Management of Data Science and Business Workflows
H420

Assignment – 3
Date – 17th November, 2023

Submitted By
Sony Shrestha
Aayush Paudel

Question 1

Definition
A Directed Acyclic Graph (DAG) is a graph structure that consists of nodes (vertices) connected by directed edges (arcs or arrows). The "directed" aspect implies that each edge has a specific direction, indicating a relationship between two nodes where one node is the source, and the other is the destination. "Acyclic" means that there are no cycles or loops in the graph, ensuring a clear and deterministic order of traversal.

DAGs in Apache Airflow
In the realm of Apache Airflow, a DAG is used to define and orchestrate a workflow - a sequence of tasks that need to be executed in a particular order. Each task within a DAG is represented by an operator, which is a class that defines what the task does.

DAG has been prepared to analyze a web server log file, which extracts IP addresses from the given log file, performs transformation to filter given IP addresses and generates a tar file out of it. Finally, the user is notified about the successful execution of DAG via an email.

The details of the DAG created is provided below:

DAG Details
DAG ID: process_web_log
Schedule Interval: daily
Start Date: datetime (2023,11,12)

Tasks
Above DAG consists of following tasks.

Task 1: scan_for_log

Task ID: scan_for_log

Task Description:
1. This task checks whether a log file named **log.txt** is present in the input directory.
2. **BashOperator** has been used to check file availability.
3. Process continues if the file is found, otherwise it fails with an error indicating “Log File Not Found”.

Dependencies:
1. None

Parameters:
1. Input Directory: This is the directory where our process will check for the log file.

Task 2: extract_data

Task ID: extract_data

Task Description:

1. This task extracts IP addresses from the input log file.
2. **PythonOperator**, which makes use of regular expressions to identify IP addresses, has been used.
3. **Extracted IP** addresses are then dumped into a file named **extracted_data.txt**.

Dependencies:

1. This task executes upon successful execution of **scan_for_log** (if **log.txt** file is present in input directory).

Parameters:

1. Input Directory: This is the directory where the log file is present.
2. Output Directory: This is the directory where **extracted_data.txt** file is generated by the task.

Task 3: transform_data

Task ID: transform_data

Task Description:

1. This task filters out all the occurrences of IP address **198.46.149.143** from the "extracted_data.txt" file and saves the output to a file named "**transformed_data.txt**".
2. **PythonOperator** is used for this purpose.

Dependencies:

1. This task executes upon successful execution of **extract_data** (if **extracted_data.txt** file has been generated by the previous task).

Parameters:

1. Input Directory: This is the directory where the **extracted_data.txt** file is present.
2. Output Directory: This is the directory where **transformed_data.txt** file is generated by the task.
3. IP address: This is the IP address which needs to be filtered out.

Task 4: load_data

Task ID: load_data

Task Description:

1. This task archives the file "**transformed_data.txt**" into a tar file named "**weblog.tar**".
2. **BashOperator** has been used for this purpose.

Dependencies:

1. This task executes upon successful execution of transform_data (if **transformed_data.txt** file has been generated by the previous task).

Parameters:

1. Input Directory: This is the directory where the **transformed_data.txt** file is present.
2. Output Directory: This is the directory where the **weblog.tar** file is generated by the task.

Task 5: send_email

Task ID: send_email

Task Description:

1. This task sends an email to the user, notifying him/her about the successful execution of the workflow.
2. **EmailOperator** has been used for this purpose.

Dependencies:

1. This task executes upon successful execution of **load_data** (if weblog.tar file has been generated by the previous task).

Parameters:

1. Input Directory: This is the directory where the **weblog.tar** file is present.
2. Recipient Email: Email address of the user to be notified about the successful execution of workflow is provided.
3. Subject: Subject of the email
4. Content: Body of the email

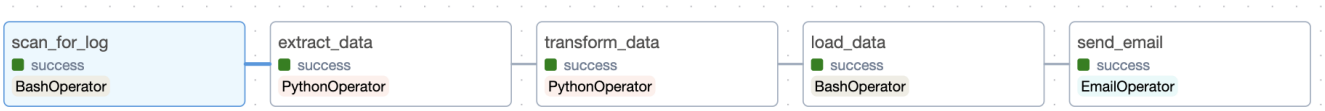
Overall Dependencies:

The tasks present in workflow are executed in the following sequence.

