Ethiopia Soil Moisture Data Downloader Using Apache Airflow

Overview

This system automatically downloads Ethiopia soil moisture data from TAMSAT satellite dataset (1986-2024) and converts it to TIFF format.

What You Get

- Automated downloads from TAMSAT database
- Ethiopia-specific clipping (removes other countries)
- TIFF format output
- Flexible year/month selection

Prerequisites

1. Install Docker Desktop

- Download: https://www.docker.com/products/docker-desktop/
- Install and ensure Docker is running
- Verify: Open terminal and run `docker --version`

Installation Steps

Technical Dependencies

- Apache Airflow 2.5.1 V
- Docker Latest version

Python Libraries

Library	Purpose	Functionality
Xarray	NetCDF data handling	handel multi-dimensional datasets
regionmask	Geographic masking	Provides country boundaries for Ethiopia clipping
rioxarray	Geospatial operations	Handles coordinate systems and TIFF export
requests	HTTP downloads	Downloads NetCDF files from TAMSAT server
h5netcdf	NetCDF engine	Efficient NetCDF file reading backend

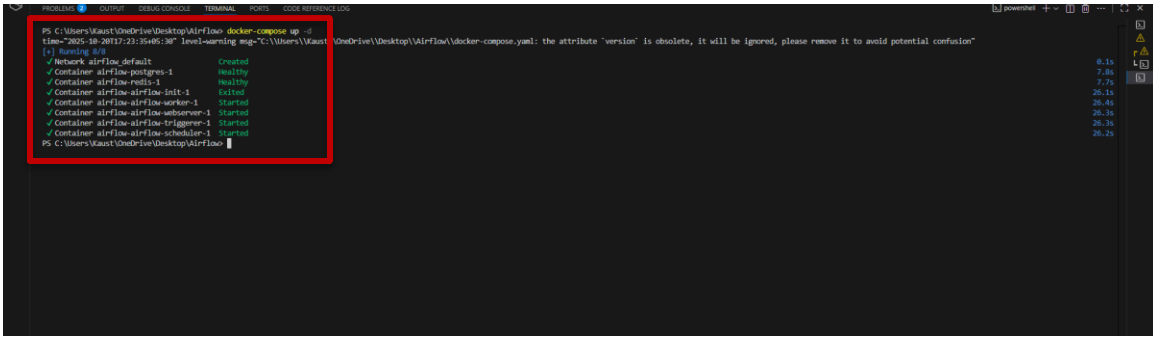
Step 1: Start the System

1. **Open VS code** in the folder

2. **Run command: <u>docker-compose up -d</u>**

3. **Wait 2-3 minutes** for all services to start

4. **Verify:** You should see "✓ Container airflow-airflow-webserver-1 Started"



