EDB
Postgres® for the AI Generation

Extend your PostgreSQL: The world of PostgreSQL extensions

Devrim Gündüz
Postgres Expert

Chicago PostgreSQL User Group Meetup
November 2024

Don’t expect a happy story!

Self Introduction

- PostgreSQL Major Contributor
- Responsible for PostgreSQL RPM repos (Red Hat, Rocky, AlmaLinux Fedora and SLES)
- Fedora and Rocky Linux contributor
- PostgreSQL community member
- Postgres expert @ EDB
- “The guy with the PostgreSQL tattoo”
- Chicago enthusiast
- London, UK.

…and sometimes:

DJing!

DJing!

Agenda

- History
- Introduction to extensions
- Extension issues
- Favourite extensions
- RPMified extensions

Are you using PostgreSQL extensions?

Extensions

- Of course!

Extensions

- Of course!
- *“Everyone is a pg_stat_statements fan, even if they are not. Even if you ask the Oracle guys, they are also a fan of pg_stat_statements”* © 2024 Devrim Gunduz

Extensions

- Of course!
- *"Everyone is a pg_stat_statements fan, even if they are not. Even if you ask the Oracle guys, they are also a fan of pg_stat_statements"* © 2024 Devrim Gunduz
    - Replace pg_stat_statement with your most favourite extension!

Pre-extension era: Contrib modules

Pre-extension era: “In-core-ish” contrib modules

- Included in the tarball
- Could include:
  - new data types
  - new operator(s)
  - new function(s)
  - new index AM
  - …
- “not in core”
- “but in core-ish”

Pre-extension era: “In-core-ish” contrib modules

- Invoke individual install/uninstall SQL scripts
    - extension_name.sql
    - uninstall_extension_name.sql
- Versioning issues
- Dependency issues
- Hard to add a new one into the core
- Historically was very hard to merge into the core
    - tsearch2

External “contrib modules”

- Not included in the tarball
- External development
- Same installation/uninstallation procedures
- Same issues

PostgreSQL meets “Extensions”

PostgreSQL meets extensions

- 9.1 (13 years!)
- Title: “Packages of external objects”
- “...make it easier to add and remove plugins, data types, functions etc with minimum fuss and without needing to compile things yourself.” (Simon Riggs)
- Contrib modules became extensions
  - still in contrib/ directory
- Easier to maintain from user point of view

PostgreSQL meets extensions

- Dependency checking (via control file)
- Contents:
  - “C” code
    - Could be skipped for extensions that does not require a .so file
  - SQL commands
  - .control file
- In-core PostgreSQL extensions:
  - https://www.postgresql.org/docs/current/contrib.html

control file

- A few basic properties of the extension.
- extension_name.control
- Must be available separately for each major PostgreSQL version
- Dependency tracking
  - Example: *h3_postgis.control:requires = 'h3, postgis, postgis_raster'*
- ..

.sql scripts

- extension_name--version.sql
- Installation scripts
- Upgrade scripts
- Located under $installation_dir/share/extension (example: /usr/pgsql-17/share/extension/

List of available extensions

- pg_available_extensions
- pg_available_extension_versions

State of the PostgreSQL Extensions

They are everywhere

- Github

They are everywhere

- Github
- Gitlab

They are everywhere

- Github
- Gitlab
- OSGeo Gitea

They are everywhere

- Github
- Gitlab
- OSGeo Gitea
- PGXN

They are everywhere

- Github
- Gitlab
- OSGeo Gitea
- PGXN
- Sourceforge

They are everywhere

- Github
- Gitlab
- OSGeo Gitea
- PGXN
- Sourceforge
- PostgreSQL git repo (...)

They are everywhere

- Github
- Gitlab
- OSGeo Gitea
- PGXN
- Sourceforge
- PostgreSQL git repo (...)
- PyPI

They are everywhere

- Github
- Gitlab
- OSGeo Gitea
- PGXN
- Sourceforge
- PostgreSQL git repo (...)
- PyPI
- Their own websites

They are everywhere

- Github
- Gitlab
- OSGeo Gitea
- PGXN
- Sourceforge
- PostgreSQL git repo (...)
- PyPI
- Their own websites
- Bitbucket

They are everywhere

- Github
- Gitlab
- OSGeo Gitea
- PGXN
- Sourceforge
- PostgreSQL git repo (...)
- PyPI
- Their own websites
- Bitbucket
- Laptops (yeah)

Laptops?

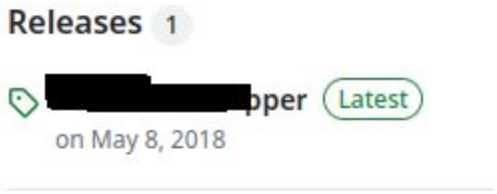
- Sometimes all I find is a tarball somewhere

Laptops?

- Sometimes all I find is a tarball somewhere
- There is a git repo, but:

Laptops?

