A Case Study Using Kubernetes/Docker Routing, Geocoding, and Basemap Microservices with QGIS and OpenLayers
Boundless provides geospatial tools and services for managing data and building applications.

Chris is a DevOps Engineer primarily focused on supporting applications running on Kubernetes. Interests include: Cloud Computing, Linux, and other Open Source technologies

Joe is the Content Services Lead and helps with: Middle Tier Development and Integrator of fun projects including: GeoServer, OpenLayers, GeoShape, GeoGig

Chris Del Pino
DevOps Engineer
cdelpino@boundlessgeo.com

Joseph Miller
Content Services Lead
jmiller@boundlessgeo.com
Why?

- One interface with lots of options
- One token with access multiple vendors
- Routing and Geocoding for QGIS analysis
- Routing and Geocoding to enhance your OpenLayers experience
One Basemap Interface

Request Structure
{scheme}://(host)/(BCS_api_version)/basemaps/{provider}/{basemap-type}/{z}/{x}/{y}.{image-type-extension}

Metadata
http://api.boundlessgeo.com/basemaps?version=0.1

Examples
http://api.boundlessgeo.com/basemaps/mapbox/satellite-streets/1/2/3.png?version=0.1
https://api.boundlessgeo.com/basemaps/planet/california/1/2/3.png?version=0.1
https://api.boundlessgeo.com/basemaps/boundless/osm/{z}/{x}/{y}.png?version=0.1
One Geocoding Interface

Examples
http://api.boundlessgeo.com/geocode/mapbox/address/x/-77.368115/y/38.905939?version=0.1
http://api.boundlessgeo.com/geocode/mapbox/address/tulsa?version=0.1
-or-
Batch CSV
Response

GeocodeResponse {
    List<GeocodePoint> geocodePoints;
    int errorCode;
    String errorMessage;
    String id;
}

GeocodePoint {
    Double x;
    Double y;
    String candidatePlace;
    String candidateSource;
    String score;
}
One Routing Interface

Examples

http://api.dev.boundlessgeo.io/v1/route/mapbox/originx/{originx}/originy/{originy}/destinationx/{destinationx}/destinationy/{destinationy}

https://api.dev.boundlessgeo.io/v1/route/mapzen/originaddress/2938%20harvest%20glen%20ct%20herndon%20va/destinationaddress/9300%20brookville%20rd%20silver%20spring%20md
https://api.dev.boundlessgeo.io/v1/route/mapbox?waypoints=-77.331398,38.852845%7C2938%20Harvest%20Glen%20Ct%20Herndon%20VA%2020171
-or-
Batch and Matrix CSV
One Routing Interface

RouteResponse {
    int errorCode = 0;
    String errorMessage;
    double distance;
    double duration;
    Geometry geometry;
    List<Leg> legs;
    String id;
    String from;
    String to;
}

Leg {
    double duration;
    double distance;
    List<Step> steps;
}

Step {
    Geometry geometry;
    double distance;
    double duration;
    String instructions;
}
Services Single Sign On

- One Self-Contained, Stateless, and Performant Token
- JSON Web Token + GRPC
- Paid Tiers
  - Mapbox Basemaps, Routing, Geocoding
  - Mapzen Routing, Geocoding
  - Planet Basemaps/Imagery
  - DigitalGlobe Basemaps
- Free As In ...
  - Boundless OSM Basemap
  - GraphHopper OSM routing (coming soon)
  - OSRM OSM routing (coming soon)
  - Nominatim OSM geocoding (coming soon)
Spring Cloud Microservices

Spring Cloud Configuration Server

GitHub | Git Bucket | Local Classpath | Etc.

Eureka, Zuul & Ribbon Interactions

1. REST (Web, CLI, SDKs)
2. Eureka Client
3. Asynchronous Notifications
4. REST calls (client-side load balancing)

Protobuf: Protocol Buffers

kibana

fluentd

GRPC

JUnit

Netflix

Boundless

Spring Cloud

kafka
What now?

• Find a good solution to run these containers
• Good options to choose from:
Why Kubernetes

- Big names behind it and using it
- Great community support
- Able to run on different cloud providers and bare metal
Our Architecture

- Running 3 clusters on AWS:
  - Development, Test, and Production
  - Each has 3 masters, spread across 3 AZ’s
  - Each has 6 nodes, 2 in each AZ
Our Architecture
How do we deploy this?

- **Kops**
  - Kubernetes tool to build and maintain configuration for Kubernetes cluster
  - Able to use this for making updates to cluster
Deployment Workflow

1. Changes pushed
2. Trigger build job
3. Create container image
4. Push to container registry
Deployment Workflow

1. New container image pushed
2. Trigger deploy job
3. Deploy job
4. Deploy to cluster
Storage

- Most of our containers are stateless
- With the exception of Postgres and Kafka
- Using EFS to store data
Monitoring

- Sysdig for checking service availability
- Assertible for validating api endpoints
Cluster Upgrade

• We can upgrade existing clusters to newer version of Kubernetes, but don’t want to affect any users
• Simply spin up new clusters, test, and cut over to new ones by updating DNS
What next?

- Horizontal pod auto scaling
- Auto scaling of infrastructure
- Federation

- On premises/air gapped with:
  - Pelias and Nominatim Geocoding
  - GraphHopper and OSRM Routing
  - Custom Routing with PgRouting
  - Custom OSM Basemaps With GeoServer EC