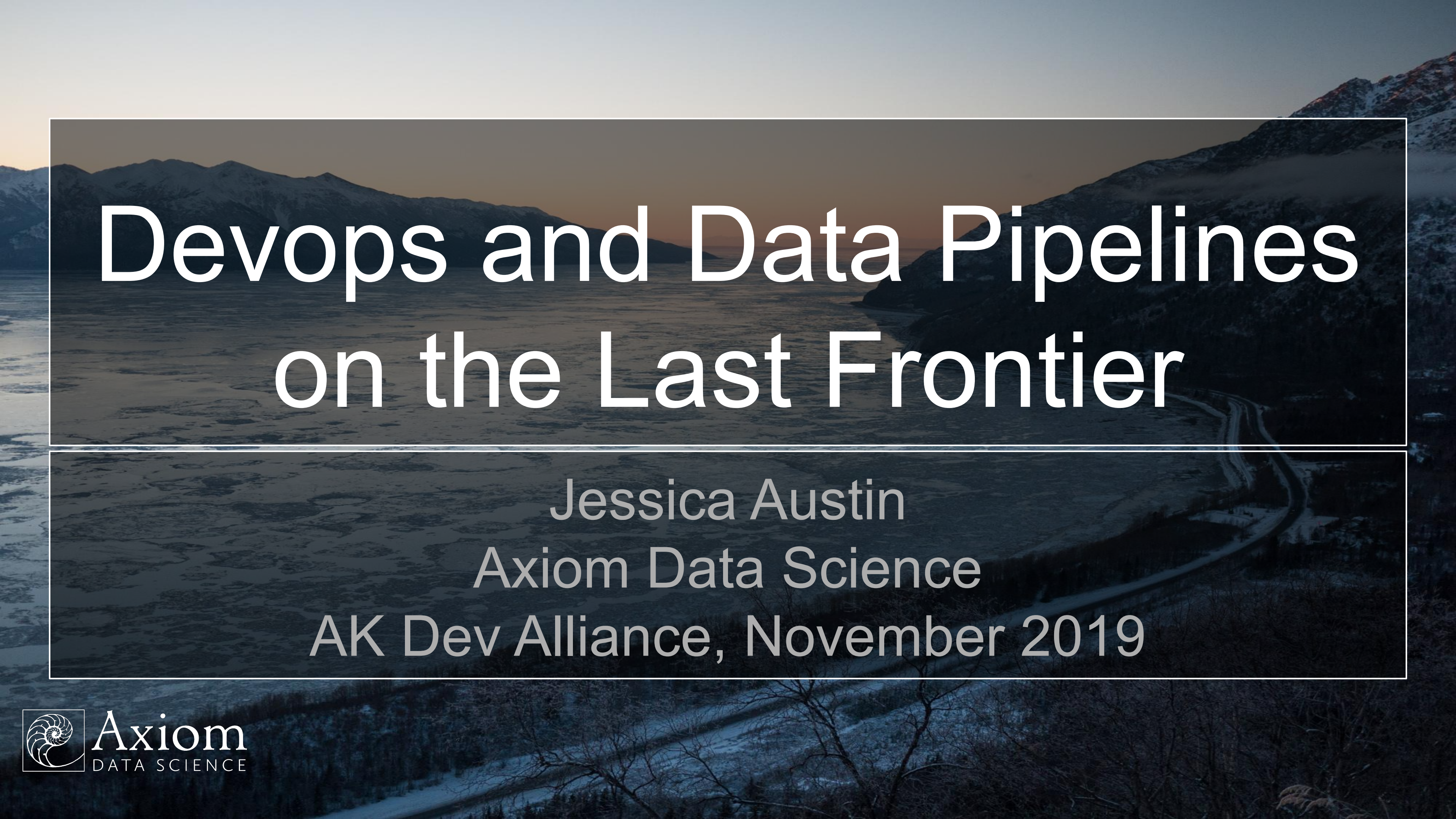


Devops and Data Pipelines on the Last Frontier

Jessica Austin
Axiom Data Science
AK Dev Alliance, November 2019

Axiom
DATA SCIENCE

About Axiom

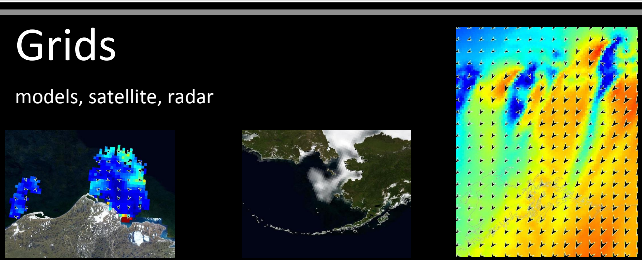
- Founded in 2006
- ~20 people
- Distributed: Anchorage, Fairbanks, Homer, Portland OR, Providence RI
- Mix of software developers, data scientists, actual scientists, librarians, PMs
- Mission-driven: to improve the synthesis and re-use of scientific data
- Broad range of partnerships, but mostly ocean, atmospheric, and arctic sciences
- Major funders: IOOS/NOAA, National Science Foundation (NSF), Office of Naval Research and DARPA

About Axiom

- We are not a consulting company, we are a technology partner
  - Data management: Ingest and standardize data, improve metadata, archive for posterity
  - Data analysis: Generate new data products
  - Data discovery: Build data portals and catalogs, develop data visualizations
- We focus on a set of core products that are useful to multiple groups

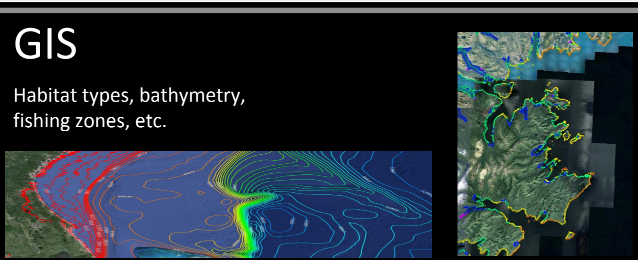
Grids

models, satellite, radar



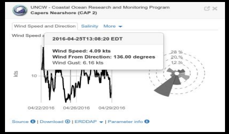
GIS

Habitat types, bathymetry,
fishing zones, etc.



Fixed Platforms

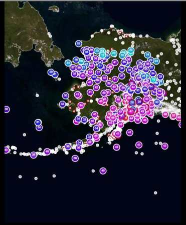
moorings, shore stations



UNCW - Coastal Ocean Research and Monitoring Program
Capers Nearshore (CAP 2)

