GFDL Data Mirroring Pipeline: Documentation

Version: 5.0
Date: 2025-11-07
Author: Abdullah Mubasshar

1. Executive Summary

This document provides a complete overview of the GFDL Data Mirroring Pipeline, a project to secure and process irreplaceable climate datasets from NOAA's GFDL. All planned project phases are now complete.

The pipeline has successfully mirrored all available Priority 1, 2, and 3 data for the target variables from the GFDL-ESM4 (historical and future scenarios) and GFDL-SPEAR (seasonal forecast historical) models. A key success was the "Full Discovery" process, which uncovered and downloaded a rich dataset including daily-frequency data and multiple ensemble members, significantly enhancing the value of the mirrored dataset.

The raw global data has been processed for five key geographical regions: Southern Africa, East Africa, West Africa, Latin America, and Southeast Asia, resulting in a ~99% reduction in data size per region. The entire pipeline is automated via a CRON job (currently disabled) for long-term maintenance.

This document serves as the definitive technical guide for the ML team to begin using the data and for future maintenance of the pipeline.

---

2. The Data Pipeline

The pipeline is a modular Python application designed for scalability and maintainability. It is controlled entirely by a central configuration file.

2.1. Architecture & Structure

The code is located at `/mnt/datalake/abdullah/gfdl_pipeline/`. The structure remains modular, where the `config.yaml` file specifies global search parameters (ensembles, grids, versions) which are then applied to groups of variables organized by time frequency (e.g., Monthly, Daily).

```
gfdl_pipeline/
├── main.py               # orchestrator: Runs the entire pipeline.
├── config.yaml          # control File: Defines ALL parameters and
targets.
├── modules/
    ├── downloader.py          # handles file downloads.
    ├── processor.py           # handles geographical subsetting.
    ├── validator.py           # contains data integrity check functions.
└── utils.py             # helper functions.
```

2.2. Multi-Region Workflow

The pipeline's workflow is now region-aware:

1. **Load Configuration:** `main.py` reads the `config.yaml` file, including the new list of `processing_regions`.
2. **Discover and Download:** It loops through the defined datasets (e.g., `ESM4_historical`). For each variable, it discovers and downloads the raw global file into the central staging directory (`/mnt/datalake/abdullah/gfdl_mirror/raw/`). This step is skipped if the file already exists.
3. **Iterate and Process:** Upon securing a raw file, the pipeline **loops through every region defined in `processing_regions`**. For each region, it:
    - Calls the `processor.py` module.
    - Subsets the raw file using the region's specific bounding box.
    - Saves the processed, region-specific file to a new, structured output directory.

2.3. How to Manage Regions

The pipeline is controlled by the `config.yaml` file. To add, remove, or modify a geographical scope, simply edit the `processing_regions` list. No code changes are required.

Example `config.yaml` structure for defining regions:

```
processing_regions:
  - name: "Southern_Africa"
    bounding_box: { min_lon: 12.0, max_lon: 41.0, min_lat: -26.0, max_lat:
-4.0 }
  - name: "East_Africa"
    bounding_box: { min_lon: 22.0, max_lon: 52.0, min_lat: -12.0, max_lat:
22.0 }
  # Add new regions here...
```

3. Data Specifications: For the ML Team

3.1. Raw Data Specifications (NOT to be used by ML Team)

The specifications for the raw data remain unchanged. It is global, large, and serves as the temporary source material for processing.

- **Location:** Archived to Azure Blob Storage.
- **Usage:** This directory is for pipeline use only. **The ML team should NOT use this data directly.**

3.2. Processed Data Specifications

This is the most critical update for the ML team. The processed data is now organised by region at the top level.

- **Location:** `/mnt/datalake/abdullah/gfdl_mirror/`
- **New Directory Structure:**

```
/gfdl_mirror/
├── Southern_Africa/
│   └── esm4/
│       └── historical/
│           ├── radiation/
│           ├── soil_moisture/
│           ├── energy_fluxes/
│           └── temperature/
│       └── scenarios/
│   └── spear/
│
├── East_Africa/
│   └── esm4/
│       └── ...
├── West_Africa/
│   └── esm4/
│       └── ...
└── Latin_America/
    └── esm4/
        └── ...
```

- **File Naming and Format:** Unchanged. Files are NetCDF4 (.nc) and retain their original, descriptive names.
- **Data Content:** Each file is a geographically subsetted version of the raw global data, containing only the data points within the bounding box of its parent region. All notes regarding nan values and the use of `xarray` still apply.

