Tuesday, 30 September 2025 1:02 PM

Earthdata Backend — Complete Command Log & Repro Steps (From Start)

> This document captures **only** the commands that are executed successfully, with brief explanations. It also provides a clean **Conda-based** reproduction path that reflects final working setup.

A) Acquire MODIS data (WSL)

1) Create a folder for MODIS downloads

```
cd /mnt/c/NASA_Project
mkdir MODIS_data
cd MODIS_data
```

2) Download the MODIS HDF file (Earthdata-auth cookies)

```
wget --load-cookies ~/.urs_cookies --save-cookies ~/.urs_cookies --keep-session-cookies -c \
"https://d13j1jds5ybppo.cloudfront.net/s3-8d80a21348b33e9c2ee6e0a8ac41d7b1/prod-lads.s3.us-west-2.amazonaws.com/MOD08_D3/MOD08_D3.A2025265.061.2025266143215.hdf?A-userid=astrodata_explorer&Expires=1758728865&Signature=YfYQD0JoAZbDzoEr14GFPSENaAWT1k~aWJ9XO0DHfOcdidK~cxRxC3VkKmpRDDmli0EyExscaTiT4i6debdo8t3-LJUmCoeGZY9mMePYrbakH92ckdTIUNrlRW3OugbW~0sXYL3oqmN93a3qCOrgabnvhg-A9iUhkPEiOlXXtiWY2oxGC1IsSfOIO4RuPM7v-mF1TbAgmCEjJ8bM0deqQI69qCjp6QT3Y8m3wUzr19OxPiD3sAWqCBzL-miyIFQduOLINv5DsAKbOoCKcdub~LFqPeUvWkQrUvUwllTuoErTh2Mz1FcZt-4ojJczjNEOok6PqyHogNXGDAbB4P7CJw__&Key-Pair-Id=K38HK9OA1FVIPQ"
```

3) Verify the download

```
ls -lh
```

4) Clean the filename (remove query suffix)

```
mv MOD08_D3.A2025265.061.2025266143215.hdf* MOD08_D3.A2025265.061.2025266143215.hdf
```

5) Verify the clean name

```
ls -lh
```

B) Inspect HDF with GDAL / h5py (initial steps shown)

> These were run early to validate tooling. Later, we moved to a Conda setup; the final reproduction path is in section **E**.

```
sudo apt update
sudo apt install python3-h5py -y
sudo apt-get install gdal-bin python3-gdal
python3 -c "import h5py; print('h5py works')"
```

List HDF metadata / subdatasets:

```
gdalinfo MOD08_D3.A2025265.061.2025266143215.hdf
```

List subdatasets in Python via GDAL (here-doc example as shared):

```
python3 - <<'EOF'
from osgeo import gdal
filename = "MOD08_D3.A2025265.061.2025266143215.hdf"
ds = gdal.Open(filename)
print("Driver:", ds.GetDriver().LongName)
print("Subdatasets:")
```

```
for s in ds.GetSubDatasets():
   print(s[0])
EOF
```

Extract one subdataset to GeoTIFF:

```
gdal_translate "HDF4_EOS:EOS_GRID:\"MOD08_D3.A2025265.061.2025266143215.hdf
\":mod08:Retrieved_Temperature_Profile_Mean" Retrieved_Temperature_Profile_Mean.tif
```

Optional plotting script (as shared) and non-GUI execution:

```
MPLBACKEND=Agg python3 plot_modis.py
explorer.exe .
```

C) HDF reading with helper scripts (from screenshots)

Attempts & fixes (kept for completeness):

```
python3 read_hdf.py              # initial tries from various folders
cp read_hdf.py /mnt/c/NASA_Project/MODIS_data/
cd /mnt/c/NASA_Project/MODIS_data
python3 read_hdf.py MOD08_D3.A2025265.061.2025266143215.hdf
python3 read_hdf.py MOD08_D3.A2025265.061.2025266143215.hdf Solar_Zenith_Mean
```

File ops around HDF placement:

```
mv *.hdf backend/                        # attempt
find . -name "*.hdf"
cp /mnt/c/Users/sakina_fatima/Downloads/*.hdf /mnt/c/NASA_Project/backend/   # attempt
cp /mnt/c/NASA_Project/MODIS_data/*.hdf /mnt/c/NASA_Project/backend/
```

D) Backend app creation & runs (FastAPI/Uvicorn)

Create backend folder (earlier step shown):

```
cd /mnt/c/NASA_Project
mkdir backend
cd backend
```

