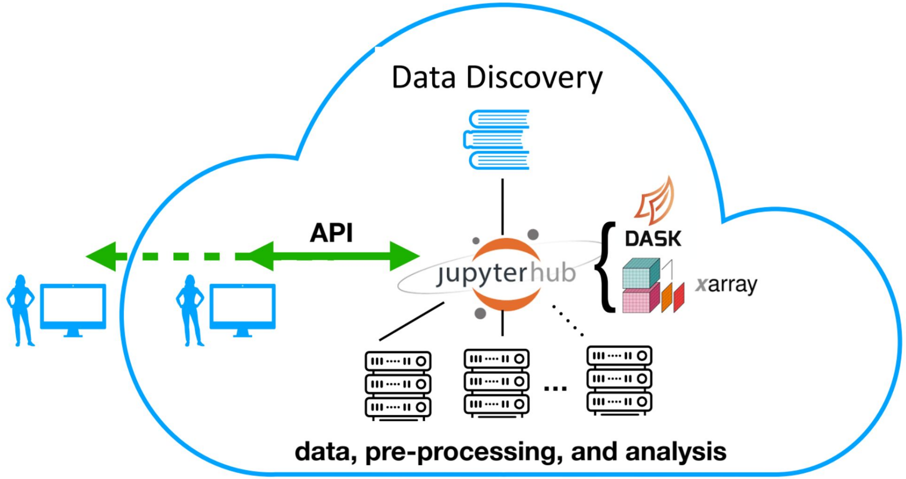
https://bit.ly/eosdis-pangeo

PANGEO



Community Tools for Analysis of NASA Earth Observation System Data in the Cloud

----

EOSDIS Webinar
July 30, 2020
Proposal (project) Number: 17-ACCESS17-0003
Co-Operative Agreement Number(s): 80NSSC18M0157, 80NSSC18M0158, 80NSSC18M0159

PIs: Anthony Arendt (1), Joe Hamman (2), Daniel Pilone (3)
Institutions: (1) University Of Washington, Seattle, (2) University Corporation For Atmospheric Research, (3) Element 84, Inc.

Project Overview

Project Team

Anthony Arendt
PI, University of Washington

Joe Hamman
PI, NCAR/CGD

Tom Augspurger
PI, Anaconda Inc.

Dan Pilone
PI, Element 84

Rob Fatland
Co-I, University of Washington

Scott Henderson
Co-I, University of Washington

Amanda Tan
University of Washington

Sebastian Alvis
UW eScience

Matt Hanson
Element 84

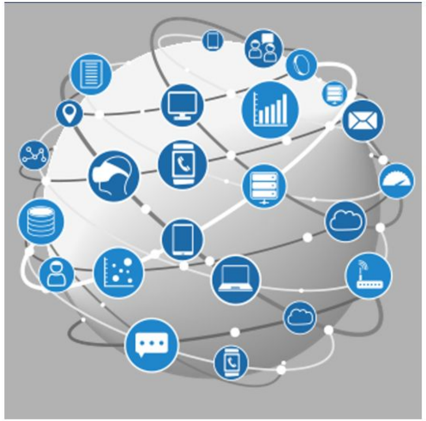
UNIVERSITY of WASHINGTON eScience Institute ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS

NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

ANACONDA.

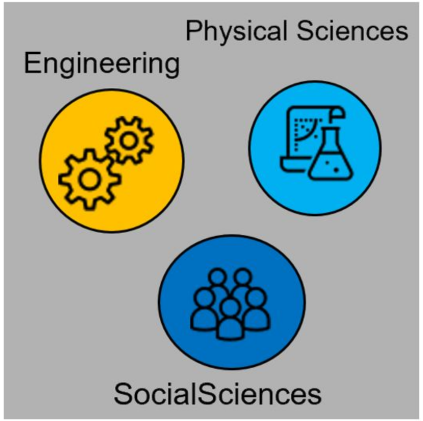
Element 84

Research Science: a New Era of Complexity



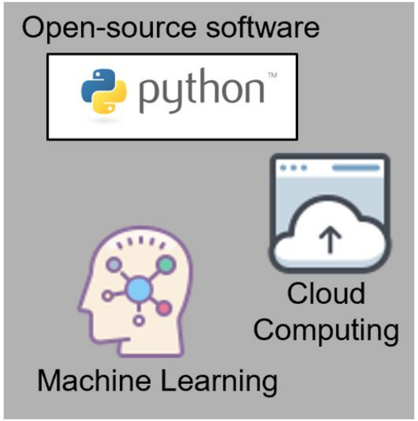
Data Deluge

Sensors, simulations, lab automation, field data



Interdisciplinarity

New insights occur at the intersection between disciplines

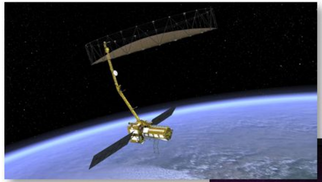


New Tools

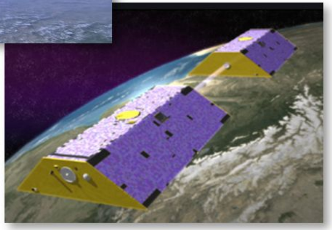
Scientists require depth of knowledge in both data science and domain science

