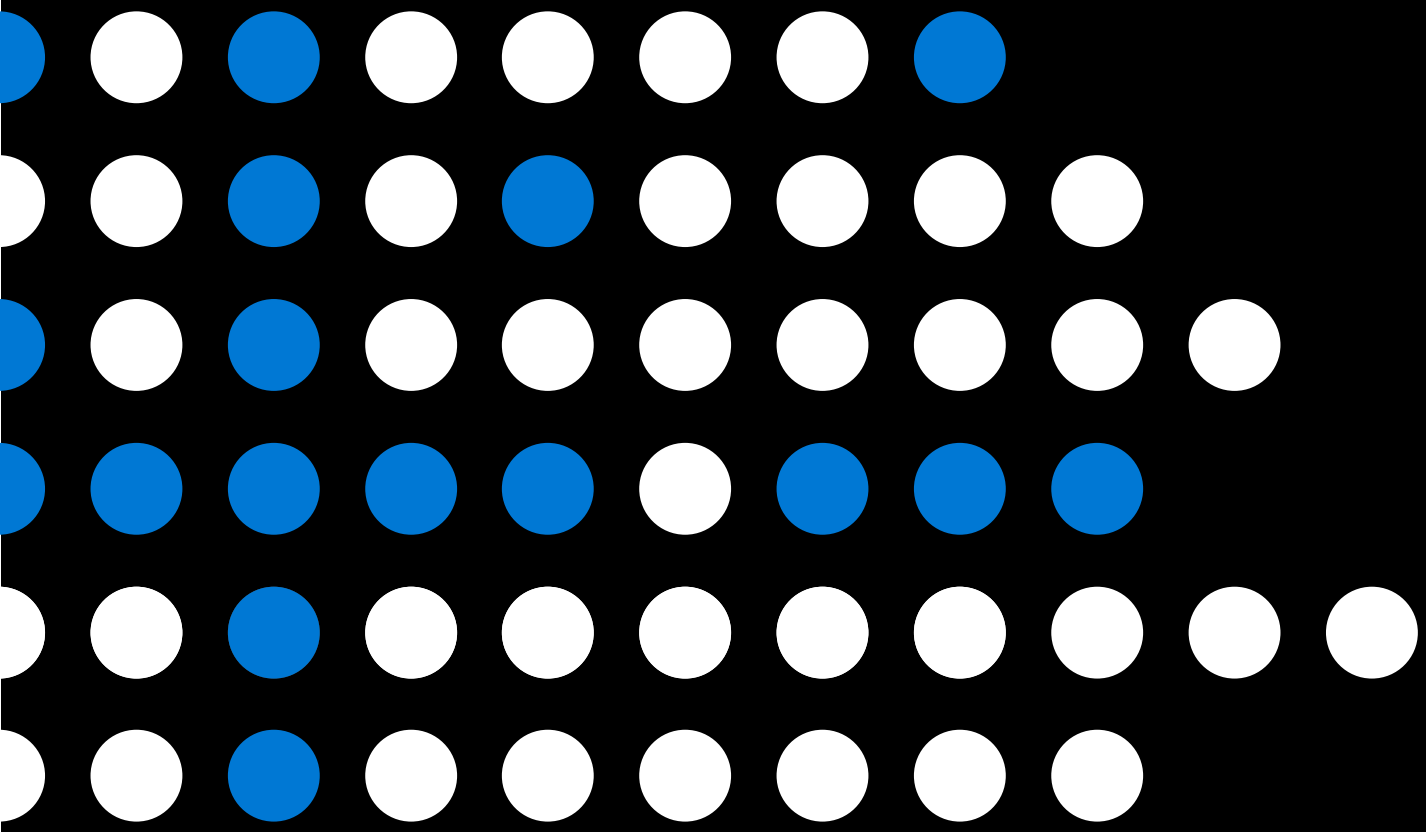
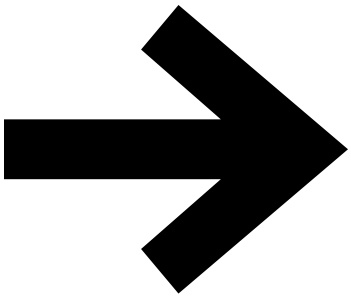
SQL Server e-book series

Microsoft

SQL Server on Linux: A guide to re-platforming and modernizing your data workloads





01 /

An overview of SQL Server on Linux

02 /

Getting the most out of SQL Server on Linux

03 /

Tools and management

04 /

Tools and services for migrations and upgrades

05 /

Conclusion

Who should read this e-book?

