ECCO Python Tutorials in the Cloud

MORE ADVANCED CALCULATIONS

Compute meridional heat transport

Compute MOC along the approximate OSNAP array from ECCO

ECCOv4 Global Volume Budget Closure

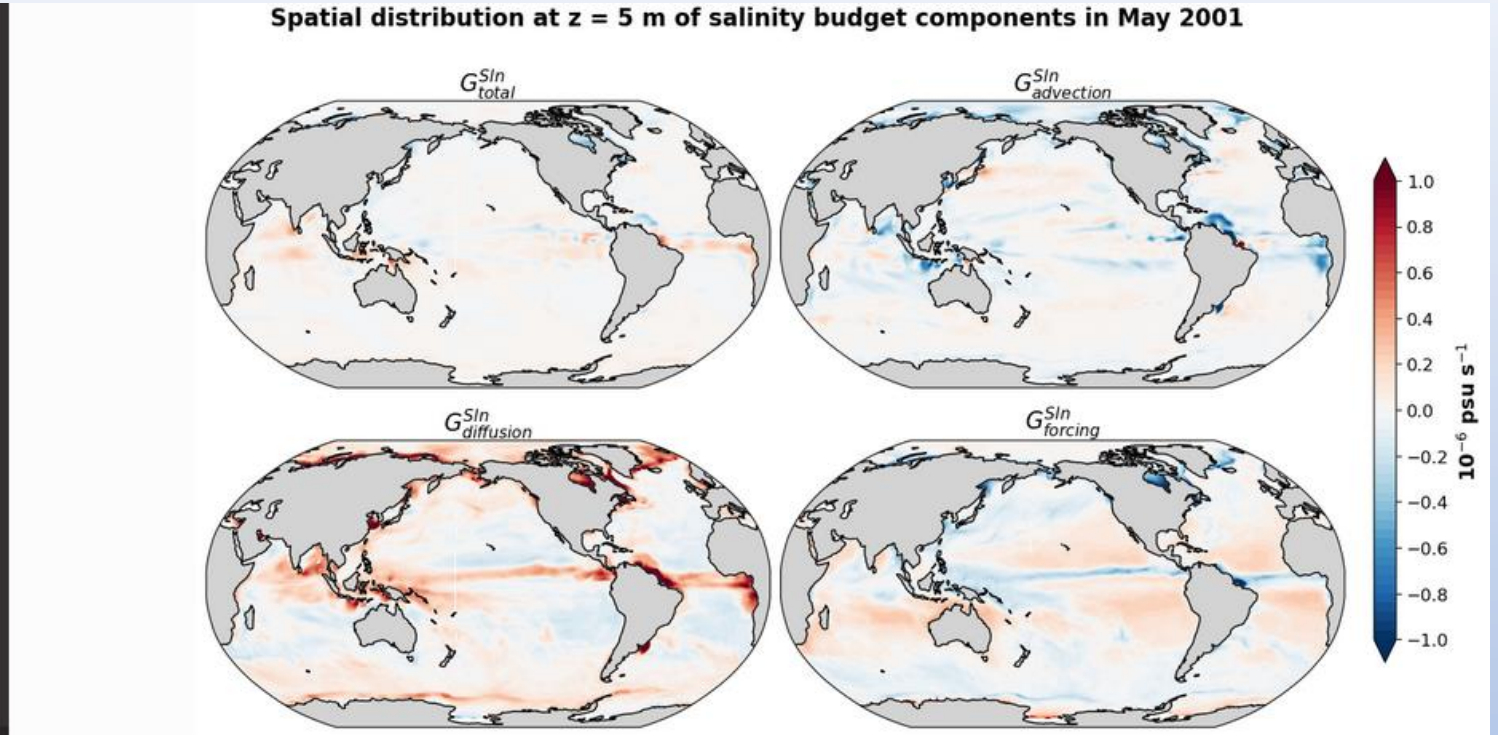
Global Heat Budget Closure

Salt, Salinity and Freshwater Budgets

Calculate ocean thermal forcing from ECCOv4r4 data, direct from PO.DAAC S3 storage

SUPPORT

Getting Help



Andrew Delman[1], Ian Fenty[2]

[1]Joint Institute For Regional Earth System Science & Engineering (JIFRESSE), University of California Los Angeles, Los Angeles, CA, USA

