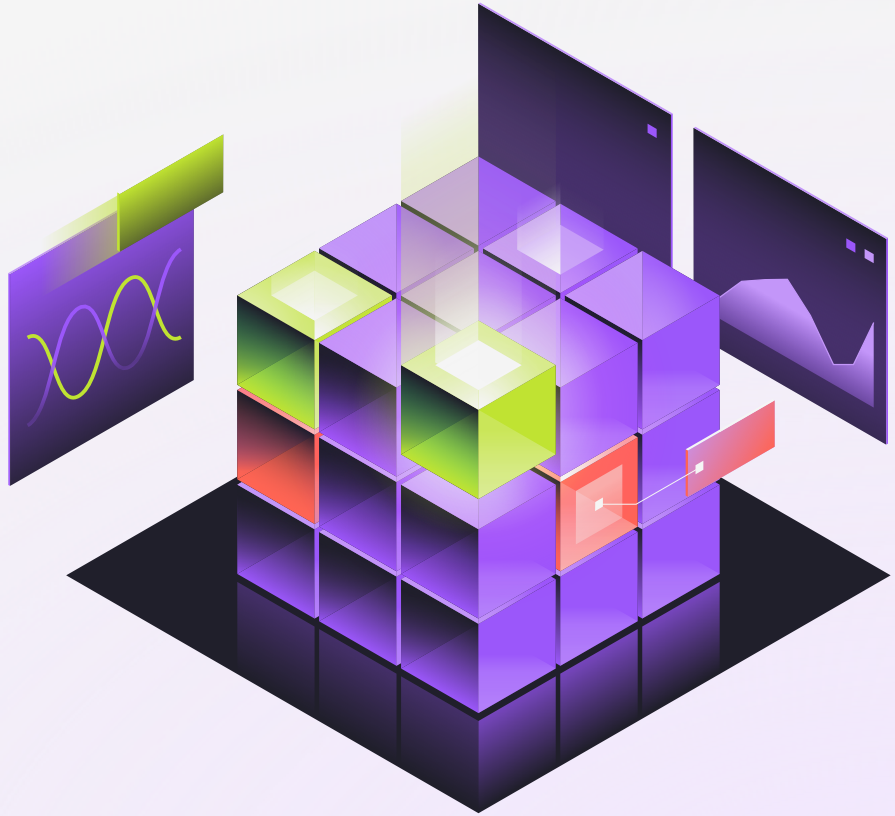
earthmover

icechunk: marrying performance & rigor for the planet



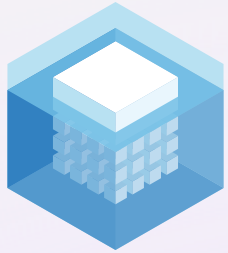
Deepak Cherian, Ph.D.
Forward Engineer
deepak@earthmover.io

PROPL 2026

THE POINT

we needed a new storage layer, icechunk is that layer

- *icechunk* is a new-ish ndarray storage engine.
- *cloud-native* & *open source*
- Smashes together a number of good ideas from earth system data systems, & databases.
- increasing engineering rigor in the earth data ecosystem
- a great foundation for computational systems
- earthmover is building a cloud platform for planetary-scale data federation



icechunk

BACKGROUND

about earthmover

Mission

> *To empower people to use scientific data to solve humanity's greatest challenges.*

- VC-backed **Public Benefit Corporation** founded in 2022
- Founders
  - Dr. Ryan Abernathey (previously: Columbia, MIT)
  - Dr. Joe Hamman (previously CarbonPlan, NCAR, UW)

Customers

RWE · Brightband · ELIANN · Sylvera · zeus·AI · The Nature Conservancy · EPRI Electric Power Research Institute · Beyond Weather · Advanced Research + Invention Agency ARIA

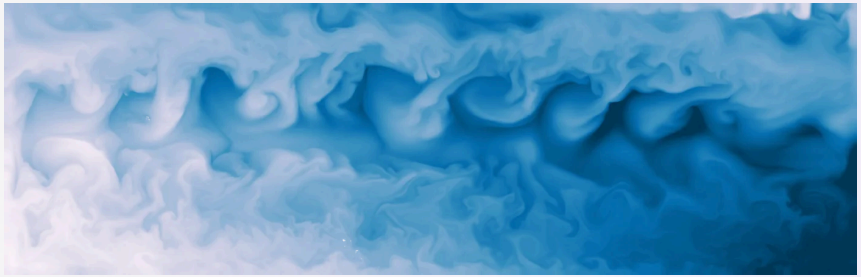
Use cases: Environmental Modeling, Energy, Commodities.

