

Microsoft SQL Server 2022

CU19 Deployment Guide

Version 16.0.4195.2 (KB5054531)

Ubuntu 24.04 LTS (Noble Numbat)

Developer Edition (64-bit)

Complete Installation, Troubleshooting & Deployment Guide

With Ubuntu 24.04 Compatibility Workarounds

Document Version: 1.0
Last Updated: November 5, 2025
Tested and Verified

Contents

1 Executive Summary	2
1.1 Key Challenges Addressed	2
1.2 Target Configuration	2
2 Prerequisites	3
2.1 System Requirements	3
2.1.1 Minimum Requirements	3
2.1.2 Recommended Requirements	3
2.2 Administrative Requirements	3
3 Installation Procedure	4
3.1 Step 1: System Verification	4
3.2 Step 2: Import Microsoft GPG Key	4
3.3 Step 3: Add SQL Server Repository	4
3.4 Step 4: Install SQL Server Package	4
3.5 Step 5: Fix OpenLDAP Dependency (CRITICAL)	5
3.5.1 5.1 Download OpenLDAP 2.5 from Ubuntu 22.04	5
3.5.2 5.2 Extract Libraries	5
3.5.3 5.3 Install OpenLDAP 2.5 Libraries	5
3.5.4 5.4 Verify Libraries	5
3.5.5 5.5 Verify No Missing Dependencies	5
3.6 Step 6: Configure SQL Server	6
3.7 Step 7: Install SQL Server Command-Line Tools	7
3.7.1 7.1 Add Microsoft Production Repository	7
3.7.2 7.2 Update and Install Tools	7
3.7.3 7.3 Add Tools to PATH (Optional)	7
3.8 Step 8: Verify Installation	7
4 Troubleshooting Guide	8
4.1 Issue 1: Service Won't Start	8
4.1.1 Check Logs	8
4.1.2 Check Dependencies	8
4.1.3 Common Causes	8
4.2 Issue 2: OpenLDAP Library Errors	8
4.3 Issue 3: Cannot Connect with SA User	8
4.3.1 Reset SA Password	8
4.3.2 Check if SQL Server is Listening	8
4.4 Issue 4: High Memory Usage	8
4.4.1 Check Current Memory Configuration	8
4.4.2 Set Memory Limit	9
5 Session History	10
5.1 Timeline of Events	10
5.2 Critical Issues and Resolutions	10
6 Post-Installation Configuration	11
6.1 Security Hardening	11
6.1.1 Change SA Password	11
6.1.2 Configure Firewall	11

6.1.3	Enable Encryption	11
6.2	Performance Tuning	11
6.2.1	Configure Memory	11
6.2.2	Enable SQL Server Agent	11
7	Service Management	12
7.1	Common Commands	12
7.2	Important File Locations	12
8	Quick Reference	13
8.1	Connection String Format	13
8.2	Common sqlcmd Commands	13
8.3	Password Change	13
9	Conclusion	13
9.1	Key Takeaways	13
9.2	Support and Updates	13
9.3	Additional Resources	14

1 Executive Summary

This document provides a comprehensive guide for installing Microsoft SQL Server 2022 (RTM-CU19) Developer Edition on Ubuntu 24.04 LTS. The guide includes:

- Step-by-step installation procedures
- Critical compatibility workarounds for Ubuntu 24.04
- Detailed troubleshooting for common issues
- Complete session history with all attempted solutions
- Post-installation configuration and optimization

1.1 Key Challenges Addressed

1. **Repository Availability:** Ubuntu 24.04 repository not available from Microsoft
2. **OpenLDAP Compatibility:** Ubuntu 24.04 ships with OpenLDAP 2.6, SQL Server requires 2.5
3. **GPG Signature Issues:** Proper repository configuration with signed-by option
4. **Authentication Issues:** SA password configuration challenges

