

What is this guy thinking about?



How many slides does Devrim have this time?





EDB

Postgres® for the AI Generation

The process of RPM packaging in the PostgreSQL ecosystem and EDB's contributions

Devrim Gündüz

Principal Engineer @ EDB

RPM packager @ PGDG

Part time DJ, serial traveller

<https://www.linkedin.com/in/devrimgunduz/>

How it started?



Quick history

2002 - "I'd like to rebuild SRPMs for my Red Hat Linux 6.X box"

2002: "Here are the binary RPMs on my website, if you want to download"

2004:

RPMS, time, and other things.

From: Lamar Owen <lowen(at)pari(dot)edu>
To: PostgreSQL-development <pgsql-hackers(at)postgresql(dot)org>
Subject: RPMS, time, and other things.
Date: 2004-10-25 17:34:36
Message-ID: 200410251334.36550.lowen@pari.edu
Views: [Whole Thread](#) | [Raw Message](#) | [Download mbox](#) | [Resend email](#)
Thread: 2004-10-25 17:34:36 from Lamar Owen <lowen(at)pari(dot)edu> 
Lists: [pgsql-hackers](#)

It is with some sadness I write this message to this group of talented people that I have come to think of as almost an extended family.

As release time for PostgreSQL 8.0 comes near, I find myself unable to spend as much time on RPMs as I once did; frankly, I find myself not quite as interested in doing so as I once did.

The situation in which I jumped feet first back in July of 1999 was one where RPMs for Red Hat releases came out infrequently, and where upgrading RPMs was fraught with peril. Times have changed: Red Hat now employs Tom Lane full-time to work on PostgreSQL; Red Hat RPM releases, handled by Tom, are both frequent and well-managed; another community maintainer, Devrim Gunduz, has stepped up to the plate and has managed the release of several RPMs at this point. These are good changes.



Quick history

2007: pgrpms.org

2011: yum.postgresql.org

2017: zypp.postgresql.org

2018: yum-archive.postgresql.org

2019: zypp-archive.postgresql.org

2024: dnf-debuginfo.postgresql.org & zypp-debuginfo.postgresql.org



The infrastructure



Infrastructure

- Main git repo: <https://git.postgresql.org/gitweb/?p=pgrpms.git>
 - Mirror: <https://github.com/pgdg-packaging/pgdg-rpms/>



Infrastructure

- Main git repo: <https://git.postgresql.org/gitweb/?p=pgrpms.git>
 - Mirror: <https://github.com/pgdg-packaging/pgdg-rpms/>
- Issue tracker: <https://github.com/pgdg-packaging/pgdg-rpms/issues>
 - Formerly: <https://redmine.postgresql.org/projects/pgrpms/issues>



Infrastructure

- Main git repo: <https://git.postgresql.org/gitweb/?p=pgrpms.git>
 - Mirror: <https://github.com/pgdg-packaging/pgdg-rpms/>
- Issue tracker: <https://github.com/pgdg-packaging/pgdg-rpms/issues>
 - Formerly: <https://redmine.postgresql.org/projects/pgrpms/issues>
- Mock builds infra: <https://github.com/pgdg-packaging/pgdg-mock-configs>



Infrastructure

- Main git repo: <https://git.postgresql.org/gitweb/?p=pgrpms.git>
 - Mirror: <https://github.com/pgdg-packaging/pgdg-rpms/>
- Issue tracker: <https://github.com/pgdg-packaging/pgdg-rpms/issues>
 - Formerly: <https://redmine.postgresql.org/projects/pgrpms/issues>
- Mock builds infra: <https://github.com/pgdg-packaging/pgdg-mock-configs>
- SRPM macros: <https://github.com/pgdg-packaging/pgdg-srpm-macros>



Infrastructure

- A beefy build server for x86_64 builds
 - 128 core, 1TB RAM, helping datacenter to warm up a bit



Infrastructure

- A beefy build server for x86_64 builds
 - 128 core, 1TB RAM, helping datacenter to warm up a bit
- AWS instances for arm64 builds



Infrastructure

- A beefy build server for x86_64 builds
 - 128 core, 1TB RAM, helping datacenter to warm up a bit
- AWS instances for arm64 builds
- OSU Open Source Lab for ppc64le builds



Infrastructure

- A beefy build server for x86_64 builds
 - 128 core, 1TB RAM, helping datacenter to warm up a bit
- AWS instances for arm64 builds
- OSU Open Source Lab for ppc64le builds
- A beefy laptop for initial packaging (a.k.a. work laptop)



Infrastructure

- A beefy build server for x86_64 builds
 - 128 core, 1TB RAM, helping datacenter to warm up a bit
- AWS instances for arm64 builds
- OSU Open Source Lab for ppc64le builds
- A beefy laptop for initial packaging (a.k.a. work laptop)
- download.postgresql.org for distributing packages



Infrastructure

- A beefy build server for x86_64 builds
 - 128 core, 1TB RAM, helping datacenter to warm up a bit
- AWS instances for arm64 builds
- OSU Open Source Lab for ppc64le builds
- A beefy laptop for initial packaging (a.k.a. work laptop)
- download.postgresql.org for distributing packages
- Magnus for everything!