Growing public sector business (ARIA, NASA, NOAA, UK Met Office, etc.)

BACKGROUND

about me

Deepak Cherian



hats i wear

Scientist (2010-2023):

- most recently Oceanographer @ NCAR (National Center for Atmospheric Research)
- TB-scale ocean modeling (regional/global)
- Observations (ocean turbulence sensors)

Programmer (2017-):

- Open-source scientific python / Pangeo project
- Xarray : nd-array analytics powered by metadata
  - `temp.mean(dim="time")` instead of `temp.mean(axis=0)`

WHY

the (earth science) scientific data/compute landscape needed an upgrade

WHY

the storage layer is fuzzy

ECMWF weather forecasts

- streaming updates: GRIB files delivered to a cloud bucket
- archive: ~ 1 year wait to pull the archive off *tape system*.

Climate model archive:

- Folders with O(100k) GB-sized files
- privileges large institutions with large resources (e.g. NCAR)

Every archive is bespoke.

WHY

the pile of files model severely lacks rigor

- you don’t know what you have until you scan all the files.
- data may be missing or corrupt. hard to interrogate
- updates are "invisible", practically speaking
- users are not isolated from physical data layout.
  - filenames and data partitioning between files are a public interface.
  - example: `20190101090000-JPL-L4_GHRSST-SSTfnd-MUR-GLOB-v02.0-fv04.1`

> *With a sufficient number of users of an API, all observable behaviors of your system will be depended on by somebody.*
>
> — **Hyrum's Law**

WHY

the pile of files is ... not good.

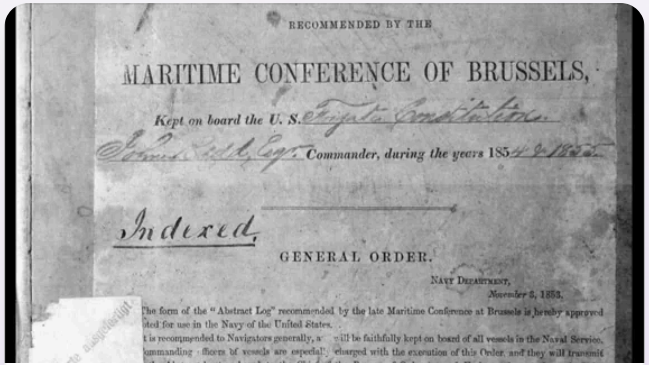
The scientific "workflow": figure out how to scan the files and assemble an n-dimensional data cube that makes sense for the problem at hand.

Incredible duplication of (hard) work.

Conceptual mismatch between compute & storage models.

WHY

how did we get here?


RECOMMENDED BY THE

MARITIME CONFERENCE OF BRUSSELS,

Kept on board the U. S. [illegible] *Commander, during the years* 185[illegible]

Indexed

GENERAL ORDER.

NAVY DEPARTMENT,
November 2, 1853.

The form of the "Abstract Log" recommended by the late Maritime Conference at Brussels is hereby approved
ted for use in the Navy of the United States.
t is recommended to Navigators generally, a [illegible] ill be faithfully kept on board of all vessels in the Naval Service.
ommanding officers of vessels are especially charged with the execution of this Order and they will transmit

BLOG POST

Cloud-native platforms are the natural evolution of atmosphere-ocean open-data

practice

WHY

the history of ocean/atmosphere open-data practice is inspiring

> *“considerable emphasis was placed on maintaining a research oceanographer’s perspective as the focus during the vision’s development.*

> *“most research oceanographers are potential data providers as well as data users”,*

> *“to design and implement a data access protocol that would work equally well for the principal investigator as for federal archives”*

Report on the First Workshop for the Distributed Oceanographic Data System

29 September – 1 October 1993

W. Alton Jones Campus

University of Rhode Island

West Greenwich, Rhode Island

WHY

the history of ocean/atmosphere open-data practice is inspiring

Principles

- isolate the file format in which the data were stored from the data user
- allow data users to access the data directly from their application packages
- allow data users to acquire only those data of interest (i.e., request data subsets)