- Sometimes all I find is a tarball somewhere
- There is a git repo, but:



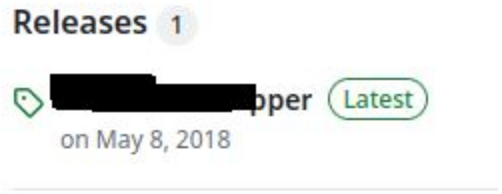
Releases 1

[redacted] Latest

on May 8, 2018

Laptops?

- Sometimes all I find is a tarball somewhere
- There is a git repo, but:



Releases 1

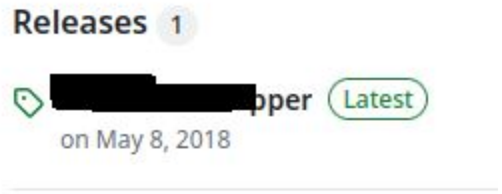
[redacted]pper Latest

on May 8, 2018

…and there is no recent commit!

Laptops?

- Sometimes all I find is a tarball somewhere
- There is a git repo, but:



Releases 1

[redacted] Latest

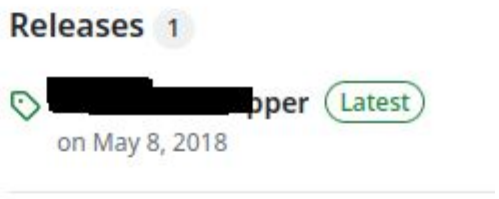
on May 8, 2018

…and there is no recent commit!

- There is a very recent release on PGXN, so I asked the author:

Laptops?

- Sometimes all I find is a tarball somewhere
- There is a git repo, but:



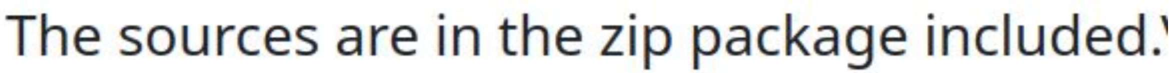
Releases 1

[redacted]pper Latest

on May 8, 2018

…and there is no recent commit!

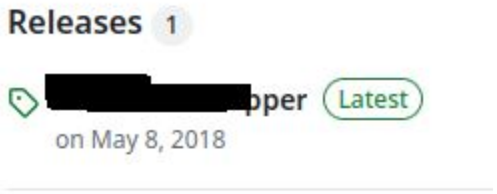
- There is a very recent release on PGXN, so I asked the author (*facepalm*)



The sources are in the zip package included.'

Laptops?

- Sometimes all I find is a tarball somewhere
- There is a git repo, but:



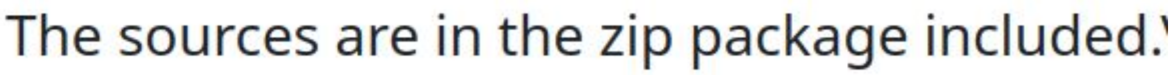
Releases 1

[redacted]pper Latest

on May 8, 2018

…and there is no recent commit!

- There is a very recent release on PGXN, so I asked the author (*facepalm*)

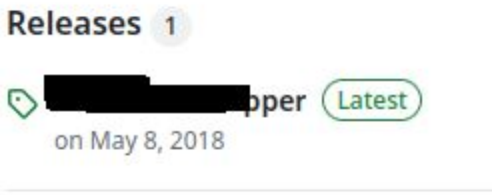


The sources are in the zip package included.

- Keep pestering, but:

Laptops?

- Sometimes all I find is a tarball somewhere
- There is a git repo, but:



Releases 1

[redacted] Latest

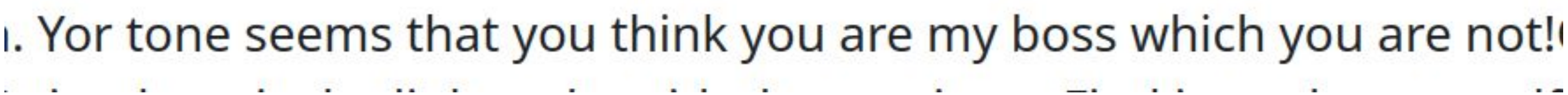
on May 8, 2018

…and there is no recent commit!

- There is a very recent release on PGXN, so I asked the author (*facepalm*):

The sources are in the zip package included.'

