NAVIGATING POSTGRESQL END OF LIFE:
UPGRADE STRATEGIES AND BEST PRACTICES

Vivek Singh

Principal Database Specialist – PostgreSQL

Amazon Web Services

Dallas/Fort Worth PostgreSQL User Group
Jan 14, 2025

AGENDA

- PostgreSQL Versioning and EOL Cycles
- Major vs Minor versions
- Risks of Not Upgrading
- Upgrade Strategies
- Pre-Upgrade Planning
- Minimizing Downtime
- Benefits of upgrading to PostgreSQL 16

AGENDA

- PostgreSQL Versioning and EOL Cycles
- Major vs Minor versions
- Risks of Not Upgrading
- Upgrade Strategies
- Pre-Upgrade Planning
- Minimizing Downtime
- Benefits of upgrading to PostgreSQL 16

POSTGRESQL VERSIONING AND EOL CYCLES

- Version Numbering: Major/Minor (e.g., 14.0, 15.1)
- Major Versions: Released annually, introduce new features
- Minor Versions: Bug fixes, security updates, no new features
- Support Lifecycle: 5 years from initial release
- End of Life (EOL): No more updates after support ends
- Community Support: Active development and security patching
- PostgreSQL 13 has EOL on November 13, 2025

MAJOR VS MINOR VERSIONS (RDS/AURORA POSTGRESQL)

	Minor upgrade	**Major upgrade**
Can upgrade Replica	Yes	Yes
Needs new custom parameter group for upgraded instance	No	Yes
Upgrades automatically (provided RDS is configured with Auto Minor Version Upgrades)	Yes	No
Updates database data files	No	Yes
Copies table statistics to upgraded instance	Yes	No
Is always backward compatible	Yes	No

AGENDA

- PostgreSQL Versioning and EOL Cycles
- Major vs Minor versions
- Risks of Not Upgrading
- Upgrade Strategies
- Pre-Upgrade Planning
- Minimizing Downtime
- Benefits of upgrading to PostgreSQL 16

RISKS OF NOT UPGRADING POSTGRESQL DATABASE

- Security Vulnerabilities
- Performance Limitations
- Compatibility Issues
- Lack of Support
- Compliance Concerns
- Increased Maintenance Costs

AGENDA

- PostgreSQL Versioning and EOL Cycles
- Major vs Minor versions
- Risks of Not Upgrading
- Upgrade Strategies
- Pre-Upgrade Planning
- Minimizing Downtime
- Post-Upgrade Testing and Validation

UPGRADE STRATEGIES

	Pros	Cons
pg_upgrade	• Fast, especially for large databases • Minimal downtime • Preserves database objects and data • Works well for minor version upgrades	• Requires sufficient disk space for both old and new clusters • Can be complex to set up, especially for inexperienced users • Not suitable if the old and new clusters have different architectures
Logical Replication	• Allows for near-zero downtime upgrades • Can upgrade across major versions • Enables partial replication of tables if needed • Allows for testing the new version before switching over	• Can be complex to set up and manage • May require schema changes to be applied manually • Not all objects are replicated (e.g., sequences, large objects) • Can put additional load on the source database
Dump and Restore	• Simple and straightforward method • Works across all PostgreSQL versions • Allows for data cleanup and reorganization during the process • Can be used to migrate to a different server or architecture	• Can be very slow for large databases • Requires significant downtime • May require a lot of disk space for the dump file • Indexes and some database objects need to be recreated
Slony or Londiste	• Allows for near-zero downtime upgrades • Can replicate across different PostgreSQL versions • Supports bi-directional replication (Slony) • Can be used for geographical distribution of data	• Complex to set up and maintain • Requires changes to the database schema • Can introduce latency in data replication • May not support all PostgreSQL features
Bucardo	• Supports multi-master replication • Can replicate between different PostgreSQL versions • Allows for near-zero downtime upgrades • Can replicate only specific tables if needed	• Complex to set up and configure • Can be slower than other methods for large data sets • Requires installation of additional software • May require resolution of conflicts in multi-master setups

