

Centre de Calcul
de l'Institut National de Physique Nucléaire
et de Physique des Particules

Découverte de Apache Airflow

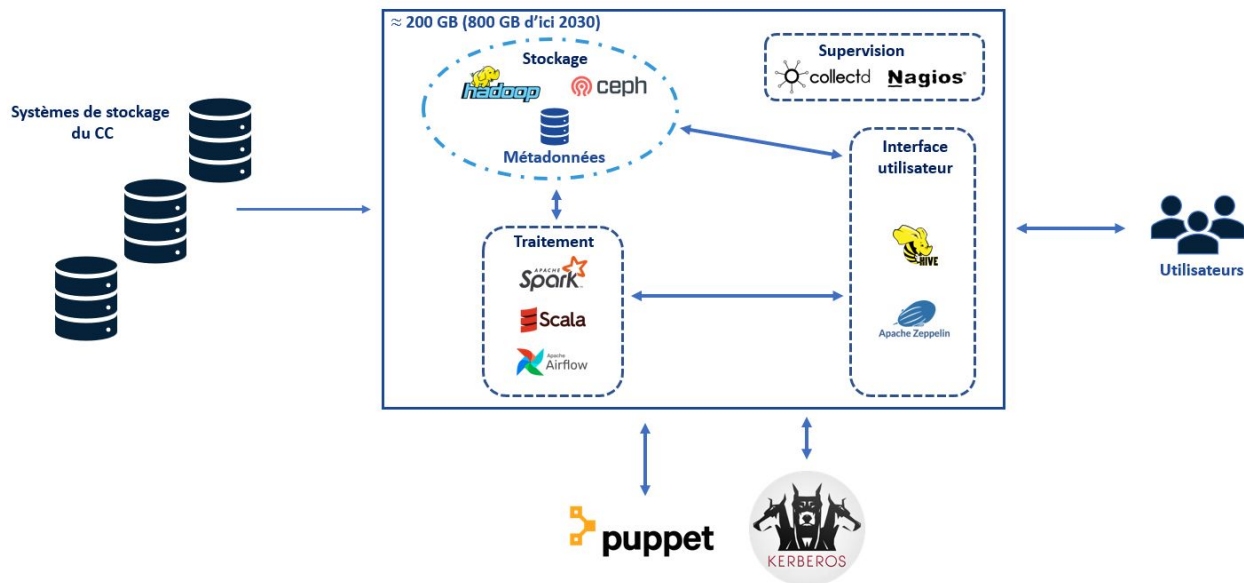
14e Journées Informatiques IN2P3/IRFU

Auteur : Florian Vernotte
Co-auteur : Osman Aïdel

- Contexte
- Présentation de Airflow
- Cas d'utilisations
- Bilan

Le projet XLDP

Offrir une vue centralisée et consolidée de l'utilisation des systèmes de stockage du CC-IN2P3



Problème :

Des flux de travail trop nombreux (+ de 200) et complexes pour être traités avec de simples cronjobs.

Solution :



- ➔ Dépendance de tâches entre elles
- ➔ Supervision des tâches facilitée

Créé en 2014 par Maxime Beauchemin



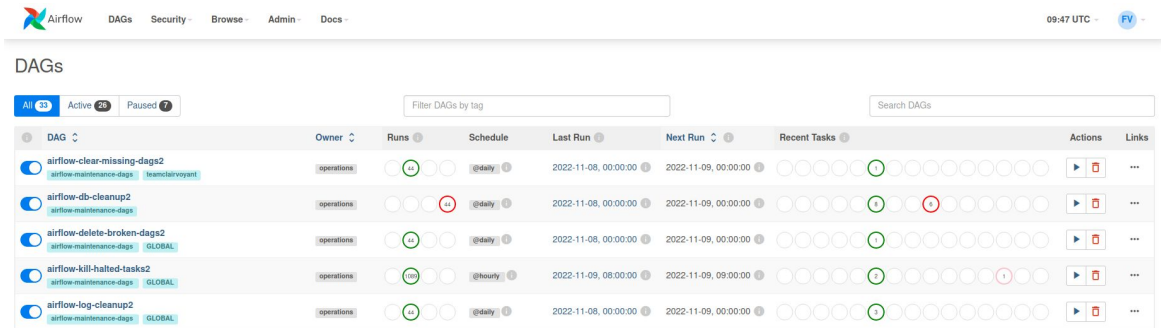
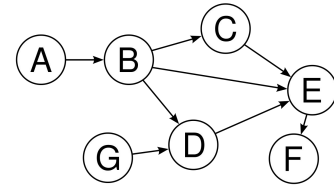
Incubé par la fondation Apache depuis 2016
Projet top level depuis 2019