This e-book is for database architects, administrators, and developers looking for more flexibility, better performance, and increased security with their data platform. By reading this e-book, you'll learn how SQL Server on Linux enables you to harness the latest capabilities of SQL Server on your choice of platform—not only Windows and Linux, but containers, too. This e-book covers preparing your Linux system for SQL Server installation, managing a mixed environment, and migrating your existing data and databases into SQL Server on Linux—with technical details on how to put these procedures into practice.

An overview of SQL Server on Linux

"With SQL Server on Linux, data and analytics leaders now have a choice of operating systems, and will increasingly choose SQL Server on Linux, rather than rival products, for an array of use cases."

Source: Gartner's Microsoft DBMS Buyers Benefit from SQL Server on Linux Four Ways.

Innovation and modernization starts with your data platform. At the same time, as data platforms transform and branch out, identifying the best platform for pushing your work forward can be a formidable task. Customers are increasingly demanding more flexibility when it comes to their choice of platform, programming languages, and infrastructure so they're able to get the maximum impact from their data estate. Flexibility with your data platform enables you to build intelligent applications with any data and language on any operating system and in the cloud. Achieving this goal, however, depends upon a modern data estate solution that can solve the issues of managing the diverse data and the associated needs.

Microsoft is committed to providing a high-quality, enterprise-level database platform that enables you to choose the best operating system or combination of operating systems for your environment—whether that environment's preferred operating system is open source, proprietary, or a mixture of both. The deployment options for SQL Server, which was traditionally previously on Windows only, have now expanded to Linux and containers platforms.

SQL Server 2017 continues the evolution of SQL Server on the platform of your choice, improving and enhancing data management and data-driven applications through new capabilities and support for more Linux distributions, including Red Hat Enterprise Linux (RHEL), Ubuntu, and SUSE Enterprise Linux (SLES), as well as container platforms like Docker and Kubernetes.

Companies can deploy on the platform—or combination of platforms—that makes the most sense for their business. The same is true for programming languages, too. Companies using Oracle, PostgreSQL, DB2, Sybase, and other systems running on Linux can now migrate to the SQL Server 2017, an industry-leading Relational Database Management System (RDBMS), and use the operating system of their choice without the requirement to install Windows Server operating system. Engineers who are most comfortable with C#, Java, Ruby, or another coding language can build apps using the best language and platform for the job, and then host the apps on any cloud provider or on-premises server that suits their needs using SQL Server 2017.

SQL Server 2017 is engineered to handle relational and non-relational data, graph data coming from diverse data sources, and run any type of application including open-source applications. In this e-book, you'll get an operational view of this RDBMS, with a focus on leveraging SQL Server capabilities to deliver powerful insights in ways that open-source developers and IT specialists need and expect. In addition to now-standard features like advanced analytics and end-to-end business intelligence, SQL Server 2017 provides in-memory performance across workloads, mission-critical high availability, and unparalleled security features and compliance with privacy regulations such as GDPR, PCI, and HIPAA to protect your data at rest and in motion—all on your choice of language and platform.