- Keep pestering, but:

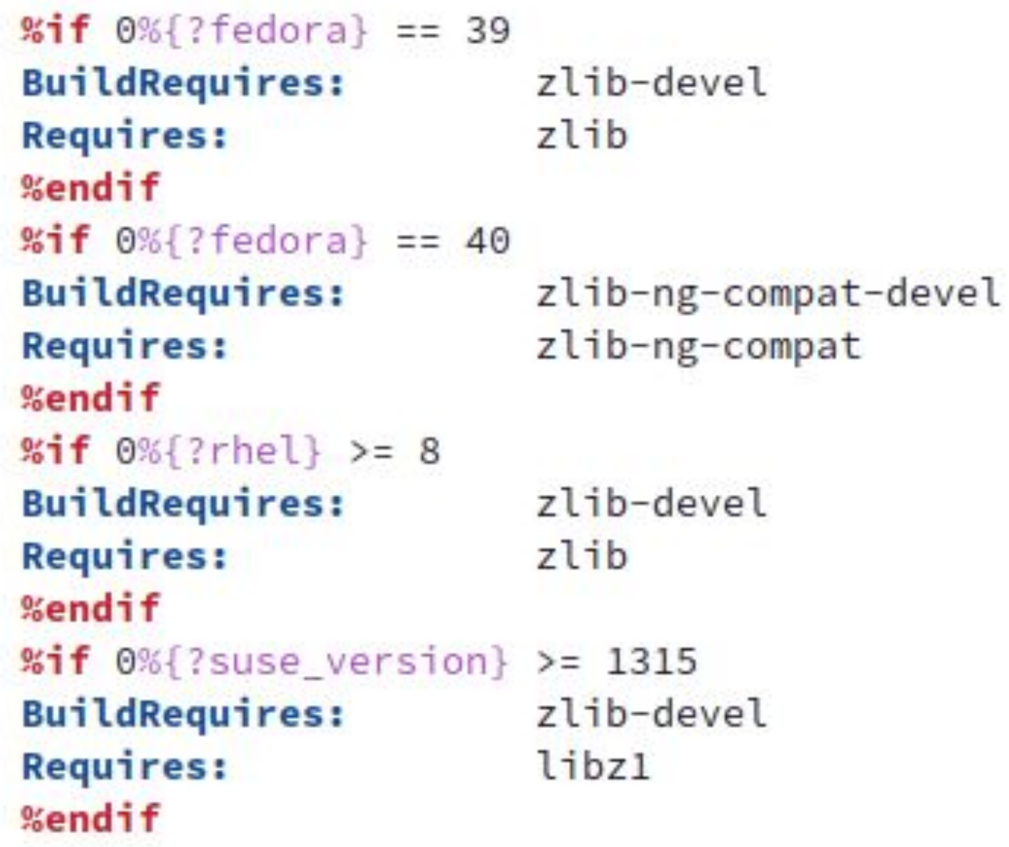


ı. Yor tone seems that you think you are my boss which you are not!

Is this all?

Platform support

“It must not be hard”

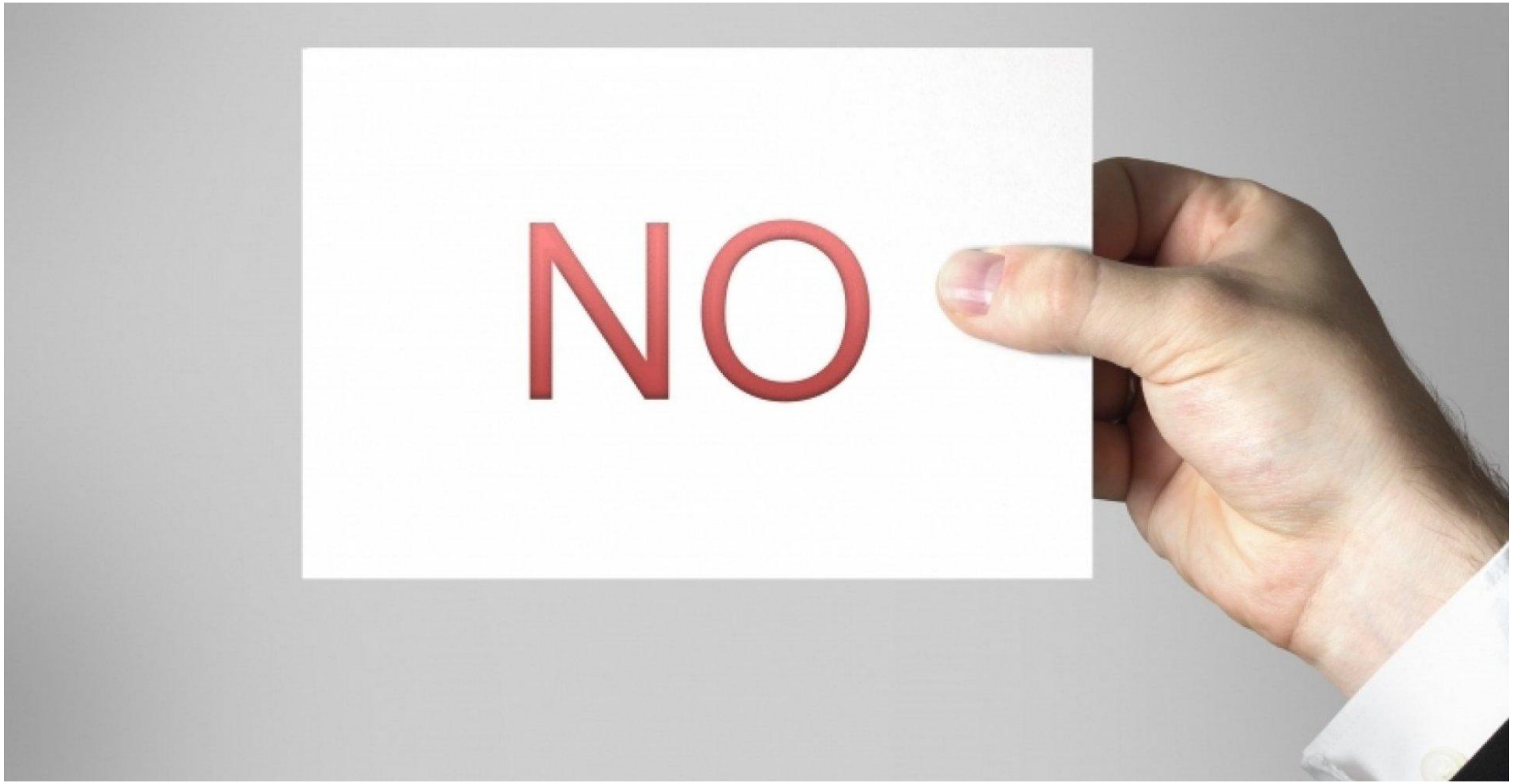


```
%if 0%{?fedora} == 39
BuildRequires:          zlib-devel
Requires:               zlib
%endif
%if 0%{?fedora} == 40
BuildRequires:          zlib-ng-compat-devel
Requires:               zlib-ng-compat
%endif
%if 0%{?rhel} >= 8
BuildRequires:          zlib-devel
Requires:               zlib
%endif
%if 0%{?suse_version} >= 1315
BuildRequires:          zlib-devel
Requires:               libz1
%endif
```

