Ce projet est cofinancé par l'Union européenne
L'EUROPE S'ENGAGE en région Auvergne-Rhône-Alpes avec le FSE
La Région Auvergne-Rhône-Alpes

ETL - Mise en production

ITÉRATION 1

Mise en place d'Airflow

OBJECTIF

- Mettre en place l'outil Airflow
- Création d'un premier DAG

1.0 — Introduction à Airflow

5 minutes— Présentiel

Maintenant que vous avez spécifié les traitements et écris un programme python permettant la collecte ou l'intégration de données.
Ces traitements vont être implémentés et intégrés dans ce que l'on appelle un scheduler, qui permet de planifier et de monitorer ce que l'on appelle des jobs.
Un outil souvent utilisé pour faire cela se nomme Airflow, il a été développé par Airbnb et est depuis utilisé par beaucoup d'autres entreprises.
Airflow manipule les jobs en créant ce que l'on appelle des DAGs (Directed Acyclic Graph) qui contiennent les tâches et sont utilisés par Airflow pour les gérer, et contient beaucoup d'utilitaires qui pourront vous être utiles…

RESSOURCES

- Vue d'ensemble de Airflow: https://www.youtube.com/watch?v=Aw0y94BJ01U

1.1 — Installation de Airflow

3 heures— Présentiel

Pour utiliser Airflow nous allons devoir l'installer. Airflow étant un paquet python, l'installation devrait être rapide, la configuration, en revanche, peut prendre un peu plus longtemps .

Pour la suite vous aurez probablement besoin d'ouvrir plusieurs terminals

Consignes

- Dans l'environnement python utilisé pour faire l'ETL, installez le paquet apache-airflow
- Créez une variable d'environnement nommée `AIRFLOW_HOME` qui contiendra le chemin ou la configuration de airflow et les dags seront stockés dans un dossier `dags`
- Utilisez les utilitaires de airflow pour initialiser airflow.
- Dans le fichier généré `airflow.cfg` définissez le répertoire principale et le répertoire où

se trouve les dags.

- Une fois le fichier rempli, exécutez les commandes suivantes:
  - `airflow webserver`, puis accéder à l'interface web d'administration à l'adresse suivante: http://localhost:8080/
  - `airflow scheduler`, vérifiez que tout s'exécute sans problème
- Si vous rencontrez des problèmes à cette étape, vous pouvez essayer la commande `airflow standalone` qui lance le web server et le scheduler en même temps.
- Écrivez votre premier DAG et exécutez le.
- Assurez-vous que les autres membres de votre îlot sont bien arrivés à installer airflow.

RESSOURCES

- Tutoriel très complet (en ligne de commande, la partie sur celery peut être sautée):
  - https://airflow-tutorial.readthedocs.io/en/latest/first-airflow.html
- Les variable d'environnement:
  - https://www.twilio.com/blog/how-to-set-environment-variables-html
- Installation de airflow
  - https://qiita.com/new-php/items/f8fda9258f77d2c7983e
  - https://www.geeksforgeeks.org/installation-guide/how-to-install-apache-airflow/
  - https://airflow.apache.org/docs/apache-airflow/stable/index.html
- Écrire un DAG
  - https://www.datacamp.com/fr/blog/what-is-a-dag
  - https://www.datacamp.com/fr/tutorial/getting-started-with-apache-airflow

LIVRABLES

❑ Airflow installé et un DAG peut s'exécuter

ITÉRATION 2

Création de pipelines d'ETL

OBJECTIF

- Créer des jobs de collecte de données dans airflow
- Créer des jobs d'intégrations de données dans airflow
- Gérer les pipelines dans Airflow

2.0 — Synchronisation

1h — Présentiel

Avant de vous lancer dans la création de vos dags personnalisés en utilisant airflow:

- Assurez vous que tout votre îlot a pu créer un DAG
- Définissez la structure du projet ainsi que la structure des DAGs, dans le dépôt git créé dans le kit 4.
- Mettez à jour votre dépôt github.

C'est partie, dans la suite il est recommandé de se répartir le travail afin de profiter avec peu de contraintes des avantages offerts par git et github.

2.1 —Découpage des tâches de l'ETL dans airflow

7h — Présentiel

Afin de passer les jobs python dans Airflow, il va être important de travailler sur l'enchaînement des tâches voulu, pour l'instant l'objectif est de passer l'ETL écrit à la main dans Airflow, ce qui permettra d'intégrer de nouveaux traitements au fil de l'eau.

Pour cela, un schéma va être créé:

Consignes

- Renseignez vous sur les différentes type de tâche et sur les DAGs
  - Qu'est-ce qu'une tâche de type Sensor?
- En utilisant un outil de dessin, dessinez le workflow voulu en nommant les types de tâche et les tâches utilisées.
- Comparez le workflow avec vos voisins
  - Modifiez le schéma pour qu'il prenne en compte les remarques de vos voisins

RESSOURCES

- Les tâches Airflow
  - https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/tasks.html
- Les DAG dans airflow:
  - https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/dags.html
- Quelques ressources en plus
  - https://marclamberti.com/blog/airflow-xcom/
  - https://medium.com/@MadhavPrajapati/deep-dive-into-passing-data-between-tasks-using-xcom-in-apache-airflow-21d719b71098
  - https://towardsdatascience.com/apache-airflow-for-data-science-how-to-run-tasks-in-parallel-32f7573068a/
  - https://blog.devgenius.io/airflow-task-parallelism-6360e60ab942
  - https://towardsdatascience.com/how-to-build-a-data-extraction-pipeline-with-apache-airflow-fa83cb8dbcdf

LIVRABLES

❑ Les DAGs dessinés dans l'outil de diagramme

2.2 —Création de l'ETL dans Airflow

7h — Présentiel

Le moment est venu de passer le code python dans airflow; les données seront stockées dans une base de données SQLite.

Consignes

- Ajoutez les jobs de scrapping dans airflow
- Ajoutez les jobs utilisant des apis dans airflow
- Ajoutez les transformations de données dans airflow

RESSOURCES

- Les tâches Airflow
  - https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/tasks.html
- Les DAG dans airflow:
  - https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/dags.html
- Quelques ressources en plus
  - https://towardsdatascience.com/how-to-build-a-data-extraction-pipeline-with-apache-airflow-fa83cb8dbcdf

Livrables

- ➔ Les DAG airflow implémentés et fonctionnels
- ➔ Les DAGs correspondent à la spécification