



Microsoft SQL Server

Enterprise applications

NetApp
March 13, 2024

Table of Contents

- Microsoft SQL Server 1
 - Microsoft SQL Server Overview 1
 - Microsoft SQL Server workloads 1
 - Database configuration 2
 - Storage configuration 11
 - Microsoft SQL Server data protection with ONTAP 22
 - Microsoft SQL Server disaster recovery 23
 - Database security 24

Microsoft SQL Server

Microsoft SQL Server Overview

SQL Server is the foundation of Microsoft's data platform, delivering mission-critical performance with in-memory technologies and faster insights on any data, whether on the premises or in the cloud.



This documentation replaces the previously published technical report *TR-4590: Best practice guide for Microsoft SQL Server with ONTAP*

Microsoft SQL Server builds on the mission-critical capabilities delivered in prior releases by providing breakthrough performance, availability, and manageability for mission-critical applications. The storage system is a key factor in the overall performance of a SQL Server database. NetApp provides several products to allow your SQL Server database to deliver enterprise-class performance while providing world-class tools to manage your environment.

Purpose and scope

This section describes best practices and offers insight into design considerations for deploying SQL Server on NetApp storage systems running NetApp ONTAP software, with the goal of achieving effective and efficient storage deployment and end-to-end data protection and retention planning. The scope of this guide is limited to technical design guidelines based on the design principles and preferred standards that NetApp recommends for storage infrastructure when deploying SQL Server. The end-to-end implementation is out of the scope of this report.

The best practices and recommendations described in this guide enable SQL Server architects and NetApp storage administrators to plan a highly available and easy-to-manage SQL Server environment and to meet stringent SLAs. NetApp assumes that the reader has working knowledge of the following:

- NetApp ONTAP software
- NetApp SnapCenter as backup software, which includes:
 - SnapCenter Plug-in for Microsoft Windows
 - SnapCenter Plug-in for SQL Server
- Microsoft SQL Server architecture and administration

For configuration compatibility across the NetApp stack, see the [NetApp Interoperability Matrix Tool \(IMT\)](#).

Microsoft SQL Server workloads

The SQL Server database platform can support many applications.

Before deploying SQL Server, you must understand the database workload requirements of the applications that your SQL Server instances support. Each application has different requirements for capacity, performance, and availability, and therefore each database should be designed to optimally support those requirements. Many organizations classify databases into multiple management tiers, using application requirements to define SLAs. SQL Server workloads can be described as follows:

- OLTP databases are often also the most critical databases in an organization. These databases usually

back customer-facing applications and are considered essential to the company's core operations. Mission-critical OLTP databases and the applications they support often have SLAs that require high levels of performance and are sensitive to performance degradation and availability. They might also be candidates for Always On Failover Clusters or Always On Availability Groups. The I/O mix of these types of databases is usually characterized by 75% to 90% random read and 25% to 10% write.

- Decision support system (DSS) databases can be also referred to as data warehouses. These databases are mission critical in many organizations that rely on analytics for their business. These databases are sensitive to CPU utilization and read operations from disk when queries are being run. In many organizations, DSS databases are the most critical during the month, quarter, and year end. This workload typically has a 100% read I/O mix.

Benchmarking

The Transaction Process Council (TPC) is a nonprofit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable TPC performance data to the industry. TPC tests simulate complete compute environments in which a population of users executes transactions against databases.

Workload type	Scenario	Read/write ratio (percentages)
OLTP	TPC-C	~75/25
	OLTP	TPC-E
~90/10		DSS
TPC-H	~100/0	

Although various workload generation options are available, we generally focus our efforts on measuring the performance of SQL Server databases when handling transactional workloads, and we use the TPC-E tools from Microsoft or TPC-H using HammerDB ([HammerDB.com](https://hammerdb.com)[^]). The detailed instructions on how to use these specific benchmarks are beyond the scope of this document.