Airflow est une plateforme permettant de créer, planifier et surveiller des flux de travail (workflow)



- Codée en Python pur
- Workflow modélisés sous forme de DAG (Directed Acyclic Graph) composé de tâches
- Web UI



DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
airflow-clear-missing-dags2 <small>airflow-maintenance-dags francislafoyest</small>	operations	12	@daily	2022-11-08, 00:00:00	2022-11-09, 00:00:00	12	▶ ⏹	...
airflow-db-cleanup2 <small>airflow-maintenance-dags</small>	operations	11	@daily	2022-11-08, 00:00:00	2022-11-09, 00:00:00	11	▶ ⏹	...
airflow-delete-broken-dags2 <small>airflow-maintenance-dags GLOBAL</small>	operations	10	@daily	2022-11-08, 00:00:00	2022-11-09, 00:00:00	10	▶ ⏹	...
airflow-kill-halted-tasks2 <small>airflow-maintenance-dags GLOBAL</small>	operations	9	@hourly	2022-11-09, 08:00:00	2022-11-09, 09:00:00	9	▶ ⏹	...
airflow-log-cleanup2 <small>airflow-maintenance-dags GLOBAL</small>	operations	8	@daily	2022-11-08, 00:00:00	2022-11-09, 00:00:00	8	▶ ⏹	...

Propriétés minimales d'un DAG dans Airflow :

- dag_id : nom identifiant uniquement un workflow
- start_date : date à partir de laquelle s'applique le dag
- schedule_interval : interval de temps entre chaque exécution

DAG : succession de tasks



Task

💡 Un dag est défini dans un fichier Python

Les noeuds d'un DAG sont appelés des tasks

 Idempotence

Une task peut être de deux types :

- Soit active (operator) ➔ elle effectue un traitement (ex: déplacer un fichier)
- Soit suspensive (sensor) ➔ elle attend qu'une condition soit satisfaite (ex: attendre la réception d'un message)

Différents états possibles pour une task

queued

running

success

failed

up_for_retry

up_for_reschedule

upstream_failed

skipped

scheduled

deferred

```
67 def list_new_file(**kwargs):
68     """
69     This task uses Variable FS_CONF_SWITCH to check all input folder for new file matching pattern.
70     It returns a list dict formatted like this [{},{ }].
71     :param kwargs:
72     :return: list returns a list dict formatted like this [{fs_name,date_of_file},{fs_name,date_of_file}].
73     """
74     import logging
75     fs_switch = Variable.get('FS_CONF_SWITCH', deserialize_json=True)
76     etl_conf_list = []
77
78     for name, values in fs_switch.items():
79         fs_conf = Variable.get(values, deserialize_json=True)
80         if 'FOLDER_INPUT_PATH' in fs_conf:
81             files = os.listdir(fs_conf['FOLDER_INPUT_PATH'])
82
83             if "INPUT_FILE_PATTERN_ARRAY" in fs_conf:
84                 extracted_date = check_file_validity(fs_conf['INPUT_FILE_REGEX'],
85                                                       files,
86                                                       fs_conf['DATE_FORMAT'],
87                                                       fs_conf['INPUT_FILE_PATTERN_ARRAY'])
88
89             else:
90                 extracted_date = check_file_validity(fs_conf['INPUT_FILE_REGEX'],
91                                                       files,
92                                                       fs_conf['DATE_FORMAT'],
93                                                       fs_conf['INPUT_FILE_PATTERN'])
94
95             for date in extracted_date:
96                 logging.info('ETL for FS {} for date {} should be treated.'.format(name, date))
97                 etl_conf_list.append({'date': date, 'fs': name})
98
99     unique_etl_conf_list = [ dict(s) for s in set( frozenset( myObject.items() ) for myObject in etl_conf_list ) ]
100     kwargs["ti"].xcom_push(key='conf_etl_list', value=unique_etl_conf_list)
101     logging.debug('The following ETL will be schedule: {}'.format(unique_etl_conf_list))
102     return unique_etl_conf_list
```