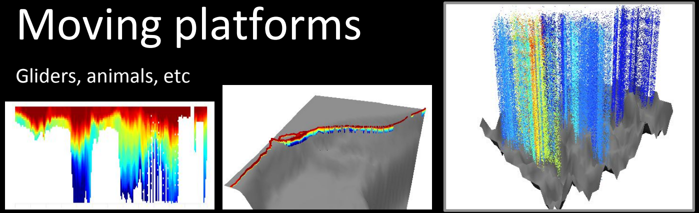
2016-04-25T13:08:20 EDT

Wind Speed: 4.09 kts
Wind From Direction: 136.00 degrees
Wind Gust: 6.16 kts



Moving platforms

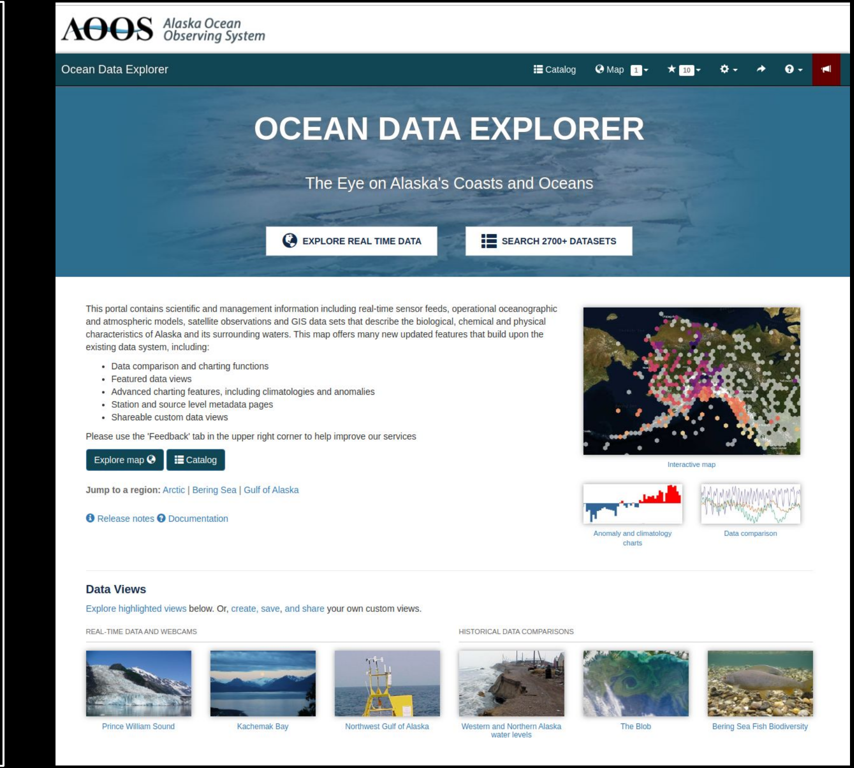
Gliders, animals, etc



Axiom
DATA SCIENCE

About Axiom

- Example data portal: Alaska Ocean Observing System (AOOS) Ocean Data Explorer
- https://portal.aoos.org/
- NSIDC Sea Ice Concentration
- Real-time sensor catalog



This portal contains scientific and management information including real-time sensor feeds, operational oceanographic and atmospheric models, satellite observations and GIS data sets that describe the biological, chemical and physical characteristics of Alaska and its surrounding waters. This map offers many new updated features that build upon the existing data system, including:

- Data comparison and charting functions
- Featured data views
- Advanced charting features, including climatologies and anomalies
- Station and source level metadata pages
- Shareable custom data views

Please use the 'Feedback' tab in the upper right corner to help improve our services

Jump to a region: Arctic | Bering Sea | Gulf of Alaska

Release notes Documentation

Data Views

Explore highlighted views below. Or, create, save, and share your own custom views.

REAL-TIME DATA AND WEBCAMS

HISTORICAL DATA COMPARISONS

Today’s presentation

- About me
  - School: MechE, Controls, Robotics
  - Work: ThoughtWorks, Grubhub.com, RDI, Axiom
  - Roles: Software dev, QA, DevOps, Data analysis
- **Feedback loops**
  - In dev: user stories, QA, DevOps, CI
  - In the community: **meetups**, conferences, publishing
- This presentation
  - Overview: Axiom DevOps and data pipeline infrastructure
  - Examples: data ingestion pipelines for weather model and environmental sensors
  - Focus on interesting technologies: Kafka, TimescaleDB, Luigi, Prometheus, Grafana

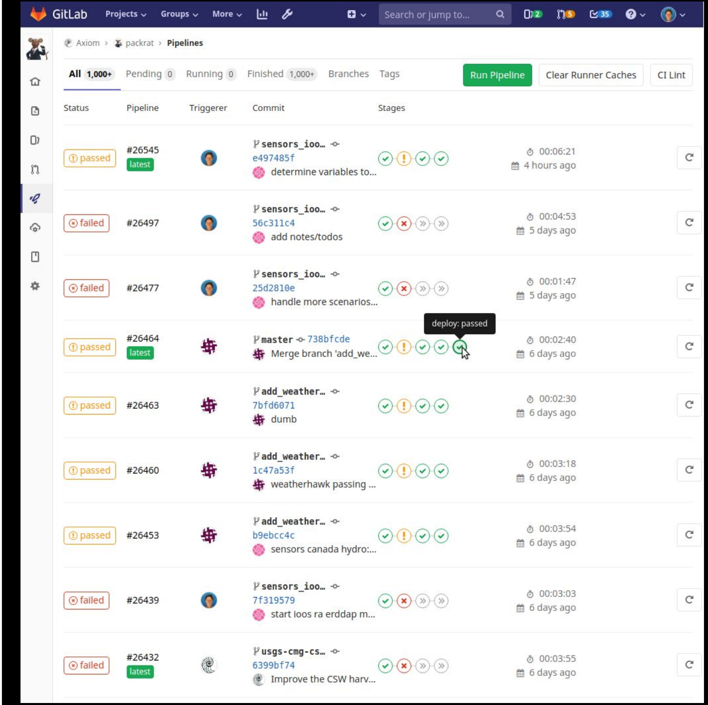
Overview: DevOps

- Private cloud in Portland, OR
  - ~5,000 processor cores
  - ~1.5 petabytes of functional storage
    - 5 petabytes of actual storage (~1,500 hard drives)
  - Level 2 Fat Tree Infiniband Network, 40 Gb/Sec node to node). 240 Gb/Sec cluster to cluster
  - Ansible for config management
- Why:
  - Cost: AWS ~$600k/mo storage+compute. We operate for ~$200k/year + 0.5 FTE
  - Complete control, infiniband network
  - DevOps makes it possible
  - We enjoy it!

Axiom DATA SCIENCE

Overview: DevOps

- Gitlab for SCM + CI
  - For OSS: Github + Travis
- Everything running in Docker
  - (other than a few edge cases)
  - Internally-hosted private docker registry
  - Each deployable gitlab project contains a Dockerfile and Gitlab CI definition
- Ansible to define app deployments
  - Can manually trigger but mostly use Gitlab CI Pipelines