Database configuration

Microsoft SQL Server shared instance versus dedicated instance

If an application has many schemas and stored procedures, it could potentially affect other apps that share a SQL Server instance.

Instance resources could potentially become divided or locked, which in turn causes performance issues for other apps that have databases hosted on the shared SQL Server instance.

Troubleshooting performance issues can be complicated because you must figure out which instance is the root cause. This question is weighed against the costs of operating system licenses and SQL Server licenses. If application performance is paramount, then a dedicated instance is highly recommended.

Microsoft licenses SQL Server per core at the server level and not per instance. For this reason, database administrators are tempted to install as many SQL Server instances as the server can handle to save on licensing costs, which can lead to major performance issues later.



NetApp recommends choosing dedicated SQL Server instances whenever possible to obtain optimal performance.

CPU configuration

Modifying server and database settings can help achieve better and more efficient database performance.

Hyperthreading

Hyperthreading is Intel's proprietary simultaneous multithreading (SMT) implementation, which improves parallelization of computations (multitasking) performed on x86 microprocessors.

Hardware that uses hyperthreading allows the logical hyperthread CPUs to appear as physical CPUs to the operating system. SQL Server then sees the physical CPUs, which the operating system presents, and so can use the hyperthreaded processors.

The caveat here is that each SQL Server version has its own limitations on the compute power it can use. For more information, see [Compute Capacity Limits by Edition of SQL Server](#).

There are two main schools of thought when licensing SQL Server. The first is known as a server + client access license (CAL) model; the second is the per processor core model. Although you can access all the product features available in SQL Server with the server + CAL strategy, there is a hardware limit of 20 CPU cores per socket. Even if you have SQL Server Enterprise Edition + CAL for a server with more than 20 CPU cores per socket, the application cannot use all those cores at a time on that instance. Figure shows the SQL Server log message after startup indicating the enforcement of the core limit.

Log entries indicate number of cores being used after SQL Server startup.

```

2017-01-11 07:16:30.71 Server      Microsoft SQL Server 2016
(RTM) - 13.0.1601.5 (X64)
Apr 29 2016 23:23:58
Copyright (c) Microsoft Corporation
Enterprise Edition (64-bit) on Windows Server 2016
Datacenter 6.3 <X64> (Build 14393: )

2017-01-11 07:16:30.71 Server      UTC adjustment: -8:00
2017-01-11 07:16:30.71 Server      (c) Microsoft Corporation.
2017-01-11 07:16:30.71 Server      All rights reserved.
2017-01-11 07:16:30.71 Server      Server process ID is 10176.
2017-01-11 07:16:30.71 Server      System Manufacturer:
'FUJITSU', System Model: 'PRIMERGY RX2540 M1'.
2017-01-11 07:16:30.71 Server      Authentication mode is MIXED.
2017-01-11 07:16:30.71 Server      Logging SQL Server messages
in file 'C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\Log\ERRORLOG'.
2017-01-11 07:16:30.71 Server      The service account is 'SEA-
TM\FUJIA2R30$'. This is an informational message; no user action
is required.
2017-01-11 07:16:30.71 Server      Registry startup parameters:
-d C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\DATA\master.mdf
-e C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\Log\ERRORLOG
-l C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\DATA\mastlog.ldf
-T 3502
-T 834
2017-01-11 07:16:30.71 Server      Command Line Startup
Parameters:
-a "MSSQLSERVER"
2017-01-11 07:16:30.72 Server      SQL Server detected 2 sockets
with 18 cores per socket and 36 logical processors per socket,
72 total logical processors; using 40 logical processors based
on SQL Server licensing. This is an informational message; no
user action is required.
2017-01-11 07:16:30.72 Server      SQL Server is starting at

```

Therefore, to use all CPUs, you should use the per-processor core license. For detailed information about SQL Server licensing, see [SQL Server 2022: Your modern data platform](#).