Let’s move on

Is this all?

No!



No!

- Developer: “I’ve got a great idea! Let me write an extension for that”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”
- Users: “It works like a charm! Let’s deploy it to the production!”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”
- Users: “It works like a charm! Let’s deploy it to the production!”
- Users: “There is a new PostgreSQL release and it does not compile”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”
- Users: “It works like a charm! Let’s deploy it to the production!”
- Users: “There is a new PostgreSQL release and it does not compile”
- Packagers: “There is a new PostgreSQL release and it does not compile”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”
- Users: “It works like a charm! Let’s deploy it to the production!”
- Users: “There is a new PostgreSQL release and it does not compile”
- Packagers: “There is a new PostgreSQL release and it does not compile”
- Developer: “ZzzzzzZZzzzzz”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”
- Users: “It works like a charm! Let’s deploy it to the production!”
- Users: “There is a new PostgreSQL release and it does not compile”
- Packagers: “There is a new PostgreSQL release and it does not compile”
- Developer: “ZzzzzzZZzzzzz”
- Packagers: “PostgreSQL $NewMajorVersion.0 is due soon”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”
- Users: “It works like a charm! Let’s deploy it to the production!”
- Users: “There is a new PostgreSQL release and it does not compile”
- Packagers: “There is a new PostgreSQL release and it does not compile”
- Developer: “ZzzzzzZZzzzzz”
- Packagers: “PostgreSQL $NewMajorVersion.0 is due soon”
- Users: “GA is out! We want to upgrade!”

No!

- Developer: “I’ve got a great idea! Let me write an extension for that”
- Users: “Yay!”
- Users: “It works like a charm! Let’s deploy it to the production!”
- Users: “There is a new PostgreSQL release and it does not compile”
- Packagers: “There is a new PostgreSQL release and it does not compile”
- Developer: “ZzzzzzZZzzzzz”
- Packagers: “PostgreSQL $NewMajorVersion.0 is due soon”
- Users: “GA is out! We want to upgrade!”
- Developer: “...”

“Non-responsive maintainers”

“Non-responsive maintainers”

- A major problem

“Non-responsive maintainers”

- A major problem
- Extensions get stalled

“Non-responsive maintainers”

- A major problem
- Extensions get stalled
- Some good people send a PR, but doors are closed

“Non-responsive maintainers”

- A major problem
- Extensions get stalled
- Some good people send a PR, but doors are closed
- Fork?

“Non-responsive maintainers”

- Not good for the users

“Non-responsive maintainers”

- Not good for the users
- Not good for the community

“Non-responsive maintainers”

- Not good for the users
- Not good for the community
- Not good for the packagers

Life of a packager

Life of a packager

Life of a packager

- Wrong slide, sorry!

Life of a packager

- Wrong slide, sorry!
- We do some actual work

Life of a packager

- Wrong slide, sorry!
- We do some actual work
- Actually it is a lot of work
    - APT based distros
    - RPM based distros
    - Windows
    - MacOS

Let the breakage begin

Breakage?

- New PostgreSQL release

Breakage?

- New PostgreSQL release
- New GCC version

Breakage?

- New PostgreSQL release
- New GCC version
- New LLVM version

Breakage?

