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

Devrim Gündüz
Principal Engineer @ EDB
RPM packager @ PGDG
Part time DJ, serial traveller
https://www.linkedin.com/in/devrimgunduz/

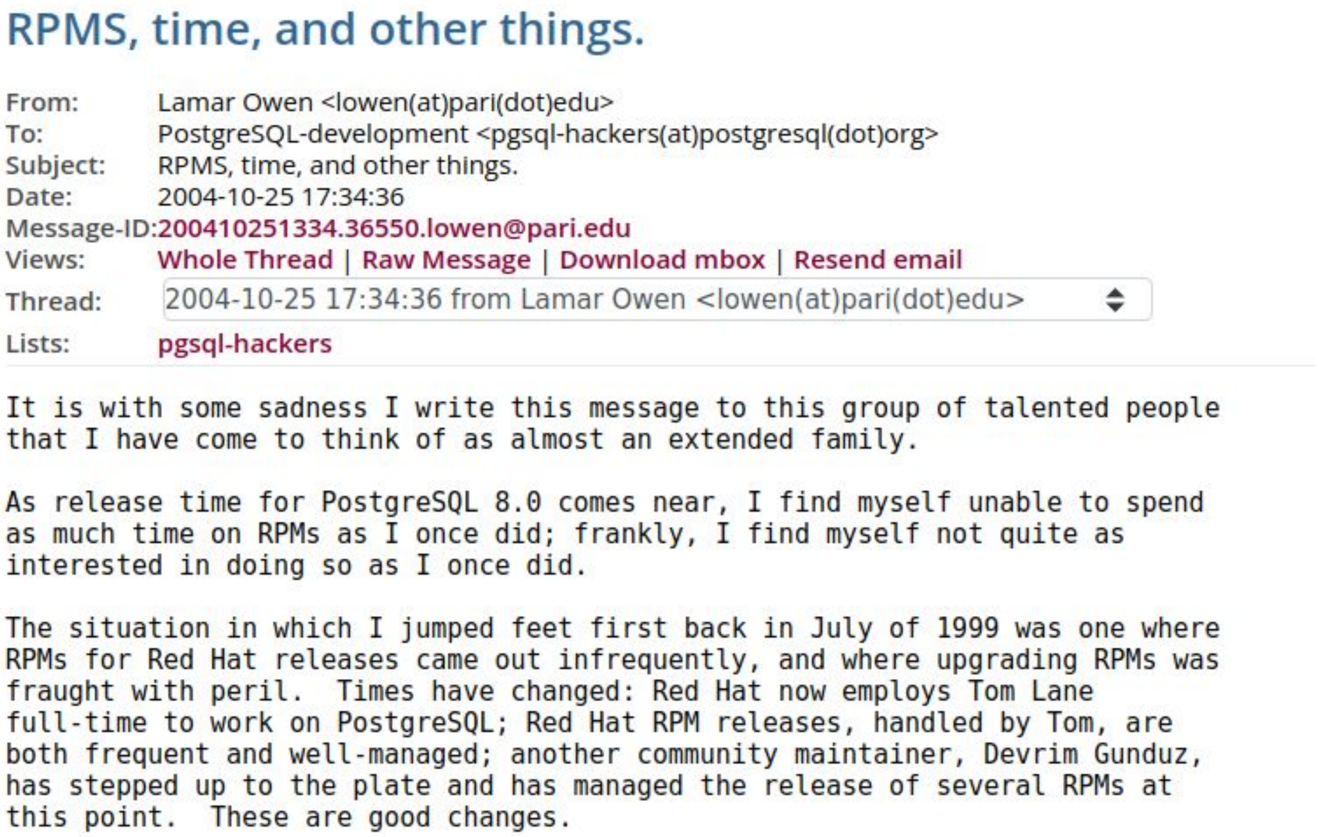
How it started?

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/
- 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 Helsinki 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!