[2]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

Primary contact: andrewdelman@g.ucla.edu

UCLA JIFRESSE

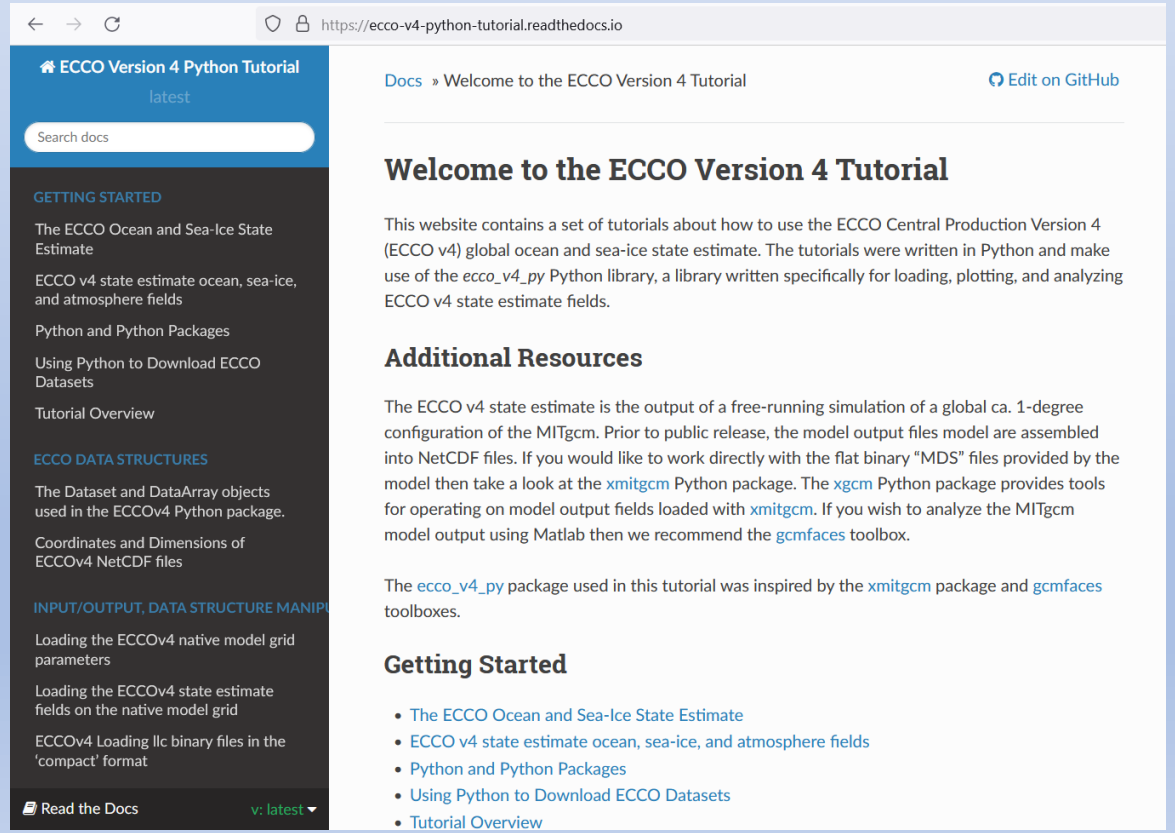
ECCO Meeting

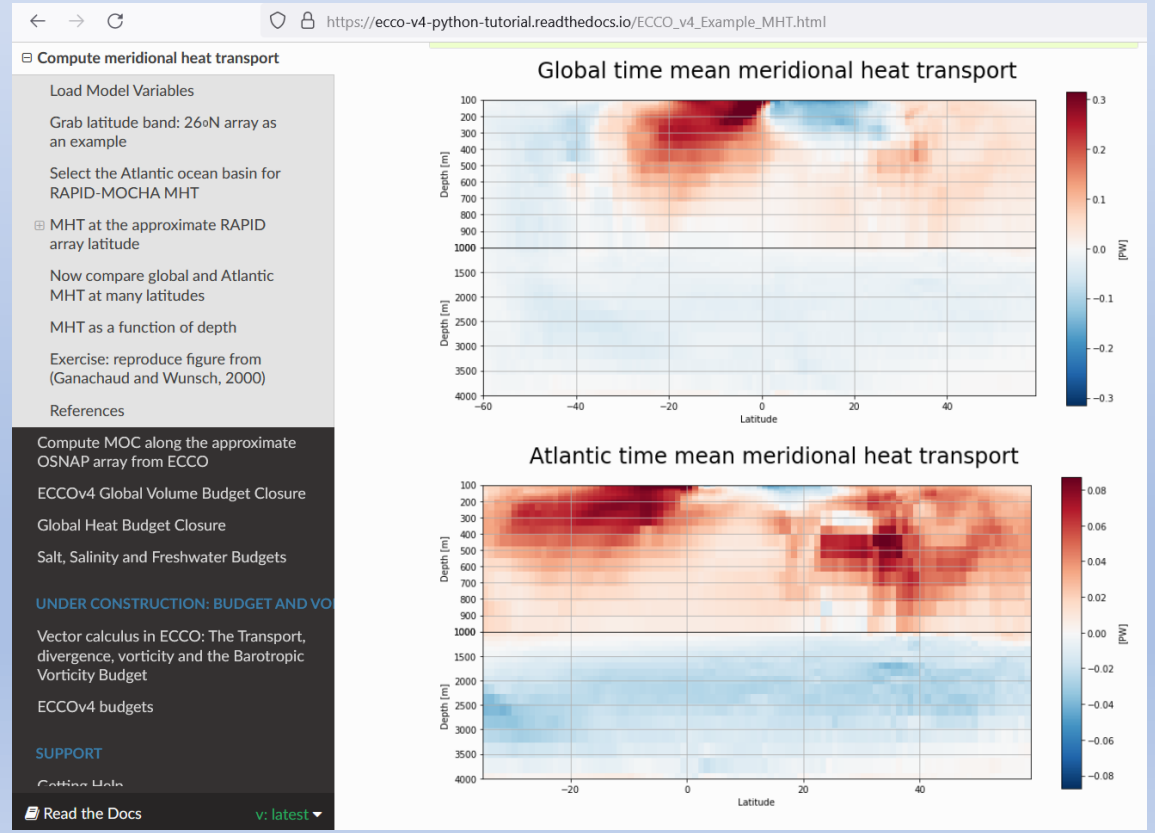
March 2024

NASA Jet Propulsion Laboratory California Institute of Technology

ECCO Python Tutorials

- Tutorials website to provide oceanographers with demos and code for ECCO analysis
  - Started by Ian Fenty a few years ago, with contributions from many others
- Comprised mostly of Jupyter notebooks that can be downloaded or git cloned to user's machine
- https://ecco-v4-python-tutorial.readthedocs.io
- The tutorials are also linked from the https://www.ecco-group.org website under Products → Analysis Tools