Définition de l'operator

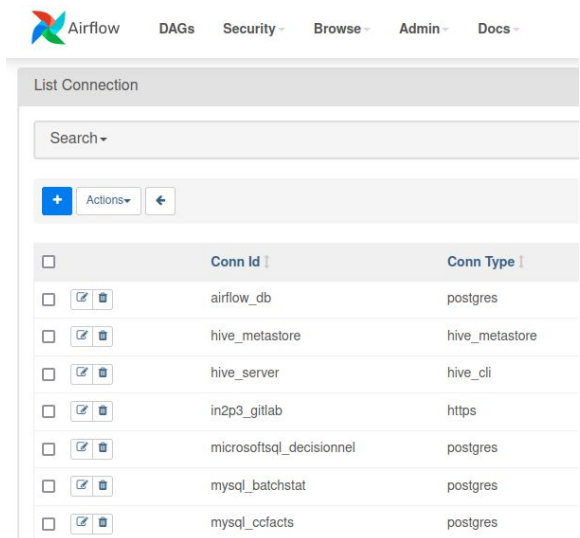
```
list_new_file = PythonOperator(task_id='list_new_file',
                               python_callable=list_new_file,
                               dag=dag)
```

Utilisation de variables communes à toutes les tasks

Transfert de données entre les tasks via XCOM

- **Opérateurs d'action** : BashOperator, PythonOperator, EmailOperator
- **Opérateurs de transfert** : Transfert de donnée d'une source vers une destination
- **Sensors** : Opérateur permettant de déclencher une action si une condition est satisfaite

Connections



Airflow DAGs Security Browse Admin Docs

List Connection

Search

+ Actions

<input type="checkbox"/>	Conn Id	Conn Type
<input type="checkbox"/>	airflow_db	postgres
<input type="checkbox"/>	hive_metastore	hive_metastore
<input type="checkbox"/>	hive_server	hive_cli
<input type="checkbox"/>	in2p3_gitlab	https
<input type="checkbox"/>	microsoftsql_decisionnel	postgres
<input type="checkbox"/>	mysql_batchstat	postgres
<input type="checkbox"/>	mysql_ccfacts	postgres

Hooks

Interface permettant d'échanger des données avec des systèmes extérieurs
ex : PostgreSQL, Hive, S3, ...

User interface : les DAGs





DAGs

All 16 Active 10 Paused 6

ETL

Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
 decisionnel_slurm_job decisionnel ETL slurm symod	airflow	 47	10 6 ***	2022-11-09, 05:10:00	2022-11-10, 05:10:00	 6	 	...
 etl_check_new_files ETL scheduler xldp	Airflow	 1090	45 *****	2022-11-10, 08:45:00	2022-11-10, 09:45:00	 2	 	...
 etl_clean_old_file ETL xldp	Airflow	 44	12 16 ***	2022-11-08, 16:12:00	2022-11-09, 16:12:00	 2	 	...
 etl_set_variables ETL variable xldp	Airflow	 45	15 00 ***	2022-11-09, 00:15:00	2022-11-10, 00:15:00	 7	 	...
 etl_spark_run ETL spark xldp	airflow	 2824  142	None	2022-11-10, 05:45:15		 9	 	...

User interface : Execution



Auto-refresh Hide Details Panel

DAG etl_spark_run / Run etl-pbs_throng-2022-11-09 / Task check_configuration

Task Instance Details Rendered Template **Log** XCom List Instances, all runs Filter Upstream

Task Actions

Ignore All Deps Ignore Task State Ignore Task Deps **Run**

Past Future Upstream Downstream Recursive Failed **Clear**

Past Future Upstream Downstream **Mark Failed**

Past Future Upstream Downstream **Mark Success**

Download Log (by attempts): 1

Status: ■ success

Task Id: check_configuration [🔗](#)
Run Id: etl-pbs_throng-2022-11-09 [🔗](#)
Operator: ShortCircuitOperator

Duration: 00:00:00
Started: 2022-11-10, 05:45:16 UTC
Ended: 2022-11-10, 05:45:17 UTC