PG_UPGRADE *OPTIONS*

- Pre-upgrade Checks: *--check*
- Link Mode: Utilize *--link*
- Jobs Optimization *--jobs*
- Data directories *--old-datadir* | *--new-datadir*
- Executable directory --old-bindir | --new-bindir
- Verbose Output: Enable *--verbose*
- Example:

```
/usr/pgsql-12/bin/pg_upgrade --old-bindir /usr/pgsql-11/bin --new-bindir /usr/pgsql-12/bin --old-datadir
/var/lib/pgsql/11/data --new-datadir /var/lib/pgsql/12/data --link --check
```

UPGRADE STRATEGIES FOR AURORA/RDS POSTGRESQL

Upgrade Option	Advantages and Disadvantages
In-place upgrade	• Can be performed from the Amazon RDS console or AWS CLI • Requires downtime
Blue/green deployments	• Can also be performed from the Amazon RDS console and can minimize risks and downtime • It may restrict some of the operations that a user can perform until the operation is complete
Out-of-place upgrade	• It reduces downtime because the upgrade happens on a copy of the database • It takes several steps to configure continuous replication with the new database instance until cutover happens

AGENDA

- PostgreSQL Versioning and EOL Cycles
- Major vs Minor versions
- Risks of Not Upgrading
- Upgrade Strategies
- Pre-Upgrade Planning
- Minimizing Downtime
- Post-Upgrade Testing and Validation

