, while LocalExecutor can run parallel tasks.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE TASK INSTANCES IN AIRFLOW?



Task instances represent a specific run of a task within a DAG.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU PASS PARAMETERS TO A TASK IN AIRFLOW?

Parameters can be passed using the `op_args` and `op_kwargs` arguments in the task definition.

```
python python_task =  
PythonOperator(task_id='python_example',  
python_callable=my_function,  
op_args=['arg1'], dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE AIRFLOW REST API?



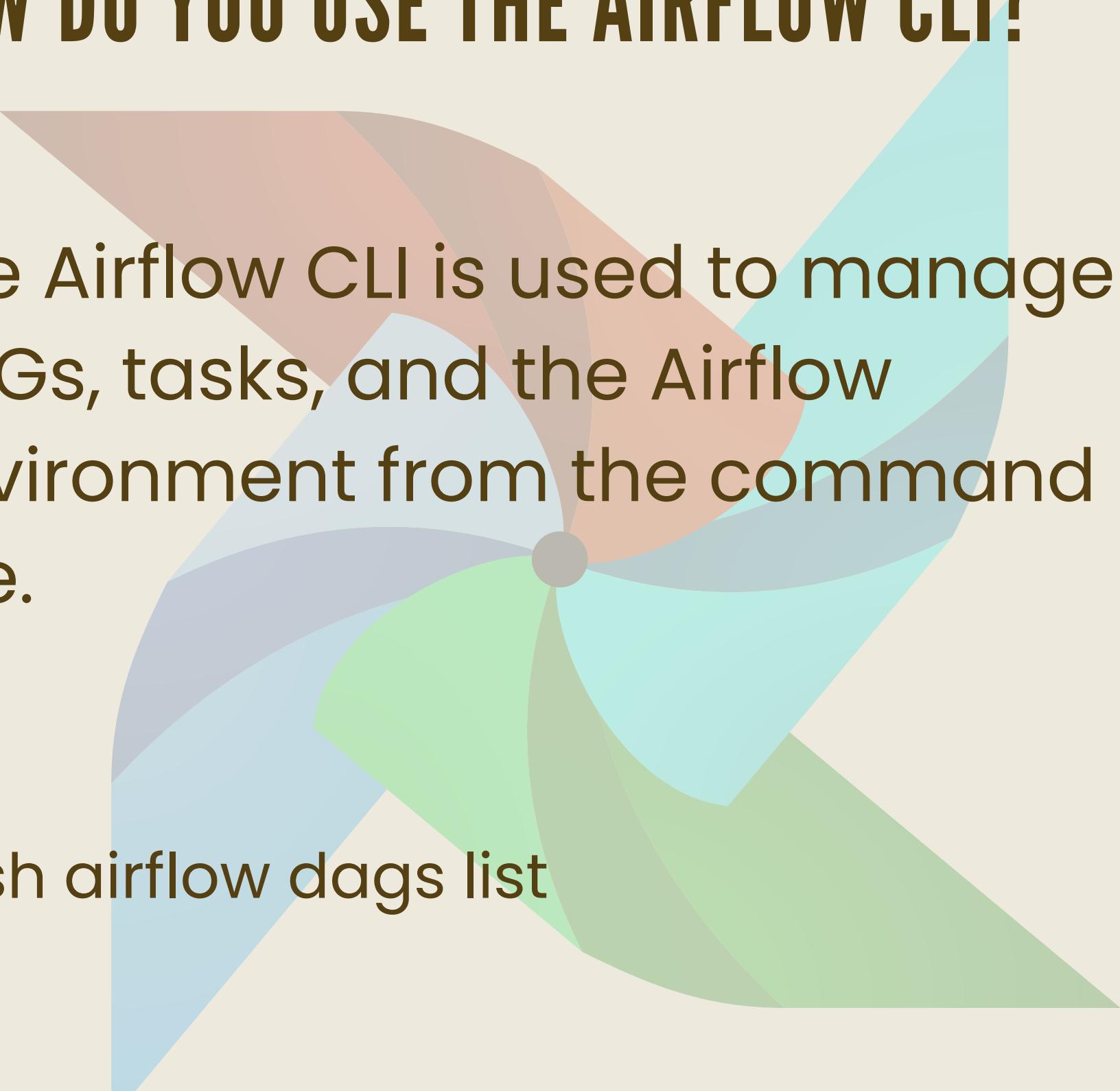
The Airflow REST API allows interaction with the Airflow instance programmatically.

```
bash curl -X GET  
"http://localhost:8080/api/v1/dags" -H  
"accept: application/json"
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU USE THE AIRFLOW CLI?



The Airflow CLI is used to manage DAGs, tasks, and the Airflow environment from the command line.

