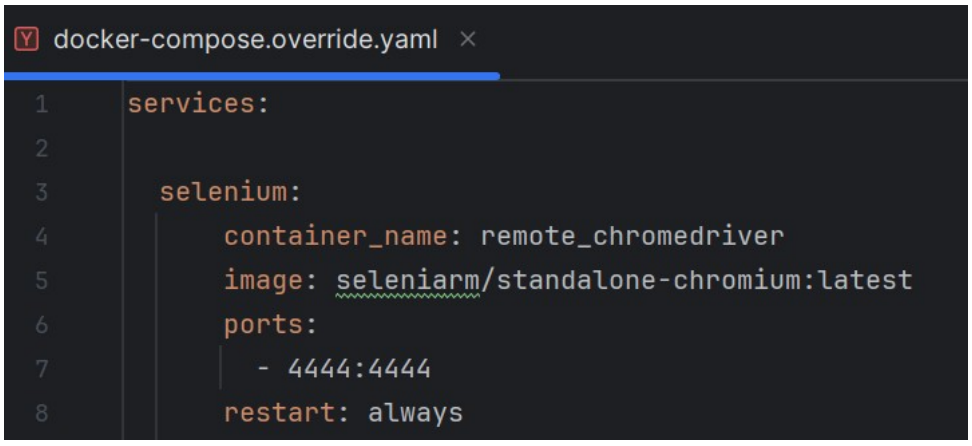
DA5402 - Assignment 2

Anirudh Rao be21b004

Airflow Initiation

In Assignment 1, Google News was scraped using a `selenium` Chrome web driver. To ensure that the web driver can be used inside the Airflow container, `selenium` is added as a service in a `docker-compose.override.yaml` file.

When '`docker compose up`' is run on the command line, Docker Compose will combine the original `docker-compose.yaml` file and the `docker-compose.override.yaml` file to initiate Airflow with a `selenium` remote Chrome web driver service at port 4444.



docker-compose.override.yaml

```yaml
services:

  selenium:
      container_name: remote_chromedriver
      image: seleniarm/standalone-chromium:latest
      ports:
        - 4444:4444
      restart: always
```

Figure 1: The `docker-compose.override.yaml` file

Module 1

First, a `BashOperator` was created to install (via `pip`) the necessary Python libraries for web scraping.

Next, a `PythonOperator` was created to import the `google_news_scraper.py` script that was written for Assignment 1. This script uses the `selenium` Chrome driver to navigate to the Google News homepage. The URL for this is not hard coded in the script and is instead easily configurable at the head of the DAG file `scraper_dag.py` written for scraping Google News. To account for lazy loading, a sleep time is incorporated to allow the website to load sufficient content for scraping. This sleep time is also configurable at the head of `scraper_dag.py`.

Module 2

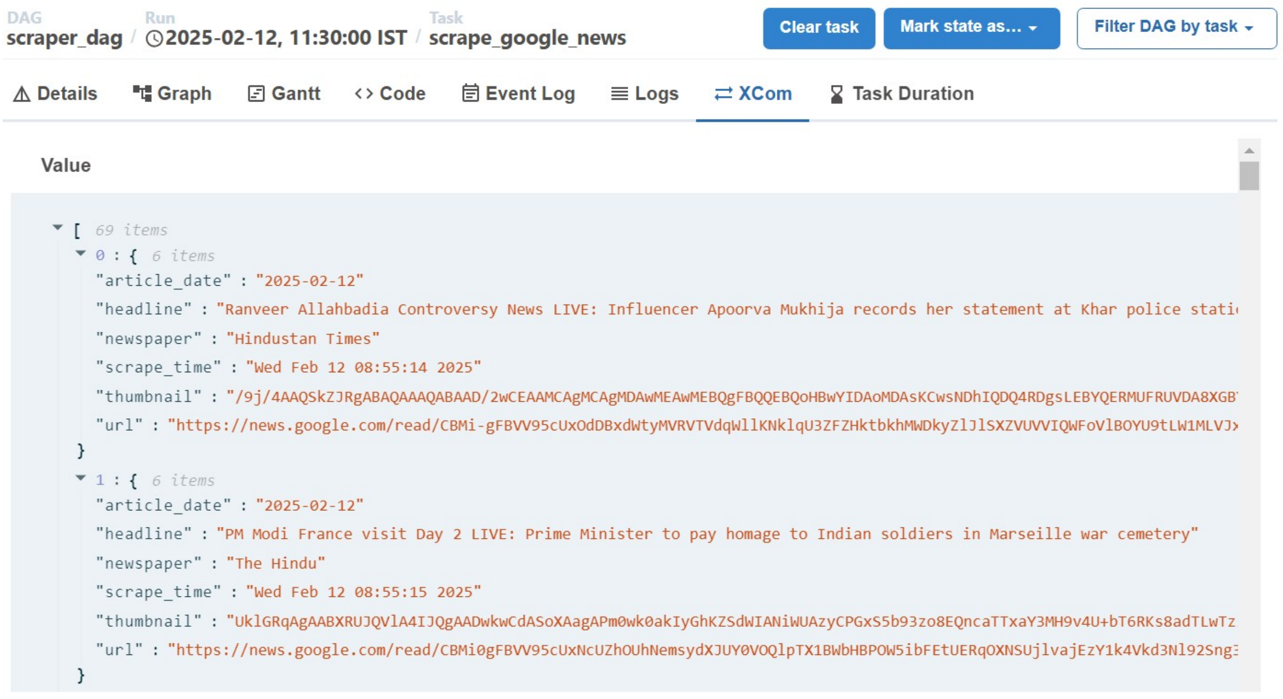
The `google_news_scraper.py` script then clicks on the ‘Top stories’ link. This link is not hard coded in the script and is instead easily configurable at the head of `scraper_dag.py`.