Platform support



A huge number of platforms

- RHEL (and its derivatives):
 - 10.2/10.1 (10.3 is due on November 2026)
 - 10.0 was EOLed in 15 Oct 2026
 - 9.8, 9.7(9.9 is due on November 2026)
 - 9.6 was EOLed in 15 Oct 2026
 - x86_64, aarch64 (arm64), ppc64le



A huge number of platforms

- RHEL (and its derivatives):
 - 10.2/10.1 (10.3 is due on November 2026)
 - 10.0 was EOLed in 15 Oct 2026
 - 9.8, 9.7(9.9 is due on November 2026)
 - 9.6 was EOLed in 15 Oct 2026
 - x86_64, aarch64 (arm64), ppc64le
- Fedora: 43 and 44 (45 testing RPMs are also out)
 - x86_64
 - latest two versions only (follows Fedora lifecycle:
<https://docs.fedoraproject.org/en-US/releases/lifecycle/>)



A huge number of platforms

- SLES: 15.7, 16.0 (15.6 is EOLed in 15 Oct 2026)
 - 16.1 is due in November 2026)
 - x86_64



A huge number of platforms

- SLES: 15.7, 16.0 (15.6 is EOLed in 15 Oct 2026)
 - 16.1 is due in November 2026)
 - x86_64
- OpenSuSE Leap 16.0
 - x86_64



A huge number of platforms

- SLES: 15.7, 16.0 (15.6 is EOLed in 15 Oct 2026)
 - 16.1 is due in November 2026)
 - x86_64
- OpenSUSE Leap 16.0
 - x86_64
- Amazon Linux 2023
 - x86_64, aarch64 (arm64)



Packaging process



Creating spec files (and patches)

- I create initial spec file on my local box (Fedora latest stable)



Creating spec files (and patches)

- I create initial spec file on my local box (Fedora latest stable)
- Do initial build and installation tests locally



Creating spec files (and patches)

- I create initial spec file on my local box (Fedora latest stable)
- Do initial build and installation tests locally
- Do mock builds:
 - Find potentially missing build dependencies
 - Find potentially missing runtime dependencies
 - Find potentially compilation issues
 - Find potential broken upgrade paths
 - Install the previous version and upgrade the package (and extension, where applicable)
 - On an old distro (RHEL 8), non-RHEL distro (SLES 16) and on latest Fedora
 - (against all supported PostgreSQL versions)



Second phase

- New: Run regression tests
 - via pgdg-srpm-macros
 - not enabled by default (intentionally)
 - Currently committing each spec file once testing is completed
 - Many packages have this infra now.



Third phase

- Check builds against supported PostgreSQL versions
- Commit to git repo
- Green light? Build packages on build instances and sign them
- Perform installation test
 - Test on Rocky Linux / Alma Linux as well
- Update <https://yum.postgresql.org/extensions/> if needed
- Update <https://yum.postgresql.org/software/> if needed.



Is that all?



Of course not!

- **Distros have different names for packages :(**

```
%if 0%{?suse_version} >= 1500
BuildRequires: libopenssl-3-devel
%endif
%if 0%{?fedora} >= 43 || 0%{?rhel} >= 9 || 0%{?amzn}
BuildRequires: openssl-devel
%endif
%if 0%{?fedora} && 0%{?fedora} <= 44 && !0%{?amzn}
BuildRequires: openssl-devel-engine
%endif
```

- **Same for zstd:**

```
%if 0%{?suse_version} >= 1500
    BuildRequires: libzstd-devel >= 1.4.0
    Requires: libzstd1 >= 1.4.0
%endif
%if 0%{?rhel} || 0%{?fedora}
    BuildRequires: libzstd-devel >= 1.4.0
    Requires: libzstd >= 1.4.0
%endif
```



Is that all?



Of course not! We love Python!

```
%if 0%{?fedora} && 0%{?fedora} == 45
%global __ospython %{_bindir}/python3.15
%global python3_pkgversion 3.15
%endif
%if 0%{?fedora} && 0%{?fedora} <= 44
%global __ospython %{_bindir}/python3.14
%global python3_pkgversion 3.14
%endif
%if 0%{?rhel} && 0%{?rhel} <= 10
%global __ospython %{_bindir}/python3.12
%global python3_pkgversion 3.12
%endif
```

```
%if 0%{?amzn} == 2023
%global __ospython %{_bindir}/python3.13
%global __python3 %{_bindir}/python3.13
%global python3_pkgversion 3.13
%endif
%if 0%{?suse_version} == 1500
%global __ospython %{_bindir}/python3.11
%global python3_pkgversion 3.11
%endif
%if 0%{?suse_version} == 1600
%global __ospython %{_bindir}/python3.13
%global python3_pkgversion 3.13
%endif
```



Is that all (another one)?



