SQL Server 2017 & 2019 Side-by-Side Installation Guide

Dr. Emmanuel M. Ekukole

verification checklist & quick troubleshooting guide included :)

SQL Server 2017 & 2019 Side-by-Side Installation Guide

Dr. Emmanuel M. Ekukole

Date: December 28, 2025
Version: 1.0
Environment: Windows 11 Pro | Dell Latitude 7490 | 16GB RAM | 512GB SSD

Table of Contents

- Overview
- Prerequisites
- **Part 1:** Clean Uninstallation
- **Part 2:** SQL Server 2017 Installation
- **Part 3:** SQL Server 2019 Installation
- **Part 4:** Post-Installation Configuration
- Troubleshooting Guide
- Verification Checklist
- Lessons Learned

default instance

named instance

now go grab a bottle of your favorite drink and let's do this!

Overview

Purpose

This guide documents the complete process of installing SQL Server 2017 and SQL Server 2019 Enterprise editions side-by-side on a single Windows 11 machine, including clean uninstallation of existing instances and successful database restoration.

Scope

- Clean removal of existing SQL Server instances
- Fresh installation of SQL Server 2017 (default instance)
- Installation of SQL Server 2019 (named instance)
- Post-installation configuration following Microsoft best practices
- AdventureWorks2017 database restoration with troubleshooting

Target Audience

- Junior DBAs setting up lab environments
- Students in database administration training
- IT professionals preparing for SQL Server certifications
- Anyone needing step-by-step SQL Server installation guidance

Time Investment

- Total Duration: [X] hours
- Uninstallation: 23 minutes
- SQL 2017 Installation: [X] minutes
- SQL 2019 Installation: [X] minutes
- Configuration & Testing: [X] minutes

Prerequisites

Hardware Requirements

✅ CPU: Intel Core i5 8th Gen or higher
✅ RAM: 16GB minimum (12GB will be allocated to SQL Server)
✅ Storage: 500GB+ free space (SSD recommended)
✅ Display: 1366x768 minimum resolution

Best practice
Partition your C:\ into 4 extra drives:
- E:\DATA
- L:\LOG
- H:\BACKUP
- T:\TEMPDB

use whatever letters you like!

Software Requirements

✅ Windows 10 Pro or Windows 11 Pro
✅ .NET Framework 3.5 (install via Windows Features)
✅ .NET Framework 4.8 or higher
✅ Windows Updates: Current
✅ SQL Server 2017 Enterprise installation media
✅ SQL Server 2019 Enterprise installation media

Access Requirements

✅ Local Administrator rights
✅ Internet access (for downloading AdventureWorks backup) available

Part 1: Clean Uninstallation

Why Clean Uninstallation Matters

Leftover files and registry entries from previous SQL Server installations can cause:

- Installation failures
- Port conflicts
- Service startup issues
- Configuration problems
- Authentication errors

Understand
uninstalling and reinstalling
server helps master
the steps much faster than any
manual would :)