- New PostgreSQL release
- New GCC version
- New LLVM version
- New “dependency” version

Wish it was that easy…

Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs

Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs
- We start annoying the extension authors along with the beta cycle

Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs
- We start annoying the extension authors along with the beta cycle
- Many extensions just work
  - Work: builds fine and passes tests

Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs
- We start annoying the extension authors along with the beta cycle
- Many extensions just work
  - Work: builds fine and passes tests
- We start annoying the extension authors

Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs
- We start annoying the extension authors along with the beta cycle
- Many extensions just work
  - Work: builds fine and passes tests
- We start annoying the extension authors
- Good people already follow the development of PostgreSQL, so they apply the fixes immediately

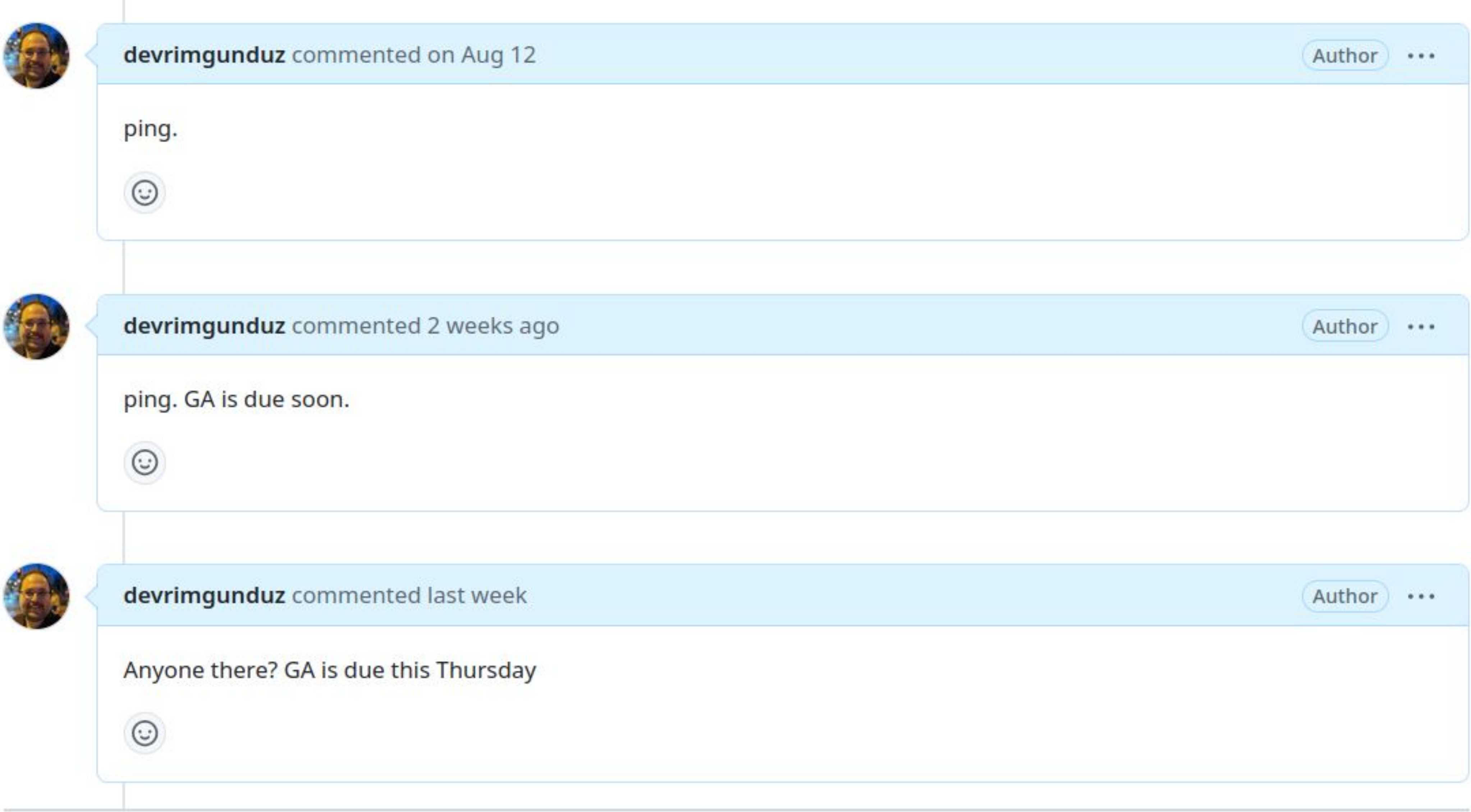
Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs
- We start annoying the extension authors along with the beta cycle
- Many extensions just work
  - Work: builds fine and passes tests
- We start annoying the extension authors
- Good people already follow the development of PostgreSQL, so they apply the fixes immediately
- But then:

Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs
- We start annoying the extension authors along with the beta cycle
- Many extensions just work
  - Work: builds fine and passes tests
- We start annoying the extension authors
- Good people already follow the development of PostgreSQL, so they apply the fixes immediately
- But then:
- GOTO -1

Slackers!



devrimgunduz commented on Aug 12

ping.