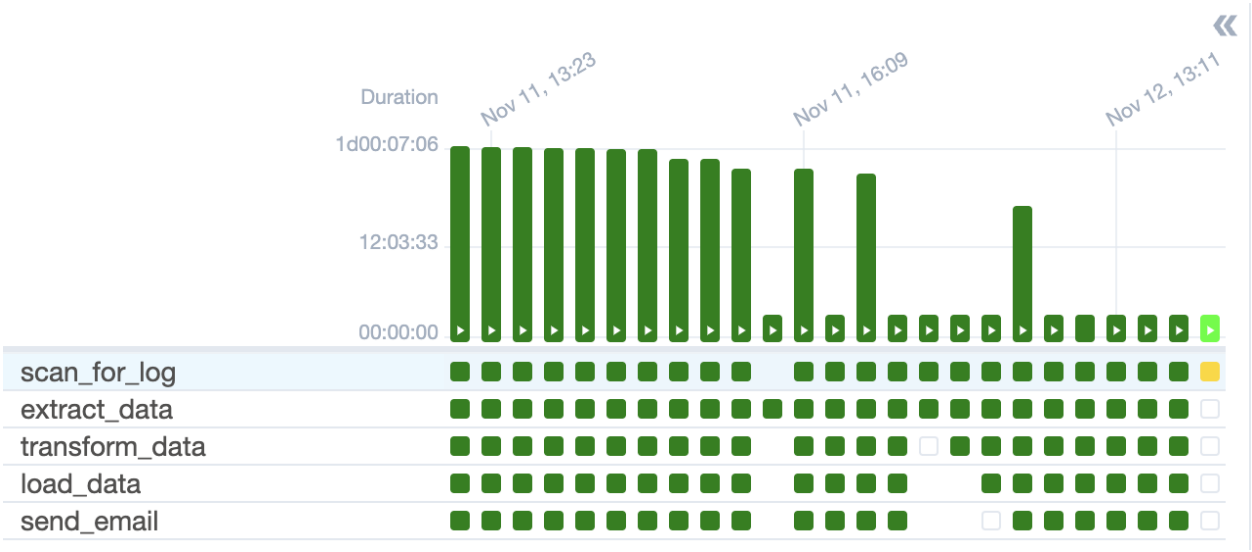
scan_for_log_task >> extract_data_task >> transform_data_task >> load_data_task >> email_task

Question 2

The following image provides the sequential flow of the task.



Number of executions were done. Corresponding execution chart has been provided below.



On average, it took 9 seconds for entire workflow to be executed successfully. Details for each task instance has been provided in diagram below.

Details for scan_for_log task which was executed successfully

Task Instance Details

Status	success
Task ID	scan_for_log
Run ID	manual__2023-11-12T13:59:29.943405+00:00
Operator	BashOperator
Trigger Rule	all_success
Duration	00:00:00
Started	2023-11-12, 13:59:32 UTC

Details for extract_data which was executed successfully

Task Instance Details

Status	success
Task ID	extract_data
Run ID	manual__2023-11-11T17:15:31.691286+00:00
Operator	PythonOperator
Trigger Rule	all_success
Duration	00:00:00
Started	2023-11-11, 17:15:35 UTC

Details for transform_data task which was executed successfully

Task Instance Details

Status	success
Task ID	transform_data
Run ID	manual__2023-11-11T17:15:31.691286+00:00
Operator	PythonOperator
Trigger Rule	all_success
Duration	00:00:00
Started	2023-11-11, 17:15:36 UTC

