

Assignment 2: Data Pipeline with Apache Airflow

Course: Data Acquisition and Management

Instructor: Muhammad Shahin

PGDM PREDICTIVE ANALYTICS

GURDARSHAN SINGH

Due Date: February 16, 2025

Executive Summary

This report presents the implementation of a data pipeline using Apache Airflow to perform ETL operations on a remote dataset. The pipeline automates data extraction, transformation, and storage while ensuring modularity and efficiency. The DAG successfully processes traffic data by downloading, extracting, filtering, transforming, and combining files before generating the final output.

Introduction

The objective of this assignment is to design and implement an ETL pipeline using Apache Airflow. The pipeline follows a systematic approach to process traffic data hosted remotely, ensuring seamless automation and reproducibility.

Implementation

1. Setting Up the Airflow Environment

- Installed and configured Apache Airflow.
- Initialized the Airflow standalone server using airflow standalone.
- Created a DAG in the dags folder with appropriate configurations.

2. DAG Definition

- DAG Name: traffic_data_ETL
- Owner: Gurdarshan_Singh
- Start Date: February 10, 2025
- Schedule: @daily

3. Tasks Implemented

1. **Create directories** – Ensured required directories exist for storing processed files.
2. **Download dataset** – Fetched the dataset from the remote URL.
3. **Extract files** – Decompressed the downloaded archive.
4. **Extract CSV data** – Selected specific columns from vehicle-data.csv.

5. **Extract TSV data** – Extracted relevant fields from tollplaza-data.tsv.
6. **Extract Fixed Width data** – Parsed structured fields from payment-data.txt.
7. **Combine extracted data** – Merged the extracted datasets into a single file.
8. **Transform data** – Converted vehicle type column to uppercase.
9. **Define dependencies** – Ensured optimal task execution order.

Feb 13 15:29 guru@Nitro-ANV15-51:~/airflow/dags

```

GNU nano 7.2
from airflow import DAG
from airflow.operators.bash import BashOperator
from airflow.operators.python import PythonOperator
from datetime import datetime, timedelta
import os
import tarfile
import requests
import pandas as pd

# Define constants
URL = 'https://elasticbeanstalk-us-east-2-340729127361.s3.us-east-2.amazonaws.com/trafficdata.tgz'
DATA_DIR = '/tmp/traffic_data'
EXTRACT_DIR = f'{DATA_DIR}/extracted'
ZIP_FILE = f'{DATA_DIR}/trafficdata.tgz'

# Default args
default_args = {
    'owner': 'Gurdarshan_Singh',
    'depends_on_past': False,
    'start_date': datetime(2025, 2, 10),
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
}

# Define DAG
with DAG(
    dag_id='traffic_data_ETL',
    default_args=default_args,
    schedule_interval='@daily',
    catchup=False
) as dag:
    # Task 1: Create directories
    create_directories = BashOperator(
        task_id='create_directories',
        bash_command=f'mkdir -p {EXTRACT_DIR}'
    )

    # Task 2: Download the file
    def download_data():
        response = requests.get(URL, stream=True)

```

^C Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-G Copy M-Q Where Was M-W Next ^F Forward

Feb 13 15:30

guru@Nitro-ANV15-51:~/airflow/dags

GNU nano 7.2 traffic_data_ETL.py *

```
    )
# Task 2: Download the file
def download_data():
    response = requests.get(URL, stream=True)
    with open(ZIP_FILE, 'wb') as file:
        for chunk in response.iter_content(chunk_size=1024):
            file.write(chunk)

download_task = PythonOperator(
    task_id='download_data',
    python_callable=download_data
)

# Task 3: Unzip the file
def extract_data():
    with tarfile.open(ZIP_FILE, 'r:gz') as tar:
        tar.extractall(EXTRACT_DIR)

extract_task = PythonOperator(
    task_id='extract_data',
    python_callable=extract_data
)

# Task 4: Extract data from CSV
def process_csv():
    df = pd.read_csv(f'{EXTRACT_DIR}/vehicle-data.csv')
    df.iloc[:, [0, 1, 2, 3]].to_csv(f'{EXTRACT_DIR}/csv_d.csv', index=False)

extract_csv_task = PythonOperator(
    task_id='extract_csv',
    python_callable=process_csv
)

# Task 5: Extract data from TSV
def process_tsv():
    df = pd.read_csv(f'{EXTRACT_DIR}/tollplaza-data.tsv', sep='\t')
    df.iloc[:, [4, 5, 6]].to_csv(f'{EXTRACT_DIR}/tsv_d.csv', index=False)

extract_tsv_task = PythonOperator(
    task_id='extract_tsv',
    python_callable=process_tsv
)
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next ^F Forward

Feb 13 15:30

guru@Nitro-ANV15-51:~/airflow/dags

```
GNU nano 7.2 traffic_data_ETL.py *
```

```
# Task 5: Extract data from TSV
def process_tsv():
    df = pd.read_csv(f'{EXTRACT_DIR}/tollplaza-data.tsv', sep='\t')
    df.iloc[:, [4, 5, 6]].to_csv(f'{EXTRACT_DIR}/tsv_d.csv', index=False)

extract_tsv_task = PythonOperator(
    task_id='extract_tsv',
    python_callable=process_tsv
)

# Task 6: Extract data from fixed-width file
def process_fixed_width():
    colspecs = [(0, 2), (3, 5)] # Adjust based on file format
    df = pd.read_fwf(f'{EXTRACT_DIR}/payment-data.txt', colspecs=colspecs)
    df.to_csv(f'{EXTRACT_DIR}/fixed_width_d.csv', index=False)

extract_fixed_width_task = PythonOperator(
    task_id='extract_fixed_width',
    python_callable=process_fixed_width
)

# Task 7: Combine extracted files
combine_data = BashOperator(
    task_id='combine_data',
    bash_command=f'paste -d " " {EXTRACT_DIR}/csv_d.csv {EXTRACT_DIR}/tsv_d.csv {EXTRACT_DIR}/fixed_width_d.csv > {EXTRACT_DIR}/combined_data.csv'
)

# Task 8: Transform data
def transform_data():
    df = pd.read_csv(f'{EXTRACT_DIR}/combined_data.csv')
    df.iloc[:, 3] = df.iloc[:, 3].str.upper()
    df.to_csv(f'{EXTRACT_DIR}/transformed_data.csv', index=False)

transform_task = PythonOperator(
    task_id='transform_data',
    python_callable=transform_data
)

