addresscloud®

# FOSS4GUK 2023
# Geospatial tools for working with Cloud Native Data

Matt Travis
github: mtravis
matt@addresscloud.com

addresscloud

## Introduction

- Been working with geo data for close to 20 years, mainly for local authorities in the UK and Dartmoor National Park
- Currently working at addresscloud as a data analyst / engineer
- Treasurer for OSGEO:UK and helped organise FOSS4GUK the 2019, 2020:Online and 2022:Local events

addresscloud®

Addresscloud was founded in 2015 (and born at FOSS4G!) and is a geocoding and location intelligence service working predominantly within the insurance sector

addresscloud.com

addresscloud

Insurers need to avoid having too much risk in a single location, but what is that location? Many systems use radial accumulation techniques but these do not represent real life risk

addresscloud.com

addresscloud®

# Cloud Native Geospatial

- ❏ Data Formats
- ❏ Data Store
- ❏ Open Source Tools

addresscloud®

# Benefits of using cloud-native data?

- Cloud-Native data formats are structured to be efficiently retrieved from cloud object storage services
- It's faster for users.
- Stream only the data you need that they need for their analysis.
- No need to download and store copies of data (or even DVDs)
- Saves users time and money

addresscloud®

# Data Formats cloud-native data?

addresscloud®

# Flatgeobuf


- Lossless binary format - fast to load and stream
- Works well with large volumes of static data, significantly faster than legacy formats
- Not editable - really for read only and data storage/transfer
- Supported by GDAL, QGIS and Tippecanoe
- Can be directly streamed and used in by Leaflet, MapLibre, etc

addresscloud®

# Geoparquet


- Based on Parquet - CSV for Big Data
- Columnar Data so quicker to read
- Currently in release candidate - v1.0 imminent
- Not editable - really for read only and data storage/transfer
- Used by Overture Maps Foundation to publish data
- Data can be partitioned so it’s quicker to retrieve.

addresscloud®

# Who is involved in GeoParquet?












addresscloud®

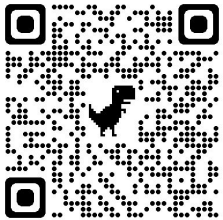
# PMTiles

- PMTiles is a single-file archive format for tiled data. A PMTiles archive can be hosted on a commodity storage platform such as S3
- https://github.com/protomaps/PMTiles

addresscloud®

# Data Store - Overture Maps Foundation (OMF)

- OMF started by big tech companies (AWS, Microsoft, Meta, TomTom) Additional members
- Based on OSM data but other sources being added e.g. MS ML-Buildings.
- Data separated into themes: Admin, Buildings, Places and Transportation
- Data is hosted within S3 buckets as parquets and
- Previously you would have to to grab the whole planet file PBF and then use tools like OSMOSIS to get what you needed.



addresscloud®

# Data Stores - Source Coop

- Collection of datasets maintained by Chris Holmes
- Includes data from Google, Overture and the OS
- Available here: https://beta.source.coop/repositories

addresscloud®

# Tippecanoe


- Builds vector tilesets from large (or small) collections of GeoJSON, FlatGeobuf, or CSV features, like these.
- Developed by Erica Fisher at Mapbox but now maintained by Felt. https://github.com/felt/tippecanoe
- Inputs include FGB, CSV and GeoJSON
- Outputs are MBTiles and recently PMTiles

addresscloud®

# DuckDB

- In memory database engine - SQLite on steroids!
- Uses Arrow under the hood
- Great for working with big data stored as parquet
- Can scan remote data and process from your own machine
- R, Python and many other bindings available
- Extensions - Spatial, Postgres, Excel(?)

addresscloud®

# DuckDB - What is it good for?

## When to use DuckDB

- Processing and storing tabular datasets, e.g. from CSV or Parquet files
- Interactive data analysis, e.g. Joining & aggregate multiple large tables
- Concurrent large changes, to multiple large tables, e.g. appending rows, adding/removing/updating columns
- Large result set transfer to client

## When to not use DuckDB

- High-volume transactional use cases (e.g. tracking orders in a webshop)
- Large client/server installations for centralized enterprise data warehousing
- Writing to a single database from multiple concurrent processes
- Multiple concurrent processes reading from a single writable database

addresscloud®

# DuckDB - Spatial

- Spatial extension - convert to other spatial formats
- Uses GDAL under the hood
- Conforms to the Simple Features for SQL specification from the Open Geospatial Consortium.

addresscloud®

# DuckDB - Spatial

```
install spatial;
load spatial;
COPY (SELECT id, ST_GeomFromWKB(geometry) as geometry
FROM read_parquet('/data/places/*')
WHERE adminLevel = 2 AND
ST_GeometryType(ST_GeomFromWKB(geometry)::geometry)
IN ('POLYGON','MULTIPOLYGON')) TO 'omf-countries.fgb'
WITH (FORMAT GDAL, DRIVER 'flatgeobuf');
```