



# Microsoft SQL Server 2017 on Linux Quick Start Guide

## How to use this guide

The information provided here represents an ongoing process of providing insight on the features and capabilities of Microsoft SQL Server 2017 on Linux. Selections for this guide are made based on the questions and feedback voiced by attendees of the SQL Server on Linux Engineering Town Hall webinar series. Each section is intended to provide a brief on specific SQL Server topics, answers to a few frequently asked questions, and one or more links to additional documentation on the topic.

## Contents

Why SQL Server on Linux?.....	2
Requirements and specifications.....	2
SQL Server licensing .....	3
SQL Server PAL explained.....	3
Installing and configuring SQL Server .....	4
SQL Server tools.....	5
Graphical tools.....	5
SQL Server Management Studio (SSMS).....	5
SQL Server Data Tools (SSDT).....	5
VS Code + mssql extension.....	5
Command line tools .....	6
sqlcmd .....	6
bcp .....	6
mssql-scripter .....	6
DBFS .....	7
SQL PowerShell .....	7
Monitoring SQL Server .....	8
DMVs .....	8
Performance monitoring.....	8
Authentication and security.....	9
High availability .....	9
Pacemaker clustering and Always On Availability Groups.....	9
Read-scale Availability Group replicas .....	10
SQL Server Migration Assistant.....	10
Data Migration Assistant .....	11
Summary .....	12
What's next.....	12
Appendix.....	13

## Why SQL Server on Linux?

*Does SQL Server on Linux support multiple instances on the same host?*

*We recommend running multiple containers on a host to have multiple distinct instances running on a host. Each container will need to listen on a different port.*

---

*Which Linux file systems can SQL Server 2017 use for data files?*

*Currently SQL Server on Linux supports ext4 and XFS. Support for other file systems will be added as needed in the future.*

If you're running a mixed operating system environment, you already know that choice is key. Whether on Microsoft Windows Server or Linux, in virtual machines or containers, enterprises are expanding their deployment options. By bringing Microsoft SQL Server to Linux, Microsoft continues to embrace open source solutions.

SQL Server 2017 brings the best features of the Microsoft relational database engine to the enterprise Linux ecosystem, including SQL Server Agent, Azure Active Directory (Azure AD) authentication, best-in-class high availability/disaster recovery, and unparalleled data security.

Note that SQL Server on Linux is not a port or rewrite. This is the same world-class Microsoft relational database management system (RDBMS) now available on more operating systems (like Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and Ubuntu) and more cloud and container platforms (like Docker).

SQL Server 2017 provides enterprise-class features in every edition. This enables you to choose which SQL Server edition to deploy based on operational needs. These include features for security, such as Transparent Data Encryption (TDE), Always Encrypted (AE), or DDM/RLS, and performance features, such as In-Memory OLTP or In-Memory Columnstore. With Linux and container support, SQL Server 2017 can provide a full-featured database engine on any platform.

## Requirements and specifications

SQL Server 2017 (RC1) is supported on Red Hat Enterprise Linux (7.3), SUSE Linux Enterprise Server (v12 SP1), Ubuntu (16.04 and 16.10), and Docker Engine (1.8 and higher). SQL Server 2017 supports XFS and ext4 file systems—no other file systems are supported.

The minimum requirements for SQL Server on Linux are subject to change. Moreover, these are minimum-level requirements and are not representative of a production environment.

The specific details for system configuration will depend on the intended workload of the server, the volume and throughput needed for those workloads, and traffic considerations.

For the most up-to-date information on requirements and specifications for SQL Server, see the [documentation page on how to install SQL Server on Linux](#).

