UNIVERSITY of WASHINGTON
eScience Institute

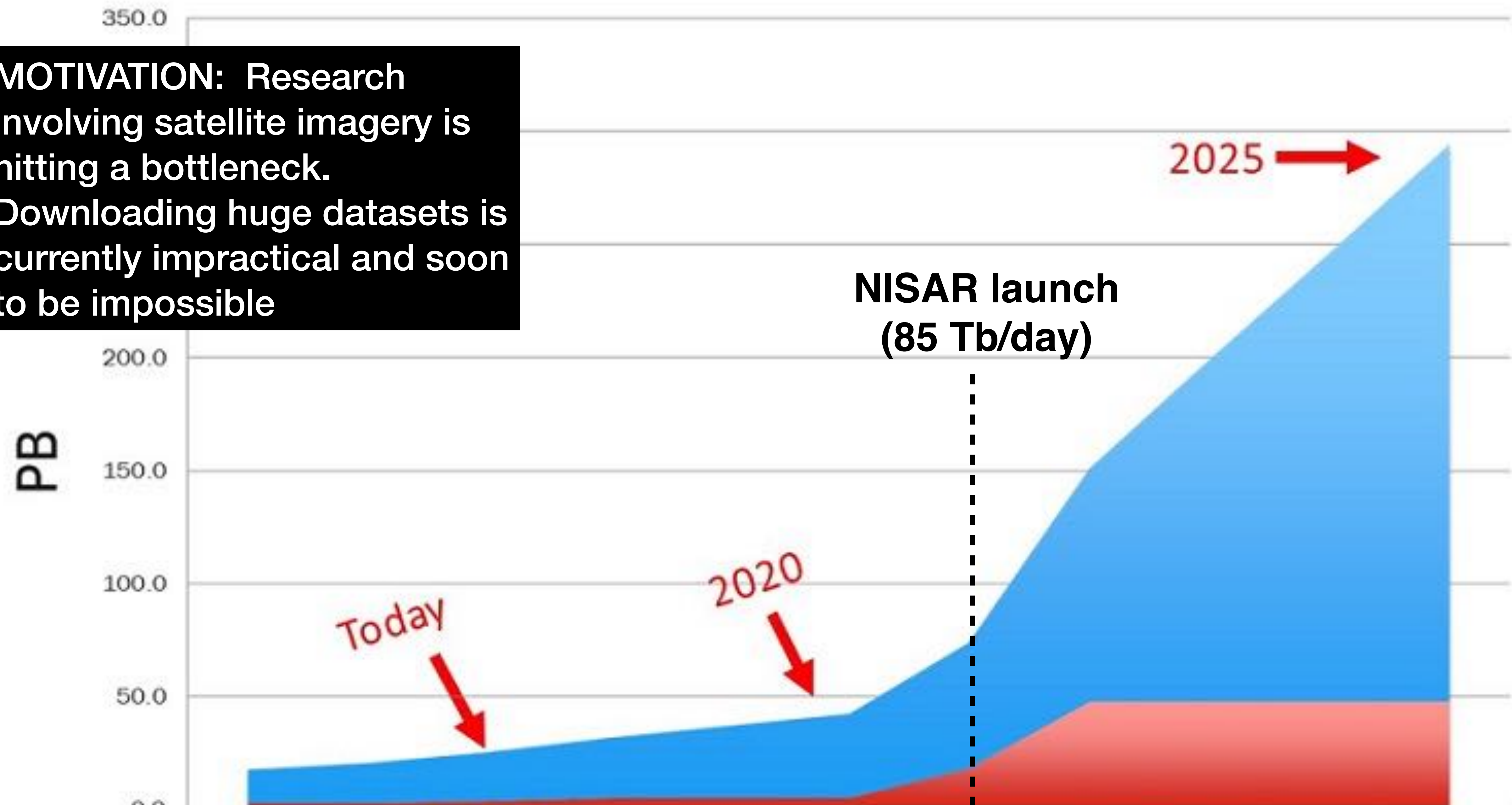
Cloud-optimized geospatial image analysis with Python

Python in the Geosciences
May 1, 2018

Scott Henderson, PhD
eScience Postdoctoral Fellow
University of Washington
scottyh@uw.edu

NASA Earth Observing System Data and Information System (EOSDIS) Archive Growth

MOTIVATION: Research involving satellite imagery is hitting a bottleneck. Downloading huge datasets is currently impractical and soon to be impossible



	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025
umulative Archive Size (PB)	15.0	17.7	21.6	26.8	32.0	37.2	55.6	103.4	151.1	198.9	246.6
chive Growth Rate (PB)	2.6	2.8	3.9	5.2	5.2	5.2	18.4	47.7	47.7	47.7	47.7

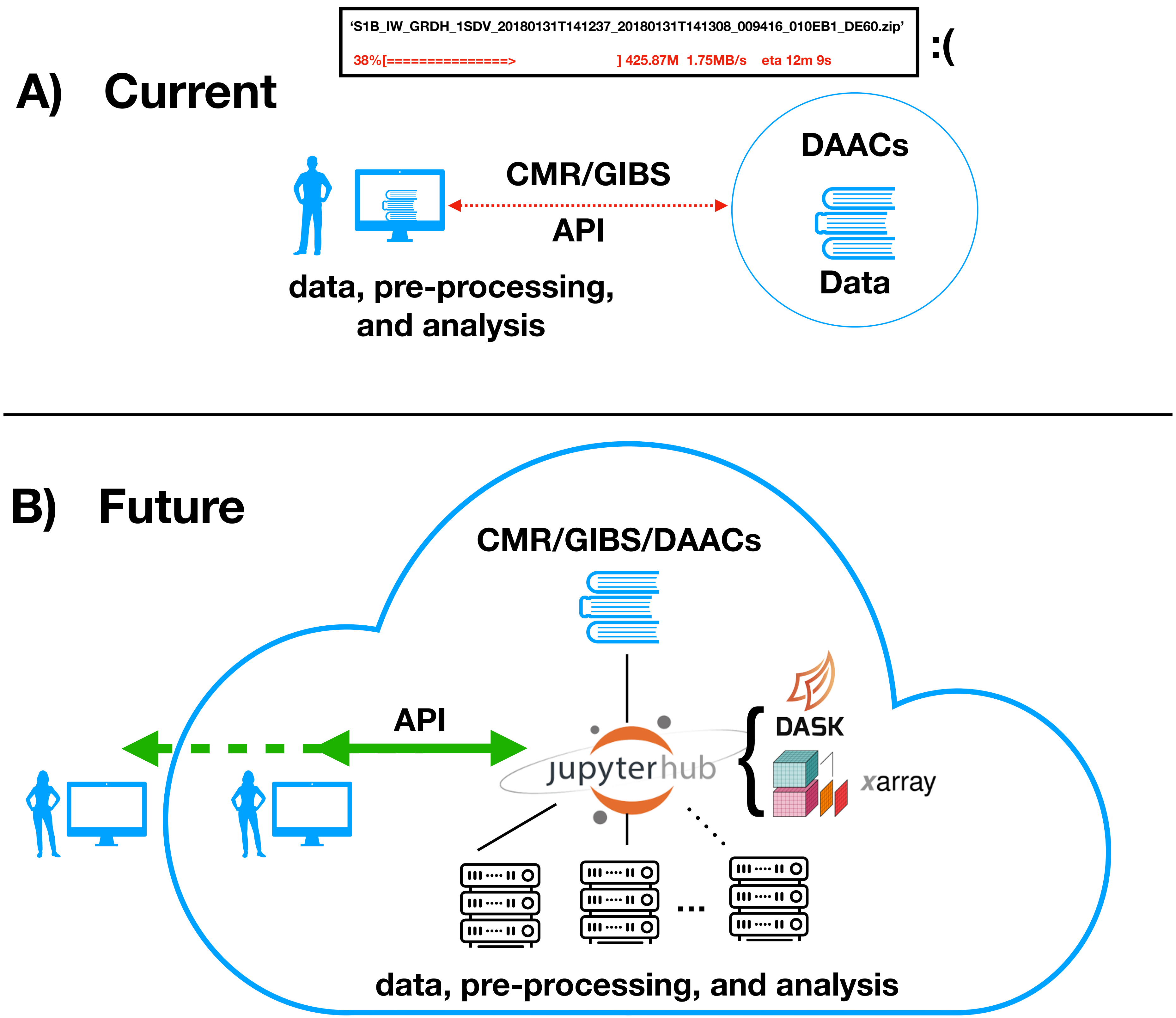
Archive Growth Rate (PB) Cumulative Archive Size (PB)