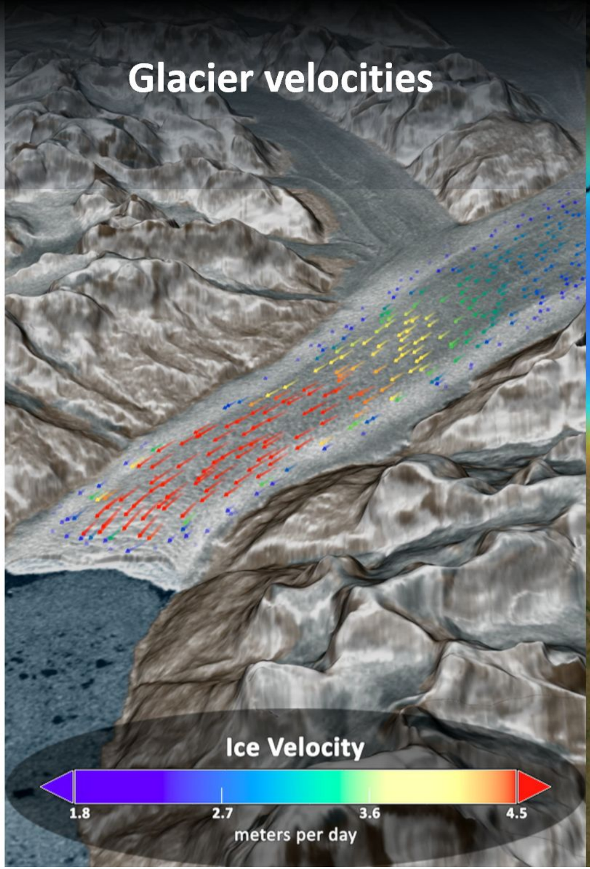
Glaciological studies

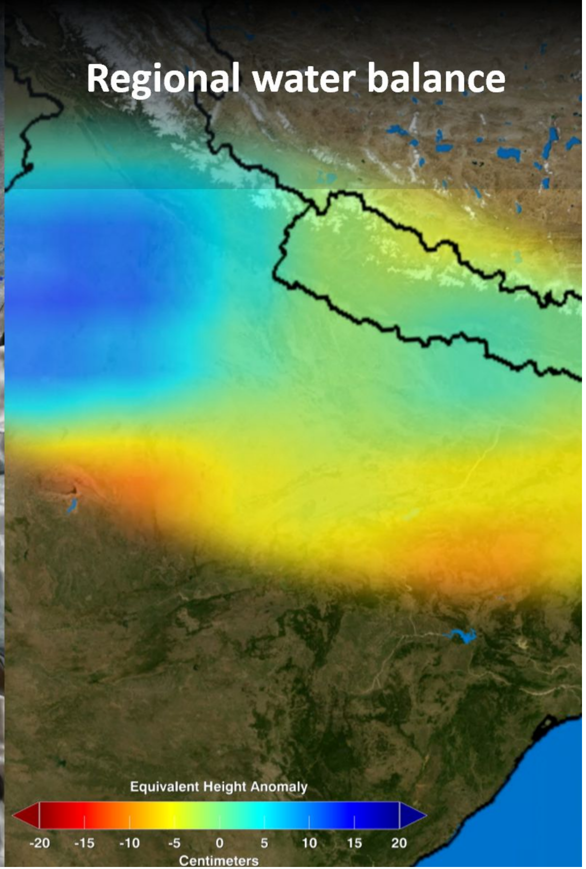
NISAR



GRACE

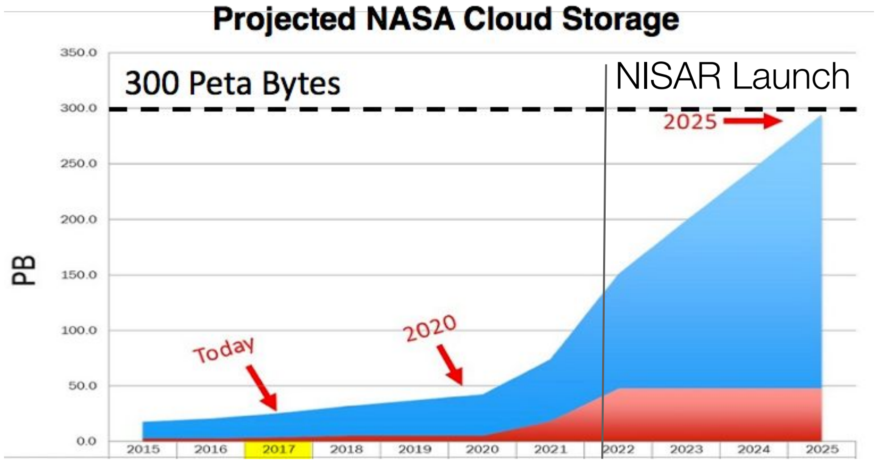


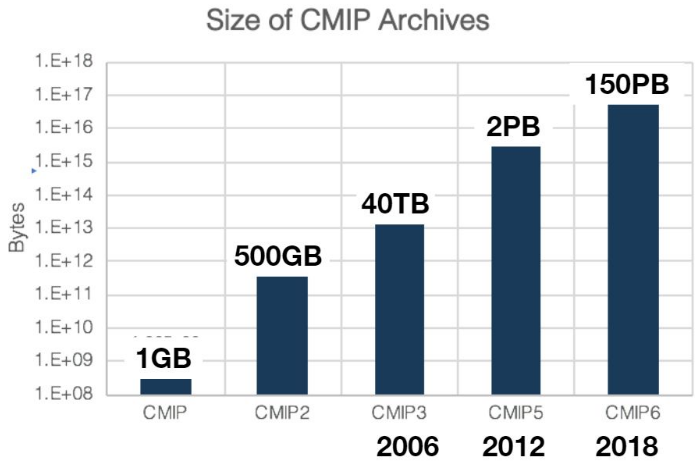




Geospatial community needs

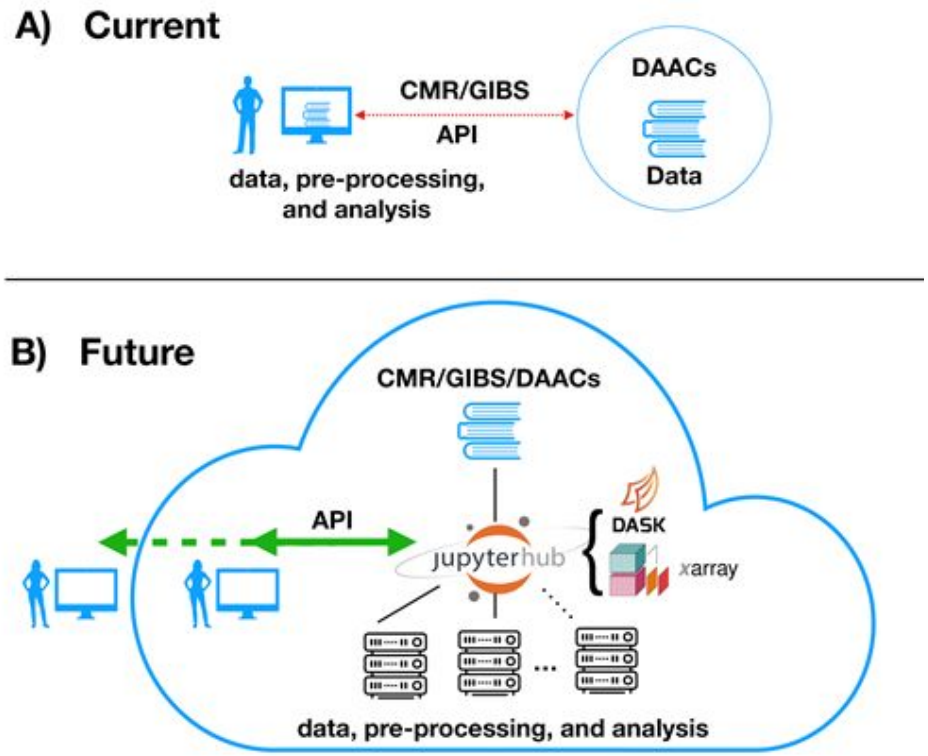
Need better tools for scalable, data proximate computing to support exploration of increasingly large data volumes:





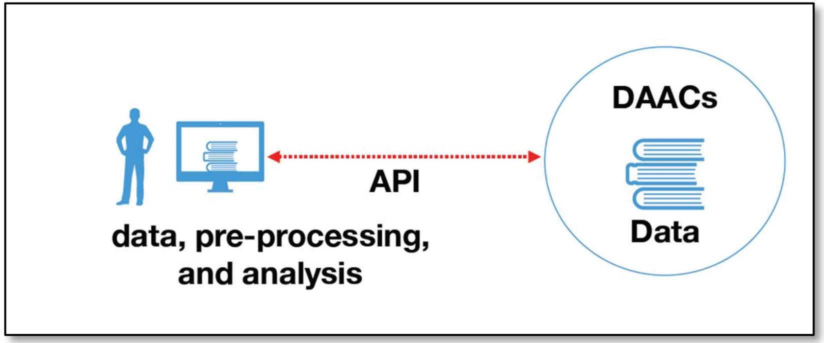
Pangeo Goals

Improved search, discovery and interactive analysis of NASA data. In particular, deployable scalable algorithms rather than downloading data.



Scientific interaction w/ NASA data

existing model



**DAAC = "Distributed Active Archive Center"*

Downloading bottleneck as researcher waits for data transfer

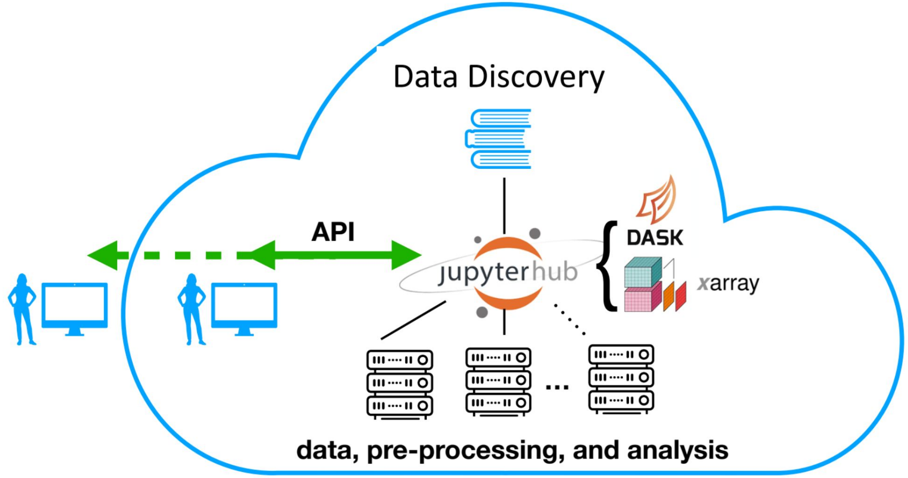
Difficult data management because researchers end up duplicating large subsets of original data with minor modifications

Difficult to share running someone else's code requires downloading all that data again!

Limited computational power since algorithms run on researcher's hardware

Benefits for Cloud-native analysis

Proposed model: Move analysis to the data



*Schematic specific to NASA data moving to AWS Cloud, but same architecture applies for HPC

Instant access to compute resources and data (no queueing)

Democratize access large computations accessed with web browser

No downloading since algorithms uploaded to data

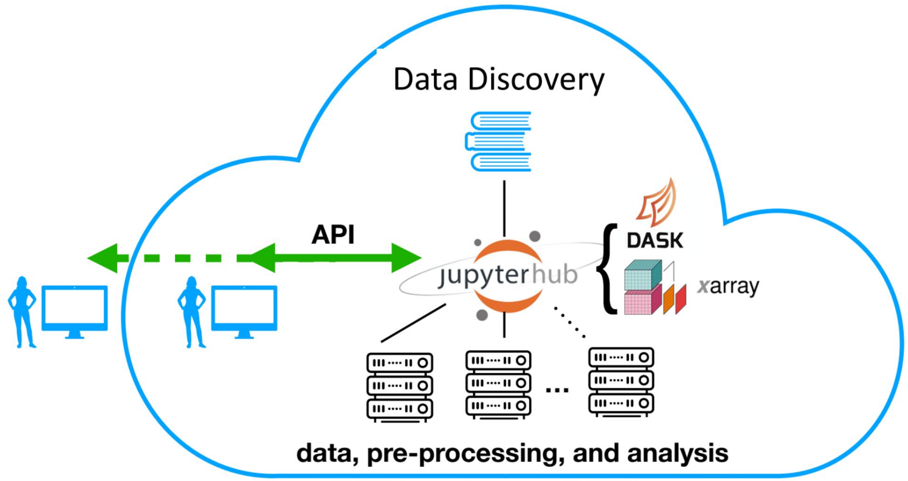
Scalable computational power used and billed by time

On-demand special resources (GPUs)

Reproducible workflows thanks to network-accessible datasets and containerized software

Concerns for Cloud-native analysis

Proposed model: Move analysis to the data



Unfamiliar cost model for cloud resources (utility pricing instead of sunk cost)

Steep learning curve to design and implement Cloud-based infrastructure

Concern over commercial management of public data

Potential vendor lock-in with major Cloud-providers (AWS, GCP, Azure…)

Ultimate Goal: Reallocate time!

