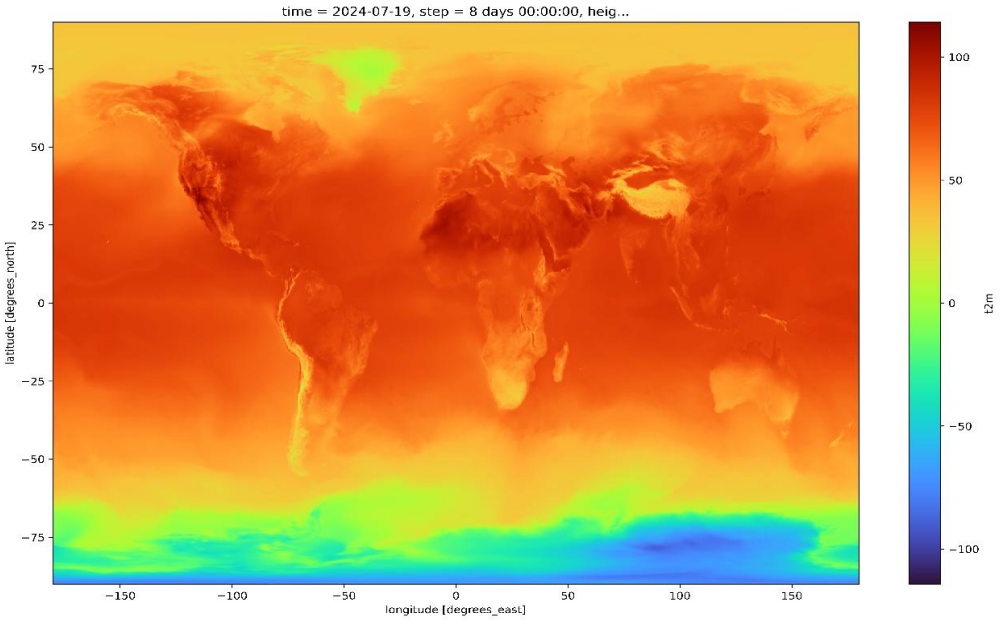
# Efficient Python Workflows for Accessing, Analyzing, Visualizing, and Indexing Large Environmental Datasets in the Cloud

## Instructors

**Ryan Paul Lafler**
*Founder, Principal Systems Architect, and Lead Consultant*
*Premier Analytics Consulting, LLC*
**Email:** rplafler@premier-analytics.com
**LinkedIn:** www.LinkedIn.com/in/RyanPaulLafler
**Business Website:** www.Premier-Analytics.com

**Miguel Angel Bravo**
*Consultant and Data Scientist*
*Premier Analytics Consulting, LLC*
**Email:** mabravo@premier-analytics.com




## Workshop Description

Join us for an immersive, hybrid, full-day Python workshop designed to equip attendees with applied skills in accessing, querying, processing, visualizing, and indexing large-scale environmental datasets stored in public cloud object storage systems. This workshop focuses on real-world climatological and meteorological use cases, featuring datasets from the Coupled Model Intercomparison Project Phase 6 (CMIP6) and NOAA's Real-Time Mesoscale Analysis (RTMA), with access via Google Cloud Storage (GCS) and Amazon S3.

Participants will learn how to efficiently build scalable Python pipelines for exploring, filtering, and visualizing high-resolution climate datasets across time and space. Hands-on examples will demonstrate how to extract time series for point locations and areal shapes, visualize multi-dimensional model output, and apply optimization strategies for handling *big* cloud-hosted environmental data.

New in this 2026 edition: we introduce powerful cloud-native indexing techniques using Kerchunk, enabling virtual Zarr access to GRIB2, NetCDF, and HDF5 files without full conversion. Attendees will gain practical experience

managing rate limiting, leveraging metadata-aware access patterns, and scaling workflows using tools such as Xarray, fsspec, Dask, and Kerchunk.

