

Postgres Upgrade Best Practices

Sunil Narain, Technical Director
Customer Success

18 November, 2021



Agenda

- Postgres version numbering
- Reasons for upgrade
- Upgrade methods
- Common upgrade issues
- Best practices
- Demo
- Q&A



Postgres version numbering

- PostgreSQL versions:
 - Major/minor versions: 9.6.x, 10.x, 11.x, 12.x, 13.x, 14.x...
 - For example, in PostgreSQL 13.4: major version number is 13 and minor patch version is 4
- EPAS versions:
 - Major/minor versions: 9.6.x.y, 10.x.y, 11.x.y, 12.x.y, 13.x.y, 14.x.y..
 - For example, in EPAS 13.4.8, major version is 13, community minor patch version is 4, EDB minor patch version is 8
- Major versions include new enhancements, bug fixes, and functionality changes. One major version is released every year.
- Minor versions include security and bug fixes. Minor patches are released quarterly.

Reasons for upgrading

- New features
- Bug fixes
- End of support for the release
- Compliance reasons

New Features introduced in each release

- PostgreSQL v10 – 11
- PostgreSQL v11 - 12
- PostgreSQL v12 - 17
- PostgreSQL v13 - 15
- PostgreSQL v14 – 12

Source : <https://www.postgresql.org/about/featurematrix/>

Bug/Security vulnerability fixes

- Every new version of Postgres/EPAS contains fixes for bugs and any security vulnerabilities
- Most recently, following two critical vulnerabilities were identified and fixed in latest minor releases:
 - **[CVE-2021-23214](#): Server processes unencrypted bytes from man-in-the-middle**
 - **[CVE-2021-23222](#): libpq processes unencrypted bytes from man-in-the-middle**
 - Read this blog for details : <https://www.enterprisedb.com/blog/postgres-mitm21-vulnerabilities>
 - FAQ : <http://www.enterprisedb.com/faq-mitm21>
 - The MITM21 webinar link/recording can be found here:
 - <https://www.enterprisedb.com/promote/webinar-important-updates-about-postgres-man-middle-attack-vulnerabilities>
 - Send email to MITM21@enterprisedb.com if you have any question about this vulnerability.

End of Support for different releases

- | | | | |
|--------------------------|-----------------------|----------------------------|----------------|
| • PostgreSQL v9.6 | - Nov 11, 2021 | EDB Postgres Advanced 9.6 | - Feb 13, 2022 |
| • PostgreSQL v10 | - Nov 10, 2022 | EDB Postgres Advanced v10. | - Nov 13, 2022 |
| • PostgreSQL v11 | - Nov 9, 2023 | EDB Postgres Advanced v11 | - Nov 20, 2023 |
| • PostgreSQL v12 | - Nov 14, 2024 | EDB Postgres Advanced v12 | - Dec 9, 2024 |
| • PostgreSQL v13 | - Nov 13, 2025 | EDB Postgres Advanced v13 | - Nov 16, 2025 |
| • PostgreSQL v14 | - Nov 12, 2026 | EDB Postgres Advanced v14 | - TBA |

Sources:

<https://www.postgresql.org/support/versioning/>
<https://www.enterprisedb.com/product-compatibility>

Upgrade methods

Minor version upgrade

- Stop Postgres / EPAS service
 - `systemctl stop edb-as-xx.service` OR
 - `systemctl stop postgresql-xx.service`
- Upgrade Postgres /EPAS binaries
 - `yum update edb-asxx-server` OR
 - `yum update postgresqlxx`
- Start Postgres / EPAS service
 - `systemctl start edb-as-xx.service` OR
 - `systemctl start postgresql-xx.service`

xx= version number

Upgrade methods

Major version upgrade

Upgrading with downtime

- Using `pg_dump` and restore
- Using `pg_upgrade`

Upgrade methods

Major version upgrade

Using `pg_dump` and restore method

- Use the `pg_dumpall` to dump all the global objects (roles and tablespaces)
- Use `pg_dump` (in plain, directory, custom archive, or tar format)
- Use `pg_dump` and `pg_dumpall` from the new version binaries
- Create any tablespaces in the new cluster
- Import global objects first (`pg_dumpall` output) into the new cluster
- Restore `pg_dump` plain format output using `psql`.
- All other formats outputs can be restored using `pg_restore`

Upgrade methods

Major version upgrade

Using pg_dump and restore method

- **Advantages**
 - One can easily rollback the upgrade and move to older version cluster.
 - Older version cluster can still be used for read workload during the upgrade
- **Disadvantages**
 - Extended downtime while restore is running in the new cluster
 - Twice the amount of storage is required
 - Standbys will need to be rebuilt

Upgrade methods

Major version upgrade

Using the pg_upgrade method

- This method is used for in-place upgrade
- Install the new version of software and run initdb to initialize the new cluster
- Run pg_upgrade utility from the new version binaries to perform the upgrade
- Two options available:
 - COPY mode – data files are copied to the new cluster
 - LINK mode – hard links are made to the data files from the old cluster to the new cluster