1.2 Target Configuration

Component	Version/Details
Operating System	Ubuntu 24.04.3 LTS (Noble Numbat)
SQL Server Version	16.0.4195.2 (RTM-CU19)
Knowledge Base	KB5054531
Edition	Developer Edition (64-bit)
Package Version	mssql-server=16.0.4195.2-4
Architecture	x86_64

2 Prerequisites

2.1 System Requirements

2.1.1 Minimum Requirements

- **Processor:** x64-compatible 1.4 GHz
- **Memory:** 2 GB RAM
- **Disk Space:** 6 GB available
- **File System:** XFS or EXT4
- **Network:** TCP port 1433 available

2.1.2 Recommended Requirements

- **Processor:** x64-compatible 2.0 GHz or faster, 4+ cores
- **Memory:** 8 GB RAM or more
- **Disk Space:** 20 GB or more (depends on database size)
- **File System:** XFS (best performance)
- **Disk:** SSD for database files

2.2 Administrative Requirements

- Root or sudo access
- Internet connection
- Ubuntu 24.04 LTS (fresh installation recommended)

3 Installation Procedure

3.1 Step 1: System Verification

Verify your Ubuntu version before proceeding:

```
1 cat /etc/os-release
```

Expected Output:

```
PRETTY_NAME="Ubuntu 24.04.3 LTS"
VERSION_ID="24.04"
VERSION_CODENAME=noble
```

3.2 Step 2: Import Microsoft GPG Key

Import the Microsoft GPG key for package verification:

```
1 curl -fsSL https://packages.microsoft.com/keys/microsoft.asc | \
2   sudo gpg --dearmor -o /usr/share/keyrings/microsoft-prod.gpg
```

Important

No output indicates success. This key is required for verifying package signatures.

3.3 Step 3: Add SQL Server Repository

Add the SQL Server 2022 repository. Note that we use Ubuntu 22.04 repository as 24.04 is not yet available:

```
1 echo "deb [arch=amd64,arm64,armhf signed-by=/usr/share/keyrings/
      microsoft-prod.gpg] https://packages.microsoft.com/ubuntu/22.04/
      mssql-server-2022 jammy main" | \
2   sudo tee /etc/apt/sources.list.d/mssql-server-2022.list
```

Ubuntu 24.04 Compatibility

Microsoft doesn't have a dedicated SQL Server repository for Ubuntu 24.04 yet. Using the Ubuntu 22.04 (jammy) repository is the recommended workaround and works correctly.

Update package lists:

```
1 sudo apt-get update
```

3.4 Step 4: Install SQL Server Package

Install SQL Server 2022 CU19:

```
1 sudo DEBIAN_FRONTEND=noninteractive apt-get install -y mssql-server
      =16.0.4195.2-4
```

Expected Output:

```
Setting up mssql-server (16.0.4195.2-4) ...
+-----+
Please run 'sudo /opt/mssql/bin/mssql-conf setup'
to complete the setup of Microsoft SQL Server
+-----+
```

3.5 Step 5: Fix OpenLDAP Dependency (CRITICAL)

This step is CRITICAL for Ubuntu 24.04!

Ubuntu 24.04 ships with OpenLDAP 2.6, but SQL Server 2022 requires OpenLDAP 2.5. Follow these steps:

3.5.1 5.1 Download OpenLDAP 2.5 from Ubuntu 22.04

```
1 cd /tmp
2 wget http://security.ubuntu.com/ubuntu/pool/main/o/openldap/libldap
      -2.5-0_2.5.16+dfsg-0ubuntu0.22.04.2_amd64.deb
```

3.5.2 5.2 Extract Libraries

```
1 dpkg -x libldap-2.5-0_2.5.16+dfsg-0ubuntu0.22.04.2_amd64.deb /tmp/
      libldap-extract
```