`bash airflow dags list`



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU CREATE A CONNECTION IN AIRFLOW?

Connections can be created using the Airflow UI, CLI, or by defining them in airflow.cfg.

```
bash airflow connections  
add my_conn --conn-type mysql  
--conn-host localhost  
--conn-login root  
--conn-password root
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A DAG RUN IN AIRFLOW?

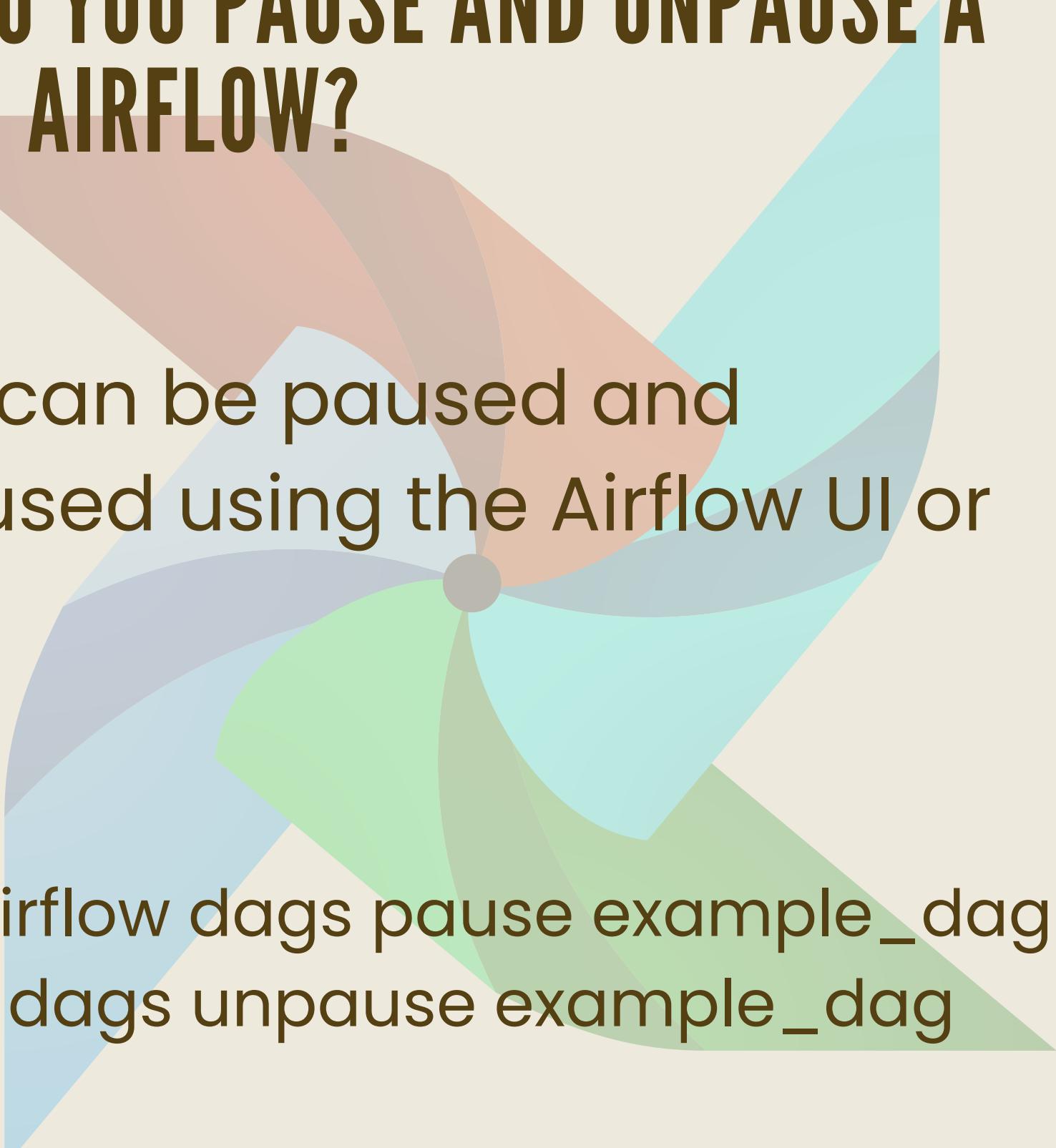


A DAG Run is an instance of a DAG, representing a specific execution of the DAG.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU PAUSE AND UNPAUSE A DAG IN AIRFLOW?