Upgrade methods

Major version upgrade

Using the pg_upgrade method

- **Advantages:**
 - No need to setup a new server. The new version is installed on the same server.
 - Faster upgrade
 - Upgrade with link mode is fastest and only requires few minutes of downtime
- **Disadvantages:**
 - COPY mode requires twice the amount of storage space
 - Not easy to roll back the upgrade when LINK mode is used.
 - Standbys need to be rebuilt

Upgrade methods

Major version upgrade

Upgrading with no/near zero downtime using logical/trigger based replication

- PostgreSQL logical replication (v10 and above)
- pglogical (9.4 and above)
- EDB Replication Server
- BDR

Upgrade methods

Major version upgrade

Upgrading with no/near zero downtime using logical replication

- Native logical replication is only available in PostgreSQL v10 and above and can be used to setup replication between nodes running different Postgres versions.
- BDR is PostgreSQL extension to build SMR/MMR between different Postgres clusters running different Postgres versions.
- EDB Replication server is a tool to setup SMR/MMR replication between different Postgres clusters running different Postgres versions.
- Setup a publication for all tables on old version cluster and a subscription on the new version
- Stop the application, once subscriber is in sync, stop the replication, and shutdown the old cluster, Restart the application.

Upgrade methods

Major version upgrade

Upgrading with no/near zero downtime using logical replication

- Advantages:
 - Little or no downtime during upgrade
 - One can easily failback to older version
 - Standbys don't need to be rebuilt
- Disadvantages:
 - Some extra effort is required to setup replication
 - Primary keys need to be added to every table (Can use REPLICA IDENTITY FULL)
 - Initial snapshot can take long time for large databases.
 - Sequences/LOBs need to be migrated manually.

Common upgrade issues

- Some changes that can break things. For example:
 - In PostgreSQL v10, pg_xlog was renamed to pg_wal and some admin functions were also renamed. This can break some maintenance/monitoring scripts.
 - Removal of WITH OIDs in PostgreSQL v12
 - Recovery.conf settings moved to postgresql.conf in v12
- Extensions and versions supported.
 - Not all versions of PostGIS are supported with every release of PG. (Refer to PostGIS version support Matrix for more info: <https://trac.osgeo.org/postgis/wiki/UsersWikiPostgreSQLPostGIS>)
- Modules/packages not supported in the new version.

Upgrade process using pg_upgrade

Initialize new version binaries and initialize the database

```
[root@trn-vm edbplus]# cd /var/lib/edb/as13
[root@trn-vm as13]# ls
backups data data1 recover
[root@trn-vm as13]# mkdir upgrade
[root@trn-vm as13]# ls -l
total 8
drwx-----. 2 enterprisedb enterprisedb 6 Nov 10 2020 backups
drwx-----. 21 enterprisedb enterprisedb 4096 Nov 8 00:00 data
drwx-----. 21 enterprisedb enterprisedb 4096 Jul 10 00:00 data1
drwxr-xr-x. 2 enterprisedb enterprisedb 6 May 11 15:43 recover
drwxr-xr-x. 2 root root 6 Nov 8 09:24 upgrade
[root@trn-vm as13]# chown enterprisedb:enterprisedb upgrade
[root@trn-vm as13]# su - enterprisedb
Last login: Thu Nov 4 11:48:20 EDT 2021 on pts/1
[enterprisedb@trn-vm ~]$ /usr/edb/as13/bin/initdb -D /var/lib/edb/as13/upgrades
The files belonging to this database system will be owned by user "enterprisedb".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.
```

Upgrade process using pg_upgrade

- Change port number of the new cluster:

```
snap_functions.sql ok
dblink_ora.sql ok
sys_stats.sql ok
finalizing initial databases ... ok
syncing data to disk ... ok
```

```
initdb: warning: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.
```

Success. You can now start the database server using:

```
/usr/edb/as13/bin/pg_ctl -D /var/lib/edb/as13/upgrades -l logfile start
```

```
[enterprisedb@trn-vm ~]$ cd /var/lib/edb/as13/upgrades
[enterprisedb@trn-vm upgrades]$ ls
base                pg_hba.conf        pg_notify          pg_stat            pg_twophase        postgresql.auto.conf
global              pg_ident.conf      pg_replslot       pg_stat_tmp       PG_VERSION         postgresql.conf
pg_commit_ts        pg_logical          pg_serial          pg_subtrans       pg_wal
pg_dynshmem         pg_multixact       pg_snapshots      pg_tblspc         pg_xact
[enterprisedb@trn-vm upgrades]$ vi postgresql.conf
```

Upgrade process using pg_upgrade

- Review all the installed extensions in the current cluster and install them in the new cluster

```

[enterprise@trn-vm upgrades]$ psql edb
psql.bin (9.6.16.23)
Type "help" for help.