MHT tutorial credit: Timothy Smith

ECCO Python Tutorials: updates as of March 2024

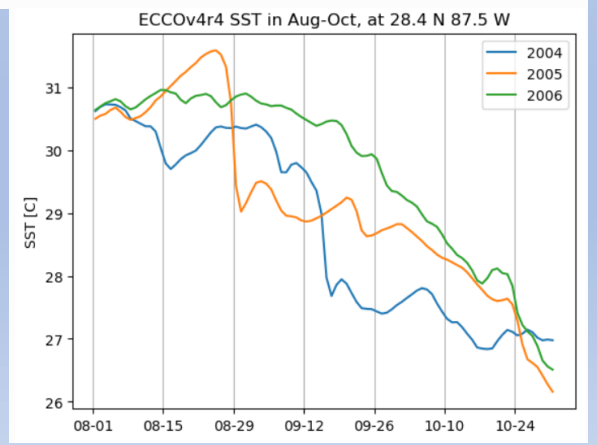
- A release last year (May 2023) updated all of the tutorials for ECCOv4 release 4, accessed through PO.DAAC
  - All tutorials have the NASA Earthdata ShortNames of the datasets needed to run them listed at the beginning, with *ecco_download.py* module assisting with downloads to user's local machine
- **Intro to PO Tutorials** illustrate fundamental PO concepts with ECCO; 3 on the website with more planned
  - Have you done something with ECCO that is fun, interesting, and/or useful to students/educators? We'll help you adapt it into a tutorial!
- Recently added Python tutorials on:
  - Downloading spatial/temporal/variable subsets of ECCO datasets using OPeNDAP
  - Computing gradients (e.g., relative vorticity, wind stress curl) on the ECCO native grid

```
[13]: ecco_podaac_download_subset(ShortName='ECCO_L4_TEMP_SALINITY_LLC0090GRID_DAILY_V4R4',\
                                  vars_to_include=['THETA'],\
                                  times_to_include=['2004-08','2004-09','2004-10',\
                                                    '2005-08','2005-09','2005-10',\
                                                    '2006-08','2006-09','2006-10'],\
                                  k_isel=[0,1,1],\
                                  tile_isel=[10,11,1],\
                                  j_isel=[28,48,1],\
                                  i_isel=[66,82,1],\
                                  subset_file_id='SST_GoM')

Download to directory C:\Users\adelman\Downloads\ECCO_V4r4_PODAAC\ECCO_L4_TEMP_SALINITY_LLC0090GRID_DA

Please wait while program searches for the granules ...
```

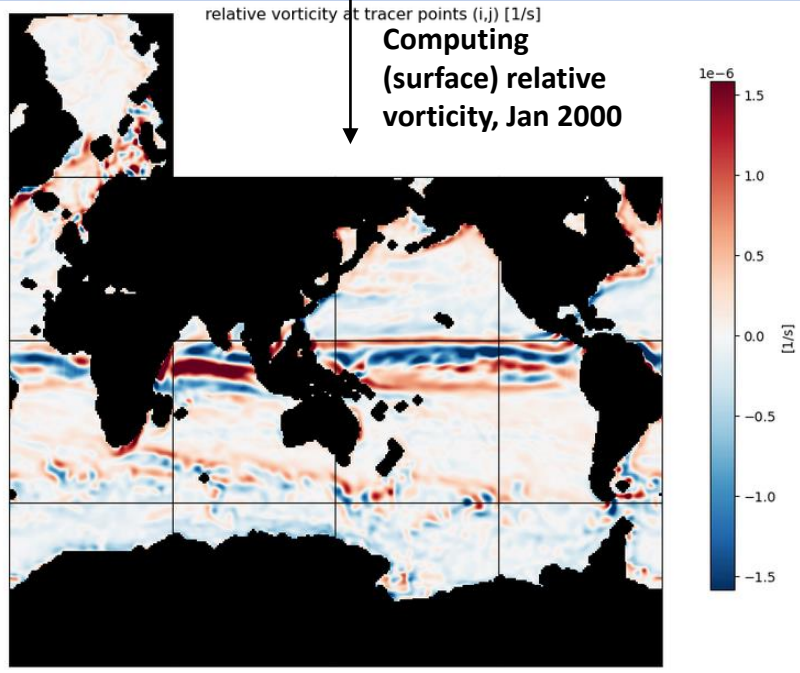
Download SST (top layer T) in Gulf of Mexico, Aug-Oct, 2004-2006




```
[57]: omega = dv_phi_dlambda - du_lambda_dphi

[58]: m=-5.8;
      title='relative vorticity at tracer points (i,j) [1/s]'
      ecco.plot_tiles(omega, cmin=-10**m, cmax=10**m,fig_size=10,
                      show_tile_labels=False,
                      rotate_to_latlon=True, layout='latlon',  cmap=colMap,
                      show_colorbar=True, cbar_label='[1/s]', show_cbar_label=True);
      plt.suptitle(title)
      plt.show()
```

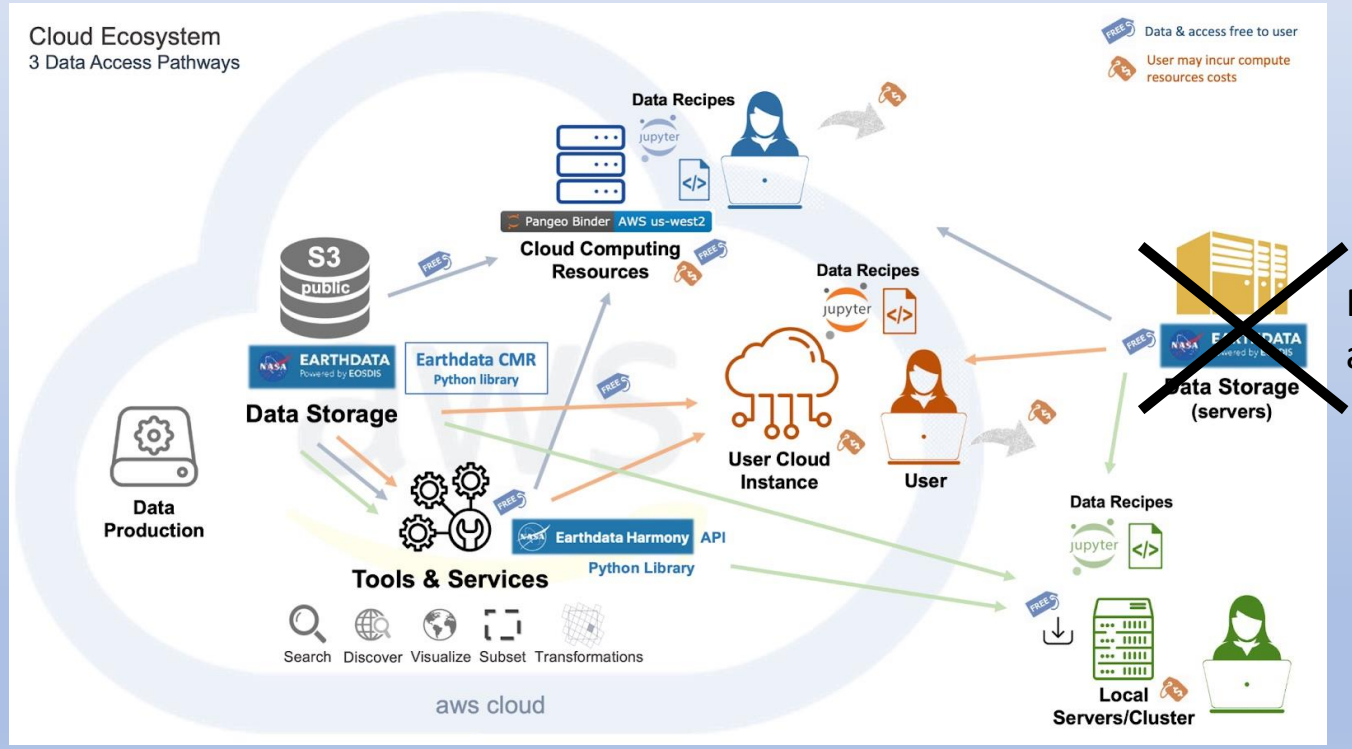
Computing (surface) relative vorticity, Jan 2000