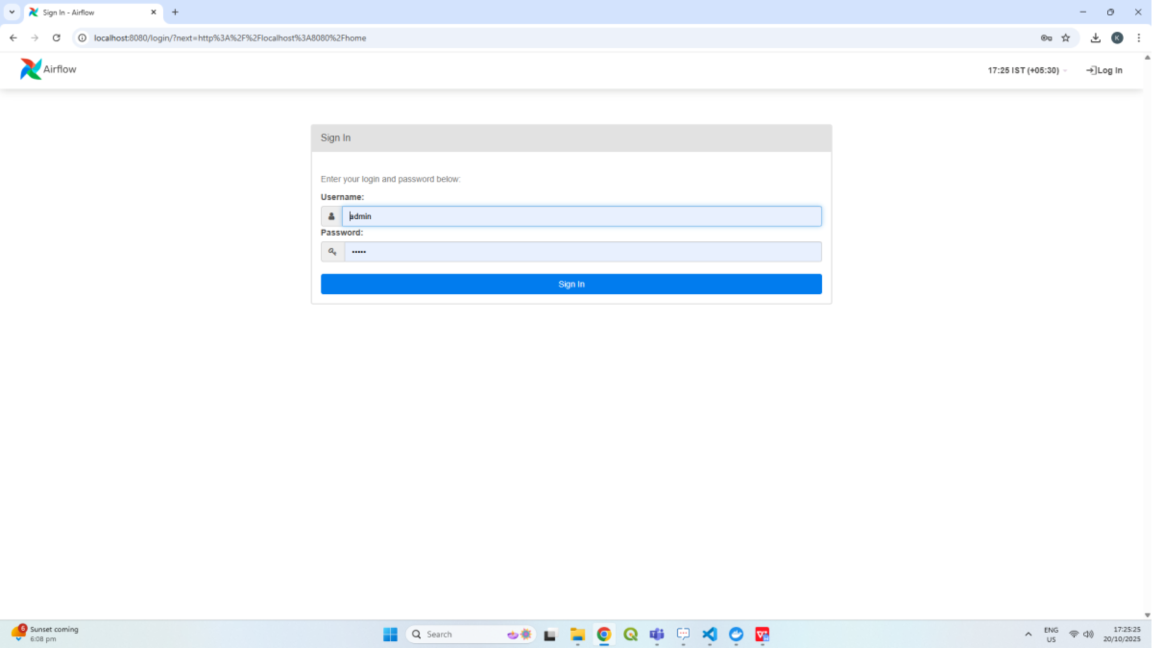
```
PS C:\Users\Kaust\OneDrive\Desktop\Airflow> docker-compose up -d
time="2025-10-20T17:23:35+05:30" level=warning msg="C:\\Users\\Kaust\\OneDrive\\Desktop\\Airflow\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 8/8
 ✓ Network airflow_default                Created        0.1s
 ✓ Container airflow-postgres-1           Healthy        7.8s
 ✓ Container airflow-redis-1              Healthy        7.7s
 ✓ Container airflow-airflow-init-1       Exited        26.1s
 ✓ Container airflow-airflow-worker-1     Started       26.4s
 ✓ Container airflow-airflow-webserver-1  Started       26.3s
 ✓ Container airflow-airflow-triggerer-1  Started       26.3s
 ✓ Container airflow-airflow-scheduler-1  Started       26.2s
PS C:\Users\Kaust\OneDrive\Desktop\Airflow>
```

Step 2: Access Web Interface

1. **Open browser** and go to: http://localhost:8080

2. **Login credentials:**

- Username: admin
- Password: admin



localhost:8080/login/?next=http%3A%2F%2Flocalhost%3A8080%2Fhome

Sign In

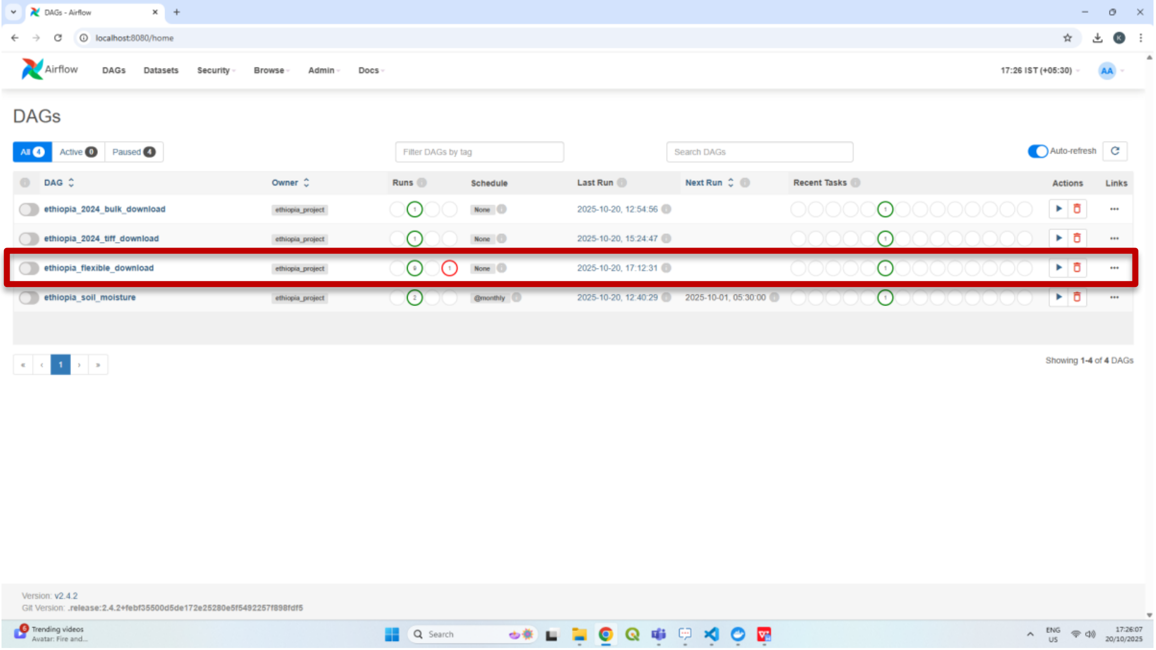
Enter your login and password below:

Username: admin

Password: •••••

Step 3: Airflow dashboard

You should see the Airflow dashboard with created DAG



localhost:8080/home

DAGs

All 4 | Active 0 | Paused 4

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
ethiopia_2024_bulk_download	ethiopia_project	1	None	2025-10-20, 12:54:56		1	▶ 🗑	•••
ethiopia_2024_tiff_download	ethiopia_project	1	None	2025-10-20, 15:24:47		1	▶ 🗑	•••
ethiopia_flexible_download	ethiopia_project	9 1	None	2025-10-20, 17:12:31		1	▶ 🗑	•••
ethiopia_soil_moisture	ethiopia_project	2	@monthly	2025-10-20, 12:40:29	2025-10-01, 05:30:00	1	▶ 🗑	•••

Showing 1-4 of 4 DAGs

Version: v2.4.2
Git Version: .release:2.4.2+febf35500d5de172e25280e5f5492257f898fdf5

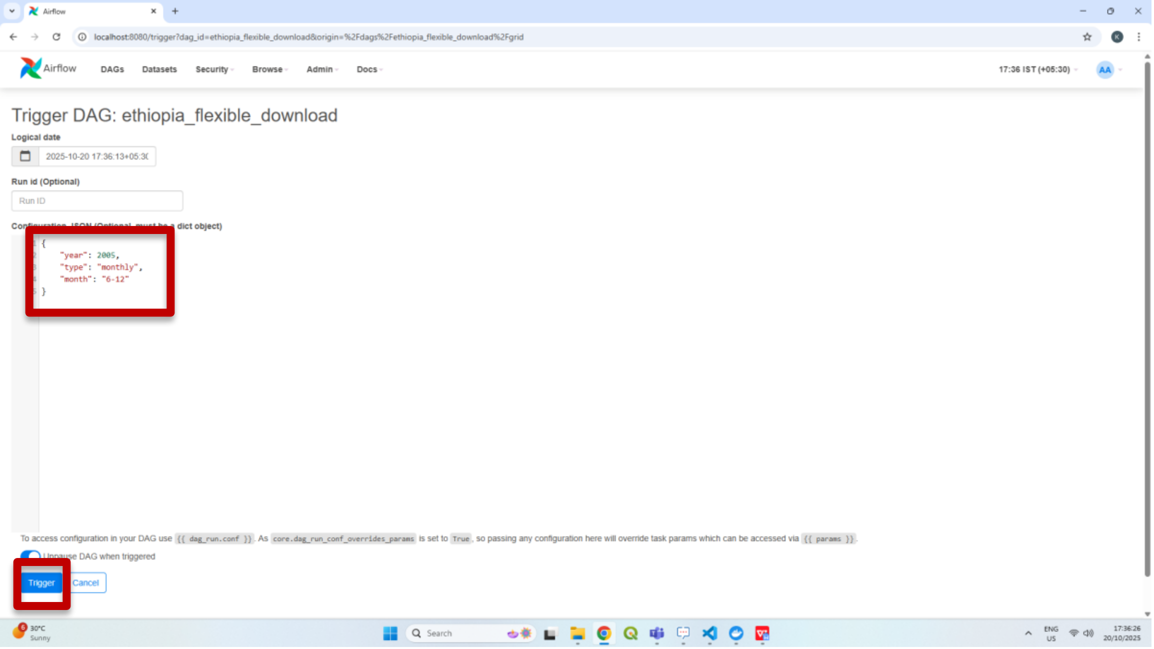
Step 4: Select Data range

Step 4.1: Find the DAG

1. **Look for:** ethiopia_flexible_download in the DAG list

Step 4.2: Configure Download

1. **Click** the play button next to the DAG
2. **Select:** <u>Trigger DAG w/ Config</u>
3. **JSON editor will open -** enter your configuration



Trigger DAG: ethiopia_flexible_download

Logical date: 2025-10-20 17:36:13+05:3(

Run id (Optional): Run ID

Configuration JSON (Optional, must be a dict object)

```
{
    "year": 2005,
    "type": "monthly",
    "month": "6-12"
}
```

To access configuration in your DAG use `{{ dag_run.conf }}`. As `core.dag_run_conf_overrides_params` is set to `True`, so passing any configuration here will override task params which can be accessed via `{{ params }}`.

