

# Cloud Data Demo

Rich Signell  
USGS, Woods Hole, MA  
...and the Pangeo  
Community

Coastal Coupling Community of Practice  
2021-02-22

# Open Architectures for Cloud-Native Earth System Analytics

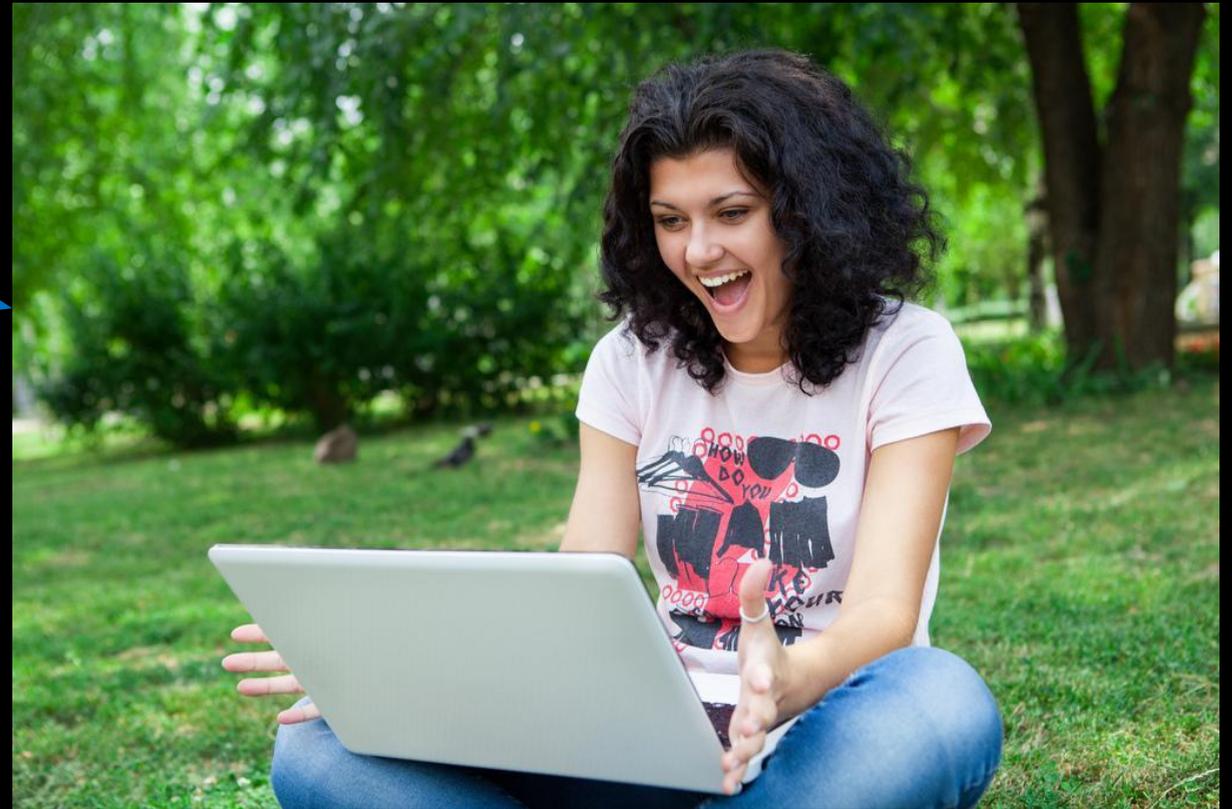
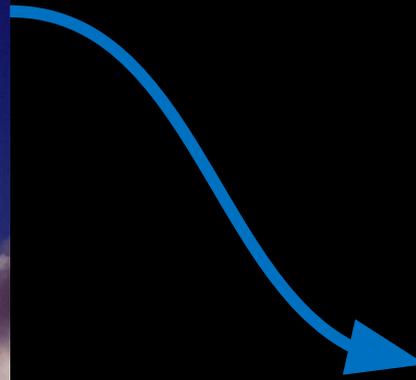
Rich Signell  
USGS, Woods Hole, MA  
...and the Pangeo  
Community

Coastal Coupling Community of Practice  
2021-02-22

# Traditional Model Data Analysis



# Model Data Analysis on the Cloud



# Closed Platforms vs. Open Architectures

**Closed platforms** aim to bring all the data into a central location and provide tools for users to perform analysis on the data.

Examples: Google Earth Engine  
Descartes Labs Platform,  
Copernicus, DesignSafe-CI

**Open architectures** assume data will be distributed and seek interoperability between different data catalogs and computational tools.

