BEAM SUMMIT

Loading Geospatial data to Google BigQuery

Dong Sun
Sean Jensen-Grey

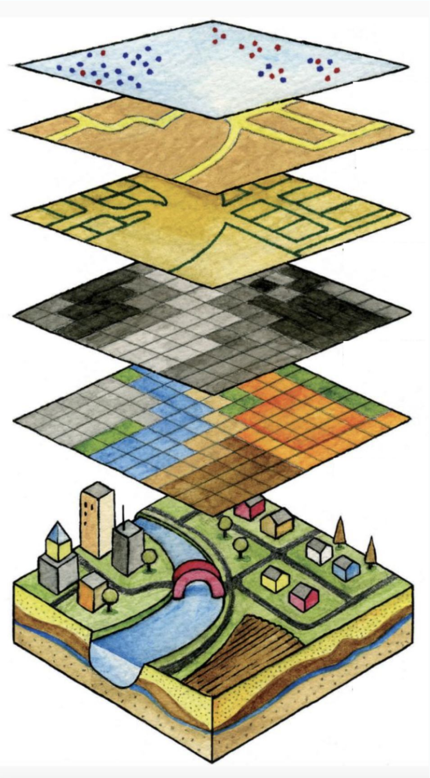
BEAM SUMMIT
NYC 2023

Agenda

- What is Geospatial Data
- Why loading to BigQuery
- How to load
    - Two ways to load
    - Challenges
    - How beam/dataflow helps
- Geospatial Analysis
- Q&A

What is Geospatial Data

- A type of data that describes objects, events, or other features **with a location** on the earth.
- A powerful tool for understanding the world around us.
- Geospatial data analysis can help organizations make better decisions with a better understanding of the spatial relationships between features.



Use Cases Enabled by Geospatial Data Analysis

Industry	Use Case
Insurance	Assess risk and analyze property damage claims, eg post severe weather events
Banking and Finance	Analyze market trends using weather or other geospatial data Manage investments with geospatial analytics/Analyze portfolio risk
Infrastructure	Infrastructure sustainability / Infrastructure location selection
Sustainability	Monitor and manage natural resources Conduct environmental impact assessments/Analyze climate data.
Utilities	Manage infrastructure and assets.
Real estate	Analyze property values, conduct site selection, and manage property portfolios
Public Sector	Land management / Emergency response planning / Environmental monitoring
Transportation & Logistics	Manage fleets / Improve customer service
Retail	Analyze customer behavior / Conduct market research / Optimize store locations
Agriculture	Manage crop yields / Improve resource management
Healthcare & Public Health	Analyze disease data / Conduct epidemiological research

Why Loading Geospatial Data to BigQuery

- BigQuery as a cloud-based data warehouse, enables low latency and large scale geospatial analysis
- Regardless of geospatial data types, there’s a way to ingest it, store it, analyze it on GCP.
- Enable integration with other data in BigQuery
- Enables geospatial data enabled machine learning models on GCP

