



UNIVERSITÉ **CÔTE D'AZUR** 

DOCUMENTATION ENVIRONNEMENT AIRFLOW

Effectué par :

NITEKA Lys Ciella
DIALLO Elhadj Mamadou Foula
PAJANY CARPIN CAOUNDIN Allan

Master 2 - Mobilité, Base de Données et Intégration
Système



Tuteur projet :

Mr. HAMMOUDI Alaaeddine

Tuteurs universitaires :

Mr. MOPOLO-MOKE Gabriel
Mr. GALLI Grégory

Table des matières

| | | |
|------|--------------------------------------|---|
| 1. | Installation d'Airflow ----- | 2 |
| 1.1. | Prérequis----- | 2 |
| 1.2. | Docker - Windows ----- | 2 |
| 1.3. | Linux (Os ou machine virtuelle)----- | 3 |
| 2. | Environnement Airflow ----- | 4 |

1. INSTALLATION D'AIRFLOW

1.1. Prérequis

Afin de pouvoir installer Airflow, vous devrez respecter certains prérequis :

| Windows | Linux |
|-------------------------------------|-----------|
| Docker Desktop ou machine virtuelle | Anaconda, |
| Python | |
| VS Code (Optionnel) | |

1.2. Docker - Windows

Pour utiliser au mieux Docker, nous vous conseillons de créer un compte et ensuite d'effectuer l'installation :

- ✓ Création compte [ici](#).
- ✓ Télécharger [Docker Destop](#)

Une fois Docker Desktop téléchargé, l'installer en suivant le processus d'installation.

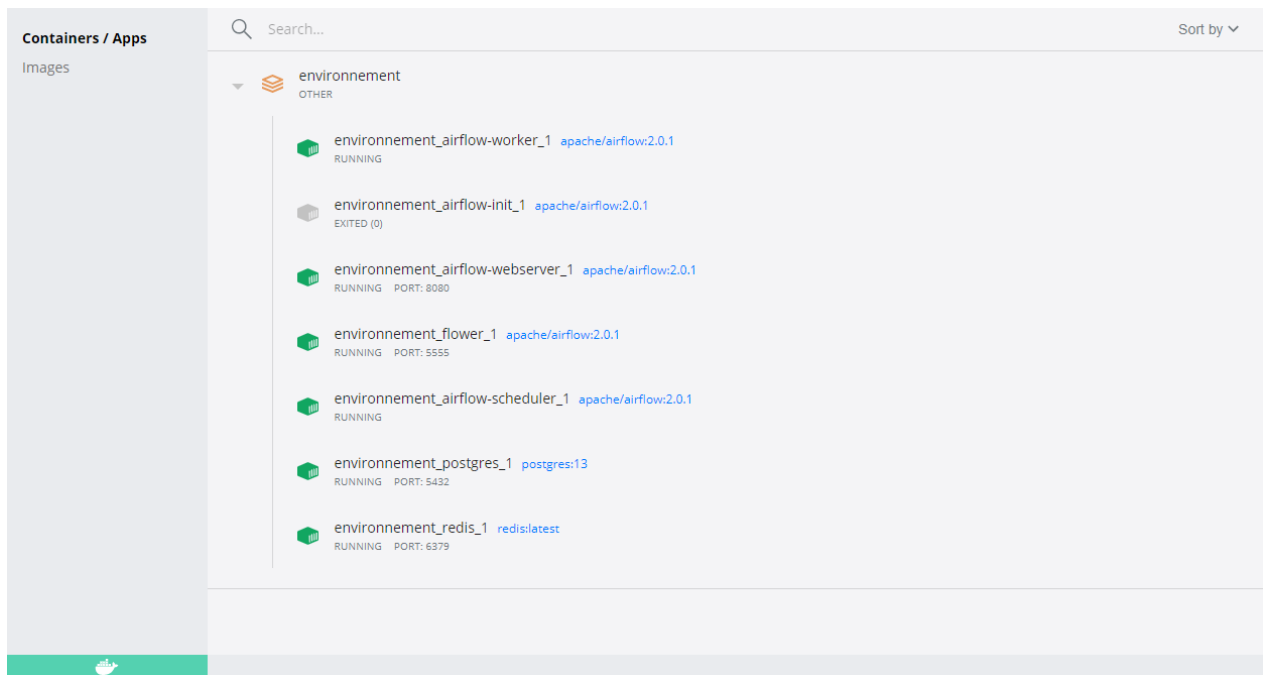
Après l'installation, le logiciel se lancera automatiquement en tâche de fond. Une icône de baleine devrait être présent dans icônes cachés de la barre de tâche. Connectez-vous à l'aide de vos identifiants créé précédemment à Docker en faisant un clic droit sur l'icône et en sélectionnant **Sign in / Create Docker ID**.