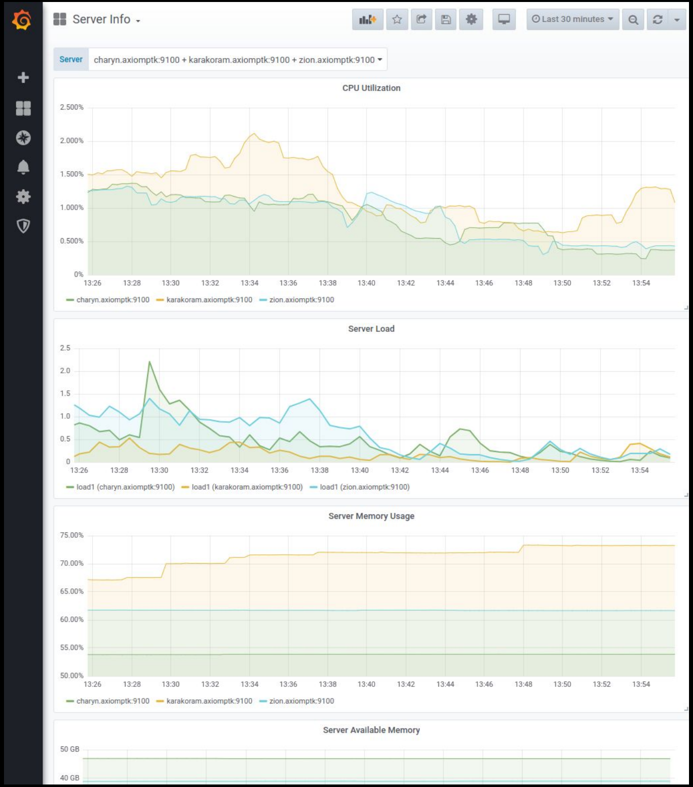
Status	Pipeline	Triggerer	Commit	Stages		
! passed	#26545 latest		sensors_ioo... e497485f determine variables to...	✓ ! ✓ ✓	00:06:21 4 hours ago	↻
✗ failed	#26497		sensors_ioo... 56c311c4 add notes/todos	✓ ✗ » »	00:04:53 5 days ago	↻
✗ failed	#26477		sensors_ioo... 25d2810e handle more scenarios...	✓ ✗ » »	00:01:47 5 days ago	↻
! passed	#26464 latest		master 738bfcde Merge branch 'add_we...	✓ ! ✓ ✓ ✓	00:02:40 6 days ago	↻
! passed	#26463		add_weather... 7bfd6071 dumb	✓ ! ✓ ✓	00:02:30 6 days ago	↻
! passed	#26460		add_weather... 1c47a53f weatherhawk passing ...	✓ ! ✓ ✓	00:03:18 6 days ago	↻
! passed	#26453		add_weather... b9ebcc4c sensors canada hydro:...	✓ ! ✓ ✓	00:03:54 6 days ago	↻
✗ failed	#26439		sensors_ioo... 7f319579 start ioos ra erddap m...	✓ ✗ » »	00:03:03 6 days ago	↻
✗ failed	#26432 latest		usgs-cmg-cs... 6399bf74 Improve the CSW harv...	✓ ✗ » »	00:03:55 6 days ago	↻

Gitlab CI pipelines

Axiom
DATA SCIENCE

Overview: DevOps

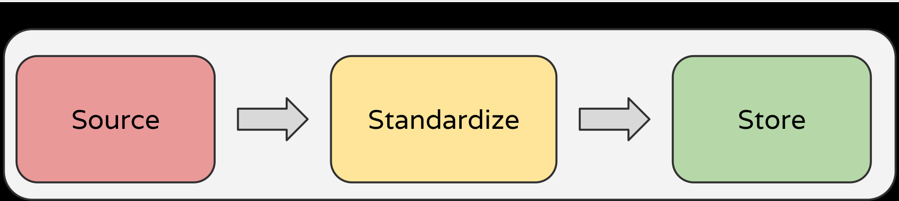
- Prometheus for metrics
  - Timeseries DB with metrics segmented by label
  - Pull model: each client provides metrics endpoint, prom scrapes periodically
  - Robust ecosystem, active development
    - e.g., [node_exporter client for hardware/OS metrics]()
  - All new Axiom apps have prom endpoint
    - Building in to older apps as we go along
- Grafana for plotting and dashboards
- Grafana + Alertmanager + nagios for alerts
  - Nagios for basic server/hardware/network issues
  - Grafana/Alertmanager for metrics-based alerts
- Kibana + Grafana Loki for app logs



Grafana showing prom metrics

Overview: Data Pipelines

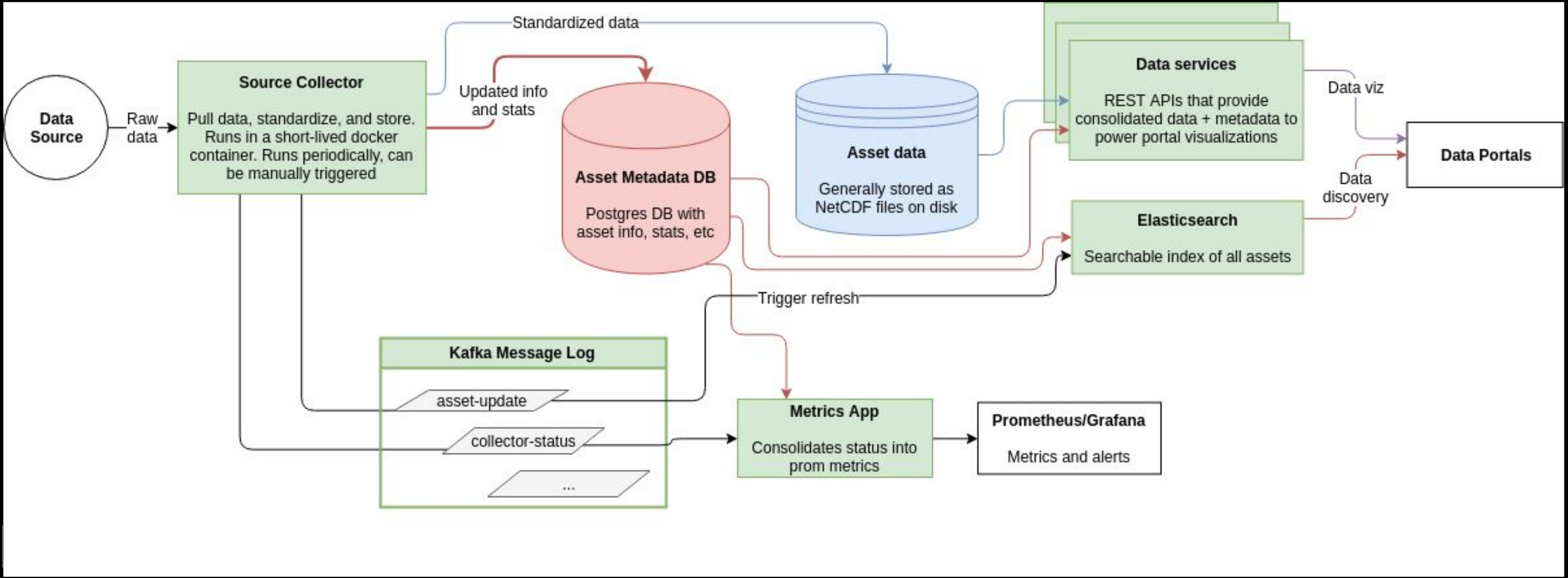
Simple version —>



Source

Standardize

Store



Data Source

Raw data

Source Collector

Pull data, standardize, and store. Runs in a short-lived docker container. Runs periodically, can be manually triggered

Standardized data

Updated info and stats

Asset Metadata DB

Postgres DB with asset info, stats, etc

Asset data

Generally stored as NetCDF files on disk

Data services

REST APIs that provide consolidated data + metadata to power portal visualizations

Data viz

Data Portals

Data discovery

Elasticsearch

Searchable index of all assets

Trigger refresh

Kafka Message Log

asset-update

collector-status

...