3.3 File Naming Convention

Filenames are descriptive and contain all necessary metadata. Example:
`ts_Amon_GFDL-ESM4_historical_r3i1p1f1_gr1_v20180701_195001-201412.nc`

Within each regional folder, the structure is:
<model>/<type>/<variable_category>/<filename.nc>

Examples:

- .../Latin_America/esm4/historical/temperature/
- .../Latin_America/esm4/scenarios/ssp245/soil_moisture/
- .../Latin_America/spear/forecast_historical/precipitation/

Component	Meaning
Variable	**ts** (Surface Temperature)
MIP Table / Frequency	**Amon** (Monthly)
Model	**GFDL-ESM4**
Experiment	**historical**
Ensemble Member	**r3i1p1f1**
Grid	**gr1**
Version	**v20180701**
Time Period	**195001 – 201412**

3.4 Key Data Characteristics

- **Format:** NetCDF4 (.nc)
- **Transformation:** Each file contains a geographically subsetted “slice” of the global data for its parent region.
- **Coordinates:** Standard Latitude/Longitude (WGS 84).
- **Units:** Standard scientific units (e.g., Temperature in K, Radiation in W m^{-2}). Units are stored in metadata and readable via xarray.
- **nan Values:** Grid cells over oceans or large water bodies contain nan values for land-based variables. Your code must handle these appropriately.

3.5 How to Load the Data (Python Example)

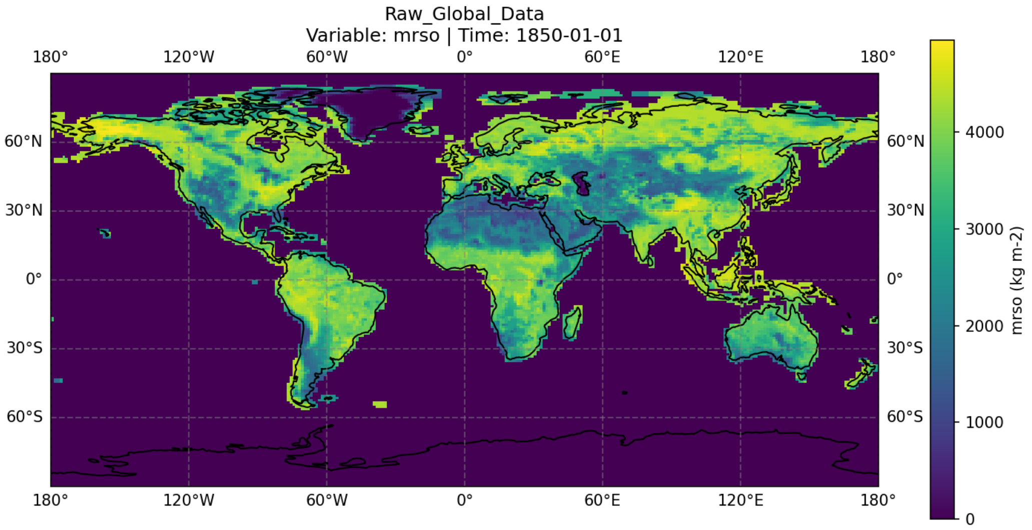
```
import xarray as xr

# Example: Load a processed file for Latin America
file_path =
"/mnt/datalake/abdullah/gfdl_mirror/Latin_America/esm4/historical/temperatu
re/ts_Amon_GFDL-ESM4_historical_r3i1p1f1_gr1_v20180701_195001-201412.nc"

# Open the dataset
ds = xr.open_dataset(file_path)

# Print a summary of the file’s content
print(ds)

# Example: Get data for a specific location (e.g., São Paulo)
sao_paulo_temp = ds['ts'].sel(lat=-23.5, lon=-46.6, method='nearest')

# Plot the time series
sao_paulo_temp.plot()
```