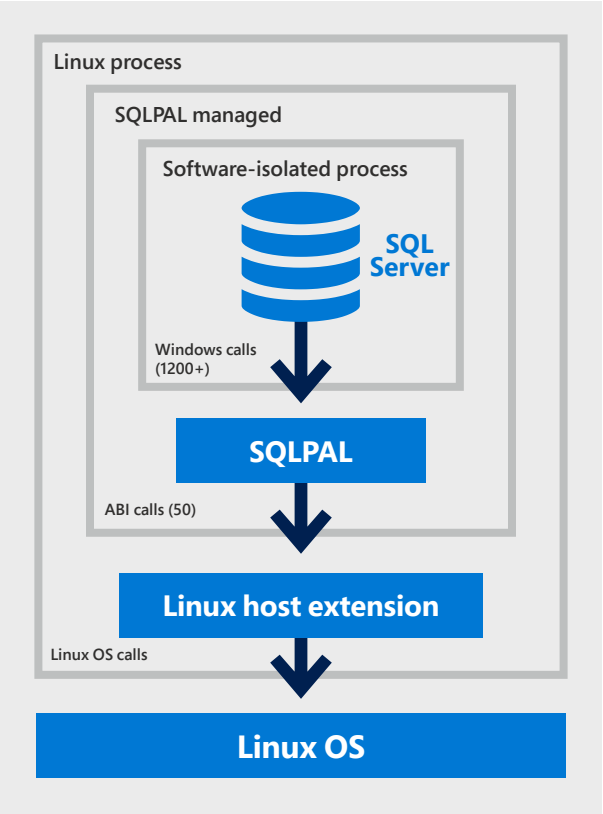
Choice across platforms

SQL Server on Windows and Linux share a common code base. That is, the SQL Server core engine hasn't been changed to allow it to run on Linux. SQL Server introduced a Platform Abstraction Layer (SQLPAL) that's responsible for abstraction of calls and communication between SQL Server and the underlying operating system. The host extension is simply a native Linux application. Low-level operating systems functions are native calls to optimize the input/output (I/O), memory, and CPU usage. When the host extension starts, it

loads and initializes SQLPAL, which then brings up SQL Server. SQLPAL launch software isolates processes that are merely a collection of threads and allocations, providing the required translation for the rest of the code.

Adding this new layer to the SQL Server architecture means that the same enterprise-level core features and benefits that have made SQL Server so powerful on Windows are available to users regardless of which OS the customer is using for SQL Server.

The response to this data platform evolution has been clear: customers are showing confidence in using SQL Server to manage their mission-critical data. Industry analysts have also responded positively. For example, Gartner has rated Microsoft as a leader with the most complete vision and highest ability to execute on any operational database management system for two consecutive years.[1] With Linux and container support, SQL Server 2017 now provides the same full-featured database engine on any platform with enterprise-grade capabilities. You can run SQL Server 2017 smoothly on Linux environments, which in turn offer integration with your existing SQL Server deployments.



It's not just Linux—SQL Server can now also run on Docker supported by Windows, Linux, and macOS

A Linux-based container can be deployed to any Linux machine (physical or virtual) running Docker and be expected to run without changes to the host operating system. With support for containers, SQL Server can run in container orchestration

[1] The complete Gartner document is available on Microsoft. Gartner does not endorse any vendor, product, or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

solutions such as Docker Swarm, Red Hat OpenShift, Kubernetes, and Microsoft Azure Container Services (AKS). With the Management Pack for SQL Server on Linux, administrators can use the System Center Operations Manager to monitor everything from the hardware up to the database engine instances and individual databases. Similarly, Docker Community Edition (CE) for Mac enables developers to run Linux containers on macOS.

SQL Server also supports installation on Windows containers. One of the key benefits of using containers in the development process is the ability to work in various environments. With SQL Server running in a container, application developers can make the database a dependency for the application. With containers, development teams can work in dev/test environments that are functionally identical to production environments. In addition, container orchestration solutions can manage the deployment of containers automatically, aiding the automation of testing and deployment.

Customer story:

Convergent Computing (CCO), a strategy and technology consulting firm, has reduced costs by 80 percent and achieved return on investment (ROI) in less than eight-and-a-half months after shifting their databases to SQL Server 2017 on both Linux and Windows.

Sometimes even faster on Linux

"SQL Server 2017 on RHEL delivers better results than the previous number 1 non-clustered TPC-H@1000GB result for SQL Server 2016 Enterprise Edition on Windows."