Implications

- NASA moving towards commercial cloud storage (Amazon Web Services, AWS) https://earthdata.nasa.gov/about/eosdis-cloud-evolution
    - Encouraging scientists to move processing and analysis to Cloud to avoid download costs
    - Major benefit for scientists in virtually unlimited computing resources
    - But, scientific community is not accustomed to cloud computing infrastructure and costs

My thoughts on “cloud-optimized”

- Easily discover data (preferably without setting up a dedicated server)
- Download image subsets instead of the entire files
- Run algorithms where imagery is stored, and download only results
- Scale analysis (global, high spatial and temporal resolution)
- Workflows deployable on any cloud-provider (AWS, GC, Azure…)
- Shouldn’t have to worry about security and costs



NASA ACCESS
2017 Proposal

Joe Hamman
Anthony Arendt
Rob Fatland
Scott Henderson
Amanda Tan
Don Setiawan

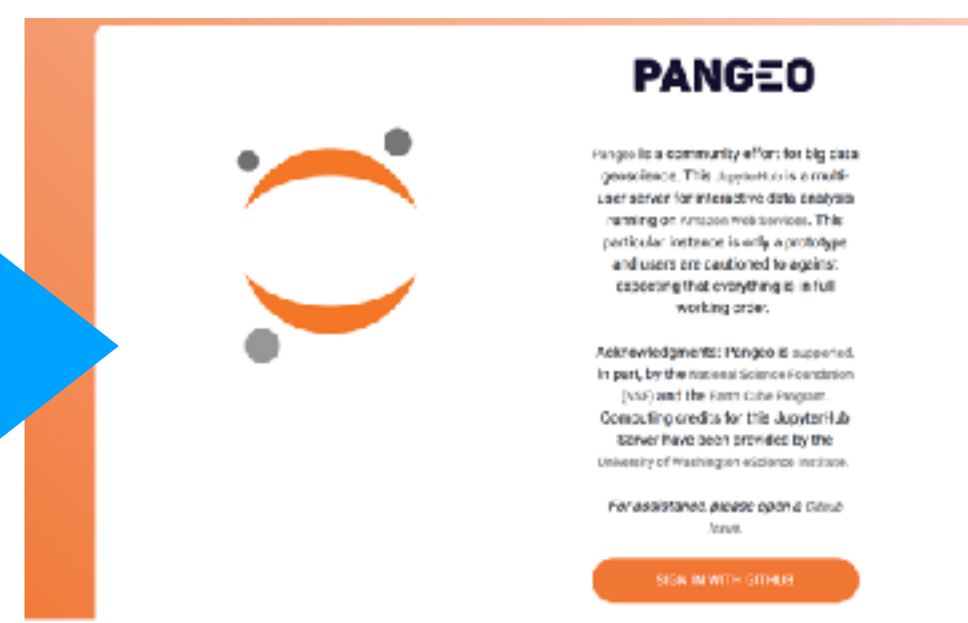
Some prototypes already working!

https://pangeo-data.github.io

Autoscaling JupyterHub Notebook on Google Cloud:
http://pangeo.pydata.org/hub/login

Description of technology by Matt Rocklin at Anaconda:
http://matthewrocklin.com/blog/work/2018/01/22/pangeo-2

Autoscaling JupyterHub Notebook on AWS:
https://pangeo-aws.cloudmaven.org



https://cloudmaven.github.io/

Data discovery

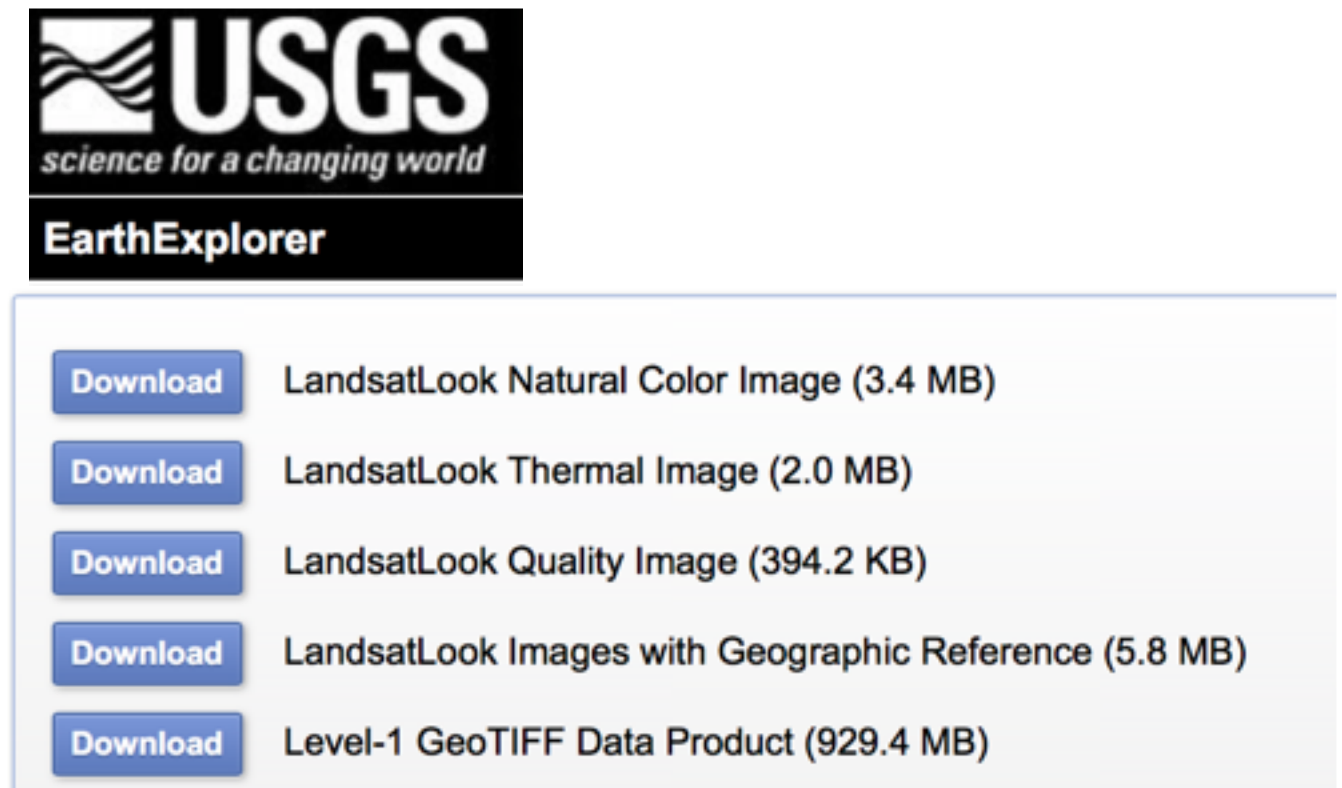
- **Current approach**: Depending on specific imagery type, go to particular web-app and click around a lot to get list of granules. Download all of them.
    - Radar: ASF DAAC, Snow: NSIDC
- **New approach**: Centralized standardized imagery search
    - NASA CMR / Earthdata Search
- **Future approach?** Why not just search with Google?
    - “Spatio-Temporal Asset Catalog” (STAC)

Transition is slow though…

EARTHDATA Search

EARTHDATA LOGIN

(or CMR API)



* After 10+ clicks and link redirections, entire image downloaded from LPDAAC

* **But Landsat-8 archive is mirrored on both Google Cloud and AWS as cloud-optimized geotiffs! Wouldn't it be nice to search and retrieve a link to the file you want in the Cloud you are working in? People are starting to achieve this:**

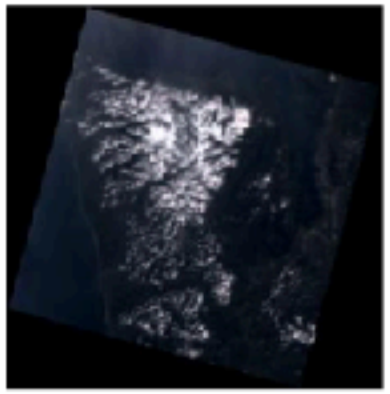
* **https://github.com/radiantearth/stac-spec/blob/master/roadmap.md**

Publicly hosted datasets

- Landsat-8 AWS hosting (https://registry.opendata.aws/landsat-8/)
- Data is stored in a structured way (object store), but no GeoServer exists for spatial queries and other operations. These applications are built on top:
    - https://landsatonaws.com
    - https://pythonhosted.org/landsat-util/
- Emerging standardization in static storage structure promises to reduce re-inventing the wheel (Spatio-Temporal Asset Catalog)
    - https://github.com/radiantearth/stac-spec
- Fees for both storage and download over internet (NOT to AWS intranet), if public AWS/NASA pays. “Requester pays” is another option.