Examples: Pangeo, Open Data  
Cube

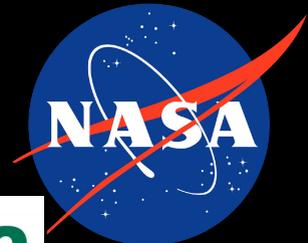
# Pangeo is a Global Community



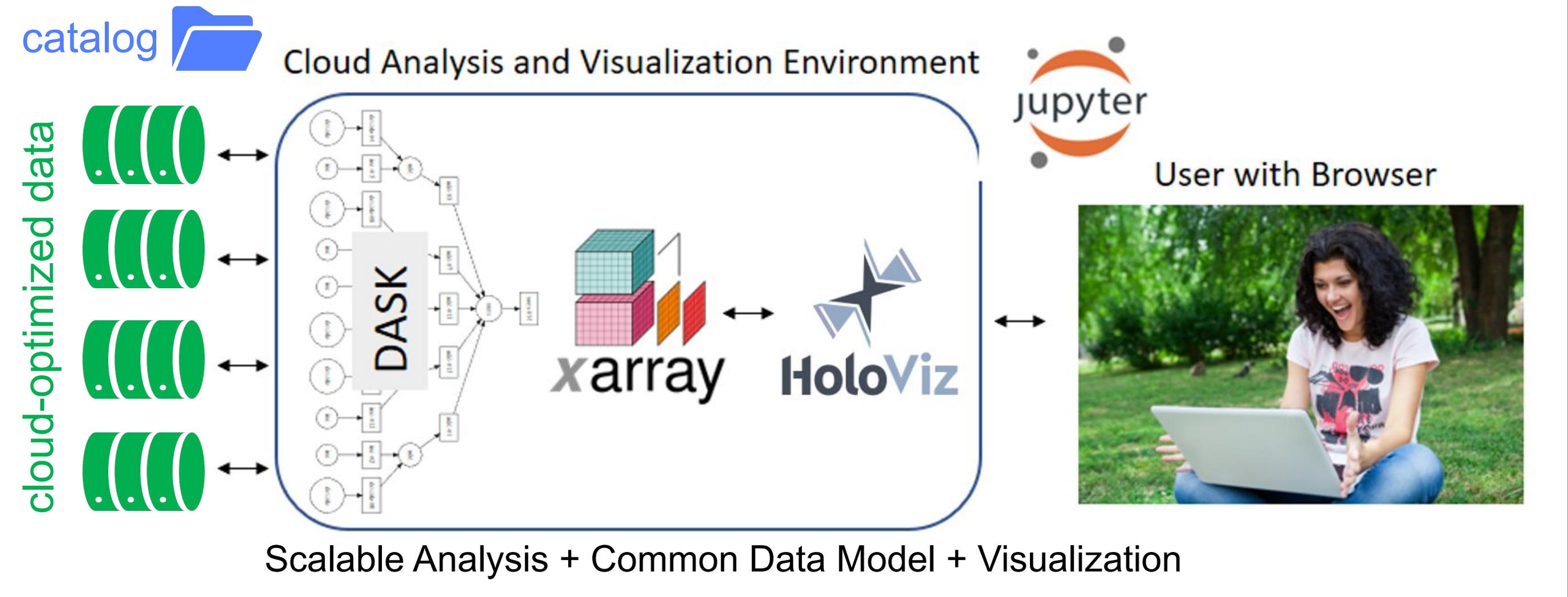
**PANGEO**

A community platform for Big Data geoscience

-  **How fast can the Met Office's solution pull data from S3?** 16  
#198 opened 17 days ago by mrocklin
-  **Pangeo use case: Advanced regridding using ESMF/ESMpy/OGGIS/xESMF/Xarray/Dask** 18  
#197 opened 17 days ago by jhamman



# Pangeo Cloud Architecture



# Pangeo is a Flexible Open-Source Framework



Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

# Cloud-optimized data

To overcome latency of object storage and take advantage of multiple cpus:

- Metadata can be read with one or just a few reads
- Data is written in chunks with compression
- Chunks are big enough so that latency is not a significant fraction of the read time (e.g. ~100mb)
- Chunk shape should be chosen based on expected use cases (e.g. not always 1 in the time dimension)

# Cloud-optimized data formats

- Zarr: ndarray data (e.g. met/ocean/hydro model output, time stacks of remote sensing data)
- Cloud-optimized GeoTIFF (COG): geospatial image data
- Parquet: tabular data
- Entwine Point Tile: point cloud data

Note: other formats like NetCDF4/HDF5, GRIB2 typically don't perform well with native readers, but there is a way...

It's possible  
to use other  
formats  
effectively  
on the  
cloud  
also...