- 6 percent higher performance
- 5 percent lower price/performance

Source: Microsoft, Red Hat, and HPE Collaboration Delivers Choice & Value to Enterprise Customers

Industry-leading performance

SQL Server 2017 performance features on Linux provide similar impact. SQL Server delivers the same—and sometimes even better—performance for both online transaction processing (OLTP) and analytic workloads on Linux deployments. It's the same SQL Server: all the core relational databases, investments made in columnstore, and In-Memory OLTP working across both deployments. SQL Server 2017 owns a world-record 1 TB TPC-H benchmark result (non-clustered) for SQL Server on RHEL.[1] SQL Server on Linux can meet the requirements of the most demanding transaction processing applications, and Microsoft has worked closely with several companies to prove these gains. Microsoft also offers SQL Server 2017 Data Warehouse Fast Track Reference Architectures for Linux to help companies build enterprise data warehouse solutions. The Fast Track program uses the core capabilities of SQL Server on Windows or Linux servers to deliver a balanced symmetric multi-processing (SMP) data warehouse with optimized performance.

[1] TPC. "TPC-H Result Highlights HPE Proliant DL380 Gen9." March 2017. http://www.tpc.org/3327

Open-source pricing

Of course, your databases and the data that's within them are priceless. But that doesn't mean that you should be required to break the bank to work with your data. With SQL Server 2017 on Linux, it's possible to realize a lower total cost of ownership (TCO) and quicker return on investment (ROI) than if you choose a competing vendor. Required capabilities and features are built into SQL Server on Linux, including in-memory capabilities, security, data warehousing, and high availability/disaster recovery (HADR) at no extra cost.

Avoiding extra licenses, training time, and hardware costs will ensure your initial investment in SQL Server pays off.

Customer story:

Ajilius, an Australian startup, migrated its PostgreSQL system to SQL Server on the Linux data platform, resulting in accelerated development and cost savings. This new platform is easier to administer and hosting expenses are down by 30 percent. Plus, they can use SQL Server built-in capabilities to help boost query performance, improve security, and increase availability.

Discover more options with SQL Server on Linux

By bringing SQL Server to Linux, Microsoft continues to embrace open-source solutions. It represents a major step for Microsoft toward making SQL Server a platform that offers choices for development languages, data types, and operating systems on-premises or in the cloud. With SQL Server 2017 on Linux, you'll get more options—backed by Microsoft support and trust. These options include SQL Server 2017 Express, which is available for free and can help you start developing fast and intelligent applications with built-in security right away. Depending on your scalability and high availability requirements, you can also choose SQL Server 2017 Standard or Enterprise. Any of these editions enable you to develop once and deploy anywhere through SQL Server's consistent programming surface area across advanced security and performance features, regardless of size.

"SQL Server is the number-one database in the world, and it's easy to work with and administer. Ninety-nine percent of BI tools on the market integrate with SQL Server straightaway, and it provides excellent standards for security, high availability, and disaster recovery. There are also resources galore including books, forums, and consultants. Today, SQL Server on Linux is our primary platform."

- Ron Dunn, Lead Developer, Ajilius

Getting the most out of SQL Server on Linux

Although SQL Server 2017 has been engineered to deliver the same high-quality experience as its Windows-based counterpart, there are certain things you can do to prepare your Linux system to best support your business goals. Whether those goals center on performance, high availability, graph databases, or security, these planning considerations will help you get the most out of SQL Server on Linux.

You can install SQL Server on Linux from the command line by installing the mssql-server package and running mssql-conf setup. Below are the sample steps for installing SQL Server 2017 on RHEL.

```
//Download the Microsoft SQL Server Red Hat repository configuration file
sudo curl -o /etc/yum.repos.d/mssql-server.repo https://packages.microsoft.com/config/rhel/7/mssql-server-2017.repo
// install SQL Serversudo yum install -y mssql-server

.// run the mssql-conf setup
sudo /opt/mssql/bin/mssql-conf setup
```

For guidance on installing SQL Server on Ubuntu and SUSE Linux Enterprise Server, see the QuickStart guides: Install SQL Server and create a database on Ubuntu, Install SQL Server and create a database on SUSE Linux Enterprise Server. To run the SQL Server 2017 container image with Docker, you need to pull and run the mssql-server-linux container image. For details on how to do this, see the QuickStart guide to Run the SQL Server 2017 container image with Docker.