3.5.3 5.3 Install OpenLDAP 2.5 Libraries

```
1 sudo cp /tmp/libldap-extract/usr/lib/x86_64-linux-gnu/liblber-2.5.so
      .0* /lib/x86_64-linux-gnu/
2 sudo cp /tmp/libldap-extract/usr/lib/x86_64-linux-gnu/libldap-2.5.so
      .0* /lib/x86_64-linux-gnu/
3 sudo ldconfig
```

3.5.4 5.4 Verify Libraries

```
1 ls -la /lib/x86_64-linux-gnu/liblber-2.5* /lib/x86_64-linux-gnu/
      libldap-2.5*
```

Expected Output:

```
-rw-r--r-- 1 root root 63992 ... /lib/x86_64-linux-gnu/liblber-2.5.
      so.0
-rw-r--r-- 1 root root 63992 ... /lib/x86_64-linux-gnu/liblber-2.5.
      so.0.1.11
-rw-r--r-- 1 root root 376512 ... /lib/x86_64-linux-gnu/libldap-2.5.
      so.0
-rw-r--r-- 1 root root 376512 ... /lib/x86_64-linux-gnu/libldap-2.5.
      so.0.1.11
```

3.5.5 5.5 Verify No Missing Dependencies

```
1 ldd /opt/mssql/bin/sqlservr | grep -i "not found"
```

Expected: No output (all dependencies satisfied)

3.6 Step 6: Configure SQL Server

Run the initial SQL Server setup:

```
1 sudo MSSQL_SA_PASSWORD='YourStrong!Password' \
2     MSSQL_PID='Developer' \
3     /opt/mssql/bin/mssql-conf -n setup accept-eula
```

Password Requirements

Replace `YourStrong!Password` with your own strong password.

Password Requirements:

- At least 8 characters
- Contains uppercase letters
- Contains lowercase letters
- Contains digits
- Contains special characters

Start and enable the SQL Server service:

```
1 sudo systemctl start mssql-server
2 sudo systemctl enable mssql-server
```

Verify service status:

```
1 sudo systemctl status mssql-server
```

Expected Output:

```
* mssql-server.service - Microsoft SQL Server Database Engine
  Loaded: loaded
  Active: active (running)
```

3.7 Step 7: Install SQL Server Command-Line Tools

3.7.1 7.1 Add Microsoft Production Repository

```
1 echo "deb [arch=amd64,arm64,armhf signed-by=/usr/share/keyrings/
      microsoft-prod.gpg] https://packages.microsoft.com/ubuntu/22.04/
      prod jammy main" | \
2 sudo tee /etc/apt/sources.list.d/mssql-prod.list
```

3.7.2 7.2 Update and Install Tools

```
1 sudo apt-get update
2 sudo ACCEPT_EULA=Y apt-get install -y mssql-tools18 unixodbc-dev
```

3.7.3 7.3 Add Tools to PATH (Optional)

```
1 echo 'export PATH="$PATH:/opt/mssql-tools18/bin"' >> ~/.bashrc
2 source ~/.bashrc
```

3.8 Step 8: Verify Installation

Test the connection:

```
1 /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -P 'YourStrong!
  Password' -C -Q "SELECT @@VERSION" -W
```

Expected Output:

```
Microsoft SQL Server 2022 (RTM-CU19) (KB5054531) - 16.0.4195.2 (X64)
Apr 18 2025 13:42:14
Copyright (C) 2022 Microsoft Corporation
Developer Edition (64-bit) on Linux (Ubuntu 24.04.3 LTS) <X64>

(1 rows affected)
```

Success!

If you see the above output, SQL Server 2022 CU19 is successfully installed and running!