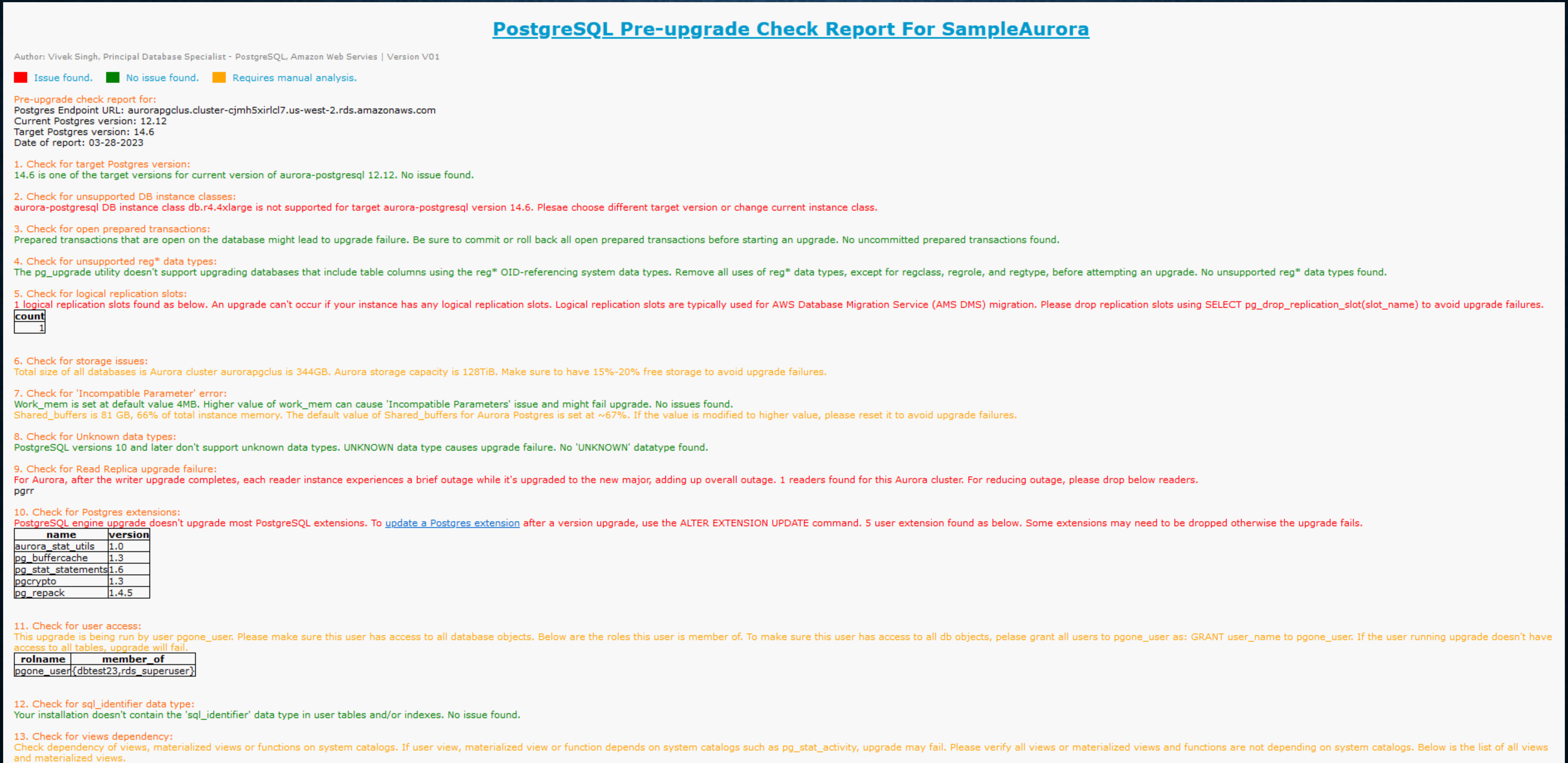
PRE UPGRADE PLANNING

- Version Compatibility Check
- Database Size Assessment
- Backup Strategy
- Test Environment Setup
- Performance Baseline
- Dependency Audit

PRE-UPGRADE CHECK FOR AURORA/RDS POSTGRESQL

- Checks on
  1. Target Postgres version
  2. Unsupported DB instance classes
  3. Open prepared transactions
  4. Unsupported reg* data types
  5. Logical replication slots
  6. Storage issues
  7. 'Incompatible Parameter' error
  8. Unknown data types
  9. Read replica upgrade failure
  10. Postgres extensions
  11. User access
  12. 'sql_identifier' data type
  13. Views dependency
  14. Incorrect primary user name
  15. Maintenance task
- link for the script

PRE-UPGRADE CHECK FOR AURORA/RDS POSTGRESQL – SAMPLE REPORT



PostgreSQL Pre-upgrade Check Report For SampleAurora

Author: Vivek Singh, Principal Database Specialist - PostgreSQL, Amazon Web Servies | Version V01

🟥 Issue found. 🟩 No issue found. 🟧 Requires manual analysis.

Pre-upgrade check report for:
Postgres Endpoint URL: aurorapgclus.cluster-cjmh5xirlcl7.us-west-2.rds.amazonaws.com
Current Postgres version: 12.12
Target Postgres version: 14.6
Date of report: 03-28-2023

1. Check for target Postgres version:
14.6 is one of the target versions for current version of aurora-postgresql 12.12. No issue found.

2. Check for unsupported DB instance classes:
aurora-postgresql DB instance class db.r4.4xlarge is not supported for target aurora-postgresql version 14.6. Plesae choose different target version or change current instance class.

3. Check for open prepared transactions:
Prepared transactions that are open on the database might lead to upgrade failure. Be sure to commit or roll back all open prepared transactions before starting an upgrade. No uncommitted prepared transactions found.

4. Check for unsupported reg* data types:
The pg_upgrade utility doesn't support upgrading databases that include table columns using the reg* OID-referencing system data types. Remove all uses of reg* data types, except for regclass, regrole, and regtype, before attempting an upgrade. No unsupported reg* data types found.

5. Check for logical replication slots:
1 logical replication slots found as below. An upgrade can't occur if your instance has any logical replication slots. Logical replication slots are typically used for AWS Database Migration Service (AMS DMS) migration. Please drop replication slots using SELECT pg_drop_replication_slot(slot_name) to avoid upgrade failures.

count
1

6. Check for storage issues:
Total size of all databases is Aurora cluster aurorapgclus is 344GB. Aurora storage capacity is 128TiB. Make sure to have 15%-20% free storage to avoid upgrade failures.

7. Check for 'Incompatible Parameter' error:
Work_mem is set at default value 4MB. Higher value of work_mem can cause 'Incompatible Parameters' issue and might fail upgrade. No issues found.
Shared_buffers is 81 GB, 66% of total instance memory. The default value of Shared_buffers for Aurora Postgres is set at ~67%. If the value is modified to higher value, please reset it to avoid upgrade failures.