Cette étape terminée, rendez-vous dans votre environnement dans travail. Exécuter le fichier « docker-compose.yaml » :

- Ouvrir un terminal et se rendre dans le dossier où le fichier se trouve
- Exécuter **docker-compose up**

Il devrait commencer par télécharger et installer certaines images spécifiques à notre environnement Airflow :

- Airflow
- Mysql
- Redis



1.3. Linux (Os ou machine virtuelle)

Concernant les systèmes Linux, nous allons utiliser l'environnement Anaconda.

1. Installation anaconda dans répertoire utilisateur
Récupérer la dernière le nom de la dernière version d'anaconda [ici](#). Ce fichier d'installation sera sous forme de fichier Bash (exemple : **Anaconda3-2020.11-Linux-x86_64.sh**).
2. Se connecter en mode root (sudo) et taper les commandes suivantes afin de télécharger anaconda :

```
cd /tmp  
curl -O https://repo.anaconda.com/archive/nomFichierAnaconda.sh
```
3. Exécuter la commande : **bash Anaconda3-2019.03-Linux-x86_64.sh**
4. Pendant l'étape d'installation, appuyer sur entrée pour passer la lecture du contrat de licence jusqu'à la fin, taper ensuite « yes » à la fin. Pendant tout le processus, il est conseillé de répondre « yes » aux questions pour garder une installation par défaut.
5. Après ceci, nous sommes invités à choisir le dossier d'installation. Nous pouvons le laisser par défaut ou le spécifier.
6. Une dernière question à répondre « yes »

A la fin de l'installation, se rendre dans le dossier d'installations d'anaconda : anaconda/bin et taper la commande :

conda activate

Ensuite installer Apache/Airflow à l'aide de python :

sudo pip install apache-airflow==2.0.1

Après l'installation d'Airflow trouver le répertoire d'installation (généralement dans le dossier user) et taper la commande suivante :

export AIRFLOW_HOME=nomRepertoire (exemple : /home/nomUser/Apache/Airflow)

Nous allons ensuite créer un utilisateur afin de permettre la connexion à l'environnement :

**airflow users create --role Admin --username nomUser --password pass --email toto@.. --
firtname fname --lastname lname**

Un message de confirmation de création d'utilisateur s'affichera ensuite.