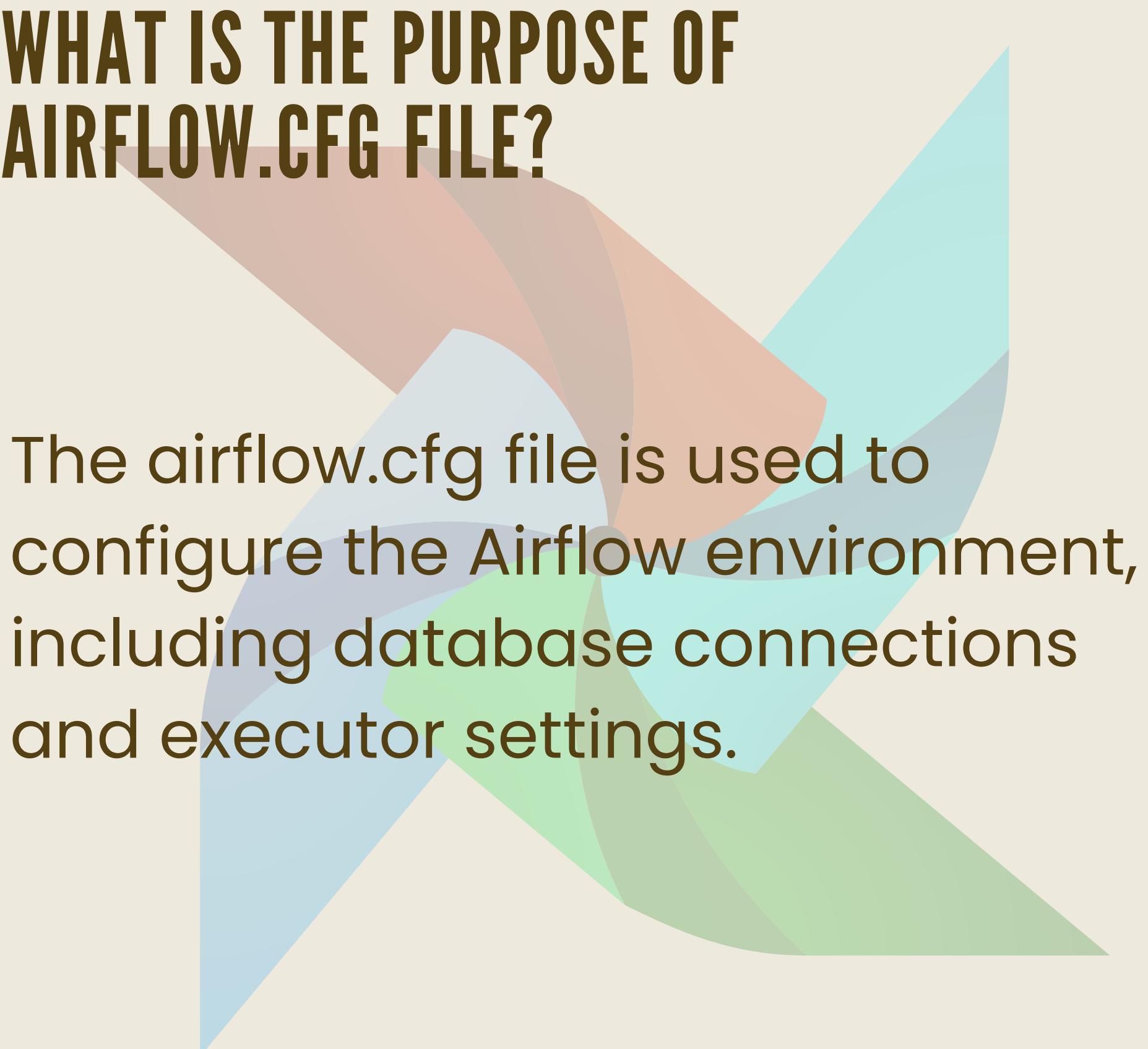
DAGs can be paused and unpause using the Airflow UI or CLI.

`bash airflow dags pause example_dag`  
`airflow dags unpause example_dag`



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE PURPOSE OF AIRFLOW.CFG FILE?



The airflow.cfg file is used to configure the Airflow environment, including database connections and executor settings.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU HANDLE TASK DEPENDENCIES DYNAMICALLY?

Task dependencies can be set dynamically within a DAG definition based on runtime conditions.

