



# EDB Customer Office Hour: PostgreSQL Upgrade Best Practices

Customer Success Team

Sep 2022



# SPEAKERS



**Manish Yadav**  
**Customer Success Solution Engineer**  
**EDB**



**Rajendra Somthankar**  
**Principal Consultant**  
**EDB**

# AGENDA

- Postgres Version Nomenclature
- Reasons for Upgrade
- Upgrade Methods
- Common Upgrade Issues
- Best Practices
- Demo & Customer Experiences
- Q&A

# VERSION NOMENCLATURE

## PostgreSQL

Major/minor versions: 10.x, 11.x, 12.x, 13.x, 14.x...

Example

PostgreSQL 13.4 → Community Minor version  
→ Major version

## EDB Postgres Advanced Server (EPAS)

Major/minor versions: 10.x.y, 11.x.y, 12.x.y, 13.x.y, 14.x.y...

Example,

EPAS 13.4.12 → EDB Minor version  
→ Community Minor Version  
→ Major version

Major versions include new enhancements, functionality changes and bug fixes. Released – Yearly  
Minor versions include security and bug fixes. Released - Quarterly.



# REASONS FOR UPGRADE

- New features/Enhancements
- Security/Bug Fixes
- End of Support for the Release
- Compliance Reasons

# NEW FEATURES/ENHANCEMENTS IN EACH RELEASE

PostgreSQL/ EPAS Version	No. of Enhancements	Major Highlights
Version 10	11	PG – introduced partitioning, logical replication, SCRAM authentication & parallelism improvements. EPAS – improved auditing, clone schema, automatic prewarm, etc.
Version 11	12	PG – partitioning improvements, JIT compilation, Stored procedure, covering index. EPAS – data redaction, autonomous transactions, edb_wait_states, etc.
Version 12	17	PG – partitioning and index improvements, with OIDs, recovery.conf file changes. EPAS – interval partitioning, Oracle compatible views.
Version 13	15	PG – faster vacuum, pg_catcheck, verify full backup, logical replication for partitioned tables, partition wise joins. EPAS – Oracle compatibility features.
Version 14	12	PG – Multi ranges, Query pipelining, logical replication stream in-progress transactions. EPAS – Object level auditing, full support for CONNECT by, sub-partitioning, full support to EDB Postgres Distributed (BDR)
Version 15	29	PG – Merge, row/column filtering with logical replication, pg_basebackup compression, pg_walinspect, privileges for config parameters EPAS – Transparent Data Encryption, password obfuscation for pg_hba.conf, Oracle compatibility functions



# SECURITY VULNERABILITY FIXES

- Every new version of Postgres/EPAS contains fixes for bugs and any security vulnerabilities
- Following critical security vulnerabilities were identified and fixed in latest minor releases:
- **CVE-2022-2625 : Extension scripts replace objects not belonging to the extension**
- Blog : <https://www.enterprisedb.com/blog/postgresql-extensions-impacted-cve-2022-2625-privilege-escalation>
- **CVE-2021-23214: Server processes unencrypted bytes from man-in-the-middle**
- **CVE-2021-23222: libpq processes unencrypted bytes from man-in-the-middle**
- Blog : <https://www.enterprisedb.com/blog/postgres-mitm21-vulnerabilities>

# END OF SUPPORT PER RELEASE

	PostgreSQL	EPAS
<b>Version 10</b>	<b>10 Nov 2022</b>	<b>13 Nov 2022</b>
Version 11	09 Nov 2023	20 Nov 2023
Version 12	14 Nov 2024	09 Dec 2024
Version 13	13 Nov 2025	16 Nov 2025
Version 14	02 Dec 2026	01 Dec 2026



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



# UPGRADE METHODS

## Minor Upgrade

- **Stop PostgreSQL**  
`systemctl stop postgresql-xx.service`
- **Upgrade PostgreSQL**  
`yum update postgresqlxx`
- **Start PostgreSQL**  
`systemctl start postgresql-xx.service`

# UPGRADE METHODS

## Major Upgrade

- **With Downtime**
  - ✓ pg\_dump and psql/pg\_restore
  - ✓ pg\_upgrade
- **With near Zero Downtime**
  - ✓ PostgreSQL Logical Replication (v10 and above)
  - ✓ EDB Postgres Distributed (BDR)
  - ✓ EDB Replication Server (xDB)