3.6 Critical Data Notes

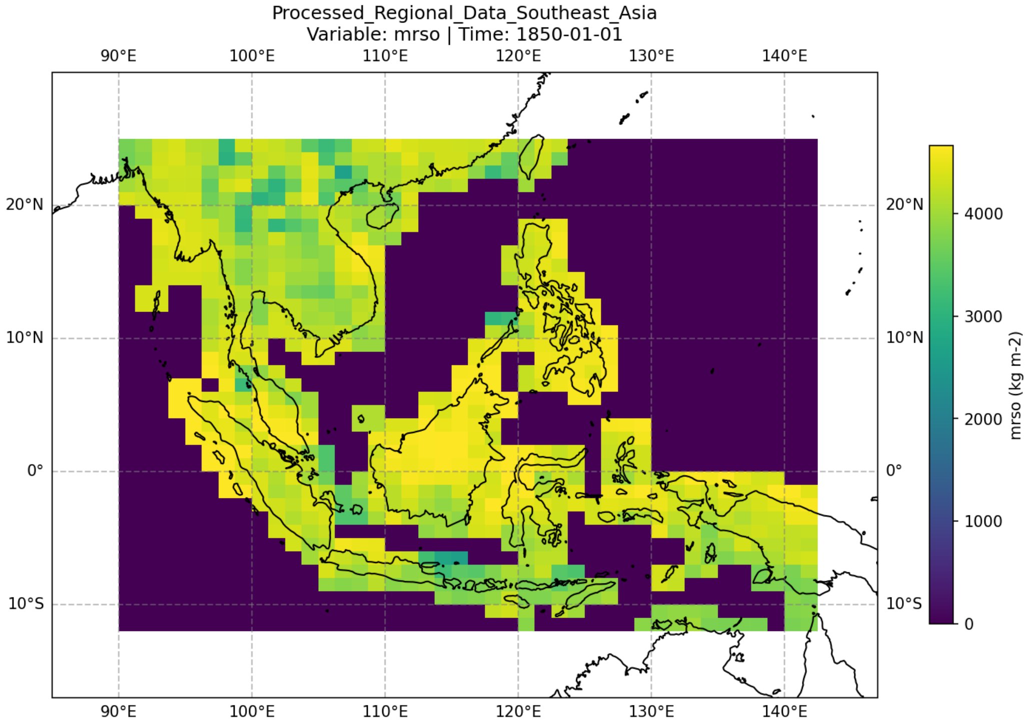
- **NaN Values:** Grid cells over oceans will contain NaN values for land-based variables. Code must handle or ignore them.
- **Units:** Stored in metadata (e.g., temperature in Kelvin, precipitation in kg m-2 s-1).

4. Example Visualisations

4.1 Global Raw Data (var “mrso”, region “global”, time period “1850”)



4.2 Processed Data (var “mrso”, region “Southeast Asia”, time period “1850”)



5. How to Run the Pipeline & Tools

Enter project directory and activate the virtual environment:

```
source /mnt/datalake/abdullah/GFDL/venv/bin/activate
```

All commands must be run from

```
/mnt/datalake/abdullah/GFDL/gfdl_pipeline/
```

Run Full Pipeline (Main Pipeline):

```
python main.py
```

Run a Specific Dataset:

```
python main.py --name <dataset_name>ESM4_SPEAR_historical
```

```
e.g., python main.py --name ESM4_SPEAR_historical
```

Run Data Integrity Checks:

```
python integrity_checker.py
```

Results are saved to:

```
/mnt/datalake/abdullah/gfdl_mirror/validation.log
```

Generate Metadata:

```
python generate_metadata.py
```

Visualise Data:

- **Text Summary:** `python visualise_cdf.py`
- **Map Visualization:** `python visualise_map.py`

Automation (CRON Job)

- **Schedule:** Every Sunday at 2:00 AM
- **Script:** `run_pipeline.sh` (handles venv and locking)
- **Logs:** `/mnt/datalake/abdullah/gfdl_mirror/cron.log`

Note:

- A weekly CRON job is already configured but disabled.
- To re-enable: open crontab (crontab -e) and uncomment the line containing run_pipeline.sh.

---

6. Data Management & Archiving

- **Raw Data:** 2.3TB+ archived to Azure Blob Storage using azcopy.
- **Processed Data:** ~1TB remains locally for fast ML access across regions.

---

7. Phase 4: Documentation & Metadata Summary

- **METADATA.md** contains full variable-level dictionary.
- **Scientific References:**
  - GFDL-ESM4: https://doi.org/10.1029/2019MS002015
  - GFDL-SPEAR: https://doi.org/10.1029/2019MS001895

- **Contact:**
  - GFDL.Climate.Model.Info@noaa.gov (data content)
  - oar.gfdl.webmaster-data1@noaa.gov (web development, system administration, etc.)

Appendix A: List of Successfully Mirrored Variables

The following variables were successfully located, downloaded, and processed. The data is analysis-ready in the specified subdirectories within `/mnt/datalake/abdullah/gfdl_mirror/[processing_region]/[source_model]`.

Important Note on Ensemble Members: The logs confirm that only the primary ensemble member, **`r1i1p1f1`**, was consistently available from the data sources. Attempts to download other ensemble members (`r2i1p1f1`, `r3i1p1f1, etc.`) resulted in "404 Not Found" errors in most cases (check the table below for detailed analysis), confirming their absence.