Traditional Project Timeline

- 80% Data Preparation (download, clean, & organize files)
- 10% Batch Processing
- 10% Think about science

Cloud-based Project Timeline

- 5% Load AODS
- 5% Parallel Processing
- 90% Think about science

*Slide by Chelle Gentemann (Farallon Institute), ESIP 2020 Summer Meeting Keynote "Empowering Transformational Science"

https://speakerdeck.com/cgentemann/empowering-transformational-science

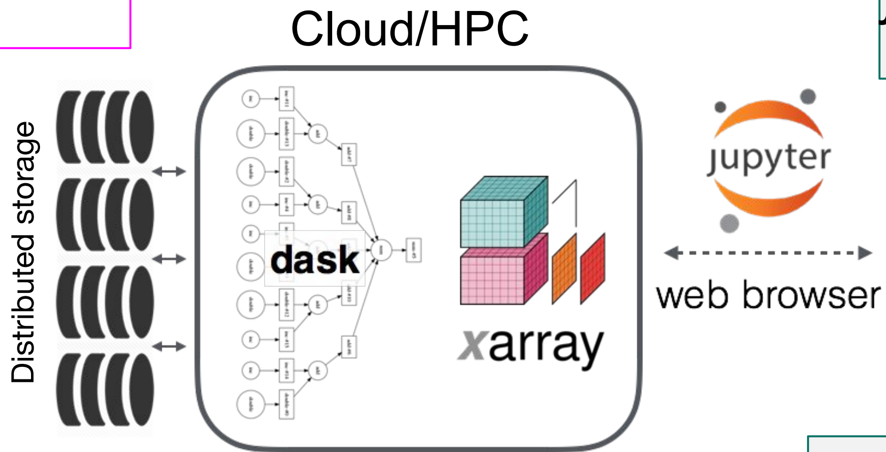
Cloud computing & JupyterHub

The Pangeo Computing Architecture

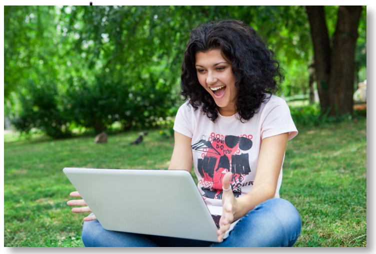
*Formats: ZARR, COG, TileDB

Analysis Ready Data
Stored ***and cataloged*** on globally-available distributed storage (e.g. S3, GCS)

*Catalogs: STAC!



Jupyter for interactive access on remote systems

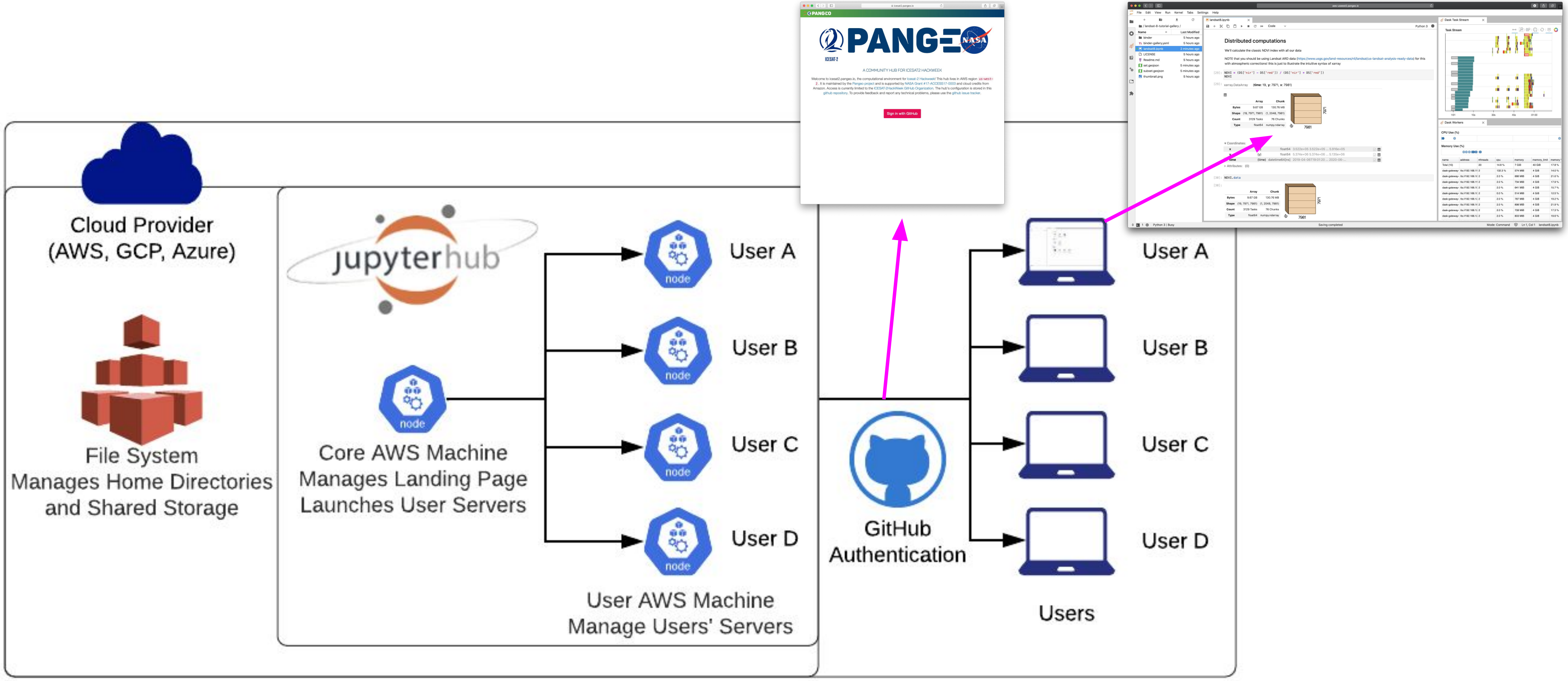


Parallel computing system built on top of ***Kubernetes (dask-gateway) or HPC (dask-jobqueue***.

Dask tells the nodes what to do.

Xarray provides data structures and intuitive interface for interacting with datasets

JupyterHub behind the scenes:



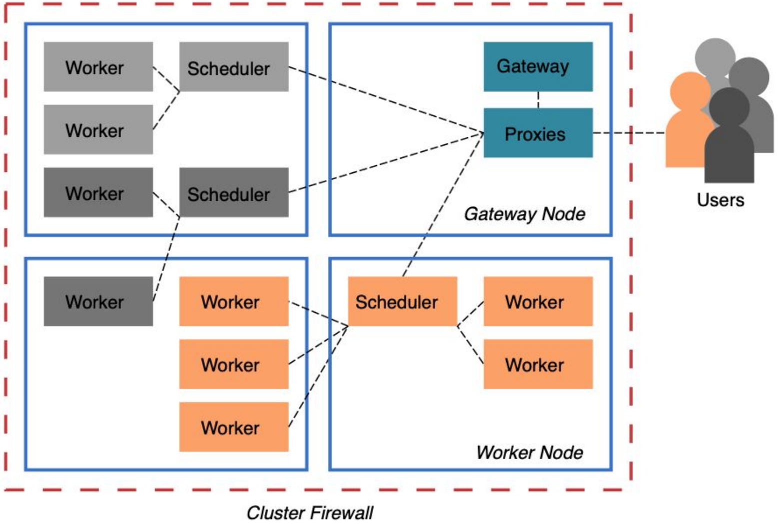
Dask-gateway for scalable computations

Pangeo with Dask Gateway

Tom Augspurger Follow
Mar 31 · 4 min read

https://medium.com/pangeo/pangeo-with-dask-gateway-4b638825f105

- Administrator handles configuration
- Scientific users only need to connect
- Separates Dask clusters from JupyterHub
- Currently implemented for Kubernetes



Dask-gateway from a user's perspective:

Dask-Gateway Cluster

If we don't specify a specific cluster, dask will use the cores on the machine we are running our notebook on instead, lets connect to a Dask-Gateway cluster. You can read more about this cluster at https://gateway.dask.org/ .

[25]:
```
from dask_gateway import GatewayCluster
from dask.distributed import Client

cluster = GatewayCluster()
client = cluster.get_client()
cluster.adapt(minimum=10, maximum=20)
cluster
```

GatewayCluster

Workers	10
Cores	20
Memory	42.95 GB

▸ Manual Scaling

▸ Adaptive Scaling

Name: icesat2-prod.a89934f463124d1cbe4266dc1e133567

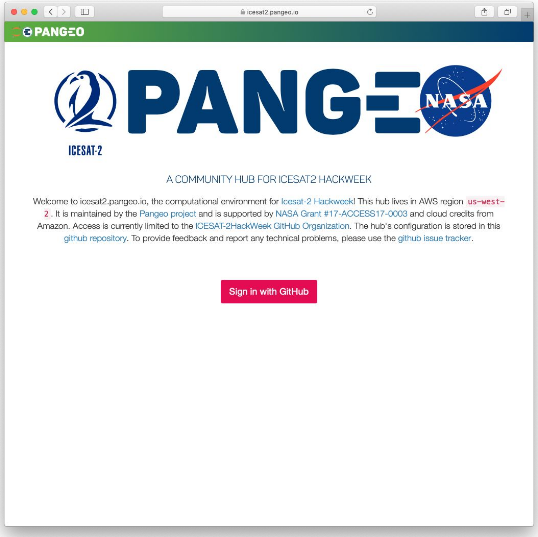
Dashboard: https://aws-uswest2.pangeo.io/services/dask-gateway/clusters/icesat2-prod.a89934f463124d1cbe4266dc1e133567/status

What is Pangeo Cloud?