Install API stack (Conda route used later):

```
# (see Section E for the consolidated conda installs)
```

Create app file and run server:

```
nano main.py
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

If the port is busy:

```
uvicorn main:app --reload --port 8001
```

Keep server alive with tmux (shown in screenshots):

```
sudo apt-get update && sudo apt-get install -y tmux
tmux new -s nasa
# (start uvicorn inside tmux; detach with Ctrl+B, then D)
```

E) Final working environment (Conda-based)

This reflects exactly what we executed in the last steps, including ToS acceptance.

1) (When prompted) accept Anaconda TOS for channels

```
conda tos accept --override-channels --channel https://repo.anaconda.com/pkgs/main
conda tos accept --override-channels --channel https://repo.anaconda.com/pkgs/r
```

2) Create and activate the env

```
conda create -n nasa_env python=3.11 -y
conda activate nasa_env
```

3) Install core stack from conda-forge

```
conda install -c conda-forge gdal=3.11 h5py fastapi uvicorn numpy pandas matplotlib -y
```

4) Verify imports (plain ASCII to avoid bash history expansion)

```
python -c "import fastapi, uvicorn, h5py, numpy, pandas, matplotlib; print('imports OK')"
```

5) Start the API

```
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

6) Enable GDAL HDF4 plugin (the fix that made reads return 200 OK)

```
gdalinfo --formats | grep HDF           # check drivers
conda install -c conda-forge gdal libgdal hdf4 -y
gdalinfo --formats | grep HDF           # verify HDF4/HDF5 now listed
```

7) Re-run the API (now reads HDF4 successfully)

```
uvicorn main:app --reload
```

Example successful requests seen in logs:

```
GET /docs HTTP/1.1 200 OK
GET /earth/files HTTP/1.1 200 OK
GET /earth/datasets/MOD08_D3.A2025265.061.2025266143215.hdf HTTP/1.1 200 OK
GET /earth/data/MOD08_D3.A2025265.061.2025266143215.hdf?variable=Solar_Zenith_Mean HTTP/1.1 200 OK
```

F) Notes captured

- Address-in-use errors occurred when a previous uvicorn was still running; resolved by stopping it or using --port 8001.
- Missing HDF4 support produced ERROR 4 until conda install -c conda-forge gdal libgdal hdf4 was executed.
- tmux used to keep the server process running persistently.

G) Minimal Reproduction (Final)

```
# 1) Data
cd /mnt/c/NASA_Project && mkdir -p MODIS_data && cd MODIS_data
wget --load-cookies ~/.urs_cookies --save-cookies ~/.urs_cookies --keep-session-cookies -c \
"https://d13j1jds5ybppo.cloudfront.net/s3-8d80a21348b33e9c2ee6e0a8ac41d7b1/prod-lads.s3.us-west-2.amazonaws.com/MOD08_D3/MOD08_D3.A2025265.061.2025266143215.hdf?A-userid=astrodata_explorer&Expires=1758728865&Signature=YfYQD0JoAZbDzoEr14GFPSENaAWT1k~aWJ9XO0DHfOcdidK~cxRxC3VkKmpRDDmli0EyExscaTiT4i6debdo8t3-LJUmCoeGZY9mMePYrbakH92ckdTIUNrlRW3OugbW~0sXYL3oqmN93a3qCOrgabnvhg-A9iUhkPEiOlXXtiWY2oxGC1IsSfOIO4RuPM7v-mF1TbAgmCEjJ8bM0deqQI69qCjp6QT3Y8m3wUzr19OxPiD3sAWqCBzL-miyIFQduOLINv5DsAKbOoCKcdub~LFqPeUvWkQrUvUwllTuoErTh2Mz1FcZt-4ojJczjNEOok6PqyHogNXGDAbB4P7CJw__&Key-Pair-Id=K38HK9OA1FVIPQ"
mv MOD08_D3.A2025265.061.2025266143215.hdf* MOD08_D3.A2025265.061.2025266143215.hdf
# 2) Backend env (conda)
conda tos accept --override-channels --channel https://repo.anaconda.com/pkgs/main
conda tos accept --override-channels --channel https://repo.anaconda.com/pkgs/r
conda create -n nasa_env python=3.11 -y
conda activate nasa_env
conda install -c conda-forge gdal=3.11 h5py fastapi uvicorn numpy pandas matplotlib -y
conda install -c conda-forge gdal libgdal hdf4 -y
# 3) Run API
cd /mnt/c/NASA_Project/backend
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

End of document.