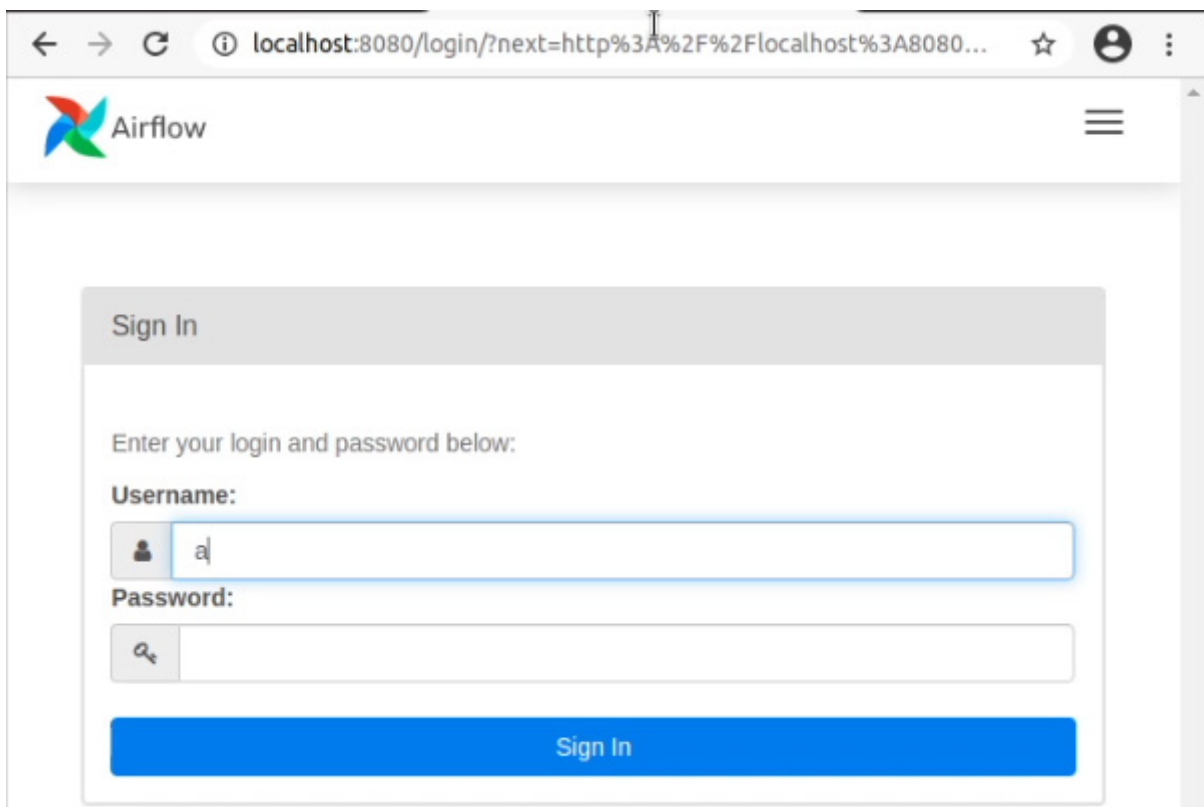
Taper ensuite les commandes suivantes :

```
Airflow db init  
airflow scheduler -D  
airflow webserver -p 8080 -D
```

Le scheduler et le webserver lancé, nous pouvons donc nous connecter à l'interface graphique web à l'adresse :

Localhost :8080

A la première connexion, il vous faudra entre votre nom d'utilisateur et votre mot de passe.

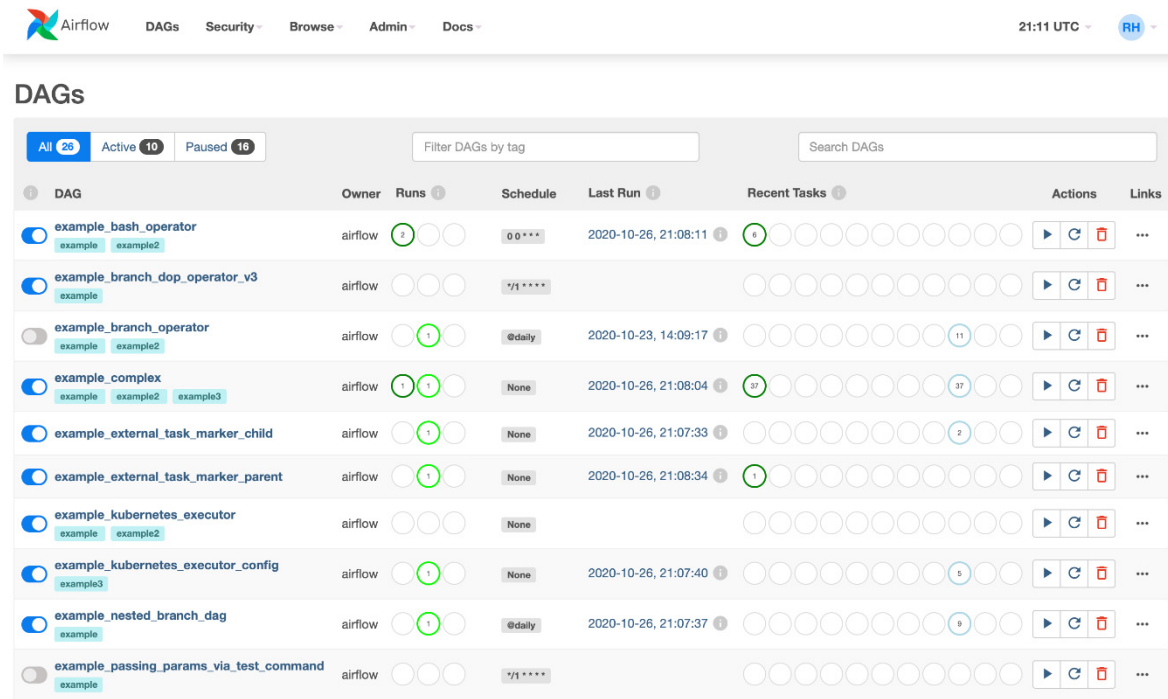


2. ENVIRONNEMENT AIRFLOW

La mise en place étant plus complexe sous Windows, nous détaillerons ici un environnement orienté docker