# Set task dependencies
create_directories >> download_task >> extract_task
extract_task >> [extract_csv_task, extract_tsv_task, extract_fixed_width_task] >> combine_data >> transform_task
```

^C Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-B Redo M-6 Copy ^Q Where Was M-W Next ^F Forward

4. Airflow DAG Execution

- DAG successfully appeared in the Airflow UI at <http://localhost:8080>
- DAG was activated and manually triggered
- Execution logs confirmed successful task completion

Screenshots

(Include screenshots of *DAG Graph View*, *Task Logs*, *Gantt Chart*, and *Data Output*)

The screenshot shows the Airflow web interface for managing Data Acquisition and Processing (DAG) tasks. The main page displays a list of DAGs, each with its name, owner, run history, and scheduling information.

DAGs Overview:

- All (2)**: Total number of DAGs.
- Active (2)**: DAGs currently running.
- Paused (0)**: DAGs currently paused.
- Running (0)**: DAGs currently running.
- Failed (0)**: DAGs that have failed.

Filter DAGs by tag and **Search DAGs** input fields are available for filtering the list.

Auto-refresh button and **Links** icon are present on the right.

DAG Details:

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
hello-world	gurdarshan singh	624	@daily	2025-02-16, 00:00:00	2025-02-17, 00:00:00	2	[Run, Log, Task, ...]	[Link]
traffic_data_ETL	Gurdarshan_Singh	11	@daily	2025-02-17, 00:34:03	2025-02-17, 00:00:00	8	[Run, Log, Task, ...]	[Link]

Pagination controls (1 of 2) and a message "Showing 1-2 of 2 DAGs" are at the bottom.

Footer:

Version: v2.10.5
Git Version: .release:b93c3db6b1641b0840bd15ac7d05bc58ff2cccbf

localhost:8080/dags/traffic_data_ETL?execution_date=2025-02-15T14%3A49%3A37.138441%2B00%3A00&tab=details&dag_run_id=manual_2025-02-15T14%3A49%3A37.138441%2B00%3A00&execution_date=2025-02-15T14%3A49%3A37.138441%2B00%3A00&tab=details&dag_run_id=manual_2025-02-15T14%3A49%3A37.138441%2B00%3A00

14:50 UTC AU

DAG: traffic_data_ETL

Schedule: @daily Next Run ID: 2025-02-15, 00:00:00 UTC

02/15/2025 02:49:37 PM All Run Types All Run States Clear Filters

Duration: 00:00:47 Feb 15, 00:00

create_directories, download_data, extract_data, extract_csv, extract_tsv, extract_fixed_width, combine_data, transform_data

DAG Run Notes

Dag Run Details

Status	success
Run ID	manual_2025-02-15T14:49:37.138441+00:00
Run type	manual
Run duration	00:00:12
Last scheduling decision	2025-02-15, 14:49:50 UTC
Queued at	2025-02-15, 14:49:37 UTC
Started	2025-02-15, 14:49:38 UTC
Ended	2025-02-15, 14:49:50 UTC
Data interval start	2025-02-14, 00:00:00 UTC

Ended: 2025-02-15, 14:49:50 UTC

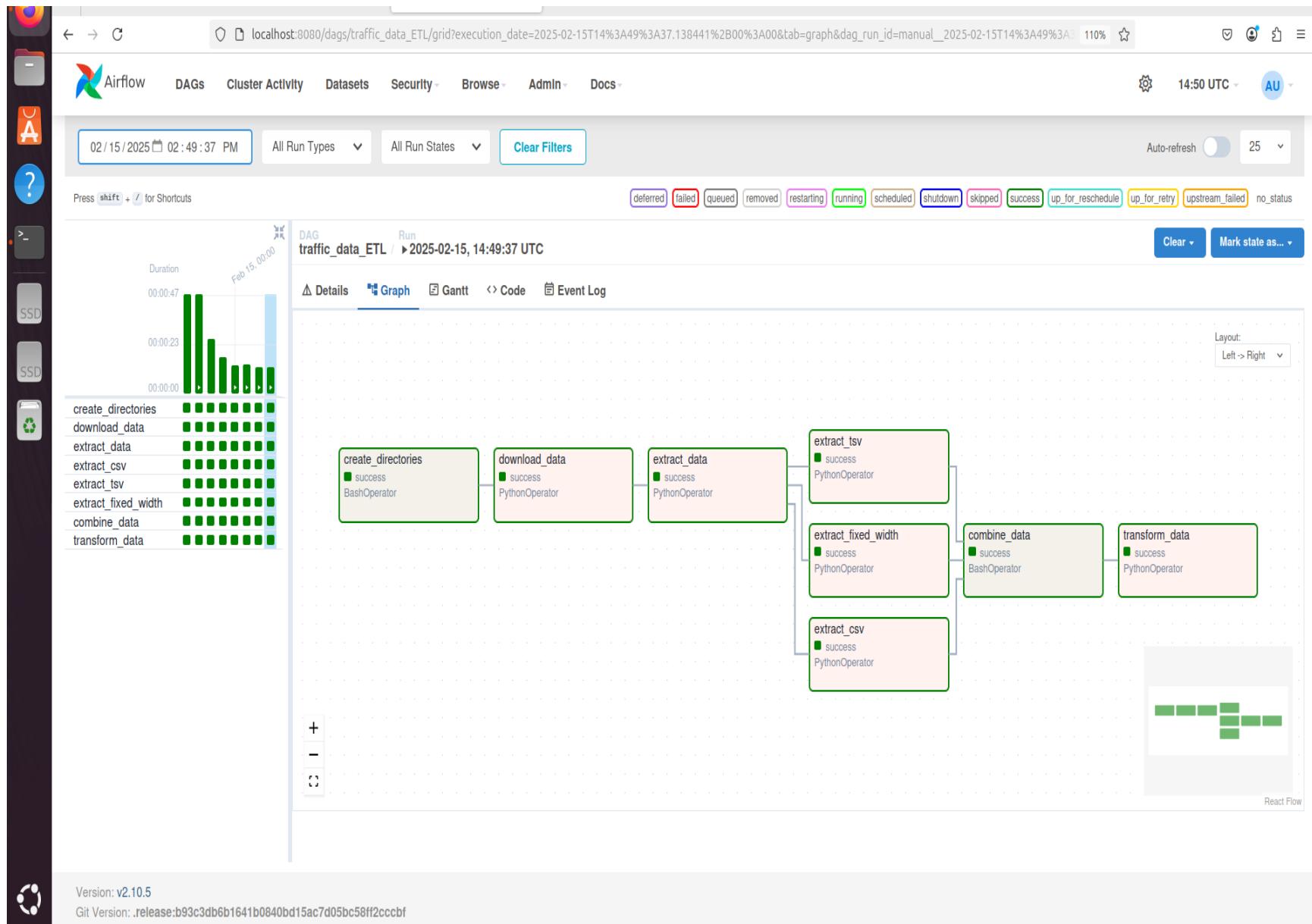
Data interval start: 2025-02-14, 00:00:00 UTC

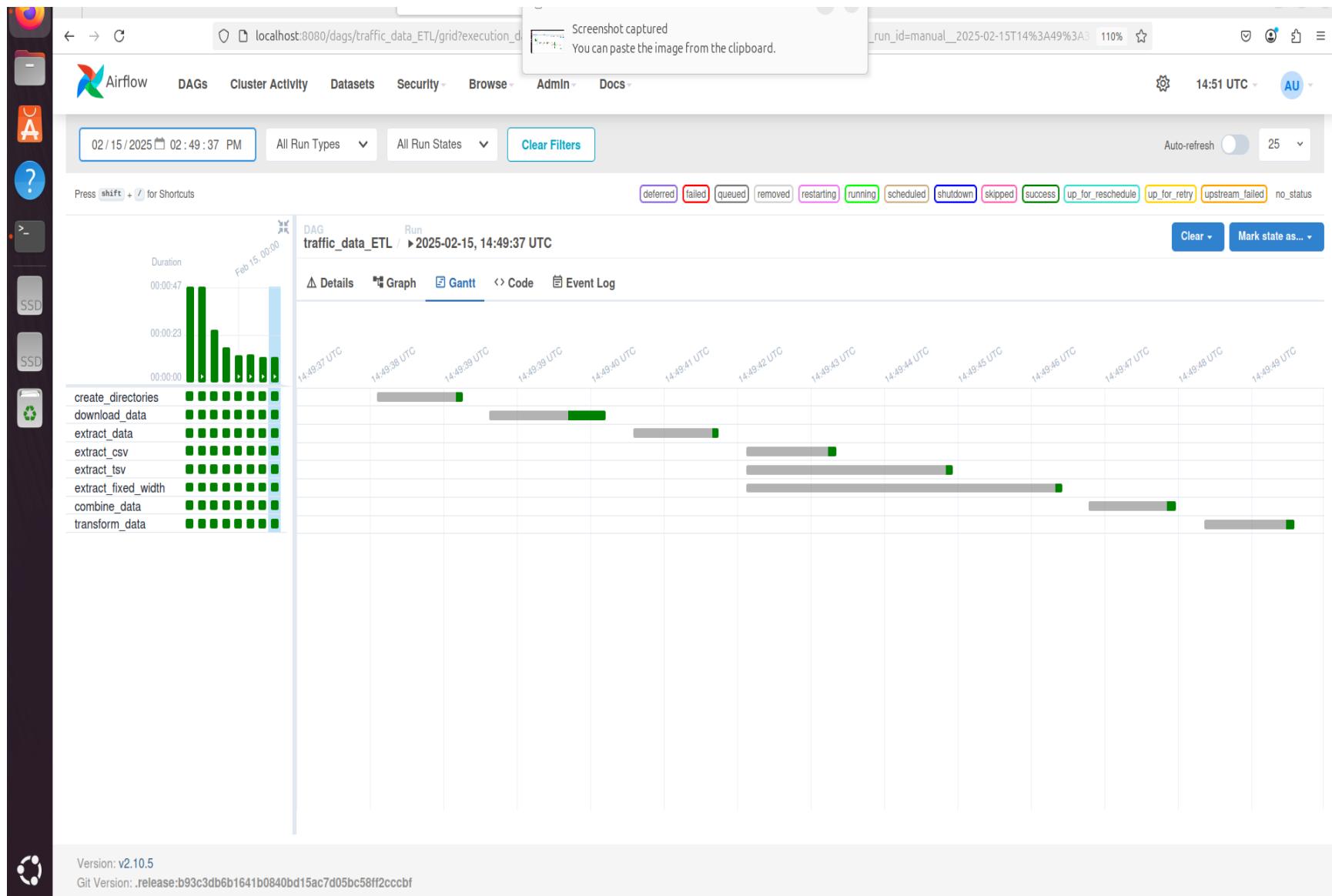
Data interval end: 2025-02-15, 00:00:00 UTC

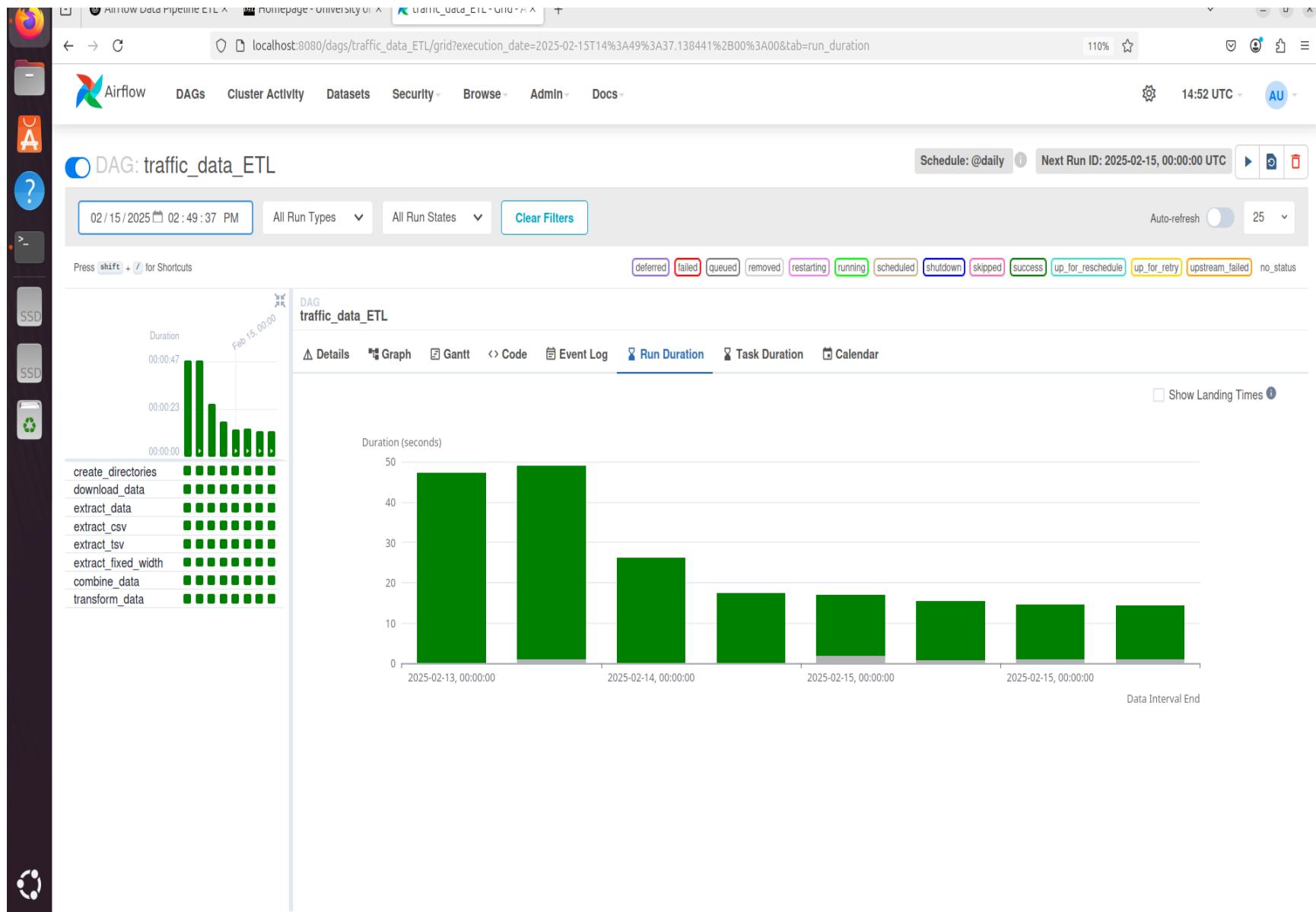
Externally triggered: True

Run config: None

Version: v2.10.5
Git Version: .release:b93c3db6b1641b0840bd15ac7d05bc58ff2cccbf







localhost:8080/dags/traffic_data_ETL/grid?execution_date=2025-02-15T14%3A49%3A37.138441%2B00%3A00&tab=event_log&dag_run_id=manual__2025-02-15T14%3A49%3A37.138441%2B00%3A00

14:51 UTC AU

DAG: traffic_data_ETL

Schedule: @daily | Next Run ID: 2025-02-15, 00:00:00 UTC | Auto-refresh 25

02 / 15 / 2025 02 : 49 : 37 PM | All Run Types | All Run States | Clear Filters

Press shift + ⌘ for Shortcuts

Duration: Feb 15, 00:00

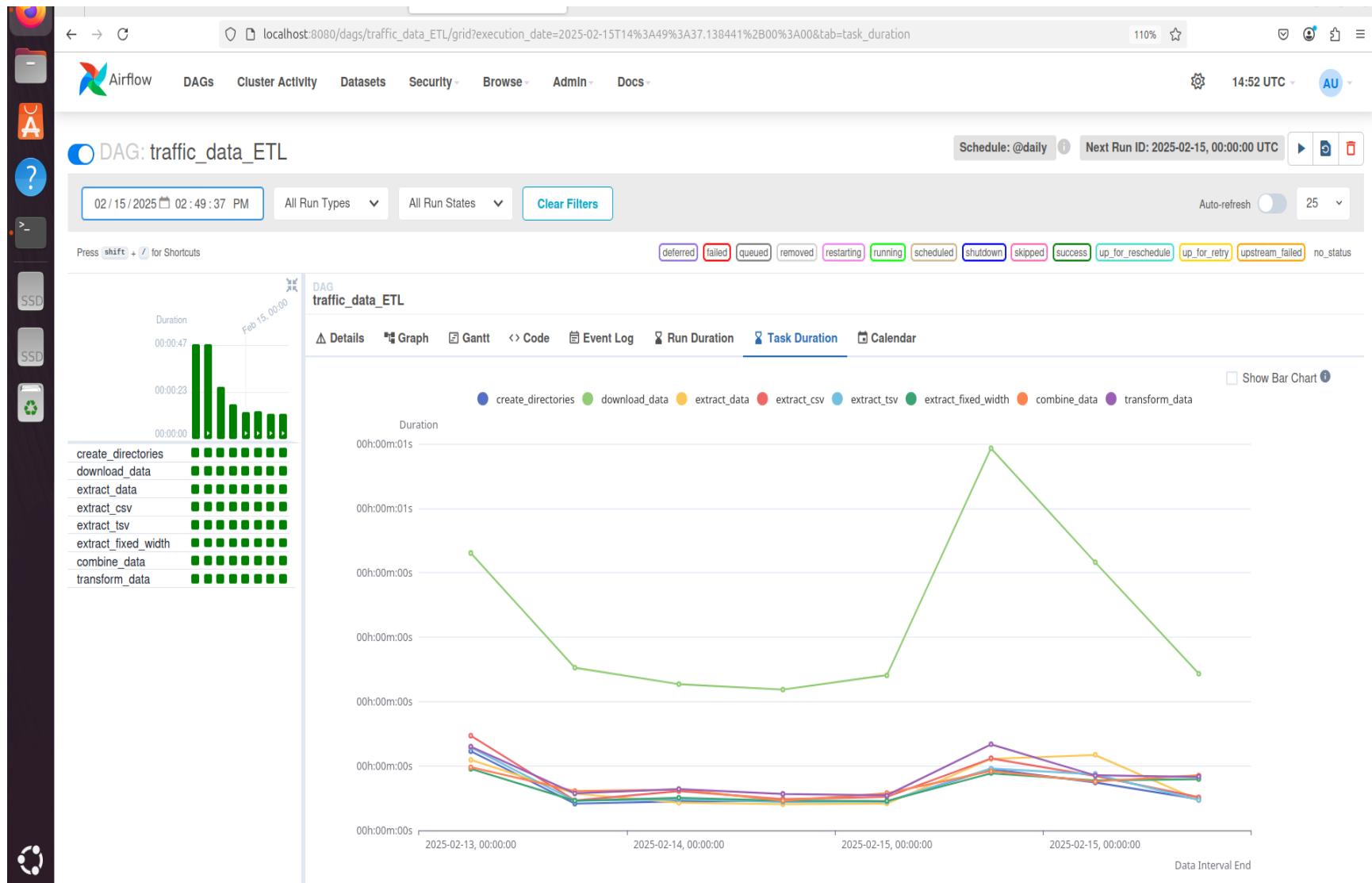
create_directories
download_data
extract_data
extract_csv
extract_tsv
extract_fixed_width
combine_data
transform_data

DAG traffic_data_ETL / Run 2025-02-15, 14:49:37 UTC

Event Log

Show Logs After: 02 / 15 / 2025 02 : 51 : 09 PM | Show Logs Before: 02 / 15 / 2025 02 : 51 : 09 PM | Events to: Include | Exclude | Select... | View full cluster Audit Log