DAG: etl_spark_run

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code Audit Log

Task Instance: load_configuration at 2022-11-10, 05:45:15

Task Instance Details <> Rendered Template **Log** XCom

Log by attempts

1

```
*** Reading local file: /ceph/xldp/log/airflow/dag_id=etl_spark_run/run_id=etl-pbs_throng-2022-11-09/task_id=load_configuration/attempt=1.log
[2022-11-10, 06:45:17 UTC] (taskinstance.py:1159) INFO - Dependencies all met for <TaskInstance: etl_spark_run.load_configuration etl-pbs_throng-2022-11-09 [queued]>
[2022-11-10, 06:45:17 UTC] (taskinstance.py:1159) INFO - Dependencies all met for <TaskInstance: etl_spark_run.load_configuration etl-pbs_throng-2022-11-09 [queued]>
[2022-11-10, 06:45:17 UTC] (taskinstance.py:1356) INFO -
-----
[2022-11-10, 06:45:17 UTC] (taskinstance.py:1357) INFO - Starting attempt 1 of 9
[2022-11-10, 06:45:17 UTC] (taskinstance.py:1358) INFO -
-----
[2022-11-10, 06:45:17 UTC] (taskinstance.py:1377) INFO - Executing <Task(PythonOperator): load_configuration> on 2022-11-10 05:45:15.327503+00:00
[2022-11-10, 06:45:17 UTC] (standard_task_runner.py:52) INFO - Started process 9798 to run task
[2022-11-10, 06:45:17 UTC] (standard_task_runner.py:79) INFO - Running: ['airflow', 'tasks', 'run', 'etl_spark_run', 'load_configuration', 'etl-pbs_throng-2022-11-09', '--job-id',
[2022-11-10, 06:45:17 UTC] (standard_task_runner.py:80) INFO - Job 958308: Subtask load_configuration
[2022-11-10, 06:45:17 UTC] (task_command.py:369) INFO - Running <TaskInstance: etl_spark_run.load_configuration etl-pbs_throng-2022-11-09 [running]>
[2022-11-10, 06:45:18 UTC] (taskinstance.py:1569) INFO - Exporting the following env vars:
```

User interface : Graph

DAG: etl_spark_run

running Schedule: None Next Run: None

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details Code Audit Log

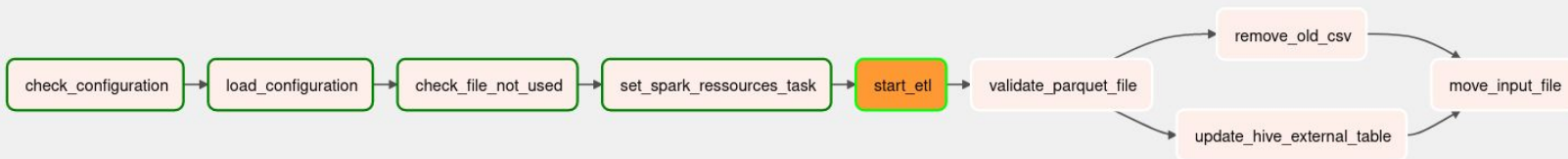
2022-11-10T10:45:17Z Runs 25 Run etl-pbs_home-2022-11-09 Layout Left > Right Update

Find Task...

BetterSparkOperator PythonOperator ShortCircuitOperator

queued running success failed up_for_retry up_for_reschedule upstream_failed skipped scheduled deferred no_status

Auto-refresh



User interface : Variables / Connections

Airflow DAGs Security Browse Admin Docs 10:44 UTC FV

List Connection

Search -

Record Count: 14

Actions

<input type="checkbox"/>	Conn Id	Conn Type	Description
<input type="checkbox"/>	postgres_bigdata_2	postgres	
<input type="checkbox"/>	mysql_cdfacts	postgres	
<input type="checkbox"/>	mysql_batchstat	postgres	
<input type="checkbox"/>	openidm_in2p3	http	openidm spark account
<input type="checkbox"/>	in2p3_github	https	connection description
<input type="checkbox"/>	microsoftsql_decisionnel	postgres	Decisionnel DB Connection
<input type="checkbox"/>	airflow_db	postgres	Postgres DB used by airflow itself
<input type="checkbox"/>	oracle_symod	postgres	SYMOD DB connection
<input type="checkbox"/>	postgres_bigdata	postgres	Postgres DB used to store storage stat from spark