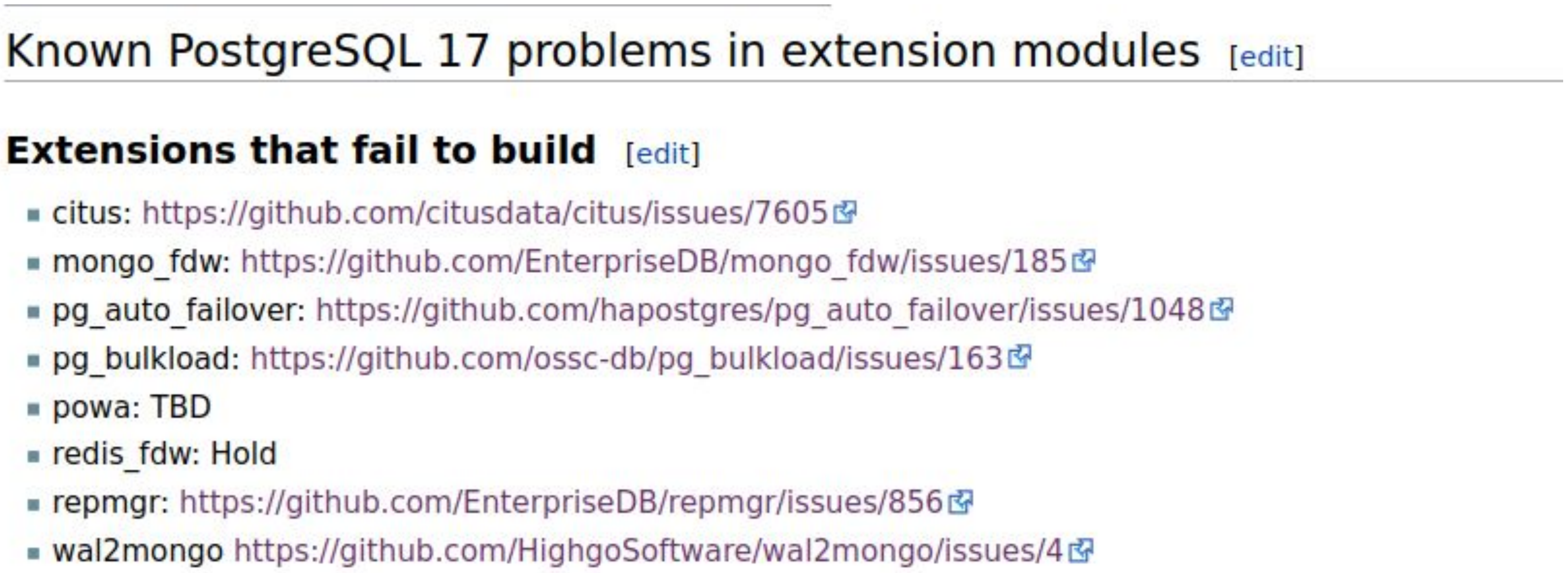
devrimgunduz commented 2 weeks ago

ping. GA is due soon.

devrimgunduz commented last week

Anyone there? GA is due this Thursday

Slackers!

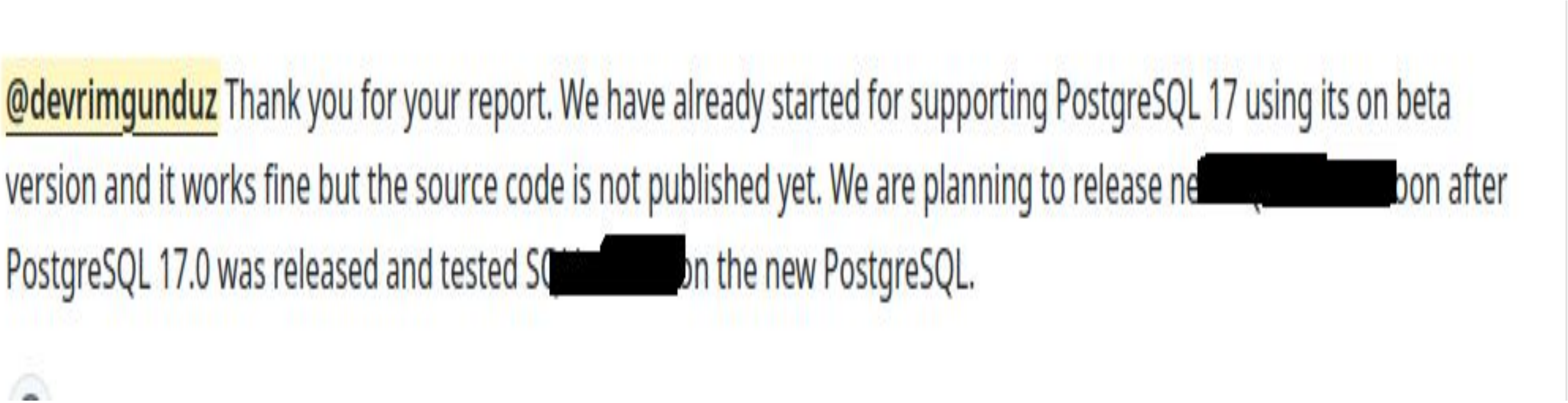


Known PostgreSQL 17 problems in extension modules [edit]