WHEN	TASK ID	MAP INDEX	TRY NUMBER	EVENT	USER	DETAILS
2025-02-15, 14:49:50 UTC	transform_data		1	success	Gurdarshan_Singh	
2025-02-15, 14:49:49 UTC	transform_data		1	running	Gurdarshan_Singh	
2025-02-15, 14:49:48 UTC	combine_data		1	success	Gurdarshan_Singh	
2025-02-15, 14:49:48 UTC	combine_data		1	running	Gurdarshan_Singh	
2025-02-15, 14:49:47 UTC	extract_fixed_width		1	success	Gurdarshan_Singh	
2025-02-15, 14:49:46 UTC	extract_fixed_width		1	running	Gurdarshan_Singh	
2025-02-15, 14:49:45 UTC	extract_tsv		1	success	Gurdarshan_Singh	
2025-02-15, 14:49:45 UTC	extract_tsv		1	running	Gurdarshan_Singh	
2025-02-15, 14:49:44 UTC	extract_csv		1	success	Gurdarshan_Singh	
2025-02-15, 14:49:44 UTC	extract_csv		1	running	Gurdarshan_Singh	
2025-02-15, 14:49:42 UTC	extract_data		1	success	Gurdarshan_Singh	
2025-02-15, 14:49:42 UTC	extract_data		1	running	Gurdarshan_Singh	



Schedule: @daily | Next Run ID: 2025-02-15, 00:00:00 UTC | Auto-refresh | 25 |

DAG: traffic_data_ETL

02/15/2025 02:49:37 PM | All Run Types | All Run States | Clear Filters

Press shift + / for Shortcuts

Duration

Feb 15, 00:00

create_directories

download_data

extract_data

extract_csv

extract_tsv

extract_fixed_width

combine_data

transform_data

Deferred Failed Queued Removed Restarting Running Scheduled Shutdown Skipped Success Up_for_reschedule Up_for_retry Upstream_failed No status

DAG traffic_data_ETL

Details Graph Gantt Code Event Log Run Duration Task Duration Calendar

Show Logs After Show Logs Before Events to Include Exclude

02/15/2025 02:51:37 PM 02/15/2025 02:51:37 PM Select...

WHEN*	RUN ID*	TASK ID*	MAP INDEX	TRY NUMBER	EVENT*	USER*	DETAILS*
2025-02-15, 14:49:50 UTC	manual_2025-02-15T14:49:37.138441+00:00	transform_data		1	success	Gurdarshan_Singh	<pre>{"host_name": "Nitro-ANV15-51", "full_command": ["'/home/guru/.local/bin/airflow', 'tasks', 'run', 'traffic_data_ETL', 'transform_data', 'manual_2025-02-15T14:49:37.138441+00:00', '--local', '--subdir', 'DAGS_FOLDER/traffic_data_ETL.py']"}</pre>
2025-02-15, 14:49:49 UTC		transform_data			cli_task_run	guru	<pre>{"host_name": "Nitro-ANV15-51", "full_command": ["'/home/guru/.local/bin/airflow', 'tasks', 'run', 'traffic_data_ETL', 'transform_data', 'manual_2025-02-15T14:49:37.138441+00:00', '--local', '--subdir', 'DAGS_FOLDER/traffic_data_ETL.py']"}</pre>
2025-02-15, 14:49:49 UTC	manual_2025-02-15T14:49:37.138441+00:00	transform_data		1	running	Gurdarshan_Singh	<pre>{"host_name": "Nitro-ANV15-51", "full_command": ["'/home/guru/.local/bin/airflow', 'tasks', 'run', 'traffic_data_ETL', 'transform_data', 'manual_2025-02-15T14:49:37.138441+00:00', '--local', '--subdir', 'DAGS_FOLDER/traffic_data_ETL.py']"}</pre>
2025-02-15,		transform_data			cli_task_run	guru	<pre>{"host_name": "Nitro-ANV15-51", "full_command": ["'/home/guru/.local/bin/airflow', 'tasks', 'run', 'traffic_data_ETL', 'transform_data', 'manual_2025-02-15T14:49:37.138441+00:00', '--local', '--subdir', 'DAGS_FOLDER/traffic_data_ETL.py']"}</pre>
2025-02-15,							

View full cluster Audit Log

localhost:8080/dags/traffic_data_ETL/grid?execution_date=2025-02-15T14%3A49%3A37.138441%2B00%3A00&tab=event_log

110% ⭐ 14:52 UTC AU

DAG: traffic_data_ETL

02 / 15 / 2025 02 : 49 : 37 PM All Run Types All Run States Clear Filters Auto-refresh 25

Press shift + / for Shortcuts

Duration Feb 15, 00:00

00:00:47

00:00:23

00:00:00

create_directories

download_data

extract_data

extract_csv

extract_tsv

extract_fixed_width

combine_data

transform_data

DAG traffic_data_ETL

Details Graph Gantt Code Event Log Run Duration Task Duration Calendar

Run ID	Trigger	Task	State	Duration	User	Logs