```
python if condition: task_1 >> task_2  
else: task_1 >> task_3
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE ROLE OF THE AIRFLOW METADATA DATABASE?

The metadata database stores information about DAGs, task instances, variables, connections, and more



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU MANAGE VARIABLES IN AIRFLOW?

Variables can be managed through the Airflow UI, CLI, or API. They are key-value pairs used to store configuration or runtime parameters.

```
bash airflow variables set key value  python  
my_var = Variable.get("my_variable")
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A SUBDAGOPERATOR?

The SubDagOperator is used to define a sub-DAG within a DAG. It is useful for organizing and reusing parts of workflows.

```
python from airflow.operators.subdag  
import SubDagOperator  
subdag_task =  
SubDagOperator(task_id='subdag',  
subdag=subdag, dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU HANDLE TASK TIMEOUTS IN AIRFLOW?

Timeouts can be handled by setting the `execution_timeout` parameter in the task definition.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='sleep 300',  
execution_timeout=timedelta(minutes=  
5), dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A TRIGGERDAGRUNOPERATOR?

The TriggerDagRunOperator triggers another DAG from within a DAG.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU TEST AN AIRFLOW DAG?

DAGs can be tested using the airflow test command or by running them in the Airflow UI in a safe environment.

```
bash airflow dags test example_dag  
2021-01-01
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE PURPOSE OF PROVIDE\_CONTEXT IN PYTHONOPERATOR?

provide\_context passes context variables to the task's callable function, allowing access to metadata and other information.

```
python def my_function(**kwargs):  
    execution_date = kwargs['execution_date']  
    python_task =  
        PythonOperator(task_id='python_example',  
                      python_callable=my_function,  
                      provide_context=True, dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE THE DIFFERENT TYPES OF EXECUTORS IN AIRFLOW?

Common executors include `SequentialExecutor`, `LocalExecutor`, `CeleryExecutor`, and `KubernetesExecutor`.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SCHEDULE A DAG TO RUN AT A SPECIFIC INTERVAL?

The schedule interval can be set using a cron expression or predefined interval strings like '@daily', '@hourly'.

```
python dag = DAG('example_dag',  
schedule_interval='@daily',  
start_date=days_ago(1))
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SET UP AIRFLOW LOGGING?

Airflow logs can be configured in the airflow.cfg file, specifying the logging level and log location.

```
ini [logging]
base_log_folder = /path/to/logs
remote_logging = True
remote_log_conn_id = my_s3_conn
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS AIRFLOW'S DEFAULT DATABASE?

Airflow uses SQLite as the default database, but it can be configured to use MySQL or PostgreSQL for production.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SET TASK PRIORITIES IN AIRFLOW?

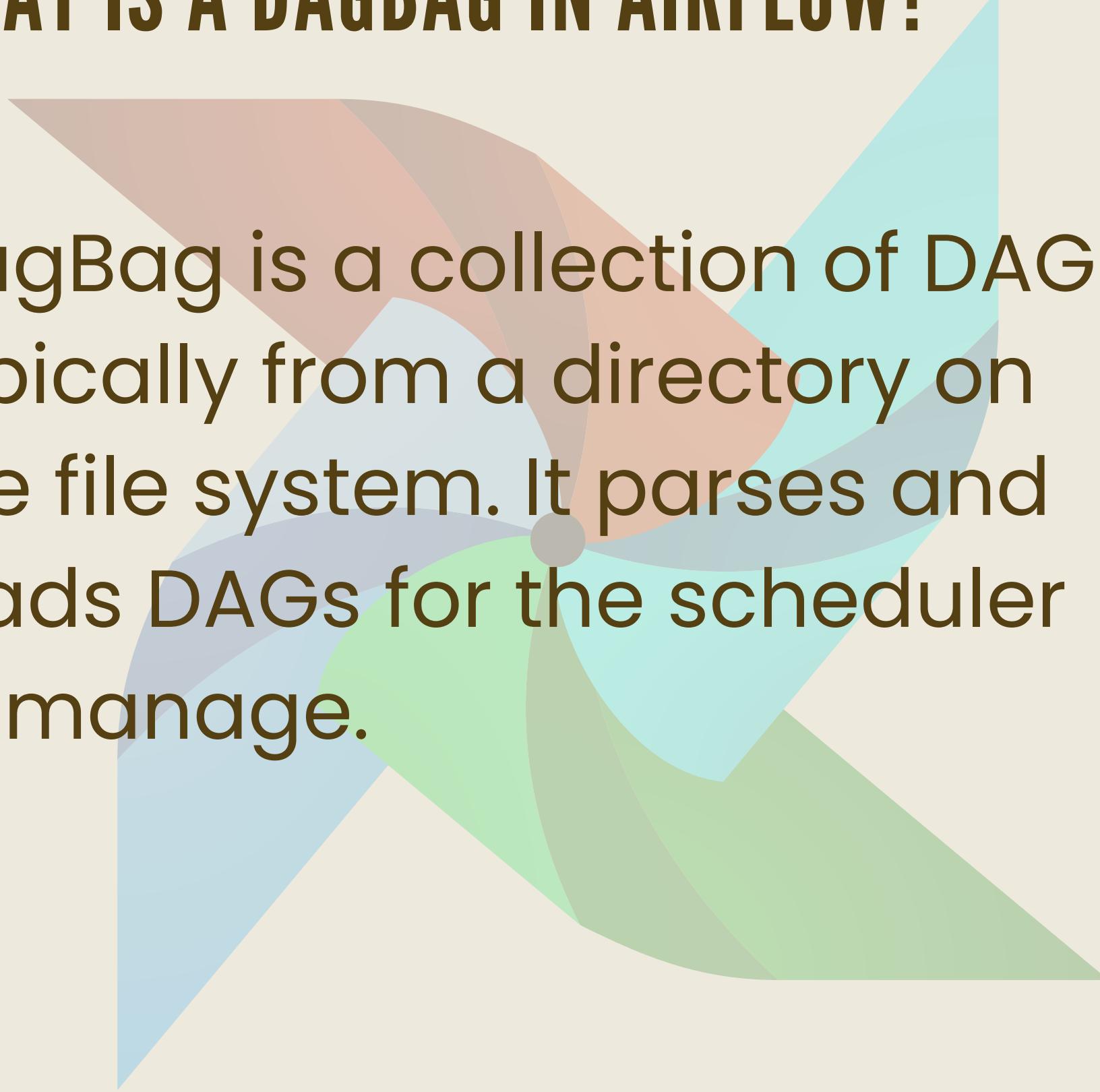
Task priorities can be set using the `priority_weight` parameter in the task definition.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='echo "Hello World",  
priority_weight=10, dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A DAGBAG IN AIRFLOW?



