Building a scaleable tiling service using Amazon API Gateway

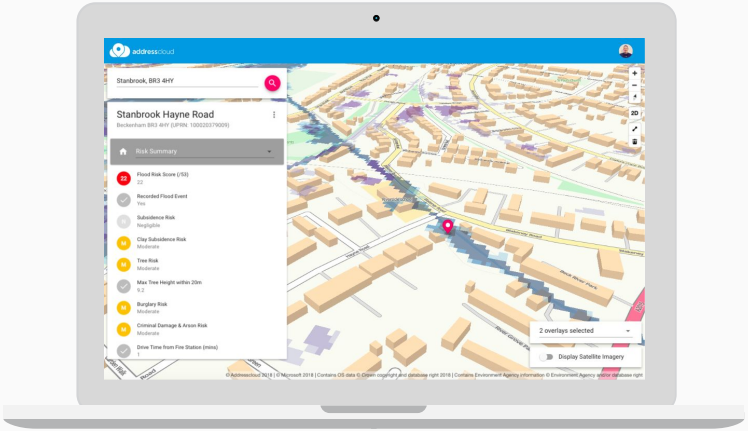
Tomas Holderness, Addresscloud

@iHolderness

addresscloud

Why did we build it?

- Protect commercial data
- Operates at scale (serverless)
- Standardise our APIs
- Improve our developer experience



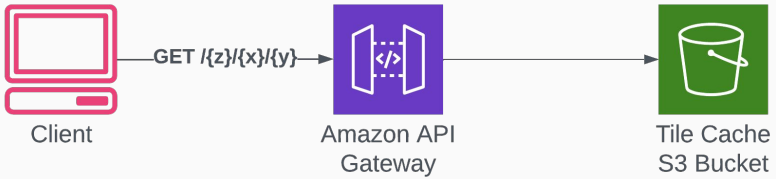
https://github.com/addresscloud/terraform-aws-tile-service

addresscloud

How does it work?

Vector tile service in the cloud

- API Gateway interface
- Slippy tiles spec API
- S3 Bucket storage
- Supports vector tile cache (PBFs)

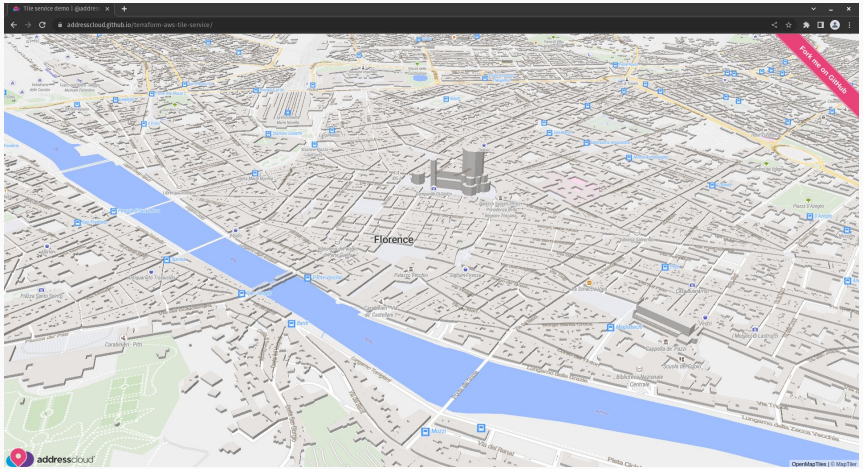


https://github.com/addresscloud/terraform-aws-tile-service

addresscloud

API Gateway

- Forwards HTTP requests to S3
- Protects S3 from public access
- In-built cache available
- Edge deployment via CloudFront
- No Lambda or compute layer

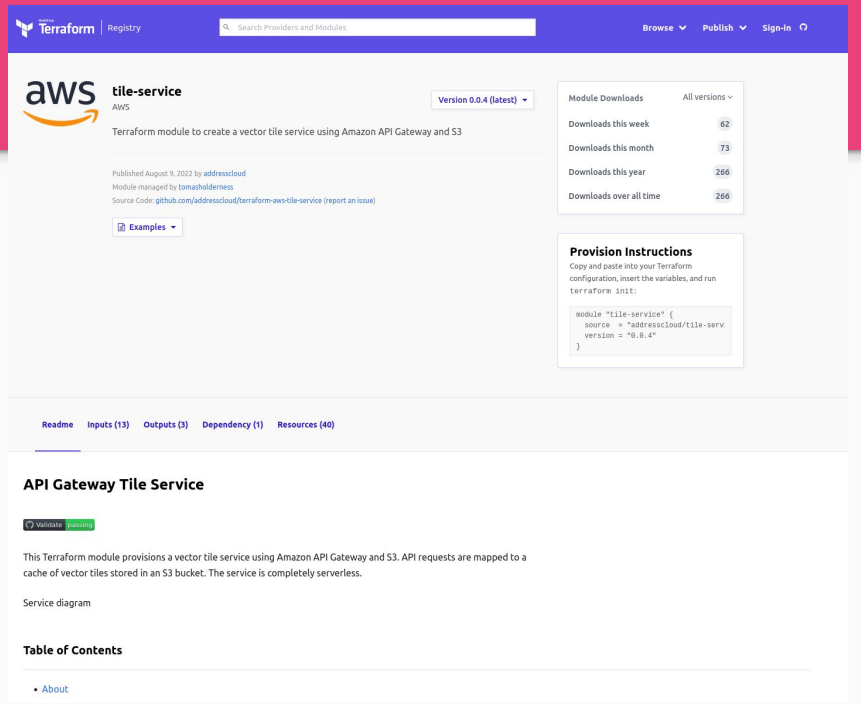


https://github.com/addresscloud/terraform-aws-tile-service

Getting started

Service packaged as Terraform module

1. Terraform run
    - create S3 bucket and API gateway
2. Upload vector tiles and tile.json
3. Instructions and examples on GitHub



https://github.com/addresscloud/terraform-aws-tile-service

addresscloud

Bonus features

- Modify API response using VTL
  - update TileJson on-the-fly
- Supports custom S3 policies
  - Use S3 replication
  - Replicate cross-region/account

The Apache Velocity Project™
http://velocity.apache.org/

https://github.com/addresscloud/terraform-aws-tile-service

addresscloud

Resources

- This talk
  - https://github.com/addresscloud/terraform-aws-tile-service
- Serverless vector tiles with Cloudfront CDN (FOSS4G 2018):
  - https://github.com/addresscloud/serverless-tiles
- Raster tiles with rio-tiler
  - https://cogeotiff.github.io/rio-tiler/