Extensions that fail to build [edit]

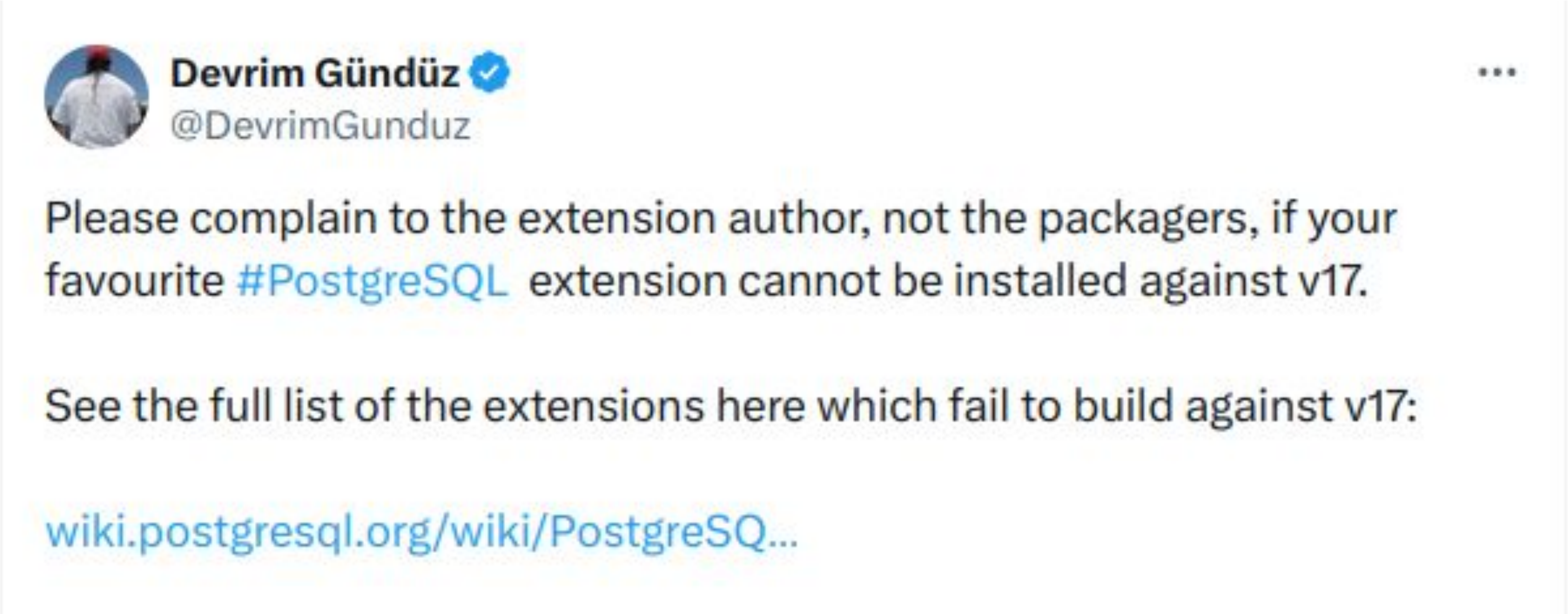
- citus: https://github.com/citusdata/citus/issues/7605
- mongo_fdw: https://github.com/EnterpriseDB/mongo_fdw/issues/185
- pg_auto_failover: https://github.com/hapostgres/pg_auto_failover/issues/1048
- pg_bulkload: https://github.com/ossc-db/pg_bulkload/issues/163
- powa: TBD
- redis_fdw: Hold
- repmgr: https://github.com/EnterpriseDB/repmgr/issues/856
- wal2mongo https://github.com/HighgoSoftware/wal2mongo/issues/4

Sure. Take your time. No rush. We can wait. Users can wait. Everyone can wait.



@devrimgunduz Thank you for your report. We have already started for supporting PostgreSQL 17 using its on beta version and it works fine but the source code is not published yet. We are planning to release ne[redacted]oon after PostgreSQL 17.0 was released and tested S[redacted]on the new PostgreSQL.

So, what did I do?



Devrim Gündüz
@DevrimGunduz

Please complain to the extension author, not the packagers, if your favourite #PostgreSQL extension cannot be installed against v17.

See the full list of the extensions here which fail to build against v17:

wiki.postgresql.org/wiki/PostgreSQ...