Unpause DAG when triggered

Download All Months of a Year:

```
{
"year": 2020,
"type": "monthly",
"month": "all"
}
```

Download mutiple Month:

{

"year": 2020,

"type": "monthly",

"month": "**<u>6-12</u>**"

}

Download single Month:

{

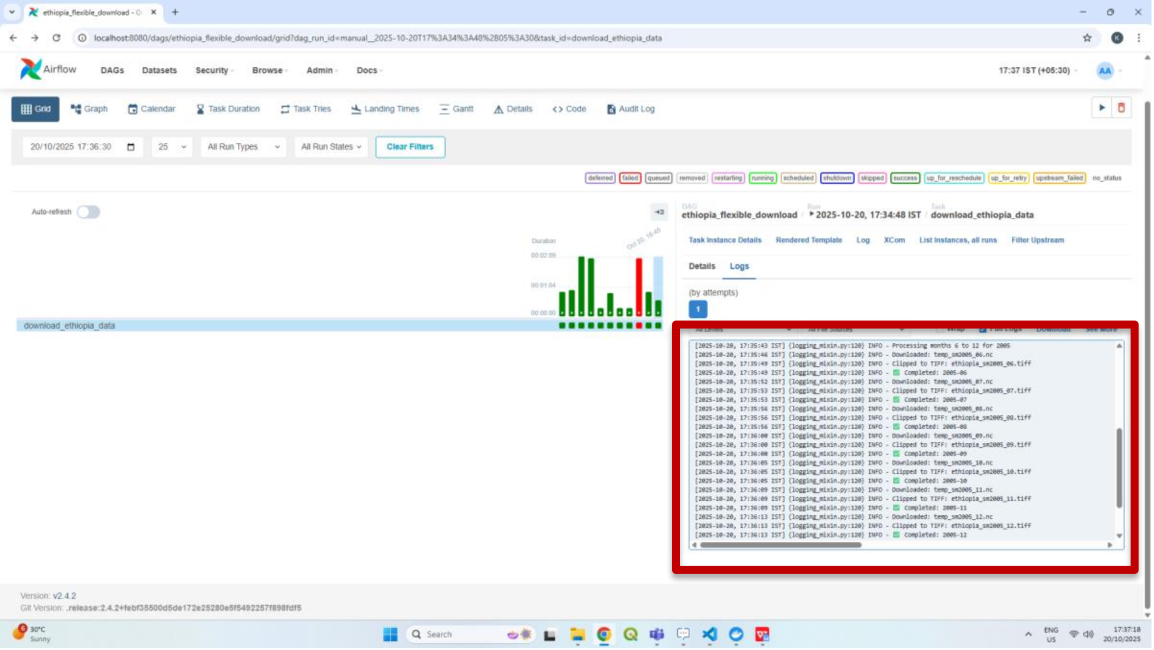
"year": 1990,

"type": "monthly",

"month": "**<u>5</u>**"