4 Troubleshooting Guide

4.1 Issue 1: Service Won't Start

4.1.1 Check Logs

```
1 sudo journalctl -u mssql-server -n 50
2 sudo cat /var/opt/mssql/log/errorlog | tail -50
```

4.1.2 Check Dependencies

```
1 ldd /opt/mssql/bin/sqlservr | grep -i "not found"
```

4.1.3 Common Causes

- Missing OpenLDAP 2.5 libraries (see Step 5)
- Insufficient memory (minimum 2 GB RAM required)
- Port 1433 already in use

4.2 Issue 2: OpenLDAP Library Errors

If you see errors like:

```
liblber-2.5.so.0: cannot open shared object file
version 'OPENLDAP_2.5' not found
```

Solution: Re-verify libraries are installed correctly:

```
1 ls -la /lib/x86_64-linux-gnu/liblber-2.5* /lib/x86_64-linux-gnu/
      libldap-2.5*
2 strings /lib/x86_64-linux-gnu/liblber-2.5.so.0 | grep OPENLDAP
```

Expected output should include: OPENLDAP_2.5_0

If missing, repeat Step 5.

4.3 Issue 3: Cannot Connect with SA User

4.3.1 Reset SA Password

```
1 sudo systemctl stop mssql-server
2 sudo MSSQL_SA_PASSWORD='NewStr0ng!Password' /opt/mssql/bin/mssql-
      conf set-sa-password
3 sudo systemctl start mssql-server
```

4.3.2 Check if SQL Server is Listening

```
1 sudo ss -tlnp | grep 1433
```

4.4 Issue 4: High Memory Usage

4.4.1 Check Current Memory Configuration

```
1 /opt/mssql-tools18/bin/sqlcmd -S localhost -U SA -C -Q "EXEC
      sp_configure 'max server memory'" -W
```

4.4.2 Set Memory Limit

```
1 sudo /opt/mssql/bin/mssql-conf set memory.memorylimitmb 2048
2 sudo systemctl restart mssql-server
```

5 Session History

This section documents the complete installation session, including all challenges encountered and their resolutions.

5.1 Timeline of Events

1. **System Verification:** Confirmed Ubuntu 24.04.3 LTS
2. **PGP Key Import:** Successfully imported Microsoft GPG key
3. **Repository Addition (Attempt 1):** Failed - Ubuntu 24.04 repository not available
4. **Repository Addition (Attempt 2):** Used Ubuntu 22.04 repository
5. **Package Update (Attempt 1):** Failed - GPG signature errors
6. **Repository Fix:** Added signed-by configuration
7. **Package Update (Attempt 2):** Success
8. **SQL Server Installation:** Successfully installed package
9. **Initial Configuration:** Completed but service failed to start
10. **Troubleshooting:** Discovered missing OpenLDAP 2.5 libraries
11. **OpenLDAP Fix (Attempt 1):** Failed - Package not available in Ubuntu 24.04
12. **OpenLDAP Fix (Attempt 2):** Failed - Symlinks to version 2.6 (version mismatch)
13. **OpenLDAP Fix (Attempt 3):** Success - Downloaded and installed OpenLDAP 2.5 from Ubuntu 22.04
14. **Service Start:** Successfully started
15. **Tools Installation:** Installed sqlcmd and related tools
16. **Connection Test (Attempt 1):** Failed - SA password authentication error
17. **Password Reset:** Successfully reset SA password
18. **Connection Test (Attempt 2):** Success - Full functionality verified

5.2 Critical Issues and Resolutions

Issue	Resolution
Ubuntu 24.04 repository not available	Used Ubuntu 22.04 (jammy) repository which is compatible
PGP signature errors	Modified repository configuration to include signed-by option pointing to GPG keyring
OpenLDAP 2.5 missing (Ubuntu 24.04 has 2.6)	Downloaded OpenLDAP 2.5 package from Ubuntu 22.04, extracted libraries, and installed manually
SA password authentication failure	Reset SA password using mssql-conf set-sa-password command after stopping service

6 Post-Installation Configuration

6.1 Security Hardening

6.1.1 Change SA Password

```
1 sudo systemctl stop mssql-server
2 sudo /opt/mssql/bin/mssql-conf set -sa-password
3 sudo systemctl start mssql-server
```