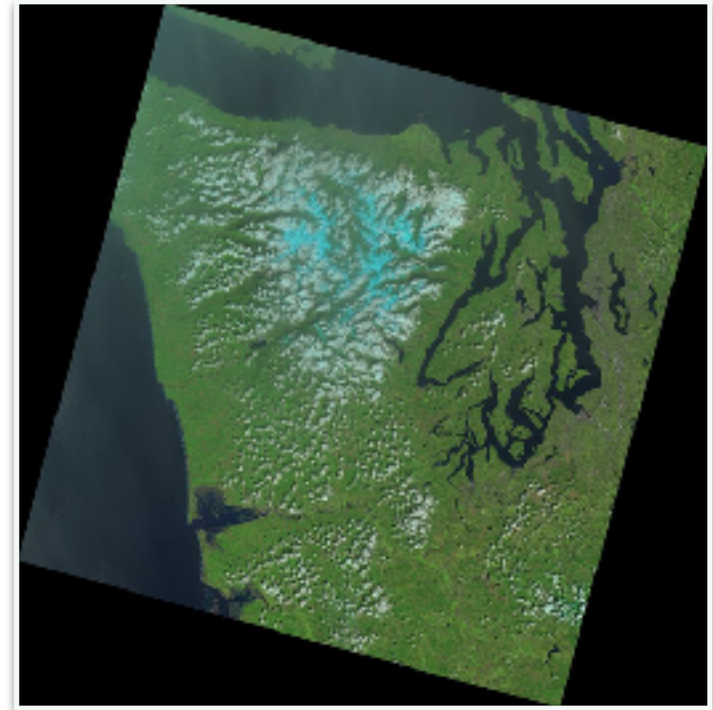
USGS 4/03 LANDSAT OVER SEATTLE: LC08_L1TP_047027_20180419_20180420_01_RT

https://landsat-pds.s3.amazonaws.com/c1/L8/047/027/LC08_L1TP_047027_20180419_20180420_01_RT/index.html



Files

- LC08_L1TP_047027_20180419_20180420_01_RT_B4_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B2.TIF (56.3MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B5.TIF.ovr (8.1MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B10_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B9_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B6_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_BQA.TIF (2.0MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B2_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B4.TIF (58.4MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B2.TIF.ovr (7.5MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B7.TIF (58.4MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B7.TIF.ovr (7.7MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B1.TIF.ovr (7.4MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B3_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B11.TIF (44.7MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B1_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B3.TIF.ovr (7.6MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B10.TIF.ovr (7.1MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B7_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_MTL.txt (8.5KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B9.TIF.ovr (4.7MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_ANG.txt (0.1MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B8.TIF.ovr (29.5MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B5_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B11_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B6.TIF (60.2MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B3.TIF (57.7MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_BQA_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B4.TIF.ovr (7.7MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B8.TIF (228.2MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B5.TIF (62.3MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B9.TIF (39.8MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_BQA.TIF.ovr (0.5MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B8_wrk.IMD (11.3KB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B1.TIF (56.2MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B11.TIF.ovr (7.0MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B6.TIF.ovr (7.9MB)
- LC08_L1TP_047027_20180419_20180420_01_RT_B10.TIF (46.2MB)



https://console.cloud.google.com/storage/browser/gcp-public-data-landsat/LC08/01/047/027/LC08_L1TP_047027_20180419_20180420_01_RT/

Google Cloud Platform · Select a project

Storage · Browser · UPLOAD FILES · UPLOAD FOLDER · CREATE FOLDER · REFRESH · SHARE PUBLICLY · DELETE

Filter by prefix...

Buckets / gcp-public-data-landsat / LC08 / 01 / 047 / 027 / LC08_L1TP_047027_20180419_20180420_01_RT

Name	Size	Type	Storage class	Last modified	Share publicly
LC08_L1TP_047027_20180419_20180420_01_RT_ANG.txt	114.52 KB	application/octet-stream	Multi-Regional	4/20/18, 6:26 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B1.TIF	66.92 MB	application/octet-stream	Multi-Regional	4/20/18, 6:26 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B10.TIF	54.06 MB	application/octet-stream	Multi-Regional	4/20/18, 6:26 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B11.TIF	52.03 MB	application/octet-stream	Multi-Regional	4/20/18, 6:27 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B2.TIF	67.1 MB	application/octet-stream	Multi-Regional	4/20/18, 6:27 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B3.TIF	68.99 MB	application/octet-stream	Multi-Regional	4/20/18, 6:27 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B4.TIF	70.6 MB	application/octet-stream	Multi-Regional	4/20/18, 6:27 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B5.TIF	79.95 MB	application/octet-stream	Multi-Regional	4/20/18, 6:28 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B6.TIF	75.14 MB	application/octet-stream	Multi-Regional	4/20/18, 6:28 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B7.TIF	71.19 MB	application/octet-stream	Multi-Regional	4/20/18, 6:28 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B8.TIF	271.77 MB	application/octet-stream	Multi-Regional	4/20/18, 6:29 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_B9.TIF	45.43 MB	application/octet-stream	Multi-Regional	4/20/18, 6:29 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_BQA.TIF	2.56 MB	application/octet-stream	Multi-Regional	4/20/18, 6:29 PM	Public link
LC08_L1TP_047027_20180419_20180420_01_RT_MTL.txt	8.51 KB	application/octet-stream	Multi-Regional	4/20/18, 6:29 PM	Public link

sat-utils (api for cloud imagery search)

https://github.com/sat-utils

development SEED

```
sat-utils/bin/sat-search --satellite_name=landsat-8 --contains=-120.466565,46.526258 --date=2017-01-01,2017-02-15 --printmd
```

```
sat-utils/bin/sat-search --satellite_name=landsat-8 --contains=-120.466565,46.526258 --cloud=0,20 --printmd --save='l8.geojson'
```

```
import geopandas as gpd
gf = gpd.read_file('l8.geojson')
gf.head()
```

*** use w/n python script: https://github.com/sat-utils/sat-search/blob/develop/test/test_main.py**

http://www.cogeo.org

Cloud Optimized GeoTIFF

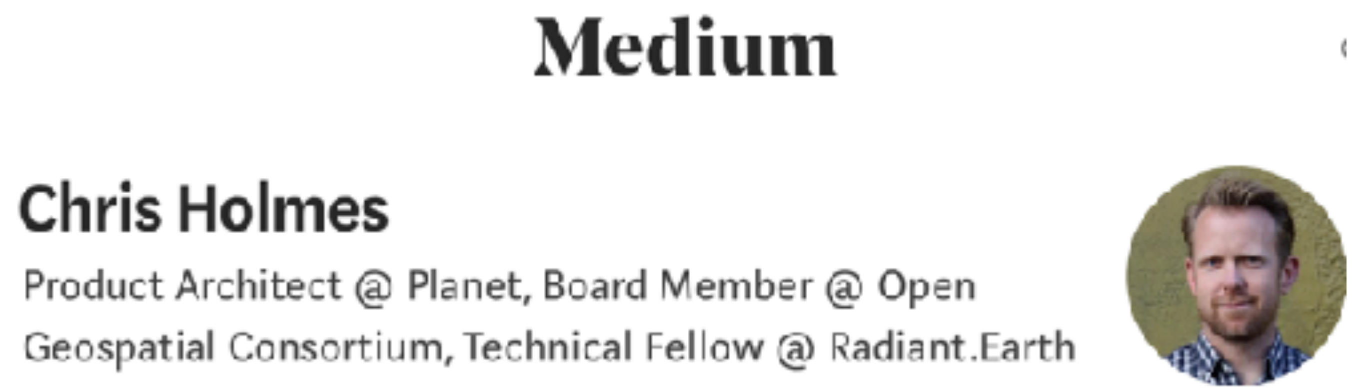
An imagery format for cloud-native geospatial processing