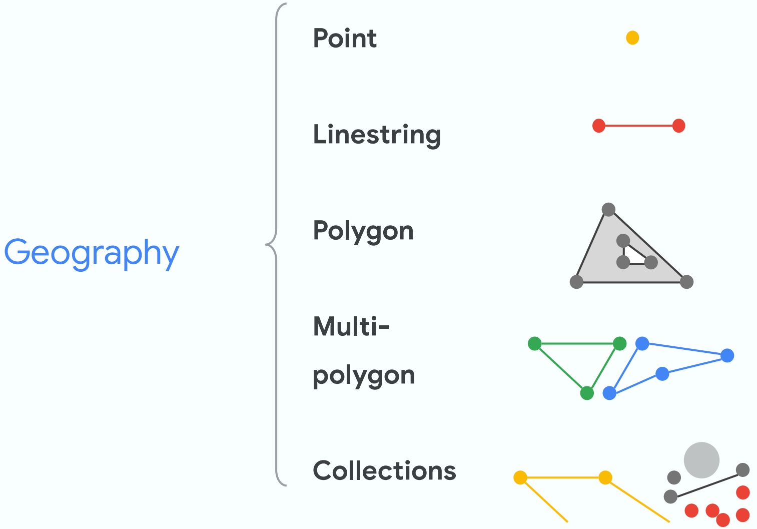
BigQuery Spatial Functions

- ST_CONTAINS(outside_geom,inside_geom) -- true if outside completely surrounds inside
- ST_INTERSECTS(a_geom, b_geom) -- true if intersection of a and b is non-empty
- ST_DISTANCE(a_geom, b_geom) -- shortest distance between two geometries
- ST_AREA(a_geom) -- area in square meters
- ST_LENGTH(a_geom) -- length of line or perimeter geometry

https://cloud.google.com/bigquery/docs/reference/standard-sql/geography_functions

Types of Geospatial Data

Vector (Tabular)

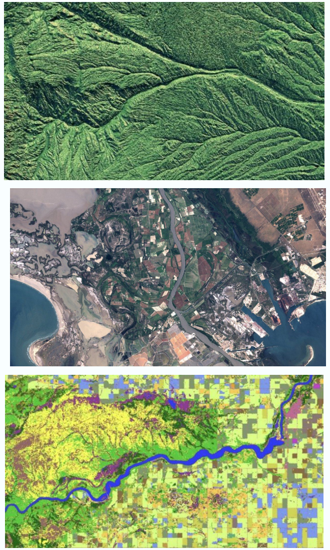


Geography

- Point
- Linestring
- Polygon
- Multi-polygon
- Collections

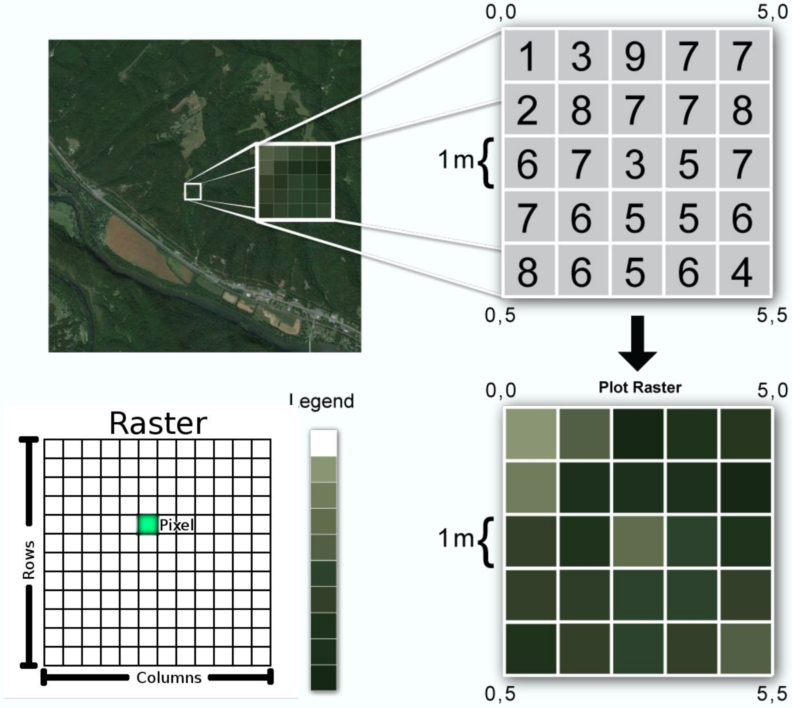
Format: shape, kml, geojson, WKT ...

Raster (Imagery)

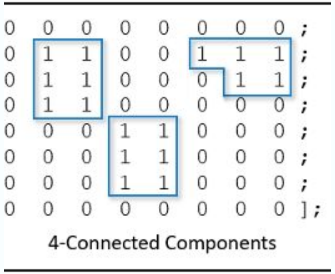


Format: GeoTiff ...

