Nominatim Apptainer Container

- Nominatim SQLite Geocoding

Version: Nominatim 5.1.0

Base: Ubuntu 24.04

Output: Portable SQLite Database

Summary:

A self-contained geocoding service that:

- Imports OpenStreetMap data using PostgreSQL
- Converts to SQLite for portability
- Includes Iowa OSM data + US metadata
- Runs without external dependencies

Use Case: Address lookup, reverse geocoding, location search

Why PostgreSQL During Build?

No Alternative (yet): Nominatim's import pipeline is PostgreSQL-only

Convert to SQLite?

Benefits:

- Single File
- Zero Dependencies: No PostgreSQL daemon needed
- May be easier to have multiple nodes batch on Argon.
- Avoids having to deal with multiple namespaces post build
    - pguser, nominatim, and www-data

Trade-off:

- No concurrent writes (but our geocoding is read-only!)

/app vs /nominatim:

During the build we are mainly operating inside of two directories:

/app/

```
┌─────────────────────────────────────────────────────────────────┐
│ /app - Build-time workspace                                     │
├─────────────────────────────────────────────────────────────────┤
│ • Temporary files needed only during container build            │
│ • PostgreSQL database and runtime files                         │
│ • Configuration files                                           │
│ • Downloaded OSM and metadata files                             │
│ • Removed in cleanup phase to reduce final image size           │
└─────────────────────────────────────────────────────────────────┘
```

/nominatim/

```
┌─────────────────────────────────────────────────────────────────┐
│ /nominatim - Runtime workspace                                  │
├─────────────────────────────────────────────────────────────────┤
│ • Nominatim project directory                                   │
│ • Final SQLite database (nominatim.sqlite)                      │
│ • .env configuration file                                       │
│ • Kept in final container for geocoding queries                 │
└─────────────────────────────────────────────────────────────────┘
```

Container File Structure

```
nominatim-container/
│
├── nominatim.def                    # Main container definition file
│
├── conf.d/                          # PostgreSQL & Nominatim configuration
│   ├── postgresql.conf              # PostgreSQL performance settings
│   ├── pg_hba.conf                  # Database authentication rules
│   ├── postgres-import.conf         # Import-time optimizations
│   ├── postgres-tuning.conf         # Runtime performance tuning
│   └── env                          # Nominatim environment variables
│
└── nominatim_project/               # Data files and scripts
    ├── config.sh                    # Nominatim configuration script
    ├── iowa-latest.osm.pbf          # OpenStreetMap data (50-200MB)
    ├── secondary_importance.sql.gz  # Additional search ranking
    ├── us_postcodes.csv.gz          # US ZIP code database
    ├── wikimedia-importance.csv.gz  # Search result weights
    └── tiger-nominatim-preprocessed-latest.csv.tar.gz  # US Census addresses
```

Required Files

Configuration (`conf.d/`):

- `postgresql.conf` - Performance tuning
- `pg_hba.conf` - Authentication
- `postgres-import.conf` - Import optimizations
- `env` - Environment variables

Required Files

Data (`nominatim_project/`):

- `iowa-latest.osm.pbf` - OpenStreetMap data
- `us_postcodes.csv.gz` - ZIP codes
- `wikimedia-importance.csv.gz` - Search ranking
- `tiger-nominatim-preprocessed-latest.csv.tar.gz` - US addresses

Build Process: Phase 1

System Setup

APT Packages:

```
apt-get install \
  postgresql-postgis \
  osm2pgsql \
  python3-pip \
  gdal-bin
```

Python Packages:

```
pip install --break-system-packages \
  nominatim-db==5.1.0 \
  nominatim-api==5.1.0
```

Phase 2: User & Database Setup

- Create non-root PostgreSQL User
- Initialize Database

```
useradd -m -s /bin/bash pguser

su pguser -c "initdb -D $PGDATA --auth=trust"
```

Why --auth=trust?

- Ran into errors during build with users
- Only relevant during build time so no real network exposure when doing geocoding.

Database Users:

Nominatim requires the following users:

```
createuser --superuser pguser              # Owns PostgreSQL database
createuser --no-superuser nominatim        # Read-only
createuser --no-superuser www-data         # Web accessor
```

Phase 3: Nominatim Import

```
nominatim import \
  --osm-file /app/data/iowa-latest.osm.pbf \
  --project-dir /nominatim \
  --threads 4
```

- People say to use ~1 thread per 2GB of RAM. For my computer this is fine.

There is a `.env` file which Nominatim uses for build variables.
This along with postgresql conf will probably need to be optimized.

Phase 4: Metadata

Import to Nominatim

- US Postcodes (ZIP code geocoding)
- Wikimedia Importance (improves result ranking)
- Tiger-Line Data (address interpolation)

Phase 5: SQLite Convert

Experimental

```
nominatim convert -o /nominatim/nominatim.sqlite
```

Converts placex table, search indexes, address hierarchies, and metadata tables

Phase 6: Cleanup

- Stop PostgreSQL
- Remove Build Artifacts

Mainly done to reduce the size of the resulting container to just have SQLite database.

Offline Container Build Strategy:

Problem:

Building containers offline (Argon) requires all dependencies be pre-downloaded and no elevated priviledges.
- APT Packages
- Python wheels
- Base Docker images

Solution: Two Stage Build

To run priveledged workflows on Argon we need an apptainer container.

So to build the nominatim.sif we will first create a dependency container which will be used as a cache when building the actual container.

1. `resources.def` container which packages all dependencies
2. `builder.def` Container which uses resources as a base to create a nominatim.sif

Overview:

```
┌─────────────────────────────────────────────────────────────┐
│ Stage 1: resources.def (Build requires internet)            │
├─────────────────────────────────────────────────────────────┤
│ Bootstrap: docker                                           │
│ From: ubuntu:24.04                                          │
│                                                             │
│ • Downloads & installs all APT packages                     │
│ • Downloads & installs all Python wheels                    │
│ • Caches dependencies for offline use                       │
│                                                             │
│ Output: resources.sif                                       │
└─────────────────────────────────────────────────────────────┘
                             and
┌─────────────────────────────────────────────────────────────┐
│ Stage 2: builder.def (Offline)                              │
├─────────────────────────────────────────────────────────────┤
│ Bootstrap: localimage                                       │
│ From: resources.sif                                         │
│                                                             │
│ • Uses pre-installed dependencies                           │
│ • Imports OSM data                                          │
│ • Builds PostgreSQL database                                │
│ • Converts to SQLite                                        │
│                                                             │
│ Output: nominatim.sif                                       │
└─────────────────────────────────────────────────────────────┘
```

`builder.def` Demo

```
Bootstrap: docker
From: ubuntu:24.04

%files
  offline_deps/apt_packages/*.deb /opt/apt_packages/
  offline_deps/python_wheels/*.whl /opt/python_wheels/

%post
  # Install all dependencies
  dpkg -i /opt/apt_packages/*.deb
  pip install --no-index --find-links=/opt/python_wheels \
    nominatim-db nominatim-api osmium psycopg aiosqlite
```

`application.def` Demo

```
Bootstrap: localimage    # Uses local .sif file
From: builder.sif

%files
  nominatim_project/iowa-latest.osm.pbf /app/data/

%post
  # All dependencies already installed
  nominatim import --osm-file /app/data/iowa-latest.osm.pbf
  nominatim convert -o /nominatim/nominatim.sqlite
```

Will be very similar to nominatim.def but instead pulls dependencies from builder.def.