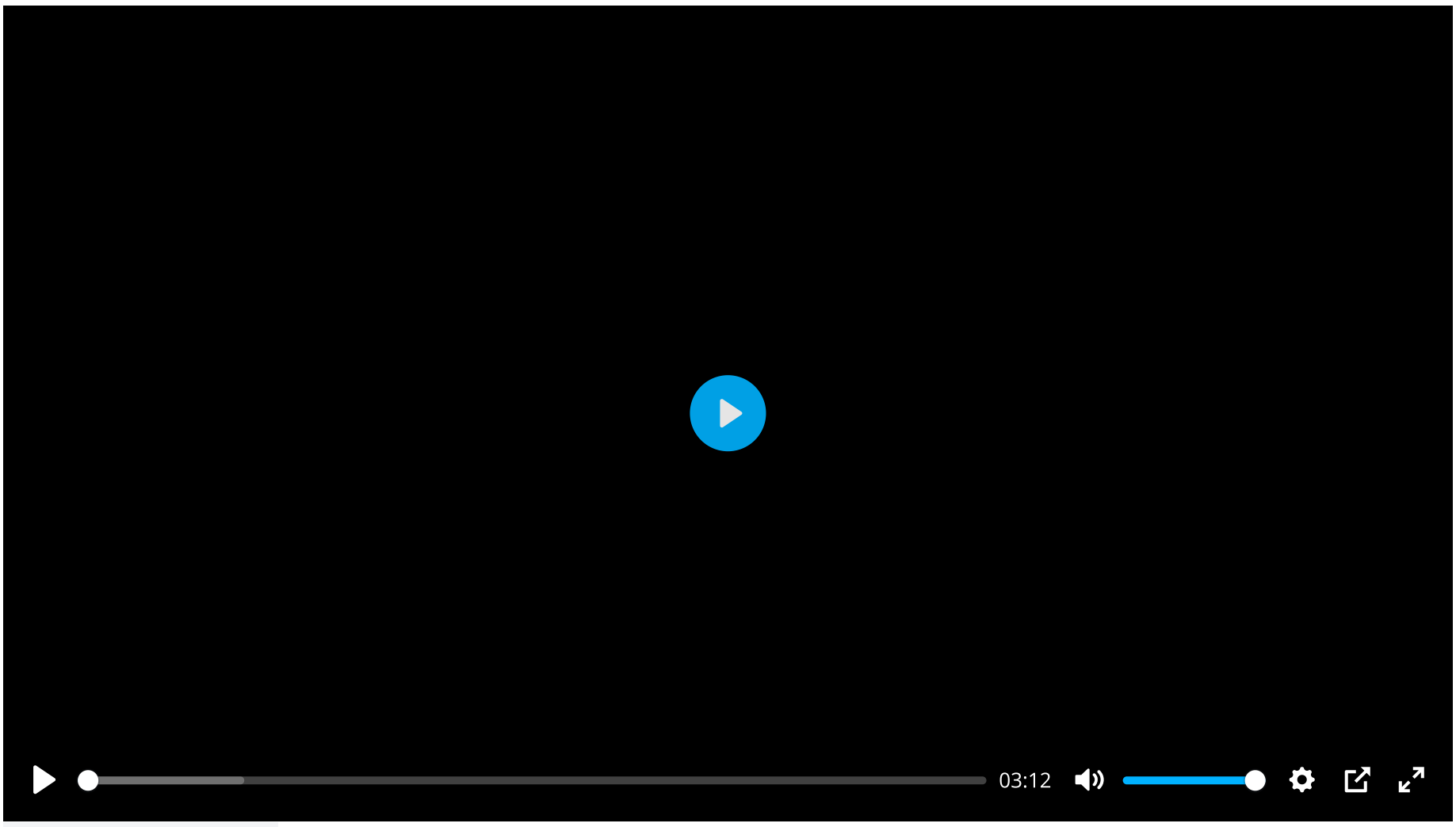
8.3 Создание и работа с DAG



Презентация к уроку 8.3.1

Текстовая расшифровка видео:

DAG

План:

- Механизм DAG;
- Создание DAG 4-мя способами;
- Стандартный конструктор;
- Менеджер контекста with;
- DAG-декоратор @dag;
- TaskFlow API. Декораторы @task;
- Параметры запуска DAG.