- **Allows for HTTP GET range requests to only retrieve portions of full file**
- **Metadata and overviews stored in front for speedy access**
- **Otherwise, it's the same-old geotiff and works in GIS programs!**

***How to create and test your own:**

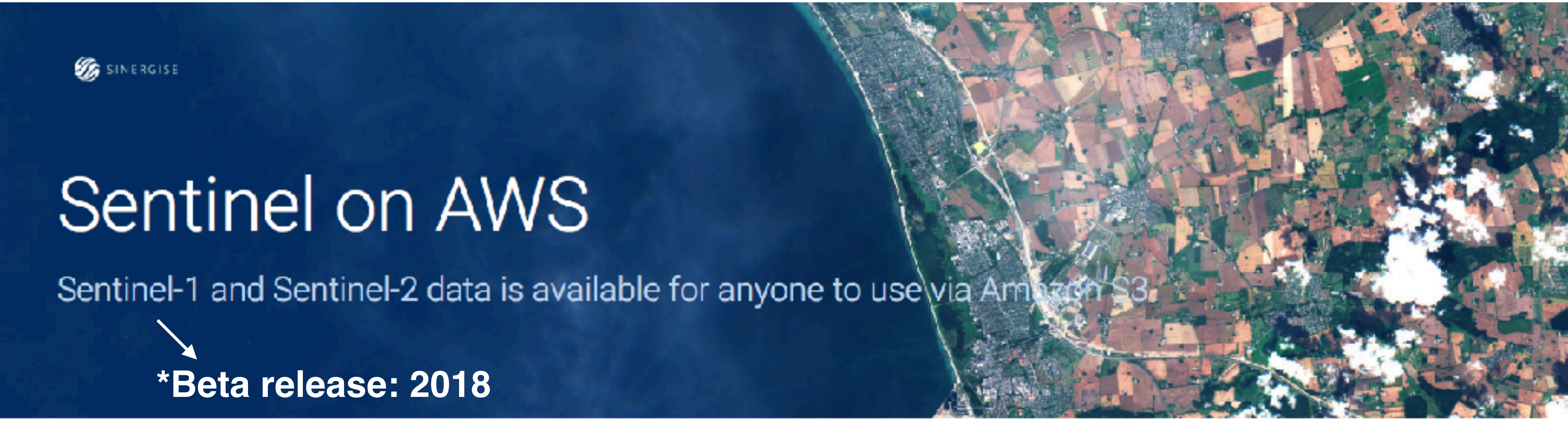
https://trac.osgeo.org/gdal/wiki/CloudOptimizedGeoTIFF

“Cloud Native Geoprocessing”



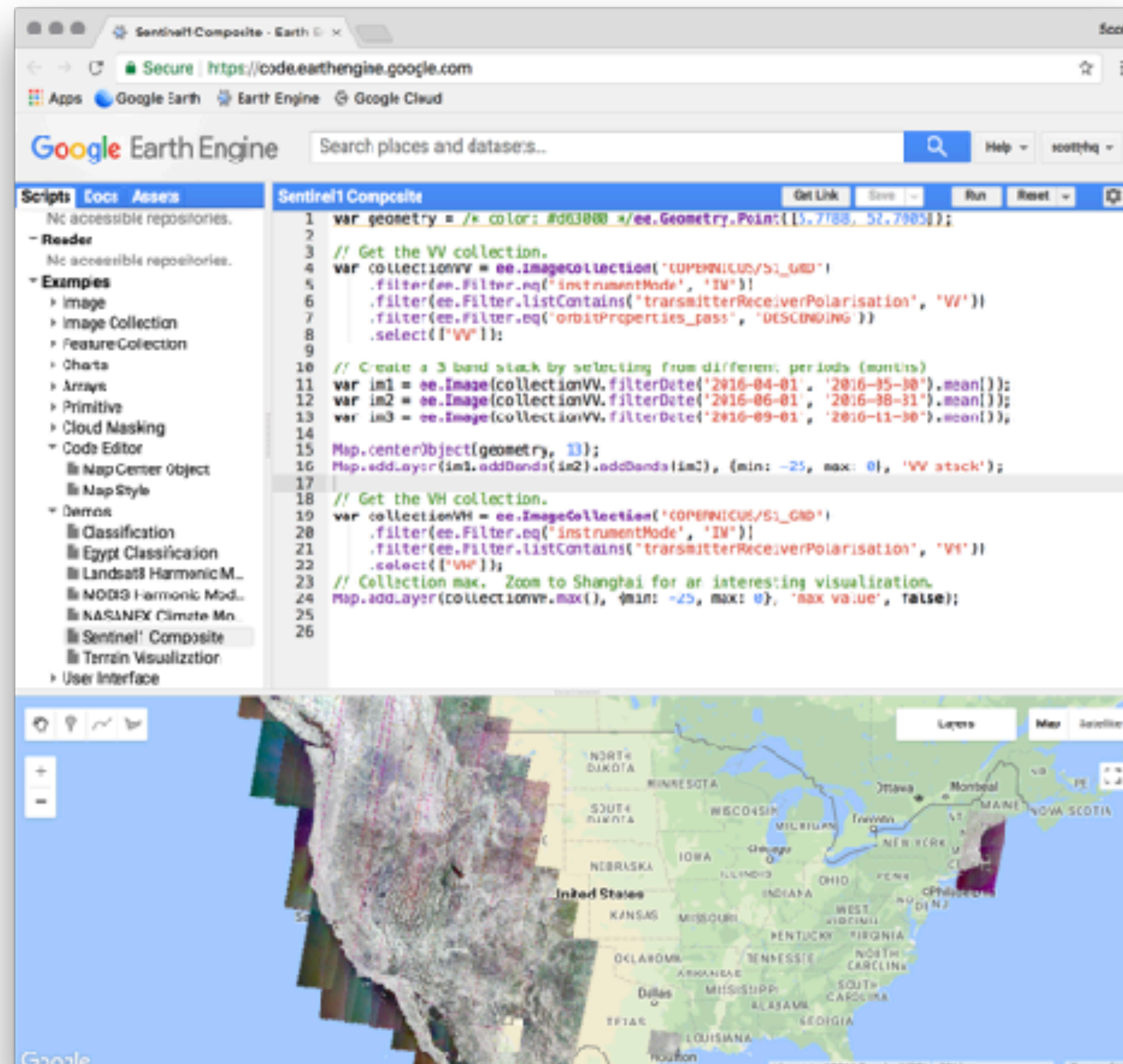
- Nice series of blog posts:
  - Part 1)
    1. The Basics
    2. The Cloud Optimized GeoTIFF
    3. Planet’s Cloud Native Geospatial Architecture
    4. Open Aerial Map’s Cloud Native Geospatial Architecture
    5. Cloud Native Geospatial Architecture Defined
    6. Metadata in a Cloud Native Geospatial World
    7. A Vision for the Cloud Native Geospatial Ecosystem
  - Part 2)
    8. Analysis-ready data
    9. Towards On-Demand Analysis Ready Data

New opportunity to conduct InSAR studies on AWS Cloud



Google Earth Engine

- Free!
- awesome web interface
- scalable algorithms applied next to data (Landsat, Sentinel1…):
  - https://earthengine.google.com/datasets/
- default language = JavaScript
- python api also available (but no unified web interface)