CPU affinity

You are unlikely ever to need to alter the processor affinity defaults unless you encounter performance problems, but it is still worth understanding what they are and how they work.

SQL Server supports processor affinity by two options:

- CPU affinity mask
- Affinity I/O mask

SQL Server uses all CPUs available from the operating system (if the per-processor core license is chosen). It creates schedulers on all the CPUs to make best use of the resources for any given workload. When multitasking, the operating system or other applications on the server can switch process threads from one processor to another. SQL Server is a resource-intensive application, and so performance can be affected when this occurs. To minimize the effect, you can configure the processors such that all the SQL Server load is directed to a preselected group of processors. This is achieved by using the CPU affinity mask.

The affinity I/O mask option binds SQL Server disk I/O to a subset of CPUs. In SQL Server OLTP environments, this extension can enhance the performance of SQL Server threads issuing I/O operations.

Max Degree of Parallelism (MAXDOP)

By default, SQL Server uses all available CPUs during query execution (if the per-processor core license chosen).

Although this is great for large queries, it can cause performance problems and limit concurrency. A better approach is to limit parallelism to the number of physical cores in a single CPU socket. For example, on a server with two physical CPU sockets with 12 cores per socket, regardless of hyperthreading, MAXDOP should be set to 12. MAXDOP cannot restrict or dictate which CPU is to be used. Instead, it restricts the number of CPUs that can be used by a single batch query.



NetApp recommends for DSS such as data warehouses, start with this setting at 50 or so and tuning up or down as appropriate. Make sure you measure for the critical queries in your application and adjust if necessary.

