EDB
Postgres® for the AI Generation

Extend your PostgreSQL: The world of PostgreSQL extensions

Devrim Gündüz
Postgres Expert @ EDB
PGConf.DE
May 2025

Linkedin:https://www.linkedin.com/in/devrimgunduz/
Instagram: devrimgunduztr
Bluesky: https://bsky.app/profile/devrim.gunduz.org

Introducing WarehousePG

https://github.com/warehouse-pg/warehouse-pg

April
2025

WarehousePG
New Open Source Project

A new Apache 2-licensed fork of the formerly **open source Greenplum Database.**
Binary compatibility with the legacy Greenplum versions most widely deployed by customers (6.x and 7.x) — plus net-new functionality and integrations.

EDB POSTGRES AI: SUPPORT FOR GREENPLUM WORKLOADS

TRANSITION TO COMMERCIALLY SUPPORTED OPEN-SOURCE WAREHOUSEPG

WarehousePG Highlights

Petabytes scale with high-speed /-concurrency performance, via MPP Cluster topology

Postgres Fork based on PG 12.12

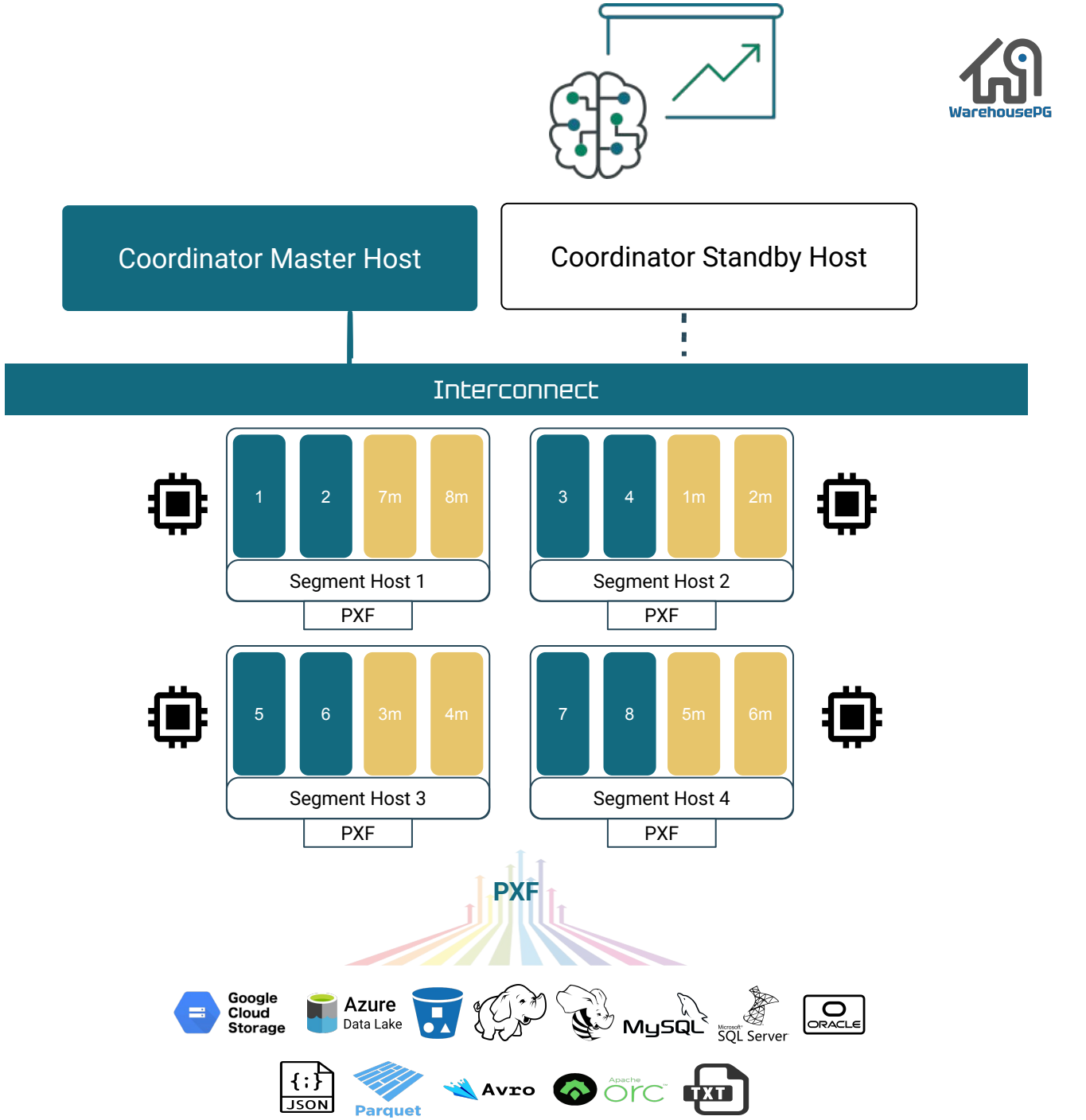
Master (coordinator) for query optimization and parallel plan and distribution