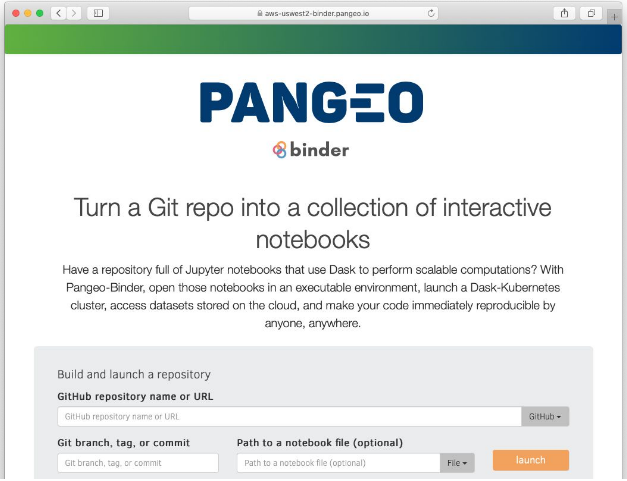
https://pangeo.io/cloud.html

"Pangeo Cloud is an experimental service providing cloud-based data-science environments (JupyterHubs and BinderHubs)."

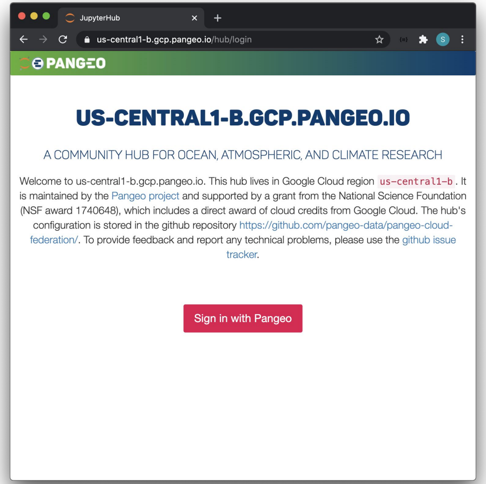
https://aws-uswest2.pangeo.io/



https://binder.pangeo.io/



https://us-central1-b.gcp.pangeo.io/

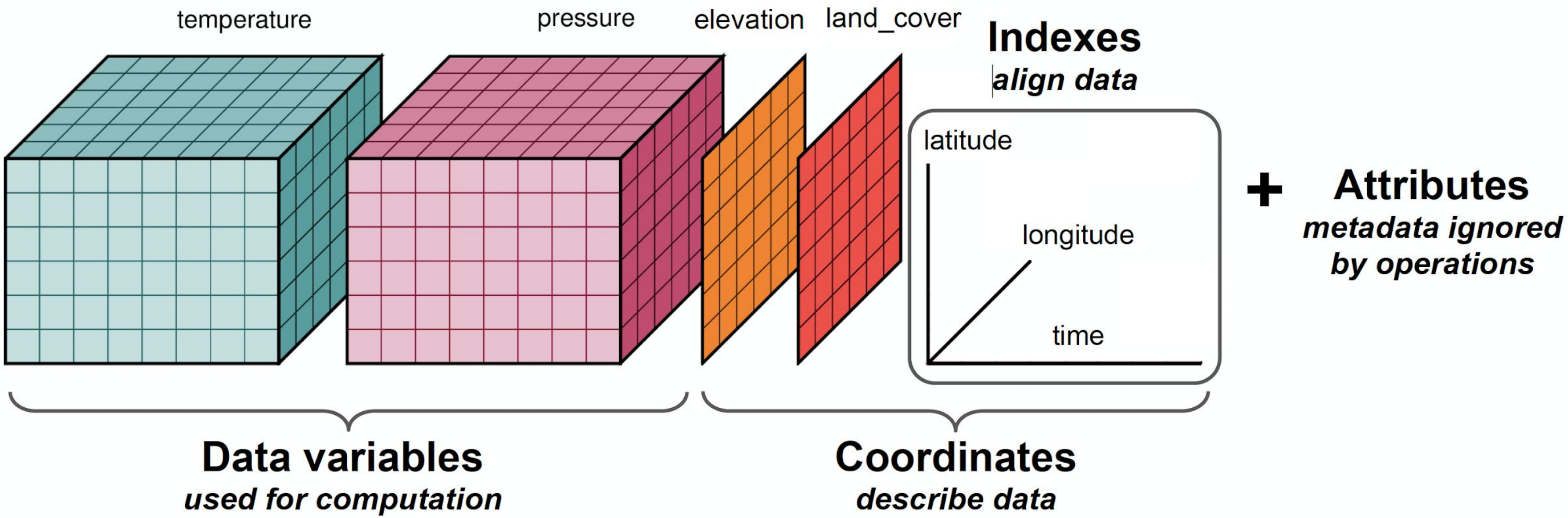


Scientific Python Software

Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

Xarray data model

XARRAY DATASET: MULTIDIMENSIONAL VARIABLES WITH COORDINATES AND METADATA



"netCDF meets pandas.DataFrame"

Credit: Stephan Hoyer

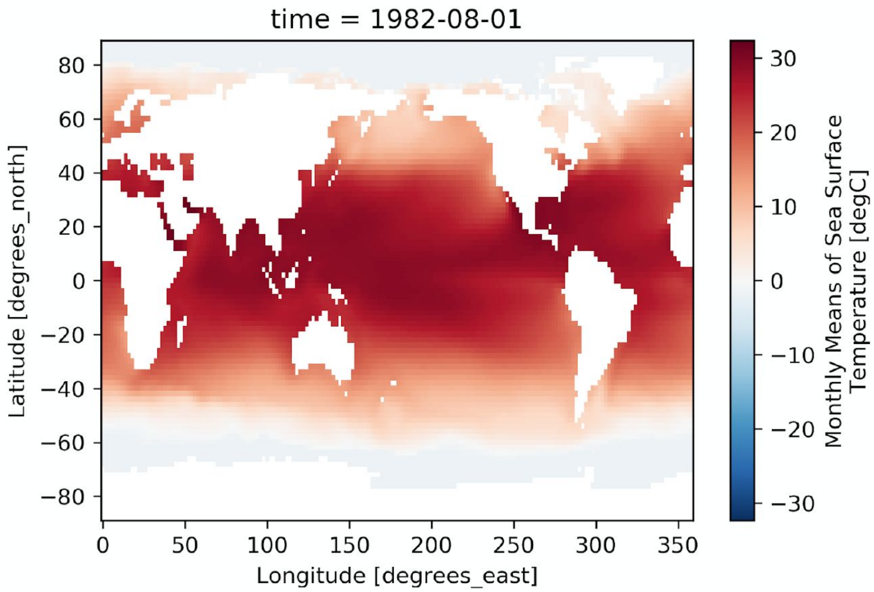
XARRAY MAKES SCIENCE EASY

```python
import xarray as xr
url = 'https://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/'
fname = 'noaa.ersst.v5/sst.mnmean.nc'
ds = xr.open_dataset(url + fname)
ds
```

```
Dimensions:     (lat: 89, lon: 180, nbnds: 2, time: 1974)
Coordinates:
  * lat         (lat) float32 88.0 86.0 84.0 82.0 80.0 78.0 76.0 74.0 72.0 ...
  * lon         (lon) float32 0.0 2.0 4.0 6.0 8.0 10.0 12.0 14.0 16.0 18.0 ...
  * time        (time) datetime64[ns] 1854-01-01 1854-02-01 1854-03-01 ...
Dimensions without coordinates: nbnds
Data variables:
    time_bnds   (time, nbnds) float64 ...
    sst         (time, lat, lon) float32 ...
Attributes:
    climatology:                     Climatology is based on 1971-2000 SST, X...
    description:                     In situ data: ICOADS2.5 before 2007 and ...
```

XARRAY: LABEL-BASED SELECTION

```
# select and plot data from one day
ds['sst'].sel(time='1982-07-31', method='nearest').plot()
```

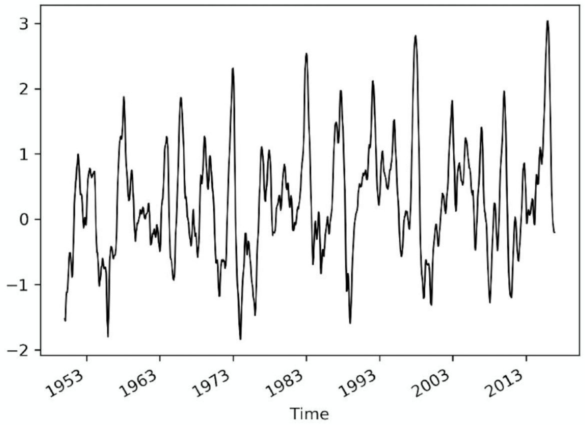


PANGEO

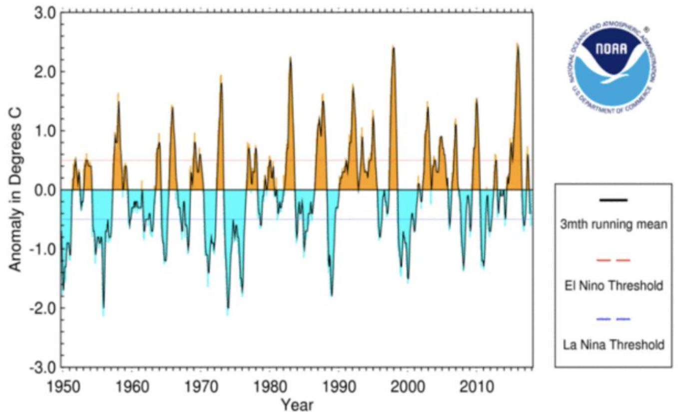
XARRAY: GROUPING AND AGGREGATION

```
sst_clim = ds['sst'].groupby('time.month').mean(dim='time')
sst_anom = ds['sst'].groupby('time.month') - sst_clim
nino34_index = (sst_anom.sel(lat=slice(5, -5), lon=slice(190, 240))
                .mean(dim=('lon', 'lat')).rolling(time=3).mean()
                .sel(time=slice('1950', '2016')))

nino34_index.plot()
```



SST Anomaly in Nino 3.4 Region (5N-5S,120-170W)



Analysis ready data

Analysis Ready Data

Example: Land Surface Model Output

Cloud Optimized GeoTIFF
An imagery format for cloud-native geospatial processing

Zarr

https://zarr.readthedocs.io
https://www.cogeo.org/

Flat File: NetCDF
one file per day

Cloud Optimized Format
("Analysis-ready")
zarr files

Distributed Computation
Dask and Xarray perform faster!