14:49:48 UTC	manual_2025-02-15T14:49:37.138441+00:00	combine_data	success	1	Gurdarshan_Singh	{"host_name": "Nitro-ANV15-51", "full_command": ["'/home/guru/.local/bin/airflow', 'tasks', 'run', 'traffic_data_ETL', 'combine_data', 'manual_2025-02-15T14:49:37.138441+00:00', '--local', '--subdir', 'DAGS_FOLDER/traffic_data_ETL.py']}"
2025-02-15, 14:49:48 UTC		combine_data	cli_task_run	guru		
2025-02-15, 14:49:48 UTC	manual_2025-02-15T14:49:37.138441+00:00	combine_data	running	1	Gurdarshan_Singh	{"host_name": "Nitro-ANV15-51", "full_command": ["'/home/guru/.local/bin/airflow', 'tasks', 'run', 'traffic_data_ETL', 'combine_data', 'manual_2025-02-15T14:49:37.138441+00:00', '--local', '--subdir', 'DAGS_FOLDER/traffic_data_ETL.py']}"
2025-02-15, 14:49:48 UTC		combine_data	cli_task_run	guru		
2025-02-15, 14:49:47 UTC	manual_2025-02-15T14:49:37.138441+00:00	extract_fixed_width	success	1	Gurdarshan_Singh	{"host_name": "Nitro-ANV15-51", "full_command": ["'/home/guru/.local/bin/airflow', 'tasks', 'run', 'traffic_data_ETL', 'extract_fixed_width', 'manual_2025-02-15T14:49:37.138441+00:00', '--local', '--subdir', 'DAGS_FOLDER/traffic_data_ETL.py']}"
2025-02-15, 14:49:46 UTC		extract_fixed_width	cli_task_run	guru		

localhost:8080/taskinstance/list/?_flt_3_dag_id=traffic_data_ETL&_flt_3_state=success

110% ⭐ 14:54 UTC AU

List Task Instance

Search ▾

Actions ▾

Record Count: 64

	State	Dag Id	Task Id	Run Id	Map Index	Logical Date	Operator	Start Date	End Date	Duration	Note	Job Id	Ho
<input type="checkbox"/>	success	traffic_data_ETL	create_directories	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	BashOperator	2025-02-13, 22:21:37	2025-02-13, 22:21:37	<1s		1282	Nit
<input type="checkbox"/>	success	traffic_data_ETL	create_directories	manual__2025-02-13T22:21:34.767622+00:00		2025-02-13, 22:21:34	BashOperator	2025-02-13, 22:21:40	2025-02-13, 22:21:40	<1s		1283	Nit
<input type="checkbox"/>	success	traffic_data_ETL	download_data	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	PythonOperator	2025-02-13, 22:21:43	2025-02-13, 22:21:44	1s		1284	Nit
<input type="checkbox"/>	success	traffic_data_ETL	download_data	manual__2025-02-13T22:21:34.767622+00:00		2025-02-13, 22:21:34	PythonOperator	2025-02-13, 22:21:47	2025-02-13, 22:21:47	<1s		1285	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_data	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	PythonOperator	2025-02-13, 22:21:51	2025-02-13, 22:21:51	<1s		1286	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_data	manual__2025-02-13T22:21:34.767622+00:00		2025-02-13, 22:21:34	PythonOperator	2025-02-13, 22:21:53	2025-02-13, 22:21:54	<1s		1287	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_csv	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	PythonOperator	2025-02-13, 22:21:56	2025-02-13, 22:21:57	<1s		1288	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_tsv	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	PythonOperator	2025-02-13, 22:21:59	2025-02-13, 22:21:59	<1s		1289	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_fixed_width	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	PythonOperator	2025-02-13, 22:22:02	2025-02-13, 22:22:02	<1s		1290	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_csv	manual__2025-02-13T22:21:34.767622+00:00		2025-02-13, 22:21:34	PythonOperator	2025-02-13, 22:22:04	2025-02-13, 22:22:05	<1s		1291	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_tsv	manual__2025-02-13T22:21:34.767622+00:00		2025-02-13, 22:21:34	PythonOperator	2025-02-13, 22:22:07	2025-02-13, 22:22:07	<1s		1292	Nit
<input type="checkbox"/>	success	traffic_data_ETL	extract_fixed_width	manual__2025-02-13T22:21:34.767622+00:00		2025-02-13, 22:21:34	PythonOperator	2025-02-13, 22:22:10	2025-02-13, 22:22:10	<1s		1293	Nit