Metrics App

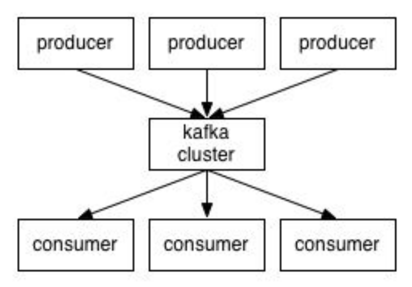
Consolidates status into prom metrics

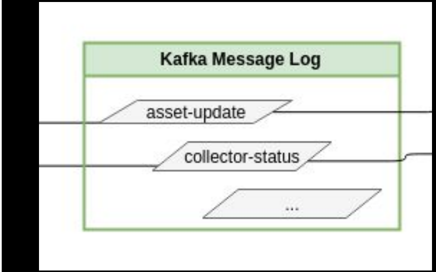
Prometheus/Grafana

Metrics and alerts

Apache Kafka in Data Pipelines

- Kafka is a distributed, publish-and-subscribe messaging system
  - All messages in Kafka are stored on a **topic**
  - Processes that publish messages to topics are called **producers**
  - Processes that subscribe to topics and listen to messages are called **consumers**
  - Each topic has a message **schema** that defines the message structure
  - Consumer pull model; can produce/consume in batches for quick I/O
  - Benefits:
    - Easily decouple processes
      - Producers/consumers don't talk directly
      - Topic is generic, so can push data from anywhere
      - Can scale producers or consumers independently
    - Topic log is history of events (great for debugging)
    - Can handle ridiculous number of messages
  - Downsides:
    - Steep learning curve, complex ecosystem, still in flux
- We use Kafka topics to link together components of our pipelines, and refresh caches that power portal visualizations

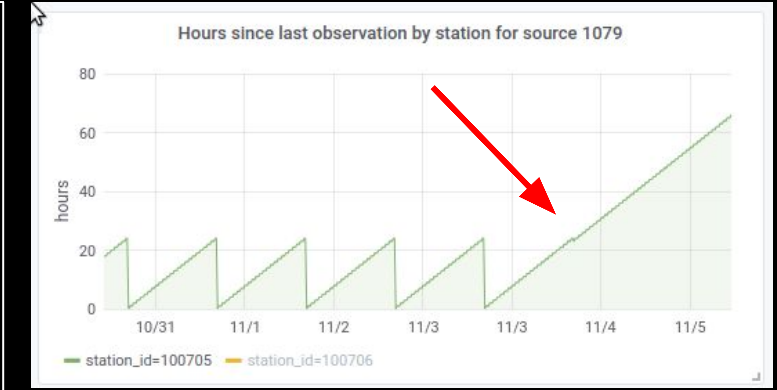


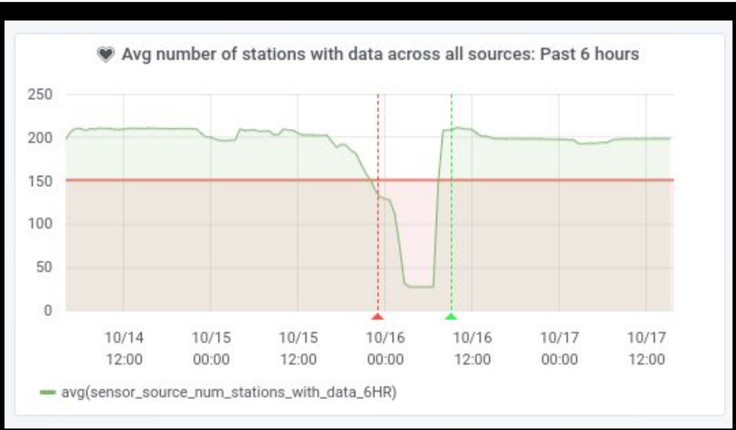


Axiom DATA SCIENCE

Data Pipeline Metrics with Prometheus and Grafana

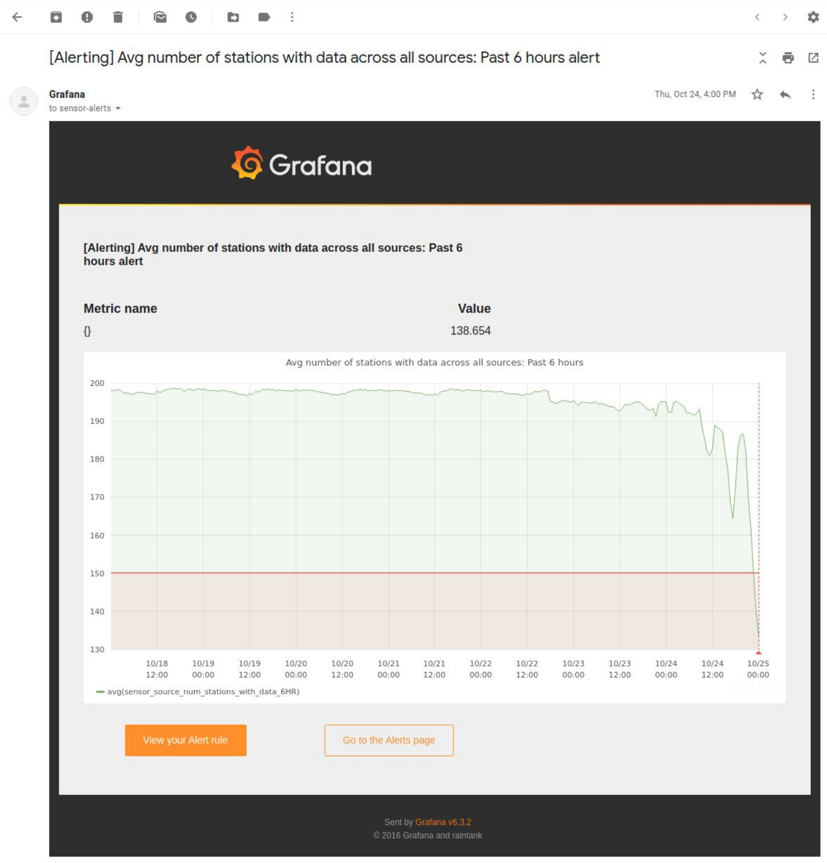
- Old school way: alert if there are a bunch of errors
  - but errors happen all the time! (source goes down, etc) and this is ok if it happens intermittently
  - and errors can happen for all sorts of reasons: source is down, bug in our code, problem with one of our services. difficult to instrument all these places
- At the end of the day, you just want to know, "**did data make it all the way through the pipeline?**"
- Metric: "time since last data point".
  - Segment by type, data source, platform ID
  - Rollup alerts for entire type (indicates catastrophic failure, address immediately)
  - Alerts for single source or platform (probably source is down or changed, address during business hours)





Alerting with Prometheus and Grafana

- Prometheus Alertmanager
  - Can define sophisticated rules and behavior
  - But managing rules is only through editing files in SCM so it's PITA to manage (prometheus/alertmanager #552)
- Grafana Alerts
  - Very intuitive to create and view alerts in a dashboard
  - It's not perfect, but in very active dev and always improving (grafana/grafana #6557)



[Alerting] Avg number of stations with data across all sources: Past 6 hours alert

Grafana
to sensor-alerts

Thu, Oct 24, 4:00 PM

Grafana

[Alerting] Avg number of stations with data across all sources: Past 6 hours alert