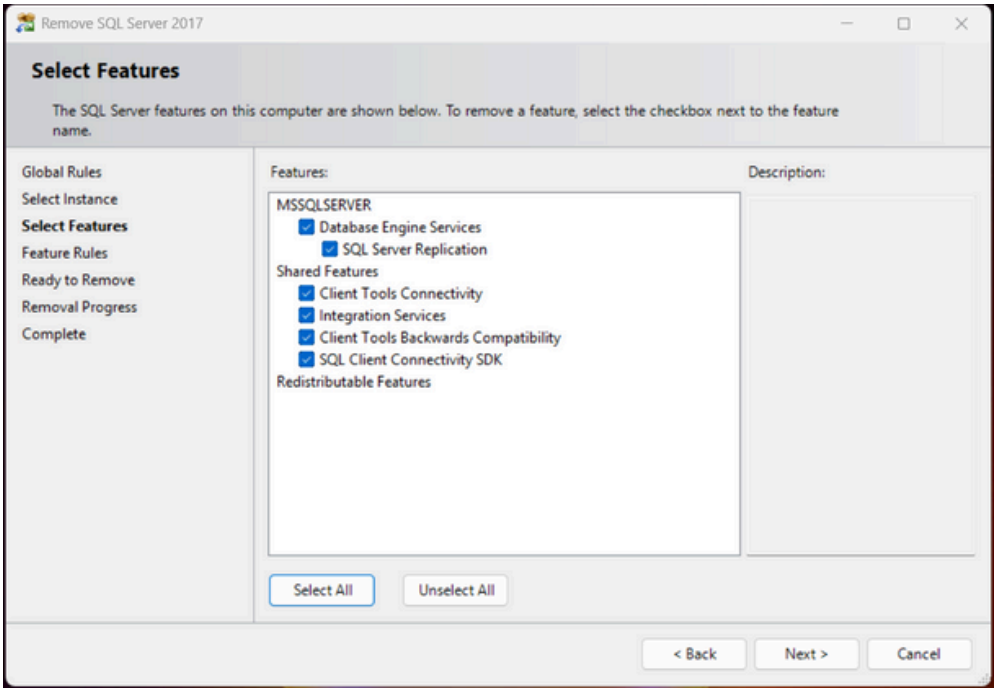
Step 1: Uninstall SQL Server 2017

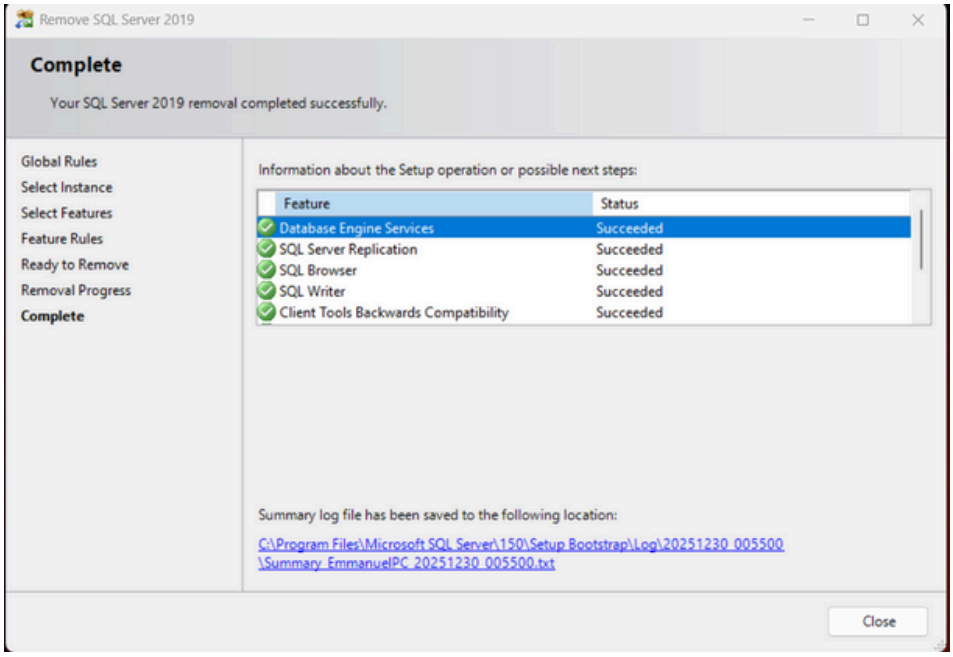
1.1 Open Programs and Features

Control Panel → Programs → Programs and Features
OR
Windows Key + X → Installed apps

1.2 Uninstall *Microsoft SQL Server 2017 (64-bit)*

- Click on *Remove* to uninstall
- *Select All* to remove features





1.3 Uninstall SQL Server 2019
Repeat the same process for SQL Server 2019 and confirm as seen above.

1.4 Verify Removal
After uninstallation, check these locations:

File System Check:
1. Open *File Explorer*
2. Navigate to: C:\Program Files\Microsoft SQL Server\
3. Verify folders are removed or minimal remnants
4. Check: C:\Program Files (x86)\Microsoft SQL Server\

Expected Result: Folders should be deleted or only contain shared components

1.5 System Restart
CRITICAL STEP - DO NOT SKIP
Restart your computer before proceeding with installation

Why This Matters:
- Releases file locks
- Clears memory
- Completes uninstallation processes
- Prevents "file in use" errors during installation

Time Required: 3-5 minutes

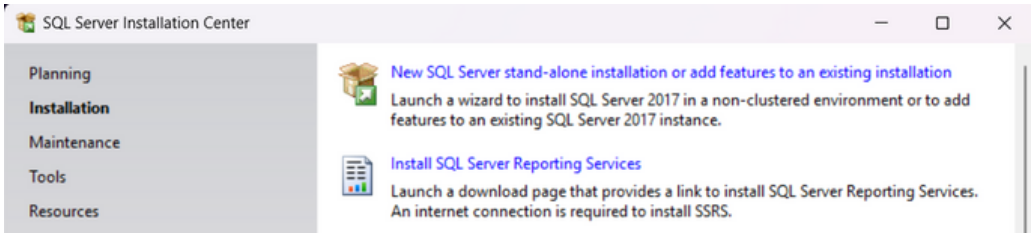
Part 2: SQL Server 2017 Installation

Step 2.1: Launch Installation Wizard

- Mount SQL Server 2017 ISO or navigate to installation media
- Run setup.exe as Administrator
- Right-click setup.exe → Run as administrator
- SQL Server Installation Center opens

Step 2.2: Choose Installation Type

- Select "Installation" from left menu
- Click "New SQL Server stand-alone installation or add features to an existing installation"



Screenshot Reference: [sql2017_installation_type.png]

Step 2.3: Product Key

- Option A: Enter product key
- Option B: Select "Evaluation" (180-day trial)
- My Selection: [Specify which you chose]
- Click Next

Step 2.4: License Terms

- ✅ Accept the license terms quickly!
- Click Next

Step 2.5: Microsoft Update

Recommendation: ❌ Don't use Microsoft Update to check for updates, just yet.