DagBag is a collection of DAGs, typically from a directory on the file system. It parses and loads DAGs for the scheduler to manage.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU USE THE BRANCHPYTHONOPERATOR?

The BranchPythonOperator allows branching based on the result of a Python function.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU DEFINE GLOBAL VARIABLES IN AIRFLOW?

Global variables can be defined in the airflow.cfg file or managed through the Airflow UI.

ini [variables] key = value



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU DEFINE THE START\_DATE FOR A DAG?

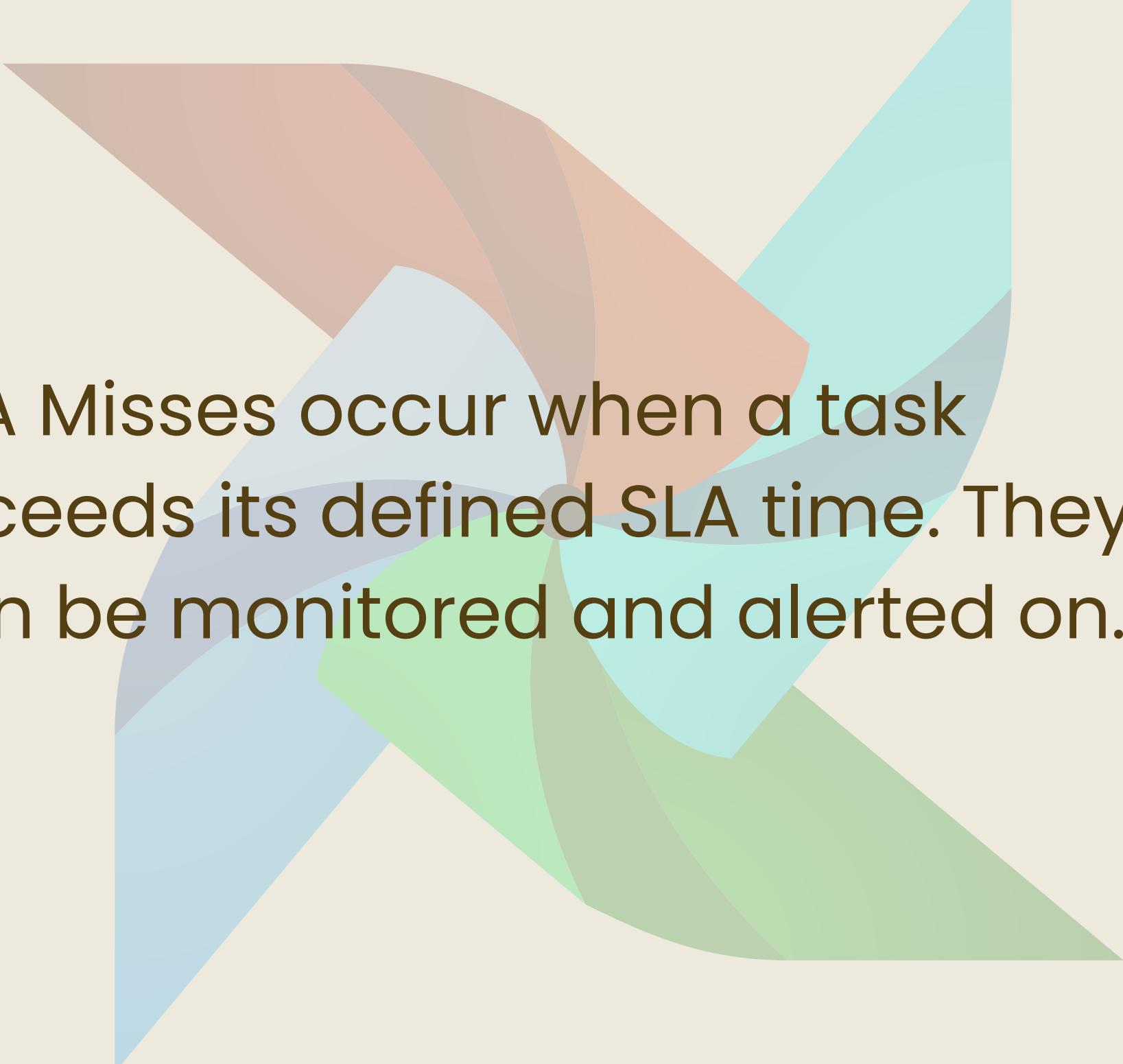
The start\_date parameter is used to define the date and time when the DAG should start running.

```
python dag = DAG('example_dag',  
start_date=datetime(2021, 1, 1))
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE SLA MISSES IN AIRFLOW?



SLA Misses occur when a task exceeds its defined SLA time. They can be monitored and alerted on.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE FUNCTION OF THE EXECUTION\_DATE IN AIRFLOW?

execution\_date represents the logical date and time for which the DAG Run is scheduled, used in templating and execution.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='echo {{ ds }}',  
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SKIP TASKS IN AIRFLOW?

Tasks can be skipped using the SkipMixin class or by setting conditions within a task.