WHY

cf/netcdf/opendap/thredds

- *CF conventions*: Semantic layer independent of actual data format.
  - metadata in controlled vocabulary `standard_name: sea_surface_temperature`
- *netCDF*: “self-describing data format” with arbitrary attributes.
  - a data model: groups/arrays/dimensions/coordinates
- *OPeNDAP*: a discipline-neutral data access layer.
  - server-side subsetting / some server side compute
- *THREDDS*: presenting virtual datacubes instead of piles of files
- and more *server-side* stuff: *GRADs*, *LAS*, *INGRID*

WHY

but it fell a bit short

- as a oceanographer (2010-) I used almost none of it (except CF/netCDF).
- OPeNDAP & THREDDS servers cannot push data fast enough to sustain streaming compute.
  - great ideas but 90s tech
- more sophisticated services are for *privileged datasets*
  - > *“most research oceanographers are potential data providers as well as data users”,*
  - not everyone can or wants to run a THREDDS or OPeNDAP server.
- hence the status quo: *standard-compliant self-describing metadata-rich* piles of files!

HOW

how can we do better?

HOW

databases do it better! and with rigor!

- transactions
- version control
- formalized, enforced schemas

HOW

the cloud lets us do better

The idea of "frictionless reproducibility": (1) open data + (2) re-executable workflows + (3) open challenges

> *In the past decade, all of these initiatives, in their best implementations, became **frictionless open services**, essentially offering immediate, permissionless, complete access to each relevant digital artifact...*

Prof. David Donoho, Stanford University

VISION

planetary-scale data federation in the cloud

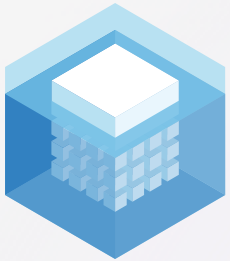
Seamless exchange of data across academia, government, and industry will accelerate research progress and empower reproducible open science.

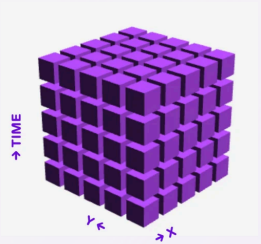
A PERFORMANT CLOUD-NATIVE DATABASE-Y STORAGE LAYER FOR N-DIMENSIONAL ARRAYS

icechunk

ICECHUNK

what is it





- chunked, compressed, nd array storage format
- *open-source*; Apache-2.0
- core in Rust + Python bindings + (more soon)
- cloud-native
- arbitrary metadata for cf-style semantic annotation
- hierarchical - groups/arrays a la netCDF
- exposes *n-dimensional* arrays / datacubes as a backend for Zarr, not files.
  - example dimensions: lat/lon/vertical/time/ensemble/scenario

ICECHUNK

promotes rigor

- isolated sessions & transactions
- safe for concurrent readers and writers
- version control (tags/branches etc.)
- snapshots are a good trigger for downstream compute

```
[21]: # Show all branches
      repo.ancestry_graph()
```

```
[21]: VD5SRKYA (hotfix) Fix sentinel value in temperature
      9S9J8T2V (main, v2.0) Zero out first half of temperature
      VYC39SKD (experiment) Fill pressure with random data
      SFZSRP31 Add pressure array
      MZV34TP6 (v1.0) Fill temperature with random data
      9X802Y23 Initialize temperature array
      1CECHNKR Repository initialized
```

ICECHUNK

example

```python
import icechunk as ic
import xarray as xr

storage = ic.s3_storage(
    bucket="icechunk-public-data", prefix="v1/era5_weatherbench2", region="us-east-1", anonymous=True,
)
repo = ic.Repository.open(storage=storage)
```

read

```python
session = repo.readonly_session("main")
ds = xr.open_zarr(
    session.store, group="1x721x1440", engine="zarr"
)
```

