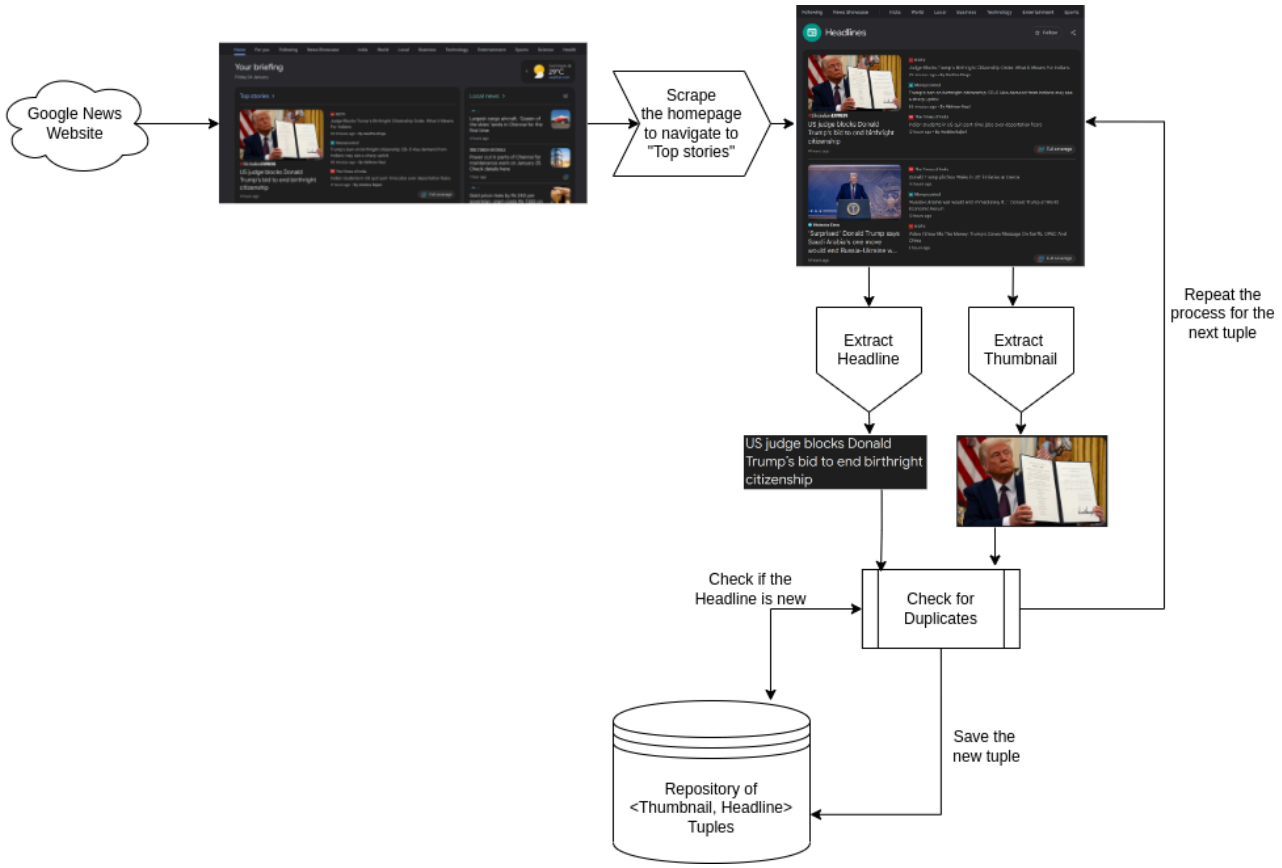
DA5402: Assignment #2

We got introduced to Apache Airflow today, which happens to be one of most popular open-source orchestration tool amongst the industry practitioners.

Let's rebuild our solution for Assignment #1 using Airflow this time. This exercise would teach you the power of using the right tool to the right problem. The workflow that we implemented in A1 is shown again here as a ready reckoner.



Module 1 [5 points]

Create a suitable Airflow Operator that would either import your Python function or directly execute your Python script to scrape the home page of Google News. The URLs to scrape may change over time, so you should keep them as configurable parameters or operator properties.

Module 2 [5 points]

Create a suitable Airflow Operator that would either import your Python function or directly executing your Python script to scrape the "Top stories" from the home page of Google News. The URLs to scrape may change over time, so you have to keep them as configurable parameters or operator properties.

Module 3 [10 points]

Create a suitable Airflow Operator that would either import your Python function or directly executing your Python script to extract the thumbnail and the headline of every story from that page. Remember that the page is set up for lazy loading. Your operator should factor lazy loading. The layout of the “Top stories” page may change over time, so ensure that you create your Operator in a way for easy updates later.

Module 4 [10 points]

Create a Postgres operator to setup your database tables with the necessary unique/primary keys. Your database should have one table for storing the image data (base64 encoded) and the other table for storing the headlines and other meta information such as URLs, scrape timestamp, article date, etc. You can run a join query with the image and headlines tables to fetch the <title, image> pair. After the “insert” operation, you should create a status file at “dags/run/status”, which should contain just one number representing the number of successful inserts. If all the records are duplicates, you will have a ‘0’ in the status file.

Module 5 [10 points]

Setup a Airflow DAG to orchestrate the workflow pipeline with necessary properties and load them up in the Airflow console. You should setup crontab, to make it runnable every hour. As long as the Airflow container is running in the background, you will keep scrapping the articles for <image, headline> tuple.

Module 6 [10 points]

Setup another Airflow DAG that would send an email to yourself, whenever a new <image, headline> tuple is added to the database. This DAG has to remember the previous state of the database tables to detect new rows. When news rows are detected, it should it’s knowledge of the previous state and also send an email to your inbox. To send emails, you may have to learn about SMTP + TLS/SSL setup with credentials. The set up is the same process when you configure your email client (Thunderbird, Outlook, etc) to send emails from your gmail or institute accounts. The only difference is you are going to do it programmatically.

Hint: *The sendmail DAG could even be one Airflow Operator. But, I want you to get creative with the modularization of the send email workflow.*

This send email DAG should get triggered by the first DAG through “FileSensor” on the status file. This means, when the first DAG completes, the sendmail DAG should get triggered. The Sendmail DAG will delete the status file upon completion.

Danger: When you submit your assignment, ensure you remove your credentials. The TAs will apply their credentials and email address to test your DAGs.

Note: You will submit only the DAG files (2 DAGs), as the rest of the environment is identical! When you visualize the DAGs are to appear as two separate workflows. The DAGs are coupled only via the status file.