En se rendant sur l'adresse suivante, on accèdera à l'interface graphique d'Airflow :
<http://localhost:8080/>

DAG : vue d'ensemble de tous les DAG de votre environnement.



| DAG | Owner | Runs | Schedule | Last Run | Recent Tasks | Actions | Links |
|---|---------|-------------|---------------|----------------------|--------------|---------------------------|-------|
| <input checked="" type="checkbox"/> example_bash_operator <small>example example2</small> | airflow | 2 (Success) | 0 0 *** | 2020-10-26, 21:08:11 | 6 (Success) | [Play] [Refresh] [Delete] | ... |
| <input checked="" type="checkbox"/> example_branch_dop_operator_v3 <small>example</small> | airflow | 0 | * / 1 * * * * | | 0 | [Play] [Refresh] [Delete] | ... |
| <input type="checkbox"/> example_branch_operator <small>example example2</small> | airflow | 1 (Success) | @daily | 2020-10-23, 14:09:17 | 11 (Success) | [Play] [Refresh] [Delete] | ... |
| <input checked="" type="checkbox"/> example_complex <small>example example2 example3</small> | airflow | 1 (Success) | None | 2020-10-26, 21:08:04 | 37 (Success) | [Play] [Refresh] [Delete] | ... |
| <input checked="" type="checkbox"/> example_external_task_marker_child <small>example</small> | airflow | 1 (Success) | None | 2020-10-26, 21:07:33 | 2 (Success) | [Play] [Refresh] [Delete] | ... |
| <input checked="" type="checkbox"/> example_external_task_marker_parent <small>example</small> | airflow | 1 (Success) | None | 2020-10-26, 21:08:34 | 1 (Success) | [Play] [Refresh] [Delete] | ... |
| <input checked="" type="checkbox"/> example_kubernetes_executor <small>example example2</small> | airflow | 0 | None | | 0 | [Play] [Refresh] [Delete] | ... |
| <input checked="" type="checkbox"/> example_kubernetes_executor_config <small>example example3</small> | airflow | 1 (Success) | None | 2020-10-26, 21:07:40 | 5 (Success) | [Play] [Refresh] [Delete] | ... |
| <input checked="" type="checkbox"/> example_nested_branch_dag <small>example</small> | airflow | 1 (Success) | @daily | 2020-10-26, 21:07:37 | 9 (Success) | [Play] [Refresh] [Delete] | ... |
| <input type="checkbox"/> example_passing_params_via_test_command <small>example</small> | airflow | 0 | * / 1 * * * * | | 0 | [Play] [Refresh] [Delete] | ... |