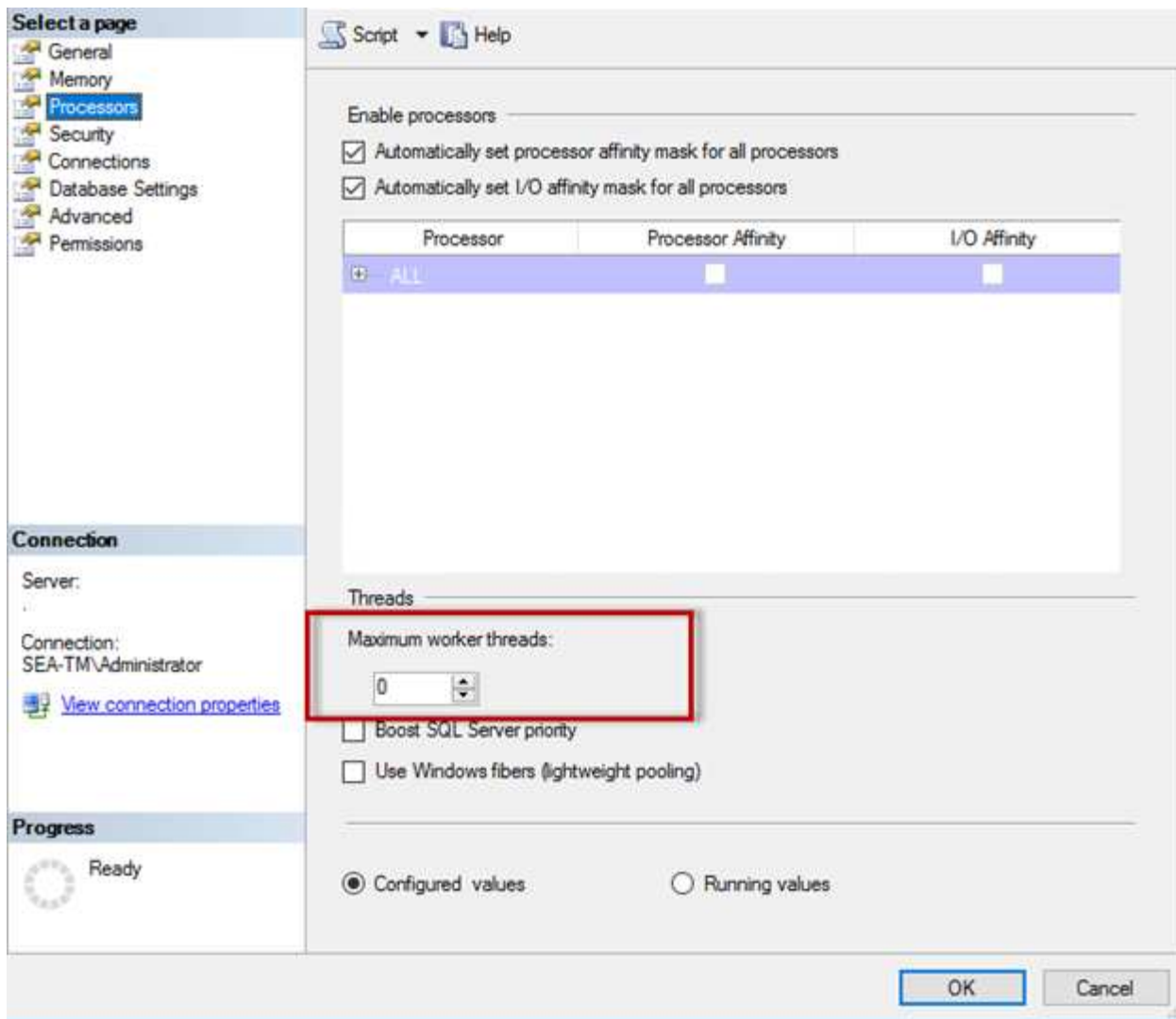
Max worker threads

The max worker threads option helps to optimize performance when large numbers of clients are connected to SQL Server.

Normally, a separate operating system thread is created for each query request. If hundreds of simultaneous connections are made to SQL Server, then one thread per query request consumes large amounts of system resources. The max worker threads option helps improve performance by enabling SQL Server to create a pool of worker threads to service a larger number of query requests.

The default value is 0, which allows SQL Server to automatically configure the number of worker threads at startup. This works for most systems. Max worker threads is an advanced option and should not be altered without assistance from an experienced database administrator (DBA).

When should you configure SQL Server to use more worker threads? If the average work queue length for each scheduler is above 1, you might benefit from adding more threads to the system, but only if the load is not CPU-bound or experiencing any other heavy waits. If either of those is happening, adding more threads does not help because they end up waiting for other system bottlenecks. For more information about max worker threads, see [Configure the max worker threads Server Configuration Option](#).



Configuring max worker threads using SQL Server Management Studio.

The following example shows how to configure the max work threads option using T-SQL.

```
EXEC sp_configure 'show advanced options', 1;

GO

RECONFIGURE ;

GO

EXEC sp_configure 'max worker threads', 900 ;

GO

RECONFIGURE;

GO
```

Microsoft SQL Server memory configuration

The following sections explain some of the critical memory configuration settings.

Max server memory

The max server memory option sets the maximum amount of memory that the SQL Server instance can use.