By the end of this Python workshop, attendees will be able to:

- Comfortably navigate and explore large CMIP6-like environmental datasets stored in public cloud repositories using Python
- Identify and differentiate between spatiotemporal data structures, file formats, and relevant Python libraries
- Build robust Python data pipelines (using functions and classes) that connect to cloud object storage systems (e.g., GCS, Amazon S3)
- Analyze ECMWF historical climate simulations and NOAA-RTMA high-resolution operational meteorological data
- Access, slice, filter, and query multi-dimensional climate data using Xarray
- Generate time series from multidimensional arrays for both points and polygonal regions
- Create virtual Zarr stores from GRIB2 and NetCDF files using Kerchunk for fast, cloud-native access
- Index large environmental datasets stored in the cloud to optimize performance and reduce access time
- Manage and process out-of-memory datasets using Dask and metadata-aware strategies
- Produce stunning visualizations of big climate and meteorological data all within Python

## Workshop Target Audience

This example-oriented Python workshop is designed for industry professionals, researchers, atmospheric scientists, environmental scientists, GIS analysts, machine learning engineers, educators, and students interested in leveraging Python to access, analyze, and index large-scale environmental datasets in the cloud.

Attendees will learn how to efficiently work with climatological, meteorological, and multi-dimensional geospatial arrays stored in public cloud repositories (e.g., S3, GCS). Emphasis is placed on real-world workflows across scientific, research, and business applications involving big data, cloud-native formats, and modern Python GIS libraries.

## Workshop Agenda

This full-day Python workshop will consist of the following topics:

| Proposed Time (Eastern) | Topic # | Topic Title |
|---|---|---|
| 8:00 AM – 8:30 AM (30 min) | **1** | Workshop Kickoff: Objectives, Python Environment Setup, and Core Libraries Overview |
| 8:30 AM – 9:00 AM (30 min) | **2** | Understanding Spatiotemporal Data: Structures, File Formats, and Encoding Schemes |
| 9:00 AM – 9:30 AM (30 min) | **3** | Accessing Big Environmental Data in the Cloud: Python Libraries and Key Concepts |
| 9:30 AM – 10:00 AM (30 min) | | **Coffee Break, Questions, and Breakout Discussions** |
| 10:00 AM – 11:00 AM (60 min) | **4** | Accessing, Exploring, and Visualizing ECMWF CMIP6 Climate Simulations in Google Cloud Storage (GCS) with Python Pipelines |
| 11:00 AM – 12:30 PM (90 min) | **5** | Generating Time Series for Points and Polygons; Visualizing Multi-Dimensional CMIP6 Temperature Arrays |

| | | |
|---|---|---|
| 12:30 PM – 1:30 PM (60 min) | | **Lunch Break, Questions, and Breakout Discussions** |
| 1:30 PM – 2:20 PM (50 min) | **6** | Building Resilient Python Pipelines for Public Cloud Buckets: Rate-Limiting, Metadata, and Caching Strategies |
| 2:20 PM – 3:30 PM (70 min) | **7** | Working with NOAA-RTMA Hourly GRIB Data in Amazon S3: Efficient Python Workflows for Operational Meteorological Data Files |
| 3:30 PM – 3:45 PM (15 min) | | **Coffee Break, Questions, and Breakout Discussions** |
| 3:45 PM – 4:45 PM (60 min) | **8** | Indexing NetCDF, GRIB2, and HDF5 in the Cloud with Kerchunk: A Metadata-Driven Workflow using NOAA-RTMA GRIB2 Files |
| 4:45 PM – 5:00 PM (15 min) | | **Final Recap, Open Q&A, and Discussion** |

## Detailed Agenda

**2) Understanding Spatiotemporal Data: Structures, File Formats, and Encoding Schemes**

- Define key characteristics of spatiotemporal data: spatial resolution, temporal resolution, coordinate reference systems, and map projections
- Distinguish between gridded (raster) and vector-based environmental datasets
    - Examples of gridded arrays and points, lines, polygons, and geometries
    - Examples of gridded file formats: NetCDF, GRIB, GRIB2, Zarr, DEM, COG, GeoTIFF, PNG, JPEG
    - Examples of vector file formats: CSV, Apache Parquet, GeoJSON, ESRI Shapefile, Geodatabase
    - Discuss tradeoffs in format selection: compression, I/O speed, indexing capabilities, scalability
- Introduce virtual Zarr stores as a format-agnostic access method for large, gridded files in cloud environments