Performance optimization

To optimize SQL Server performance, there are various best practices for disk layout design that apply on Linux and Windows. For example, disk striping enables you to add additional data disks in your environment for your log, data, and tempdb files. With disk striping, you can then mount these files and configure tempdb, transaction logs, and data files to stripe the I/O across disks. With disk striping, you can segment a single larger disk into multiple smaller disks to optimize the I/O operations per second (IOPS). You can analyze the number of IOPS measurements and bandwidth required for the associated data disks.

```
Use Master;
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = tempdev, FILENAME = ‘/var/opt/mssq1/data/tempdb01/tempdb01.mdf’, SIZE =- 1024, FILEGROWTH = 8192 KB) ;
GO
ALTER DATABASE tempdb MODIFY FILE (NAME = templog, FILENAME – ‘/var/opt/mssqIl/datalog/tempdbOI/tempIog.ldf’, SIZE = 1024KB , FILEGROWTH = 8192KB  );
GO
```

Here are some other best practices and post-installation options that can help you maximize performance for database applications that connect to SQL Server on Linux:

- To maintain efficient Linux and SQL Scheduling behavior, it's recommended to use the ALTER SERVER CONFIGURATION command to set PROCESS AFFINITY for all the NUMANODEs and/or CPUs.
- To reduce the risk of tempdb concurrency slowdowns in high-performance environments, configure multiple tempdb files according to guidelines found through Microsoft Support. The automatic tempdb configuration feature introduced in SQL Server 2016 doesn't offer an option to configure multiple tempdb files when installing SQL Server on Linux. However, additional tempdb files can be added using the ADD FILE Transact-SQL (T-SQL) command.
- If you plan to use In-Memory OLTP in SQL Server 2017, you must ensure your database compatibility level is at 140 (SQL Server 2017). Be aware that if you've upgraded from a previous version, the compatibility level may not have been changed.

  Check the compatibility level of your database using the following T-SQL command:

```
ALTER DATABASE YourDBName
SET COMPATIBILITY_LEVEL = 140;
GO
```

Memory configuration settings

- Use mssql-conf to configure the memory limit and ensure there's enough free physical memory for the Linux operating system.

Linux OS configuration settings

- On multi-node Non-Uniform Memory Access (NUMA) installations, auto NUMA balancing needs to be disabled to allow SQL Server to operate at maximum efficiency on a NUMA system.

```
sysctl -w kernel.numa_balancing=0
```

- You can also change the kernel settings value for virtual address space to 256K, as the default value of 65K may be insufficient for a SQL Server installation.

```
sysctl -w vm.max_map_count=262144
```

- Use the `noatime` attribute to disable last accessed timestamps with any file system that is used to store SQL Server data and log files.

- For the most consistent performance experience, you must leave the Transparent Huge Pages (THP) option enabled. In most cases this is enabled by default.
- Virtual machine (VM) features like Hyper-V Dynamic Memory shouldn't be used with SQL Server installations. When using VMs, be sure to assign sufficient fixed-memory sizes.

For details on enabling some of the performance features of SQL Server on Linux, such as how to create columnstore indexes or configuration for In-Memory OLTP, see Walkthrough for the performance features of SQL Server on Linux.

High-availability configurations

SQL Server 2017 empowers you to improve how you use your resources on Windows and Linux platforms, boosting mission-critical uptime, failover, and manageability through high availability. It enables various HADR scenarios—Always On Failover Cluster Instances (FCIs), Always On Availability Groups, and log shipping—that can help your organization achieve a wide range of availability service-level agreements (SLAs).

High Availability with Always On Failover Cluster Instances

Here are features that you'll want to be aware of while setting up FCIs on Linux:

Clustering. The clustering layer is managed by the cluster controller Pacemaker, which coordinates communications and resource management. On RHEL, it's provided by the RHEL High Availability Add-On. On SLES, the necessary packages are provided by the SUSE Linux Enterprise High Availability Extension (HAE).