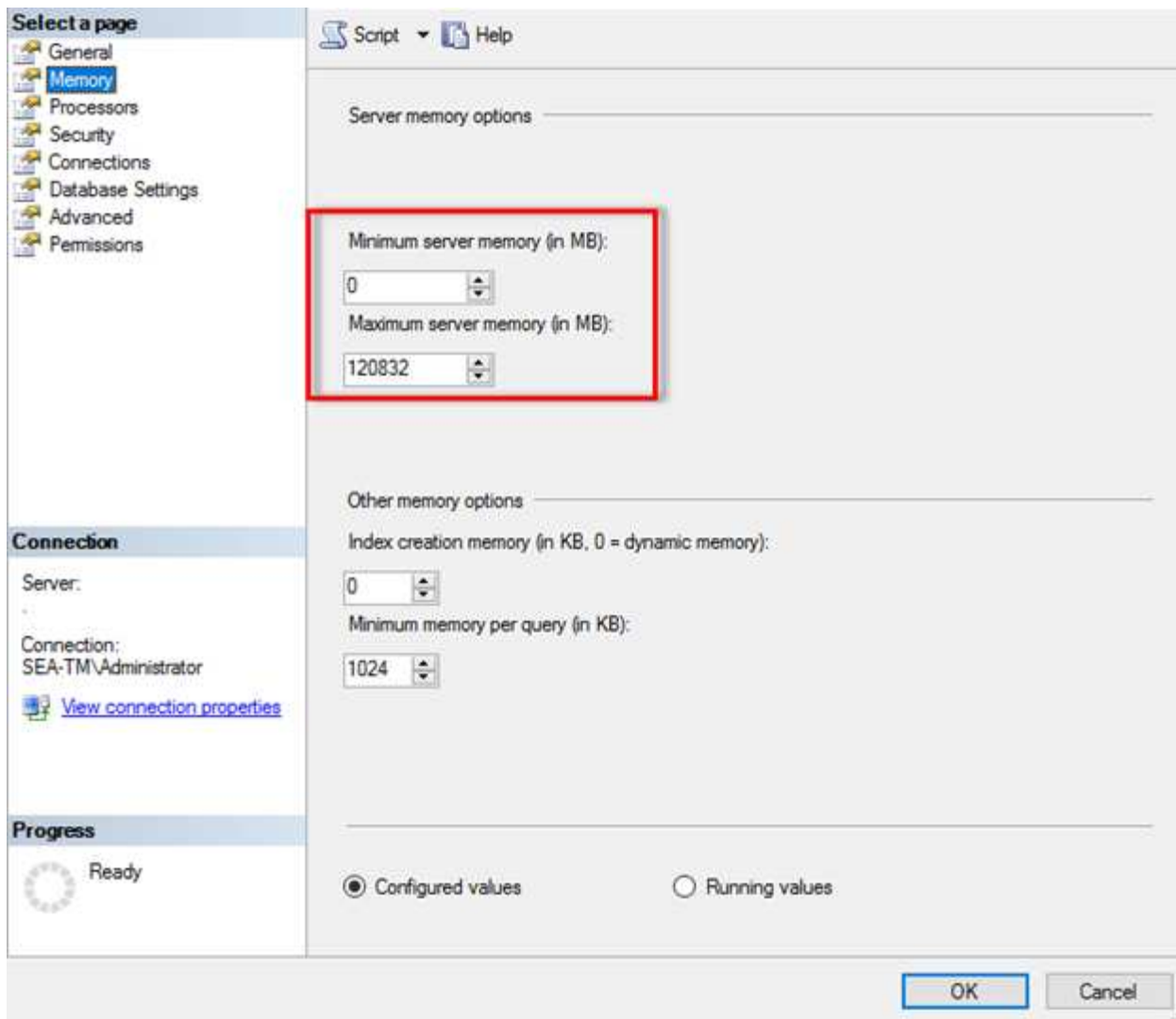
It is generally used if multiple applications are running on the same server where SQL Server is running and you want to guarantee that these applications have sufficient memory to function properly.

Some applications only use whatever memory is available when they start and do not request more even if needed. That is where the max server memory setting comes into play.

On a SQL Server cluster with several SQL Server instances, each instance could be competing for resources. Setting a memory limit for each SQL Server instance can help guarantee best performance for each instance.



NetApp recommends leaving at least 4GB to 6GB of RAM for the operating system to avoid performance issues.



Adjusting minimum and maximum server memory using SQL Server Management Studio.

Using SQL Server Management Studio to adjust minimum or maximum server memory requires a restart of the SQL Server service. You can adjust server memory using transact SQL (T-SQL) using this code:

```
EXECUTE sp_configure 'show advanced options', 1

GO

EXECUTE sp_configure 'min server memory (MB)', 2048

GO

EXEC sp_configure 'max server memory (MB)', 120832

GO

RECONFIGURE WITH OVERRIDE
```

Nonuniform memory access

Nonuniform memory access (NUMA) is a memory-access optimization method that helps increase processor speed without increasing the load on the processor bus.