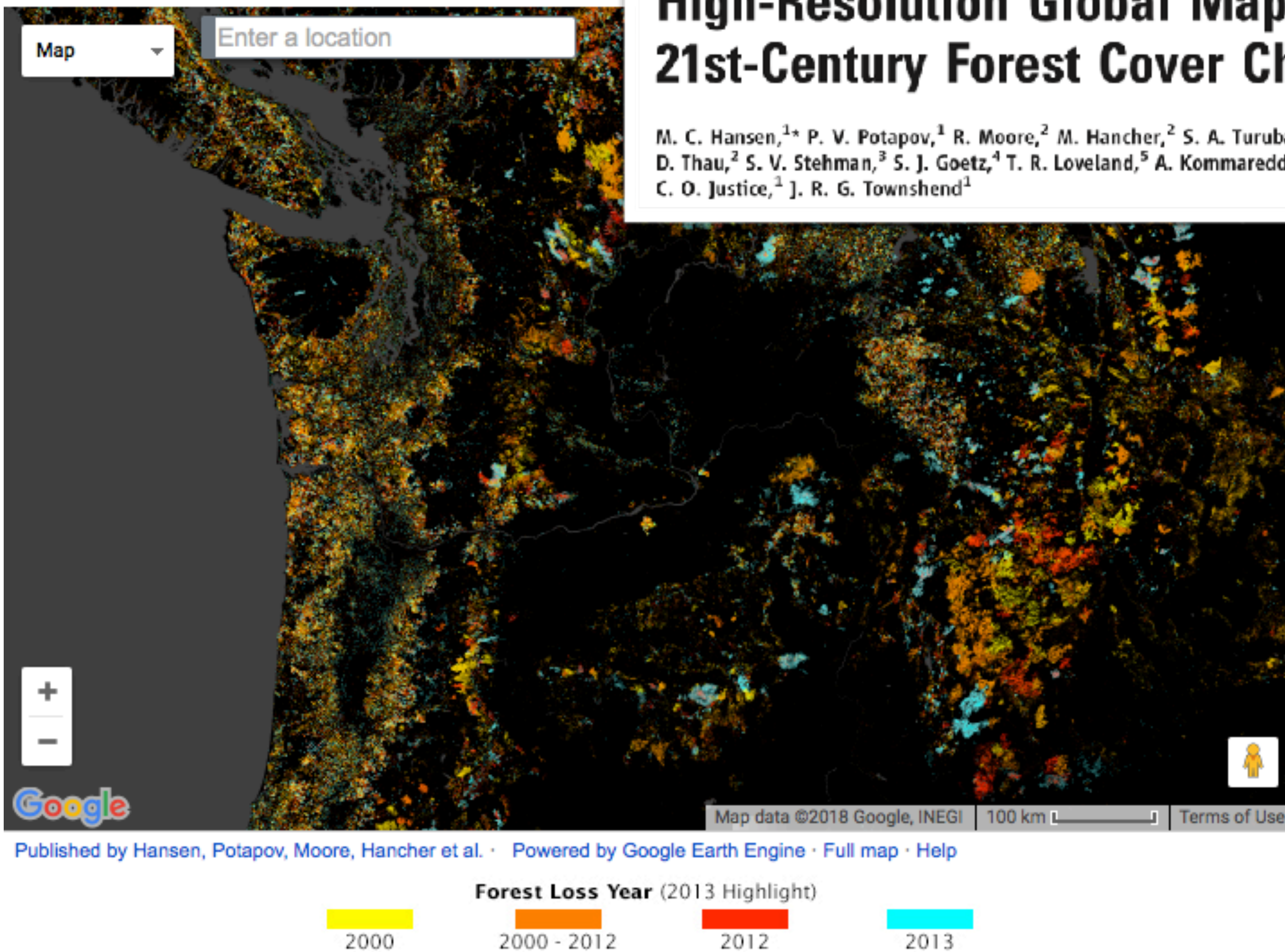
Big Data Helps Scientists Dig Deeper

Supercomputing + 40 Years of Landsat Images = A New Era for Earth Science

Science, Vol 342, November 2013

High-Resolution Global Maps of 21st-Century Forest Cover Change

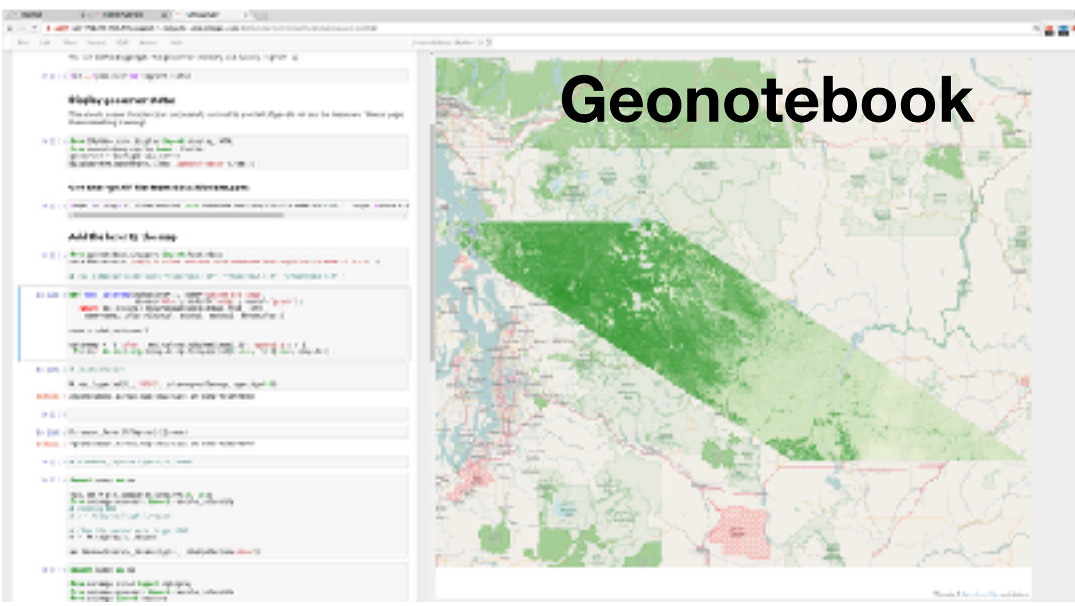
M. C. Hansen,[1*] P. V. Potapov,[1] R. Moore,[2] M. Hancher,[2] S. A. Turubanova,[1] A. Tyukavina,[1] D. Thau,[2] S. V. Stehman,[3] S. J. Goetz,[4] T. R. Loveland,[5] A. Kommareddy,[6] A. Egorov,[6] L. Chini,[1] C. O. Justice,[1] J. R. G. Townshend[1]