How to load - Challenges

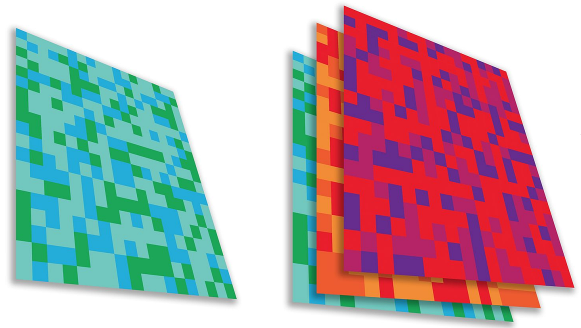


1	3	9	7	7
2	8	7	7	8
6	7	3	5	7
7	6	5	5	6
8	6	5	6	4



```
0   0   0   0   0   0   0   0  ;
0   1   1   0   0   1   1   1  ;
0   1   1   0   0   0   1   1  ;
0   1   1   0   0   0   0   0  ;
0   0   0   1   1   0   0   0  ;
0   0   0   1   1   0   0   0  ;
0   0   0   1   1   0   0   0  ;
0   0   0   0   0   0   0   0  ];
```

4-Connected Components



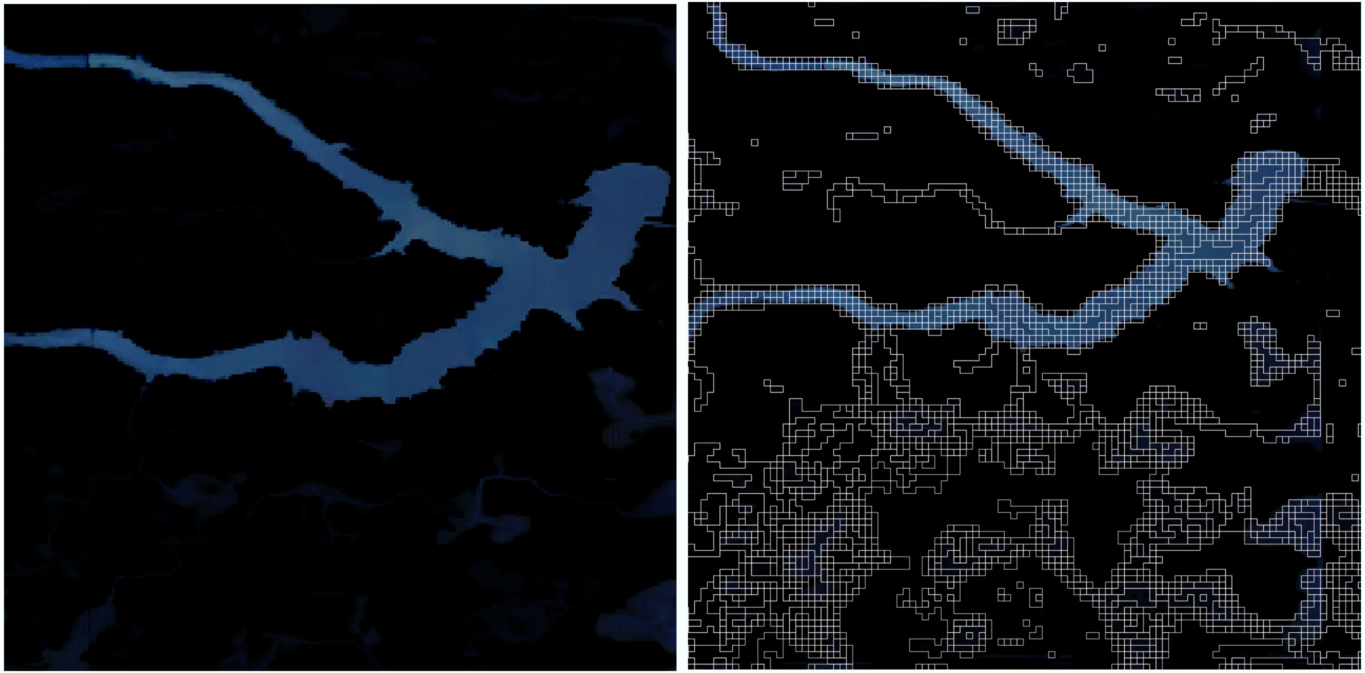
Single Band Raster

Multi Band Raster

Polygonize the high resolution image with many pixels in it is very compute intensive.

