SQL Server Upgradation Process

Upgrade Step	Description	Key Actions
Upgrade Type	Choose the appropriate upgrade method: in-place or side-by-side.	- **In-Place Upgrade:** Upgrades the existing instance. - **Side-by-Side Upgrade:** Installs a new instance and migrates databases. - Choose based on downtime and complexity.
Pre-Upgrade Checklist	Perform system checks, backups, and tests before upgrading SQL Server.	- Verify system requirements for the target SQL Server version. - Take full backups of all databases (full, differential, log). - Test compatibility using SQL Server Data Migration Assistant (DMA).
Hardware/OS Compatibility	Ensure the hardware and OS meet the requirements for the new SQL Server version.	- Confirm CPU, memory, and storage specifications. - Verify OS version compatibility. - Check whether additional resources are needed for new features.
Service Pack and Cumulative Updates	Apply the latest service pack or cumulative update for the current SQL Server version before upgrading.	- Update the existing SQL Server to the latest service pack or cumulative update. - This ensures a smoother upgrade process and fewer compatibility issues.

Database Compatibility Level Check	Verify the database compatibility level for any breaking changes in the new version.	- Identify compatibility level of databases. - Plan to upgrade the compatibility level to leverage new features. - Use the `ALTER DATABASE` command to change compatibility.
System Database Backups	Take backups of system databases (master, model, msdb) before the upgrade.	- Back up system databases manually, as they contain critical configuration settings. - Copy these backups to a safe location.
Disk Space Availability	Ensure sufficient disk space for upgrade files, logs, and database growth.	- Verify available disk space, especially for TempDB and log files. - Clean up unnecessary files before the upgrade.
Stop Maintenance Jobs	Disable or stop any SQL Server Agent jobs before starting the upgrade process.	- Disable SQL Server Agent jobs (e.g., backups, index maintenance) during the upgrade process to avoid conflicts.
Upgrade SQL Server Instance	Perform the actual upgrade using SQL Server Setup Wizard or command-line tool.	- Use SQL Server Setup (`setup.exe`) to begin the upgrade. - Follow wizard instructions to upgrade instance, features, and components (e.g., SSIS, SSAS, SSRS).
Post-Upgrade Health Check	Verify the health of the SQL Server instance and databases after the upgrade.	- Check the SQL Server logs for any errors or warnings. - Ensure all databases are online and accessible. - Test key functionality (queries, jobs, backups).

Database Compatibility Updates	Update the database compatibility level if required to enable new features or improve performance.	- Change compatibility level after confirming the application is compatible. - Use the Query Store to monitor the impact of compatibility changes.
Rebuild Indexes/Update Statistics	Rebuild indexes and update statistics to improve query performance on the upgraded system.	- Perform index maintenance on all databases to account for any schema changes. - Update statistics using `sp_updatestats` or custom maintenance jobs.
Re-enable Maintenance Jobs	Enable or resume SQL Server Agent jobs after confirming the system is stable.	- Re-enable disabled jobs (backups, index rebuilds) and monitor for any issues. - Verify that SQL Agent jobs run as expected post-upgrade.
Security and Permissions Check	Ensure that database security settings, logins, and permissions are intact after the upgrade.	- Check that logins and users are properly mapped to databases. - Ensure permissions are intact for all roles. - Test application connectivity and access.
Post-Upgrade Monitoring	Continuously monitor performance and error logs for at least a week after the upgrade.	- Use SQL Server Profiler or Extended Events to track performance. - Set up monitoring for CPU, memory, I/O, and query performance. - Address any post-upgrade issues promptly.

Backup After Upgrade	Take a fresh full backup of all databases once the upgrade has been completed.	- Perform a full backup of all upgraded databases to ensure they are restorable in the new version. - Store backups in a secure location.
Documentation and Reporting	Document the upgrade process, issues encountered, and the final configuration.	- Create a detailed report of the upgrade, including any performance improvements and steps for rollback. - Keep documentation for auditing and future reference.

Types of SQL Server Upgrades

Upgrade Type	Description	Advantages	Disadvantages
In-Place Upgrade	Upgrades the existing SQL Server instance to a new version.	- Less complex, no need to migrate databases manually. - Shorter downtime compared to side-by-side upgrades.	- No rollback option other than restoring from backup. - Can potentially inherit issues from the old system.
Side-by-Side Upgrade	A new SQL Server instance is installed alongside the old one, and databases are migrated.	- Allows for rollback by retaining the old system. - Less risk of downtime if migration is planned properly.	- Requires more time and resources for setup and migration. - May require additional hardware or virtual machines.

SQL Server Version Compatibility

Source Version	Target Version (Supported)	Upgrade Path
SQL Server 2008/R2	SQL Server 2012, SQL Server 2014, SQL Server 2016, SQL Server 2017, SQL Server 2019	Direct upgrade supported to newer versions up to SQL Server 2019.
SQL Server 2012	SQL Server 2014, SQL Server 2016, SQL Server 2017, SQL Server 2019, SQL Server 2022	Direct upgrade to these versions.
SQL Server 2014	SQL Server 2016, SQL Server 2017, SQL Server 2019, SQL Server 2022	Direct upgrade to these versions.
SQL Server 2016	SQL Server 2017, SQL Server 2019, SQL Server 2022	Direct upgrade to these versions.
SQL Server 2017	SQL Server 2019, SQL Server 2022	Direct upgrade to SQL Server 2019 and SQL Server 2022.
SQL Server 2019	SQL Server 2022	Direct upgrade to SQL Server 2022.

Best Practices for SQL Server Upgrade

Best Practice	Details
Test in a Non-Production Environment	Always test the upgrade in a development or staging environment before upgrading the production instance. Ensure all applications work correctly with the new version.
Back Up Everything	Take full backups of all databases, system databases (master, msdb, model), and configuration files before starting the upgrade.
Use SQL Server Data Migration Assistant (DMA)	Analyze database compatibility, feature parity, and performance using DMA to assess potential issues during the upgrade process.
Plan for Downtime	Communicate the upgrade schedule to stakeholders and plan for sufficient downtime to accommodate the upgrade and any unforeseen issues.
Monitor Closely Post-Upgrade	After the upgrade, closely monitor performance, error logs, and the functioning of key services for at least a week to ensure everything is running smoothly.
Rollback Plan	Have a rollback plan in place (restore backups or revert to the old instance in case of side-by-side upgrades) in case the upgrade fails or encounters issues.

Summary

This table provides a comprehensive guide to SQL Server upgrades, covering pre-upgrade planning, the actual upgrade process, and post-upgrade tasks. By following this structured process and best practices, you can ensure a successful SQL Server upgrade with minimal downtime and risks.