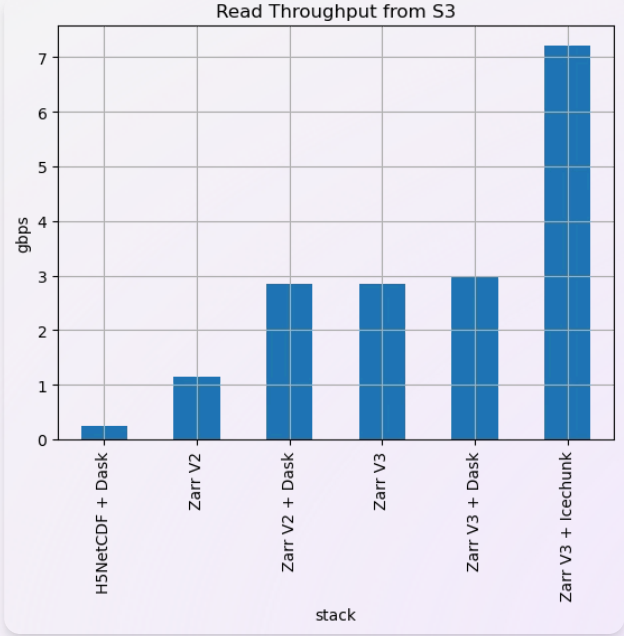
write

```python
session = repo.writable_session("main")
ds = ... # construct update
ds.to_zarr(session.store, group="1x721x1440")
 # write still invisible to _all_ readers
session.commit(
  message="I updated the data!",
  properties={"timestamp": "2026-06-14 00:00"},
) # writes are now visible to _new_ readers
```

ICECHUNK

is performant

Rust lets us get close to saturating network hardware



ICECHUNK

is cloud-native

cloud-optimized: metadata separate from data, cheap to open.

cloud-native: optimized for cloud object store.

> *Icechunk's format follows S3 recommendations, by uniformly distributing its chunks over the key space after a prefix ... we got S3 to sustain 230,000 reads/sec.*
>
> — **Sebastian Galkin**, Earthmover

👆 is what we need for performant *streaming* compute for extremely popular public datasets.

BLOG POST

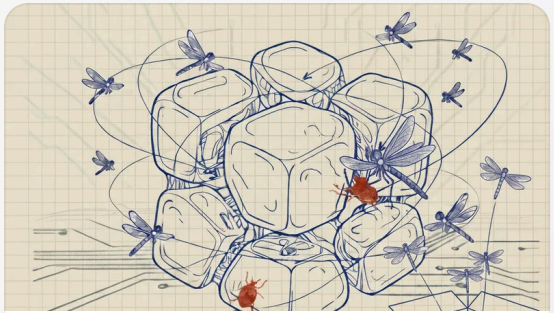
Exploring Icechunk scalability: untangling S3's prefix story

April 10, 2025

ICECHUNK

is rigorously developed

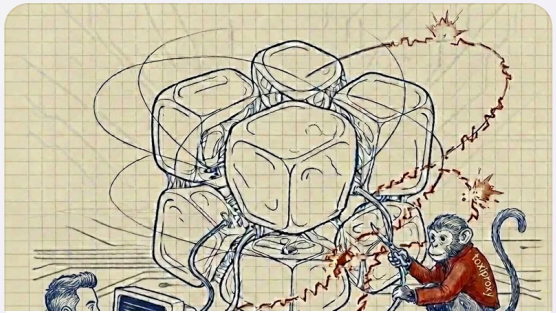
a battery of property tests, stateful tests, concurrency permutation tests, fault-injection tests, cross-version tests.



BLOG POST

Ship fast, break nothing: Engineering rigor in Icechunk with property and stateful testing

April 22, 2026



BLOG POST

Ship fast, break nothing: Engineering rigor in Icechunk with concurrency & fault injection testing

April 28, 2026

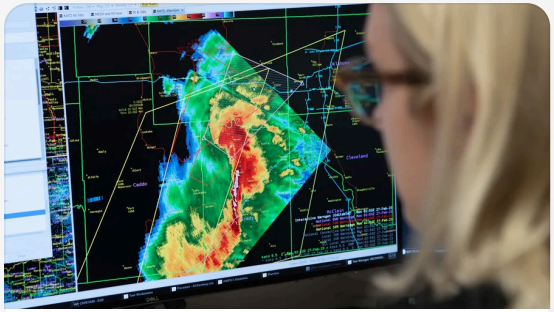
BLOG POST

Ship fast, break nothing: Engineering rigor in Icechunk with cross-version stateful compatibility testing