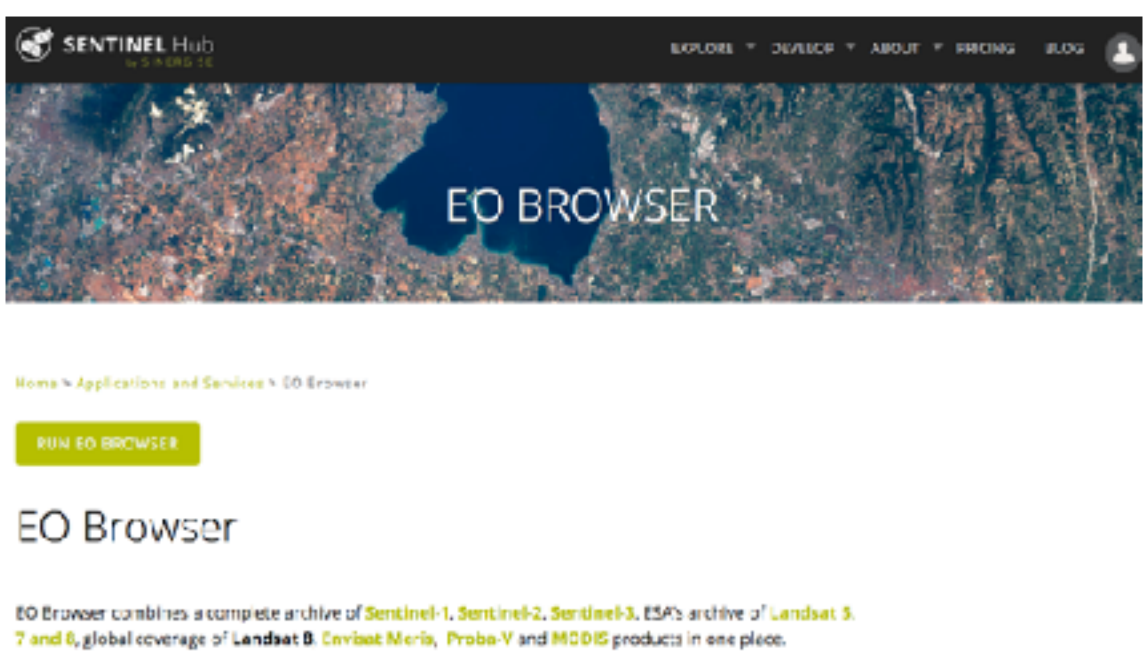


https://earthobservatory.nasa.gov/Features/LandsatBigData/

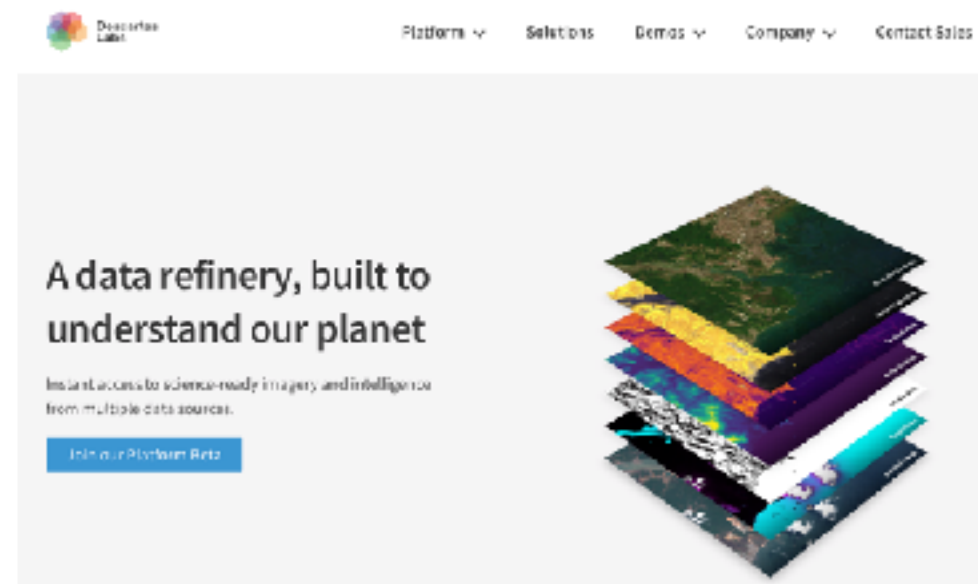
Emerging alternatives

Free / Semifree





Commercial



Raster Foundry

GeoTrellis

Cloud computing doesn't have to be expensive (in fact it can be free!)

Jupyter demos / sharing code:

binder

(beta)

*4G memory limit, data doesn't persist

Microsoft Azure Notebooks Preview

*4G memory limit per user and a 1G data limit

* ??

For larger projects:

Academic research credits:

https://aws.amazon.com/research-credits/

https://cloud.google.com/edu/

https://www.microsoft.com/en-us/research/ac

Google Earth Engine

*Really amazing. Javascript required :(

AWS Free Tier

https://aws.amazon.com/free/

*12 months, all kinds of caveats…

I just want to try things out on my mac…

* Since cloud platforms use linux, it's best to develop in linux environment:

https://github.com/jupyter/docker-stacks

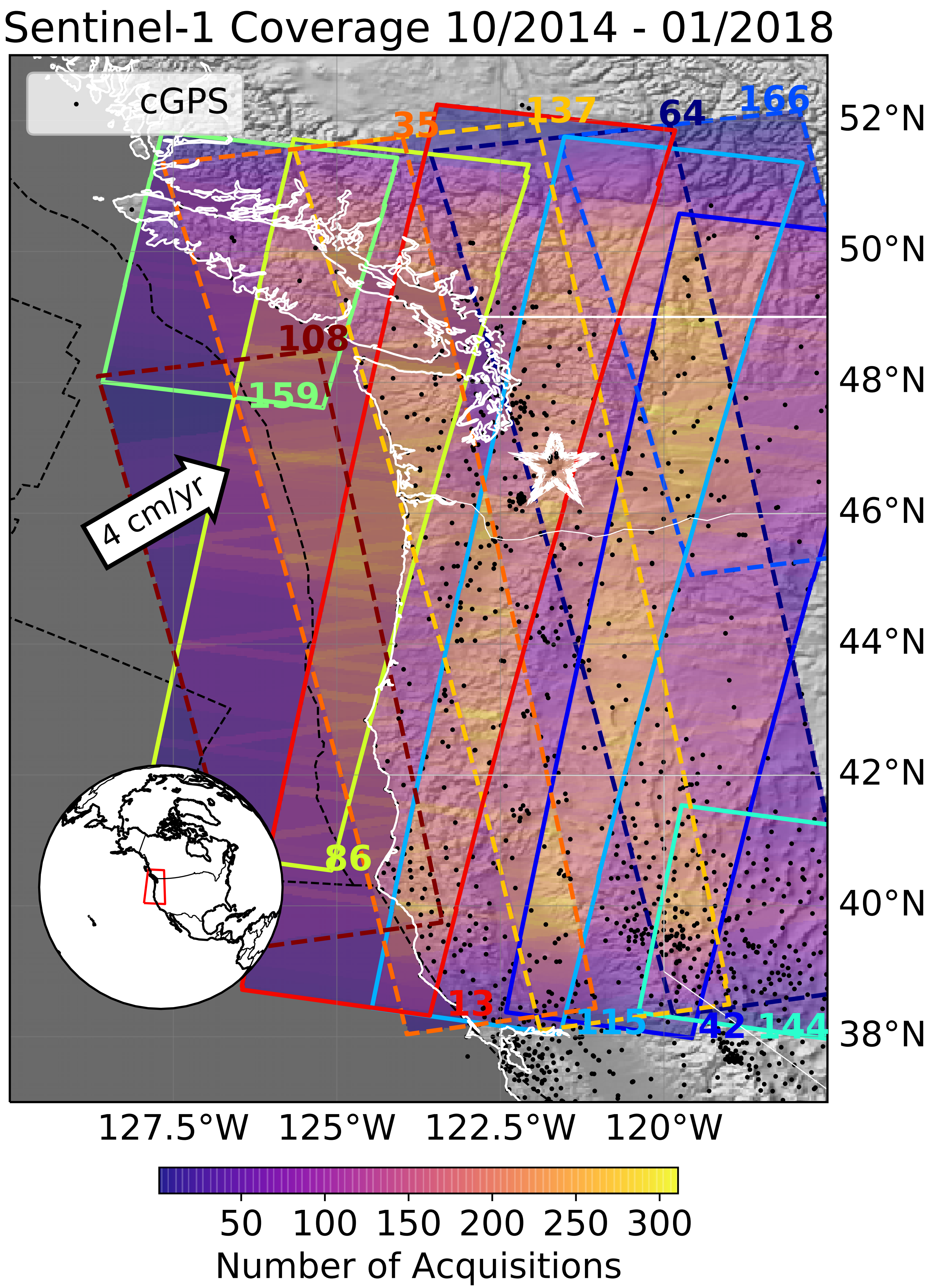
https://hub.docker.com/u/jupyter

```
docker run -it --rm -p 8888:8888 -v `pwd`:/home/jovyan/work -e NB_UID=$UID -e GRANT_SUDO=yes --user root jupyter/base-notebook start.sh jupyter lab
```

DEMO!

https://github.com/scottyhq/landsat-aws-demo

- Customizable landsat-8 NDVI analysis on AWS (launch locally, launch on AWS in same region as data, launch on google cloud w/ Binder)
- GOALS: utilize powerful and free tools, distinguish between what is happening on the server/cloud and locally on your laptop.



