Report: Implementation of Apache Airflow DAG for Data Extraction, Transformation, and Storage

Objective:

The objective of this project was to implement an Apache Airflow DAG to automate the processes of data extraction, transformation, and storage. The tasks involved utilizing web scraping techniques to extract data from specified URLs, preprocessing the extracted data, and storing it in a CSV file. Additionally, the project aimed to integrate version control using Git and Data Version Control (DVC) to track changes to the data.

Tasks Completed:

1. Data Extraction:
    - Utilized web scraping techniques to extract data from the websites 'https://www.dawn.com/' and 'https://www.bbc.com/'.
    - Extracted links and article metadata (title, description) from the landing pages of the specified URLs.
    - Implemented error handling to ensure robustness during data extraction.
2. Data Transformation:
    - Preprocessed the extracted text data to clean and format it appropriately for further analysis.
    - Utilized regular expressions and string manipulation techniques to remove HTML tags and non-alphanumeric characters from the text data.
    - Standardized the text data to lowercase for consistency.
3. Data Storage and Version Control:
    - Saved the preprocessed data to a CSV file (extracted.csv) for storage.
    - Integrated Git and DVC to implement version control for the data.
    - Automated the process of pushing changes to the Git repository using Python functions within the DAG.

Workflow:

The workflow of the implemented Apache Airflow DAG is as follows:

1. Extract Task (extract_task): Extracts data from the specified URLs using web scraping techniques.
2. Preprocess Task (preprocess_task): Preprocesses the extracted data to clean and format it.
3. Save Task (save_task): Saves the preprocessed data to a CSV file (extracted.csv).
4. DVC Push Task (dvc_push_task): Integrates DVC to track versions of the data and pushes changes to the Git repository.
5. Git Push Task (git_push_task): Pushes changes to the Git repository for version control.

Challenges Encountered:

1. Dependency Management: Ensuring that all necessary Python libraries and dependencies were installed and compatible with Airflow.
2. Error Handling: Implementing robust error handling mechanisms to handle exceptions during data extraction and processing.
3. Integration with Git and DVC: Configuring Git and DVC to work seamlessly within the Airflow environment and automating the process of pushing changes.

Observations:

1. The use of Airflow provided a scalable and flexible framework for orchestrating the data pipeline, allowing for easy monitoring and management of tasks.
2. Integrating version control with Git and DVC enhanced the reproducibility and traceability of the data pipeline, facilitating collaboration and ensuring data integrity.

Conclusion:

The implementation of the Apache Airflow DAG for data extraction, transformation, and storage was successful in automating the specified tasks. The workflow provided a structured approach to handle the complexities of data processing and management, while the integration with version control systems added transparency and accountability to the process.

Recommendations:

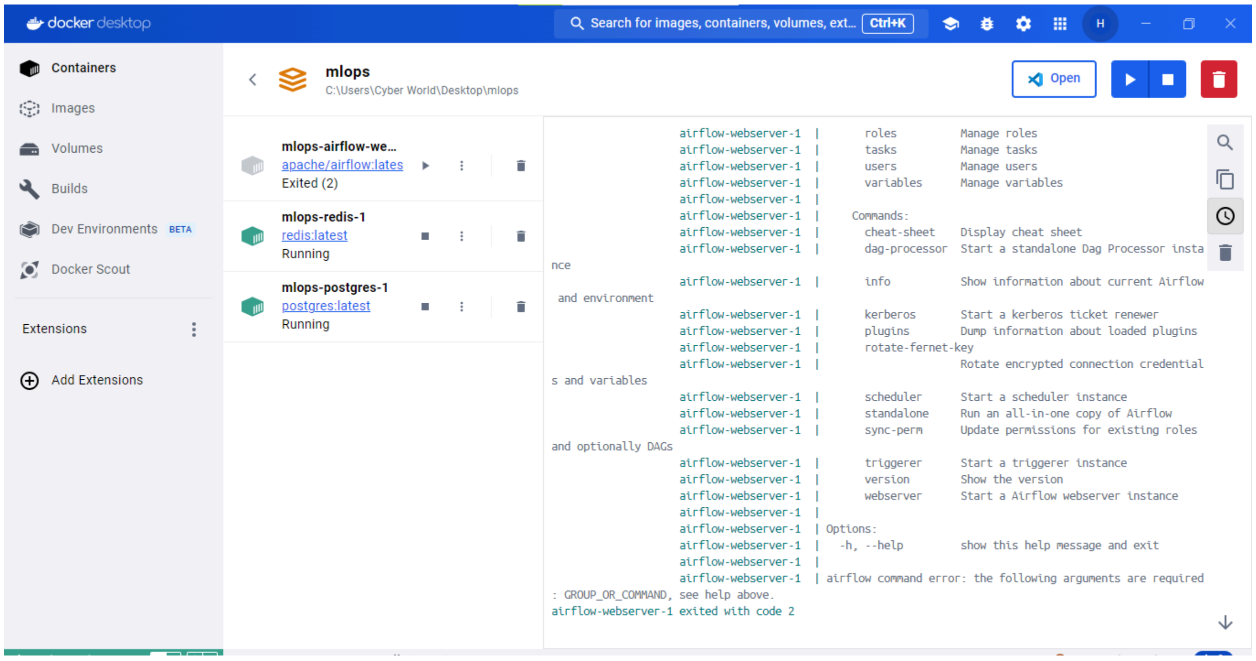
1. Continuously monitor and optimize the performance of the DAG to ensure efficient execution.

2. Explore additional features and capabilities of Airflow for advanced workflow orchestration and scheduling.
3. Document the DAG workflow, dependencies, and configurations for future reference and maintenance.

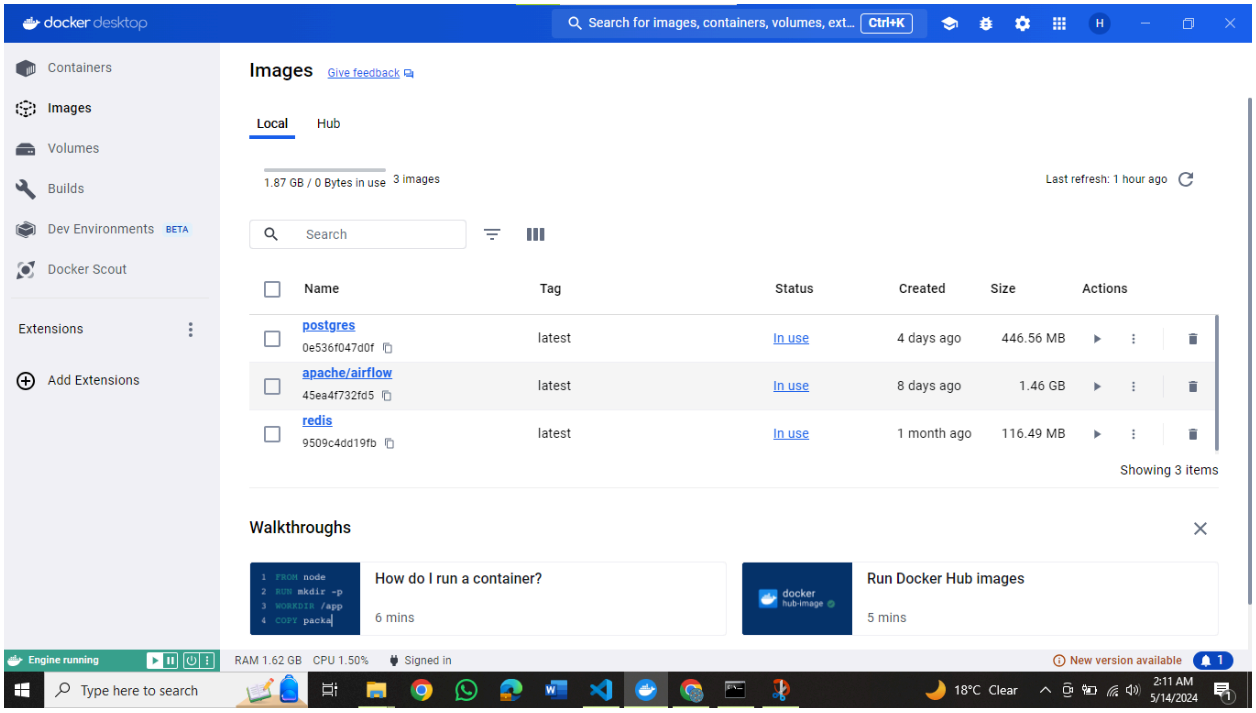
Acknowledgments:

We acknowledge the support and guidance provided by the project team and stakeholders throughout the implementation of the data pipeline.

GitHub Repository Link: https://github.com/hashaam-011



```
                    airflow-webserver-1  |      roles          Manage roles
                    airflow-webserver-1  |      tasks          Manage tasks
                    airflow-webserver-1  |      users          Manage users
                    airflow-webserver-1  |      variables      Manage variables
                    airflow-webserver-1  |
                    airflow-webserver-1  |    Commands:
                    airflow-webserver-1  |      cheat-sheet    Display cheat sheet
                    airflow-webserver-1  |      dag-processor  Start a standalone Dag Processor insta
nce
                    airflow-webserver-1  |      info           Show information about current Airflow
 and environment
                    airflow-webserver-1  |      kerberos       Start a kerberos ticket renewer
                    airflow-webserver-1  |      plugins        Dump information about loaded plugins
                    airflow-webserver-1  |      rotate-fernet-key
                    airflow-webserver-1  |                     Rotate encrypted connection credential
s and variables
                    airflow-webserver-1  |      scheduler      Start a scheduler instance
                    airflow-webserver-1  |      standalone     Run an all-in-one copy of Airflow
                    airflow-webserver-1  |      sync-perm      Update permissions for existing roles
and optionally DAGs
                    airflow-webserver-1  |      triggerer      Start a triggerer instance
                    airflow-webserver-1  |      version        Show the version
                    airflow-webserver-1  |      webserver      Start a Airflow webserver instance
                    airflow-webserver-1  |
                    airflow-webserver-1  | Options:
                    airflow-webserver-1  |   -h, --help        show this help message and exit
                    airflow-webserver-1  |
                    airflow-webserver-1  | airflow command error: the following arguments are required
: GROUP_OR_COMMAND, see help above.
airflow-webserver-1 exited with code 2
```



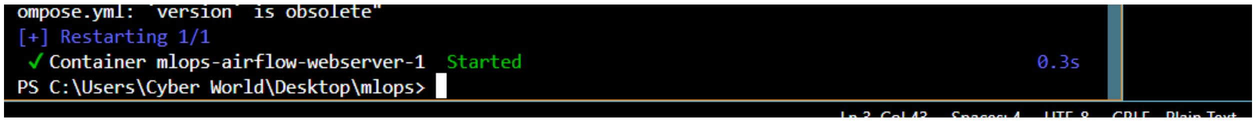
Images

1.87 GB / 0 Bytes in use 3 images

Last refresh: 1 hour ago

Name	Tag	Status	Created	Size	Actions
postgres 0e536f047d0f	latest	In use	4 days ago	446.56 MB	
apache/airflow 45ea4f732fd5	latest	In use	8 days ago	1.46 GB	
redis 9509c4dd19fb	latest	In use	1 month ago	116.49 MB	

Showing 3 items



```
ompose.yml: `version` is obsolete"
[+] Restarting 1/1
 ✓ Container mlops-airflow-webserver-1  Started                                    0.3s
PS C:\Users\Cyber World\Desktop\mlops>
```