Number of instances and nodes. A Linux-based FCI can have only have a single instance—that is, only one instance of SQL Server per Linux server unless you're using containers.

IP Address and hostname. Each SQL Server instance requires its own IP address and hostname. These are used internally by Pacemaker to communicate with specific instances. Pacemaker setup includes the creation of a virtual resource with its own IP, allowing applications to connect to data without needing to identify cluster-internal resources.

Shared storage. All FCIs on Linux or Windows Server require some form of shared storage. The options available for shared storage on Linux are iSCSI and Network File System (NFS), and Server Message Block (SMB) on Windows Server. In a configuration that spans multiple locations, data stored in one datacenter must be synchronized across locations. For default user data and log file locations, the system databases must always exist at /var/opt/mssql/data for all instances.

FCI resource group. The FCI on Linux nodes must be created in a resource group. You can create the FCI resource using the pcs resource command. After ensuring that the FCI is online, you can issue the normal SQL statements using SSMS or sqlcmd.

```
sudo pcs resource create FCIResourceName
ocf:mssql:fci op defaults timeout=60s
--group RGName
```

For further guidance on how to create a SQL Server FCI on Linux, see Microsoft documentation.

High availability with Always On Availability Groups

For Always On Availability Groups under Linux-based SQL Server installations, you'll want to consider characteristics and configuration settings such as:

Configure setting for SQL Server Availability Group on Linux. To create an availability group on Linux servers, you need to enable availability groups on each Linux node with endpoints and certificates. Use T-SQL (or the New Availability Group Wizard in SSMS on Windows) to create an availability group with the desired cluster type. After an availability group is created on SQL Server, you'll also need to create the corresponding resources in Pacemaker. To get started with Availability Groups in SQL Server 2017 on Linux, see Create and configure an availability group for SQL Server on Linux.

Cluster type. For an availability group on Linux, set the cluster type to External or None. An External type means using Pacemaker with the availability group while None means that there's no requirement to use Pacemaker. An External cluster type with Pacemaker helps you query the instances of SQL Server in the availability group and orchestrate failover to maintain high availability. A cluster type of None only supports manual failover from a primary to a secondary replica and is primarily targeted for the read-scale out scenario. For listener connectivity in this context, you can use an IP address resource created in Pacemaker that can run on any of the nodes. Read more about listener functionality under Linux.

Number of replicas and cluster nodes. SQL Standard Edition supports two nodes in an availability group and one database per availability group, while Enterprise Edition can have up to nine nodes in an availability group. Configuration-only nodes do not count against these limits. Use a configuration-only replica if you want to configure a two-replica with the ability to automatically fail over to another replica. Read about replicas and cluster nodes.

Configuration-only replica and quorum. Make sure to check that Pacemaker is configured properly so it continues to operate. That also means ensuring quorum and STONITH are implemented properly from a Pacemaker perspective, in addition to any SQL Server requirements such as a configuration-only replica. For more information, see Configuration-only replica and quorum.

SQL Server resource agent for Pacemaker. SQL Server 2017 added a sequence number option to sys.availability_groups. This allows Pacemaker to identify how up-to-date secondary replicas are with the primary replica. Pacemaker updates the sequence_number with each availability group configuration change. Examples of configuration changes include failover, replica addition, or removal.

REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT. SQL Server 2017 also offers this option for the cluster resource setting, which can be changed via Pacemaker. This setting guarantees that the specified number of secondary replicas log the transaction data before the primary replica commits each transaction. You can set the value of this parameter for high availability and data protection. For more details on design patterns and its capabilities, see How the configuration affects default resource settings.

Security features

SQL Server 2017 offers several built-in security features on Linux deployment to help you protect and secure your organization's sensitive data both in motion and at rest.

Transparent Data Encryption (TDE) encrypts database at rest, without requiring any application changes.

Backup Encryption enables you to encrypt data files while creating backups.

Always Encrypted enciphers sensitive data on the client side.

SQL Authentication adds an authentication layer via a username and password.