Notes on processing individual frame:

- **c5.4xlarge instance**
- **30m resolution**

time	2 hrs
final folder space	10 Gb

***Minimum requirements for entire region: 25 Tb**

- **Currently working w/ beta version of ISCE Sentinel stacking which works with a common master frame to save space and processing**
- **Running with GPU give 10x speedup, but more costly**

How many pairs?

$$m \leq \frac{n(n-1)}{2}$$

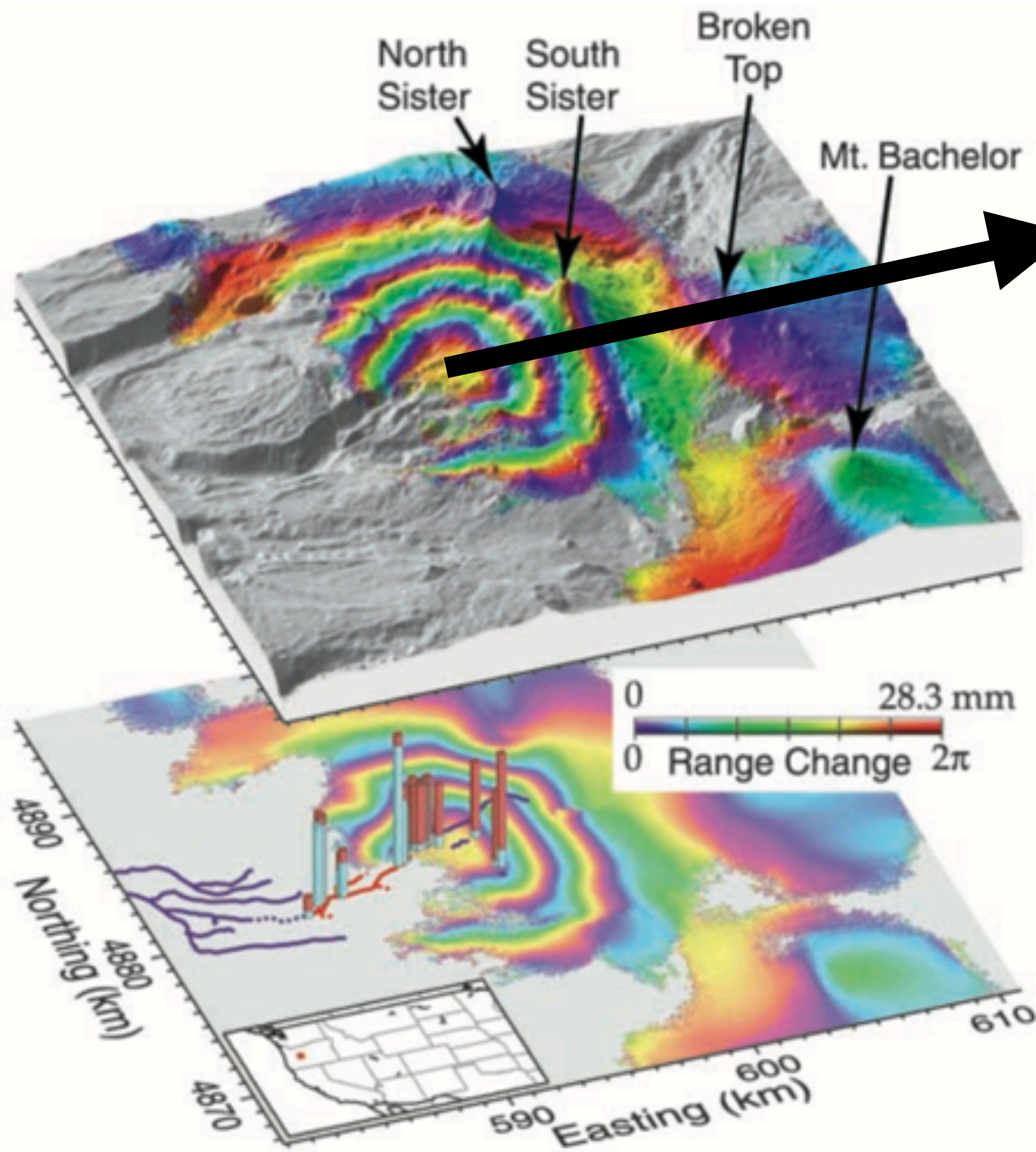
* 12 + 24 day pairs = (n-1) + (n-2) = 2n-3
* 3 consecutive = 3n-6

n dates	m pairs	3 consecutive	2 consecutive
3	3	3	3
4	6	6	5
5	10	9	7
6	15	12	9
10	45	24	17
20	190	54	37
50	1225	144	97

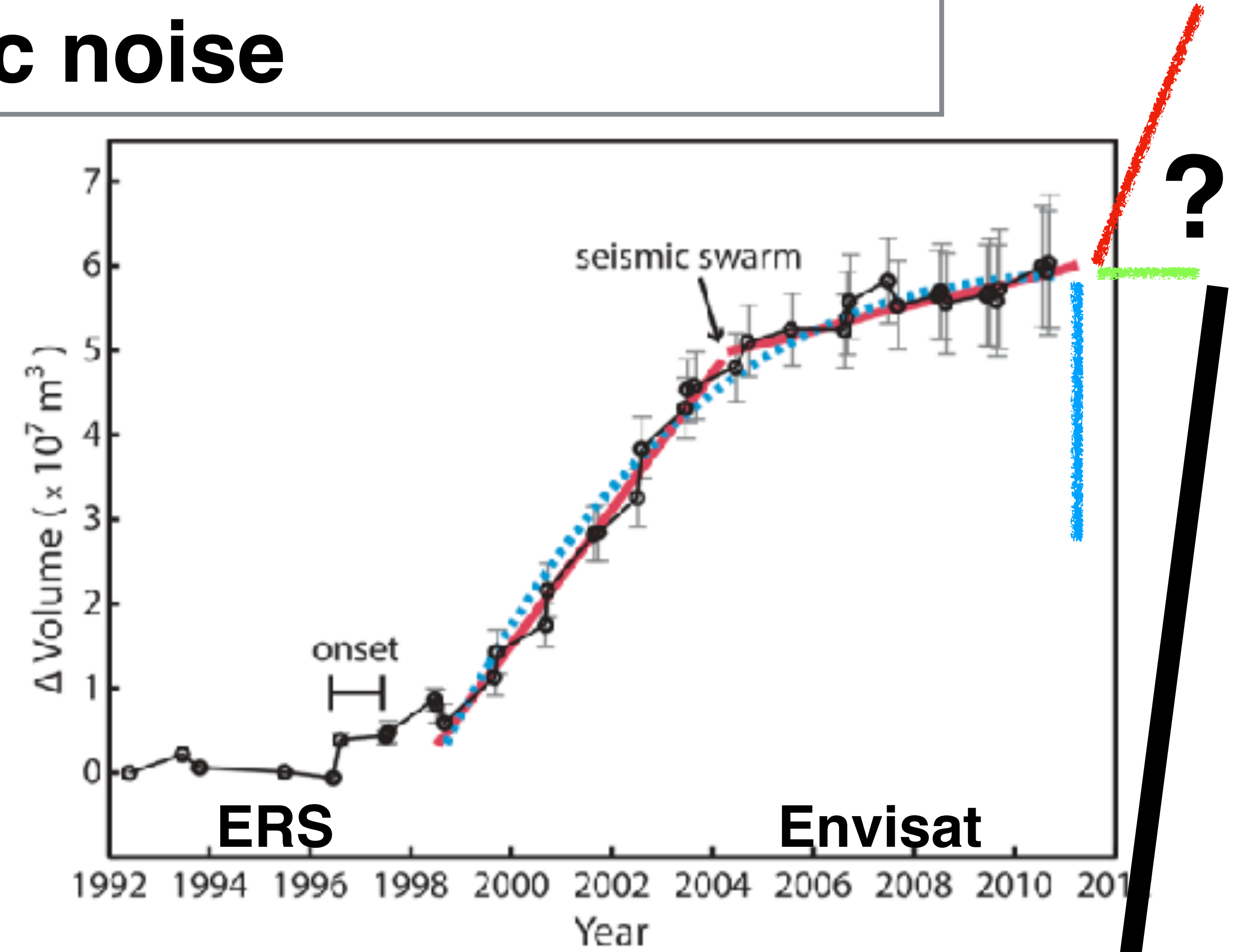
* need at least 4 dates
* redundancy increases as n^2

Why process many pairs? Low-amplitude signals/ atmospheric noise

Aseismic deep magmatic intrusion at Three Sisters, Oregon



Wicks et al 2002, GRL



Riddick & Schmidt 2011, G^3

Weekly observations, rather than monthly or yearly, will allow more precise timing of dynamic transitions, and correlation with other measurements (GPS, seismic, gas)

https://volcanoes.usgs.gov/volcanoes/three_sisters/three_sisters_geo_hist_129.html

AWS Batch

Fully Managed Batch Processing at Any Scale

“Batch dynamically provisions the optimal quantity and type of compute resources (e.g., CPU or memory optimized instances) based on the volume and specific resource requirements of the batch jobs submitted. With AWS Batch, there is no need to install and manage batch computing software or server clusters that you use to run your jobs, allowing you to focus on analyzing results and solving problems.”

… Sort of. In my experience there are many options to consider and lots of design decisions to make. A lot of this has to do with Docker and file storage management.