Of course not! Rust

- Avoided Rust for a long time
- Packaging Rust packages is harder than packaging C extensions:
 - <https://github.com/pgdg-packaging/pgdg-rpms/wiki/Packaging-Rust-vs-C-from-PGDG-RPMs-point-of-view>
- Bit the bullet recently for postgresql_anonymizer, pgvectorscale and pg_parquet
 - (and some of support packages)
- More may or may not come.



Is that all (another one)?



Of course not! Go!

- Extras repos for such packages
- consul, consul-template, etcd, pgwatch, WAL-G, Idap2pg
- No source compilation
- Use at your own risk!



Kick off!



Final phase

- Sync to staging server
- Sync to download mirrors
- Final installation test via yum/zypp repos



Lots of background work



Non-visible work

- Security
- Continuous update of automation and scripting
- Avoiding every single piece of failure
 - Package signatures, etc.
- Interaction with other packagers
- Following latest packaging features
 - Not just the RPM side, also the OS changes
- Building on Fedora first: uses latest GCC, CLANG, etc. I report any compilation issues to upstream.
 - This speeds up RHEL packaging, and also helps broader community



Let the breakage begin!



PostgreSQL RPM repository now supports multiple RHEL minor versions (Status update on the PGDG RPM repository and the November 2025 minor release breaking update)

Last updated: 29 Dec 2025 10:35 CST

Red Hat's decision to ship a major OpenSSL update (3.2 → 3.5) together with the RHEL 10.1 and 9.7 releases caused unexpected breakage—not only for users of Rocky Linux, AlmaLinux, and other derivatives, but also for RHEL 9.6 users (including EUS).



What does it mean?

- A full month of stress
 - (Found myself discussing with myself against a mirror)
 - (Did I mention that Magnus is a great person?)
- 6 new build instances, 8 new test instances
 - (Magnus)



List of shame



Beta/RC testing and “List of shame”

- We build daily snapshot RPMs of development (alpha) version (currently v20)
- Once a release reaches beta state, we start building extensions against them
 - Currently v19
- Create bug report for each extension that fails to build
- https://wiki.postgresql.org/wiki/PostgreSQL_19_Extension_Bugs
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs
- Follow up them regularly
- Pester them regularly



“Do you have some news for us?”



YES! No more EPEL!

- Getting rid of EPEL dependency *in style*!
 - Currently WIP-ish!
 - Committed spec file updates just today.
 - Ran all build and upgrade tests on all platforms!
 - Use our own directories to avoid clashes with the OS packages
 - PGDG GIS stack will use those new packages
 - Use the latest and greatest for CVE-free'ish world
 - Use patches from upstream when needed
 - Extreme flexibility



“Do you have some more news for us?”



YES! Amazon Linux 2023 meets GIS stack!

- AL 2023 does not have EPEL like repo!
- Difficulty level: Low to moderate
- Added 1 package to the repo: nlohmann_json
- Edited the rest to add Amazon Linux support!
- They are live!
- PostGIS, ogr_fdw, pgRouting, and support packages like SFCGAL, libgeotiff17, GDAL, GeOS, CGAL, etc.



“Do you have some more news for us?”



YES! PGDG RPM repo meets with AI!

- pg_statviz can now ask Claude, a local Ollama model, OpenAI or Gemini to explain your PostgreSQL statistics on every distro we ship it for.
- 45+ new packages added to the repo
- https://people.planetpostgresql.org/devrim/index.php?/archives/132-PostgreSQL-RPM-repository-meets-AI-with-pg_statviz.html
- Enjoy!



EDB's contribution to the packaging ecosystem



EDB is leading the packaging work

- Community RPMs
 - Human power, x86_64 build instances, etc.
- macOS
 - Human power, hosting, build instances
- Windows
 - Human power, hosting, build instances



EDB is leading the packaging work

- Q/A testing of all packages (including community packages)
- Helps promote of the packaging work
- All about giving back to the community



and the end!



THANKS FOR LISTENING!

ANY QUESTIONS?



Help me and the conference
get better!

Give feedback for this
session and the conference.



PostgreSQL Conference
Europe 2026

Valencia

Oct 20th - 23rd



EDB

Postgres® for the AI Generation

The process of RPM packaging in the PostgreSQL ecosystem and EDB's contributions

Devrim Gündüz

Principal Engineer @ EDB

RPM packager @ PGDG

Part time DJ, serial traveller

<https://www.linkedin.com/in/devrimgunduz/>