8. Check for Unknown data types:
PostgreSQL versions 10 and later don't support unknown data types. UNKNOWN data type causes upgrade failure. No 'UNKNOWN' datatype found.

9. Check for Read Replica upgrade failure:
For Aurora, after the writer upgrade completes, each reader instance experiences a brief outage while it's upgraded to the new major, adding up overall outage. 1 readers found for this Aurora cluster. For reducing outage, please drop below readers.
pgrr

10. Check for Postgres extensions:
PostgreSQL engine upgrade doesn't upgrade most PostgreSQL extensions. To update a Postgres extension after a version upgrade, use the ALTER EXTENSION UPDATE command. 5 user extension found as below. Some extensions may need to be dropped otherwise the upgrade fails.

name	version
aurora_stat_utils	1.0
pg_buffercache	1.3
pg_stat_statements	1.6
pgcrypto	1.3
pg_repack	1.4.5

11. Check for user access:
This upgrade is being run by user pgone_user. Please make sure this user has access to all database objects. Below are the roles this user is member of. To make sure this user has access to all db objects, pelase grant all users to pgone_user as: GRANT user_name to pgone_user. If the user running upgrade doesn't have access to all tables, upgrade will fail.

rolname	member_of
pgone_user	{dbtest23,rds_superuser}

12. Check for sql_identifier data type:
Your installation doesn't contain the 'sql_identifier' data type in user tables and/or indexes. No issue found.

13. Check for views dependency:
Check dependency of views, materialized views or functions on system catalogs. If user view, materialized view or function depends on system catalogs such as pg_stat_activity, upgrade may fail. Please verify all views or materialized views and functions are not depending on system catalogs. Below is the list of all views and materialized views.