A ~15 meter ground resolution one degree tile image could have more than 10million pixels

Geotiff to Polygon



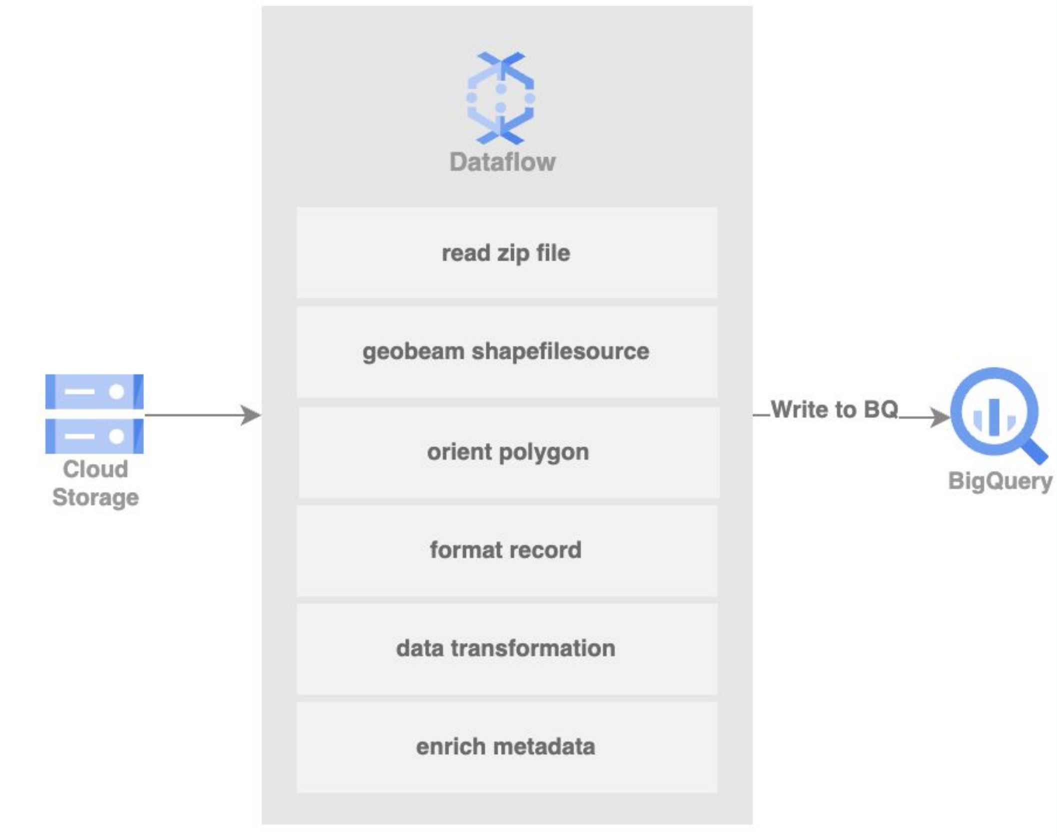
How to Load - 2 ways to load

- Dataflow with GeoBeam library
  - GeoBeam load shapefiles very effectively.
  - Works well with small to medium size images (< ~a few hundreds MB).
  - Great examples
  - Runs fine in the local runner
- Dataflow running gdal commands
  - Can handle complex data processing (retile raster, reproject, polygonizer etc)
  - Better parallelism and memory consumption