## SQL Server licensing

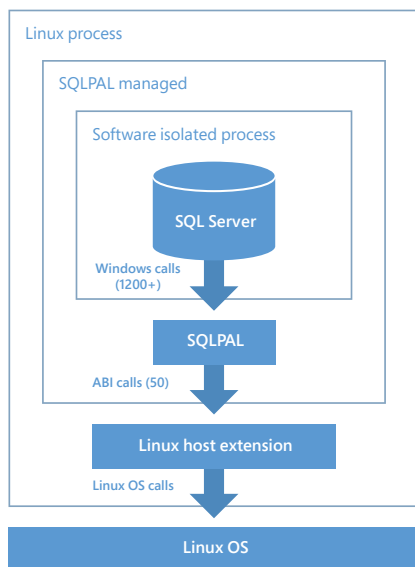
As SQL 2017 is a preview release (RC1), licensing for SQL Server has not been finalized. Licensing details will be made available as the general availability release for SQL Server 2017 approaches. (For more information, see [Get ready for SQL Server 2017](#).)

## SQL Server PAL explained

The SQL Server Platform Abstraction Layer (SQLPAL) is responsible for abstraction of the calls and semantics of the underlying operating system and its libraries from the SQL Server software itself. This approach has enabled the SQL Server team to deliver the same reliability and performance for which SQL Server is known to Linux operating environments.

All resources in the process are managed by SQLPAL. With this new architecture, even the Win32 and NT APIs would be based on SQLPAL, so every memory allocation and thread would be controlled by SQLPAL. This also benefits SQL Server on Windows.

For more detailed information on SQL Server PAL, see the [SQL Server Team's blog post](#) on how Linux support for SQL Server was achieved, or watch [SQL Server on Linux: The HOW Part 1](#) and [Part 2](#).



## Installing and configuring SQL Server 2017

*How do I get SQL Server installed on my Linux servers?*

*Microsoft maintains package repositories for installing SQL Server, and supports installation via native package managers such as yum, zypper, and apt.*

*Has Microsoft created an app like the SQL Server Configuration Manager on Linux?*

*Yes, there is a configuration tool for SQL Server on Linux: mssql-conf.*

Microsoft maintains repositories for installing, updating, and upgrading SQL Server on Linux distributions. This enables administrators to install SQL Server using the native Linux package manager with their distro such as yum for Red Hat Enterprise Linux, zypper for SUSE Linux Enterprise Server, and apt for Ubuntu.

At a basic level, the setup process breaks down as follows:

1. Add the appropriate package repository information for your Linux distribution
2. Run the native package installer command for your distro
3. Run the configuration script to complete setup
4. Optionally, enable communication for the appropriate (default 1433) port on your firewall

You can also refer to detailed steps for installing SQL Server 2017 on [Linux and Docker containers](#).

Configuration management for SQL Server on Linux can be done using the mssql-conf script. This script installs with SQL Server for Red Hat Enterprise Linux, SUSE Enterprise Linux, and Ubuntu.

Some of the configuration capabilities of the mssql-conf script enable you to accomplish these tasks:

Change the TCP port used by SQL Server to listen for remote connections

Change directory paths for data, log, dump, or backup files

Enable Availability Groups

Set trace flags

Set collation

(For more information, see [Configure SQL Server on Linux with mssql-conf](#).)

## *Can you run SQL Server on Linux/Docker in production?*

*While SQL Server on Linux and in Docker is available in public preview, you must join the [SQL Server Early Adoption Program](#) to run them in a production environment.*

---

## *Can I use the SQL Server Management Studio client on Windows to access SQL Server on Linux?*

*Yes, you can use all your existing tools that run on Windows to access SQL Server on Linux. These include tools from Microsoft such as SQL Server Management Studio (SSMS), SQL Server Data Tools (SSDT), and OSS and third-party tools.*

There are also [Transact-SQL commands](#) for database management tasks, and dynamic management views that can provide information about the configuration of a SQL Server instance. You can run these commands in any client tool that supports connecting to SQL Server and running Transact-SQL queries. Examples include [sqlcmd](#), [Visual Studio Code](#), and [SQL Server Management Studio](#).

Microsoft has a [public repository](#) for SQL Server Docker images in the [Docker Store](#). Step-by-step instructions for installing Docker and deploying containers are available on the [Microsoft documentation website](#).