- Ensures that your current setup doesn't break up with the latest security patches
- Manual patching is best practice here!
- Click Next

Step 2.6: Install Setup Files

- Automatic installation of setup support files
- Wait for completion (~2-3 minutes)

Step 2.7: Install Rules Check

System checks for:
✅ Windows Firewall
✅ .NET Framework
✅ Disk space
✅ Administrator rights

Expected Result: All checks pass (green checkmarks)
⚠️ If any failures:

- Review error details
- Fix issues before proceeding
- Click "Re-run" after fixes
- Click Next

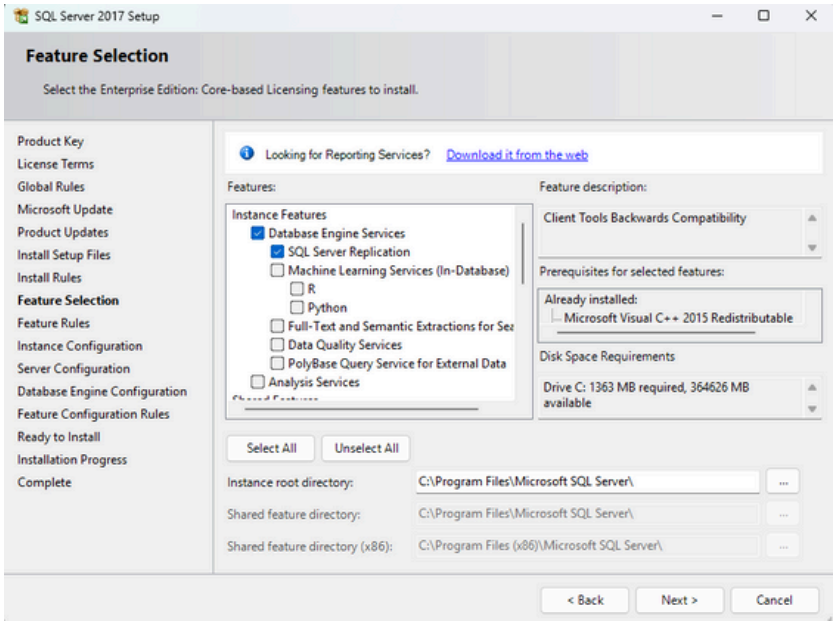
Step 2.8: Feature Selection (Typical)

Features to Select:
✅ Database Engine Services (REQUIRED)
✅ SQL Server Replication
✅ Client Tools Backwards Compatibility
✅ Client Tools Connectivity
✅ Integration Services

Instance Root Directory:
C:\Program Files\Microsoft SQL Server

Shared Feature Directory:

- C:\Program Files\Microsoft SQL Server\
- C:\Program Files (x86)\Microsoft SQL Server\



- Click *Next*

Step 2.9: Instance Configuration

Critical Decision: Default vs Named Instance

My Selection:

- Instance Type: Default instance
- Instance name: MSSQLSERVER
- Instance ID: MSSQLSERVER

Why Default Instance:

- Simplest to connect to (just use server name)
- Standard for primary SQL Server installation
- No port specification needed in connection strings

Instance root directory:

C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER

- Click Next

Step 2.10: Server Configuration

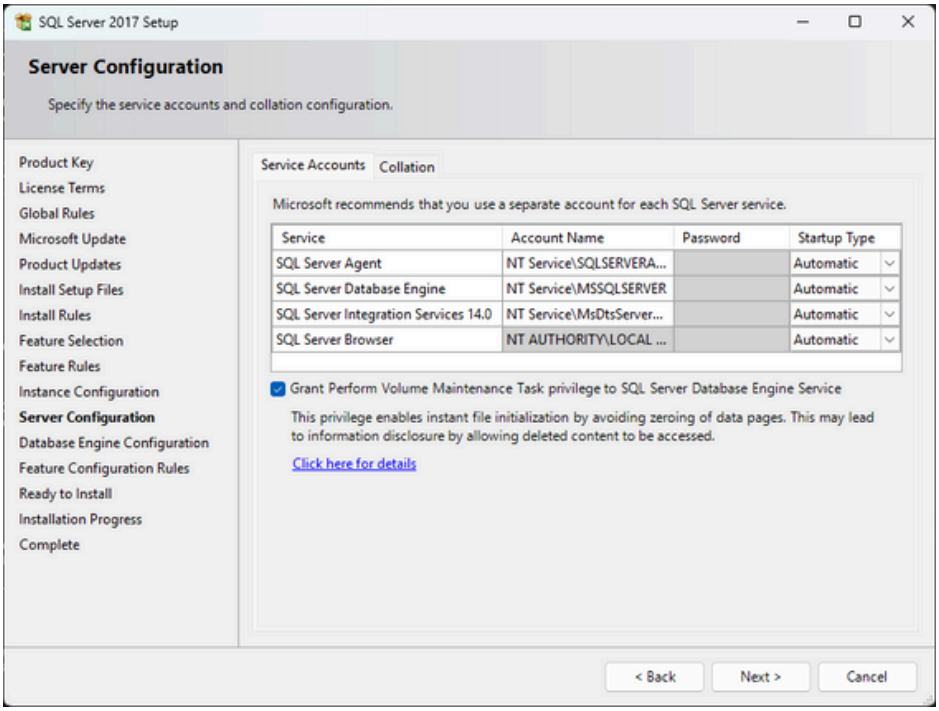
Service Accounts Configuration:

Why NT AUTHORITY\SYSTEM:

- Built-in high-privilege account
- No password management required
- Suitable for development/testing environments
- In production, use domain service accounts

Collation:

- Default: *SQL_Latin1_General_CP1_CI_AS*



- Click *Next*

Step 2.11: Database Engine Configuration

Authentication Mode

My Selection: ✅ Mixed Mode (SQL Server authentication and Windows authentication)

Why Mixed Mode:

- Allows both Windows and SQL logins
- More flexible for testing
- Can create SA account for emergency access

- SA Password: [Set strong password - DOCUMENT SECURELY]
- Minimum 8 characters
- Uppercase, lowercase, numbers, special characters
- Example format: SqlAdmin2024!

SQL Server Administrators:

✅ *Add Current User* ([Your Windows Account])
This gives you full administrative rights

Data Directories

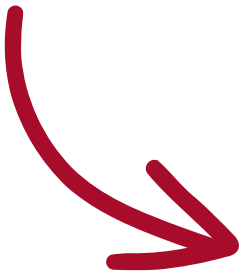
Recommended configuration:

Data root directory - *ALLOW DEFAULT AS IS*
C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL

User database directory
E:\DEFAULT\DATA

User database log directory
L:\DEFAULT\LOG

Backup directory
H:\DEFAULT\BACKUP



Best practice
time to use those drives.. you would've created folders and subfolders before this step but you could still do so now :)

Best Practice Note:
In production environments:

- Data files: Separate fast drive (SSD)
- Log files: Separate drive from data
- Backups: Network location or separate drive

TempDB Configuration

- Default Settings (Acceptable for Development):
- Click Next

Step 2.12: Ready to Install

Final Review Screen:

- Review all selections
- Note installation path
- Verify feature list
- Check instance name

Configuration File:
C:\Program Files\Microsoft SQL Server\140\Setup Bootstrap\Log\[Timestamp]\ConfigurationFile.ini

- Useful for automated installations
- Can be saved for documentation
- Click *Install*

Step 2.13: Installation Progress

What Happens:

- Setup support files installation
- Feature installation (Database Engine, Replication, etc.)
- Component configuration
- Service startup

Time Required: 15-30 minutes (depending on hardware)

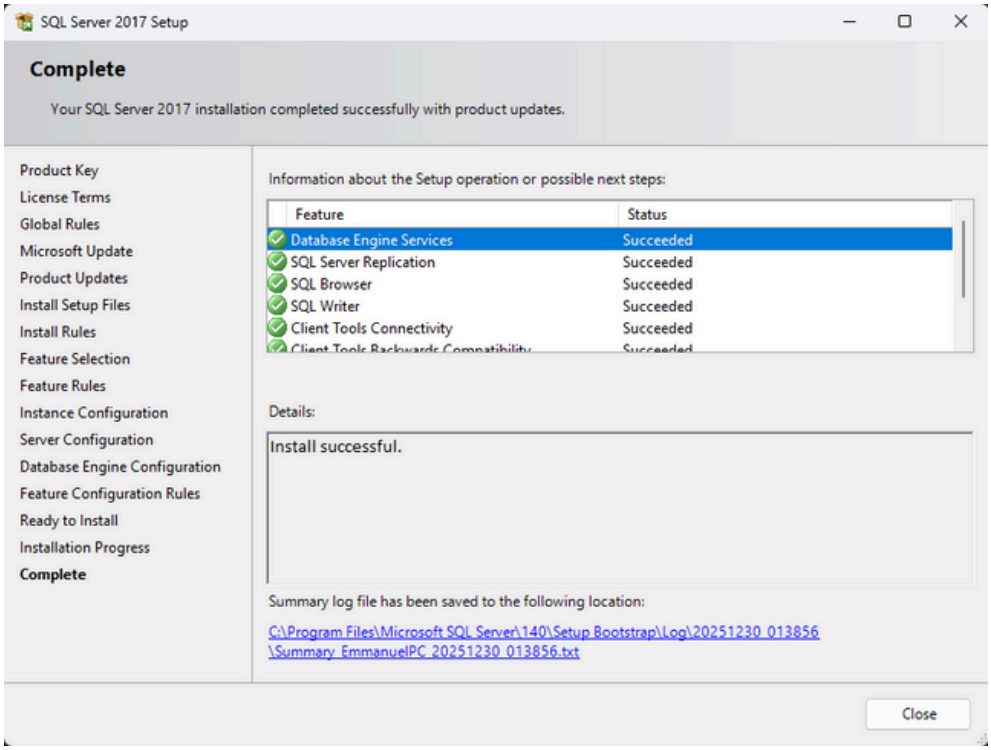
Monitor Progress:

- Green checkmarks = completed successfully
- Red X = failure (note error for troubleshooting)

Step 2.14: Installation Complete

Success Indicators:
✅ All features show "Success"
✅ SQL Server Database Engine: Success
✅ SQL Server Replication: Success
✅ Management Tools: Success

Summary Log Location:
C:\Program Files\Microsoft SQL Server\140\Setup Bootstrap\Log\Summary.txt



⚠️ If Any Failures:

- Click on failed component
- Review error details
- Check detailed log in Setup Bootstrap\Log folder
- See *Troubleshooting Guide* below
- Click *Close*

😎 Kudos! You made it up to this point without a cup of coffee!