Step 5: Start Workflow

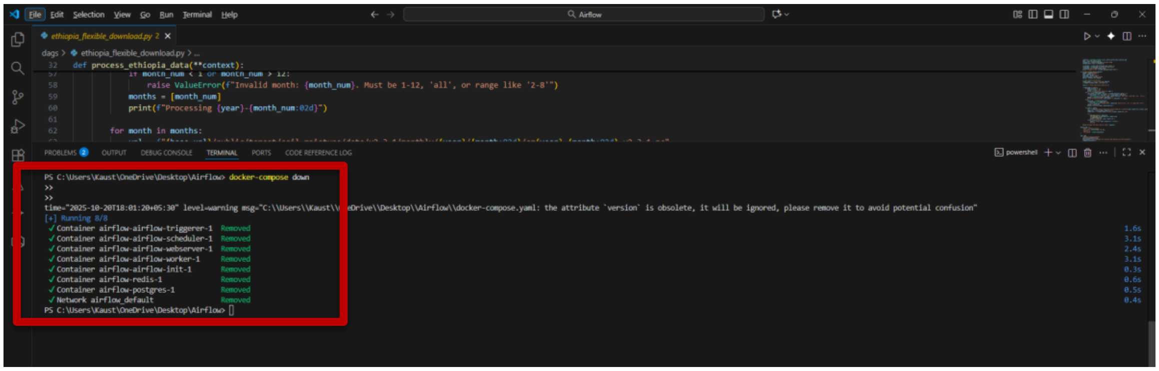
1. **Click:** "Trigger" button
2. **Monitor progress** by clicking on the DAG run
3. **View logs** to see download status



```
[2025-10-20, 17:35:43 IST] {logging_mixin.py:120} INFO - Processing months 6 to 12 for 2005
[2025-10-20, 17:35:46 IST] {logging_mixin.py:120} INFO - Downloaded: temp_sm2005_06.nc
[2025-10-20, 17:35:49 IST] {logging_mixin.py:120} INFO - Clipped to TIFF: ethiopia_sm2005_06.tiff
[2025-10-20, 17:35:49 IST] {logging_mixin.py:120} INFO - ✅ Completed: 2005-06
[2025-10-20, 17:35:52 IST] {logging_mixin.py:120} INFO - Downloaded: temp_sm2005_07.nc
[2025-10-20, 17:35:53 IST] {logging_mixin.py:120} INFO - Clipped to TIFF: ethiopia_sm2005_07.tiff
[2025-10-20, 17:35:53 IST] {logging_mixin.py:120} INFO - ✅ Completed: 2005-07
[2025-10-20, 17:35:56 IST] {logging_mixin.py:120} INFO - Downloaded: temp_sm2005_08.nc
[2025-10-20, 17:35:56 IST] {logging_mixin.py:120} INFO - Clipped to TIFF: ethiopia_sm2005_08.tiff
[2025-10-20, 17:35:56 IST] {logging_mixin.py:120} INFO - ✅ Completed: 2005-08
[2025-10-20, 17:36:00 IST] {logging_mixin.py:120} INFO - Downloaded: temp_sm2005_09.nc
[2025-10-20, 17:36:00 IST] {logging_mixin.py:120} INFO - Clipped to TIFF: ethiopia_sm2005_09.tiff
[2025-10-20, 17:36:00 IST] {logging_mixin.py:120} INFO - ✅ Completed: 2005-09
[2025-10-20, 17:36:05 IST] {logging_mixin.py:120} INFO - Downloaded: temp_sm2005_10.nc
[2025-10-20, 17:36:05 IST] {logging_mixin.py:120} INFO - Clipped to TIFF: ethiopia_sm2005_10.tiff
[2025-10-20, 17:36:05 IST] {logging_mixin.py:120} INFO - ✅ Completed: 2005-10
[2025-10-20, 17:36:09 IST] {logging_mixin.py:120} INFO - Downloaded: temp_sm2005_11.nc
[2025-10-20, 17:36:09 IST] {logging_mixin.py:120} INFO - Clipped to TIFF: ethiopia_sm2005_11.tiff
[2025-10-20, 17:36:09 IST] {logging_mixin.py:120} INFO - ✅ Completed: 2005-11
[2025-10-20, 17:36:13 IST] {logging_mixin.py:120} INFO - Downloaded: temp_sm2005_12.nc
[2025-10-20, 17:36:13 IST] {logging_mixin.py:120} INFO - Clipped to TIFF: ethiopia_sm2005_12.tiff
[2025-10-20, 17:36:13 IST] {logging_mixin.py:120} INFO - ✅ Completed: 2005-12
```