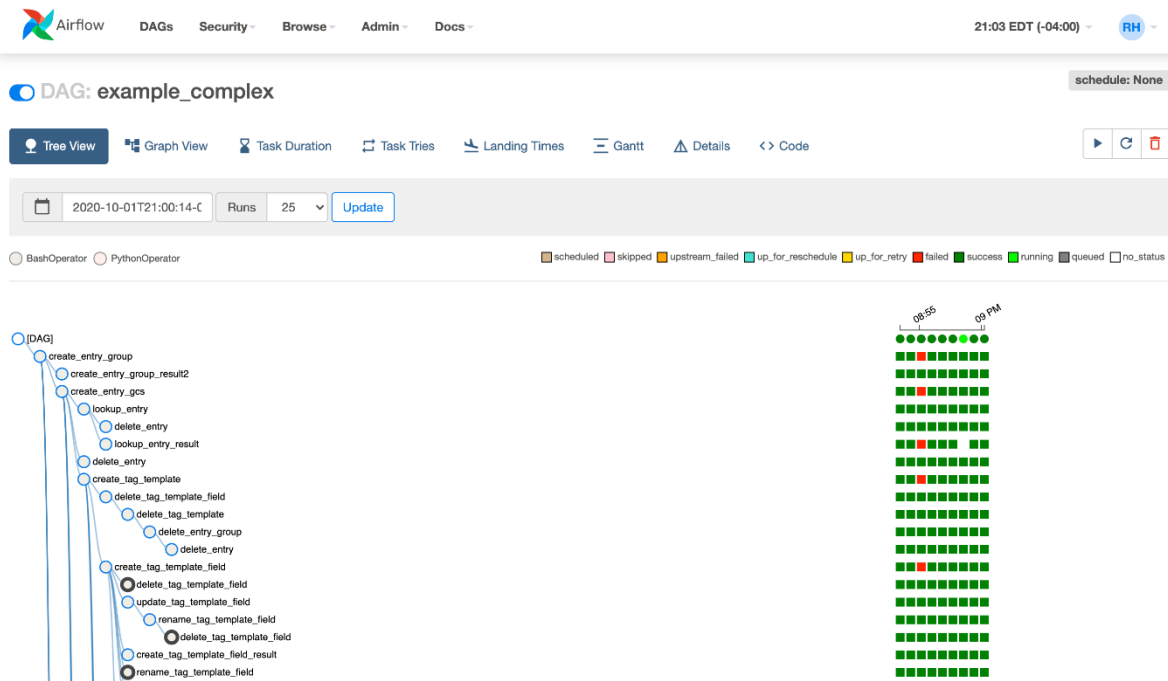
- **DAG** : Nom du DAG
- **Owner** : Propriétaire
- **Runs** : Statuts des précédentes exécutions de nos DAGs. Trois types d'états des exécutions de nos DAGs ainsi que le nombre d'exécutions.
 - Success (vert foncé)
 - Running (vert clair)
 - Failed (rouge)
- **Schedule** : Le type de planification
- **Last Run** : Dernière date de lancement
- **Récents Tasks** : Statut des tâches des DAGs en cours d'exécution ou de la dernière exécution.
- **Actions** : Lancement, rafraichissement et suppression des DAGs.
- **Links** : Redirection vers un des outils de monitoring choisi (Tree view, Graph view, ...).

Dans les onglets au-dessus :

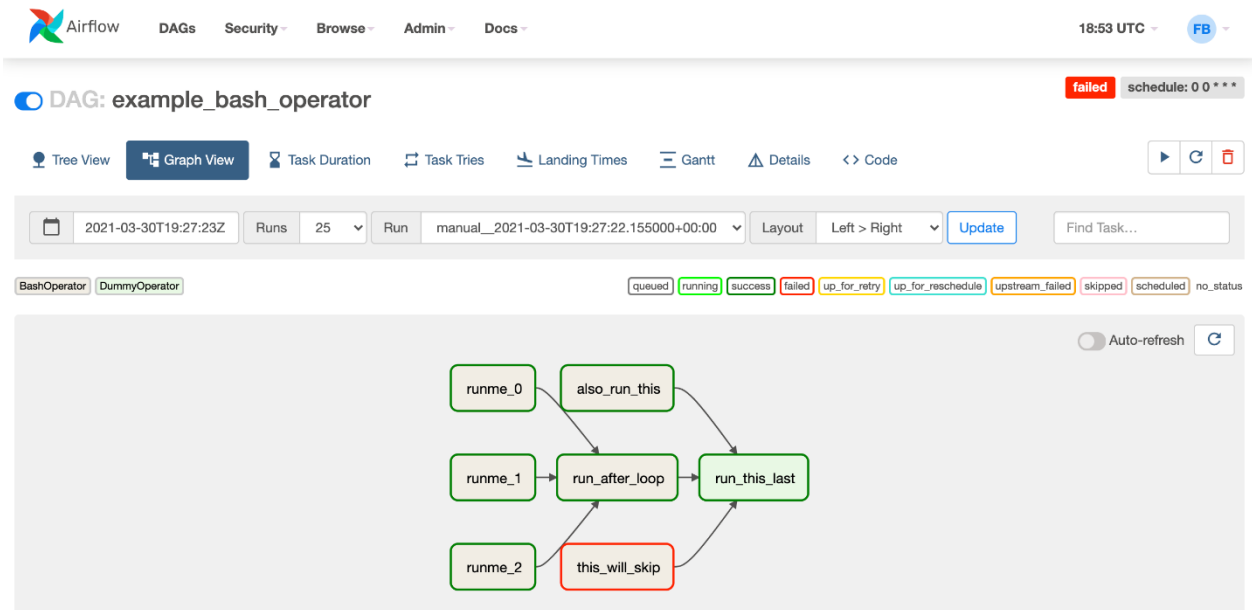
- **DAGs** : Affiche les DAGs (page d'accueil)
- **Security** : Gestion de la sécurité (utilisateurs, rôles, permissions...)