- LIS_HIST_200304010000.d01.nc
- LIS_HIST_200304020000.d01.nc
- LIS_HIST_200304030000.d01.nc
- LIS_HIST_200304040000.d01.nc
- LIS_HIST_200304050000.d01.nc
- LIS_HIST_200304060000.d01.nc
- LIS_HIST_200304070000.d01.nc
- LIS_HIST_200304080000.d01.nc

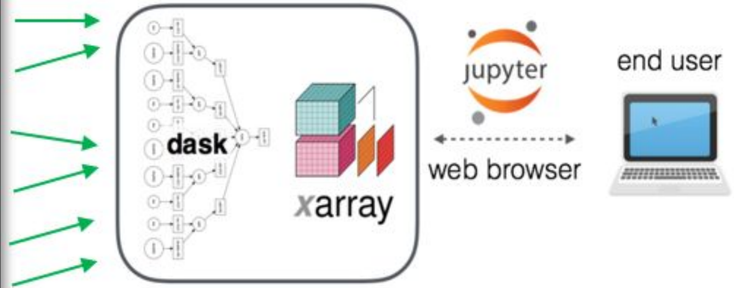
→

- Evap_tavg
- lat
- long
- Qs_tavg
- Qsb_tavg
- Qsm_tavg
- Rainf_tavg
- Snowf_tavg
- time
- TWS_tavg

→

- 0.0.0
- 0.0.1
- 0.0.2
- 0.0.3
- 0.0.4
- 0.0.5
- 0.0.6

→



https://github.com/intake/intake-stac

- Spatio-Temporal Asset Catalogs (STAC) are an emerging standard among imagery providers to simplify and unify search capabilities

- Intake is a Python-specific library for data catalog management

- Intake-STAC facilitates exploring STAC catalogs and loading imagery directly into Python for interactive computation

Example: Static STAC Catalogs

[1]:
```
%%time

import intake # Automatically will discover intake-stac installed

item = intake.open_stac_item('https://sat-api-dev.developmentseed.org/collections/landsat-8-l1/items/LC80090142019038LGN00')
da = item.B1(chunks=dict(band=1,x=2048,y=2048)).to_dask()
da
```

```
CPU times: user 1.34 s, sys: 153 ms, total: 1.49 s
Wall time: 3.18 s
```

[1]: xarray.DataArray (**band**: 1, **y**: 8591, **x**: 8541)

dask.array<chunksize=(1, 2048, 2048), meta=np.ndarray>

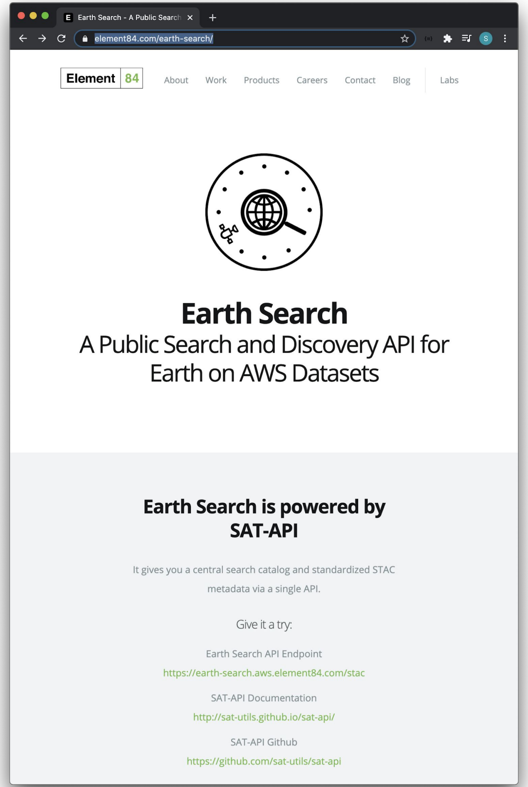
▼ Coordinates:

band	(band)	int64	1
y	(y)	float64	7.406e+06 7.405e+06 ... 7.148e+06
x	(x)	float64	2.37e+05 2.37e+05 ... 4.932e+05

▼ Attributes:

transform :	(30.0, 0.0, 236985.0, 0.0, -30.0, 7405515.0)
crs :	+init=epsg:32622
res :	(30.0, 30.0)
is_tiled :	1
nodatavals :	(nan,)

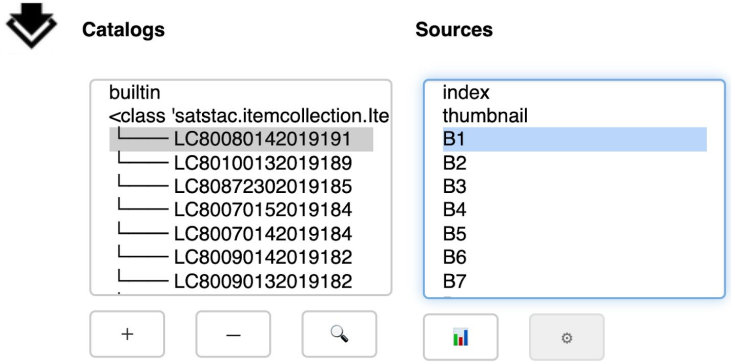
Example: Search with STAC-APIs



[4]:
```
# Search STAC API
results = satsearch.Search.search(
    collection='landsat-8-l1',
    bbox=[-55, 65, -53, 66],
    datetime='2019-06-01/2019-07-15',
    property=["landsat:tier=T1"])

# Load with Intake-STAC
catalog = intake.open_stac_item_collection(results.items())
intake.gui.add(catalog)
intake.gui
```

[4]:



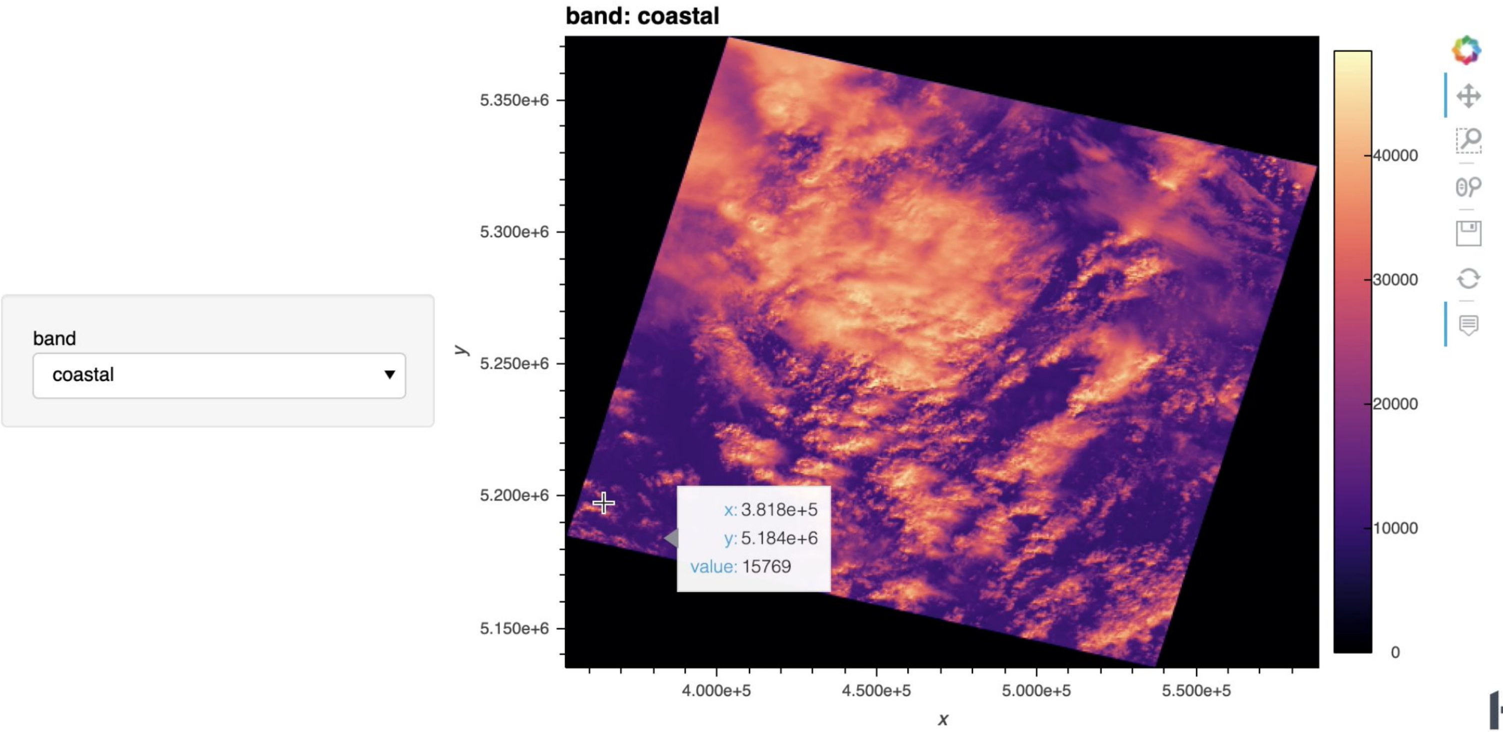

```
name: B1
container: xarray
plugin: ['rasterio']
description: Band 1 (coasta
l)
direct_access: True
user_parameters: []
metadata:
  type: image/x.geotiff
  eo:bands: [0]
  title: Band 1 (coastal)
  href: https://landsat-pd
s.s3.amazonaws.com/c1/L8/00
8/014/LC08_L1TP_008014_2019
```

