USING NIX TO BUILD AND DISTRIBUTE A GEOSPATIAL SOFTWARE

FOSS4G Europe 2024, Tartu, Estonia, 2024

Ivan Mincik, @imincik, Nix Geospatial Team

A talk about **inovative ways of running a software** on a computer.

TYPE OF GEOSPATIAL SOFTWARE

- Core libraries (C, C++) + addons
- Python/R/Other libraries
- Desktop apps + plugins/addons
- Databases + extensions
- Web services + plugins
- CLIs

USER EXPECTATIONS

- Run **any** of software on **any machine**
- Run any software **without breaking another one**
- Freedom to decide **when to update** or **not to update** at all

USER EXPECTATIONS

- Be able to **reproduce installation** on **any machine** at **any time**

ADVANCED USER EXPECTATIONS

- Have a **full control of whole dependency graph**
- Be able to **customize the software** including **dependencies**

DEVELOPER EXPECTATIONS

- Be able to **start hacking** without too much effort
- Have a **quick** and **reliable feedback loop** with the users and other developers

FROM SOURCE TO USER

Source code -> **Package** -> **Repository** -> **User**

(devs) -> (package maintainers) -> (users)

Deb, RPM, PPA, AUR, Conda, Hombrew, Flatpak, AppImage, Snap, Pip, Npm, OSGeo4W, Chocolatey, Docker, ...

PROBLEMS

- **Delivery** of software **depends on package maintainers**

PROBLEMS

- Big **duplication of package maintainers work**
- All solutions are **missing some packages**

PROBLEMS

- **Breakages when combining** multiple solutions/sources
- **Software inconsistencies** when combining multiple solutions/sources

PROBLEMS

- Some solutions are **platform/distro specific**
- Some solutions **don't allow to install multiple versions** of software
- Some solutions depend on **proprietary** components
- Some solutions are **just workarounds** (containers for running GUI)

NIX

Ground-up design, not following known **broken patterns.**

(E. Dolstra, PhD theses, 2006)

FEATURES OF NIX

- **Reproducibility** as a core feature:
  - between multiple builds
  - between multiple machines
  - over the time
- **Full control** over whole **dependency graph**
- **No software conflicts**
- Runs on **all Linux, Mac, Win WSL 2**

FEATURES OF NIX

- Per-project **isolated environments** for **all types of software**
- Software versions are **locked** and **updated when requested**
- Great **customization support**
- Dozens of other **very unique features**

WHAT IS NIX ?

- **Nix** - the package manager/build system
- **Nix** - the language
- **Nixpkgs** - the largest packages repo
- **Nix modules system** - the declarative configuration management
- **NixOS** - the unique operating system

+ dozens of other community projects (Home Manager, ..)

DEMO

(magic Nix commands)

NIX FOR USERS

RUN GRASS

- No GRASS installed

```
$ grass

The program 'grass' is not in your PATH.
```

- Run GRASS from Internet repo (geospatial-nix)

```
$ nix run github:imincik/geospatial-nix#grass -- --version

GRASS GIS 8.3.2
```

RUN GRASS (OTHER VERSION)

- Run GRASS in other version

```
$ nix run github:imincik/geospatial-nix/58d8cff#grass -- --version

GRASS GIS 8.3.1
```

INSTALL GRASS

- Install GRASS

```
$ nix profile install github:imincik/geospatial-nix#grass
```

SHELL ENVIRONMENT

- No QGIS installed

```
$ qgis

The program 'qgis' is not in your PATH.
```

- Create shell environment with GRASS and QGIS

```
$ nix shell github:imincik/geospatial-nix#{grass,qgis}

$ grass --version
GRASS GIS 8.3.2

$ qgis --version
QGIS 3.36.3-Maidenhead 'Maidenhead' (exported)
```

EXIT SHELL ENVIRONMENT

- Exit shell environment

```
$ exit  # no grass and qgis anymore :(
```

NIX FOR ADVANCED USERS

GRASS CUSTOMIZATION (OVERRIDE DEPENDENCIES)

- Build with development version of GDAL

```
$ nix run -L --impure --expr \
  "let \
    f = builtins.getFlake "github:imincik/geospatial-nix"; \
    p = f.packages.x86_64-linux; \

  in p.grass.override { gdal = p.gdal-master; }"
```

```
$ g.version -e

GRASS 8.3.2 (2024)
PROJ: 9.4.1
GDAL/OGR: 3.10.0dev
GEOS: 3.12.1
SQLite: 3.43.2
```

GRASS CUSTOMIZATION (MODIFY BUILD CONFIGURATION)

- Build without X support

```
$ nix run -L --impure --expr \
  "let \
    f = builtins.getFlake "github:imincik/geospatial-nix"; \
    p = f.packages.x86_64-linux; \

  in p.grass.overrideAttrs (old: { configureFlags = old.configureFlags ++ [ "--without-x" ]; })"
```

GRASS CUSTOMIZATION (ADD PATCH)

- Build with patch from PR

```
p.grass.overrideAttrs (_: { patches = [
    n.fetchpatch {
      url = "https://github.com/<OWNER>/<REPO>/commit/<GIT-REVISION>.patch";
      hash = "";
    }
]})
```

GRASS IN CONTAINER

- Nix is a better Docker image builder than Docker

```
$ nix build --impure --expr \
  "let \
    f = builtins.getFlake "github:imincik/geospatial-nix"; \
    p = f.packages.x86_64-linux; \
    n = f.inputs.nixpkgs.legacyPackages.x86_64-linux; \

  in n.dockerTools.buildImage \
    { name = "grass"; config.Cmd = [ "${p.grass}/bin/grass" "--version" ]; }"
```

```
$ docker load < ./result

$ docker run grass:<TAG>

GRASS GIS 8.3.2
```