## SQL Server tools

### Graphical tools

#### **SQL Server Management Studio (SSMS)**

SSMS is a free tool [available on the web](#) that offers comprehensive tools for the design and management of SQL Server databases, Analysis Services, Integration Services, and Reporting Services. SSMS is built on Visual Studio—meaning rich editing plus IntelliSense and extensibility—with more than 50 independent software vendors offering SSMS extension solutions. SSMS provides database administrations (DBAs) with the interface to perform HA/DR, database configuration and maintenance plans, SQL Server Agent management, query performance, and much more.

#### **SQL Server Data Tools (SSDT)**

SSDT is a free tool included in all versions of Visual Studio (2015–2017) that supports application and database development tasks, as well as database design and testing. SSDT supports Analysis Services, Integration Services, and Reporting Services, and it offers tools such as schema and data comparison. Since SSDT is integrated with Visual Studio, IDE features (such as rich editors, IntelliSense, and source control) are built in.

#### **VS Code + mssql extension**

Visual Studio Code is an open source multi-platform editor/IDE available on GitHub. VS Code supports extension via plug-in, of which thousands have been made available by the community. The mssql extension, created by Microsoft, brings many of the capabilities of SSDT to VS Code, allowing for database design, connection management, and query execution to any SQL Server instance (Windows, Linux, or Azure).

*Are commands like sqlcmd and bcp available on Linux?*

*Yes, sqlcmd and bcp are natively available on Linux, macOS, and Windows. In addition, use the new mssql-scripter command line tool on Linux, macOS, or Windows to generate T-SQL scripts for your SQL database running anywhere.*

## Command line tools

### sqlcmd

The sqlcmd command line utility is available on Linux, macOS, and Windows and lets you connect to SQL Server running anywhere, and execute adhoc T-SQL statements, system procedures, and script files at the command prompt.

See more details on [using sqlcmd with SQL Server on Linux](#).

### bcp

The bcp command line utility is available on Linux, macOS, and Windows and lets you efficiently bulk copy a large number of rows into SQL Server tables or export data out of tables into data files in a user-specified format. Except when used with the queryout option, bcp requires no knowledge of Transact-SQL. The bcp command line utility works with SQL Server running on-premises or in the cloud, on Linux, Windows, or Docker, and with Azure SQL Database and Azure SQL Data Warehouse.

See more details on [using bcp with SQL Server on Linux](#).

### mssql-scripter

Use the mssql-scripter command line utility on Linux, macOS, and Windows to generate data definition language (DDL) and data manipulation language (DML) T-SQL scripts for database objects in SQL Server running anywhere, Azure SQL Database, and Azure SQL Data Warehouse. You can choose to save the generated T-SQL script to a .sql file or pipe it to well-known Linux utilities such as sed, awk, and grep for further transformations. You can edit the generated script as desired or check it into source control and use tools such as sqlcmd to execute it in your existing SQL database deployment processes and DevOps pipelines.

mssql-scripter is built using Python and is open source. Get it from [the Python website](#) and contribute to [the source code on GitHub](#).

## DBFS

DBFS uses Filesystem in Userspace (FUSE) to mount [SQL Server Dynamic Management Views \(DMVs\)](#) as a virtual file system. This gives you the ability to explore information about your database using native Bash commands. DBFS creates two zero byte files for each DMV—one for showing the data in CSV format and one for showing the data in JSON format. When a file is read, the relevant information from the corresponding DMV is queried from SQL Server and displayed just like the contents of any CSV or JSON text file. DBFS can also be used in scenarios where access to DMV data from the context of a script with CLI tools such as grep, awk, and sed are desired.

Features:

- Access data in .json format if you are connected to SQL Server 2016 or later
- Compatible with Bash tools such as grep, sed, and awk
- Live DMV data at time of access
- Works with both SQL Server on Windows and SQL Server on Linux

Head over to our GitHub repo to get started with DBFS and [contribute to the source code](#).

## SQL PowerShell

SQL Server 2017 supports Windows PowerShell, which is a powerful scripting shell that lets administrators and developers automate server administration and application deployment. The Windows PowerShell language supports more complex logic than Transact-SQL scripts, giving SQL Server administrators the ability to build robust administration scripts. Windows PowerShell scripts can also be used to administer other Microsoft server products. This gives administrators a common scripting language across servers.