Airflow DAGs Security Browse Admin Docs 10:43 UTC FV

Parcourir... Aucun fichier sélectionné. Import Variables

List Variable

Search -

Record Count: 215

Page size Actions







<input type="checkbox"/>	Key	Val	Description	Is Encrypted
<input type="checkbox"/>	airflow_db_cleanup__max_db_e...	45		True
<input type="checkbox"/>	airflow_log_cleanup__enable_de...	True		True
<input type="checkbox"/>	airflow_log_cleanup__max_log_...	45		True
<input type="checkbox"/>	DCACHE_EGEE_CONFIGURAT...	{ "INPUT_FILE_REGEX": "dc...		True
<input type="checkbox"/>	DCACHE_HIVE_QL	CREATE EXTERNAL TABLE ...		True
<input type="checkbox"/>	DCACHE_LCG_CONFIGURATION	{ "INPUT_FILE_REGEX": "dc...		True
<input type="checkbox"/>	DECI_JAR	/ceph/xldp/data/decisionnel/...		True
<input type="checkbox"/>	decisionnel_path	/ceph/xldp/data/decisionnel/a...		True
<input type="checkbox"/>	ETL_SPARK_COMMON_CONF	{ "spark.conf.entryName": "...		True
<input type="checkbox"/>	FS_CONF_SWITCH	{ "docache_log": "DCACHE_L...		True
<input type="checkbox"/>	HPSS_CONFIGURATION	{ "INPUT_FILE_REGEX": "hp...		True
<input type="checkbox"/>	HPSS_HIVE_QL	CREATE EXTERNAL TABLE ...		True