<input type="checkbox"/>	success	traffic_data_ETL	combine_data	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	BashOperator	2025-02-13, 22:22:12	2025-02-13, 22:22:12	<1s		1294	Nit
<input type="checkbox"/>	success	traffic_data_ETL	combine_data	manual__2025-02-13T22:21:34.767622+00:00		2025-02-13, 22:21:34	BashOperator	2025-02-13, 22:22:15	2025-02-13, 22:22:15	<1s		1295	Nit
<input type="checkbox"/>	success	traffic_data_ETL	transform_data	scheduled__2025-02-12T00:00:00+00:00		2025-02-12, 00:00:00	PythonOperator	2025-02-13, 22:22:18	2025-02-13, 22:22:18	<1s		1296	Nit

localhost:8080/taskinstance/list/?_flt_3_dag_id=traffic_data_ETL&_flt_3_state=sucess

110% ⭐ 14:54 UTC AU

Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs

<input type="checkbox"/>			success	traffic_data_ETL	combine_data	manual_2025-02-15T14:20:44.849654+00:00	2025-02-15, 14:20:44	BashOperator	2025-02-15, 14:20:57	2025-02-15, 14:20:57	<1s	1334	Nit
<input type="checkbox"/>			success	traffic_data_ETL	transform_data	manual_2025-02-15T14:20:44.849654+00:00	2025-02-15, 14:20:44	PythonOperator	2025-02-15, 14:20:59	2025-02-15, 14:20:59	<1s	1335	Nit
<input type="checkbox"/>			success	traffic_data_ETL	create_directories	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	BashOperator	2025-02-15, 14:29:37	2025-02-15, 14:29:37	<1s	1338	Nit
<input type="checkbox"/>			success	traffic_data_ETL	download_data	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	PythonOperator	2025-02-15, 14:29:39	2025-02-15, 14:29:39	<1s	1339	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_data	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	PythonOperator	2025-02-15, 14:29:41	2025-02-15, 14:29:41	<1s	1340	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_csv	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	PythonOperator	2025-02-15, 14:29:42	2025-02-15, 14:29:42	<1s	1341	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_tsv	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	PythonOperator	2025-02-15, 14:29:44	2025-02-15, 14:29:44	<1s	1342	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_fixed_width	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	PythonOperator	2025-02-15, 14:29:45	2025-02-15, 14:29:45	<1s	1343	Nit
<input type="checkbox"/>			success	traffic_data_ETL	combine_data	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	BashOperator	2025-02-15, 14:29:47	2025-02-15, 14:29:47	<1s	1344	Nit
<input type="checkbox"/>			success	traffic_data_ETL	transform_data	manual_2025-02-15T14:29:35.486097+00:00	2025-02-15, 14:29:35	PythonOperator	2025-02-15, 14:29:48	2025-02-15, 14:29:48	<1s	1345	Nit
<input type="checkbox"/>			success	traffic_data_ETL	create_directories	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	BashOperator	2025-02-15, 14:49:39	2025-02-15, 14:49:39	<1s	1346	Nit
<input type="checkbox"/>			success	traffic_data_ETL	download_data	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	PythonOperator	2025-02-15, 14:49:40	2025-02-15, 14:49:41	<1s	1347	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_data	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	PythonOperator	2025-02-15, 14:49:42	2025-02-15, 14:49:42	<1s	1348	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_csv	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	PythonOperator	2025-02-15, 14:49:43	2025-02-15, 14:49:44	<1s	1349	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_tsv	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	PythonOperator	2025-02-15, 14:49:45	2025-02-15, 14:49:45	<1s	1350	Nit
<input type="checkbox"/>			success	traffic_data_ETL	extract_fixed_width	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	PythonOperator	2025-02-15, 14:49:46	2025-02-15, 14:49:47	<1s	1351	Nit
