

**THERE IS NO CLOUD**



**IT'S JUST  
SOMEONE ELSE'S COMPUTER**

# What is the Cloud? (a user's perspective)

- A network of remote servers accessible via the internet
- **Store/Manage/Process** data
- Scalable, collaborative, and real-time data access
- Operates on a pay-as-you-go model for cost efficiency – some free options...

# Amazon Web Services

- Comprehensive cloud platform
- Computing, storage, database
- Popular for hosting web applications and big data analytics.

# Digital Ocean

- Simpler than AWS
- Computing, storage, database
- Popular for hosting web applications and big data analytics.



# Console Home

Reset to default layout

+ Add widgets

## Recently visited

Billing and Cost Management

S3

Support

RDS

CloudShell

AWS Health Dashboard

Resource Access Manager

IAM

AWS App Runner

EC2

CloudWatch

AWS Organizations

View all services

## Cost and usage

Current month costs

\$60.49

↓ 9% compared to last month for same period

Forecasted month end costs

\$82.75

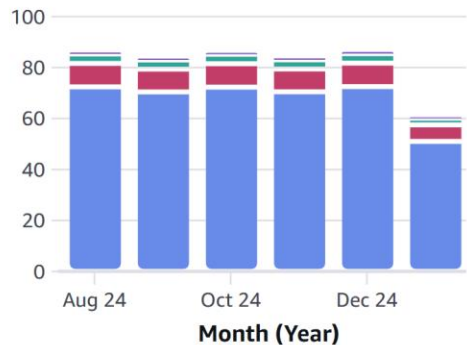
↓ 4% compared to last month's total costs

Savings opportunities

\$32.07/mo

Reduce spend by 37% compared to last month

Cost (\$)



Relational Database Service  
Virtual Private Cloud  
S3  
CloudShell  
Tax

Go to Billing and Cost Management

## Welcome to AWS

Getting started with AWS

## AWS Health

Open issues

## Applications (0)

Region: Canada (Central)

Create application

← Back to Droplets



hydro-debian-1

in first-project / 1 GB Memory / 1 AMD vCPU / 25 GB Disk + 100 GB / TOR1 - Debian 11 x64

Upsize Droplet

ON

Reserved IP: [Enable now](#)

Console:  ?

Graphs

Access

Power

Volumes

Resize

Networking

Backups

Snapshots

Kernel

History

Destroy

Tags

Recovery

We value your feedback. Tell us about your experience with the metrics agent so far.

Share feedback



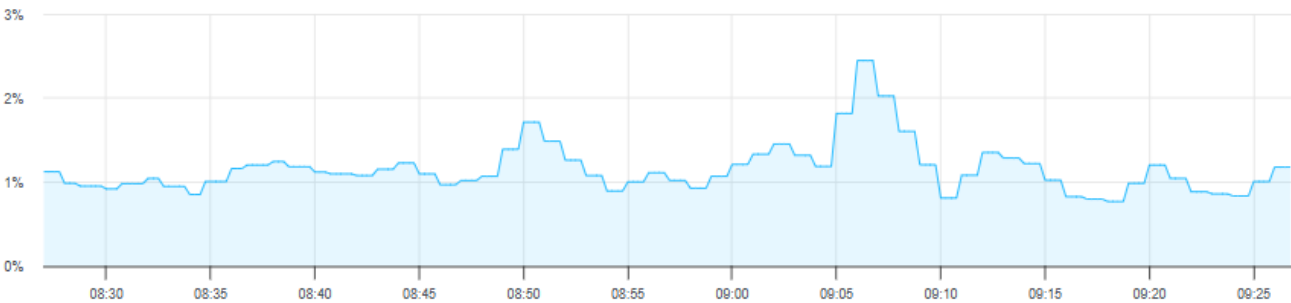
Select period

1 hour

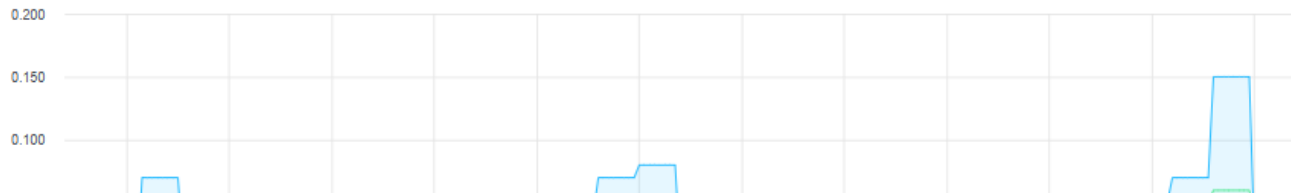


Learn 

CPU %



Load (1/5/15)