Multiple Segments

- Segment is a Postgres Process which holds data
- Own CPUs, Hard disk and RAM
- Scalable

HA achieved through Standby for Master and Mirror for Segments

High-speed Interconnection for continuous data processing pipelining



Don’t expect a happy story!

EDB
Postgres® for the AI Generation

Self Introduction

- PostgreSQL Major Contributor
- Responsible for PostgreSQL RPM repos (Red Hat, Rocky, AlmaLinux Fedora and SLES)
- Fedora and Rocky Linux contributor
- PostgreSQL user since 1998
- PostgreSQL community member
- Postgres expert @ EDB
- “The guy with the PostgreSQL tattoo”
- Serial traveller
- London, UK / Istanbul, TR / Chicago, US / Airport lounges, around the world.

…and sometimes:

DJing!

DJing!

EDB
Postgres® for the AI Generation

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”* © 2025 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”* © 2025 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

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
  - …

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”

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

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

- Invoke individual install/uninstall SQL scripts
  - extension_name.sql
  - uninstall_extension_name.sql
- Versioning issues

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

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

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 has been very hard to merge into the core
  - tsearch2

External “contrib modules”

- Not included in the tarball

External “contrib modules”

- Not included in the tarball
- External development

External “contrib modules”

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

External “contrib modules”

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

External “contrib modules”

- Not included in the tarball
- External development
- Same installation/uninstallation procedures
- Same issues
- PGXS: PostgreSQL Extension Build System

PostgreSQL meets “Extensions”

PostgreSQL meets extensions

- 9.1 (13.5 years!)

PostgreSQL meets extensions

- 9.1 (13.5 years!)
- Title: “Packages of external objects”

PostgreSQL meets extensions

- 9.1 (13.5 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)

PostgreSQL meets extensions

- 9.1 (13.5 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

PostgreSQL meets extensions

- 9.1 (13.5 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)

PostgreSQL meets extensions

- Dependency checking (via control file)
- Contents:
  - “C” code
    - Could be skipped for extensions that does not require a .so file

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

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

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.

control file

- A few basic properties of the extension.
- extension_name.control

control file

- A few basic properties of the extension.
- extension_name.control
- Must be available separately for each major PostgreSQL version

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

.sql scripts

- extension_name--version.sql
- Installation scripts

.sql scripts

- extension_name--version.sql
- Installation scripts
- Upgrade scripts

.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

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:

Laptops?

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

…and there is no recent commit!

Laptops?