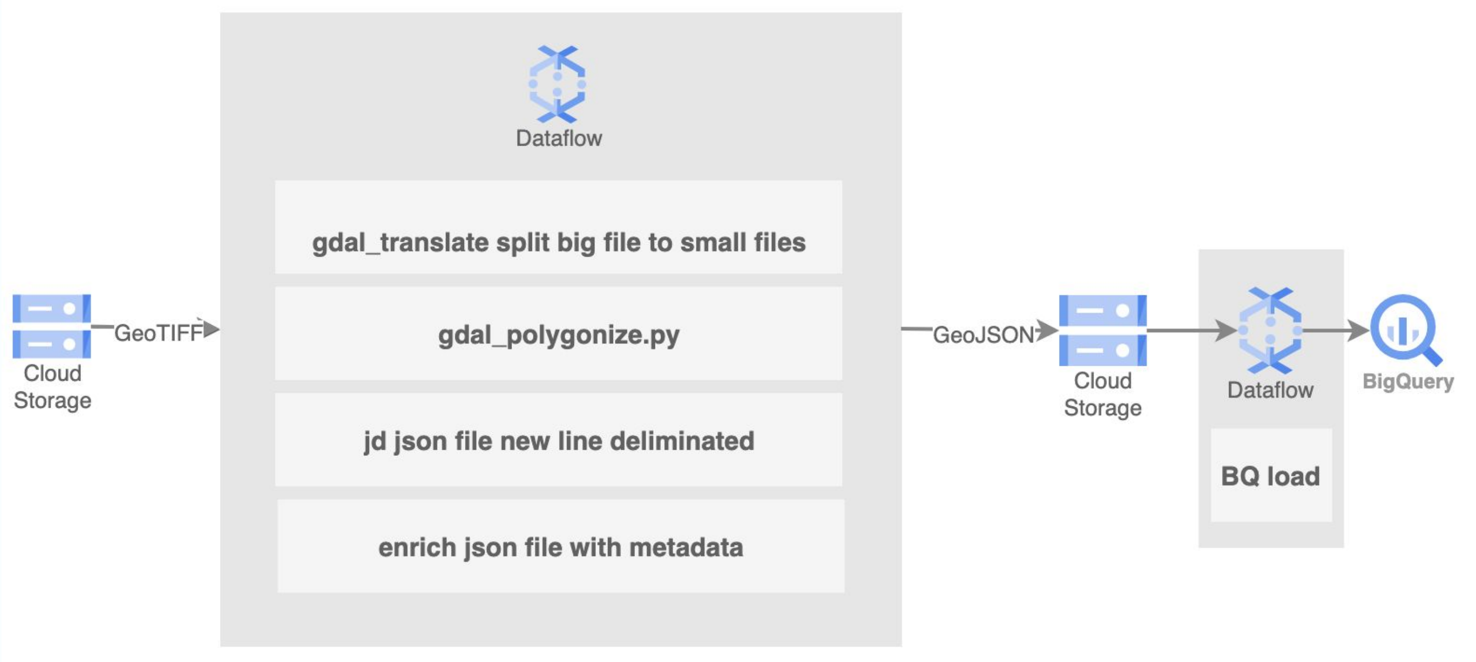
https://github.com/GoogleCloudPlatform/dataflow-geobeam/tree/main/geobeam/examples