**3) Accessing Big Environmental Data in the Cloud: Python Libraries and Key Concepts**

- Understand Python programming constructs for building reusable, modular cloud workflows
- Introduce key Python libraries for big data and cloud-native analysis:
    - **Xarray** for labeled multi-dimensional arrays
    - **Dask** for parallel and out-of-core computation of big data
    - **fsspec** and **s3fs**/**gcsfs** for accessing cloud storage (S3, GCS)
    - **Kerchunk** for efficient indexing of separately stored gridded data files
    - **Pandas**, **NumPy**, and extensions (**GeoPandas**) for scalable tabular, array, vector processing
- Review visualization libraries: Matplotlib, Seaborn, Plotly
- Discuss “cloud-optimized” patterns: lazy loading, caching, chunking, metadata-driven reads

**4) Accessing, Exploring, and Visualizing ECMWF CMIP6 Climate Simulations in Google Cloud Storage (GCS) with Python Pipelines**

- Explore structure and layout of CMIP6 datasets in Google Cloud Storage buckets
- Navigate and interpret folder structure containing climate simulations within CMIP6 GCS
- Engineer Python pipelines that access and deliver specified data from GCS into the Python session
- Incorporate Python libraries to efficiently handle and process multidimensional environmental data
- Implement chunk-aware reads and lazy evaluation for high-volume CMIP6 data
- Develop, test, and deploy pipelines to query different climate attributes from ECMWF historical simulated datasets
- Filter, load, and explore variables across model ensembles and scenarios
- Visualize high-resolution ECMWF historical simulations inside of Python

**5) Generating Time Series for Points and Polygons; Visualizing Multi-Dimensional CMIP6 Temperature Arrays**

- Introduce air temperature (multi-dimensional) gridded arrays and how to filter across many dimensions
- Filter and aggregate ECMWF records over temporal and spatial dimensions
- Subset ECMWF CMIP6 data spatially (by point or polygon) and temporally (by timestep or average)
- Extract locations and generate comparative time series
- Calculate zonal statistics over user-defined regions and examine comparative trends
    - Examine effects of air temperature during a heat wave over the central US across different pressure levels
    - Visualize, customize, and export comparative time series plots with **Matplotlib**

**6) Building Resilient Python Pipelines for Public Cloud Buckets: Rate-Limiting, Metadata, and Caching Strategies**

- Learn common challenges when accessing environmental data from public cloud object storage providers (e.g., Amazon S3, GCS)
- Handle rate-limiting issues from repeated reads of public datasets (e.g., CMIP6 on GCS)
- Explore the use of **fsspec.CachingFileSystem**, **simplecache**, or **filecache** layers to locally cache remote cloud reads
- Apply retry logic and backoff strategies using **fsspec**, **aiohttp**, or **requests**
- Enable metadata-aware reads using Zarr consolidated metadata or Kerchunk-generated JSON index files
- Use caching mechanisms to avoid repeated downloads of metadata or file headers
- Organize access patterns and chunking strategies to minimize bandwidth and request costs

**7) Working with NOAA-RTMA Hourly GRIB Data in Amazon S3: Efficient Python Workflows for Operational Meteorological Data Files**

- Understand the NOAA-RTMA GRIB2 dataset layout, naming conventions, and temporal granularity (hourly)
    - Investigate how variables are stored and can be retrieved
- Develop pipelines to access hourly GRIB2 files directly from Amazon S3 using fsspec and s3fs (public bucket access)
    - Ensure pipelines are customizable and can access different hourly meteorological data in S3 bucket (using its folder structure)
- Download GRIB2 files in chunks for quicker access in temporary storage
- Visualize hourly temperature with zonal and meridional wind components to generate detailed maps