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

  …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:

…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:

…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:

…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

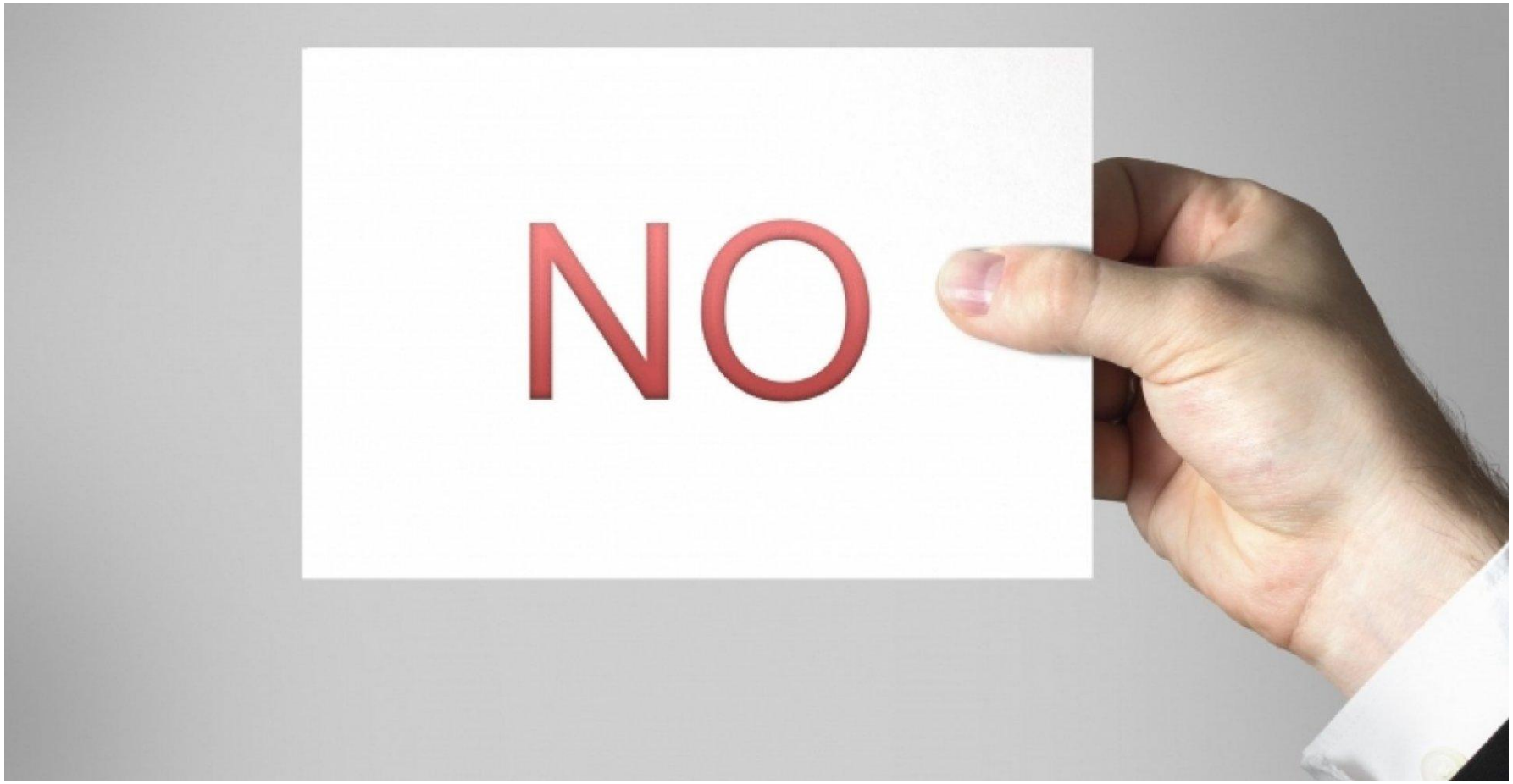
“It must not be hard”

```
%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
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs (WIP)

Road to the major release

- https://wiki.postgresql.org/wiki/PostgreSQL_17_Extension_Bugs
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs (WIP)
- 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
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs (WIP)
- 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
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs (WIP)
- 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
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs (WIP)
- 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
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs (WIP)
- 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
- https://wiki.postgresql.org/wiki/PostgreSQL_18_Extension_Bugs (WIP)
- 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 — Author

ping.

devrimgunduz commented 2 weeks ago — Author

ping. GA is due soon.

devrimgunduz commented last week — Author

Anyone there? GA is due this Thursday

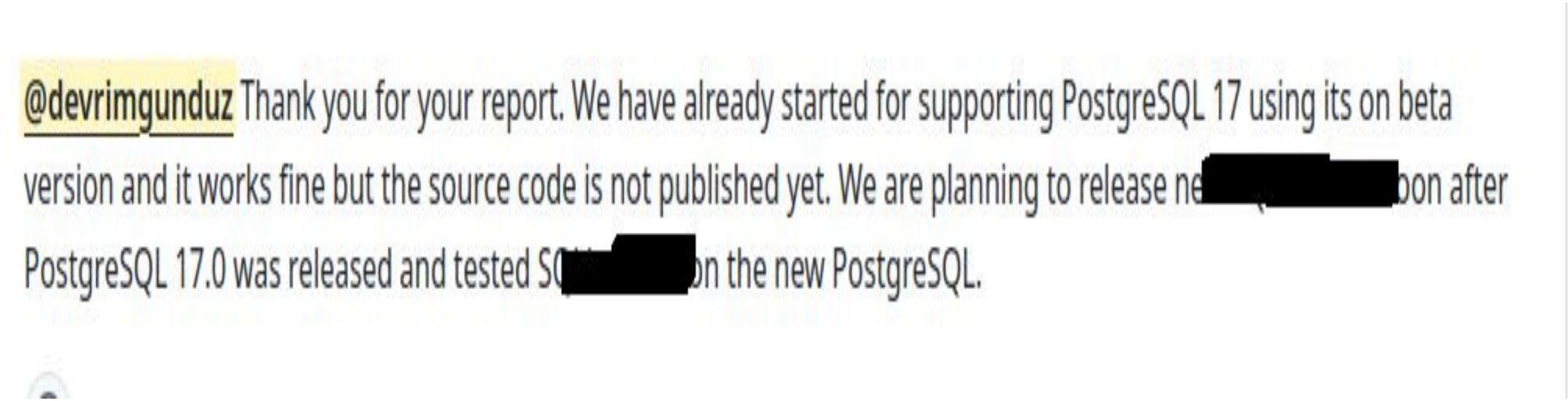
Slackers!

Known PostgreSQL 17 problems in extension modules [edit]

Extensions that fail to build [edit]

- wal2mongo https://github.com/HighgoSoftware/wal2mongo/issues/4

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



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

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.