Tutorials on the Cloud

- You may know that PO.DAAC datasets are now archived on the NASA Earthdata Cloud, hosted by Amazon Web Services (AWS)
- But perhaps the idea of working in the Cloud seems as nebulous as…a cloud.

From PO.DAAC website

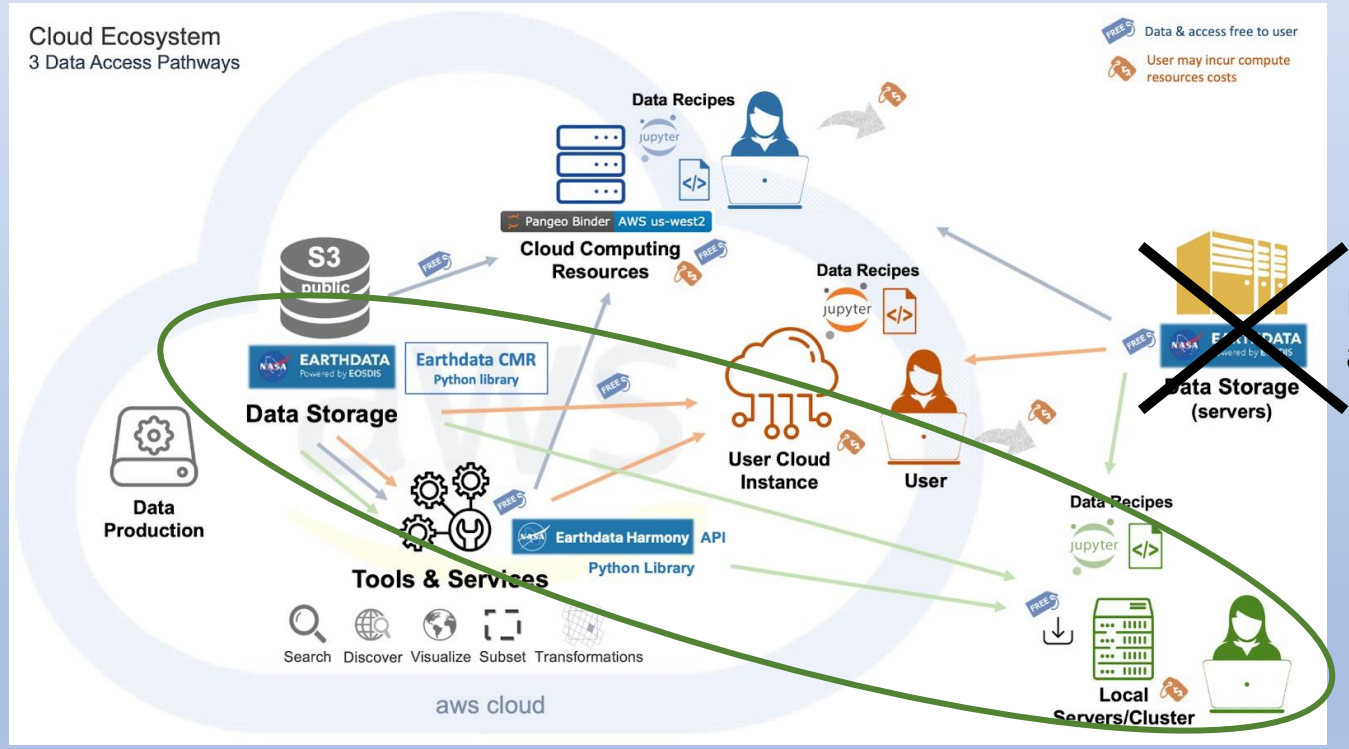


No more PO.DAAC data storage at JPL, only on AWS Cloud

Tutorials on the Cloud

- You may know that PO.DAAC datasets are now archived on the NASA Earthdata Cloud, hosted by Amazon Web Services (AWS)
- But perhaps the idea of working in the Cloud seems as nebulous as…a cloud.