https://gdal.org/programs/gdal_polygonize.html

Steps of Loading Shape File with Geobeam



Steps of Loading GeoTiff Files with GDAL



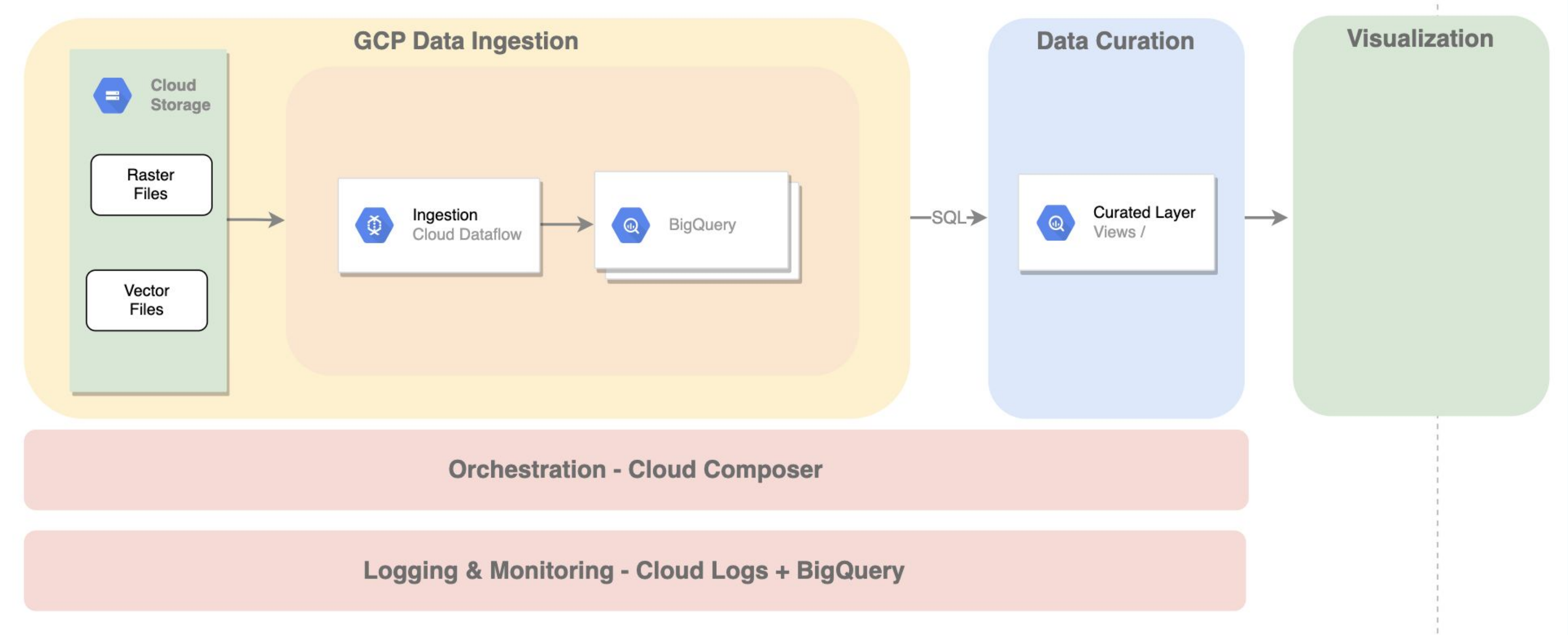
Dataflow
gdal_translate split big file to small files
gdal_polygonize.py
jd json file new line deliminated
enrich json file with metadata
Cloud Storage
GeoTIFF
GeoJSON
Cloud Storage
Dataflow
BQ load
BigQuery

- *gdal_polygonize.py nass.tif -b 1 -f "GeoJSON" nass.geojson*
- *jq -c '.features[]' nass.geojson > nass-nl.geojson*
- *bq load --source_format=NEWLINE_DELIMITED_JSON --json_extension=GEOJSON -autodetect=true geodata.ny_nass nass-nl.geojson*