If you documented your own process like I did, you might as well stop right here and forget about the rest of this playbook. You have some confidence now to go ahead and install a named instance side-by-side to this current default one.

i challenge you to go ahead and do it. As expected, you'd probably miss out a step or two like I did on my first run. That's exactly how you'd learn to properly install and configure SQL Server.

Install, Uninstall, Reinstall, Repeat.

Fun fact: I uninstalled and reinstalled both servers while editing this guide!

However if you want to continue with me after a glass of your favorite chilled drink, that's all up to you. If not, skip *Part 3* and move on directly to post-installation best practices in *Part 4!*

Part 3: SQL Server 2019 Installation

Why Install SQL Server 2019 Separately

Use Cases for Multiple Versions:

- Testing application compatibility
- Learning new features
- Simulating production/dev environments
- Migration preparation

Step 3.1: Launch SQL Server 2019 Setup

Same process as 2017:

- Run setup.exe as Administrator
- Select "Installation" → "New SQL Server stand-alone installation"

Step 3.2: Product Key & License

[Follow same steps as SQL 2017]

Step 3.3: Feature Selection

Same features as 2017 OR adjust based on needs

Step 3.4: Instance Configuration - CRITICAL DIFFERENCE

⚠️ IMPORTANT: Must use Named Instance (cannot have two default instances)

My Configuration:

- Instance Type: Named instance
- Instance name: SQL2019
- Instance ID: SQL2019

Connection String Difference:

- SQL 2017 (default): localhost or . or (local)
- SQL 2019 (named): localhost\SQL2019 or .\SQL2019

Instance root directory:

C:\Program Files\Microsoft SQL Server\MSSQL15.SQL2019

Part 4: Post-Installation Configuration

Step 4.1: Verify Both Instances Running

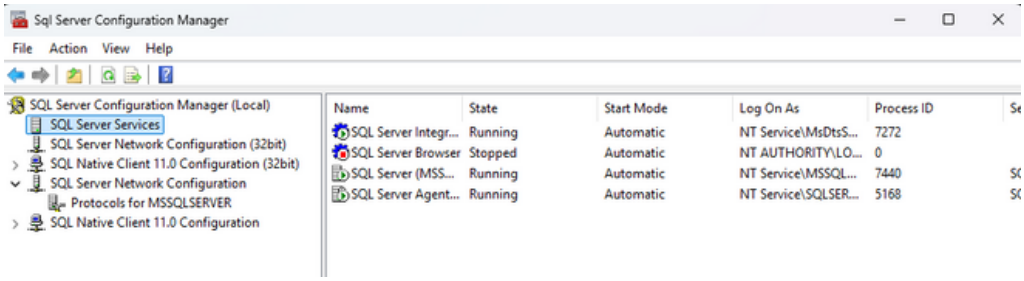
Method 1: Services (services.msc)

1. Windows Key + R
2. Type: services.msc
3. Press Enter

4. Look for:
   - SQL Server (MSSQLSERVER) - Status: Running
   - SQL Server (SQL2019) - Status: Running
   - SQL Server Agent (MSSQLSERVER) - Status: Running
   - SQL Server Agent (SQL2019) - Status: Running

Method 2: SQL Server Configuration Manager

1. Open SQL Server Configuration Manager
2. SQL Server Services
3. Verify both instances show "Running"



Only the default instance shown here

Step 4.2: Connect via SSMS

- Install/Verify SQL Server Management Studio 20 (or whatever preferred version)
- Open SSMS
- **Connect to SQL Server 2017:**
- Server name: localhost or . or (local)
- Authentication: Windows Authentication
- Click *Connect*

- **Connect to SQL Server 2019:**
- File → Connect Object Explorer
- Server name: localhost\SQL2019
- Authentication: Windows Authentication
- Click *Connect*

Expected Result: Both servers appear in Object Explorer

Step 4.3: Verify SQL Server Versions

For SQL Server 2017:

```
-- Connect to default instance
SELECT @@VERSION AS SQLServerVersion;
SELECT SERVERPROPERTY('ProductVersion') AS Version,
       SERVERPROPERTY('ProductLevel') AS ServicePack,
       SERVERPROPERTY('Edition') AS Edition;
```

Expected Output:
Microsoft SQL Server 2017 (RTM-CU31) - 14.0.3456.2 (X64)
Enterprise Edition (64-bit)

For SQL Server 2019:

```
-- Connect to .\SQL2019
SELECT @@VERSION AS SQLServerVersion;
```

Expected Output:
Microsoft SQL Server 2019 (RTM-CU18) - 15.0.4261.1 (X64)
Enterprise Edition (64-bit)

Step 4.4: Configure Maximum Server Memory

Why This Matters:

- SQL Server will consume ALL available RAM by default
- Starves Windows OS
- Can cause system instability