From PO.DAAC website



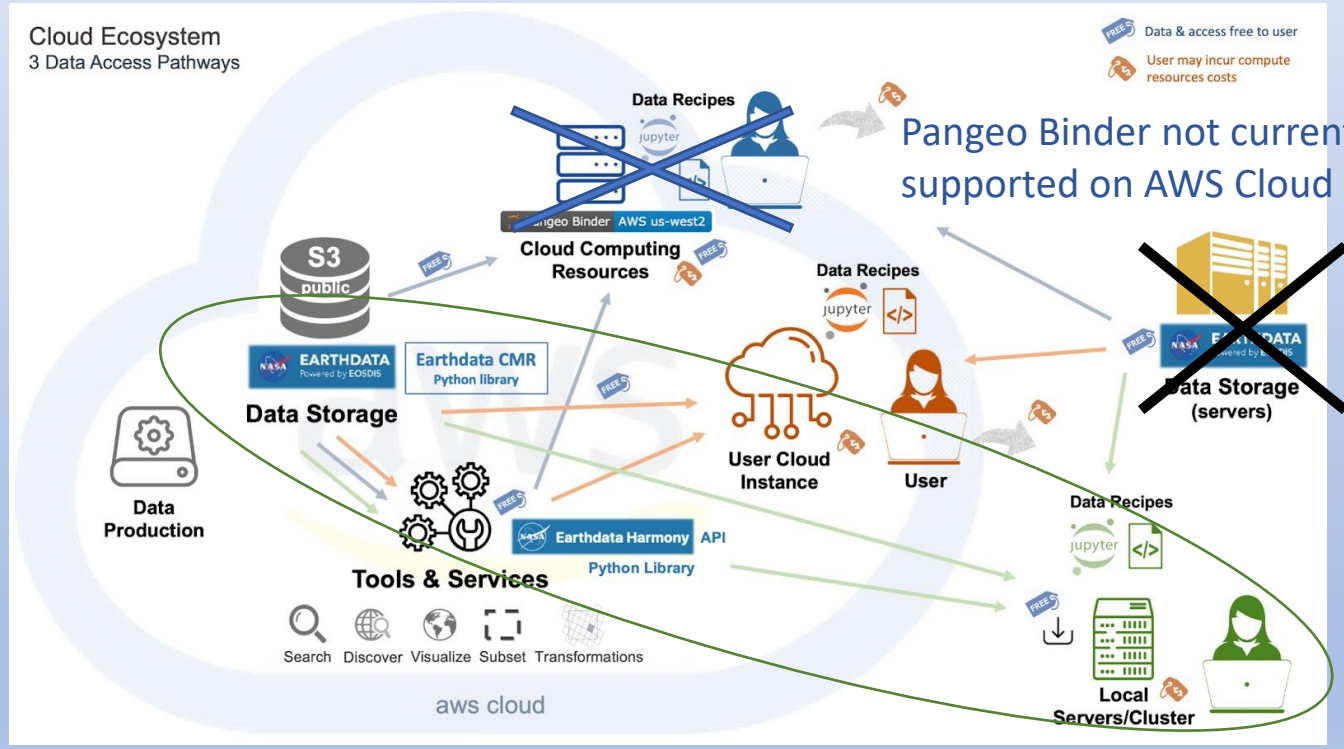
No more PO.DAAC data storage at JPL, only on AWS Cloud

User downloads data and works on local machine

Tutorials on the Cloud

- You may know that PO.DAAC datasets are now archived on the NASA Earthdata Cloud, hosted by Amazon Web Services (AWS)
- But perhaps the idea of working in the Cloud seems as nebulous as…a cloud.

From PO.DAAC website



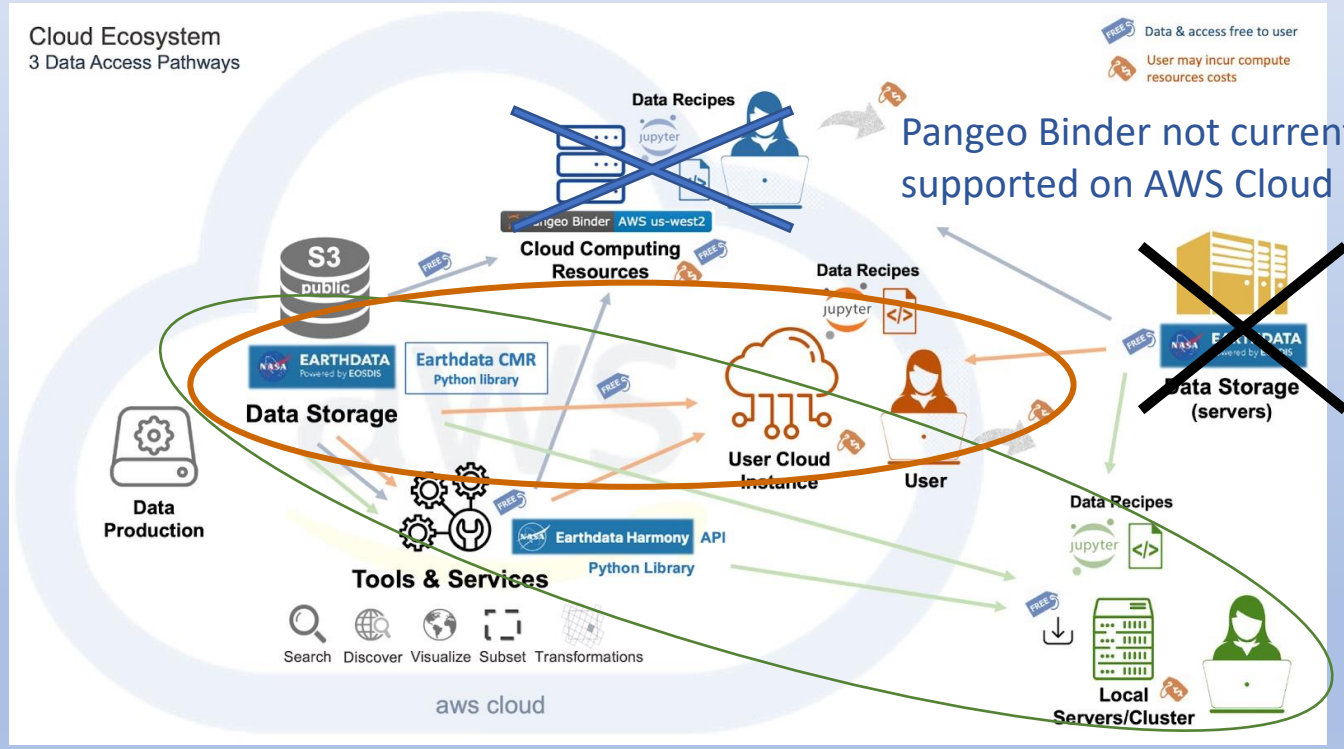
Pangeo Binder not currently supported on AWS Cloud

User downloads data and works on local machine

Tutorials on the Cloud

- You may know that PO.DAAC datasets are now archived on the NASA Earthdata Cloud, hosted by Amazon Web Services (AWS)
- But perhaps the idea of working in the Cloud seems as nebulous as…a cloud.