```
python from airflow.models import  
SkipMixin  
class MyTask(SkipMixin, BaseOperator):  
    def execute(self, context):  
        if condition: self.skip(context['dag_run'],  
                               context['ti'].task_id,  
                               context['execution_date'])
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU HANDLE TASK DEPENDENCIES DYNAMICALLY?

Task dependencies can be set dynamically within a DAG definition based on runtime conditions.

```
python if condition: task_1 >> task_2  
else: task_1 >> task_3
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE DIFFERENCE BETWEEN THE START\_DATE AND END\_DATE IN AIRFLOW?

start\_date defines when a DAG should start running, while end\_date defines when it should stop running.

```
python dag = DAG('example_dag',  
start_date=datetime(2021, 1, 1),  
end_date=datetime(2021, 12, 31))
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU TRIGGER A DAG FROM ANOTHER DAG?

The TriggerDagRunOperator is used to trigger another DAG from within a DAG.

```
python from
airflow.operators.dagrun_operator
import TriggerDagRunOperator
trigger_task =
TriggerDagRunOperator(task_id='trigger
_dag', trigger_dag_id='example_dag',
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE USE OF AIRFLOW'S TASKFLOW API?

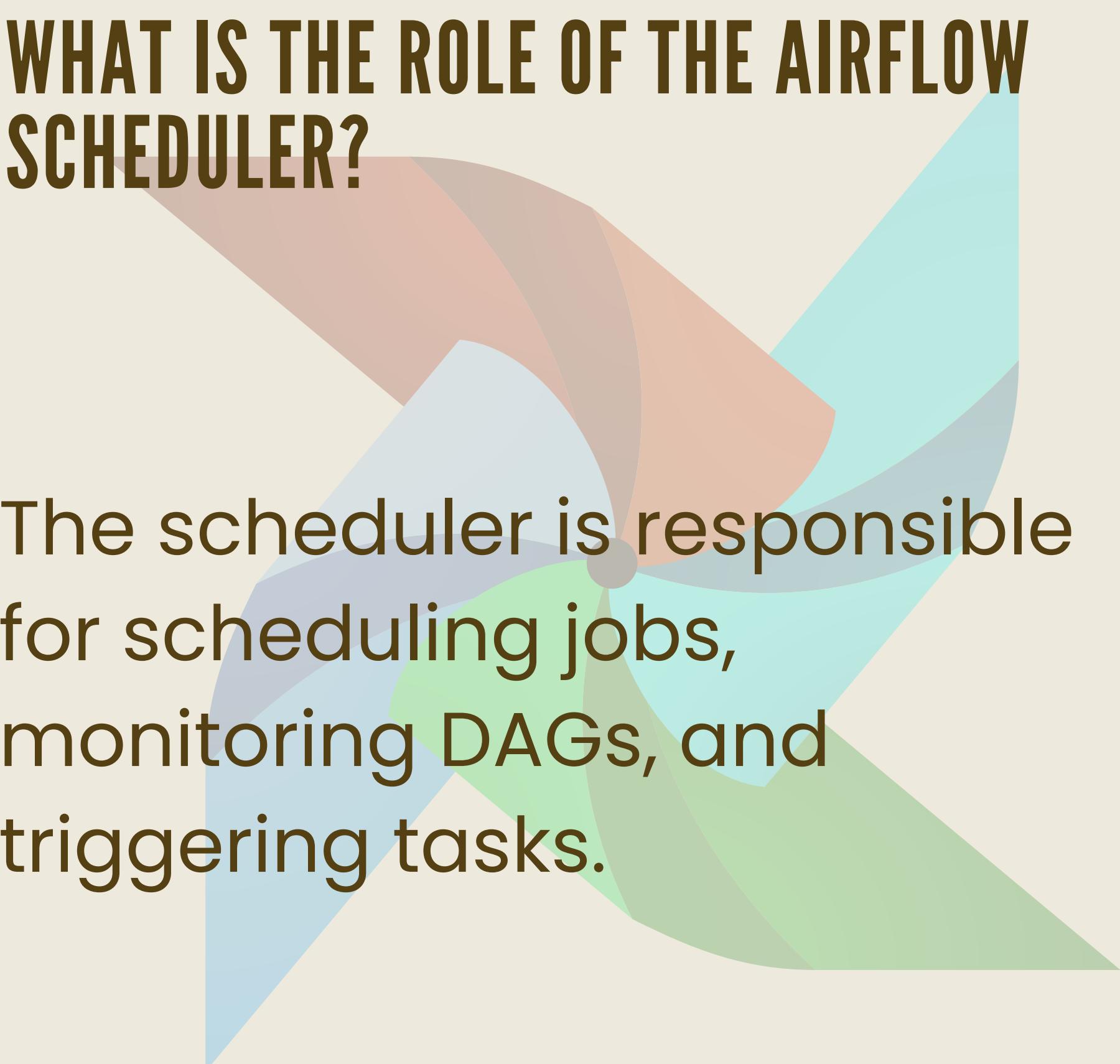
The Taskflow API simplifies the creation and management of task dependencies and data passing between tasks.

