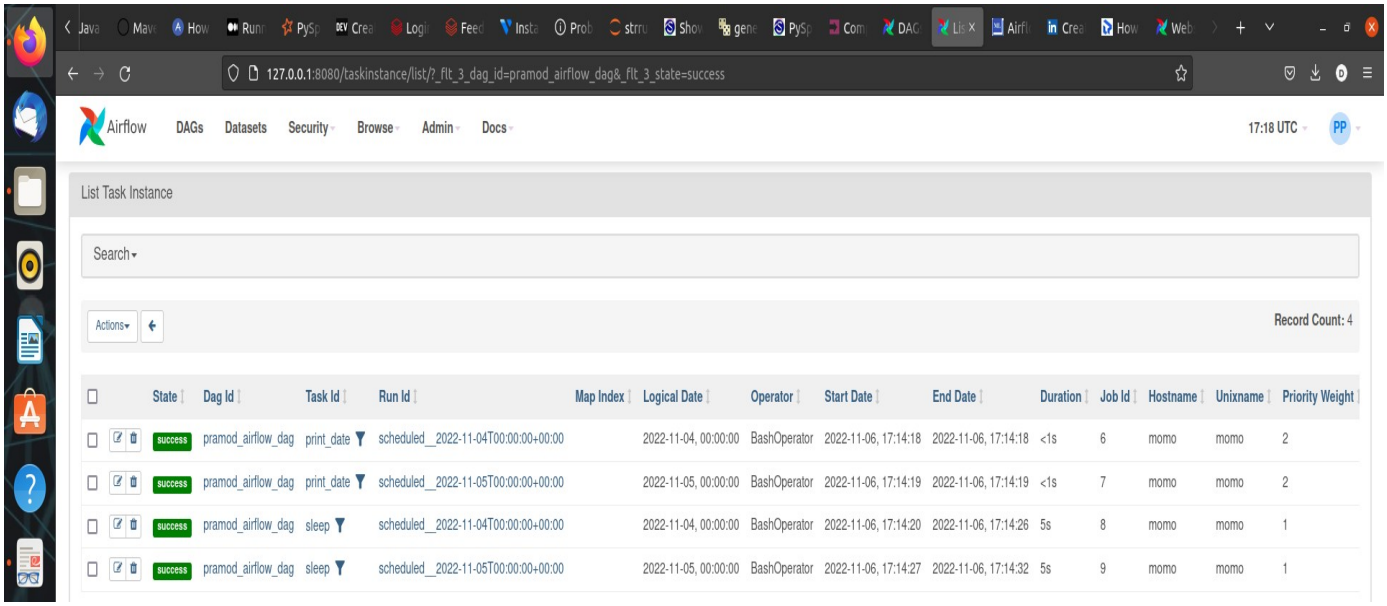
Pour ce chapitre nous avons vu le framwork **Airflow** et comment les jobs sont planifiés et exécutés à travers son core **DAG** qui est une collection de tâches à exécuter sur un ordre donné.
Pour pouvoir gérer un workflow avec **Airflow,** l'ensemble des tâches à exécuter sont définies dans le ficher DAG qui est un script python contenant l'ensemble des dépendances et taĉes à exécuter ainsi que l'ordre d'exécution des tâches.

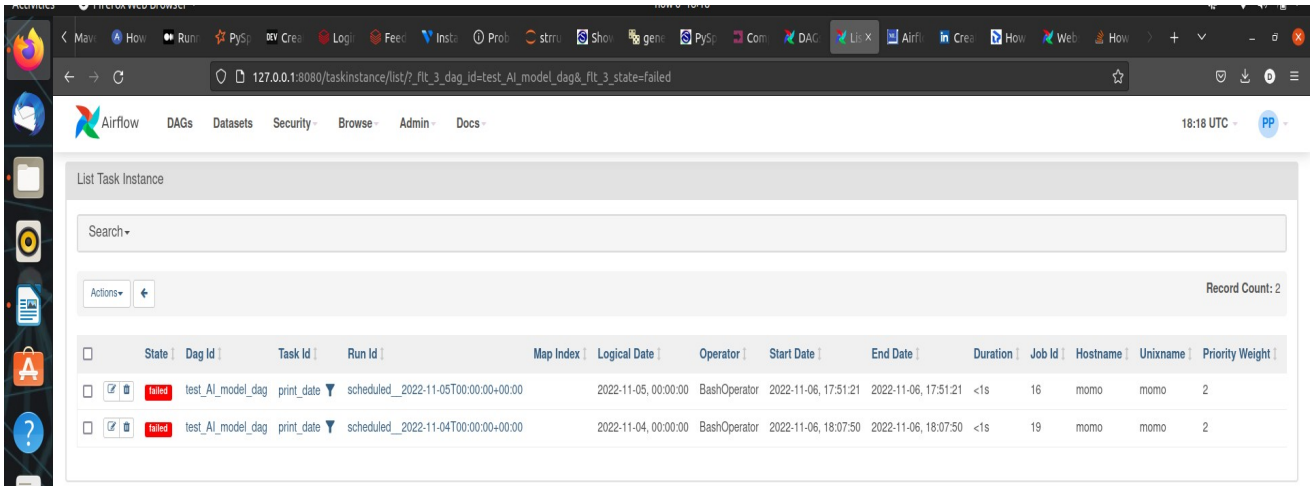
Pour cela nous avions la possibilité d'utiliser workflow avec le container Docker ou de l'installer en local pour ensuite l'utliser.
L'utilisation avec Docker nous a causé des soucis, après avoir pull l'image de workflow sur Docker Hub nous n'arrivions pas à voir le dossier airflow pour y mettre nos fichiers dag.
Donc nous avons utilisé la deuxième opttion qui est d'installer Airflow afin de pouvoir l'utiliser.

Pour cela, nous avons créer un environnement Python avec anaconda et ensuite activer ce dernier à travers la commande **conda activate** .
Ensuite nous avons installé les dépendances de Airflow avec la commande **apt-get install** et enfin nous avons utiliser la commande **pip** pour installer Airflow.
Ainsi nous avons créer un sous dossier dag, dans le répertoir airflow, qui va contenir l'ensemble de fichers dag et script bash, python, …

Nous avons fait un premier exemple assez simple à travers le ficher **first_dag.py** permettant juste d'afficher la date actuelle à travers une commande bash et le dag a été exécuté avec succes sur l'interface web de Airflow comme le montre l'image ci-dessous:



Contrairement à notre deuxième dag où seule la première tâche a été exécutée avec succes au moment où l'exécution de la deuxième tâche a échoué comme le montre l'image ci-dessous:



Sur cette image nous voyons bien que l'exécution de la tâche print_date, qui consiste à exécuter un script python testant un modèle IA, a échoué. Notre problème ici c'est de ne pas être en mesure de voir le problème à travers les logs sur l'interface web mais plutôt nous sommes obligé d'aller sous dossier logs du répertoire airflow.
Ainsi après avoir consulté les logs il se trouve qu'il y avait un problème avec le chemin vers notre modèle(No such file or directory error) et après correction la tâche print_date arrive à s'exécuter.