NASA Earth Data Cloud

Mark Carroll Data Science Group Lead NASA Goddard Space Flight Center



"Managed" commercial cloud for Earth Observations on AWS

- Improves the efficiency of NASA's data systems operations maintaining free/open data policy
- Initially architected for NASA DAACs (applications and mission data ingest, archive, distribution)
- Operational July 2019. First in, Common Metadata Repository (CMR) Application, Earthdata Search Application, 6 PB Sentinel-1 mission data.
- 23 PB of mission data in EDC as of May 2022 with plan to onboard 50/75 top datasets by Q4 CY 2022, establishing initial "data lake".

Earthdata Cloud (EDC)



NASA Earthdata Cloud (EDC) – http://earthdata.nasa.gov

What is the Earthdata Cloud Platform?

Earthdata Cloud Platform (NGAP - NASA General Application Platform) is a multi-account, Infrastructure-as-a-Service (IaaS) cloud platform operating on Amazon Web Services (AWS) under a single Earth System Data and Information System (ESDIS) owned top level "payer account", providing shared cloud services and controls to EOSDIS. ESDIS Managed / EED Operated.



Maximizes Autonomy

Be a platform, not a gate; foster experimentation/innovation and support production needs of application owners



Maximizes Flexibility

Provide projects the freedom to implement solutions that fit their problem domains



Shared Services & Controls

Platform manages common shared services & controls to reduce duplication, system complexity, and cost across EOSDIS



Cloud Platform Infrastructure

Earthdata Cloud Cookbook

Cheatsheets & Guides



https://nasa-openscapes.github.io/earthdata-cloud-cookbook/cheatsheet.html#tools-services-roadmap

openscapes

0,000000

Last update: 8 July 2022

Tools & Services Roadmap

demo

 Practical guide for learning and selecting the right tool or service for a given use case

Earthdata Cloud Cookbook

Supporting NASA Earth science research teams' migration to the cloud

- Curated collection of tutorials have been iterated on and adapted following versions and feedback from interactive training events
- Focus on the common steps across DAACs/users
- For self-paced learning
- Links back to underlying GitHub repo
- Under active, open development

A place to learn, share, and experiment with NASA Earthdata on the Cloud. We know this has a lot of moving parts, and we are iterating as we go, and welcome feedback and contributions.



NASA

Earthdata

Cloud

Cookbook

Q

Welcome

How To's

Tutorials

Our Cookbook

In Development

Troubleshooting

Contributing

Cheatsheets & Guides

>

NASA Earthdata Cloud Cookbook

Supporting NASA Earth science research teams' migration to the cloud

AUTHOR NASA Openscapes Team

PUBLISHED May 1, 2022 openscapes

Welcome

Welcome to the NASA Openscapes EarthData Cloud Cookbook!

This Cookbook is learning-oriented to support scientific researchers using NASA Earthdata from Distributed Active Archive Centers (DAACs) as they migrate their workflows to the cloud. It has resources to learn how to work with NASA Earthdata on the cloud, as well as well as documentation for how to contribute to these resources. It is under active, open development.

The Cloud Paradigm

NASA Distributed Active Archive Centers (DAACs) are in the process of moving their data holdings to the cloud. In the new paradigm, data storage (orange in the illustration) will migrate to the cloud (green) and DAACprovided tools and services built on top of the data are co-located in the Earthdata Cloud.

https://nasa-openscapes.github.io/earthdata-cloud-cookbook/

Tools and access development

earthaccess: simplifying access





Overview

TL;DR: earthdata is a Python package to search, preview and access NASA datasets (on-prem or in the cloud) with a few lines of code.

from earthdata import Auth, DataGranules, Store

first we authenticate with NASA EDL
auth = Auth().login(strategy="netrc")

Then we build a Query with spatiotemporal parameters
GranuleQuery = DataGranules().concept_id("C1575731655-LPDAAC_ECS").bounding_box(-134.7,58.9,-133.9,59.2)

We get the metadata records from CMR
granules = GranuleQuery.get()

Now it{s time to download (or open) our data granules list with get()
files = Store(auth).get(granules, local_path='./data')

Now to the important science!

https://nasa-openscapes.github.io/

How can users interact with the data?

Work with the data in the cloud

Jupyter Notebooks

► EC2

Amazon tools

Download the data locally

Why does all of this matter?

- Accelerate science
- Work with curated data sets
- Facilitate collaboration
- Expand the use of data
 - No downloads required
 - Not limited by local storage and compute
- Upcoming missions (NISAR, SWOT, SBG, etc.) have greatly increased data volumes
- Greater understanding likely comes from combining data from multiple missions

Thanks!

Questions?

- http://earthdata.nasa.gov
- https://nasa-openscapes.github.io/