User interface : Gestion des rôles


List Roles

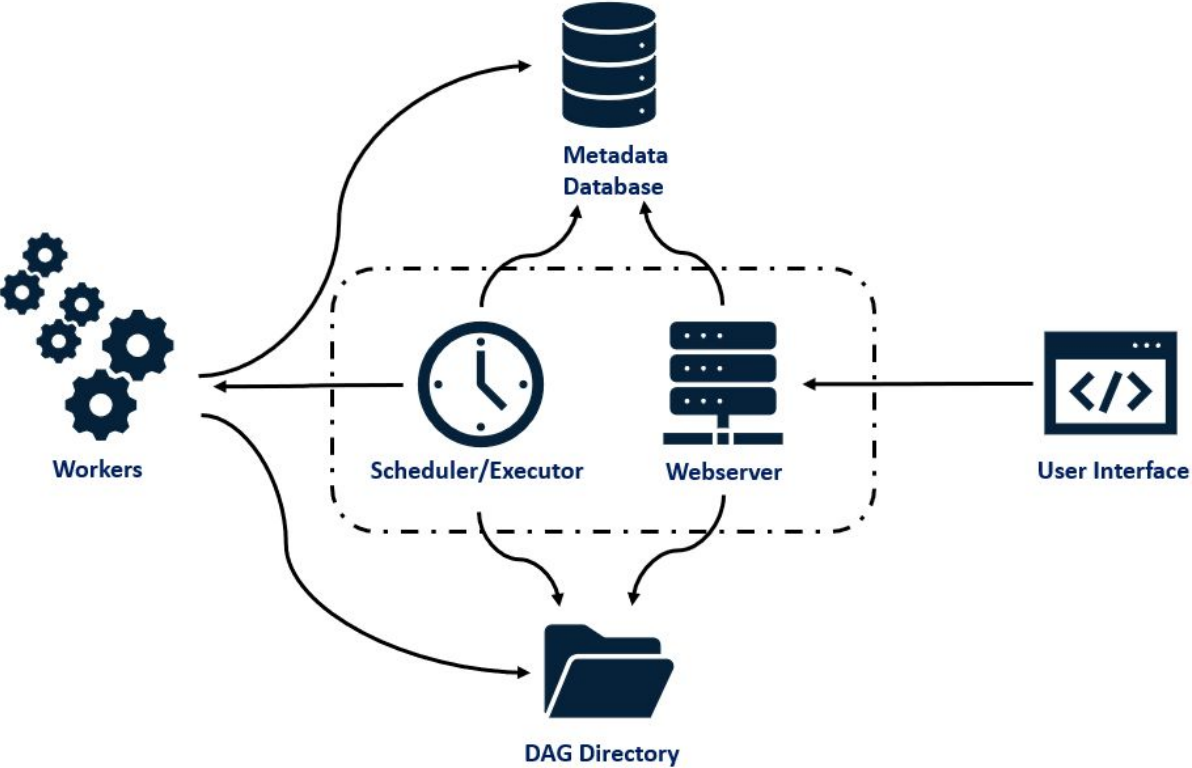
Search

[+](#) Actions [←](#) Record Count: 6

<input type="checkbox"/>	Name	Permissions
<input type="checkbox"/>   	Viewer	[can read on Audit Logs, can read on DAGs, can read on DAG Dependencies, can read on DAG Code, can read on DAG Runs, can read on ImportError, can read on Jobs, can read on My Password, can edit on My Password, can read on My Profile, can edit on My Profile, can read on Plugins, can read on SLA Misses, can read on Task Instances, can read on Task Logs, can read on XComs, can read on Website, menu access on Browse, menu access on DAG Runs, menu access on Documentation, menu access on Docs, menu access on Jobs, menu access on Audit Logs, menu access on Plugins, menu access on SLA Misses, menu access on Task Instances, menu access on DAG Dependencies]
<input type="checkbox"/>   	User	[can read on Audit Logs, can read on DAGs, can read on DAG Dependencies, can read on DAG Code, can read on DAG Runs, can read on ImportError, can read on Jobs, can read on My Password, can edit on My Password, can read on My Profile, can edit on My Profile, can read on Plugins, can read on SLA Misses, can read on Task Instances, can read on Task Logs, can read on XComs, can read on Website, menu access on Browse, menu access on DAG Runs, menu access on Documentation, menu access on Docs, menu access on Jobs, menu access on Audit Logs, menu access on Plugins, menu access on SLA Misses, menu access on Task Instances, can edit on DAGs, can delete on DAGs, can create on Task Instances, can edit on Task Instances, can delete on Task Instances, can create on DAG Runs, can edit on DAG Runs, can delete on DAG Runs, menu access on DAG Dependencies]



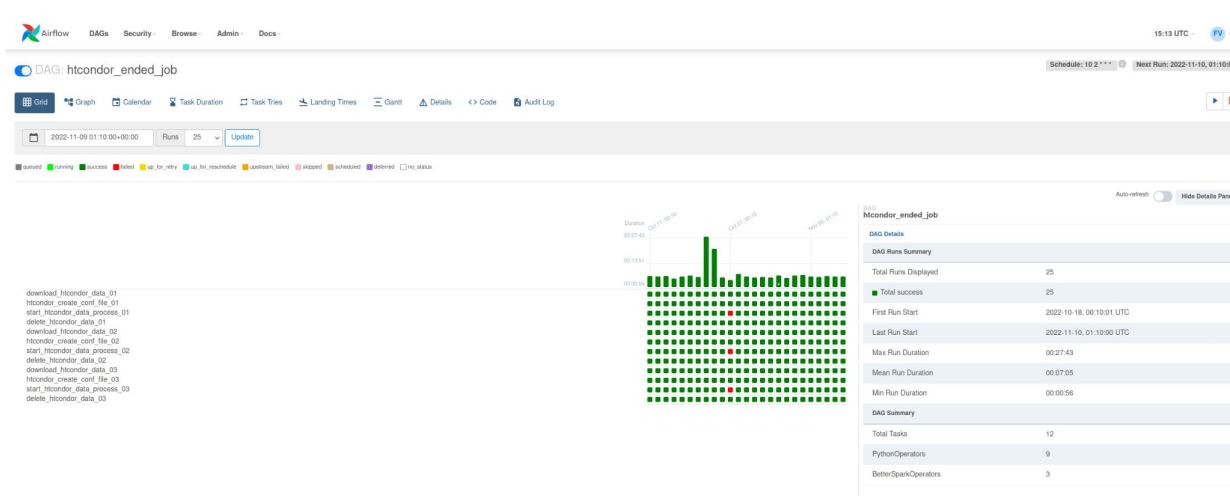
<input type="checkbox"/>   	Florian	Vernotte	vernotte	True	[Admin]
--	---------	----------	----------	------	---------



- Création de pipelines de données
- Gestion de workflow type ETL (Extract Transform Load)
- Planification et coordination des tâches (ex : jobs Spark)

Équipe décisionnel

Utilisation du Airflow de la plateforme XLDP pour générer les statistiques sur le batch du CC-IN2P3



 **Projet open source**
(Communauté très active, documentation fournie)

 **Simple d'utilisation**
(Consultation des logs depuis l'UI, réexécution de tasks échouées, ...)

 **Extensible**
(Possibilité de développer ses propres opérateurs, hooks)

 **Dépendance entre tasks**
(Ex : une task peut attendre la fin de l'exécution d'une autre)

 **Quelques défauts dans l'UI**
(Affichage des DAGs si ils sont nombreux)

 **Déploiement / maintenance**
(Mises à jour pas toujours simples à effectuer)



Merci pour votre attention
14e Journées Informatiques IN2P3/IRFU