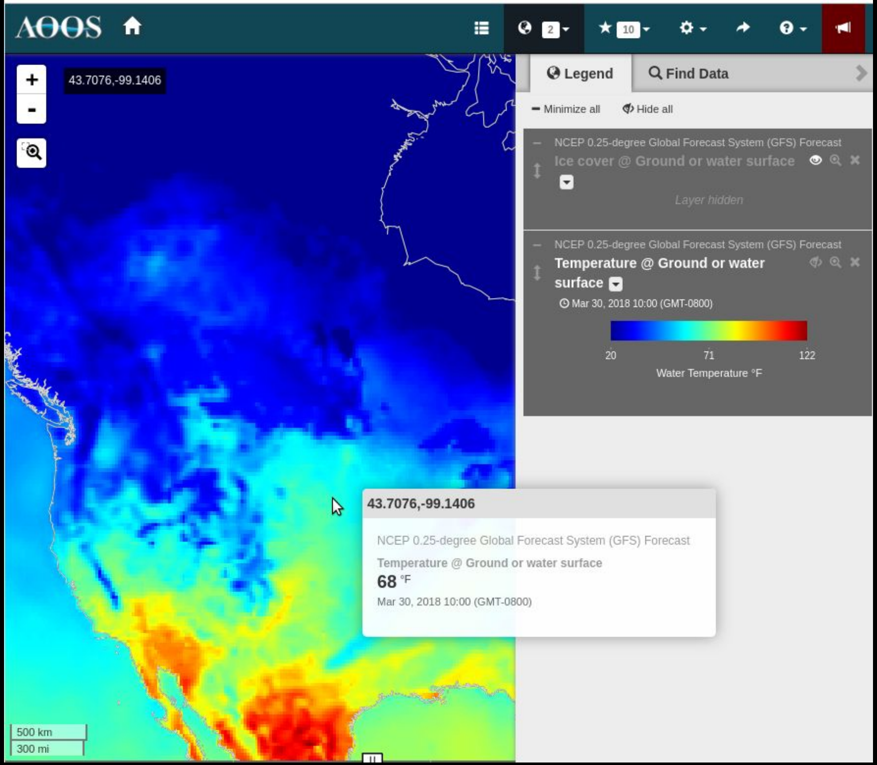
Metric name	Value
{}	138.654

Avg number of stations with data across all sources: Past 6 hours

Sent by Grafana v6.3.2
© 2016 Grafana and raintank

Example: GFS weather model ingest pipeline

- Data source: NOAA NWS
- Input: GRIB2 gridded data
- 4 forecasts/day (23GB total per day)
- Output: netcdf files
- Serve with WMS
- Data pipeline:
  - Download, enhance, store
  - Trigger downstream updates
- Requirements:
  - Don't re-download any data
  - Retry if something failed



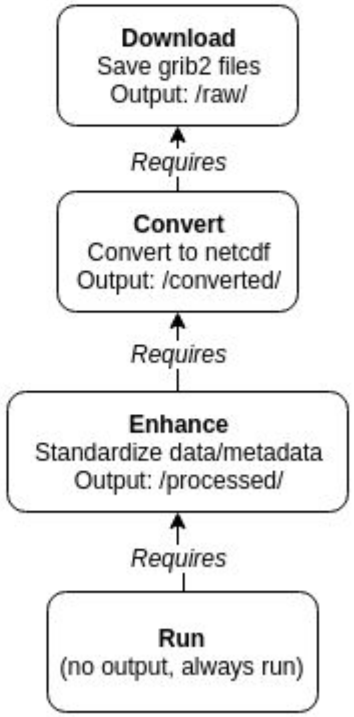
NCEP 0.25-degree Global Forecast System (GFS) Forecast
Ice cover @ Ground or water surface
Layer hidden

NCEP 0.25-degree Global Forecast System (GFS) Forecast
Temperature @ Ground or water surface
Mar 30, 2018 10:00 (GMT-0800)

43.7076,-99.1406
NCEP 0.25-degree Global Forecast System (GFS) Forecast
Temperature @ Ground or water surface
68 °F
Mar 30, 2018 10:00 (GMT-0800)

Example: GFS weather model ingest pipeline

- Pipeline runs using Spotify's Luigi
  - Python package, built for large batch jobs
  - Framework for defining Tasks
    - Tasks have outputs
    - Tasks can depend on other Tasks
    - If a Task's output exists, don't re-run it!
  - Provides scheduler for running tasks
    - Allows failure + retry
    - Basic UI + API + notifications
  - Overall thoughts: great for large datasets, mature, robust, moderate learning curve



Download
Save grib2 files
Output: /raw/

Requires

Convert
Convert to netcdf
Output: /converted/

Requires

Enhance
Standardize data/metadata
Output: /processed/

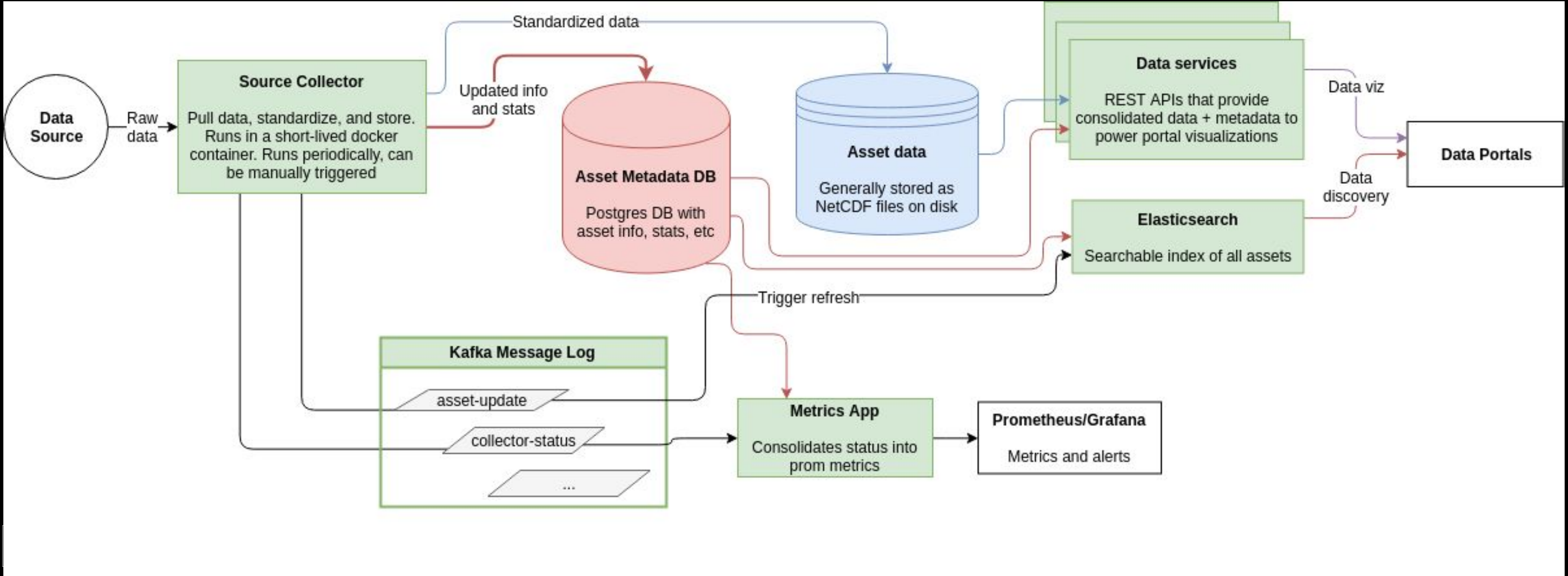
Requires

Run
(no output, always run)