Active Directory (AD) Authentication offers single sign-on through Active Directory and Kerberos.

Granular permissions help you control access to individual tables or columns of data.

Row-Level Security (RLS) enables you to manage access to rows in a table based on a customizable policy.

Dynamic data masking (DDM) limits sensitive data exposure by obscuring it to non-privileged users.

Fine-grained audit features help you enforce a data audit policy and track user activity.

To learn more about these security features, see Overview of SQL Server Security.

Tools and management

Managing a mixed environment—including Linux and Windows deployments—requires many different tools. Microsoft makes it easier by offering several cross-platform tools for SQL Server.

SQL Operations Studio

SQL Operations Studio is a new cross-platform tool for SQL Server. Built upon VSCode, this free tool runs on Windows, macOS, and Linux, and is ideal for managing SQL Server on-premises, in containers, or in the cloud. It offers a modern, keyboard-focused T-SQL coding experience with built-in features such as multiple tab windows, a rich T-SQL editor, IntelliSense, keyword completion, code snippets, code navigation, and source control integration (Git). SQL Operations Studio enables on-demand T-SQL queries, viewing and saving results in common formats such as text, JSON, or Microsoft Excel. You can use it to organize your database connections and browse database objects in a familiar object browsing experience on the platform of your choice.

Visual Studio Code and SQL Server extensions

Visual Studio (VS) Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. VS Code is free and available on your favorite platform—Linux, macOS, and Windows. It has a rich ecosystem of extensions found in the VS Code Extension Marketplace.

The mssql extension for VS Code enables you to connect to SQL Server, query with T-SQL statements, and view the results. You can save results as a JSON or CSV file to use the data in your applications with just a few clicks. While typing T-SQL code, you get rich T-SQL language features like T-SQL IntelliSense (code completion), syntax highlighting, linting, code navigation, and code snippets.

You can also use VS Code to establish a connection to the SQL Server Integration Services (SSIS) Catalog using an SSIS extension, and then use T-SQL statements to deploy an SSIS project to the SSIS Catalog.

mssql-cli command-line query tool

An interactive command-line tool for querying SQL Server, mssql-cli includes many modern features such as auto-completion and syntax highlighting. Based on Python, mssql-cli uses a preferred installer program (pip) to install it. Once it's installed, you can launch the tool from the command line using mssql-cli --help. As an open-source tool that works across platforms, you can install mssql-cli on Windows, macOS, or Linux. For more information on how to install mssql-cli and usage, see documentation on the mssql-cli GitHub repository.

SQL Server Management Studio (SSMS)

Even though SQL Server Management Studio isn't made to run on Linux, it can be used—from a Windows server—to manage SQL Server instances on Linux OS remotely. SSMS is a powerful database administration tool that can help you back up, restore, view, and edit databases, plus create and edit T-SQL queries, scripts, and database objects. See specific instructions on how to use SSMS to connect remotely to your SQL Server instance on Linux.

System Center Management Pack (Management Pack)

Microsoft System Center Management Pack for SQL Server 2017 is another tool that enables the discovery and monitoring of SQL Server 2017 Database Engines,

databases, and other related components on Windows and Linux deployments. There's no need for you to disrupt ongoing monitoring approaches and managing systems—all you have to do is to import a new version of the Management Pack and configure it to facilitate tracking of SQL Server. Using Management Pack, you can monitor SQL 2017 on both Windows and Linux. Plus, Management Pack supports agentless monitoring on Linux. Using the agentless mode, you can shift monitoring workloads to management servers included in the SQL Server Monitoring Pool. This enables you to remove System Center Operations Manager (SCOM) and data processing overhead from the SQL Server host and move it to the SQL Server Monitoring Pool. The monitoring provided by the management pack includes performance, availability, and configuration monitoring, as well as performance and events data collection. All monitoring workflows have predefined thresholds and complementary knowledge base articles.

Tools and services for migrations and upgrades

There are various ways to migrate your existing data and databases into SQL Server on Linux. Microsoft can help you manage this transition to SQL Server with migration tools for many popular relational database management solutions.