Formula for 16GB System:

- Total RAM: 16GB (16,384 MB)
- Reserve for OS: 4GB (4,096 MB)
- Max for SQL Server: 12GB (12,288 MB)

If running TWO instances: 6GB each (6,144 MB)
Configuration Script (run on BOTH instances):

For SQL 2017:

```
-- Connect to default instance
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;

EXEC sp_configure 'max server memory', 6144; -- 6GB
RECONFIGURE;
```

```
-- Verify
EXEC sp_configure 'max server memory';
```

For SQL 2019:

```
-- Connect to .\SQL2019
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;

EXEC sp_configure 'max server memory', 6144; -- 6GB
RECONFIGURE;

-- Verify
EXEC sp_configure 'max server memory';
```

Step 4.5: Enable TCP/IP Protocol

Why Enable TCP/IP:

- Required for remote connections
- Required for named instances
- Industry standard
- More flexible than Named Pipes

Steps:

- Open SQL Server Configuration Manager
- SQL Server Network Configuration
- Protocols for MSSQLSERVER
- Right-click TCP/IP → *Enable*
- Protocols for SQL2019
- Right-click TCP/IP → *Enable*
- Restart both SQL Server services

Verify:

```
-- Check listening protocols
EXEC xp_readerrorlog 0, 1, N'Server is listening on';
```

Step 4.6: Configure Firewall (Optional for Remote Access)

If you need remote connections:
Windows Firewall → Advanced Settings → Inbound Rules
Add Rule:
- Port: 1433 (SQL 2017 default instance)
- Port: [dynamic for SQL2019 - check Configuration Manager]
- Protocol: TCP
- Allow the connection

Troubleshooting Guide

Issue 1: Installation Fails at Database Engine Services

Error Message:
The following error has occurred:
SQL Server Database Engine Services Instance Features installation failed.

Root Causes:

- Missing .NET Framework
- Insufficient permissions
- Leftover registry entries
- Antivirus interference

Solution A: Install .NET Framework 3.5

1. Control Panel → Programs → Turn Windows features on or off
2. Check ".NET Framework 3.5 (includes .NET 2.0 and 3.0)"
3. Click OK, restart
4. Retry installation

Solution B: Check Setup Logs

Location: *C:\Program Files\Microsoft SQL Server\140\Setup Bootstrap\Log*
File: Summary.txt

Look for:
- "Error" entries
- Exit code (non-zero = failure)
- Specific component that failed

Solution C: Clean Previous Installation

1. Use Microsoft's SQL Server Uninstall utility
2. Manually delete registry keys (Advanced users only)
3. Remove all SQL Server folders
4. Restart computer
5. Retry installation

Issue 2: Cannot Connect to SQL Server

Error Message:
A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible.

Step 1: Verify Service is Running

1. Open services.msc
2. Find "SQL Server (MSSQLSERVER)"
3. Status should be "Running"
4. If stopped, right-click → Start

Step 2: Check SQL Server Configuration Manager

1. Open SQL Server Configuration Manager
2. SQL Server Services
3. SQL Server (MSSQLSERVER) - should be green/running
4. If red, check error logs

Step 3: Verify TCP/IP is Enabled

1. SQL Server Configuration Manager
2. SQL Server Network Configuration
3. Protocols for MSSQLSERVER
4. TCP/IP should show "Enabled"
5. If disabled, right-click → Enable → Restart SQL Service

Step 4: Check SQL Server Error Log

EXEC sp_readerrorlog;

Look for:

- "Server is listening on"
- Any error messages
- Port numbers

Step 5: Test Connection String

Try these variations:
- localhost
- .
- (local)
- 127.0.0.1
- [Your Computer Name]
- [Your Computer Name]\MSSQLSERVER

Issue 3: Services Won't Start

Symptoms:

- SQL Server service shows "Stopped"
- Cannot start service manually
- Error 1067 or similar

Check 1: Service Account Permissions

- SQL Server Configuration Manager →
- SQL Server Services →
- Right-click service → Properties →
- Log On tab → Verify NT AUTHORITY\SYSTEM

Check 2: Port Conflicts

-- *Check if port 1433 is in use*
netstat -ano | findstr :1433

-- *Change port if needed in SQL Server Configuration Manager*

Check 3: Corruption

1. Check error logs:
   C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Log\ERRORLOG

2. Look for specific error messages
3. May need to rebuild master database (advanced)

Issue 4: Cannot Connect to Named Instance

Symptoms:

- SQL Server 2019 connection fails
- "Server not found" error
- Named instance not accessible

Solution 1: Verify Instance Name

- Correct format: localhost\SQL2019
- NOT: localhost:SQL2019
- NOT: localhost/SQL2019

Solution 2: Enable SQL Server Browser

1. services.msc
2. Find "SQL Server Browser"
3. Right-click → Properties
4. Startup type: Automatic
5. Start the service

Solution 3: Enable TCP/IP

SQL Server Configuration Manager → SQL Server Network Configuration → Protocols for SQL2019 → Right-click TCP/IP → Enable →
Restart SQL Server (SQL2019) service