Module 3

`selenium` scrolls through the ‘Top stories’ page until the height of the webpage stays constant. This scrolling is performed using the configured sleep time, keeping in mind the lazy loading of the webpage. `selenium` then uses CSS selectors to scrape the headline, thumbnail, newspaper, article URL, and article date. To account for Google News changing its layout, these CSS selectors are not hard-coded and are instead configurable at the head of `scraper_dag.py`. The thumbnail is stored in a `base64` format. The original image can be obtained using the `PIL` library.

Module 4

A `PostgresOperator` is used to create a database table called ‘news’ using Airflow’s Postgres connection. Then, a `PythonOperator` is used to execute a script that pulls the scraped articles’ data from XCom and inserts it into the database. The headline is used as a de-duplication constraint to ensure only unique articles are entered into the database. After the insert is completed, a status file is written at `/temp/dags/run/status`. This contains just one number - the number of successful inserts. If all articles scraped are duplicates, the status file reads 0.



```
[ 69 items
  0 : { 6 items
    "article_date" : "2025-02-12"
    "headline" : "Ranveer Allahbadia Controversy News LIVE: Influencer Apoorva Mukhija records her statement at Khar police stati
    "newspaper" : "Hindustan Times"
    "scrape_time" : "Wed Feb 12 08:55:14 2025"
    "thumbnail" : "/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAMCAgMCAgMDAwMEAwMEBQgFBQQEBQoHBwYIDAoMDAsKCwsNDhIQDQ4RDgsLEBYQERMUFRUVDA8XGB
    "url" : "https://news.google.com/read/CBMi-gFBVV95cUxOdDBxdWtyMVRVTVdqWllKNklqU3ZFZHktbkhMWDkyZlJlSXZVUVVIQWFoVlBOYU9tLW1MLVJ
  }
  1 : { 6 items
    "article_date" : "2025-02-12"
    "headline" : "PM Modi France visit Day 2 LIVE: Prime Minister to pay homage to Indian soldiers in Marseille war cemetery"
    "newspaper" : "The Hindu"
    "scrape_time" : "Wed Feb 12 08:55:15 2025"
    "thumbnail" : "UklGRqAgAABXRUJQVlA4IJQgAADwkwCdASoXAagAPm0wk0akIyGhKZSdWIANiWUAzyCPGxS5b93zo8EQncaTTxaY3MH9v4U+bT6RKs8adTLwTz
    "url" : "https://news.google.com/read/CBMi0gFBVV95cUxNcUZhOUhNemsydXJUY0VOQlpTX1BWbHBPOW5ibFEtUERqOXNSUjlvajEzY1k4Vkd3Nl92Sng
  }
```

Figure 2: Data scraped in XCom

Module 5

The four operators are orchestrated using an Airflow DAG called `scraper_dag`. It is scheduled using cron to execute every hour (`0 * * * *`).

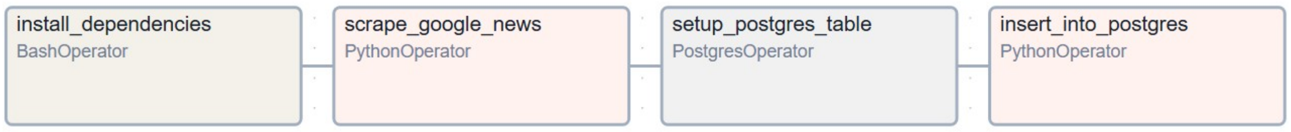


Figure 3: Graph for `scraper_dag`

Module 6

Another Airflow DAG called `email_dag` is created with filename `email_dag.py`. It uses a `FileSensor` to detect the presence of the status file is written at `/temp/dags/run/status`. It uses a poke interval of 1 minute and a timeout of 1 hour 1 minute, after which the task is denoted as 'failed'.

If detected, it uses a `PythonOperator` to check the Postgres database for new entries. A `json` file at `/tmp/previous_record_count.json` is used to remember the previous state of the database tables to detect new entries.

If new entries are detected (using the return value of the previous task in XCom), a `PythonOperator` uses the `email_sender.py` script to send an email to the email address mentioned in the script. This utilizes the `smtplib` and `ssl` libraries. The user will have to configure their email address and password in the `email_sender.py` script.

Finally, a `TriggerDagRunOperator` is used to trigger `email_dag` and reactivate the `FileSensor`. The process of file sensing, checking for new entries, and sending an email then occurs.

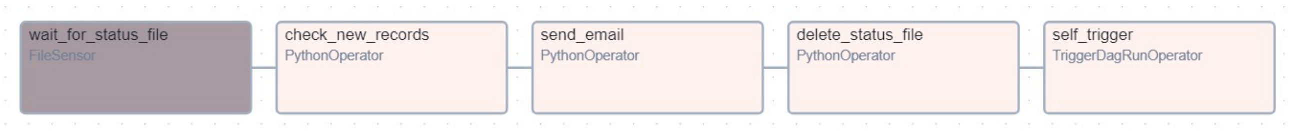
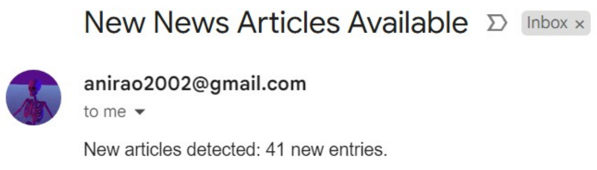


Figure 4: Graph for `email_dag`



New News Articles Available Inbox ×

anirao2002@gmail.com

to me

New articles detected: 41 new entries.

Figure 5: Example of an email sent