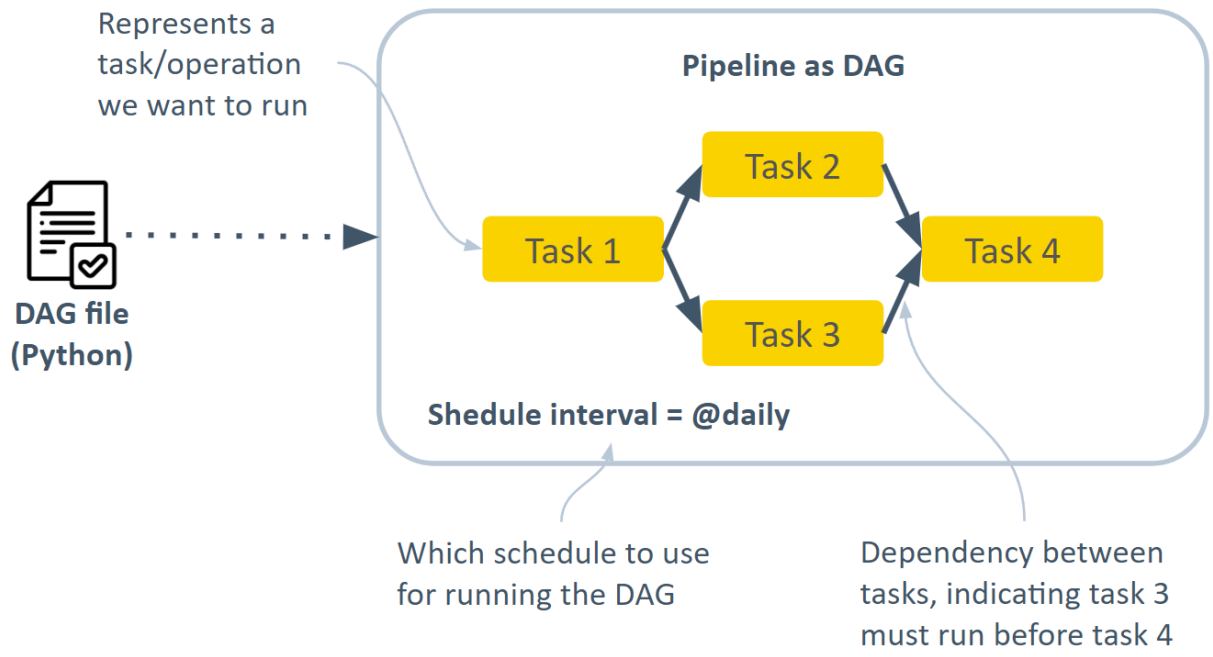
Механизм DAG

DAG (directed acyclic graph) – это направленный граф без циклов. Это примитив Airflow, где:

- **Airflow** как оркестратор – менеджмент компании;
- **DAG** – менеджер отделов;
- **Tasks** – работники.

DAG считывается с директории, которая указана у Airflow. По умолчанию это папка «Dags». В нее мы кладем Python-файл с объектом DAG. Далее, он считывается и отображается в Web-UI.

Мы можем указывать зависимость тасков:



Мы можем указывать интервал выполнения. В указанном выше примере интервал «daily», то есть «ежедневно».

Обязательно укажите **начало** и **ID** DAG.

Создание DAG 4-мя способами

Способы создания DAG:

- Через стандартный конструктор, передающий DAG операторам;
- Через менеджера контекста with;
- Через DAG-декоратор @dag;
- Через TaskFlow API.

Стандартный конструктор

С помощью стандартного конструктора мы импортируем объект «DAG» из Airflow, создаем DAG как объект, указываем его параметры и ссылаемся на этот объект в тасках (в операторах):

```
my_dag = DAG("my_dag_name", start_date=pendulum.datetime(2023, 1, 1, tz="UTC"), schedule="@daily", catchup=False)
op = EmptyOperator(task_id="task", dag=my_dag)
```

Менеджер контекста with

Когда мы используем менеджер контекста, мы используем его как классический менеджер контекста с его файлами и другими объектами. Все таски указываются внутри него с отступом и без указания DAG ID:

```
with DAG(
    "my_dag_name", start_date=pendulum.datetime(2023, 1, 1, tz="UTC"),
    schedule="@daily", catchup=False
) as dag:
    op = EmptyOperator(task_id="task")
```

DAG-декоратор @dag

DAG-декоратор указывается перед функциями. В декораторе мы указываем функции, относящиеся к данному DAG, а внутри функции указываем оператора:

```
@dag(start_date=pendulum.datetime(2023, 1, 1, tz="UTC"),
     schedule="@daily", catchup=False)
def generate_dag():
    op = EmptyOperator(task_id="task")
…
dag = generate_dag()
```

TaskFlow API. Декораторы @task

TaskFlow API позволяет указывать декораторы не только для DAG, но и для тасков. Эта фича появилась в Airflow2.0, вместе с ней появились hooks-крючки:

```
@dag(schedule=None,start_date=pendulum.datetime(2023, 1, 1, tz="UTC"),catchup=False,tags=["example"],)
def example_dag_decorator(email: str = "example@example.com"):
    get_ip = GetRequestOperator(task_id="get_ip", url="http://httpbin.org/get")
    @task(multiple_outputs=True)
    def prepare_email(raw_json: dict[str, Any]) -> dict[str, str]:
        external_ip = raw_json["origin"]
        return {"subject": f"Server connected from {external_ip}","body": f"Seems like today your serv
    email_info = prepare_email(get_ip.output)
    EmailOperator(
        task_id="send_email", to=email, subject=email_info["subject"], html_content=email_info["body"]

example_dag = example_dag_decorator()
```

TaskFlow API позволяет указывать параметры тасков, а также не использовать декоратор «DAG», указывая декораторами таски:

```
from airflow.decorators import task
from airflow.operators.email import EmailOperator
@task
def get_ip():
    return my_ip_service.get_main_ip()
@task
def compose_email(external_ip):
    return {'subject':f'Server connected from {external_ip}',
        'body': f'Your server executing Airflow is connected from the external IP {external_ip}<br>'}
email_info = compose_email(get_ip())
EmailOperator(task_id='send_email',to='example@example.com',subject=email_info['subject'],html_content
```

Параметры запуска DAG

```
my_dag = DAG("my_dag_name", start_date=pendulum.datetime(2023, 1, 1, tz="UTC"), schedule="@daily", catchup=False)
```

Первый параметр – **DAG ID** (my_dag_name). Здесь должны быть буквенные символы, числовые символы, тире, нижние подчеркивания.

Start_date – это datetime-параметр, который обязательно нужно указывать. Это то время, с которого DAG начинает выполнение, когда вы сдвигаете ползунок.

Scheduler – это интервал выполнения. Его необязательно указывать, однако в примере он указан как «daily», то есть выполняется ежедневно.

Catchup – это параметр, выполняющий DAG за предыдущие дни. По умолчанию – true. Лучше использовать false во избежание выполнения всех DAG's за большой промежуток времени, если они вам не нужны.

Документация по параметрам DAG