From PO.DAAC website



Pangeo Binder not currently supported on AWS Cloud

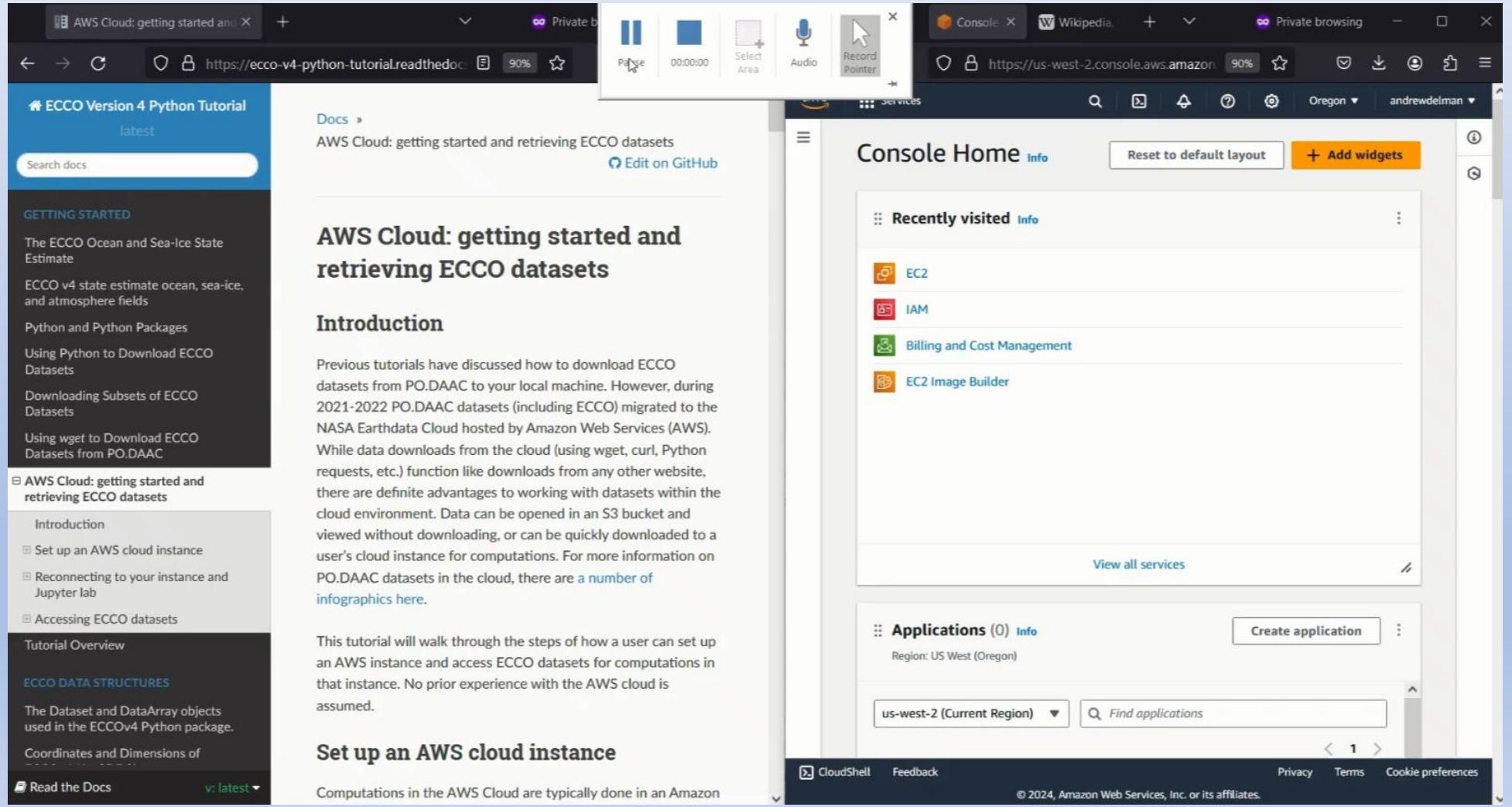
Now consider the case where a user does computations in the cloud, on their own "instance"

User downloads data and works on local machine

- Goal here is to "de-mist-ify" the cloud a bit, and to show how we are setting up the ECCO Python tutorials to run in the cloud (these can also be used to inform your own codes/workflows)

Setting up an “instance” in the AWS Cloud

- An instance is a self-contained computing environment with its own OS, memory, and storage
- A “free-tier” instance is available, though if possible it is recommended you seek project or institutional support



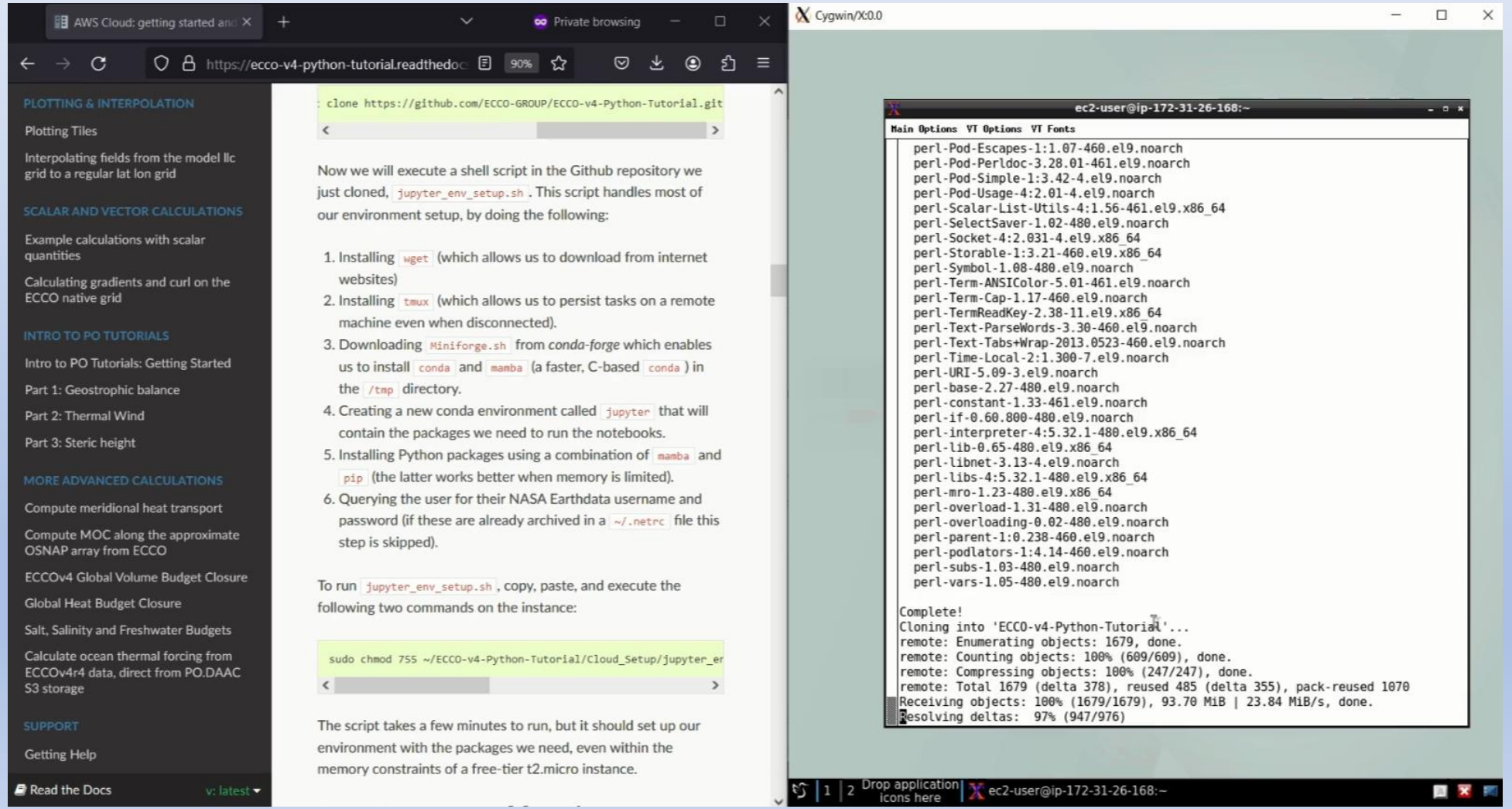
Setting up an “instance” in the AWS Cloud