Step 6: Stop System

docker-compose down



```
def process_ethiopia_data(**context):
        if month_num < 1 or month_num > 12:
            raise ValueError(f"Invalid month: {month_num}. Must be 1-12, 'all', or range like '2-8'")
        months = [month_num]
        print(f"Processing {year}-{month_num:02d}")

    for month in months:
```

```
PS C:\Users\Kaust\OneDrive\Desktop\Airflow> docker-compose down
>>
>>
time="2025-10-20T18:01:20+05:30" level=warning msg="C:\\Users\\Kaust\\OneDrive\\Desktop\\Airflow\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 8/8
 ✓ Container airflow-airflow-triggerer-1  Removed                                                    1.6s
 ✓ Container airflow-airflow-scheduler-1  Removed                                                    3.1s
 ✓ Container airflow-airflow-webserver-1  Removed                                                    2.4s
 ✓ Container airflow-airflow-worker-1     Removed                                                    3.1s
 ✓ Container airflow-airflow-init-1       Removed                                                    0.3s
 ✓ Container airflow-redis-1              Removed                                                    0.6s
 ✓ Container airflow-postgres-1           Removed                                                    0.5s
 ✓ Network airflow_default                Removed                                                    0.4s
PS C:\Users\Kaust\OneDrive\Desktop\Airflow> []
```

Where Files Are Saved

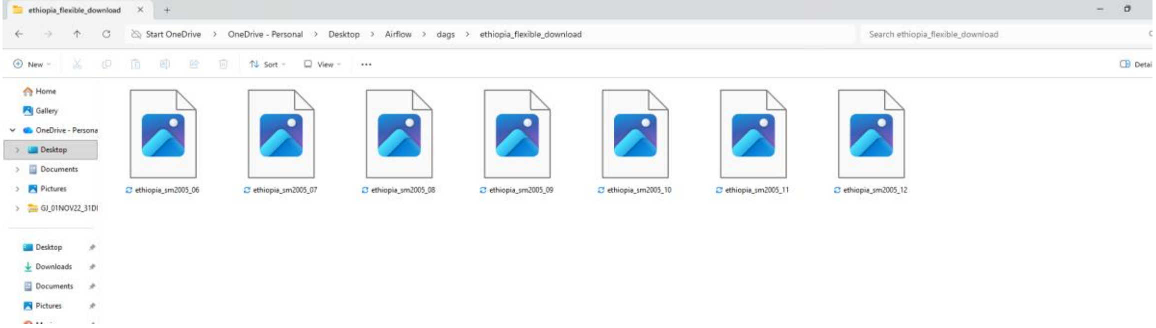
Local Storage Location

Your downloaded TIFF files are saved to:

[Your-Extraction-Folder]/dags/ethiopia_flexible_download/

Example: If you extracted to **C:\MyProjects\Ethiopia-Airflow\** , files will be in:

C:\MyProjects\Ethiopia-Airflow\dags\ethiopia_flexible_download\



ethiopia_sm2005_06
ethiopia_sm2005_07
ethiopia_sm2005_08
ethiopia_sm2005_09
ethiopia_sm2005_10
ethiopia_sm2005_11
ethiopia_sm2005_12