```
python from airflow.decorators import
dag, task
@dag(schedule_interval='@daily',
start_date=days_ago(2))
def example_dag(): @task def
extract(): return 'data' @task def
process(data): return f'processed
{data}' data = extract() process(data)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE ROLE OF THE AIRFLOW SCHEDULER?



The scheduler is responsible for scheduling jobs, monitoring DAGs, and triggering tasks.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU CONFIGURE RETRIES FOR A TASK IN AIRFLOW?

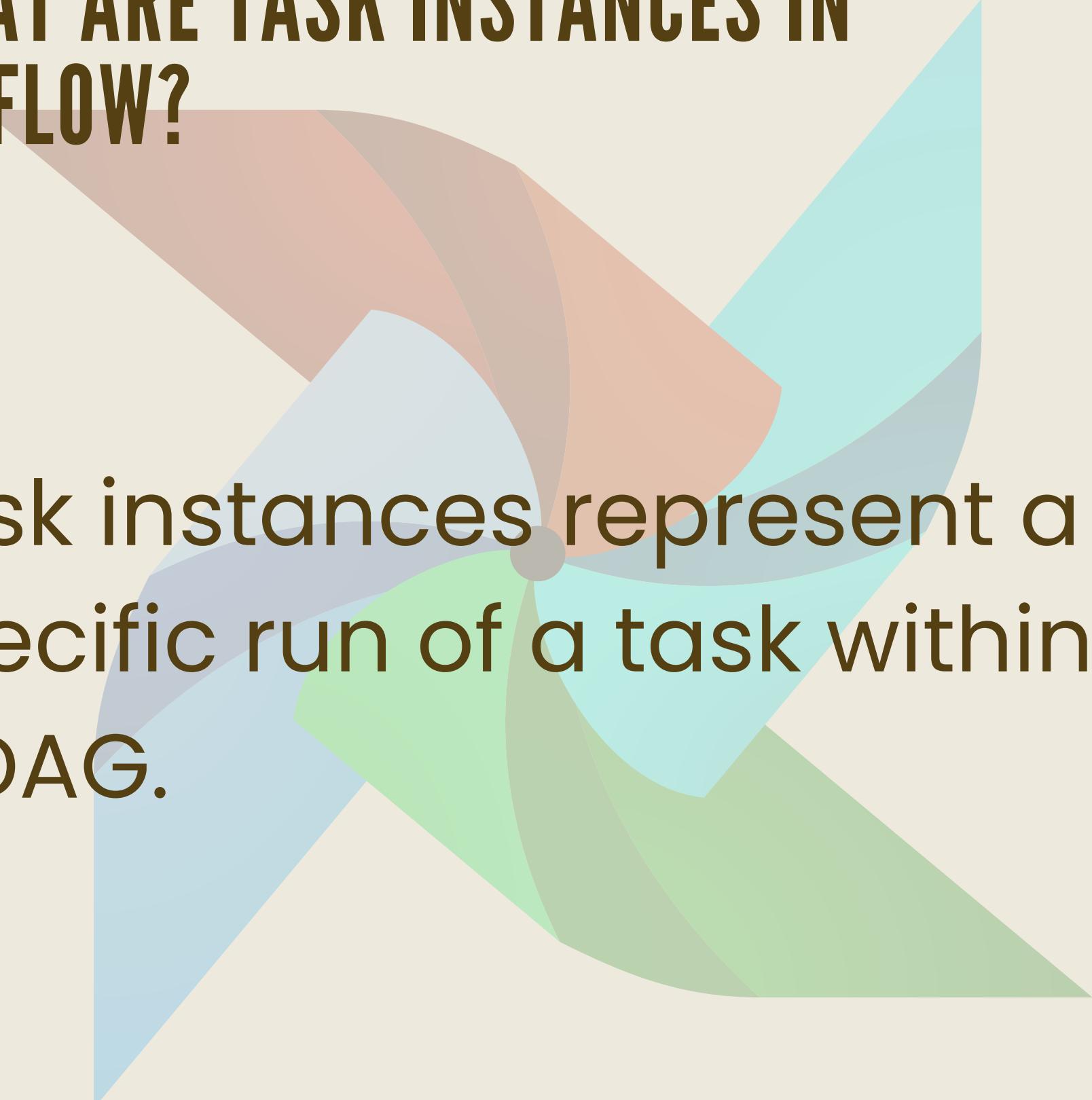
Retries can be configured using the retries and retry\_delay parameters in the task definition.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='exit 1', retries=3,  
retry_delay=timedelta(minutes=5),  
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE TASK INSTANCES IN AIRFLOW?