6.1.2 Configure Firewall

```
1 # Allow SQL Server default port
2 sudo ufw allow 1433/tcp
3
4 # Or allow from specific network
5 sudo ufw allow from 192.168.1.0/24 to any port 1433
```

6.1.3 Enable Encryption

```
1 sudo /opt/mssql/bin/mssql-conf set network.forceencryption 1
2 sudo systemctl restart mssql-server
```

6.2 Performance Tuning

6.2.1 Configure Memory

```
1 # Set maximum memory (recommended: leave 25% for OS)
2 # For 8 GB system: set to 6144 MB
3 sudo /opt/mssql/bin/mssql-conf set memory.memorylimitmb 6144
4 sudo systemctl restart mssql-server
```

6.2.2 Enable SQL Server Agent

```
1 sudo /opt/mssql/bin/mssql-conf set sqlagent.enabled true
2 sudo systemctl restart mssql-server
```

7 Service Management

7.1 Common Commands

```
1 # Check service status
2 sudo systemctl status mssql-server
3
4 # Start service
5 sudo systemctl start mssql-server
6
7 # Stop service
8 sudo systemctl stop mssql-server
9
10 # Restart service
11 sudo systemctl restart mssql-server
12
13 # View real-time logs
14 sudo journalctl -u mssql-server -f
15
16 # View SQL Server error log
17 sudo tail -f /var/opt/mssql/log/errorlog
```

7.2 Important File Locations

Component	Location
SQL Server Binaries	/opt/mssql/
Database Files	/var/opt/mssql/data/
Log Files	/var/opt/mssql/log/
Configuration	/var/opt/mssql/mssql.conf
Systemd Service	/usr/lib/systemd/system/mssql-server.service
Command-Line Tools	/opt/mssql-tools18/bin/
ODBC Driver	/opt/microsoft/msodbcsql18/

8 Quick Reference

8.1 Connection String Format

```
Server=localhost ,1433;Database=master;User Id=SA;Password=
YourPassword;TrustServerCertificate=True;
```

8.2 Common sqlcmd Commands

```
1 # Interactive connection
2 sqlcmd -S localhost -U SA -C
3
4 # Execute single query
5 sqlcmd -S localhost -U SA -C -Q "SELECT name FROM sys.databases"
6
7 # Execute SQL file
8 sqlcmd -S localhost -U SA -C -i /path/to/script.sql
9
10 # Export query results to file
11 sqlcmd -S localhost -U SA -C -Q "SELECT * FROM sys.databases" -o
     output.txt
```

8.3 Password Change

```
1 sudo systemctl stop mssql-server
2 sudo /opt/mssql/bin/mssql-conf set -sa-password
3 # Enter new password when prompted
4 sudo systemctl start mssql-server
```

9 Conclusion

This guide has provided a complete walkthrough for installing Microsoft SQL Server 2022 (RTM-CU19) on Ubuntu 24.04 LTS, including critical workarounds for compatibility issues.

9.1 Key Takeaways

1. Ubuntu 24.04 requires using the Ubuntu 22.04 repository
2. OpenLDAP 2.5 must be manually installed from Ubuntu 22.04 packages
3. Repository configuration must include the signed-by option
4. SA password may need to be reset after initial configuration

9.2 Support and Updates

For the latest updates and security patches:

```
1 # Check for updates
2 apt-cache policy mssql-server
3
4 # Update to latest CU
5 sudo apt-get update
6 sudo apt-get upgrade mssql-server
```

9.3 Additional Resources

- Official SQL Server on Linux Documentation: <https://docs.microsoft.com/en-us/sql/linux/>
- SQL Server 2022 Release Notes: <https://docs.microsoft.com/en-us/sql/sql-server/>
- sqlcmd Utility: <https://docs.microsoft.com/en-us/sql/tools/sqlcmd-utility>

Document Version 1.0

Last Updated: November 5, 2025

Tested and Verified on Ubuntu 24.04.3 LTS