# Cloud-Performant NetCDF4/HDF5 with Zarr, Fsspec, and Intake

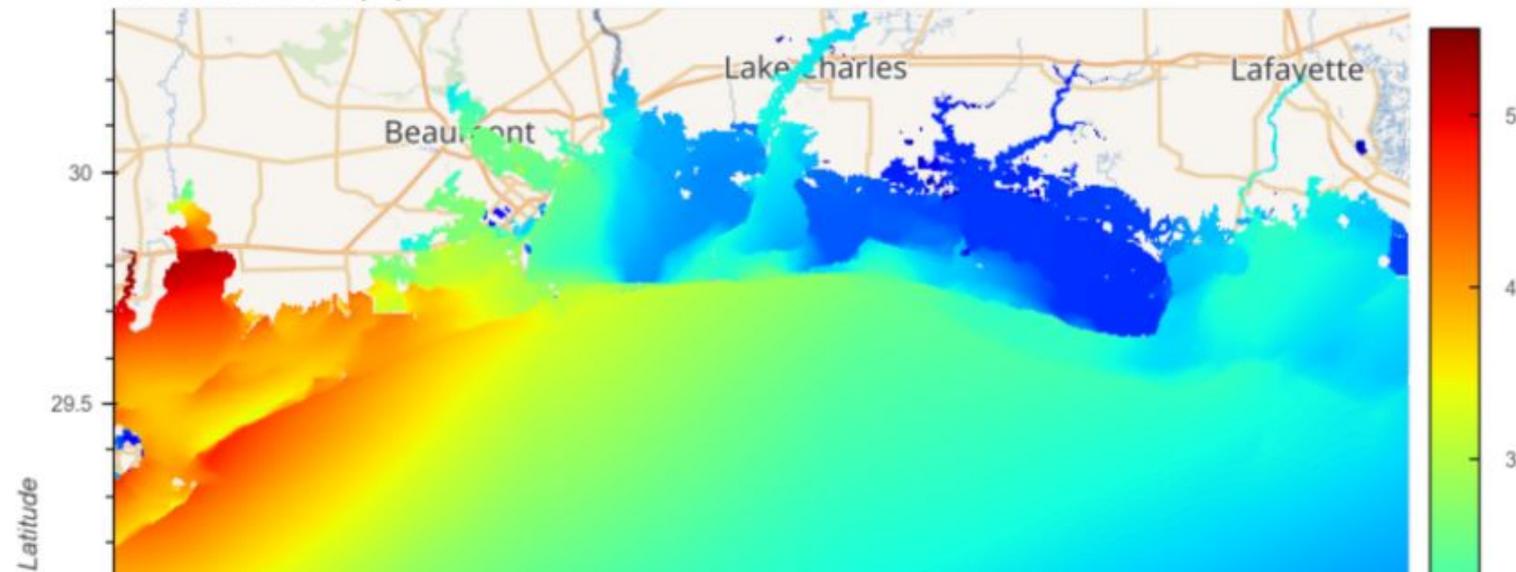


Richard Signell [Follow](#)

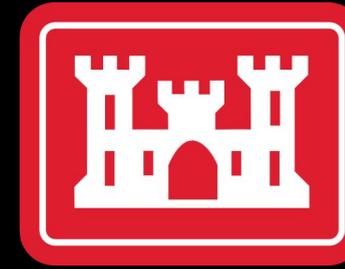
Dec 14, 2020 · 6 min read



max water level (m)



# Cloud Data Pilot Project thanks to ESIP, AWS and Qhub



Quansight.  
YOUR DATA EXPERTS



QUANSIGHT

## QHUB - ESIPFED-QHUB

AUTOSCALING COMPUTE ENVIRONMENT ON AWS

Welcome to [jupyter.qhub.esipfed.org](https://jupyter.qhub.esipfed.org). It is maintained by Quansight staff. The hub's configuration is stored in a github repository based on <https://github.com/Quansight/qhub/>. To provide feedback and report any technical problems, please use the [github issue tracker](#).

[Sign in with GitHub](#)

## Announcing QHub

Updated: Oct 23, 2020

Today, we are announcing the release of QHub, a new open source project from Quansight that enables teams to build and maintain a cost-effective and scalable compute/data science platform in the cloud or on-premises. QHub can be deployed with minimal in-house DevOps experience.

See the demonstration here:

Dharhas Pothina - Introducing QHub | JupyterCon 2020

Watch later Share

```
name: "quansight/qhub-jupyter-lab:4c8c28332be1fde32f786c54a5961b5f3d789e16"
owner_override:
  name: "quansight/qhub-jupyter-lab:4c8c28332be1fde32f786c54a5961b5f3d789e16"
  image: "quansight/qhub-jupyter-lab:4c8c28332be1fde32f786c54a5961b5f3d789e16"
dash_workers:
  "small_worker":
    worker_cores_limit: 1
    worker_cores: 1
    worker_memory_limit: 10
    worker_memory: 10
    image: "quansight/qhub-dash-worker:4c8c28332be1fde32f786c54a5961b5f3d789e16"
  "medium_worker":
    worker_cores_limit: 1.5
    worker_cores: 1.5
    worker_memory_limit: 20
    worker_memory: 20
    image: "quansight/qhub-dash-worker:4c8c28332be1fde32f786c54a5961b5f3d789e16"
environments:
  "environment-default.yaml":
    name: default
    channels:
      - conda-forge
      - defaults
    dependencies:
      - python=3.7
      - ipynb
      - ipynb-glueviz
      - dask=2.14.0
      - distributed=2.14.0
      - dask-gateway=0.6.1
      - numpy
      - numba
      - pandas
      - flask
  "environment-example-2.yaml":
    name: example-2
    dependencies:
      - numpy
      - numba
      - pandas
      - flask
  "environment-example-2.yaml":
    name: example-2
```

Recent commits

3.2k qhub-config.yaml

# Conclusion: Open Cloud Architecture advantages

- No downloading to local machines
- No local compute infrastructure
- Pay only for what you use
- Unmatched compute power
- Builds Community
- Reproducible
- Inclusive