Category	Variable ID	Description	Source Model	Available Frequencies	Ensembles
Energy Fluxes	hfls	Latent Heat Flux	ESM4	Monthly (Amon)	r1i1p1f1
Energy Fluxes	hfss	Sensible Heat Flux	ESM4	Monthly (Amon)	r1i1p1f1
Soil Moisture	mrro	Root Zone Soil Moisture	ESM4	Monthly (Lmon), Daily (day)	r1i1p1f1
Soil Moisture	mrso	Total Soil Moisture	ESM4	Monthly (Lmon), Daily (Day)	r1i1p1f1, r2i1p1f1, r3i1p1f1
Soil Moisture	mrsol	Soil Moisture by Layer	ESM4	Monthly (Emon)	r1i1p1f1
Radiation	rsds	Downward Shortwave Radiation	ESM4	Monthly (Amon), Daily (day)	r1i1p1f1, r3i1p1f1
Radiation	rss	Net Shortwave Radiation	ESM4	Monthly (Emon)	r1i1p1f1
Radiation	rsus	Upward Shortwave Radiation	ESM4	Monthly (Amon)	r1i1p1f1
Temperature	ts	Surface Temperature	ESM4	Monthly (Amon)	r1i1p1f1, r2i1p1f1, r3i1p1f1
Temperature	tsl	Soil Temperature by Layer	ESM4	Monthly (Lmon)	r1i1p1f1
Temperature	tas	Near-Surface Air Temp	SPEAR	Monthly (Amon), Daily (Day)	r1–r30 available

Category	Variable ID	Description	Source Model	Available Frequencies	Ensembles
Temperature	tasmax	Daily Maximum Near-Surface Air Temp	SPEAR	Daily (Day)	r1–r30 available
Temperature	tasmin	Daily Minimum Near-Surface Air Temp	SPEAR	Daily (Day)	r1–r30 available
Precipitation	pr	Precipitation	SPEAR	Monthly (Amon), Daily (Day)	r1–r30 available
Humidity	hus	Specific Humidity	SPEAR	Monthly (Amon)	r1–r30 available
Wind	sfcWind	Near-Surface Wind Speed	SPEAR	Monthly (Amon)	r1–r30 available

Appendix B: Attempted but Unavailable Variables

The robust pipeline attempted to find and download the following variables using multiple common MIP tables, but they were not available from the data source for the `historical` experiment.

Implication for ML Team: These variables are not present in the mirrored dataset and should not be included in analysis plans for the historical period.

Category	Variable ID	Description	Status
Radiation	`par`	Photosynthetically Active Radiation	**Unavailable** - File not found on server.
Energy Fluxes	`hfgs`	Ground Heat Flux	**Unavailable** - File not found on server.

Appendix C: Project Decisions Log

Priority 3 (NARCCAP): Cancelled.
Reason: Dataset only covers North America and does not overlap with project regions (Africa, Latin America, SE Asia). Proceeding would have consumed storage and pipeline time without benefit.

Appendix D: Geographical Scope

The following table details the bounding boxes used for each processed region.

Region Name	`min_lon`	`max_lon`	`min_lat`	`max_lat`
Southern_Africa	12.0	41.0	-26.0	-4.0
East_Africa	22.0	52.0	-12.0	22.0
West_Africa	-20.0	20.0	-4.0	25.0
Latin_America	-118.0	-34.0	-56.0	33.0
Southeast_Asia	90.0	142.0	-12.0	25.0

Geographical Scope

Southern African countries as in CDO Pipeline (DONE)

- Zambia
- Malawi
- Mozambique
- Zimbabwe
- Angola
- DRC
- Tanzania

- Namibia

East Africa (DONE)

- Kenya
- Uganda
- Rwanda
- Burundi
- Ethiopia
- Eritrea
- Djibouti
- Somalia
- South Sudan
- Sudan (sometimes classified as East Africa)

West Africa (DONE)

- Nigeria
- Ghana
- Senegal
- Mali
- Burkina Faso
- Niger
- Guinea
- Sierra Leone
- Liberia
- Ivory Coast (Côte d'Ivoire)
- Benin
- Togo
- Gambia
- Guinea-Bissau
- Cape Verde
- Mauritania

Latin America (DONE)

- Brazil
- Argentina
- Colombia
- Chile
- Mexico
- Costa Rica
- Peru
- Paraguay

- Bolivia
- Venezuela

Southeast Asia

- Philippines
- Indonesia
- Thailand
- Vietnam
- Malaysia
- Singapore
- Cambodia