See more details on using [SQL PowerShell on Windows](#) with SQL Server on Linux.



## Monitoring SQL Server

*Is it possible to view Activity Monitor when connected through SSMS on Windows for an instance running on Linux?*

*Yes, you can use SSMS on Windows to connect remotely, and use tools/features such as Activity Monitor commands on a Linux instance.*

---

*What tools are available to monitor SQL Server performance on Linux?*

*You can use system dynamic management views to collect various types of information about SQL Server, including Linux process information.*

### DMVs

SQL Server Dynamic Management Views (DMVs) are built-in views and functions that return information about the server that can be used to monitor health and performance of an instance of SQL Server. The focus of these views and functions ranges from database information such as file and log space usage or partition statistics to Availability Group states. Starting with SQL Server 2017, a set of DMVs was specifically designed to access catalog metadata for SQL Server running on Linux. These include (but are not limited to):

- CPU information
- Memory
- SQL Server threads

You can access live DMVs mounted to a virtual filesystem using DBFS and FUSE. Users with permissions to access DBFS need only view the contents of the virtual files in the mounted virtual directory to see the same data as if running a SQL query to view the DMV data. There is no need to log in to the SQL Server using a GUI or command line tool or run SQL queries.

For more information, see [System dynamic management views](#).

### Performance monitoring

#### Live Query statistics

Display real-time statistics about query execution steps. Because this data is available while the query is executing, these execution statistics are extremely useful for debugging query performance issues.

The Activity Monitor in SQL Server Management Studio is useful for ad hoc views of current activity and graphically displays information about processes running on an instance of SQL Server, such as blocked processes, locks, and user activity.

For more information, see [Choose the right tool to manage SQL Server on Linux](#).

## Authentication and security

With RC1, SQL Server will support AD authentication using Kerberos protocol. This enables integrated security logins with SQL Server on Linux systems—meaning applications can leverage AD authentication when connecting to databases, without the need to store credentials in application configuration files. Currently, only AD logins can be authenticated. OS account logins are not supported.

For more information, see [Using Integrated Authentication](#).

## High availability

### Pacemaker clustering and Always On Availability Groups

Always On Availability Groups for Linux in SQL Server 2017 uses a cluster manager to provide business continuity. Always On Availability Groups works on all supported Linux distributions, namely Red Hat Enterprise Linux, Ubuntu, and the SUSE Linux Enterprise Server. The clustering layer is based on the distribution-specific, high-availability (HA) add-on that is built on top of Pacemaker. Integration with Pacemaker enables all the orchestration for monitoring, failure detection, and automatic failover required for a comprehensive high-availability solution.

With Windows Server Failover Clustering (WSFC) there is a tight integration between the cluster and SQL Server instance (FCI) or availability group replicas (AG), but the design approach for Linux is different. In Linux, orchestration between cluster and SQL Server is only outside-in. This makes it possible to create availability groups without a cluster prerequisite and enable scenarios such as massive read scale-out or cross-platform data migration. Also, this pluggable design opens the possibility of integrating with other cluster solutions.

As a consequence, it is expected that Always On DMVs that query cluster information return empty rows. You can still create a listener to use it for transparent reconnection after failover, but you will have to manually register the listener name in the DNS server with the IP used to create the virtual IP resource.

A Linux cluster uses fencing to return the cluster to a known state. The way to configure fencing depends on the distribution and environment.

*Are Always On and clustering supported in Linux?*

*Failover clustering and high availability on Linux are achieved with Pacemaker on Linux.*

*Is it possible to configure replication from Linux to Windows and vice versa?*

*Read-scale replicas can be used between Windows and Linux to replicate data one-way.*

---

*Is it possible to migrate existing SQL 2008 DB from Windows to Linux?*

*Yes, there are several methods of achieving this.*

## Read-scale Availability Group replicas

SQL Server allows for secondary Availability Group members to be accessed as a read-only data source. While these availability groups do not make for an HA/DR scenario, the ability to replicate data seamlessly across domains or regions offers some unique benefits.