<input type="checkbox"/>			success	traffic_data_ETL	combine_data	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	BashOperator	2025-02-15, 14:49:48	2025-02-15, 14:49:48	<1s	1352	Nit
<input type="checkbox"/>			success	traffic_data_ETL	transform_data	manual_2025-02-15T14:49:37.138441+00:00	2025-02-15, 14:49:37	PythonOperator	2025-02-15, 14:49:49	2025-02-15, 14:49:50	<1s	1353	Nit

Version: v2.10.5

Git Version: .release:b93c3db6b1641b0840bd15ac7d05bc58ff2cccbf

Airflow Task Instances											14:54 UTC	AU
ID	State	Dag ID	Task ID	Execution Date	Last Run	Operator	Start Time	End Time	Duration	Log URL	Run ID	dag_id
1298	success	traffic_data_ETL	create_directories	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	BashOperator	2025-02-14, 05:18:53	2025-02-14, 05:18:53	<1s		1298	Nit
1300	success	traffic_data_ETL	download_data	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	PythonOperator	2025-02-14, 05:18:58	2025-02-14, 05:18:58	<1s		1300	Nit
1302	success	traffic_data_ETL	extract_data	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	PythonOperator	2025-02-14, 05:19:04	2025-02-14, 05:19:04	<1s		1302	Nit
1303	success	traffic_data_ETL	extract_csv	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	PythonOperator	2025-02-14, 05:19:07	2025-02-14, 05:19:07	<1s		1303	Nit
1304	success	traffic_data_ETL	extract_tsv	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	PythonOperator	2025-02-14, 05:19:09	2025-02-14, 05:19:09	<1s		1304	Nit
1305	success	traffic_data_ETL	extract_fixed_width	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	PythonOperator	2025-02-14, 05:19:12	2025-02-14, 05:19:12	<1s		1305	Nit
1306	success	traffic_data_ETL	combine_data	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	BashOperator	2025-02-14, 05:19:14	2025-02-14, 05:19:14	<1s		1306	Nit
1307	success	traffic_data_ETL	transform_data	scheduled_2025-02-13T00:00:00+00:00	2025-02-13, 00:00:00	PythonOperator	2025-02-14, 05:19:16	2025-02-14, 05:19:16	<1s		1307	Nit
1310	success	traffic_data_ETL	create_directories	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	BashOperator	2025-02-15, 14:12:37	2025-02-15, 14:12:37	<1s		1310	Nit
1312	success	traffic_data_ETL	download_data	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	PythonOperator	2025-02-15, 14:12:39	2025-02-15, 14:12:40	<1s		1312	Nit
1314	success	traffic_data_ETL	extract_data	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	PythonOperator	2025-02-15, 14:12:42	2025-02-15, 14:12:42	<1s		1314	Nit
1315	success	traffic_data_ETL	extract_csv	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	PythonOperator	2025-02-15, 14:12:44	2025-02-15, 14:12:44	<1s		1315	Nit
1316	success	traffic_data_ETL	extract_tsv	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	PythonOperator	2025-02-15, 14:12:46	2025-02-15, 14:12:46	<1s		1316	Nit
1317	success	traffic_data_ETL	extract_fixed_width	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	PythonOperator	2025-02-15, 14:12:47	2025-02-15, 14:12:48	<1s		1317	Nit
1318	success	traffic_data_ETL	combine_data	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	BashOperator	2025-02-15, 14:12:49	2025-02-15, 14:12:50	<1s		1318	Nit
1319	success	traffic_data_ETL	transform_data	scheduled_2025-02-14T00:00:00+00:00	2025-02-14, 00:00:00	PythonOperator	2025-02-15, 14:12:52	2025-02-15, 14:12:52	<1s		1319	Nit
1320	success	traffic_data_ETL	create_directories	manual_2025-02-15T14:17:48.110005+00:00	2025-02-15, 14:17:48	BashOperator	2025-02-15, 14:17:51	2025-02-15, 14:17:51	<1s		1320	Nit
1321	success	traffic_data_ETL	download_data	manual_2025-02-15T14:17:48.110005+00:00	2025-02-15, 14:17:48	PythonOperator	2025-02-15, 14:17:52	2025-02-15, 14:17:53	<1s		1321	Nit
