

The Ultimate PostgreSQL Upgrade Checklist



This checklist is a simple roadmap for seasoned and new DBAs and developers taking on a PostgreSQL upgrade. It can be used as a guide to help navigate the complex process of upgrading, minimizing risks, and maximizing the efficiency of a new PostgreSQL version.

Complete a pre-upgrade assessment

Review the release notes of the target PostgreSQL version to understand new features and any deprecated functionalities.

Check the compatibility of your current systems and infrastructure with the new PostgreSQL version.

Conduct a detailed performance analysis of your current database to set a benchmark for post-upgrade comparison.

Prepare a detailed upgrade plan

Clearly define each step in the upgrade process, encompassing aspects like backups, thorough testing, and the actual upgrade procedure.

Distribute specific tasks among any team members, ensuring that responsibilities for each upgrade phase are explicitly assigned and understood.

Execute and test a complete backup

Use tools like `pg_dump` to make a comprehensive backup of the current database, including all vital schemas, tables, and objects.

After backup, verify the file for completeness and integrity, ensuring no data corruption.

Conduct a test restore in a staging environment to confirm the data's restorability and functionality.

Document the backup process details, noting the date, time, and method used for future reference.

Store the backup in a secure, reliable location, ideally off-site or in the cloud, to protect against data loss.

Set up a staging environment for a trial upgrade

Establish a staging environment that mirrors the production setting regarding hardware, software, and configurations.

Check that all extensions are compatible with the new PostgreSQL version, updating or replacing them as required.

Perform a trial upgrade in the staging environment to pinpoint potential issues and refine the upgrade strategy.

Test all applications linked to the database for compatibility with the upgraded PostgreSQL version, focusing on functionality, performance, and stability.

Evaluate the performance of the staging environment after the upgrade to compare with the pre-upgrade performance metrics of the existing database.

Verify the integrity of the data and the correctness of transactions to ensure there is no data corruption or loss.

Prepare for emergencies

Formulate a detailed rollback plan to swiftly revert to the previous PostgreSQL version in case of critical issues.

Ensure that the team is adequately trained and prepared to execute the rollback plan efficiently if needed.

Perform simulations to assess the effectiveness and readiness of the rollback strategy.

Keep clear and easily accessible documentation of emergency procedures for quick reference in urgent situations.

Confirm that backups are readily accessible and can be restored efficiently if an emergency rollback is necessary.

Schedule the upgrade

Choose a time for the upgrade when system usage is minimal, usually during off-peak hours, to reduce disruption.

Inform all key stakeholders, such as management, IT staff, and end users, about the scheduled downtime and expected duration.

Allocate resources

Verify that the existing hardware meets the requirements for the new PostgreSQL version, upgrading or enhancing components as necessary.

Ensure sufficient storage space for the upgrade process, including backups and temporary files.

Assess and potentially increase memory and CPU capacity to support the upgrade and post-upgrade operations.

Arrange for additional resources, like extra servers or increased bandwidth, to handle any unexpected demands during the upgrade.

Review and update configurations

Carefully examine existing PostgreSQL configurations, paying special attention to parameters likely impacted by the upgrade.

Update and fine-tune configurations to maximize the benefits of new features and enhancements in the upgraded version.

Maintain a detailed log of all modifications made to the configurations for future reference and auditing purposes.

Perform the upgrade

Decide on an in-place upgrade or a dump-and-restore method, considering factors like the environment, database size, and acceptable downtime.

Confirm that staging and production environments are fully prepared, with all pre-upgrade checks completed.

Start the upgrade process following the selected approach.

Closely monitor the upgrade for errors, warnings, or unusual activities, and be prepared to intervene if necessary.

Complete post-upgrade validation & optimization

After the upgrade, conduct extensive tests on the database to ensure its functionality, focusing on data integrity and performance.

Check that all applications interacting with the database are connecting correctly and operating as expected.

Compare the performance of the upgraded database with the pre-upgrade benchmarks.

Test backup and recovery procedures in the new environment to confirm data safety.

Review system logs and error reports for any unusual occurrences or issues post-upgrade.

Identify any performance bottlenecks or issues and carry out necessary optimizations or adjustments.

Check the data

Confirm that all data has been successfully transferred to the upgraded version without missing records.

Implement thorough data integrity checks to confirm the consistency and accuracy of data after the upgrade.

Take snapshots of crucial data sets before and after the upgrade to compare and ensure data consistency.

Review transaction logs for any irregularities or errors that could signal issues with data integrity.

Execute key database queries and generate key reports to verify their correct functioning and the accuracy of results.

Ensure all database indexes and constraints are intact and operating as expected post-upgrade.

Confirm that users and applications have the appropriate access to the data and that permissions and roles are correctly maintained.

Review security features

Examine the security settings in the upgraded PostgreSQL environment, including access controls and user permissions, to align with organizational security policies and compliance standards.

Modify and strengthen security policies and settings to address new features or changes introduced in the upgraded PostgreSQL version.

Perform tests to assess the effectiveness of security protocols, like intrusion detection systems, firewalls, and encryption methods.

Reevaluate and fine-tune user roles and privileges to ensure they provide minimum necessary access, minimizing potential security risks.

Update documentation

Record each step of the upgrade process, noting any challenges faced and how they were resolved for future reference.

Revise all operational documents and guidelines to reflect the changes and new functionalities in the updated PostgreSQL version.

Educate the team about new features, deprecated functionalities, and any changes in the PostgreSQL upgrade that may affect their work.

Update the backup and recovery documentation to align with the new system's configuration and capabilities.

Revise system monitoring procedures and documentation to efficiently monitor the performance and health of the upgraded PostgreSQL system.

Set up and continue monitoring

Consistently monitor the upgraded PostgreSQL database using tools like Percona Monitoring and Management to identify and resolve any post-upgrade issues.

Regularly analyze performance metrics and adjust database configurations for optimal performance and efficiency.

Scrutinize system logs for early indications of problems or potential areas for improvement.

Keep an eye on how applications interact with the database after the upgrade to ensure smooth integration and performance.

Establish routine assessments to evaluate and enhance database operations proactively.

We hope this general checklist will guide you seamlessly through your PostgreSQL upgrade. However, should you encounter any challenges or require specialized assistance, Percona is here to help. Whether you need continuous database management, strategic support for migrations, database design, high-availability (HA) and disaster recovery (DR) planning, or other complex projects, our team of PostgreSQL experts has the proficiency to meet your needs.

At Percona, we distinguish ourselves through a unique combination of technical excellence, steadfast commitment to open source, technology-agnostic expertise, and a deep focus on customer success. Partner with us for a comprehensive upgrade service tailored to fit your budget and objectives, ensuring a smooth transition to the latest PostgreSQL version.

[Learn more about Percona upgrade services](#)