- An instance is a self-contained computing environment with its own OS, memory, and storage
- A “free-tier” instance is available, though if possible it is recommended you seek project or institutional support

3 minutes later…

Setting up an “instance” in the AWS Cloud

- An instance is a self-contained computing environment with its own OS, memory, and storage
- A “free-tier” instance is available, though if possible it is recommended you seek project or institutional support



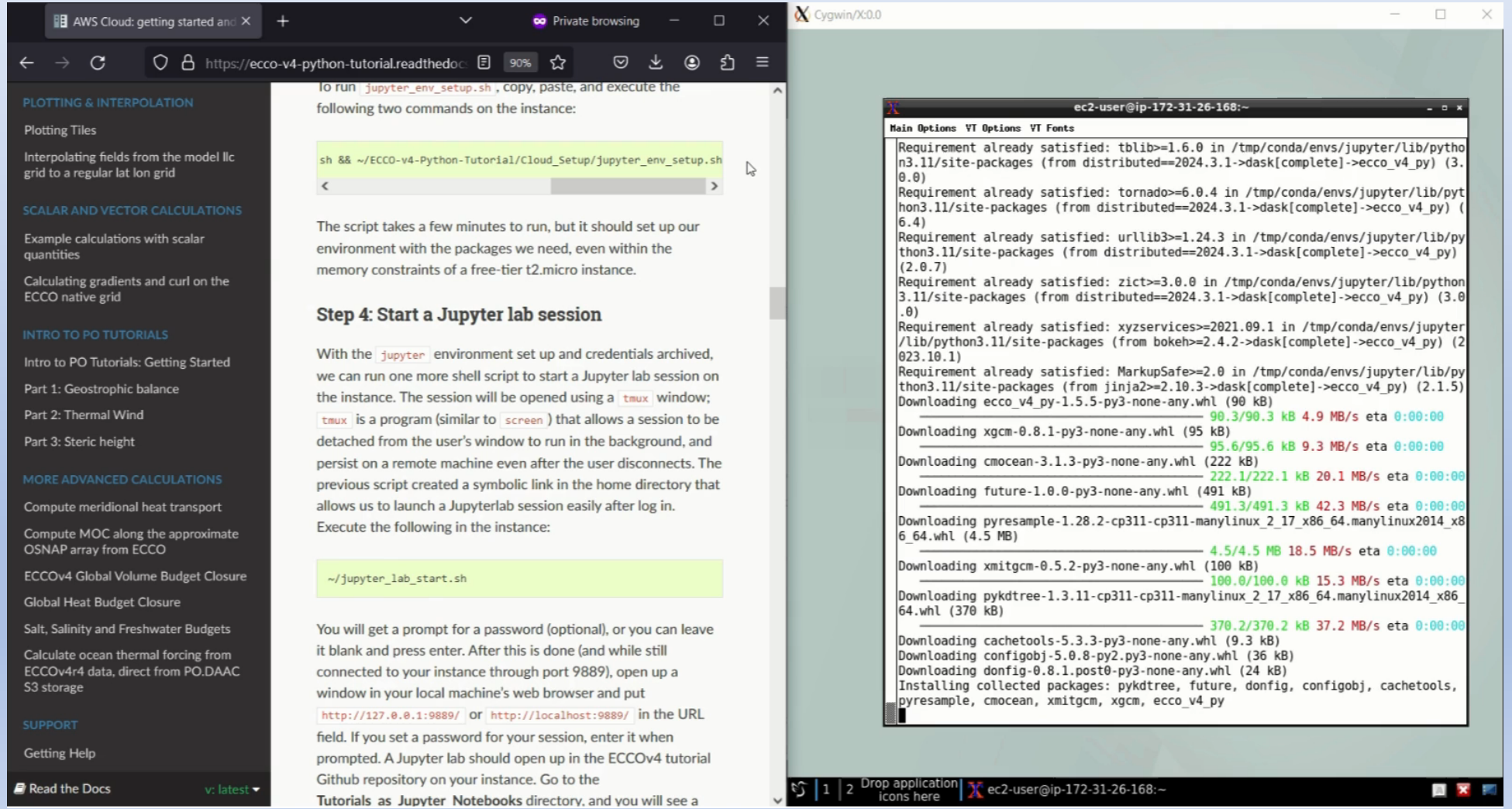
Setting up an “instance” in the AWS Cloud