Screenshot Reference: [browser_service_enabled.png]

Issue 5: Mixed Mode Authentication Not Working

Symptoms:

- Cannot log in with SA account
- "Login failed for user 'sa'"
- Windows Authentication works but SQL doesn't

Solution:

```
-- Verify authentication mode (connect with Windows Auth first)
SELECT SERVERPROPERTY('IsIntegratedSecurityOnly') AS AuthMode;
-- 0 = Mixed Mode, 1 = Windows Only

-- If result is 1, change to Mixed Mode:
```

Manual Method:

1. SSMS → Connect with Windows Authentication
2. Right-click server → *Properties*
3. Security page
4. Server authentication:
   ✅ SQL Server and Windows Authentication mode
5. Click *OK*
6. Restart SQL Server service

Enable SA Account:

```
-- SA is disabled by default in newer versions
USE master;
GO
ALTER LOGIN sa ENABLE;
GO
ALTER LOGIN sa WITH PASSWORD = 'YourStrongPassword123!';
GO
```

Issue 6: Installation Fails During Feature Installation

Symptoms:

- Red X on Database Engine Services
- Setup fails midway
- Error logs show component failures

Common Causes & Solutions:

Cause 1: .NET Framework Missing

Control Panel → Programs → Turn Windows features on/off →
✅ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
Restart computer

Cause 2: Previous Installation Remnants

1. Use Microsoft's SQL Server Uninstall Tool:
   - Download from Microsoft
   - Run to completely clean system
2. Manually delete folders:
   - C:\Program Files\Microsoft SQL Server\
   - C:\Program Files (x86)\Microsoft SQL Server\
3. Restart and retry

Cause 3: Insufficient Disk Space

- Free up at least 6GB on system drive
- Move page file if needed
-

Issue 7: Memory Configuration Not Persisting

Symptoms:

- Max memory settings revert after restart
- SQL Server consumes all RAM
- System becomes slow

Solution:

```
-- Verify current settings
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;
EXEC sp_configure 'max server memory';

-- If not correct, reset and restart service
EXEC sp_configure 'max server memory', 6144;
RECONFIGURE WITH OVERRIDE;

-- Restart SQL Server service in services.msc
-- Then verify again
```

Issue 8: High Memory Usage Despite Configuration

Symptoms:

- Task Manager shows SQL Server using more than configured
- System still sluggish
- Both instances consuming excessive memory

Understanding Memory:

SQL Server memory settings control buffer pool only, not:

- Thread memory
- CLR allocations
- Linked server queries
- Memory-intensive queries

Solutions:

For Development Environment:

```
-- On BOTH instances, reduce further:
EXEC sp_configure 'max server memory', 4096; -- 4GB each
RECONFIGURE;
```

Monitor Memory Usage:

```
-- Check actual memory usage
SELECT
    physical_memory_in_use_kb/1024 AS [Physical Memory Used MB],
    locked_page_allocations_kb/1024 AS [Locked Page Allocations
MB],
    total_virtual_address_space_kb/1024 AS [Total Virtual Address
Space MB],
    virtual_address_space_committed_kb/1024 AS [Virtual Address
Space Committed MB]
FROM sys.dm_os_process_memory;
```

☑ Verification Checklist

Post-Installation Verification

Service Status:

☐ SQL Server (MSSQLSERVER) - Running, Automatic
☐ SQL Server Agent (MSSQLSERVER) - Running, Automatic
☐ SQL Server (SQL2019) - Running, Automatic
☐ SQL Server Agent (SQL2019) - Running, Automatic
☐ SQL Server Browser - Running, Automatic (if needed for remote access)

Connection Tests:

☐ Can connect to localhost (SQL 2017) via SSMS
☐ Can connect to localhost\SQL2019 via SSMS
☐ Windows Authentication works on both instances
☐ SQL Authentication works on both instances (if configured)
☐ Can see system databases on both instances

Version Verification:

```
-- Run on SQL 2017
SELECT @@VERSION;

-- Should show: SQL Server 2017 (14.0.x)

-- Run on SQL 2019
SELECT @@VERSION;

-- Should show: SQL Server 2019 (15.0.x)
```

Memory Configuration:

```
-- Run on BOTH instances
EXEC sp_configure 'max server memory';

-- Should show: 6144 MB (or your configured value)
```

Network Configuration:

☐ TCP/IP enabled on both instances
☐ Named Pipes enabled (optional)
☐ Firewall rules configured (if needed for remote access)

Lessons Learned

Key Takeaways

1. Clean Uninstallation is Critical

- Don't skip the uninstallation steps
- Always restart after uninstalling
- Check for leftover files manually
- Previous installations cause 70% of installation failures

2. Instance Naming Matters

- Default instance: Simpler connections
- Named instance: Required for multiple versions
- Document your naming convention
- Connection string format is critical: server\instance

3. Memory Configuration is Not Optional

- Default settings will consume all RAM
- Configure BEFORE adding workloads
- Monitor after configuration
- Adjust based on actual usage patterns

4. File Path Attention

- Most restore failures are path-related
- Always use "Relocate all files to folder" option
- Verify paths before clicking OK
- SQL Server needs full permissions to target folders

5. Authentication Setup

- Mixed Mode provides flexibility
- Document SA password securely
- Windows Auth is more secure for production
- Enable SA account explicitly if needed

Common Mistakes to Avoid

❌ **Mistake 1:** Not restarting after uninstallation
✅ **Fix:** Always restart - saves hours of troubleshooting

❌ **Mistake 2:** Using same instance name for both versions
✅ **Fix:** Use descriptive named instances (SQL2019, SQL2022, etc.)