+ Code + Text

✓  
0s `from google.colab.patches import cv2_imshow`

For example, here we download and display an image:

✓  
3s

```
[5] !curl -o logo.png https://bevingtona.github.io/images/2019-02-12_0029_circle.jpg
import cv2
img = cv2.imread('logo.png', cv2.IMREAD_UNCHANGED)
cv2_imshow(img)
```



% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload Upload	Total	Spent	Left	Speed
100 1093k	100 1093k	0 0	4439k 0	--:--:--	--:--:--	--:--:--	4426k



✓ 3s completed at 9:40 AM





# Choose the Colab plan that's right for you

Whether you're a student, a hobbyist, or a ML researcher, Colab has you covered

Colab is always free of charge to use, but as your computing needs grow there are paid options to meet them.

[Restrictions apply. learn more here](#)

## Pay As You Go

CA\$13.99 for 100 Compute Units

CA\$67.20 for 500 Compute Units

You currently have 0 compute units.  
Compute units expire after 90 days.  
Purchase more as you need them.

- ✓ No subscription required.  
Only pay for what you use.
- ✓ Faster GPUs  
Upgrade to more powerful GPUs.

Recommended

## Colab Pro

CA\$13.99 per month

- ✓ 100 compute units per month  
Compute units expire after 90 days.  
Purchase more as you need them.
- ✓ Faster GPUs  
Upgrade to more powerful GPUs.
- ✓ More memory  
Access our highest memory machines.
- ✓ Terminal

## Colab Pro+

CA\$67.20 per month

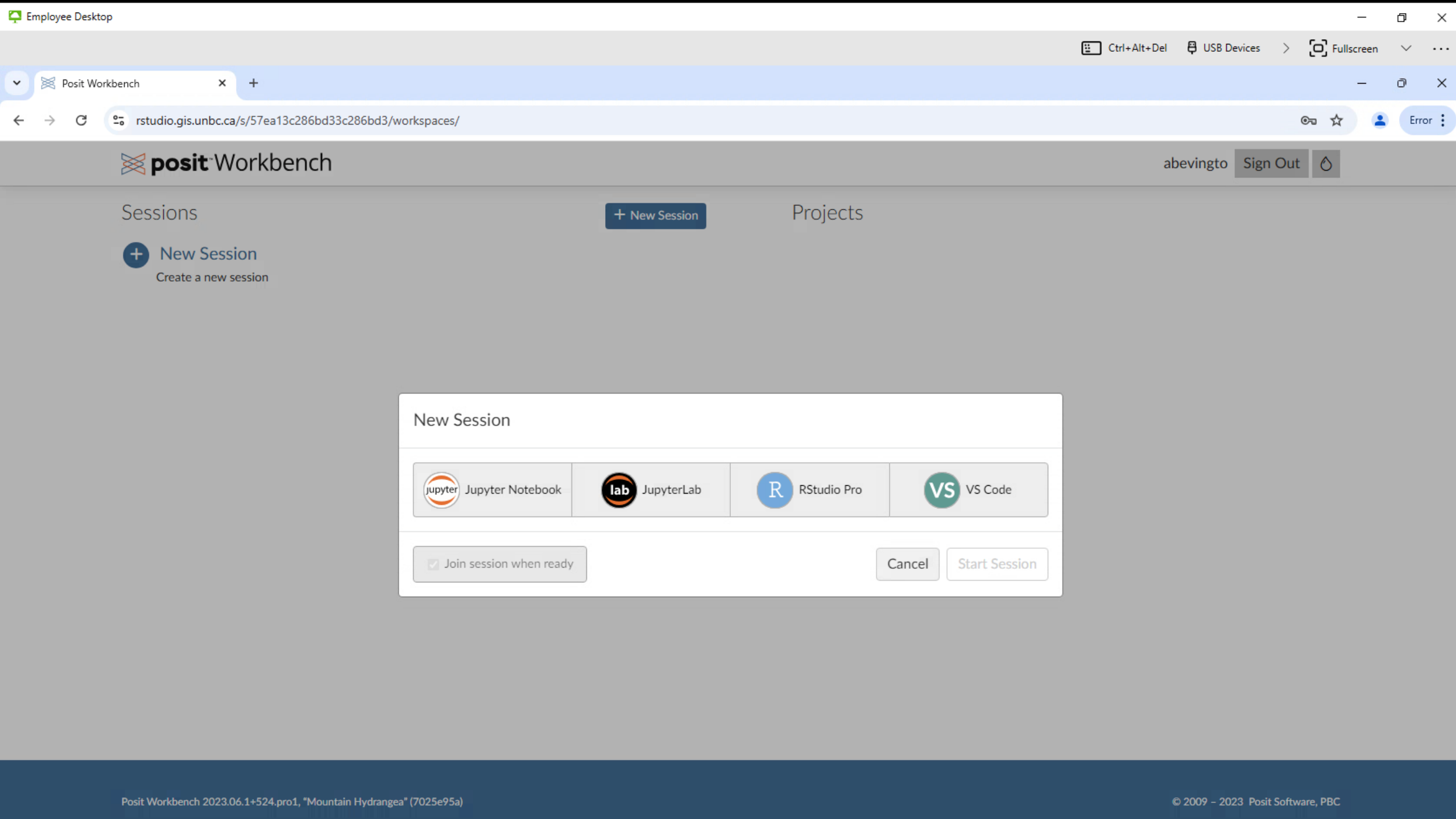
All of the benefits of Pro, plus:

- ✓ An additional 400 compute units for a total of 500 per month.  
Compute units expire after 90 days.  
Purchase more as you need them.
- ✓ Faster GPUs  
Priority access to upgrade to more powerful premium GPUs.
- ✓ Background execution

## Colab Enterprise

Pay for what you use

- ✓ Integrated  
Tightly integrated with Google Cloud services like BigQuery and Vertex AI.
- ✓ Enterprise notebook storage  
Replace your usage of Google Drive notebooks with GCP notebooks, stored and shared within your cloud console.
- ✓ Productive  
Generative AI powered code




Sessions


+ New Session


Projects


+ New Session  
Create a new session

New Session

 Jupyter Notebook

 JupyterLab

 RStudio Pro

 VS Code

☒ Join session when ready

Cancel Start Session

RStudio Pro

rstudio.gis.unbc.ca/s/996a0cfc78a31c0030824/?launcher=1

R

File Edit Code View Plots Session Build Debug Profile Tools Help

abevingto Sessions · lab3 Project: (None) R 4.4.2

Go to file/function Addins

Untitled1\*

Source on Save Run Source

```
1 install.packages("rgee")
2 library(rgee)
3
```

3:1 (Top Level) R Script

Console Terminal Workbench Jobs

R 4.4.2 · ~/

```
** testing if installed package keeps a record of temporary installation path
* DONE (reticulate)
* installing *source* package 'rgee' ...
** package 'rgee' successfully unpacked and MD5 sums checked
** using staged installation
** R
** inst
** byte-compile and prepare package for lazy loading
```

Environment History Connections Tutorial

Import Dataset 263 MiB

R Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

Folder Blank File Upload Delete Rename

Home > K

	Name	Size	Modified
↑	..		
🕒	.Rhistory	14.3 KB	Jan 16, 2025, 3:01 PM
📁	Contacts		
📁	Desktop		
📁	Documents		
📁	Downloads		
📁	Favorites		
📁	Geo-SAM		
📄	Geo-SAM.zip	148.9 MB	Jan 9, 2025, 12:33 PM

File

Edit

View

Run

Kernel

Git

Tabs

Settings

Help

+

+

↑

↺

✎

Filter files by name

/

Name	Last Modified
Research	a day ago
K	7 days ago
Downloads	7 days ago
Documents	7 days ago
Pictures	9 days ago
L	10 days ago
R	13 days ago
Untitled.ipynb	seconds ago