GFS Luigi Task Pipeline

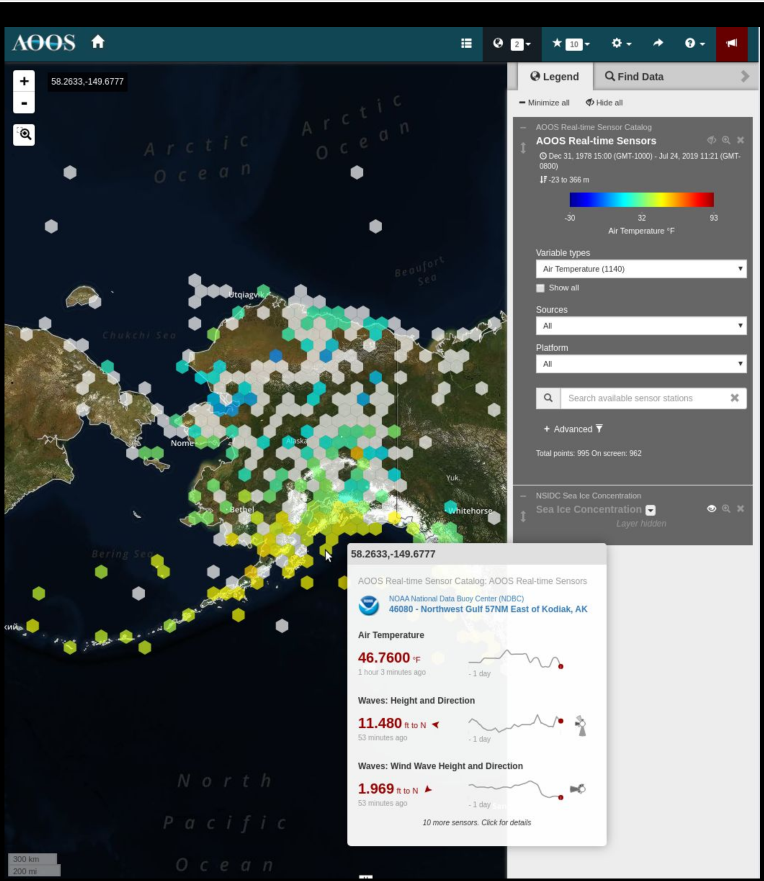
Example: GFS weather model ingest pipeline

- All this runs in short-lived docker container, triggered by fcron project
- After completion, send Kafka message


Data Source
Raw data
Source Collector
Pull data, standardize, and store. Runs in a short-lived docker container. Runs periodically, can be manually triggered
Standardized data
Updated info and stats
Asset Metadata DB
Postgres DB with asset info, stats, etc
Asset data
Generally stored as NetCDF files on disk
Data services
REST APIs that provide consolidated data + metadata to power portal visualizations
Data viz
Data Portals
Data discovery
Elasticsearch
Searchable index of all assets
Trigger refresh
Kafka Message Log
asset-update
collector-status
...
Metrics App
Consolidates status into prom metrics
Prometheus/Grafana
Metrics and alerts

Example: environmental sensor data pipeline

- Environmental sensors
  - Timeseries data
  - Weather data, ocean state, water quality, etc
  - Focus on real-time data
- ~40k stations across 100+ data sources
- ~50,000,000 new observations per week
- We've been redesigning this from the ground up using Kafka, TimescaleDB, and Prometheus



AOOS Real-time Sensor Catalog

AOOS Real-time Sensors

Dec 31, 1978 15:00 (GMT-1000) - Jul 24, 2019 11:21 (GMT-0800)

-23 to 366 m

Variable types: Air Temperature (1140)

□ Show all

Sources: All

Platform: All

Search available sensor stations

+ Advanced

Total points: 995 On screen: 962

NSIDC Sea Ice Concentration

Sea Ice Concentration

Layer hidden

58.2633,-149.6777

AOOS Real-time Sensor Catalog: AOOS Real-time Sensors

NOAA National Data Buoy Center (NDBC)