https://www.element84.com/earth-search/

Interactive visualizations

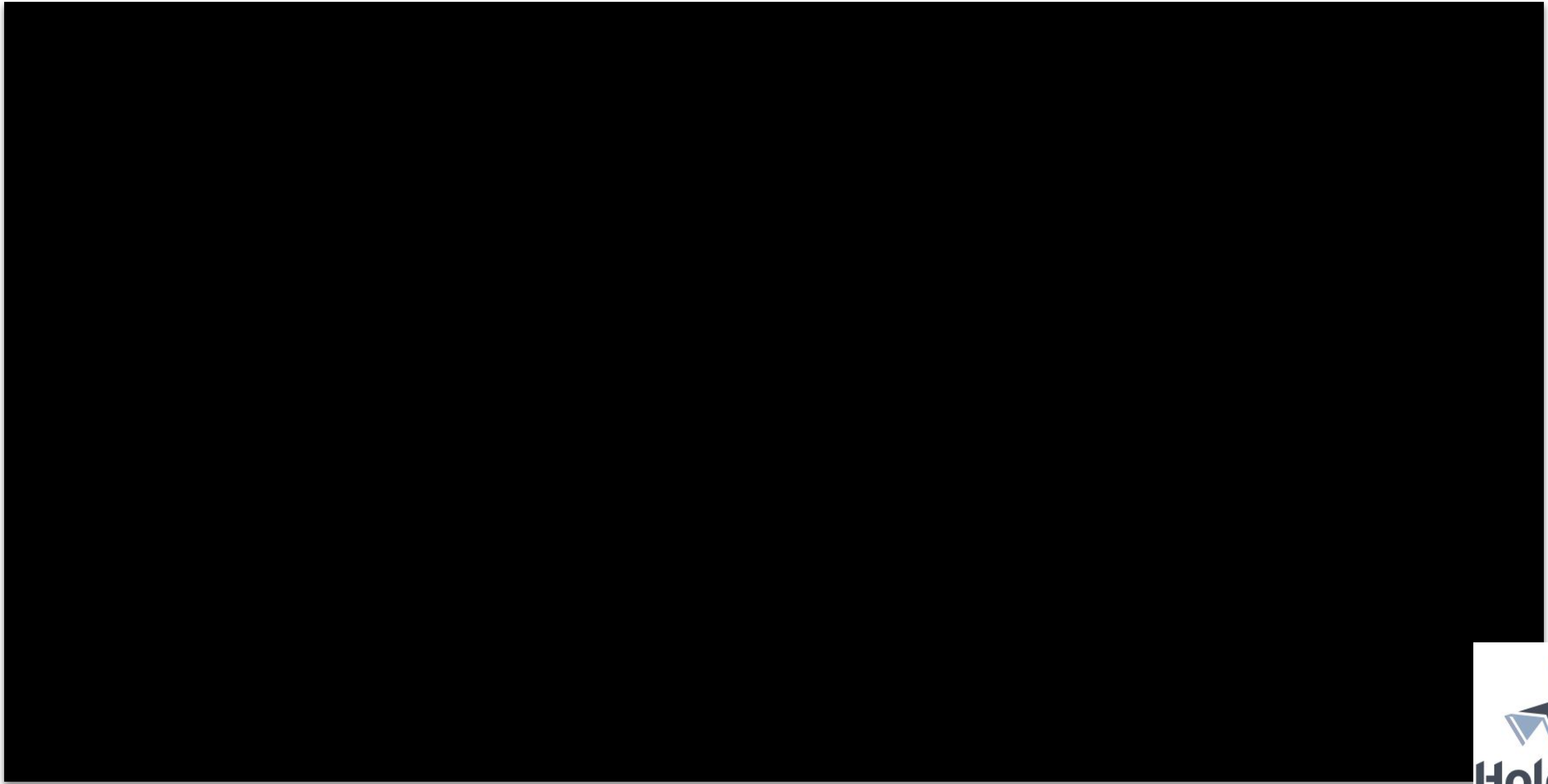
```
import hvplot.xarray

da.hvplot.image(groupby='band', rasterize=True, dynamic=True, cmap='magma',
                width=700, height=500,  widget_location='left')
```



HoloViz

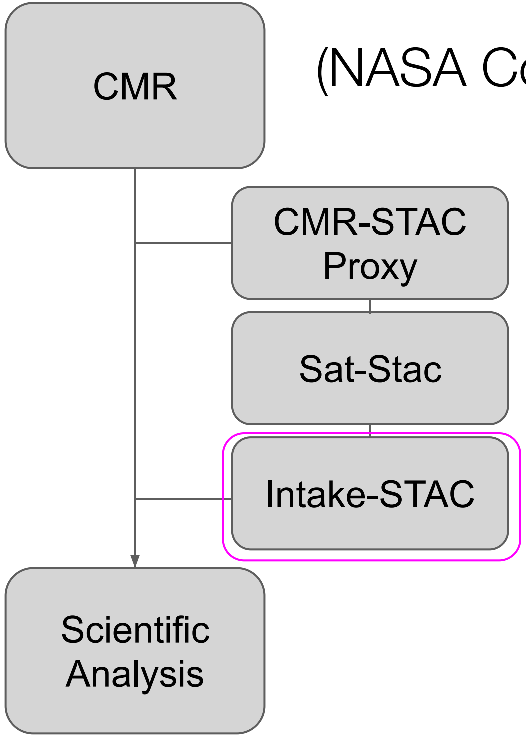
Interactive visualizations



HoloViz

PANGEO

Future Effort: Integrations with NASA CMR



(NASA Common Metadata Repository)

https://github.com/Element84/cmr-stac-api-proxy

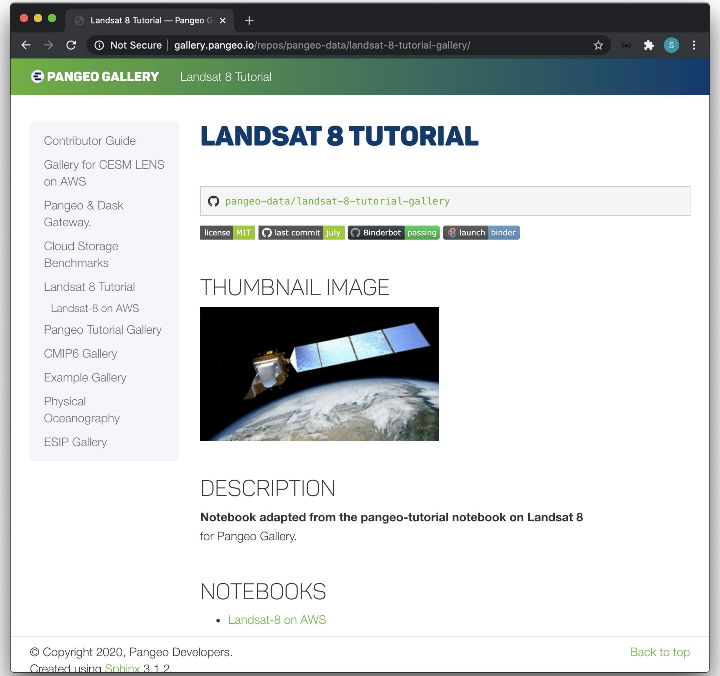
- If CMR Searches return STAC metadata for collections on AWS S3, we can reuse intake-STAC machinery

DEMO!

PANGEO

Scaling out to large archives

http://gallery.pangeo.io/repos/pangeo-data/landsat-8-tutorial-gallery/




```
[26]: print('Dataset size: [Gb]', DS.nbytes/1e9)
      DS
```

Dataset size: [Gb] 77.357853784

[26]: xarray.Dataset

▶ Dimensions: (**time**: 19, **x**: 7981, **y**: 7971)

▼ Coordinates:

y	(y)	float64	5.374e+06 5.374e+06 ... 5.135e+06
x	(x)	float64	3.522e+05 3.522e+05 ... 5.916e+05
time	(time)	datetime64[ns]	2019-04-06T19:01:20 ... 2020-06-27T19:01:35

▼ Data variables:

coastal	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...
blue	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...
green	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...
red	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...
nir	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...
swir16	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...
swir22	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...
cirrus	(time, y, x)	float64	dask.array<chunksize=(1, 2048, 2048), meta=np.n...