- An instance is a self-contained computing environment with its own OS, memory, and storage
- A “free-tier” instance is available, though if possible it is recommended you seek project or institutional support

6 minutes later…

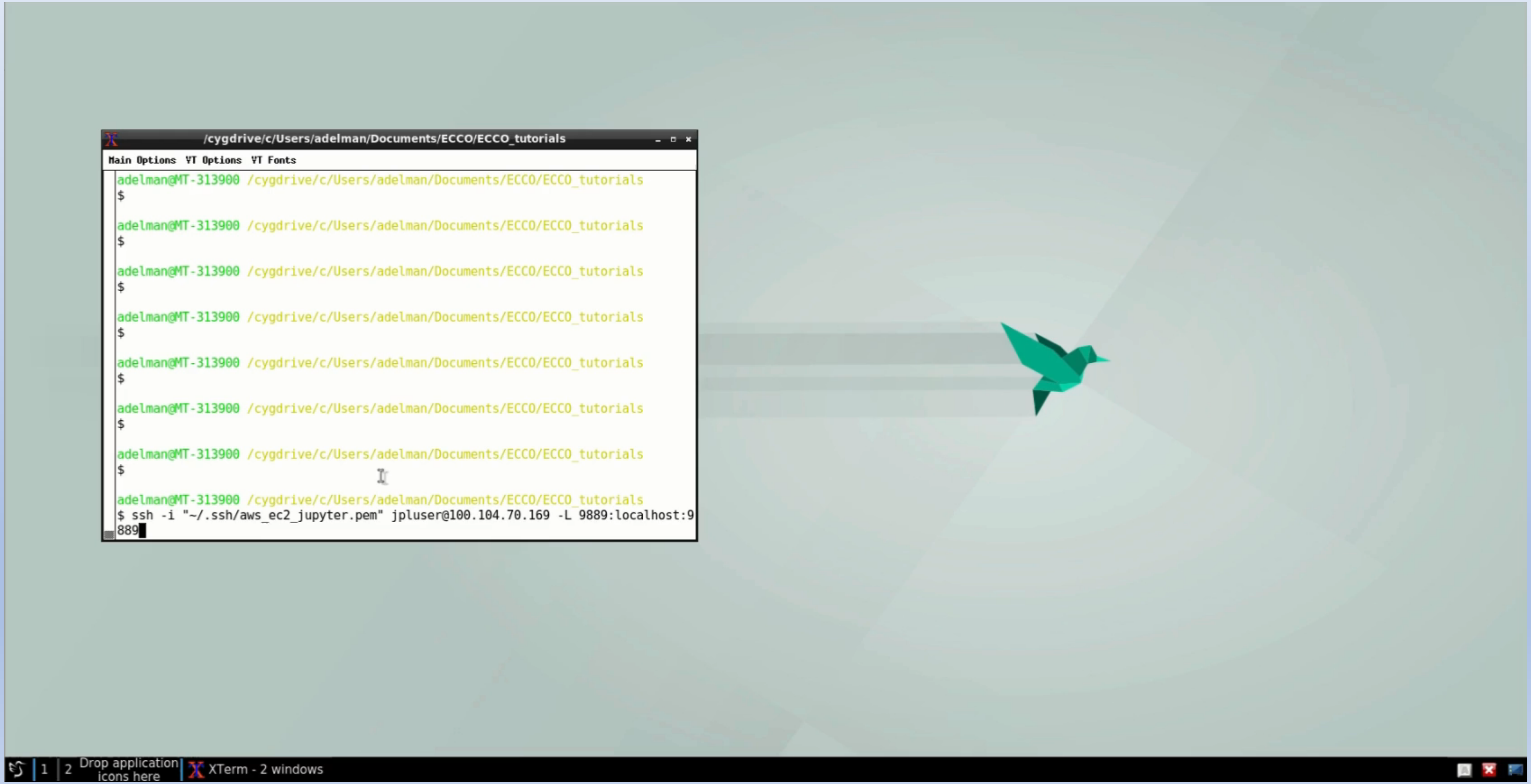
Setting up an “instance” in the AWS Cloud

- An instance is a self-contained computing environment with its own OS, memory, and storage
- A “free-tier” instance is available, though if possible it is recommended you seek project or institutional support



Running a tutorial on an AWS Cloud instance

- Sample calculation: ECCOv4r4 global mean ocean temperature (full depth) for 12 months in 2010
- Downloading the files needed to my laptop for this calculation took 3 minutes (on hotel wi-fi), versus…



Final remarks/thoughts

- A short snippet of code is being added to each of the tutorials to enable users to access ECCO output and run tutorials on the AWS Cloud
  - Powered by the *ecco_s3_retrieve.py* module (analogous to *ecco_download.py*, but for in-cloud access)

```
ShortNames_list = ["ECCO_L4_TEMP_SALINITY_LLC0090GRID_MONTHLY_V4R4",\
                   "ECCO_L4_GEOMETRY_LLC0090GRID_V4R4"]

## download files to instance (only if download size is less than available storage threshold from max_avail_frac)
retrieved_files = ecco_podaac_s3_get_diskaware(ShortNames=ShortNames_list,\
                                               StartDate="2010-01",EndDate="2010-12",\
                                               max_avail_frac=0.5,n_workers=2)

## open files in workspace
ds = xr.open_mfdataset(retrieved_files[ShortNames_list[0]],\
                       data_vars='minimal',coords='minimal',\
                       compat='override',\
                       chunks={'time':1,'k':10,'tile':13,'j':90,'i':90})

ds_grid = xr.open_mfdataset(retrieved_files[ShortNames_list[1]],chunks={'k':10,'tile':13,'j':90,'i':90})
```

- AWS Cloud instances can be run for free (for 12 months) but to run code successfully, it needs to be designed in a way to minimize memory footprint (e.g., using *dask* chunking, “lazy” computations)

- **Here’s the URL for the tutorials again: https://ecco-v4-python-tutorial.readthedocs.io**
  - Can also be accessed through the ECCO Group website under Products -> Analysis Tools

- Questions or ideas (including tutorial ideas)? Please contact me: andrewdelman@g.ucla.edu.