Open Science Platform for Disaster Risk Reduction

Joost van Ulden, William Chow, Drew Rotheram-Clarke, Damon Ulmi, Anthony Fok

Table of Contents

1. Executive Summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2. Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3. Platform Requirements

To support the Pathways activities and deliverables a platform with the following features and functions were desired:

- Collaborative development of science assets and related documentation
- Centralized access to the science outputs
- Data and related services to align with industry best practices and standards

The operational context of the platform required that it was to be aligned with Government of Canada policies, directives, and standards. These included:

- Government of Canada Policy on Service and Digital:
  - Government of Canada Directive on Service and Digital
    - Mandatory Procedures on Application Programming Interfaces
  - Government of Canada Standard on Web Accessibility
  - Government of Canada Standard on Web Interoperability
  - Government of Canada Standard on Web Usability
- NRCan Open Science Initiative
- NRCan Scientific Integrity Policy

3.1. Principles

The principles under which the platform would operate were established as:

- Transparent
- Open
- Collaborative
- Accessible
- Robust

4. Architecture

Do we need to do some diagram to show the flow?

The OpenDRR platform extract-transform-load (ETL) pipeline consists of several open source technologies, namely PostgreSQL with PostGIS extension, Python, ElasticSearch/Kibana and PyGeoAPI. Each application is containerized using separate Docker containers which are orchestrated using Docker-Compose.

Source data for this project includes the National Human Settlement Layers (physical exposure and social vulnerability), the National Seismic Risk model for Canada (CanadaSRM2, probabilistic), Canada's National Earthquake Scenario Catalogue (deterministic), and boundary geometries are stored on the Open Disaster Risk Reduction Platform (OpenDRR) GitHub site using a large file storage service called Git LFS. The stack of technologies can be configured to run on a local machine with PostgreSQL and ElasticSearch environments running locally or can be configured to communicate with cloud deployments of these services. For example, the stack can be configured to communicate simultaneously with a PostgreSQL database on Amazon Web Services (AWS) Relational Database Service (RDS) and a Containerized ElasticSearch environment on an AWS Elastic Container Service (ECS). A common use case is to run the ETL process on a local desktop or AWS Elastic Compute Cloud (EC2) with a local containerized PostgreSQL environment and a cloud deployed ElasticSearch environment which can support a public or private application programming interface (API).

The ETL process currently relies on a main shell script titled 'add_data.sh' which downloads the required source data files including comma separated value (csv), and GeoPackages (gpkg) into the local filesystem and performs necessary transformations to load the data into a PostgreSQL database. The script is written to be flexible enough to load any number of conformant deterministic earthquake scenarios and national seismic risk model data? and allows some flexibility in the contents of the input data as long as a minimal number of required fields are present. Once the required source data has been loaded into the database, a number of Structured Query Language (SQL) scripts help transform the data into a framework of meaningful earthquake risk indicators which in turn gets pushed to ES/Kibana ??? .

Discuss the postgresql database? Briefly? Schemas? The PostgreSQL database is created and has

PostGIS extension enabled to allow for spatial data queries. The source datas are loaded into their respected schema names and the results are generated into their respected schemas with results_ prefix.????

Results are calculated and/or aggregated at different spatial scales dependent on the data. For National Human Settlement Layers the results are in settled areas. The probabilistic and deterministic data are calculated at the building level and aggregated up to the settled area and census subdivision levels.

These results are aggregated at several different spatial scales, building, sauid, and csd level. The building level is the finest spatial resolution and aggregates all buildings of simmilar construction type for a given neighborhood. The Sauid level of aggregation groups all building types across the whole neighbourhood polygon. The Census Sub-divison (CSD) aggregation is a coarse aggregtion of all assets within a given CSD as defined by Statistics Canada.

5. Lessons Learned

5.1. Open Source, Open Data, Open Science, Open Process

- (May seem scary at first?) but many benefits
- full openness, transparency, accountability
- GitHub platform offers a lot of features and flexibility that we need
- Industry best practices
- Mutually beneficial relationship with GEM (OpenQuake)
- pygeoapi
- etc.

5.2. Security

Example: Apache Log4j vulnerability

- We got confirmation from ITSEC that our ES endpoint was attacked. Since we were running 7.12.0 the attack failed. Lesson: keep your software up to date. (Credit: Joost,)
  - Mitigate Log4j2 / Log4Shell in Elasticsearch
- (a little bit about) keeping up-to-date vs. compatibility-breaking changes (e.g. Python 3.8 to 3.9); be prepared for extra work hours (manpower?) to resolve compatibility issues during upgrade…

5.3. [optional] The use of Agile principles on GitHub; benefits and challenges

5.4. [optional] opendrr.github.io got falsely flagged as phishing or malicious

- CIRA, Google, VirusTotal; took a while to get off e.g. Microsoft SmartScreen
- Possibly due to the use of non Canada.ca template on a non .ca domain?
- https://github.com/OpenDRR/opendrr/issues/122

6. Future Development

6.1. Risk Profiler (v1)

- Expected to be published before April 2022 (kudos to HabitatSeven)

6.2. Increase in scope: Import and process other datasets besides earthquake

- Flood: https://github.com/NRCan/CanFlood (Open-source flood risk modelling toolbox)
- Wildfire
- Mudslides

6.3. Continued automation, optimization, etc.

- Increased use of GitHub Actions workflows
- Full CI (Continuous Integration, Continuous Delivery)
- Cost-savings (e.g. move off Git LFS wherever possible; cheaper alternatives…)
- More automated testing, to catch errors as early as possible (preferrably at the pull request stage)
- Test-driven development (TDD) / Bahaviour driven development (BDD)?

6.4. Promote the project

- as a model with example that that other groups (countries/governments/companies etc.) can use
- (as open-source project) invite participation from open-source, scientific community, and the general public

- through web site(s), seminars/talks, publications

6.5. More…