Packaging process

Creating spec files (and patches)

- I create initial spec file on my local box (Fedora latest stable)
- Do initial build and installation tests
- Find potentially missing build dependencies
- Find potentially missing runtime dependencies

Second phase

- Check builds against supported PostgreSQL versions
- Commit to git repo
- Do mock builds
- Fix any potential issues
- Green light? Build packages on build instances and sign them
- Perform installation test
- Test on Rocky Linux / Alma Linux
- Update https://yum.postgresql.org/extensions/ if needed

Is that all?

Of course not!

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

```
%if 0%{?suse_version} >= 1500
  Requires:       libopenssl3
  BuildRequires:  libopenssl-3-devel
%endif
%if 0%{?fedora} >= 42 || 0%{?rhel} >= 8
  Requires:       openssl-libs >= 1.1.1k
  BuildRequires:  openssl-devel
%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:

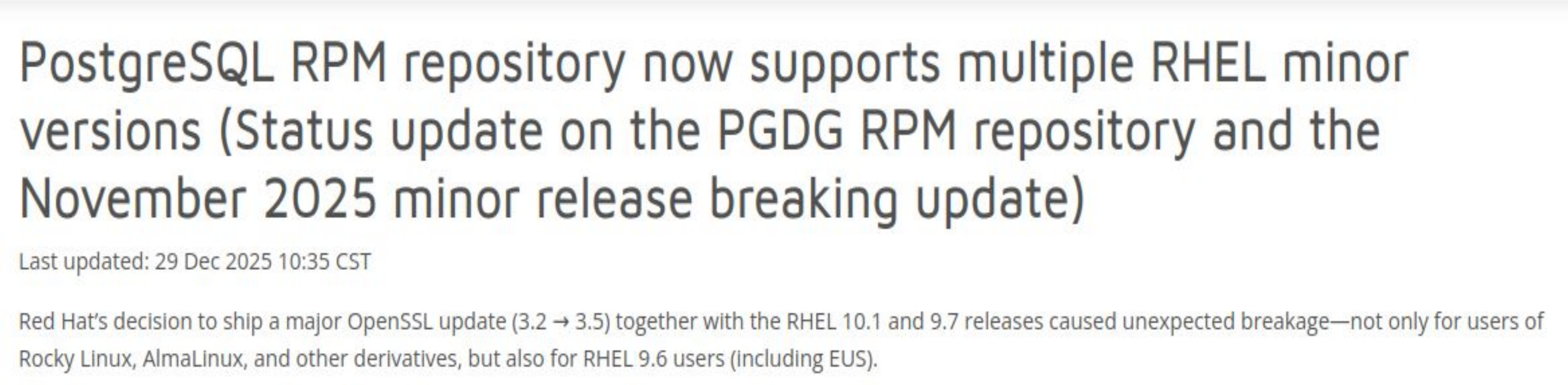
```
%if 0%{?fedora} && 0%{?fedora} == 43
%global __ospython %{_bindir}/python3.14
%global python3_pkgversion 3.14
%endif
%if 0%{?fedora} && 0%{?fedora} <= 42
%global __ospython %{_bindir}/python3.13
%global python3_pkgversion 3.13
%endif
%if 0%{?rhel} && 0%{?rhel} <= 10
%global __ospython %{_bindir}/python3.12
%global python3_pkgversion 3.12
%endif
%if 0%{?suse_version} == 1500
%global __ospython %{_bindir}/python3.11
%global python3_pkgversion 311
%endif
%if 0%{?suse_version} == 1600
%global __ospython %{_bindir}/python3.13
%global python3_pkgversion 313
%endif
```

Kick off!

Final phase

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

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

and the end!

aaaand!

EDB
Postgres® for the AI Generation

The process of RPM packaging in the PostgreSQL ecosystem

Devrim Gündüz
Principal Engineer @ EDB
RPM packager @ PGDG
Part time DJ, serial traveller
https://www.linkedin.com/in/devrimgunduz/