46080 - Northwest Gulf 57NM East of Kodiak, AK

Air Temperature

46.7600 °F

1 hour 3 minutes ago

Waves: Height and Direction

11.480 ft to N

53 minutes ago

Waves: Wind Wave Height and Direction

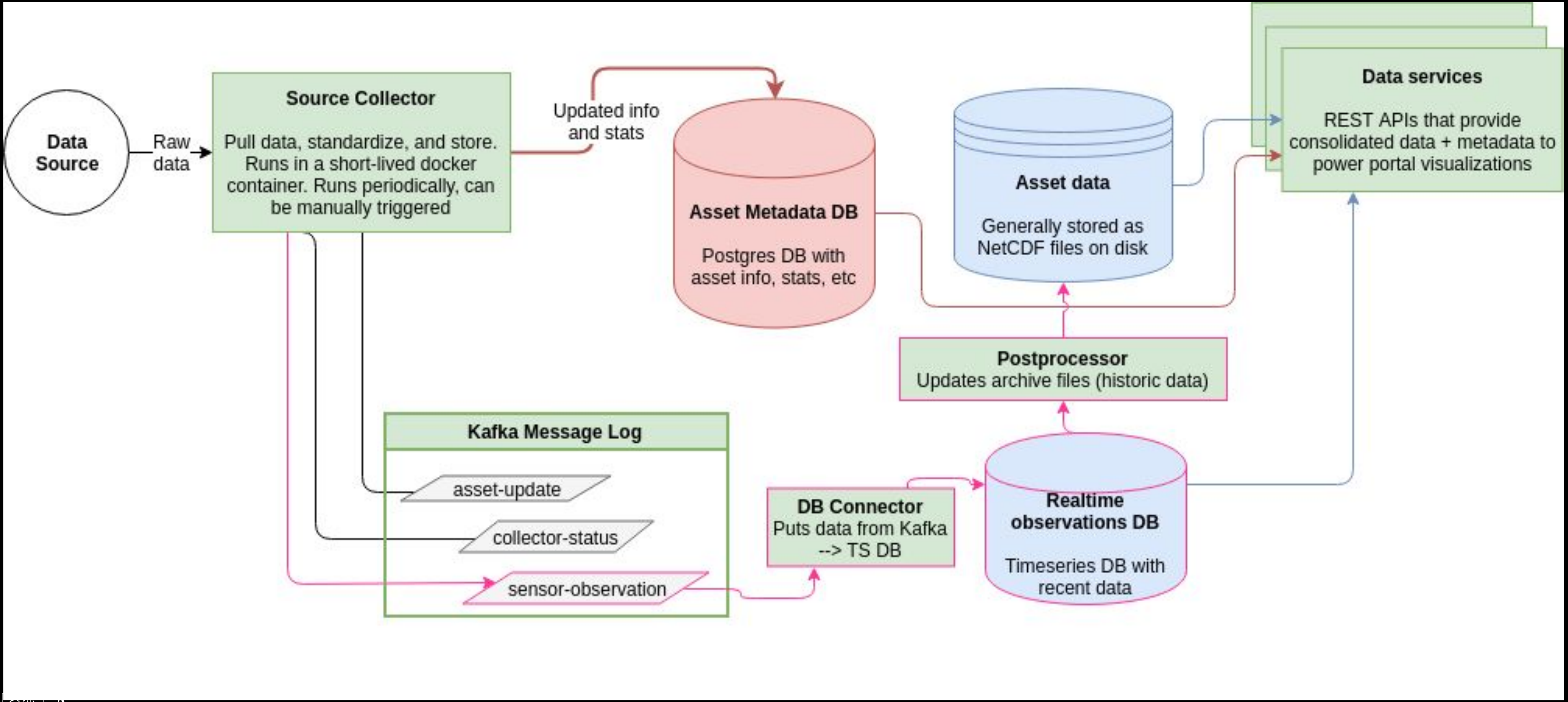
1.969 ft to N

53 minutes ago

10 more sensors. Click for details

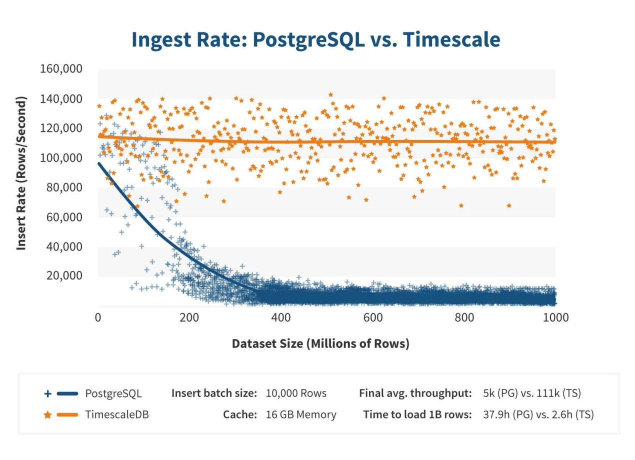
Axiom DATA SCIENCE

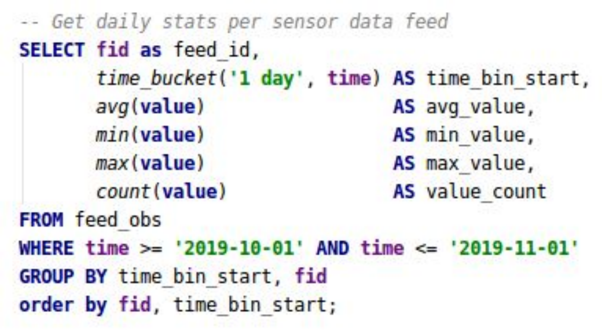
Example: environmental sensor data pipeline


Data Source
Raw data
Source Collector
Pull data, standardize, and store. Runs in a short-lived docker container. Runs periodically, can be manually triggered
Updated info and stats
Asset Metadata DB
Postgres DB with asset info, stats, etc
Asset data
Generally stored as NetCDF files on disk
Data services
REST APIs that provide consolidated data + metadata to power portal visualizations
Postprocessor
Updates archive files (historic data)
Kafka Message Log
asset-update
collector-status
sensor-observation
DB Connector
Puts data from Kafka --> TS DB
Realtime observations DB
Timeseries DB with recent data

TimescaleDB for real-time data

- TimescaleDB is a time-series database built on top of Postgres (it's an extension)
  - Exposes what look like singular tables, called **hypertables**, that are actually an abstraction of many individual tables holding the data, called **chunks**
  - Chunks are created by partitioning the hypertable's data into one or multiple dimensions (e.g., time and device ID)
- Higher data ingest rate
- Better performance for typical timeseries queries
- Time-specific functions

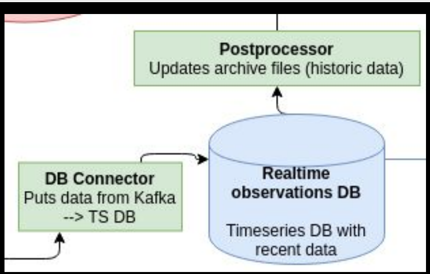




```
-- Get daily stats per sensor data feed
SELECT fid as feed_id,
       time_bucket('1 day', time) AS time_bin_start,
       avg(value)                 AS avg_value,
       min(value)                 AS min_value,
       max(value)                 AS max_value,
       count(value)               AS value_count
FROM feed_obs
WHERE time >= '2019-10-01' AND time <= '2019-11-01'
GROUP BY time_bin_start, fid
order by fid, time_bin_start;
```

TimescaleDB for real-time data

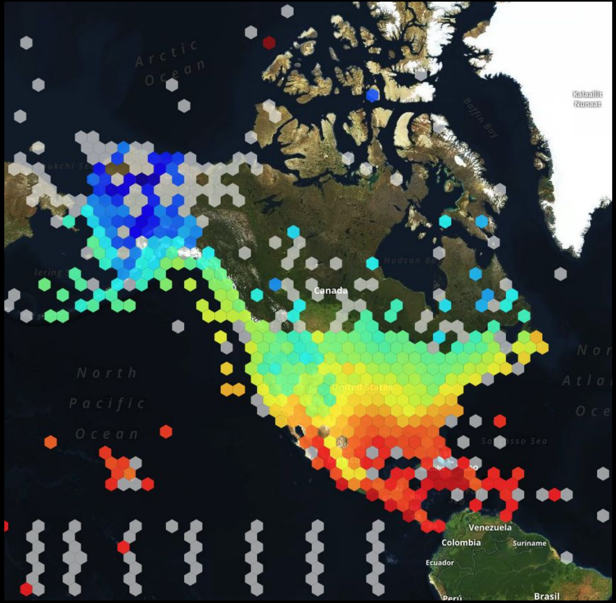
- We're using TimescaleDB as the "real-time" sensor cache (last 30 days of data)
  - Quickly generate "global" heatmaps with latest data per sensor type
  - Buffer for frequent data updates
    - Every 15 mins: get new data
    - Every day: more advanced processing
- Overall impressions
  - Very simple to set up and use (compared to influxdb, etc)
  - In very active dev, lots of investment
  - Single machine right now, clustering in private beta