Task instances represent a specific run of a task within a DAG.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU HANDLE TASK FAILURES IN AIRFLOW?

Task failures can be handled by setting up retries, using the `on_failure_callback`, or configuring alerts.

```
python def failure_callback(context):
    print("Task failed")
task =
BashOperator(task_id='bash_example',
bash_command='exit 1',
on_failure_callback=failure_callback,
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE USE OF EXECUTION\_DATE IN AIRFLOW?

execution\_date represents the logical date and time for which the DAG Run is scheduled, used in templating and execution.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='echo {{ ds }}',  
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS A SENSOR IN AIRFLOW?

A sensor is a special kind of operator that will wait for a certain condition to be met before it proceeds.

```
python from airflow.sensors.filesystem  
import FileSensor
```

```
file_sensor_task =  
FileSensor(task_id='wait_for_file',  
filepath='/path/to/file', dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU USE TEMPLATES IN AIRFLOW?

Templates in Airflow are used to dynamically generate task parameters using Jinja templating.

```
python from datetime import datetime  
task =  
BashOperator(task_id='templated_task',  
bash_command='echo {{ ds }}',  
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT ARE THE DIFFERENT TYPES OF EXECUTORS IN AIRFLOW?

Common executors include SequentialExecutor, LocalExecutor, CeleryExecutor, and KubernetesExecutor.



Shwetank Singh  
GritSetGrow - GSGLearn.com

# WHAT IS THE PURPOSE OF EXECUTION\_DATE IN AIRFLOW?

execution\_date represents the logical date and time for which the DAG Run is scheduled, used in templating and execution.

```
python task =  
BashOperator(task_id='bash_example',  
bash_command='echo {{ ds }}',  
dag=dag)
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SECURE AN AIRFLOW INSTANCE?

Security in Airflow can be managed through RBAC (Role-Based Access Control) and SSL encryption.

```
bash airflow users create  
--role Admin  
--username admin  
--email admin@example.com  
--firstname admin  
--lastname user  
--password admin
```



Shwetank Singh  
GritSetGrow - GSGLearn.com

# HOW DO YOU SCHEDULE A DAG TO RUN AT A SPECIFIC INTERVAL?

The schedule interval can be set using a cron expression or predefined interval strings like '@daily', '@hourly'.

```
python dag =  
DAG('example_dag',  
schedule_interval='@daily',  
start_date=days_ago(1))
```



Shwetank Singh  
GritSetGrow - GSGLearn.com



The background features a central white circle containing the text "THANK YOU". Behind the circle is a complex arrangement of overlapping semi-transparent colored shapes. These shapes include shades of brown, tan, light blue, teal, green, and grey. They overlap in various ways, creating a layered effect that suggests depth and movement. The overall composition is clean and modern, with a focus on the central message.

**THANK YOU**