If NUMA is configured on the server where SQL Server is installed, no additional configuration is required because SQL Server is NUMA aware and performs well on NUMA hardware.

Index create memory

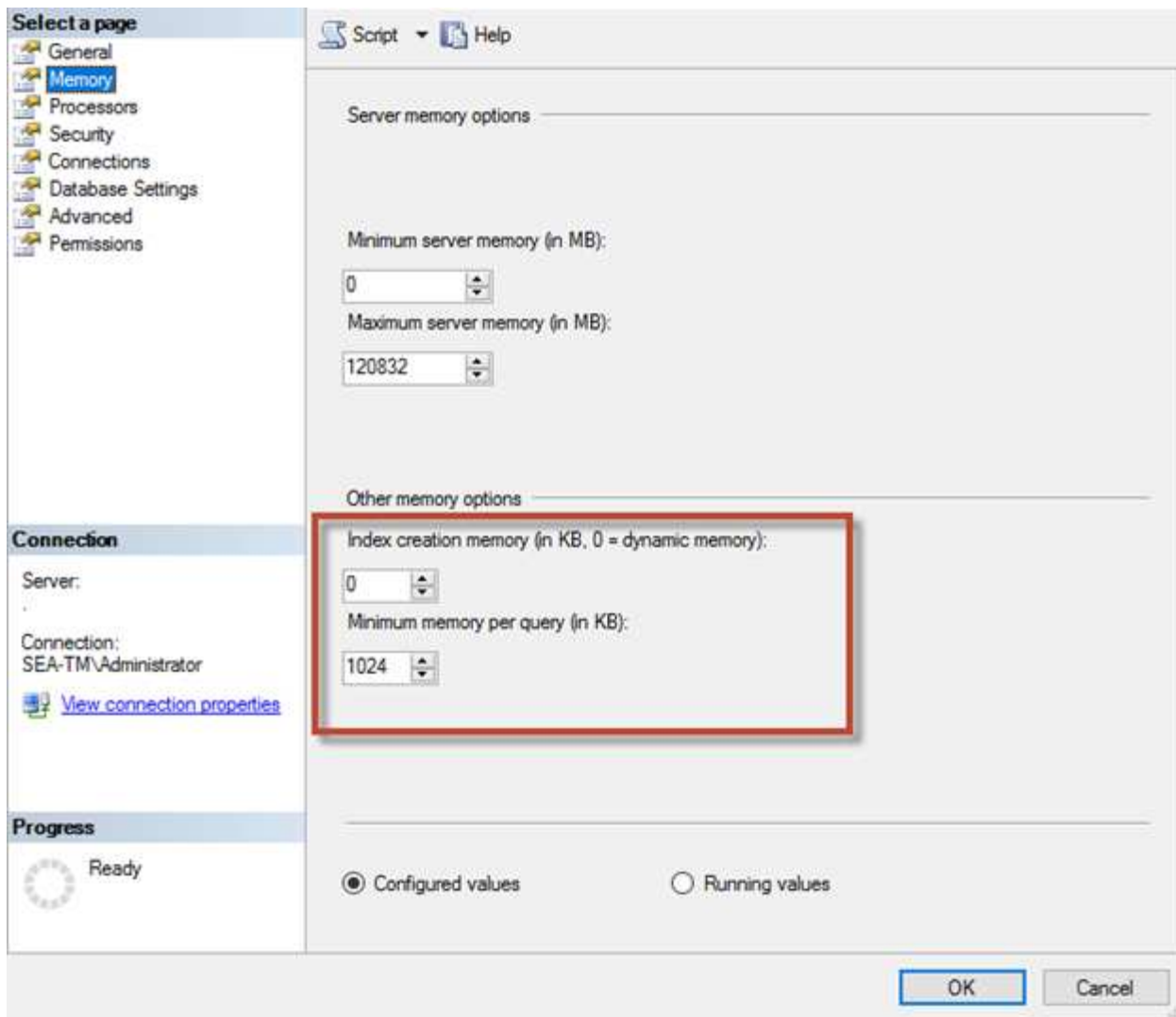
The index create memory option is another advanced option that you should not usually change.