- **Browse** : Permet d'avoir les informations sur les DAGs en cours, les logs, les TaskInstances ...
- **Admin** : Paramétrages et configuration d'Airflow (connexions, plugins...)
- **Docs** : Liens vers la documentation générale

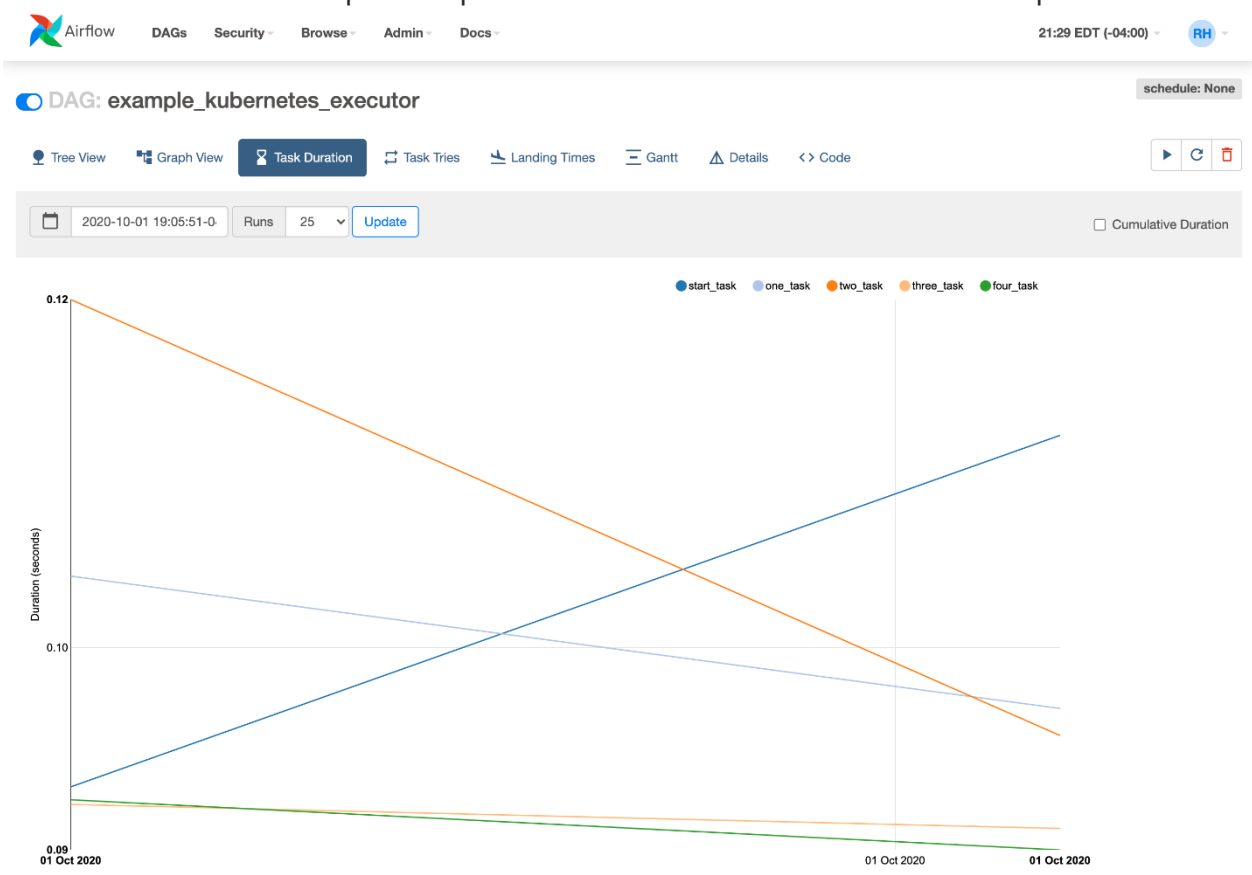
Vue arborescente : représentation arborescente d'un DAG qui s'étend dans le temps.