1322	success	traffic_data_ETL	extract_data	manual_2025-02-15T14:17:48.110005+00:00	2025-02-15, 14:17:48	PythonOperator	2025-02-15, 14:17:54	2025-02-15, 14:17:54	<1s		1322	Nit
1323	success	traffic_data_ETL	extract_csv	manual_2025-02-15T14:17:48.110005+00:00	2025-02-15, 14:17:48	PythonOperator	2025-02-15, 14:17:56	2025-02-15, 14:17:56	<1s		1323	Nit
1324	success	traffic_data_ETL	extract_tsv	manual_2025-02-15T14:17:48.110005+00:00	2025-02-15, 14:17:48	PythonOperator	2025-02-15, 14:17:57	2025-02-15, 14:17:57	<1s		1324	Nit
1325	success	traffic_data_ETL	extract_fixed_width	manual_2025-02-15T14:17:48.110005+00:00	2025-02-15, 14:17:48	PythonOperator	2025-02-15, 14:17:50	2025-02-15, 14:17:50	<1s		1325	Nit

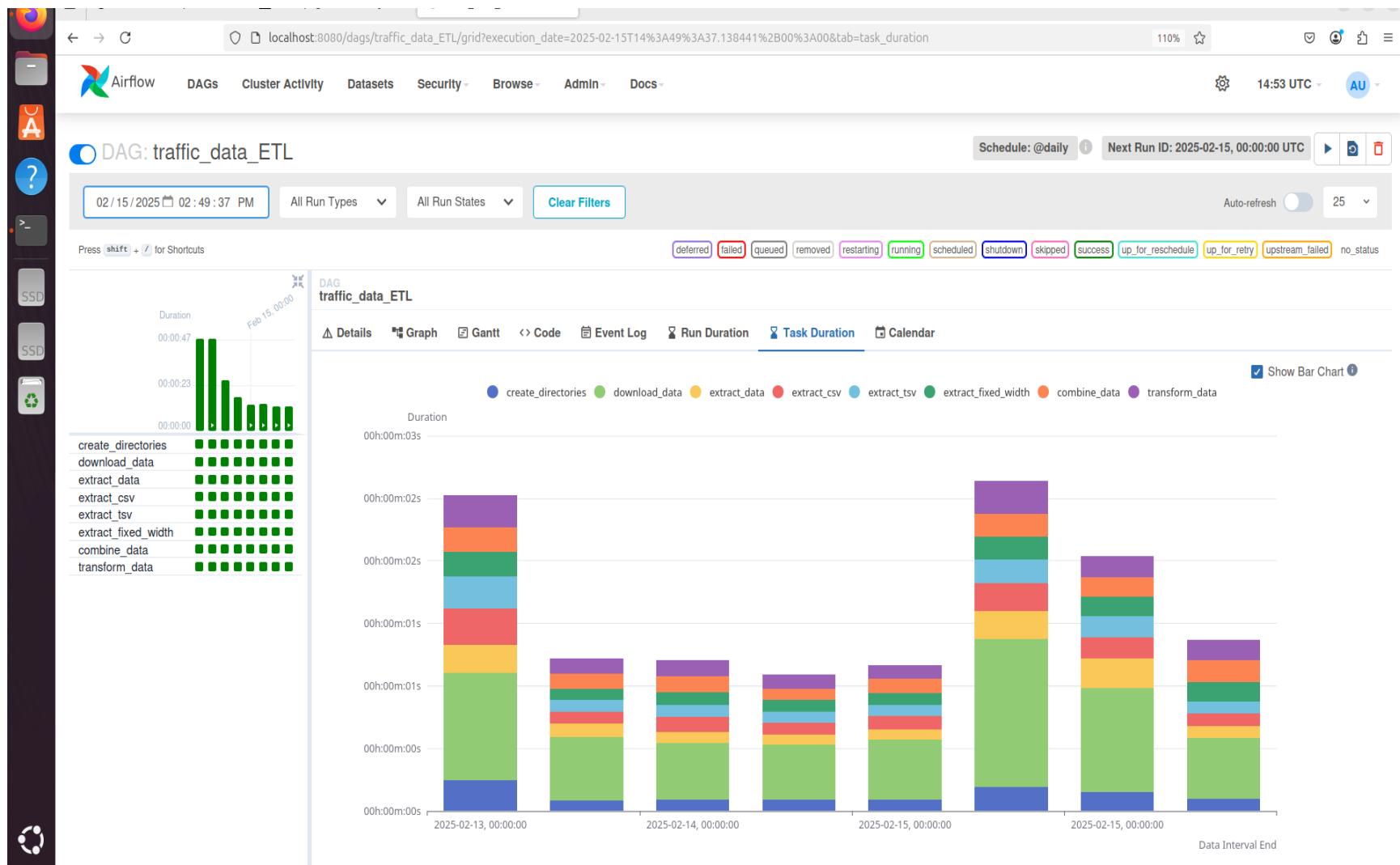
The screenshot shows the Airflow web interface running on localhost:8080. The URL in the address bar is `localhost:8080/dagrun/list/?_flt_3_dag_id=traffic_data_ETL&_flt_3_state=success`. The page title is "List Dag Run".

The interface includes a sidebar with various icons for file operations like copy, move, delete, and search. The main header has links for "Airflow", "DAGs", "Cluster Activity", "Datasets", "Security", "Browse", "Admin", and "Docs". On the right, there are status indicators for "110%", a star icon, and a user icon labeled "AU". The timestamp "14:53 UTC" is also displayed.

The main content area is titled "List Dag Run" and contains a table with the following data:

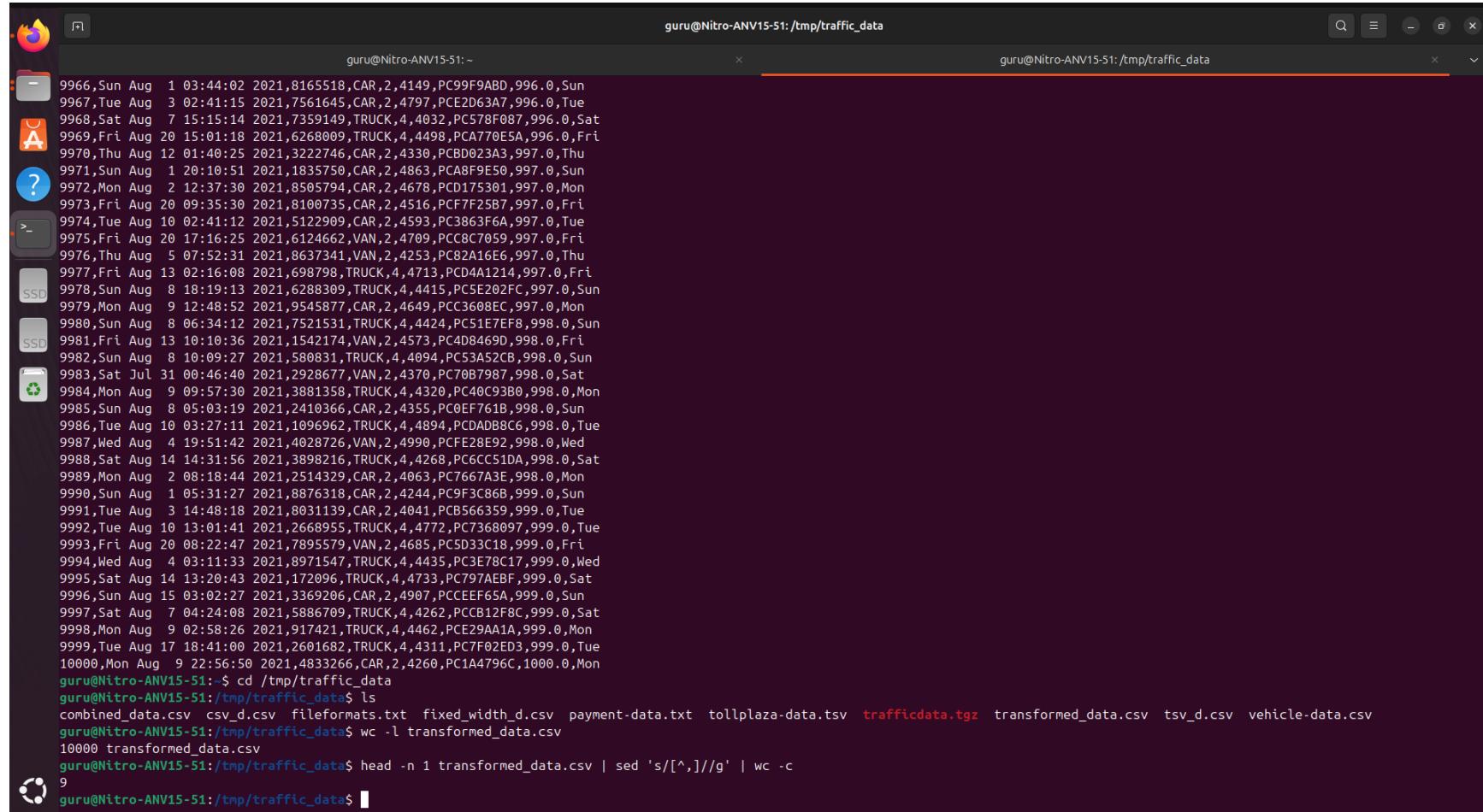
	State	Dag Id	Logical Date	Run Id	Run Type	Queued At	Start Date	End Date	Note	External Trigger	Conf	Duration
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-15, 14:49:37	manual_2025-02-15T14:49:37.138441+00:00	manual	2025-02-15, 14:49:37	2025-02-15, 14:49:38	2025-02-15, 14:49:50	True	{}	12s	
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-15, 14:29:35	manual_2025-02-15T14:29:35.486097+00:00	manual	2025-02-15, 14:29:35	2025-02-15, 14:29:36	2025-02-15, 14:29:49	True	{}	12s	
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-15, 14:20:44	manual_2025-02-15T14:20:44.849654+00:00	manual	2025-02-15, 14:20:44	2025-02-15, 14:20:45	2025-02-15, 14:20:59	True	{}	13s	
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-15, 14:17:48	manual_2025-02-15T14:17:48.110005+00:00	manual	2025-02-15, 14:17:48	2025-02-15, 14:17:50	2025-02-15, 14:18:03	True	{}	13s	
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-14, 00:00:00	scheduled_2025-02-14T00:00:00+00:00	scheduled	2025-02-15, 14:12:35	2025-02-15, 14:12:35	2025-02-15, 14:12:53	False	{}	17s	
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-13, 22:21:34	manual_2025-02-13T22:21:34.767622+00:00	manual	2025-02-13, 22:21:34	2025-02-13, 22:21:35	2025-02-13, 22:22:22	True	{}	47s	
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-13, 00:00:00	scheduled_2025-02-13T00:00:00+00:00	scheduled	2025-02-14, 05:18:51	2025-02-14, 05:18:51	2025-02-14, 05:19:17	False	{}	25s	
<input type="checkbox"/>	success	traffic_data_ETL	2025-02-12, 00:00:00	scheduled_2025-02-12T00:00:00+00:00	scheduled	2025-02-13, 22:21:35	2025-02-13, 22:21:35	2025-02-13, 22:22:22	False	{}	47s	

At the bottom left, there is a footer with the text "Version: v2.10.5" and "Git Version: .release:b93c3db6b1641b0840bd15ac7d05bc58ff2cccbf".



Results

- The extracted and transformed data was saved in the /tmp/traffic_data directory.
- The final processed file, transformed.csv, contained the expected transformed data.



The screenshot shows a terminal window with two tabs. The left tab displays a large amount of log-like data with timestamp, date, and vehicle details. The right tab shows the command-line interface used for the process.