Untitled.ipynb

+

✂

📄

📄

▶

■

↺

▶▶

Code

⌵

🕒

git

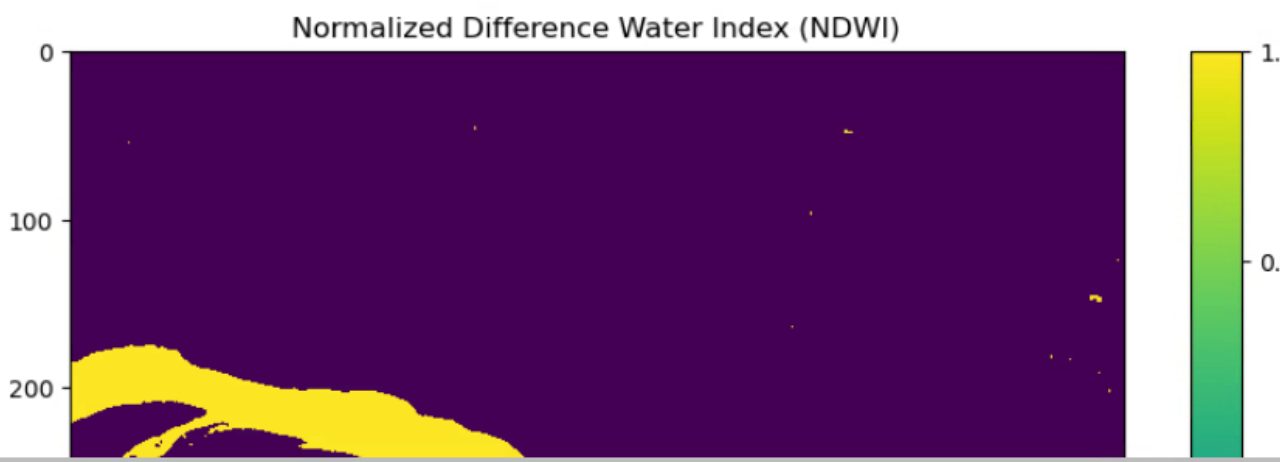
GIS Base Environment

```
•[1]: import rasterio
      from rasterio.plot import show
      import matplotlib.pyplot as plt

[7]: ndwi_file = "K/Lab2_SiteC_sentinel-2-l2a_2021_2024_6_10_0_10/20210627191911024_water.tif"

      # Read the GeoTIFF using rasterio
      with rasterio.open(ndwi_file) as src:
          ndwi = src.read(1)

      # Display the NDWI raster using matplotlib
      plt.figure(figsize=(10, 8))
      plt.imshow(ndwi, cmap='viridis', interpolation='none') # Use a colormap like 'viridis'
      plt.colorbar(label="NDWI Value") # Add a colorbar
      plt.title("Normalized Difference Water Index (NDWI)")
      plt.xlabel("X Coordinate")
      plt.ylabel("Y Coordinate")
      plt.show()
```



Simple ☐ 0 1

GIS Base Environment | Idle

Mode: Command Ln 1, Col 26 English (United States) Untitled.ipynb 1



# Now you have code that you want running in the cloud

1. Spin up a server (often Linux!)
  2. Install your language and libraries
  3. Run your code on demand or on a schedule
    1. Cronjobs
    2. AWS Lambda
    3. Google Cloud Scheduler
- **Problems**
    - Scaling
    - Environment Drift
    - Portability
    - Management Overhead
  - **Solution**
    - Docker: Package your code and environment into a portable container
    - Kubernetes: Manage containers across servers



# You still need the data to be accessible from your cloud processing

- **Spatial Databases**

- Designed to store and query spatial data like maps and coordinates.
- Examples include PostGIS, SpatiaLite, and Oracle Spatial.
- Essential for GIS, remote sensing, and location-based services.

- **STAC (SpatioTemporal Asset Catalog)**

- A specification for organizing geospatial assets.
- Facilitates search and discovery of satellite imagery and maps.
- Widely used for indexing and sharing remote sensing datasets.

# Hosting geospatial data services (OGC Compatible)

- **WMS** (Web Map Service): Serves georeferenced images. Cannot be directly downloaded, edited or queried. Some customization possible from user.
- **WMTS** (Web Map Tile Service): Similar to WMS but faster and less flexible. Tiles are fixed in style and resolution.
- **Vector tiles**: Serves geospatial data as vectors instead of raster images for visualization. Rendered client-side, allowing for dynamic styling and interactivity. Very fast!
- **Web Feature Service (WFS)**: Raw vector data that allows querying and downloading of vector data. Think `bcddata` package.
- ...

# Example 1: Microsoft Planetary Computer

- Host large STAC catalogs
- Allow computation on their servers with python via Dask
- Free

```
import numpy as np
import xarray as xr

import rasterio.features
import stackstac
import pystac_client
import planetary_computer

import xrspatial.multispectral as ms

from dask_gateway import GatewayCluster
```

## Create a Dask cluster

We're going to process a large amount of data. To cut down on the execution time, we can use Dask. See [Scale With Dask](#) for more on using Dask.

```
cluster = GatewayCluster() # Creates the Dask Scheduler.

client = cluster.get_client()

cluster.adapt(minimum=4, maximum=24)
print(cluster.dashboard_link)
```

<https://pcc-staging.westeurope.cloudapp.azure.com/comput>

# Example 2: Google Earth Engine

- Host large catalogs (only accessible from GEE)
- Allow computation on their servers using GEE functions
- Options
  - JS API and online interface
  - Python API
  - Python `geemap`
  - R `rgee`
- Free ...with options to buy storage