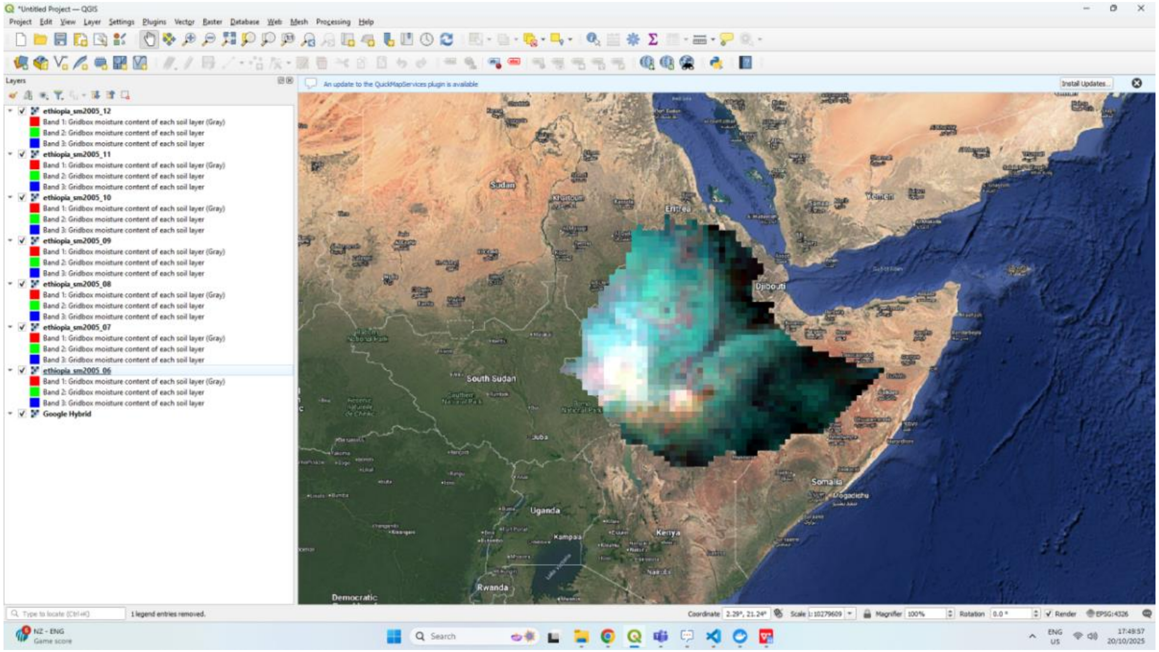
File Naming Convention

Files are named as: ethiopia_sm[YEAR]_[MONTH].tiff

Examples:

- ethiopia_sm2005_06.tiff (January 2020)
- ethiopia_sm2005_07.tiff (February 2020)
- ethiopia_sm2005_08/.tiff (December 1990)

Tiff in QGIS



An update to the QuickMapServices plugin is available

- ☑ ethiopia_sm2005_12
  - 🟥 Band 1: Gridbox moisture content of each soil layer (Gray)
  - 🟩 Band 2: Gridbox moisture content of each soil layer
  - 🟦 Band 3: Gridbox moisture content of each soil layer
- ☑ ethiopia_sm2005_11
  - 🟥 Band 1: Gridbox moisture content of each soil layer (Gray)
  - 🟩 Band 2: Gridbox moisture content of each soil layer
  - 🟦 Band 3: Gridbox moisture content of each soil layer
- ☑ ethiopia_sm2005_10
  - 🟥 Band 1: Gridbox moisture content of each soil layer (Gray)
  - 🟩 Band 2: Gridbox moisture content of each soil layer
  - 🟦 Band 3: Gridbox moisture content of each soil layer
- ☑ ethiopia_sm2005_09
  - 🟥 Band 1: Gridbox moisture content of each soil layer (Gray)
  - 🟩 Band 2: Gridbox moisture content of each soil layer
  - 🟦 Band 3: Gridbox moisture content of each soil layer
- ☑ ethiopia_sm2005_08
  - 🟥 Band 1: Gridbox moisture content of each soil layer (Gray)
  - 🟩 Band 2: Gridbox moisture content of each soil layer
  - 🟦 Band 3: Gridbox moisture content of each soil layer
- ☑ ethiopia_sm2005_07
  - 🟥 Band 1: Gridbox moisture content of each soil layer (Gray)
  - 🟩 Band 2: Gridbox moisture content of each soil layer
  - 🟦 Band 3: Gridbox moisture content of each soil layer
- ☑ ethiopia_sm2005_06
  - 🟥 Band 1: Gridbox moisture content of each soil layer (Gray)
  - 🟩 Band 2: Gridbox moisture content of each soil layer
  - 🟦 Band 3: Gridbox moisture content of each soil layer
- ☑ Google Hybrid

Type to locate (Ctrl+K) 1 legend entries removed.

Coordinate 2.29°, 21.24° Scale 1:10279609 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:4326

17:49:57
20/10/2025

How Local Storage Works

- Docker Volume Mapping: The system automatically maps container storage to your local folder
- Real-time Sync: Files appear in your local folder as soon as they're processed
- No Manual Copy Needed: Files are directly accessible on your PC

Available Data Range

Years Available: 1983 - 2024

- Historical Data: 1983-2024 (complete years)

File Locations Summary

- Configuration: docker-compose.yaml
- DAG Code: dags/ethiopia_flexible_download.py
- Downloaded Data: dags/ethiopia_flexible_download/
- System Logs: logs/ folder