**8) Indexing NetCDF, GRIB2, and HDF5 in the Cloud with Kerchunk: A Metadata-Driven Workflow using NOAA-RTMA GRIB2 Files**

- Introduce Kerchunk and its role in cloud-native indexing of monolithic formats (GRIB2, NetCDF, HDF5)
- Explain the concept of reference-based access, where JSON files act as maps to internal byte ranges in cloud-hosted files
- Demonstrate how to scan individual files and generate Kerchunk reference JSONs to emulate Zarr-like access without data conversion
- Check the metadata structure for NOAA-RTMA GRIB2 files and ensure consistency across datasets
- Scan and generate JSON reference files for multiple GRIB2 files from S3
- Combine multiple JSON references using MultiZarrToZarr to enable unified access to time-series collections or ensembles
- Perform on NOAA-RTMA GRIB2 files for better indexing and time series retrieval

## Instructor Bios

**Ryan Paul Lafler** is the Founder and Lead Consultant of **Premier Analytics Consulting, LLC**, a California-certified small business based in San Diego specializing in applied AI and machine learning systems, distributed data engineering, statistical analysis, enterprise GIS, and custom full-stack analytics platform development. As a principal architect and consultant, Ryan designs and delivers infrastructure-aware AI/ML solutions, production-grade analytics platforms, scalable data infrastructure, GIS and spatial analytics systems, and statistical modeling workflows for enterprise organizations, public-sector agencies, and research institutions. Through consulting and contracting roles, he has cross-industry programming expertise in Python, R, SQL/NoSQL, SAS®, and modern JavaScript frameworks, and implements structured quality control, validation, and governance practices for automated analytics and AI-assisted workflows. He also serves as an Adjunct Professor in the Big Data Analytics Graduate Program, the Department of Mathematics and Statistics, and the Global Campus Program at San Diego State University. He earned his Master of Science in Big Data Analytics (2023) following the defense and publication of his thesis, and his Bachelor of Science in Statistics with a Minor in Quantitative Economics (2020), both from San Diego State University.

**Miguel Angel Bravo** is a Consultant and Data Scientist for Premier Analytics Consulting, LLC, where he develops applied machine learning systems, AI integrations, big data pipelines, and data-driven full-stack systems for research and enterprise analytics. His work spans production-ready ML workflows, containerized AI systems, open-source GIS workflows, and real-time analytics using Python, FastAPI, Docker, AWS, and modern MLOps practices. Miguel holds a Master of Science in Big Data Analytics from San Diego State University and a Bachelor of Science in Electronics, Robotics, and Mechatronics Engineering from the University of Málaga, with research experience in environmental modeling, geospatial analytics, and AI-driven decision support systems.

## About Premier Analytics Consulting, LLC

Premier Analytics Consulting, LLC is a California Certified Small Business founded and led by Ryan Paul Lafler that provides cross-industry services including infrastructure-aware AI and machine learning systems, distributed data engineering, statistical analysis and modeling, enterprise GIS solutions, and custom full-stack platform development for organizations working with complex data environments. The firm partners with enterprise organizations, public-sector agencies, consulting firms, and research institutions through prime contracts, subcontracts, consulting and advisory engagements, and technical partnerships in flexible remote and hybrid environments. Premier Analytics Consulting focuses on designing and implementing reliable, secure, and production-ready AI, analytical, statistical, data engineering, and GIS systems that support open-source modernization and enterprise decision-making, research, operations, and long-term organizational data strategy across industries and technical domains.

➤ *Learn more about our services and engagement structures:* **www.Premier-Analytics.com**

**UEI:** P2HZGKY3FXN3 | **CAGE:** 15Q83
**D-U-N-S:** 13-156-8659
**California DGS Small (Micro) Business:** Certification No. 2049971
**NAICS Codes:** 541511 • 541512 • 541690 • 541715 • 541519