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

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.

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"

or, read!

Find many information in one website

Installing PostgreSQL Extensions via RPMs

This document includes detailed information about installing and configuring PostgreSQL extensions on RPM based systems.

Extension how-to docs

- Foreign Data Wrappers
- DBA Tools
- Application Developers
- Data types and extra functions
- Replication
- Procedural Languages
- Monitoring
- Analytics
- Oracle compatibility
- Indexes
- Security
- Geospatial
- Major Features
- Other features

https://yum.postgresql.org/extensions

Updating an extension

Updating an extension

- Install the updated extension

Updating an extension

- Install the updated extension
- ALTER EXTENSION … UPDATE;

Updating an extension

- Install the updated extension
- ALTER EXTENSION … UPDATE;
- Some extension versions may not need this

Updating an extension

- Install the updated extension
- ALTER EXTENSION … UPDATE;
- Some extension versions may not need this
  - “Upgrade path”

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”

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?

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 … ;

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.
```

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

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
- PL/xslt
- pgpcre
- q3c
- pg_roaringbitmap
- icu_ext
- …

Thank you.

EDB
Postgres® for the AI Generation

Extend your PostgreSQL: The world of PostgreSQL extensions

Devrim Gündüz
Postgres Expert @ EDB
SCaLE 22x
March 2025

Linkedin:https://www.linkedin.com/in/devrimgunduz/
Instagram: devrimgunduztr
Bluesky: https://bsky.app/profile/devrimgunduz.bsky.social