It controls the maximum amount of RAM initially allocated for creating indexes. The default value for this option is 0, which means that it is managed by SQL Server automatically. However, if you encounter difficulties creating indexes, consider increasing the value of this option.

Min memory per query

When a query is run, SQL Server tries to allocate the optimum amount of memory for it to run efficiently.

By default, the min memory per query setting allocates \geq 1024KB for each query to run. It is a best practice is to leave this setting at the default value of 0 to allow SQL Server to dynamically manage the amount of memory allocated for index creation operations. However, if SQL Server has more RAM than it needs to run efficiently, the performance of some queries can be boosted if you increase this setting. Therefore, as long as memory is available on the server that is not being used by SQL Server, any other applications, or the operating system, then boosting this setting can help overall SQL Server performance. If no free memory is available, increasing this setting might hurt overall performance.



Buffer pool extensions

The buffer pool extension provides seamless integration of an NVRAM extension with the database engine buffer pool to significantly improve I/O throughput.

The buffer pool extension is not available in every SQL Server edition. It is available only with the 64-bit SQL Server Standard, Business Intelligence, and Enterprise editions.

The buffer pool extension feature extends the buffer pool cache with nonvolatile storage (usually SSDs). The extension allows the buffer pool to accommodate a larger database working set, forcing the paging of I/O between the RAM and the SSDs and effectively offloading small random I/Os from mechanical disks to SSDs. Because of the lower latency and better random I/O performance of SSDs, the buffer pool extension significantly improves I/O throughput.

The buffer pool extension feature offers the following benefits:

- Increased random I/O throughput
- Reduced I/O latency
- Increased transaction throughput
- Improved read performance with a larger hybrid buffer pool

- A caching architecture that can take advantage of existing and future low-cost memory

NetApp recommends configuring the buffer pool extensions to:



- Make sure that an SSD-backed LUN (such as NetApp AFF) is presented to the SQL Server host so that it can be used as a buffer pool extension target disk.
- The extension file must be the same size as or larger than the buffer pool.

The following example shows a T-SQL command to set up a buffer pool extension of 32GB.

```
USE master

GO

ALTER SERVER CONFIGURATION

SET BUFFER POOL EXTENSION ON

    (FILENAME = 'P:\BUFFER POOL EXTENSION\SQLServerCache.BUFFER POOL
EXTENSION', SIZE = 32 GB);

GO
```

Storage configuration

Microsoft SQL Server database files and filegroups

A SQL Server database is a collection of objects that allows you to store and manipulate data.