Hackweeks

PANGEO

A community platform for Big Data geoscience

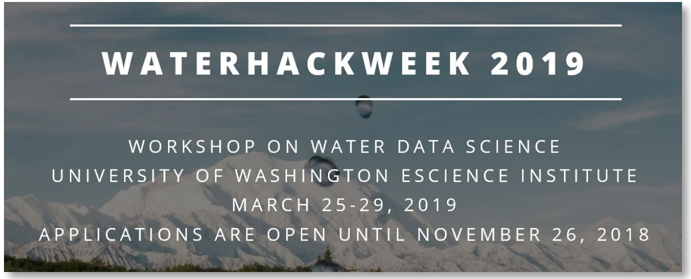
Growing and evolving since 2017!

contributors 62 | discourse 341 users | chat on gitter | follow @pangeo_data 2.4k

"Pangeo is first and foremost a *community* promoting open, reproducible, and scalable science." http://pangeo.io

Hackweeks to support community training

https://geohackweek.github.io



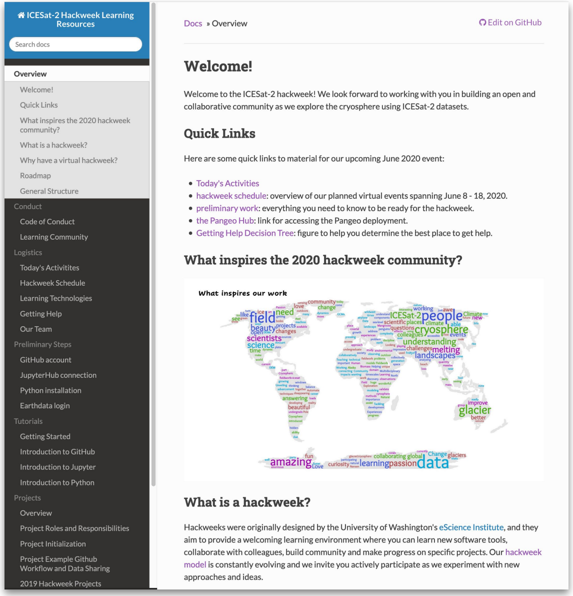
https://waterhackweek.github.io

UNIVERSITY of WASHINGTON
eScience Institute

https://icesat-2hackweek.github.io

What is a Hackweek?

A welcoming learning environment designed to build an *open and collaborative research community* while introducing participants to new software tools



Slides courtesy of Jessica Scheik, ESIP 2020

A typical Hackweek includes...

- Community building activities
- Curated computational environment (Pangeo)
- Hands on tutorials
- Interactive peer-to-peer learning
- Project time to "hack" on something of interest to you
- Access to a team of experts in your field AND open-source software

PANGEO

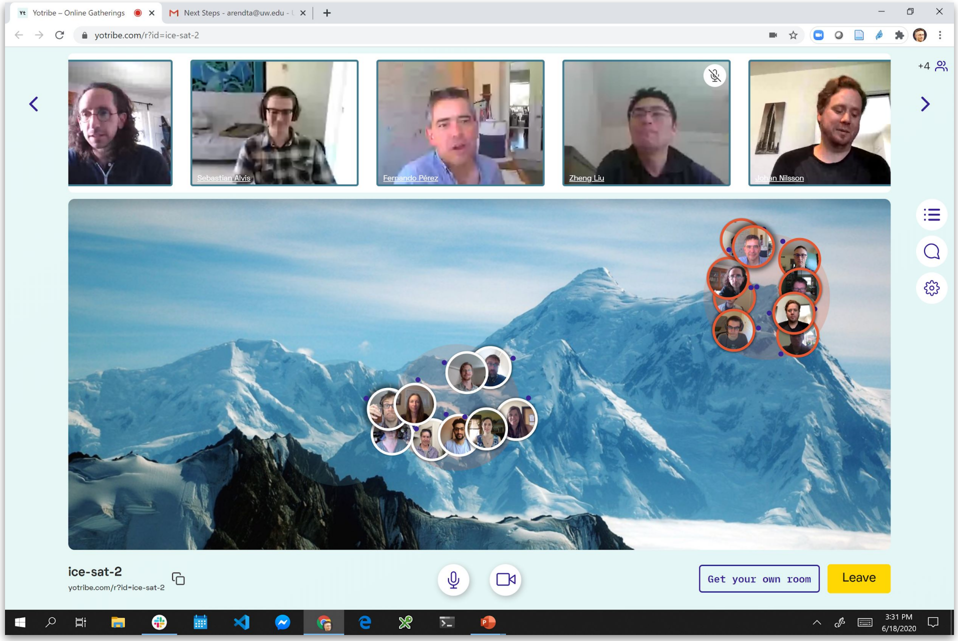
Cryosphere themed ICESat-2 Hackweek

ICESat-2
ICE, CLOUD, AND LAND ELEVATION SATELLITE-2

NASA

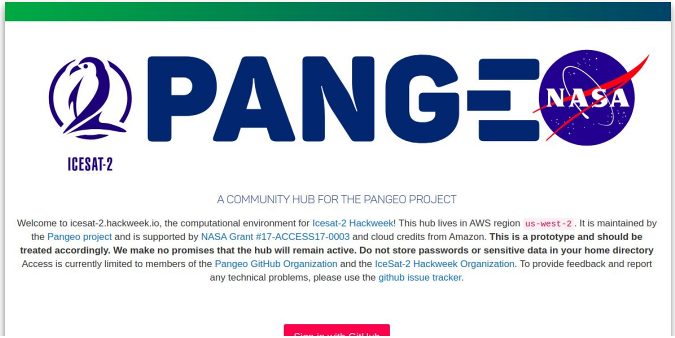
UNIVERSITY of WASHINGTON
eScience Institute

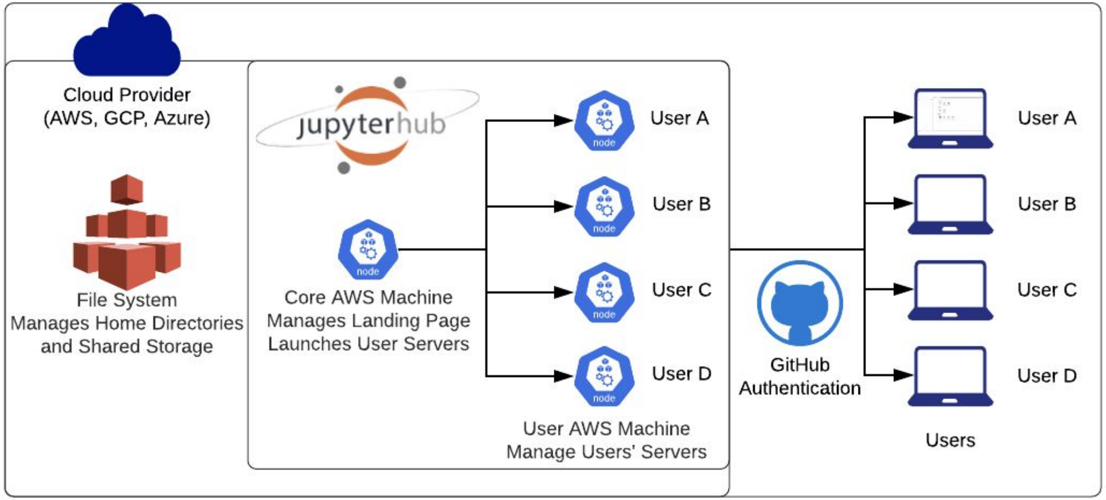
- June 2020
- 80+ participants
- First 100% virtual hackweek! Used Zoom+Slack.
- Only ~2 months to transition to virtual
- A great success!



Yotribe virtual happy hour!

https://icesat2-hackweek.io



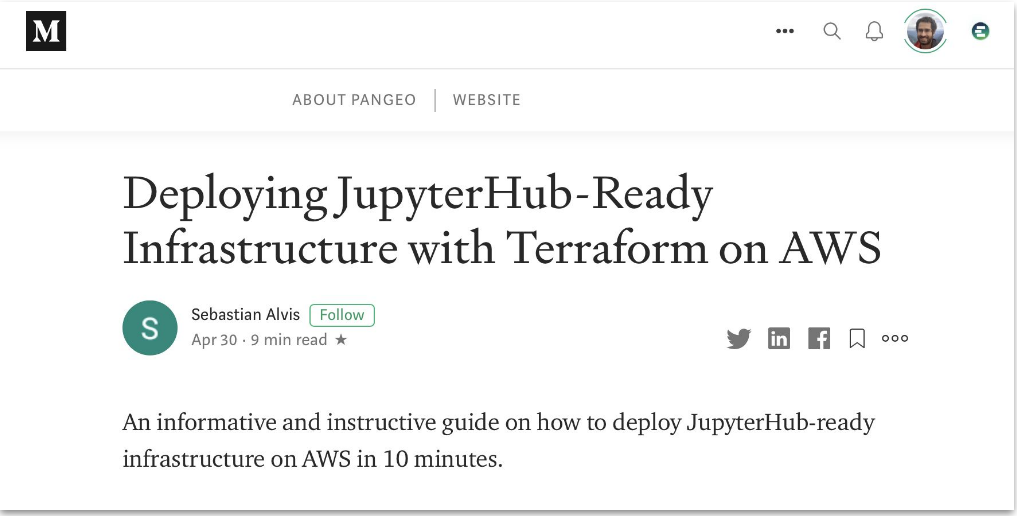


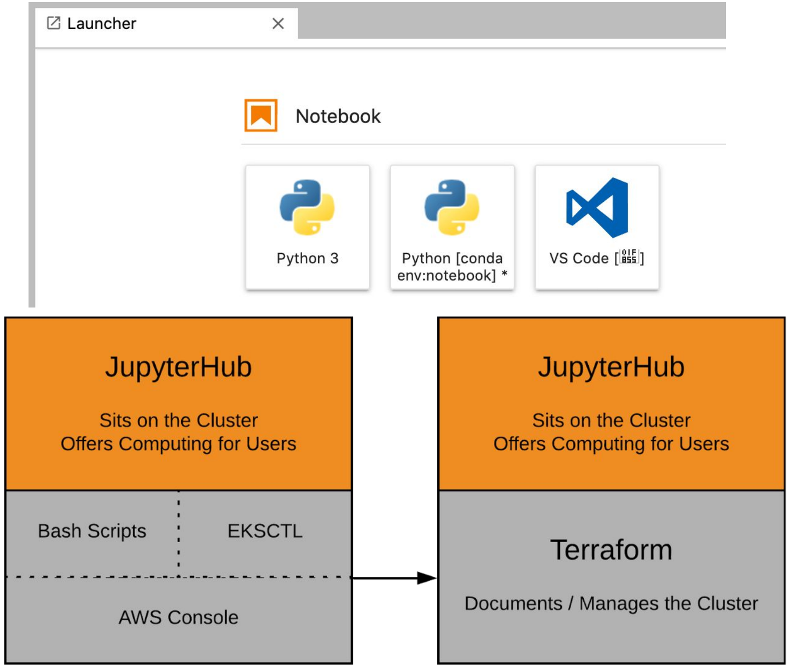
- Custom JupyterHub deployed for 2 months
- Key technology for facilitating collaboration
- On AWS-uswest2 (where NASA is starting to host icesat2 data)
- Total Cloud bill ~$1000