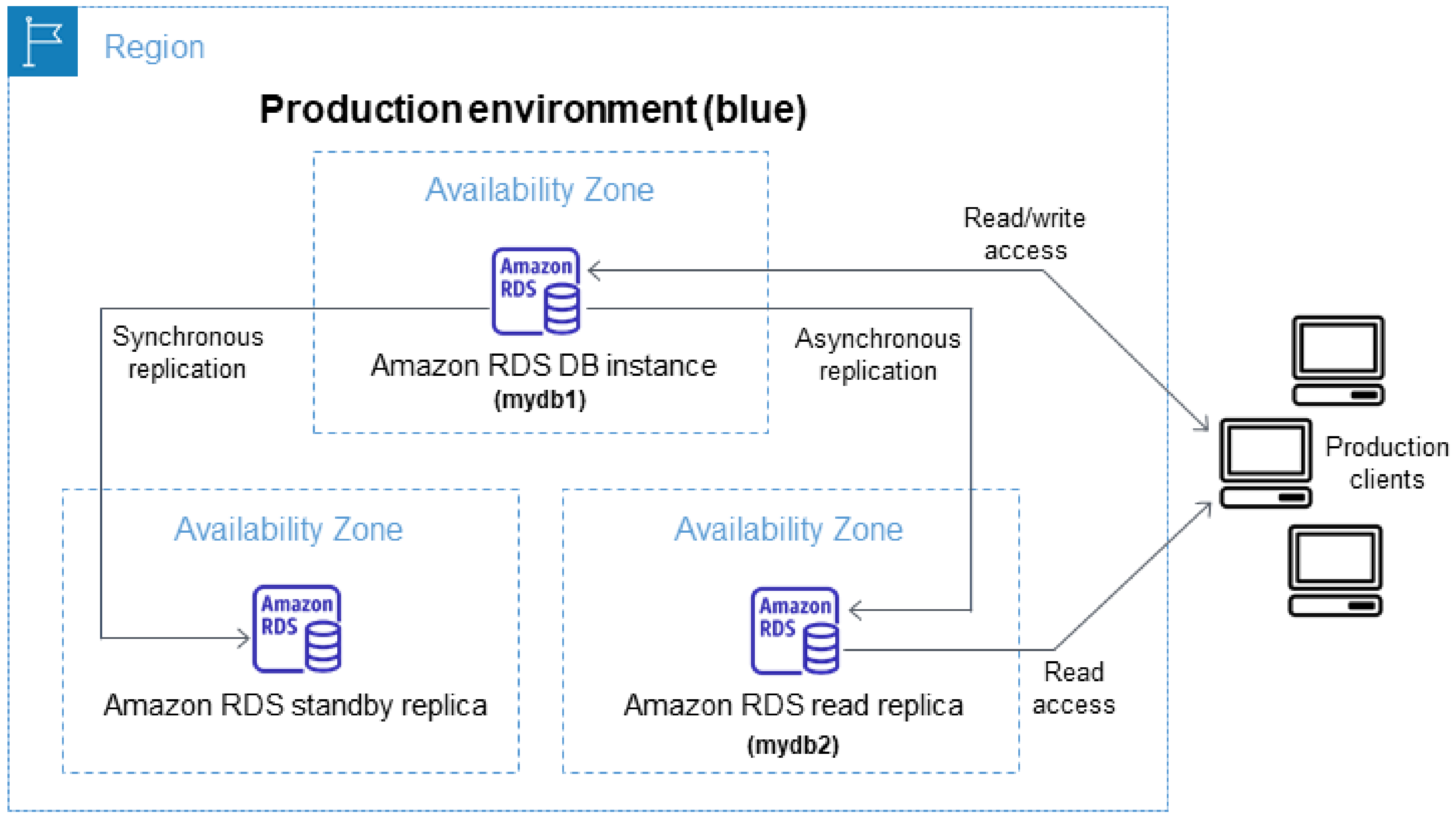
AGENDA

- PostgreSQL Versioning and EOL Cycles
- Major vs Minor versions
- Risks of Not Upgrading
- Upgrade Strategies
- Pre-Upgrade Planning
- Minimizing Downtime
- Benefits of upgrading to PostgreSQL 16

MINIMIZING DOWNTIME

- Logical Replication
- Controlled Switchover
- Connection Pooling
- Phased Migration
- Data Validation
- Rollback Plan

AMAZON RDS BLUE/GREEN DEPLOYMENTS



POST-UPGRADE TESTING AND VALIDATION

- Functionality Testing
- Performance Benchmarking
- Application Integration
- Data Integrity Checks
- Security Validation
- Parameters/ Monitoring Setup

AGENDA

- PostgreSQL Versioning and EOL Cycles
- Major vs Minor versions
- Risks of Not Upgrading
- Upgrade Strategies
- Pre-Upgrade Planning
- Minimizing Downtime
- Benefits of upgrading to PostgreSQL 16

BENEFITS OF UPGRADING TO POSTGRESQL 16

- **Performance improvements**
  - New optimizations in query planner
  - Bulk loading
  - Load balancing for libpq
  - Improved vacuum
  - Performance boosts

- **Logical replication**
  - Bidirectional Replication
  - Logical replication from standby (only for RDS PostgreSQL)
  - Performance enhancements for logical replication

BENEFITS OF UPGRADING TO POSTGRESQL 16

- **Monitoring**
  - IO monitoring
  - pg_stat_all_tables
  - pg_stat_all_indexes
  - Improved 'auto_explain'
  - Improved query tracking

- **Security**
  - More Detailed Access Control Options
  - Security-Focused Client Connection Parameters
  - Kerberos Credential Delegation

DEMO – POSTGRESQL 12 VS16 - BENCHMARKING

- Video detached due to size

RESOURCES

- Announcement: Amazon Aurora PostgreSQL 12.x end of standard support is February 28, 2025
- Announcement: Amazon RDS PostgreSQL 12 end of standard support is February 28, 2025
- Aurora user guide: Overview of Amazon Aurora Blue/Green Deployments
- AWS blog: Estimating the charges for Amazon RDS Extended Support
- AWS blog: Upgrade your Amazon RDS for PostgreSQL or Amazon Aurora PostgreSQL database, Part 1: Comparing upgrade approaches
- AWS blog: Best practices for upgrading Amazon RDS to major and minor versions of PostgreSQL

Q&A

Thank you!

Vivek Singh

Principal Database Specialist
Amazon Web Services

LinkedIn