Conda Environment Setup: HydroGPU Project

This document outlines the steps to create and populate a Conda environment with essential libraries for geospatial data processing, including rasterio, geopandas, and gdal, alongside fundamental data science libraries like numpy and pandas, and the earthengine-api. The setup prioritizes the rapidsai and conda-forge channels for optimized performance and broader package availability.

1. Prerequisites

Before proceeding, ensure you have:

- **Conda (Miniconda or Anaconda) installed** on your system. If not, download and install it from the official Conda website.
- **NVIDIA GPU (Optional but Recommended for RAPIDS):** If you plan to leverage RAPIDS' GPU acceleration, ensure you have a compatible NVIDIA GPU and the appropriate NVIDIA drivers installed for your cudatoolkit version (here, 11.8).

2. Create the Base Environment

This command creates a new Conda environment named my_rapids_env and pre-configures it with the specified channels and a base Python version, along with the CUDA toolkit required for RAPIDS.

- **Command:**
  ```
  conda create -n my_rapids_env -c rapidsai -c conda-forge -c defaults
  python=3.10 cudatoolkit=11.8
  ```
- **Explanation:**
  - conda create: The command to create a new Conda environment.
  - -n my_rapids_env: Specifies the name of the new environment as my_rapids_env. You can choose any name you prefer.
  - -c rapidsai: Adds the rapidsai channel. This is crucial for installing RAPIDS libraries.
  - -c conda-forge: Adds the conda-forge channel. This is a community-driven channel providing a vast collection of open-source packages, often with more up-to-date versions and better cross-platform compatibility than defaults. It's highly recommended for scientific computing.
  - -c defaults: Includes the default Conda channel. It's listed last to ensure rapidsai and conda-forge packages are prioritized.
  - python=3.10: Installs Python version 3.10 into this environment.
  - cudatoolkit=11.8: Installs the CUDA toolkit version 11.8. This is essential for

RAPIDS to function and should match your NVIDIA driver capabilities. **Adjust this version if your GPU or drivers require a different CUDA toolkit.**

3. Activate the Environment

Before installing additional packages, you must activate the newly created environment.

- **Command:**
  ```
  conda activate my_rapids_env
  ```
- **Explanation:** This command switches your terminal's active environment to my_rapids_env, meaning all subsequent conda install commands will apply to this specific environment.

4. Install Required Libraries

After activating the environment, you can install the specific libraries needed for your projects. Conda will try to resolve dependencies using the channels already configured for the environment (i.e., rapidsai, conda-forge, defaults).

- **Commands:**
  ```
  conda install rasterio
  conda install matplotlib
  conda install numpy
  conda install pandas
  conda install geopandas
  conda install gdal
  conda install earthengine-api
  ```
- **Explanation:**
  - rasterio: A powerful library for reading and writing geospatial raster data.
  - matplotlib: A comprehensive library for creating static, animated, and interactive visualizations in Python.
  - numpy: The fundamental package for numerical computing with Python, providing support for arrays and matrices.
  - pandas: A fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool.
  - geopandas: Extends pandas to allow spatial operations on geometric types. It combines the capabilities of pandas and shapely for geospatial data.
  - gdal: A translator library for raster and vector geospatial data formats. geopandas and rasterio often depend on this.

    - earthengine-api: The official Python client library for Google Earth Engine.

5. Verify Installation (Optional)

You can verify that the packages are installed correctly by trying to import them in a Python interpreter within your activated environment.

- **Commands:**
  ```
  python
  ```
  Then, inside the Python interpreter:
  ```
  import rasterio
  import matplotlib
  import numpy
  import pandas
  import geopandas
  import osgeo.gdal # gdal is imported via its osgeo module
  import ee # earthengine-api's main module
  print("All packages imported successfully!")
  exit()
  ```
  If no errors occur during import, the packages are correctly installed.