Slide courtesy of Sebastian Alvis, UW

Deploy your own Pangeo

https://medium.com/pangeo/terraform-jupyterhub-aws-34f2b725f4fd

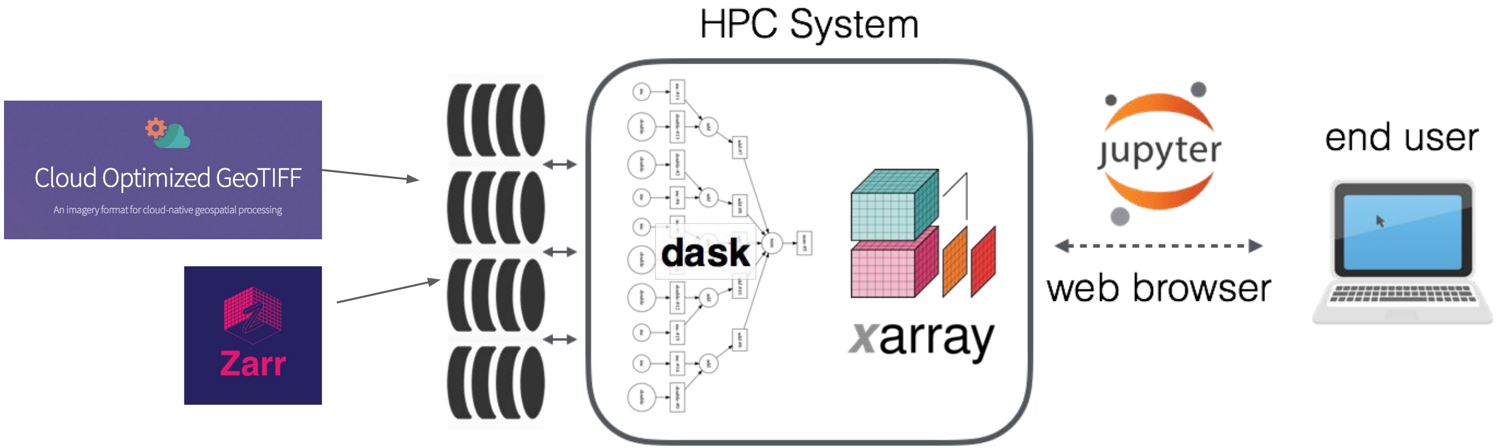




Sebastian Alvis, medium.com

Conclusions and Lessons Learned

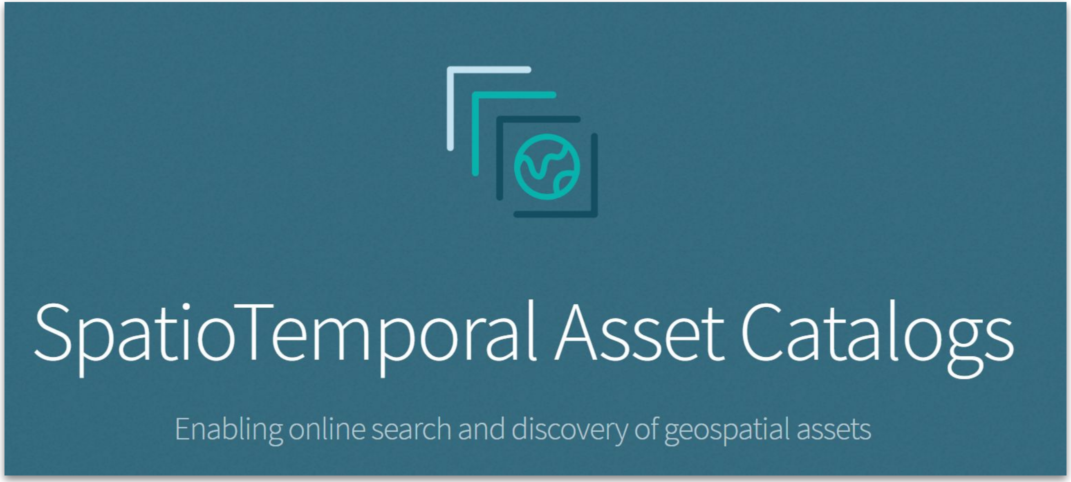
When hosting your data in the cloud, consider cloud-optimized formats.



Explore existing open-source solutions and avoid reinventing the wheel.

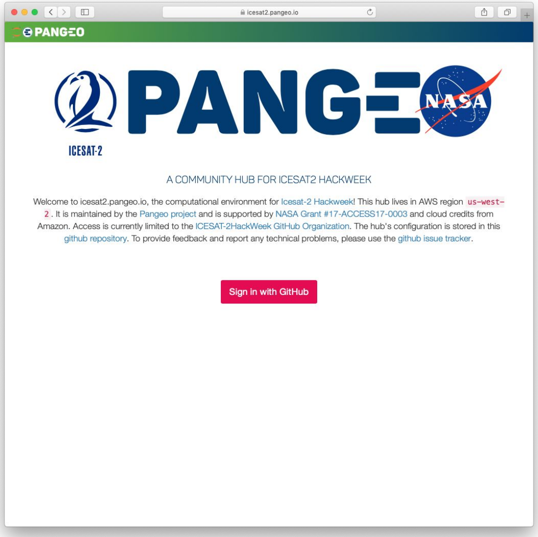
Credit: Stephan Hoyer, Jake Vanderplas (SciPy 2015)

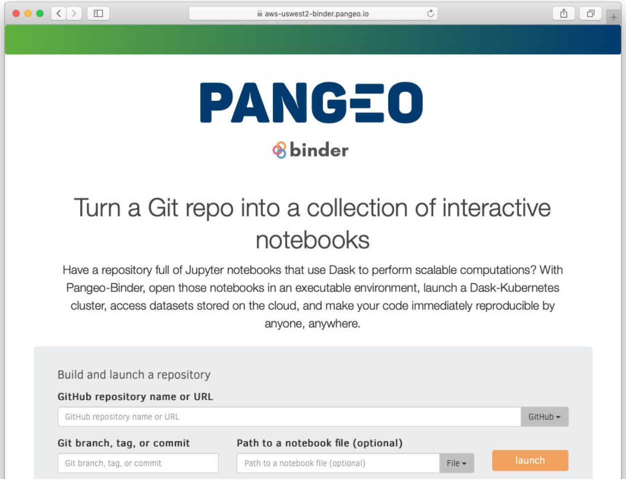
Infusion / interoperability is more likely when we adopt consistent community standards.

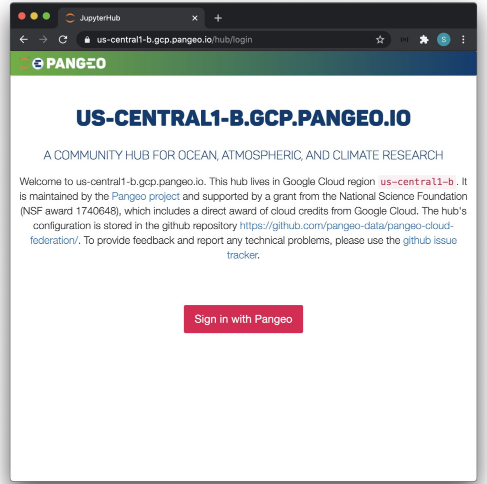


Education and outreach are critical to facilitate community adoption of cloud technologies.

Researchers will benefit from having clear funding models to support future adoption of JupyterHub and Binder toolkits.







Funding and other Contributors

Met Office

NSF

EarthCube

Google Cloud Platform

USGS

aws

NASA

NUMFOCUS
OPEN CODE = BETTER SCIENCE

Rh g
RHODIUM GROUP

UNIVERSITY of WASHINGTON
eScience Institute
ADVANCING DATA-INTENSIVE DISCOVERY IN ALL FIELDS

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Geophysical Fluid Dynamics Laboratory
GFDL
NOAA

JOHNS HOPKINS
UNIVERSITY

BIDS
BERKELEY INSTITUTE
FOR DATA SCIENCE

jupyter

ANACONDA
Powered by Continuum Analytics

ECMWF

PANGEO

A community platform for Big Data geoscience

contributors 62 | discourse 341 users | chat on gitter | follow @pangeo_data 2.4k

http://pangeo.io

@pangeo_data

https://github.com/pangeo-data