Details for load_data task which was executed successfully

Task Instance Details

Status	success
Task ID	load_data
Run ID	manual__2023-11-11T17:15:31.691286+00:00
Operator	BashOperator
Trigger Rule	all_success
Duration	00:00:00
Started	2023-11-11, 17:15:38 UTC

Details for send_mail task which was executed successfully

Task Instance Details

Status	success
Task ID	send_email
Run ID	manual__2023-11-12T13:59:29.943405+00:00
Operator	EmailOperator
Trigger Rule	all_success
Duration	00:00:01
Started	2023-11-12, 13:59:38 UTC

Details for scan_for_log task which failed because of log file not found scenario

Task Instance Details

Status	failed
Task ID	scan_for_log
Run ID	manual__2023-11-12T14:03:39.909381+00:00
Operator	BashOperator
Trigger Rule	all_success
Duration	00:00:00
Started	2023-11-12, 14:08:42 UTC

Details for scan_for_log task which was scanning for the input log file

Task Instance Details

Status	up_for_retry
Task ID	scan_for_log
Run ID	manual__2023-11-12T14:03:39.909381+00:00
Operator	BashOperator
Trigger Rule	all_success
Duration	00:00:00
Started	2023-11-12, 14:03:41 UTC

Question 3
Additional task has been created in airflow that is responsible for sending an email to the user, notifying him/her about the successful execution of the workflow.

Task Detail for **send_email** has been provided below.

Task 5: send_email

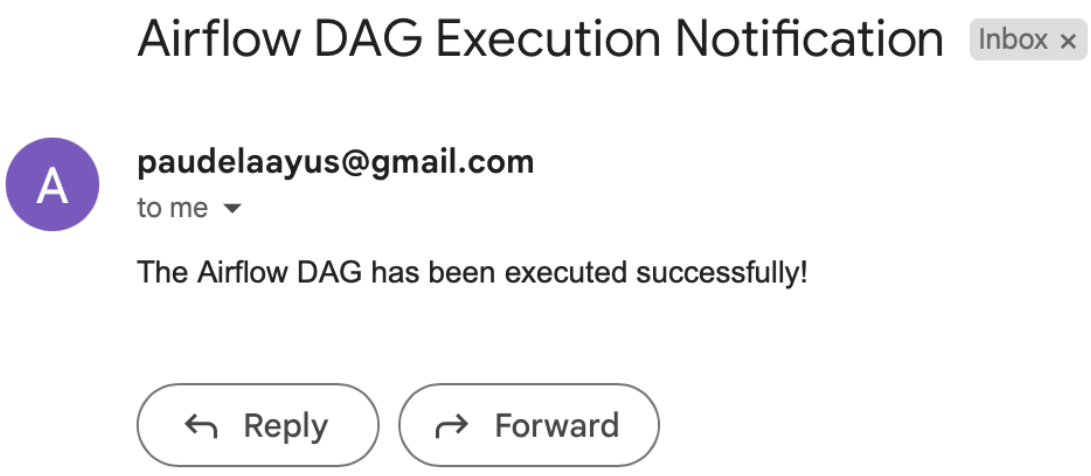
Task ID: send_email

Task Description:
1. This task sends an email to the user, notifying him/her about the successful execution of the workflow.
2. **EmailOperator** has been used for this purpose.

Dependencies:
1. This task executes upon successful execution of load_data (if **weblog.tar** file has been generated by the previous task).

Parameters:
1. Input Directory: This is the directory where the **weblog.tar** file is present.
2. Recipient Email: Email address of the user to be notified about the successful execution of workflow is provided.
3. Subject: Subject of the email
4. Content: Body of the email

Email is received as follows.



In order to make sure that Email operator works as expected, aiflow.cfg file is configured for SMTP as follows:

```
[smtp]
smtp_host = smtp.gmail.com
smtp_starttls = True
smtp_ssl = False
smtp_user = your_gmail_username@gmail.com
smtp_password = your_gmail_password
smtp_port = 587
smtp_mail_from = your_airflow_email@gmail.com
```