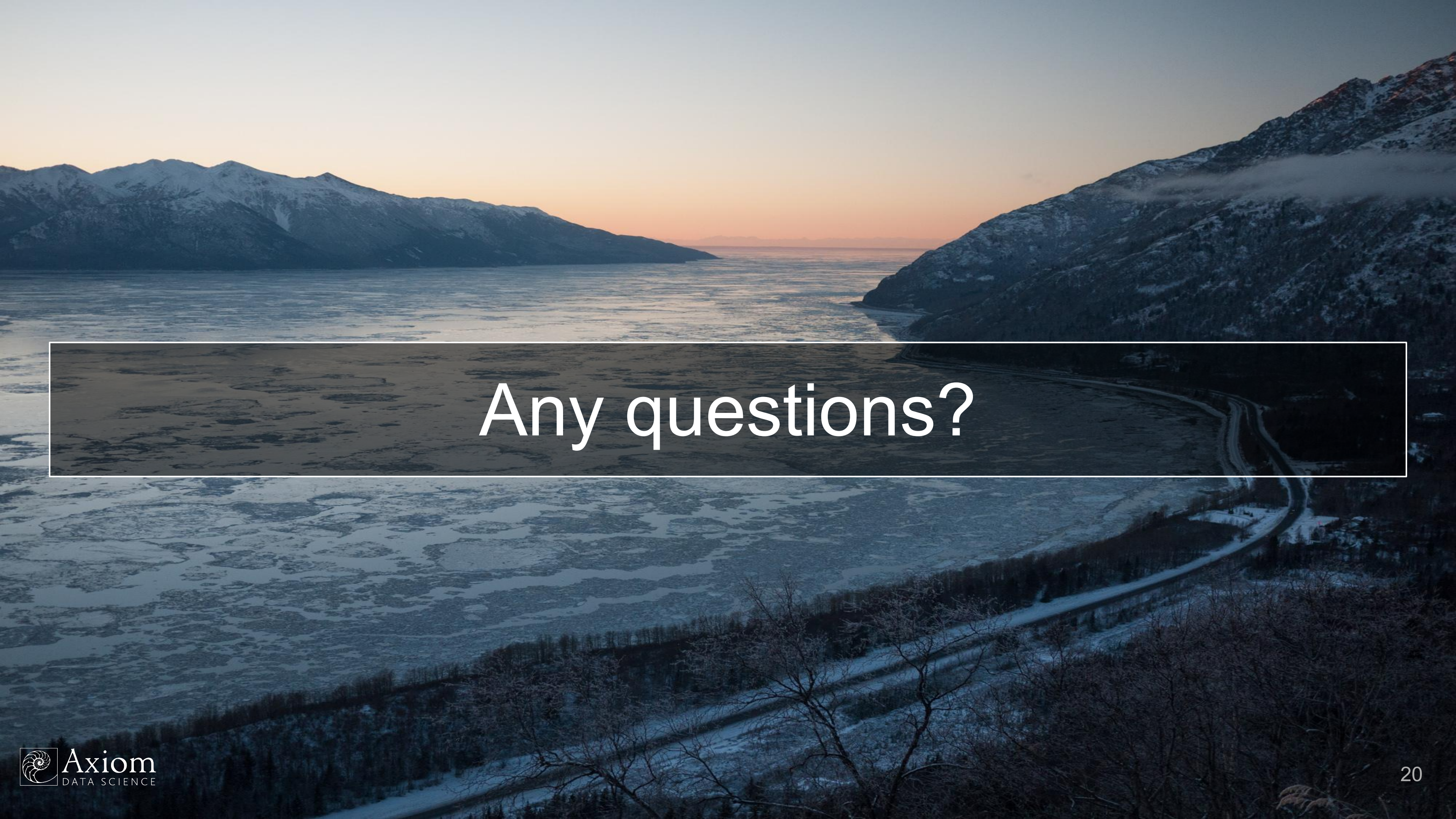


Postprocessor
Updates archive files (historic data)

DB Connector
Puts data from Kafka --> TS DB

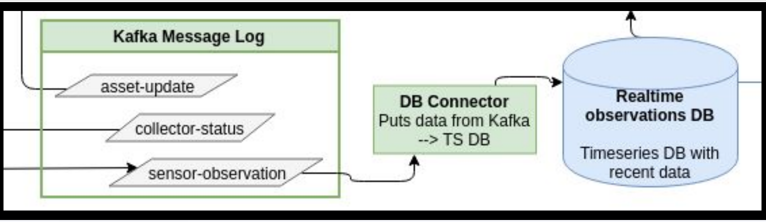
Realtime observations DB
Timeseries DB with recent data

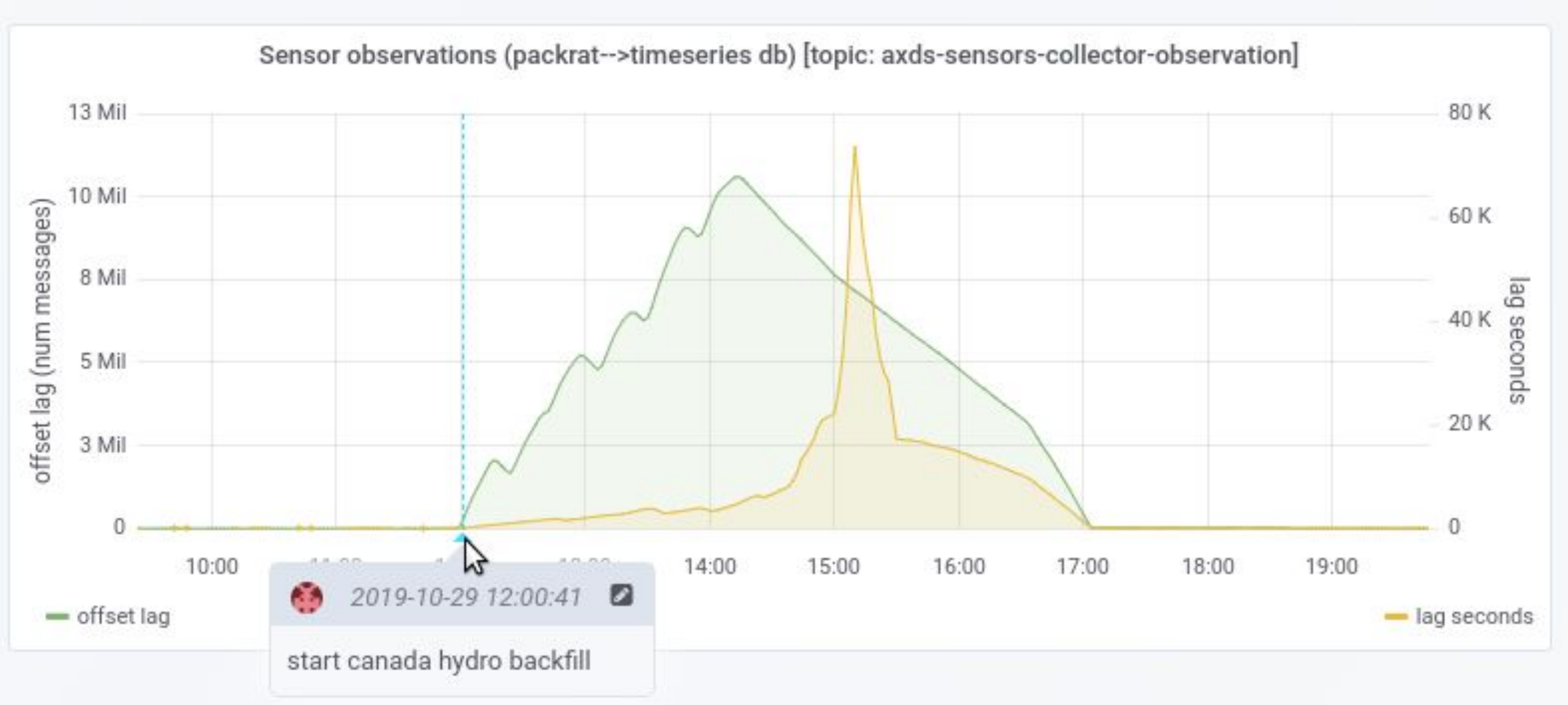




Any questions?

Kafka as a buffer for large data ingestions





2019-10-29 12:00:41

start canada hydro backfill

Elasticsearch for data discovery

- lots of little pieces -- how to consolidate?
  - elasticsearch with shared document structure ("asset")
    - id, type, label, start, end, geospatial extent, variable names, etc
- Include some examples here – screenshot of catalog, maybe screenshot of JSON
- Mention that we have an “asset metadata update” and “asset data update” topic, and include some examples
  - Both automated processes and humans would trigger messages on this topic
  - This topic is great as a history of updates
-