NIX FOR DEVELOPERS

PR: ADD NIX FILES TO THE PROJECT

- flake.nix
- flake.lock
- package.nix

GRASS DEVELOPMENT ENVIRONMENT

- Get GRASS source code

```
$ git clone https://github.com/OSGeo/grass.git
```

- Build GRASS from source

```
$ nix develop

Welcome to a GRASS development environment !
Build GRASS using following commands:

 1.   ./configure --prefix=$(pwd)/app
 2.   make -j16
 3.   make install

...
```

```
$ ./configure

$ make -j8
```

RUN GRASS DIRECTLY FROM SOURCE CODE

- Run latest development version

```
$ nix run github:OSGeo/grass/#grass -- --version
```

- Run specific git revision

```
$ nix run github:OSGeo/grass/<GIT-REVISION>#grass -- --version
```

- Run from PR

```
$ nix run github:OSGeo/grass/<PR-BRANCH>#grass -- --version
```

INSTALL GRASS DIRECTLY FROM SOURCE CODE

- Run latest development version

```
$ nix profile install github:OSGeo/grass/#grass -- --version
```

- Run specific git revision

```
$ nix profile install github:OSGeo/grass/<GIT-REVISION>#grass -- --version
```

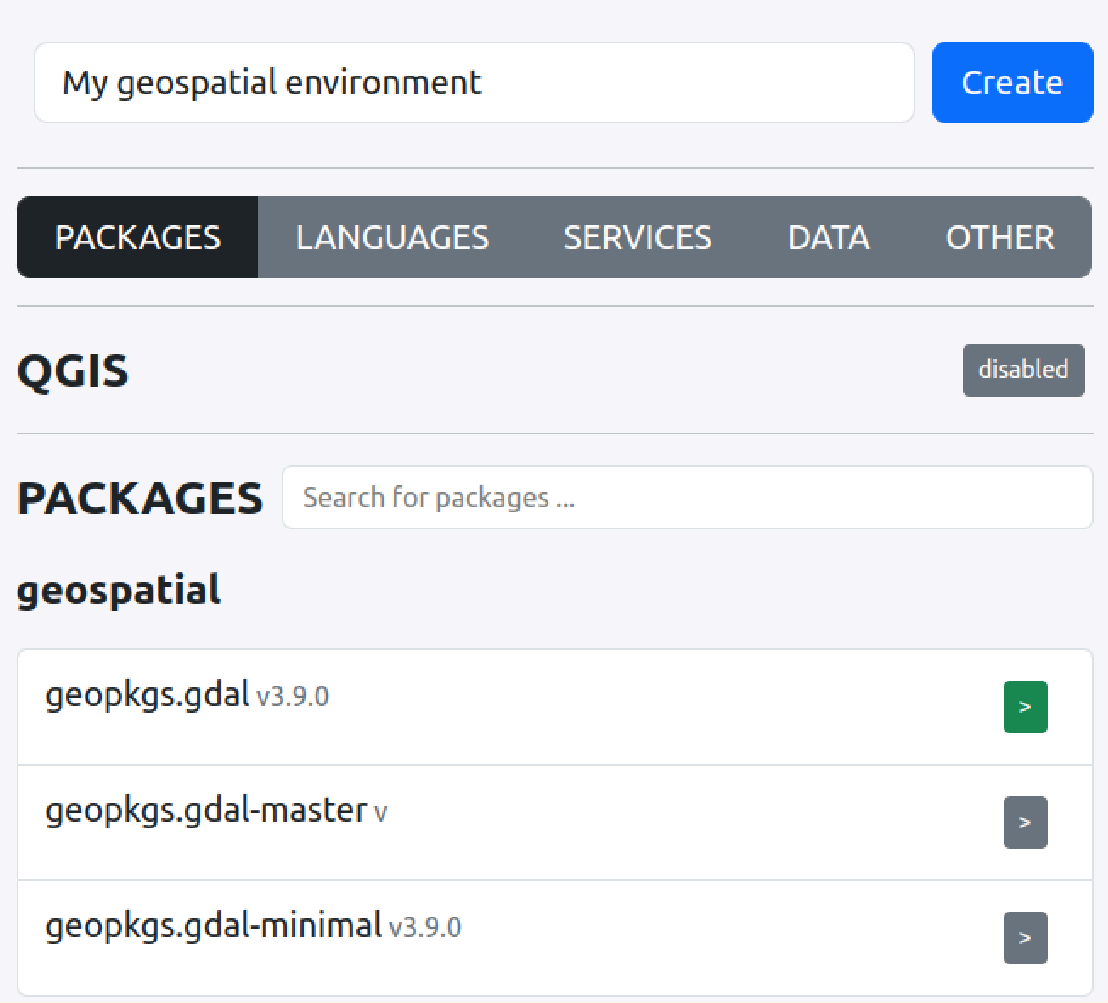
- Run from PR

```
$ nix profile install github:OSGeo/grass/<PR-BRANCH>#grass -- --version
```

INTERESTED ?

- Matrix: #geospatial:nixos.org
- Email: ivan.mincik@gmail.com

GEOSPATIAL NIX.TODAY



My geospatial environment

QGIS

PACKAGES

geospatial

geopkgs.gdal v3.9.0 >

geopkgs.gdal-master v >

geopkgs.gdal-minimal v3.9.0 >

GEOSPATIAL NIX.TODAY

https://geospatial-nix.today/

NIX DOCUMENTATION

https://nix.dev/