# UPGRADE METHODS

## Major Upgrade

### **pg\_dump/pg\_restore Method**

- **Advantages**

- ✓ One can easily rollback the upgrade and move to older version cluster
- ✓ Free from bloats
- ✓ 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 Upgrade

### **pg\_dump/pg\_restore Method**

- Use the pg\_dumpall to dump all the global objects (roles and tablespaces)
- Use pg\_dump {in plain(-Fp), directory(-Fd), custom archive(-Fc), tar format(-Ft)}
- Create any tablespaces in the new cluster
- Restore global objects into the new cluster
- Restore plain format(-Fp) output using psql utility
- Restore remaining formats(-Fd, -Fc, -Ft) outputs using pg\_restore

# UPGRADE METHODS

## Major Upgrade

### **pg\_upgrade Method**

- **Advantages**
  - ✓ In-place upgrade - do not require extra hardware
  - ✓ Upgrade with Link mode is faster and only catalog objects are dump and restored
- **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 Upgrade

### **pg\_upgrade Method**

- Install the new version of PostgreSQL/EPAS 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. With this option no user data transfer involved

# UPGRADE METHODS

## Major Upgrade

### Upgrading with near Zero downtime

- **Advantages**
  - ✓ Minimal downtime during upgrade
  - ✓ One can easily failback to older version
  - ✓ Standbys don't need to be rebuilt
- **Disadvantages**
  - ✓ Extra efforts required to setup replication
  - ✓ Primary keys need to be added to each table (use REPLICA IDENTITY FULL)
  - ✓ Initial snapshot takes longer time for large databases
  - ✓ Sequences/LOBs need to be migrated manually

# UPGRADE METHODS

## Major Upgrade

### Upgrading with near Zero downtime

- **Native Logical Replication** – Introduced in PGv10 and can be used to setup replication between nodes running on different Postgres versions.
- **EDB Postgres Distributed (BDR)** - PostgreSQL extension to replicate data between Postgres clusters running on different Postgres versions.
- **EDB Replication Server (xDB)**- Setup SMR/MMR replication between Postgres clusters running on different Postgres versions.

# UPGRADE METHODS

## Logical Replication

- Adjust parameters in existing database i.e.

```
Wal_level = logical, max_wal_senders,  
max_replication_slots, max_logical_replication_workers
```

- Initialize a new cluster with new version

- Create database & restore the schema from existing database i.e.

```
pg_dump -s -h old_db -no-publications -d dbpg10 | \  
psql -h new_db -d dbpg12
```



# UPGRADE METHODS

## Logical Replication (cont'd)

- Create PUBLICATION on existing database(dbpg10)

```
CREATE PUBLICATION migration FOR ALL TABLES;
```

- Create SUBSCRIPTION on new version database(dbpg12)

```
CREATE SUBSCRIPTION migration_new CONNECTION 'dbname=pgdb10 port=5432'  
PUBLICATION migration WITH (copy_data = true, slot_name = migration');
```



# UPGRADE METHODS

## Logical Replication(cont'd)

- Time to perform upgrade !!!
  - Stop application
  - Verify data synchronization
  - Applications to point a new version database server
  - Start application

# 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 BEST PRACTICES

- Carefully read and understand the bug fixes, enhancements, deprecated features in each release
- Pick the upgrade method that is best suited as per the downtime requirements without adding much complexities
- Install all required extensions/libraries in the new version prior to upgrading
- Log the timing while performing upgrade in test environment for better estimation
- Test all your application modules on new version database in test environment

# UPGRADE BEST PRACTICES

- When running `pg_upgrade` in copy mode, make sure to allocate twice the amount of storage on the server
- 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
- Should run `ANALYZE` after the upgrade to refresh all the catalog statistics



# Resources

**PG\_UPGRADE DOC:** <https://www.postgresql.org/docs/14/pgupgrade.html>

**EPAS UPGRADE GUIDE:** [https://www.enterprisedb.com/docs/epas/latest/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/>

**POSTGIS UPGRADE DOC:** [https://www.enterprisedb.com/docs/postgis/latest/03\\_upgrading\\_postgis/](https://www.enterprisedb.com/docs/postgis/latest/03_upgrading_postgis/)

**POSTGIS VERSION INFO:** <https://trac.osgeo.org/postgis/wiki/UsersWikiPostgreSQLPostGIS>

Thank you



# DEMO

# Q & A



THANK YOU  
(We can't wait to see what you create)