```
guru@Nitro-ANV15-51:~
```

```
9966,Sun Aug  1 03:44:02 2021,8165518,CAR,2,4149,PC99F9ABD,996.0,Sun
9967,Tue Aug  3 02:41:15 2021,7561645,CAR,2,4797,PCE2D63A7,996.0,Tue
9968,Sat Aug  7 15:15:14 2021,7359149,TRUCK,4,4032,PC578F087,996.0,Sat
9969,Fri Aug 20 15:01:18 2021,6268009,TRUCK,4,4498,PCA770E5A,996.0,Fri
9970,Thu Aug 12 01:40:25 2021,3222746,CAR,2,4330,PCBD023A3,997.0,Thu
9971,Sun Aug 12 20:10:51 2021,1835750,CAR,2,4863,PCA8F9E50,997.0,Sun
9972,Mon Aug  2 12:37:30 2021,8505794,CAR,2,4678,PCD175301,997.0,Mon
9973,Fri Aug 20 09:35:30 2021,8100735,CAR,2,4516,PCF7F25B7,997.0,Fri
9974,Tue Aug 10 02:41:12 2021,5122909,CAR,2,4593,PC3863F6A,997.0,Tue
9975,Fri Aug 20 17:16:25 2021,6124662,VAN,2,4709,PCC8C7059,997.0,Fri
9976,Thu Aug  5 07:52:31 2021,8637341,VAN,2,4253,PC82A16E6,997.0,Thu
9977,Fri Aug 13 02:16:08 2021,698798,TRUCK,4,4713,PCD4A1214,997.0,Fri
9978,Sun Aug  8 18:19:13 2021,6288309,TRUCK,4,4415,PC5E202FC,997.0,Sun
9979,Mon Aug  9 12:48:52 2021,9545877,CAR,2,4649,PCC3608EC,997.0,Mon
9980,Sun Aug  8 06:34:12 2021,7521531,TRUCK,4,4424,PC51E7EF8,998.0,Sun
9981,Fri Aug 13 10:10:36 2021,1542174,VAN,2,4573,PC4D8469D,998.0,Fri
9982,Sun Aug  8 10:09:27 2021,580831,TRUCK,4,4094,PC53A52CB,998.0,Sun
9983,Sat Jul 31 00:46:40 2021,2928677,VAN,2,4370,PC70B7987,998.0,Sat
9984,Mon Aug  9 09:57:30 2021,3881358,TRUCK,4,4320,PC40C93B0,998.0,Mon
9985,Sun Aug  8 05:03:19 2021,2410366,CAR,2,4355,PC9EF761B,998.0,Sun
9986,Tue Aug 10 03:27:11 2021,1096962,TRUCK,4,4894,PCDAD88C6,998.0,Tue
9987,Wed Aug  4 19:51:42 2021,4028726,VAN,2,4990,PCFE28E92,998.0,Wed
9988,Sat Aug 14 14:31:56 2021,3898216,TRUCK,4,4268,PC6CC51DA,998.0,Sat
9989,Mon Aug  2 08:18:44 2021,2514329,CAR,2,4063,PC7667A3E,998.0,Mon
9990,Sun Aug  1 05:31:27 2021,8876318,CAR,2,4244,PC9F3C86B,999.0,Sun
9991,Tue Aug  3 14:48:18 2021,8031139,CAR,2,4041,PCB566359,999.0,Tue
9992,Tue Aug 10 13:01:41 2021,2668955,TRUCK,4,4772,PC7368097,999.0,Tue
9993,Fri Aug 20 08:22:47 2021,7895579,VAN,2,4685,PCSD33C18,999.0,Fri
9994,Wed Aug  4 03:11:33 2021,8971547,TRUCK,4,4435,PC3E78C17,999.0,Wed
9995,Sat Aug 14 13:20:43 2021,172096,TRUCK,4,4733,PC797AEFB,999.0,Sat
9996,Sun Aug 15 03:02:27 2021,3369206,CAR,2,4907,PCCEE65A,999.0,Sun
9997,Sat Aug  7 04:24:08 2021,5886709,TRUCK,4,4262,PCCB12F8C,999.0,Sat
9998,Mon Aug  9 02:58:26 2021,917421,TRUCK,4,4462,PCE29AA1A,999.0,Mon
9999,Tue Aug 17 18:41:06 2021,2601682,TRUCK,4,4311,PC7F02ED3,999.0,Tue
10000,Mon Aug  9 22:56:50 2021,4833266,CAR,2,4260,PC1A4796C,1000.0,Mon
guru@Nitro-ANV15-51:~$ cd /tmp/traffic_data
guru@Nitro-ANV15-51:/tmp/traffic_data$ ls
combined_data.csv  csv_d.csv  fileformats.txt  fixed_width_d.csv  payment-data.txt  tollplaza-data.tsv  trafficdata.tgz  transformed_data.csv  tsv_d.csv  vehicle-data.csv
guru@Nitro-ANV15-51:/tmp/traffic_data$ wc -l transformed_data.csv
10000 transformed_data.csv
guru@Nitro-ANV15-51:/tmp/traffic_data$ head -n 1 transformed_data.csv | sed 's/[,]/,/g' | wc -c
9
guru@Nitro-ANV15-51:/tmp/traffic_data$
```

Observations

1. DAG Structure & Task Execution

- The DAG consists of several tasks, each responsible for different stages of the ETL process:
 - **create_directories**: Ensures necessary directories are available before data extraction.
 - **download_data**: Fetches raw traffic data, possibly from an API or external source.
 - **extract_data**: Extracts relevant portions from the downloaded data.
 - **extract_csv, extract_tsv, extract_fixed_width**: Handle extraction from different file formats.
 - **combine_data**: Merges data from multiple sources into a unified dataset.
 - **transform_data**: Applies necessary data cleaning and processing before storage or analysis.
- The execution follows a structured workflow where tasks are executed in a sequence, ensuring data integrity and correctness.

2. Task Execution Status

- All tasks in the DAG have been successfully executed, as indicated by the green status markers.
- No failed or pending tasks were observed, suggesting that the ETL pipeline is functioning without errors.
- The DAG execution timestamps indicate consistent processing, with runs scheduled daily.

3. Task Duration Analysis

- The **Task Duration Chart** highlights the execution time for individual tasks over multiple DAG runs.
- **download_data** consistently records the highest execution time, indicating that fetching raw data is a time-consuming process.
- **combine_data** also takes a relatively longer time compared to other tasks, which is expected as it integrates multiple data sources.

- Other tasks, such as **create_directories** and **extract_data**, execute within milliseconds to seconds, suggesting efficient performance.

4. Execution Trends Over Time

- The **Run Duration Chart** provides insights into total DAG execution time across multiple days (February 13–15, 2025).
- The DAG run on **February 13, 2025**, took the longest time (~50 seconds), while subsequent runs show reduced execution times.
- The decreasing execution time trend indicates potential optimizations in data handling or reduced processing loads.
- Recent DAG executions have stabilized, maintaining a consistent duration across multiple runs.

5. Gantt Chart Observations

- The **Gantt Chart View** provides a timeline representation of task execution.
- Tasks are executed sequentially, with minimal idle time between them, suggesting efficient scheduling.
- Green bars in the chart confirm successful completion, while gray bars indicate queued tasks waiting for execution.
- The **combine_data** task appears to have the longest execution span, aligning with findings from the Task Duration analysis.

6. Airflow Monitoring & Interface Features

- The Airflow UI provides multiple visualization options such as:
 - **Graph View:** Shows task dependencies and execution flow.
 - **Gantt Chart View:** Displays task execution duration and scheduling gaps.
 - **Run Duration View:** Tracks execution time trends over multiple DAG runs.
 - **Task Duration View:** Highlights individual task performance and bottlenecks.
- The system actively monitors DAG runs, ensuring any performance issues or failures are easily identifiable.

Conclusion

The **traffic_data_ETL** pipeline in Apache Airflow executes successfully with structured task dependencies and efficient scheduling. The **download_data** and **combine_data** tasks require the longest execution time, making them key areas to monitor for performance improvements. The DAG's execution time has decreased over multiple runs, suggesting optimizations or variations in data volume. The monitoring capabilities in Airflow allow detailed tracking of task performance, ensuring smooth operation of the ETL workflow.