May 21, 2026

ICECHUNK

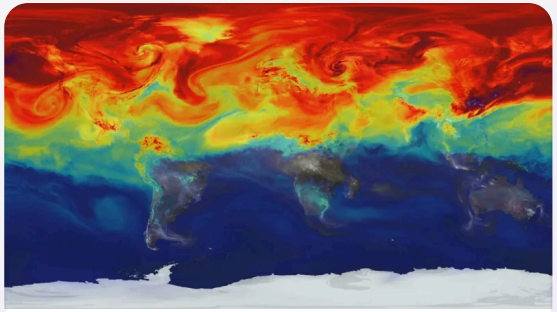
increasing uptake among public sector data providers



ANNOUNCEMENT

Icechunk Adopted by the National Weather Service: Earthmover Joins Booz Allen on NWS CIRRUS

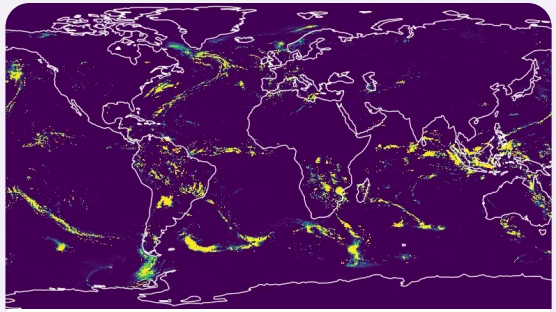
June 4, 2026



ANNOUNCEMENT

Earthmover Selected to Power ARIA's "Forecasting Tipping Points" Simulation Catalogue

February 9, 2026



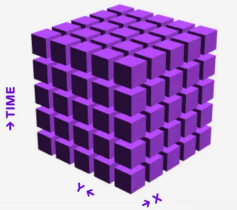
BLOG POST

Solving NASA's Cloud Data Dilemma: How Icechunk Revolutionizes Earth Data Access

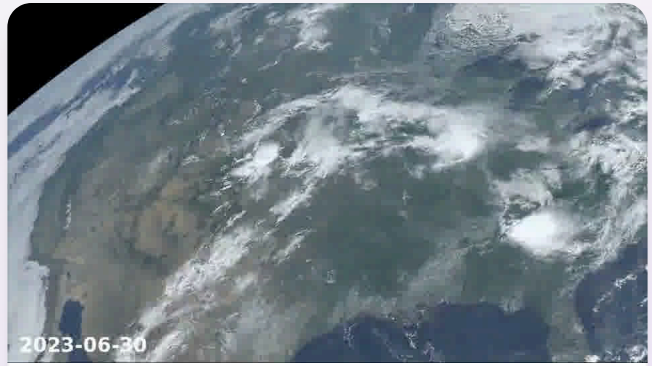
March 27, 2025

ICECHUNK

natively supports "virtual" datasets built from piles of files



- A *chunk* is a compressed binary blob object.
- a *manifest* maps chunk key (`0/10/30`) to a blob name.
- blob could be a byte range within an existing object (e.g. a netCDF file)
- VirtualiZarr : construct virtual Icechunk datacubes from *existing* cloud archives