SQL Server Migration Assistant (SSMA)

SSMA allows you to convert database schemas to SQL Server schemas, upload the schemas, and migrate data to the target server. It 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. SSMA is a Windows application, so we recommend using SSMA when you have a Windows machine that can connect to a remote SQL Server instance on Linux. SSMA supports a variety of source databases including Oracle, MySQL, SAP Adaptive Server Enterprise (ASE), DB2, and Microsoft Access. Plus, SSMA helps automate migration tasks such as assessing your source database, converting the source database schema to SQL Server schema, migrating the schema and the data, and, lastly, testing the migration. For help choosing the version of SSMA based on the source database, see Microsoft documentation.

Data Migration Assistant (DMA)

DMA detects compatibility issues that can affect database functionality on your new version of SQL Server before you migrate. If issues are detected, DMA can offer suggestions on how to address them before proceeding. It recommends performance and reliability improvements for your target environment, including newer features that are available through a version upgrade. The migration workflow helps you migrate database schemas, data and users, server roles, and SQL Server logins. While the migration of Active Directory (AD) logins is officially supported by SQL Server 2017 on Linux, this requires additional configuration for setting up AD logins on SQL Server 2017 on Linux. (For more information, see Configuration steps.) Standard SQL Authentication works as expected without any additional setup.

Additionally, Data Migration Services (DMS) simplify the migration of existing on-premises SQL Server, Oracle, and MySQL databases to Microsoft Azure, whether your target database is SQL Database, SQL Database Managed Instance, or SQL Server on an Azure virtual machine. The automated workflow guides you through the changes you need to make before performing the migration. The options presented through the guided process are easy to understand and implement.

Database Experimentation Assistant (DEA)

DEA helps in evaluating a targeted version of SQL for upgrade. It's another important tool for customers upgrading from previous SQL Server versions (starting 2005 and above) to any newer version of SQL Server. DEA provides analysis reports such as queries metrics for compatibility errors, degraded queries and query plans, and other workload comparison reports to help you upgrade successfully. This tool helps you set up automated workload capture and replay of production database, perform statistical analysis on traces collected using both old and new instances, and visualize the collected data through analysis report. You can install DEA through the Microsoft Download Center.

Availability groups as a migration path

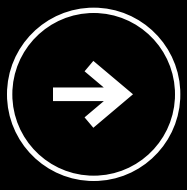
For migrating from one datacenter or ecosystem to another, availability group replicas offer a way to move quickly without shuffling around backups and with minimal downtime. You can use the cross-platform Always On Availability Group configuration, where part of the replica is on a Windows server and the other replica is on a Linux

server, for migration from one platform to the other. However, since Windows and Linux use different controllers to manage replica failover (WSFC and Pacemaker), any availability group containing a mix of operating systems needs to use a cluster type of None and therefore requires manual failover. Also, since Distributed Availability Groups support two completely different availability groups, you can migrate your organization's high-availability configuration from one platform to another without risking exposure to events that can cause extended downtime. For more information, see documentation on using distributed availability groups.

Conclusion

The modern data platform spans across heterogeneous environments, where operational databases, data warehouses, and different data sources combine to drive insights from data. To innovate, you need flexibility when it comes to your choice of platform, programming languages, and data infrastructure to get the most from your data.

SQL Server 2017 is the first version of SQL Server to support Windows, Linux, macOS, and Docker containers. This new support represents a major step for SQL Server as a platform of choice for development and cross-platform operating systems, giving you more flexibility with your data solutions. You can also benefit from other cutting-edge technologies already available in SQL Server with In-Memory OLTP, In-Memory Clustered Columnstore, Availability Groups, Adaptive Query Processing, Automatic Tuning, and enterprise-level security all built-in. With SQL Server 2017, organizations like yours have a full range of options and features for building or extending data ecosystems across operating systems and programming languages.



Resources
Learn more about running SQL Server 2017 on your favorite platform.

Get help installing SQL Server 2017 on Linux:
Red Hat Enterprise Linux (RHEL) QuickStart installation guide.

Docker QuickStart installation guide.