Vue graphique : visualisation des dépendances d'un DAG et de leur état actuel pour une exécution spécifique.



Durée de la tâche : temps total passé sur différentes tâches au fil du temps.



Affichage du code : moyen rapide d'afficher le code source d'un DAG.

Airflow DAGs Security Browse Admin Docs 21:30 EDT (-04:00) RH

DAG: example_bash_operator schedule: 0 0 * * *

Tree View Graph View Task Duration Task Tries Landing Times Gantt Details <> Code

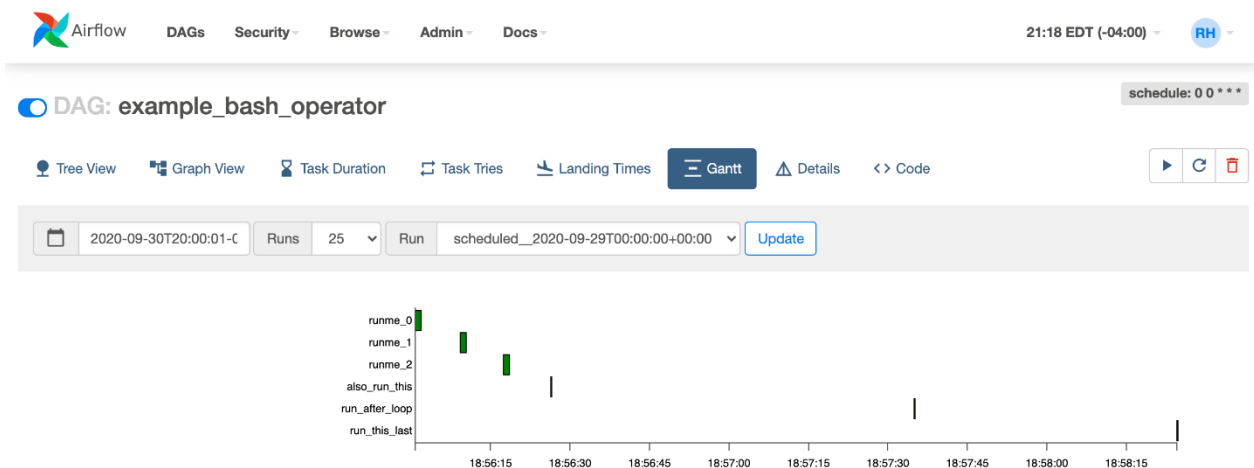
```

1 #
2 # Licensed to the Apache Software Foundation (ASF) under one
3 # or more contributor license agreements. See the NOTICE file
4 # distributed with this work for additional information
5 # regarding copyright ownership. The ASF licenses this file
6 # to you under the Apache License, Version 2.0 (the
7 # "License"); you may not use this file except in compliance
8 # with the License. You may obtain a copy of the License at
9 #
10 # http://www.apache.org/licenses/LICENSE-2.0
11 #
12 # Unless required by applicable law or agreed to in writing,
13 # software distributed under the License is distributed on an
14 # "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
15 # KIND, either express or implied. See the License for the
16 # specific language governing permissions and limitations
17 # under the License.
18
19 """Example DAG demonstrating the usage of the BashOperator."""
20
21 from datetime import timedelta
22
23 from airflow import DAG
24 from airflow.operators.bash import BashOperator
25 from airflow.operators.dummy_operator import DummyOperator
26 from airflow.utils.dates import days_ago
27
28 args = {
29     'owner': 'airflow',
30 }
31
32 dag = DAG(
33     dag_id='example_bash_operator',
34     default_args=args,
35     schedule_interval='0 0 * * *'.

```

Toggle Wrap

Vue Gantt : durée et chevauchement d'un DAG.



/

La partie 3.3 (p.13) du rapport, explique l'environnement d'Airflow.

Ce [lien](#) donnera plus de détail à quant à l'utilisation d'Airflow.

Pour ajouter une connexion Mysql :

Se rendre dans l'onglet Admin, puis Connections et Add a new record :

- Conn Id = mysql_connexion
- Conn Type = MySQL
- Host = mysql
- Schema = airflow
- Login = root
- Password = root
- Port = 3306