❌ **Mistake 3:** Skipping memory configuration
✅ **Fix:** Configure immediately after installation

❌ **Mistake 4:** Wrong backup file location
✅ **Fix:** Use SQL Server's Backup directory

❌ **Mistake 5:** Not documenting passwords
✅ **Fix:** Use password manager or secure documentation

🕐 Time-Saving Tips

Before Starting:

- Download all installation media first
- Have SQL Server Uninstall Tool ready
- Disable antivirus temporarily
- Close unnecessary applications
- Document current configuration

During Installation:

- Take screenshots at each step
- Save configuration files
- Note any warnings or errors
- Don't rush through wizards

After Installation:

- Test immediately
- Configure before adding databases
- Create restore scripts
- Document connection strings
- Save this guide with your notes

Resources for Further Learning

Microsoft Documentation:
SQL Server Installation Guide: https://docs.microsoft.com/sql/database-engine/install-windows/

Best Practices: https://docs.microsoft.com/sql/sql-server/install/

AdventureWorks Samples: https://github.com/Microsoft/sql-server-samples

Community Resources:
SQL Server Central: https://www.sqlservercentral.com
DBA Stack Exchange: https://dba.stackexchange.com
Brent Ozar's Blog: https://www.brentozar.com/blog/

My Environment Specifics:
Hardware: Dell Latitude 7490
CPU: Intel Core i5-8350U (8th Gen)
RAM: 16GB DDR4
Storage: 512GB NVMe SSD
OS: Windows 11 Pro (Version 22H2)

SQL 2017: Enterprise Evaluation Edition (Build 14.0.3456.2)
SQL 2019: Enterprise Evaluation Edition (Build 15.0.4261.1)
SSMS: Version 20.x

Final Notes

Document Your Environment
Your Installation Date: _____

Your Instance Names:
SQL Server 2017: _____
SQL Server 2019: _____

Your Connection Strings:
SQL 2017: _____
SQL 2019: _____

Your SA Passwords: (Store securely!)
SQL 2017 SA: [Use password manager]
SQL 2019 SA: [Use password manager]

Issues Encountered:
1. _____
2. _____
3. _____

Total Time Spent:
Uninstallation: _____ minutes
SQL 2017 Install: _____ minutes
SQL 2019 Install: _____ minutes
Configuration: _____ minutes
Troubleshooting: _____ minutes

TOTAL: _____ hours

Success Criteria
You've successfully completed this installation when:

✅ Both SQL Server instances are running
✅ You can connect to both via SSMS
✅ Memory is configured properly (6GB each)
✅ Services start automatically after reboot
✅ You've documented all configurations
✅ You understand how to troubleshoot common issues

Appendix A: Quick Reference Commands

Check SQL Server Version

```
SELECT @@VERSION;
SELECT SERVERPROPERTY('ProductVersion') AS Version;
```

Check Running Services (PowerShell)

```
Get-Service | Where-Object {$_.Name -like "*SQL*"} | Select Name,
Status, StartType
```

Check Memory Configuration

```
EXEC sp_configure 'max server memory';
```

List All Databases

```
SELECT name, database_id, create_date
FROM sys.databases
ORDER BY name;
```

Check Database Size

```
EXEC sp_spaceused;
```

Backup Database (T-SQL)

```
BACKUP DATABASE AdventureWorks2017
TO DISK = 'C:\Backup\AdventureWorks2017.bak'
WITH FORMAT, INIT, COMPRESSION;
```

Restore Database (T-SQL)

```
RESTORE DATABASE AdventureWorks2017
FROM DISK = 'C:\Backup\AdventureWorks2017.bak'
WITH MOVE 'AdventureWorks2017'
     TO 'C:\Program Files\Microsoft SQL
Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2017.mdf',
MOVE 'AdventureWorks2017_log'
     TO 'C:\Program Files\Microsoft SQL
Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\AdventureWorks2017_log.ldf',
REPLACE;
```

Appendix B: Error Code Reference

Error Code	Description	Typical Solution
5	Access denied	Fix file/folder permissions
1807	Database already exists	Drop existing or use WITH REPLACE
3156	Incorrect file path	Verify paths on Files page
18456	Login failed	Check authentication mode
1067	Service won't start	Check error log, verify configuration
17182	TDS protocol error	Enable TCP/IP protocol
26	Error locating server	Check instance name, enable browser

END OF INSTALLATION GUIDE

Document Status: Complete
Last Updated: December 28, 2025

Validated Environment: Windows 11 Pro, SQL Server 2017/2019 Enterprise

This guide is based on real-world installation experience and includes solutions to actual problems encountered. Your specific environment may vary.