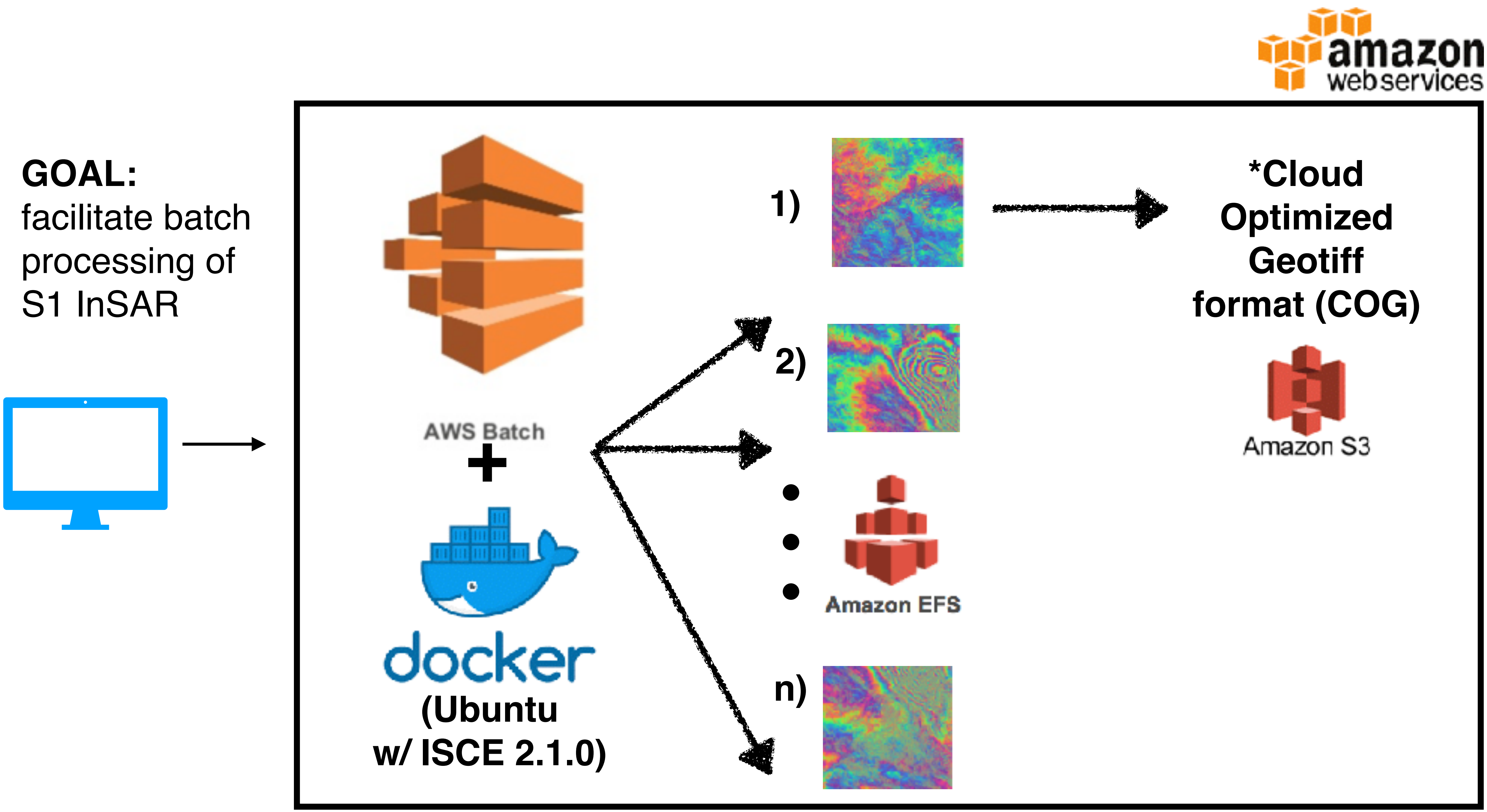


A great guide!
https://github.com/cloudmaven/Rosetta/wiki

InSAR in the Cloud

https://github.com/scottyhq/dinoSARaws

GOAL:
facilitate batch processing of S1 InSAR



http://int-20170828-20170816.s3-website-us-west-2.amazonaws.com

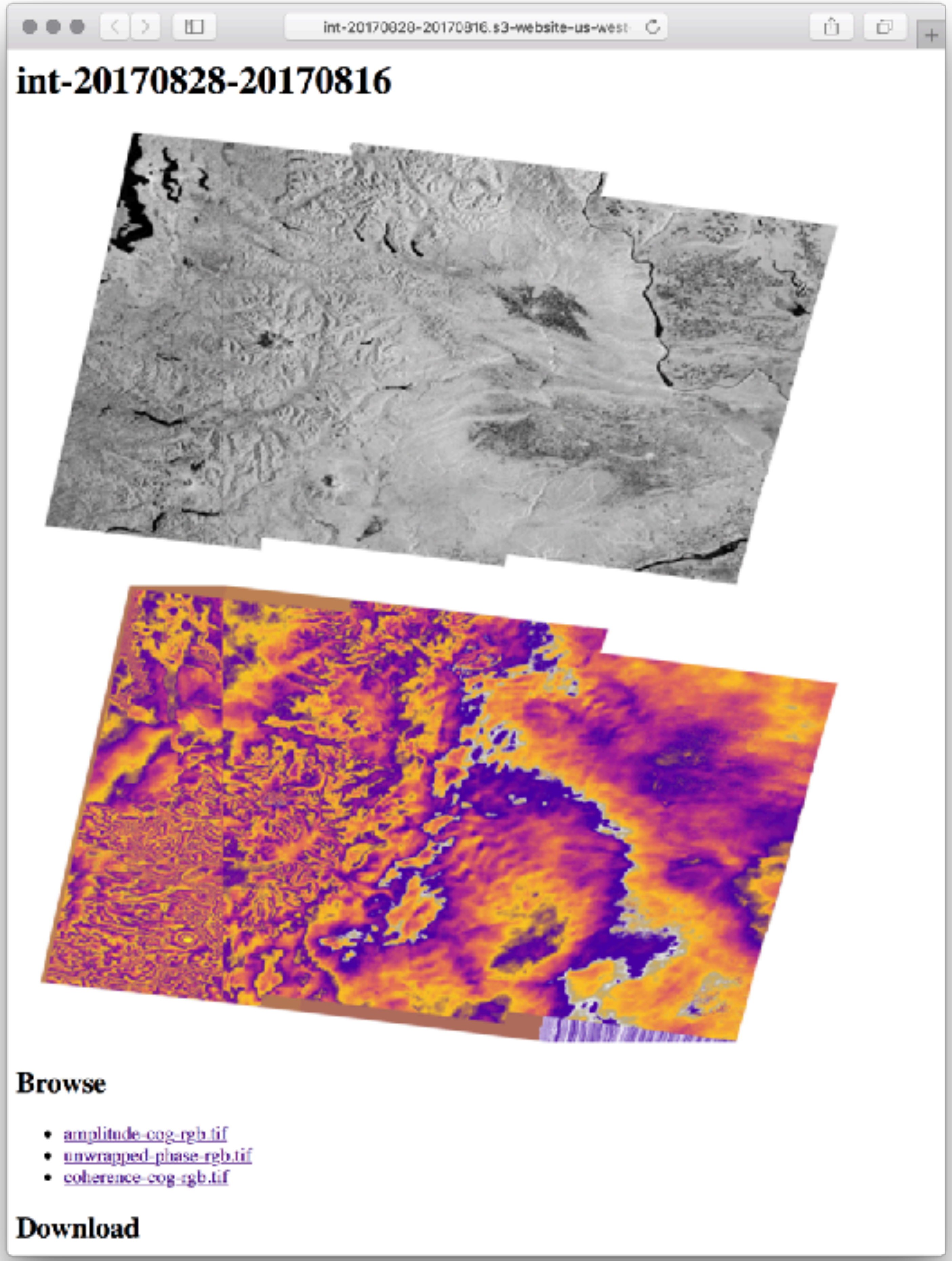
InSAR in the cloud update

User-friendly outputs!

* Slippy Maps!

http://int-20170828-20170816.s3-website-us-west-2.amazonaws.com/amplitude-cog-rgb/leaflet.html

* Shareable interactive analysis



Thanks!