BLOG POST

Old format, no problem!: Cloud-optimizing the GOES-16 archive as Virtual Zarr

June 2, 2026

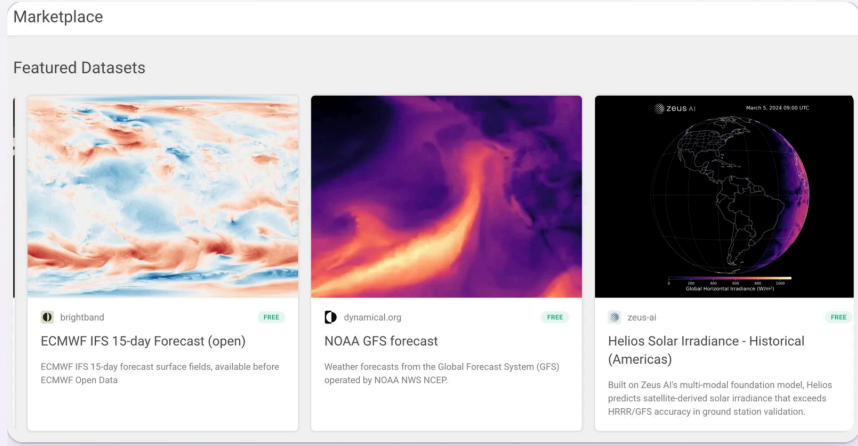
EARTHMOVER

icechunk in the cloud

ARRAYLAKE: ICECHUNK IN THE CLOUD

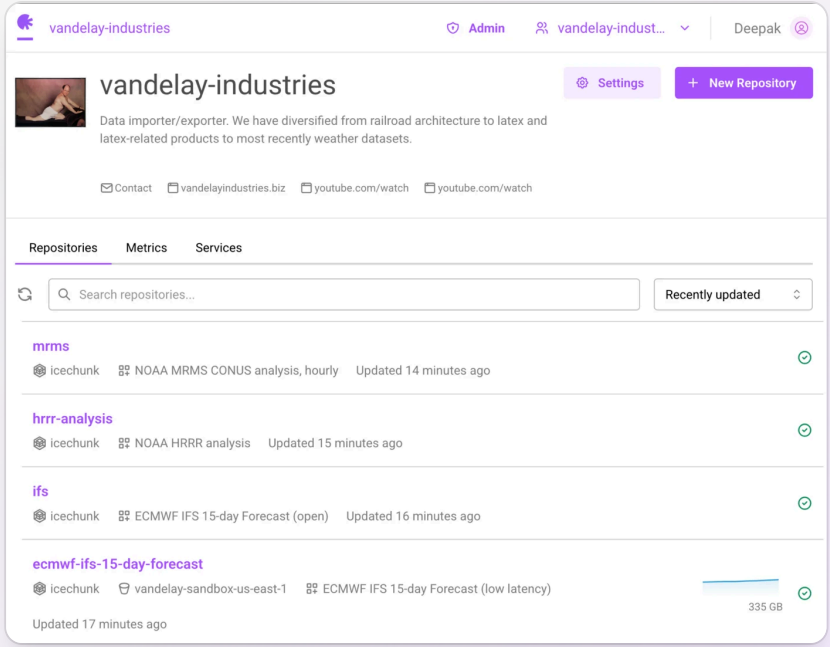
a *marketplace* of analysis-ready, cloud optimized data

- free & paid datasets
- academic & private & public
- updating *subscriptions* with notifications



ICECHUNK IN THE CLOUD

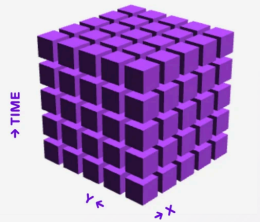
icechunk powers data *subscriptions*



NEXT

what's missing

- requires "homogeneous" datasets / datacubes



- more cross-language bindings
  - rust / python / JS / typescript today
- schemas! updates are hard to get right; "best practices" are not rigorous enough.

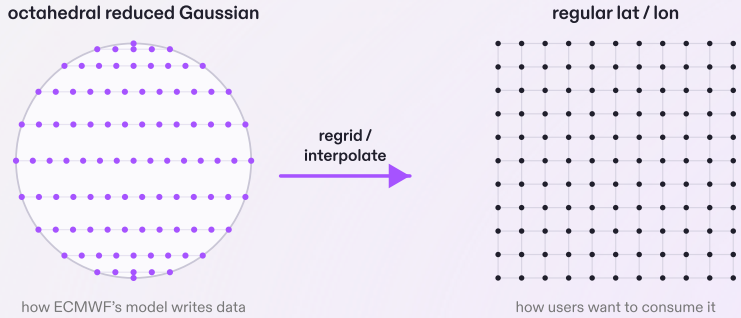
NEXT

what's missing (broader picture)

close collaboration between storage & compute layers can unlock a lot

- incremental view maintenance
- streaming compute / updates
- on-the-fly compute: regridding, derived fields

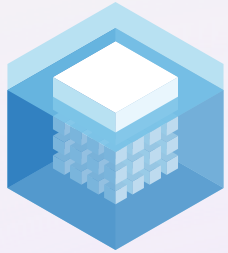
users are still not fully isolated from physical layout of the data — we are missing a semantic layer.



THE POINT

we needed a new storage layer, icechunk is that layer

- **try out icechunk today!**
- *icechunk* is a new-ish ndarray storage engine.
- *cloud-native & open source*
- Smashes together a number of good ideas from earth system data systems, & databases.
- increasing engineering rigor in the earth data ecosystem
- a great foundation for computational systems
- earthmover is building a cloud platform for planetary-scale data federation



icechunk

earthmover

earthmover.io