```

```

edb=# \dx

```

List of installed extensions			
Name	Version	Schema	Description
adminpack	1.0	pg_catalog	administrative functions for PostgreSQL
dblink	1.2	enterprisedb	connect to other PostgreSQL databases from within a database
dbms_scheduler	1.0	sys	Creates catalog objects for DBMS_SCHEDULER package
edb_dblink_libpq	1.0	pg_catalog	EnterpriseDB Foreign Data Wrapper for PostgreSQL
edb_dblink_oci	1.0	pg_catalog	EnterpriseDB Foreign Data Wrapper for Oracle
edbspl	1.0	pg_catalog	EDB-SPL procedural language
file_fdw	1.0	enterprisedb	foreign-data wrapper for flat file access
pgagent	3.4	pgagent	A PostgreSQL job scheduler
pldbgapi	1.0	pg_catalog	server-side support for debugging PL/pgSQL functions
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language
postgis	2.3.5	enterprisedb	PostGIS geometry, geography, and raster spatial types and functions
postgres_fdw	1.0	enterprisedb	foreign-data wrapper for remote PostgreSQL servers

(12 rows)

Upgrade process using pg_upgrade

Upgrade PostGIS to the version supported on v13

- `yum upgrade edb-as96-postgis-3.1.1 -y`

Upgrade process using pg_upgrade

- Run pg_upgrade with compatibility check:

```
[enterprisedb@trn-vm tmp]$ /usr/edb/as13/bin/pg_upgrade -b /opt/edb/as9.6/bin -B /usr/edb/as13/bin -d /opt/edb/as9.6/data -D /var/lib/edb/as13/upgrades -p 5444 -P 9544 -c
Performing Consistency Checks
-----
Checking cluster versions                                ok
Checking database user is the install user              ok
Checking database connection settings                   ok
Checking for prepared transactions                     ok
Checking for reg* data types in user tables             ok
Checking for contrib/isn with bigint-passing mismatch  ok
Checking for tables WITH OIDS                           fatal

Your installation contains tables declared WITH OIDS, which is not
supported anymore. Consider removing the oid column using
ALTER TABLE ... SET WITHOUT OIDS;
A list of tables with the problem is in the file:
tables_with_oids.txt

Failure, exiting
```

Upgrade process using pg_upgrade

- Get more information about tables with OIDs:

```
[enterprisedb@trn-vm tmp]$ cat tables_with_oids.txt
In database: edbstore
  _edb_replicator_pub.rrst_edbuser_emp
  _edb_replicator_pub.rrst_edbuser_dept
In database: reptest
  _edb_replicator_pub.rrst_public_t2
  _edb_replicator_pub.rrst_public_dept
  _edb_replicator_pub.rrst_public_emp
  _edb_replicator_pub.rrst_public_emp2
  _edb_replicator_pub.rrst_public_t1
In database: roy
  _edb_replicator_pub.rrst_testschema_t1
In database: testdb
  _edb_replicator_pub.rrst_testschema_t1
  _edb_replicator_pub.rrst_testschema_books
  _edb_replicator_pub.rrst_testschema_mmr_seq_tbl
[enterprisedb@trn-vm tmp]$ █
```

- Fix above issues and re-run the `pg_upgrade -check`.
- Proceed with the upgrade when `pg_upgrade -check` returns OK for all the checks.

Upgrade best practices

- Carefully read and understand the bug fixes, enhancements, deprecated features in each release
- Pick the upgrade method that is best suited for your downtime requirements without adding any complexities.
- When running `pg_upgrade` in copy mode, make sure to allocate twice the amount of storage on the server.
- Install all required extensions/libraries in the new version prior to upgrading.
- Make sure to upgrade PostGIS to the version that is supported in the new version. There might be an intermediate steps involved if you are migrating from a very old release to a latest version.
- Take a backup of your database prior to the upgrade
- Run `analyze` after the upgrade to refresh all the necessary statistics.
- Test all your application features with the upgraded database in test environment.

Resources

- **PG_UPGRADE DOC:** <https://www.postgresql.org/docs/14/pgupgrade.html>
- **POSTGIS UPGRADE DOC:** https://www.enterprisedb.com/docs/postgis/latest/03_upgrading_postgis/
- **EPAS UPGRADE GUIDE:** https://www.enterprisedb.com/docs/epas/latest/epas_upgrade_guide/
- **UPGRADE BLOG:** <https://www.enterprisedb.com/blog/how-to-upgrade-postgresql-and-upgrading-methods-costs-risks>
- **PG END OF SUPPORT DATES:** <https://www.postgresql.org/about/featurematrix/>
- **EPAS END OF SUPPORT DATES:** <https://www.enterprisedb.com/product-compatibility>
- **POSTGRES VERSIONING:** <https://www.postgresql.org/support/versioning/>
- **POSTIGIS VERSION INFO:** <https://trac.osgeo.org/postgis/wiki/UsersWikiPostgreSQLPostGIS>
- **WEBINAR OIN UPGRADE METHODS**
https://techsupport.enterprisedb.com/customer_portal/videos/ba8f2490c6db480ea77361f0dc5b2183/

Demo

- Demo of upgrading EDB Postgres Advanced Server from 9.6 to v13 using pg_upgrade method.



Q&A