### Data migration

Whether migrating from one datacenter or ecosystem to another, AG replicas offer a way to migrate data quickly without shuffling around backups and with minimum downtime. Using Distributed Availability Groups, organizations can migrate their high-availability configuration from one platform to another without risking exposure to events that can cause extended downtime.

### Application data

Read-only secondary replicas can be load-balanced to serve data to applications that only require read capabilities, such as backup, reporting, or business intelligence.

For more information, see [Availability Groups for SQL Server on Linux](#).

## Migration

You can migrate your existing data and databases into SQL Server on Linux in a number of ways. Microsoft offers migration tools to manage the transition to SQL Server for many popular relational database management solutions.

### SQL Server Migration Assistant

SQL Server Migration Assistant (SSMA) allows you to convert database schemas to SQL Server schemas, upload the schemas, and migrate data to the target server. SSMA manages the migration process with a graphical user interface that walks you step by step through the conversion, upload, and data migration process, highlighting any issues encountered during the conversion.

*Can I migrate my data from MySQL to SQL Server on Linux?*

*Yes. SSMA supports migration from several types of database engines: Microsoft Access, DB2, MySQL, Oracle, and SAP ASE (formerly SAP Sybase ASE).*

## Data Migration Assistant

Data Migration Assistant (DMA) enables you to upgrade to a modern data platform by detecting compatibility issues that can affect database functionality on your new version of SQL Server. It recommends performance and reliability improvements for your target environment.

The migration workflow helps you migrate database schemas, data and users, server roles, and SQL Server and Windows logins.

The latest version of DMA includes improved recommendations on using Stretch Database to maximize storage savings, including specific tables to stretch.

SSMA and DMA provide the means to easily migrate database and data for most mainstream solutions.

For more specialized scenarios, such as migrating structured data, see [Migrate databases and structured data to SQL Server on Linux](#).

## Summary

Microsoft continues to improve support for open source solutions, with SQL Server now available for Linux operating systems. The SQLPAL enables SQL Server to operate on Linux distributions with the same reliability and performance previously exclusive to Windows environments, now with support for AD authentication. The ability to run on containers enables DevOp support for data applications, as well as the ability to run multiple instances of SQL Server on a single host.

Microsoft offers several native and cross-platform tools and resources to monitor and manage server instances across Linux, Windows, or mixed operating environments. Linux clustering with Pacemaker, coupled with Always On Availability Groups, makes for an effective HA/DR solution, with full monitoring, failure detection, and failover. For those organizations looking to migrate from other database management systems, Microsoft offers tools such as SSMA to migrate and convert database schemas and migrate data, or Data Migration Assistant to upgrade or migrate data from various database types. As general availability for SQL Server 2017 approaches, Microsoft continues to bring more features and innovation with each preview release, enabling data-driven organizations to choose a solution that best meets their existing ecosystem—without the need for large-scale change.

## What's next

Read more about SQL Server 2017 on Windows and Linux and download the preview:

<https://www.microsoft.com/en-us/sql-server/sql-server-2017>

SQL Server 2017 release notes:

<https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-release-notes>

## Appendix

SQL Server 2017 Linux installation, requirements, and specifications:

[Installation guidance for SQL Server on Linux](#)

SQL Server 2017 Docker containers:

Documentation – [Run the SQL Server 2017 container image with Docker Repository](#)

SQL Platform Abstraction Layer:

Blog post – [SQL Server on Linux: How? Introduction](#)

Videos – [SQL Server on Linux: The HOW Part 1](#) and [Part 2](#)

SQL Server Management Studio:

[Download SSMS](#)

SQL Server tools documentation:

[sqlcmd](#)

[bcp](#)

[mssql-scripter](#)

[DBFS](#)

SQL PowerShell:

[Use PowerShell on Windows to Manage SQL Server on Linux](#)

Monitoring SQL Server:

[System Dynamic Management Views](#)

[Choose the right tool to manage SQL Server on Linux](#)

Authentication and security:

[Using integrated authentication](#)

Always On Availability Groups:

[Availability groups for SQL Server on Linux](#)

Migration:

[Migrate databases and structured data to SQL Server on Linux](#)