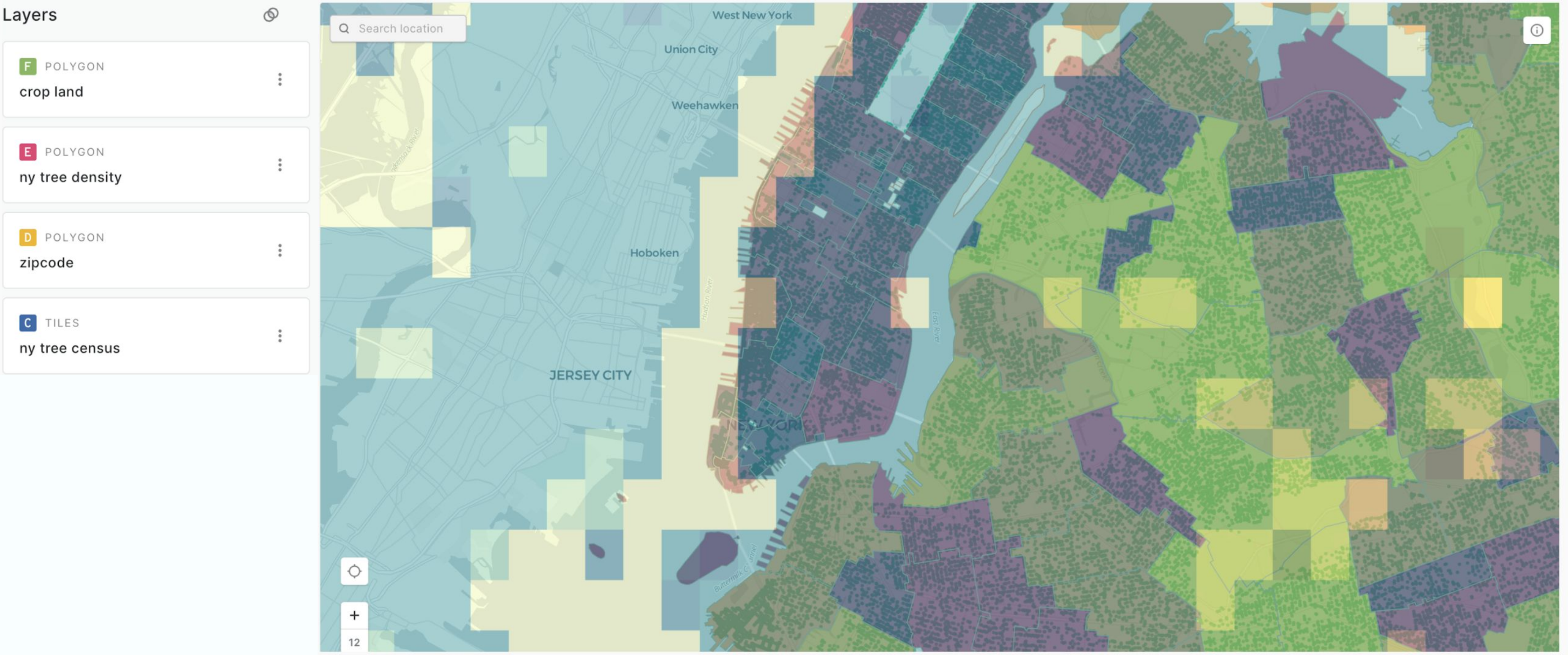
How beam/dataflow helps

- GeoBeam, built on top of Apache Beam
  - Provides a set of FileBasedSource classes that make it easy to read, process, and write geospatial data
  - Provides a set of helpful Apache Beam transforms and utilities that make it easier to process GIS data in Dataflow pipelines.
- Dataflow customized container worker for
  - Easy deployment of 3rd party libraries as you can specify them in build files and dataflow will handle the deployment.
- Use Dataflow as compute and run command lines allows to focus on data handling and not underneath beam complexity
- Easy parallel processing with auto scaling, composer, and Thread Pool Executor while converting geotiff to geojson
- Integrated with GCP for monitoring logging

Reference Architecture



Demo



Layers

F POLYGON
crop land

E POLYGON
ny tree density

D POLYGON
zipcode

C TILES
ny tree census

QUESTIONS?