Installing an extension

Installation

- First of all: Install the extension package for the specific PostgreSQL version
  - Preferred: RPM, DEB, etc.
  - Compilation from source
  - Other installation options
- **May require adding extension name to shared_preload_libraries and restarting the server**

  ```
  [postgres] # CREATE EXTENSION citus;
  ERROR:  Citus can only be loaded via shared_preload_libraries
  HINT:  Add citus to shared_preload_libraries configuration variable in postgresql.conf in master and workers. Note that citus should be at the beginning of shared_preload_libraries.
  ```

- May require dependencies to be installed
  - Package managers often solve this problem

Installation

- CREATE EXTENSION
  - VERSION
  - CASCADE
  - SCHEMA
- Dependencies:

```
[postgres] # CREATE EXTENSION h3_postgis ;
ERROR:  required extension "postgis_raster" is not installed
HINT:  Use CREATE EXTENSION ... CASCADE to install required extensions too.
```

- CASCADE:

```
[postgres] # CREATE EXTENSION h3_postgis CASCADE;
NOTICE:  installing required extension "postgis_raster"
```

Updating an extension

Updating an extension

- Install the updated extension
- ALTER EXTENSION … UPDATE;
- Some extension versions may not need this
  - “Upgrade path”
  - “NOTICE: version "12.1-1" of extension "foobar" is already installed”
  - Problem?
    - Restart?
- Duplicate .so files?

Uninstalling an extension

Updating an extension

- DROP EXTENSION … ;
- Dependencies:

  ```
  [postgres] # DROP EXTENSION postgis_raster ;
  ERROR:  cannot drop extension postgis_raster because other objects depend on it
  DETAIL:  extension h3_postgis depends on extension postgis_raster
  HINT:  Use DROP ... CASCADE to drop the dependent objects too.
  ```

- CASCADE:

  ```
  [postgres] # DROP EXTENSION postgis_raster CASCADE;
  NOTICE:  drop cascades to extension h3_postgis
  DROP EXTENSION
  ```

- CASCADE: Also drops objects that depends on the extension!

So, favourite extensions

PLs

- PL/pgSQL (only default “extension”)
- PL/Python
- PL/Perl
- PL/TCL
- PL/Java
- PL/Lua
- PL/Bash (sh(
- PL/R

Also:

- plpgsql_check
- plprofiler

PostGIS

- Really an extension? ;)
- Spatial database “extender” for PostgreSQL
- https://postgis.net

pgRouting

- Extends PostGIS, with routing and network analysis functionality.
- Comes with great number of routing algorithms.
- Available in many packaging formats, including RPMs.
- Depends on CGAL and Boost (...)
- https://www.pgrouting.org

pg_stat_statements

- A must-install for everyone
- If you disagree with it, just do agree :)

pgvector

- Vector similarity search for Postgres
- Stores your vectors with the rest of your data

pg_partman

- Makes PostgreSQL partitioning manageable and usable
- Actively developed

Citus

- Horizontal scaling of PostgreSQL
- sharding & replication
- 100% open source

TimescaleDB

- PostgreSQL based time-series database
- Partial open source
- automatic partitioning across time and space (partitioning key)

pgAudit

- Auditing extension for PostgreSQL
  - Very extensive!
- Recently changed versioning numbers
  - pgaudit 17.x :PostgreSQL 17
  - pgAudit 16.x : PostgreSQL 16
  - pgAudit 1.7.x : PostgreSQL 15
  - pgAudit 1.6.x: PostgreSQL 14
  - …

FDWs

- postgres_fdw
- oracle_fdw
- tds_fdw
- mysql_fdw
- hdfs_fdw
- mongo_fdw
- sqlite_fdw
- firebird_fdw
- jdbc_fdw
- odbc_fdw
- ogr_fdw
- pgbouncer_fdw
- file_fdw

Some other extensions

- count_distinct
- credcheck
- ddlx
- emaj
- h3-pg
- hll
- hypopg
- ip4r
- login_hook
- multicorn2
- pg_cron
- pg_dbms_job
- pg_dbms_lock
- pg_catcheck
- pg_background
- postgresql-anonymizer
- pg_auth_mon
- pg_extra_time
- pg_tle
- pg_repack
- pg_ivm
- pg_net
- pg_squeeze
- pg_uri
- pgtt
- rum
- set_user
- topn
- wal2json
- wal2mongo
- postgresql-unit

Some other extensions

- plproxy
- pg_readonly
- pg_qualstats
- pg_prioritize
- pg_hint_plan
- pg_store_plans
- pg_stat_kcache
- pgplsh
- pg_statement_rollback
- pgmemcache
- pgl_ddl_deploy
- pgauditlogtofile
- pgsphere
- pgsodium
- pldebugger
- plpgsql_check
- plprofiler
- pgsql_http
- pgsql_gzip
- pgsql_tweaks
- pointcloud
- semver
- …

Thank **you**.

EDB
Postgres® for the AI Generation

Extend your PostgreSQL: The world of PostgreSQL extensions

Devrim Gündüz
Postgres Expert

Chicago PostgreSQL User Group Meetup
November 2024