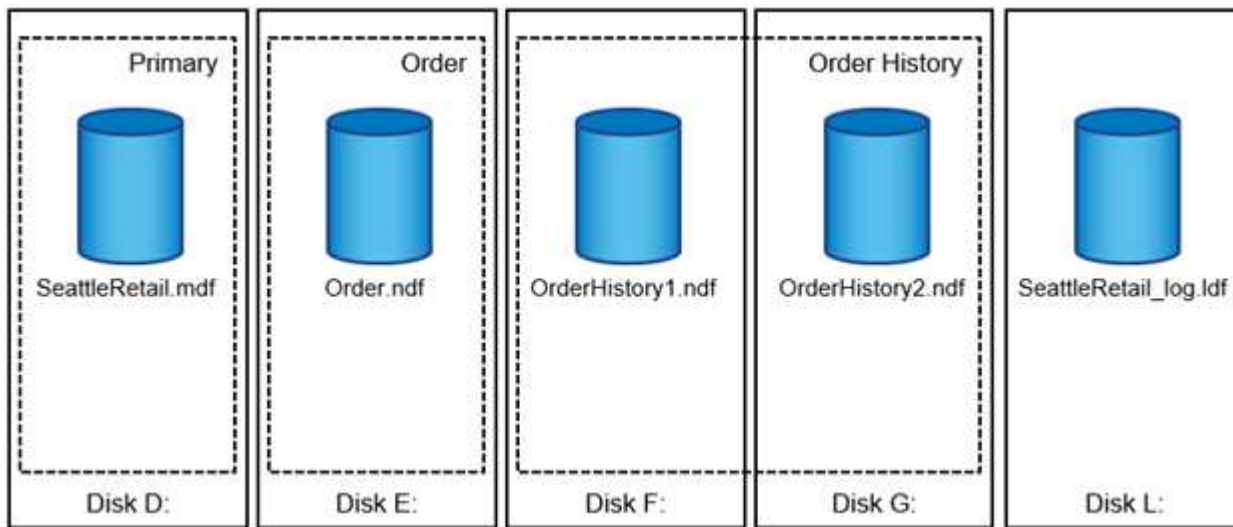
In theory, SQL Server (64-bit) supports 32,767 databases per instance and 524,272TB of database size, although the typical installation usually has several databases. However, the number of the databases SQL Server can handle depends on the load and hardware. It is not unusual to see SQL Server instances hosting dozens, hundreds, or even thousands of small databases.

Each database consists of one or more data files and one or more transaction log files. The transaction log stores the information about database transactions and all data modifications made by each session. Every time the data is modified, SQL Server stores enough information in the transaction log to undo (roll back) or redo (replay) the action. A SQL Server transaction log is an integral part of SQL Server's reputation for data integrity and robustness. The transaction log is vital to the atomicity, consistency, isolation, and durability (ACID) capabilities of SQL Server. SQL Server writes to the transaction log as soon as any change to the data page happens. Every Data Manipulation Language (DML) statement (for example, select, insert, update, or delete) is a complete transaction, and the transaction log makes sure that the entire set-based operation takes place, making sure of the atomicity of the transaction.

Each database has one primary data file, which, by default, has the .mdf extension. In addition, each database can have secondary database files. Those files, by default, have .ndf extensions.

All database files are grouped into filegroups. A filegroup is the logical unit, which simplifies database

administration. They allow the separation between logical object placement and physical database files. When you create the database objects tables, you specify in what filegroup they should be placed without worrying about the underlying data file configuration.



The ability to put multiple data files inside the filegroup enables you to spread the load across different storage devices, which helps to improve the I/O performance of the system. The transaction log in contrast does not benefit from the multiple files because SQL Server writes to the transaction log in a sequential manner.

The separation between logical object placement in the filegroups and physical database files allows you to fine-tune the database file layout, getting the most from the storage subsystem. For example, independent software vendors (ISVs) who are deploying their products to different customers can adjust the number of database files based on the underlying I/O configuration and the expected amount of data during the deployment stage. Those changes are transparent to the application developers, who are placing the database objects in the filegroups rather than database files.



NetApp recommends avoiding the use of the primary filegroup for anything but system objects. Creating a separate filegroup or set of filegroups for the user objects simplifies database administration and disaster recovery, especially in the case of large databases.

You can specify initial file size and autogrowth parameters at the time when you create the database or add new files to an existing database. SQL Server uses a proportional fill algorithm when choosing which data file it should write data into. It writes an amount of data proportionally to the free space available in the files. The more free space in the file, the more writes it handles.



NetApp recommends that all files in the single filegroup have the same initial size and autogrowth parameters, with the grow size defined in megabytes rather than percentages. This helps the proportional fill algorithm evenly balance write activities across data files.

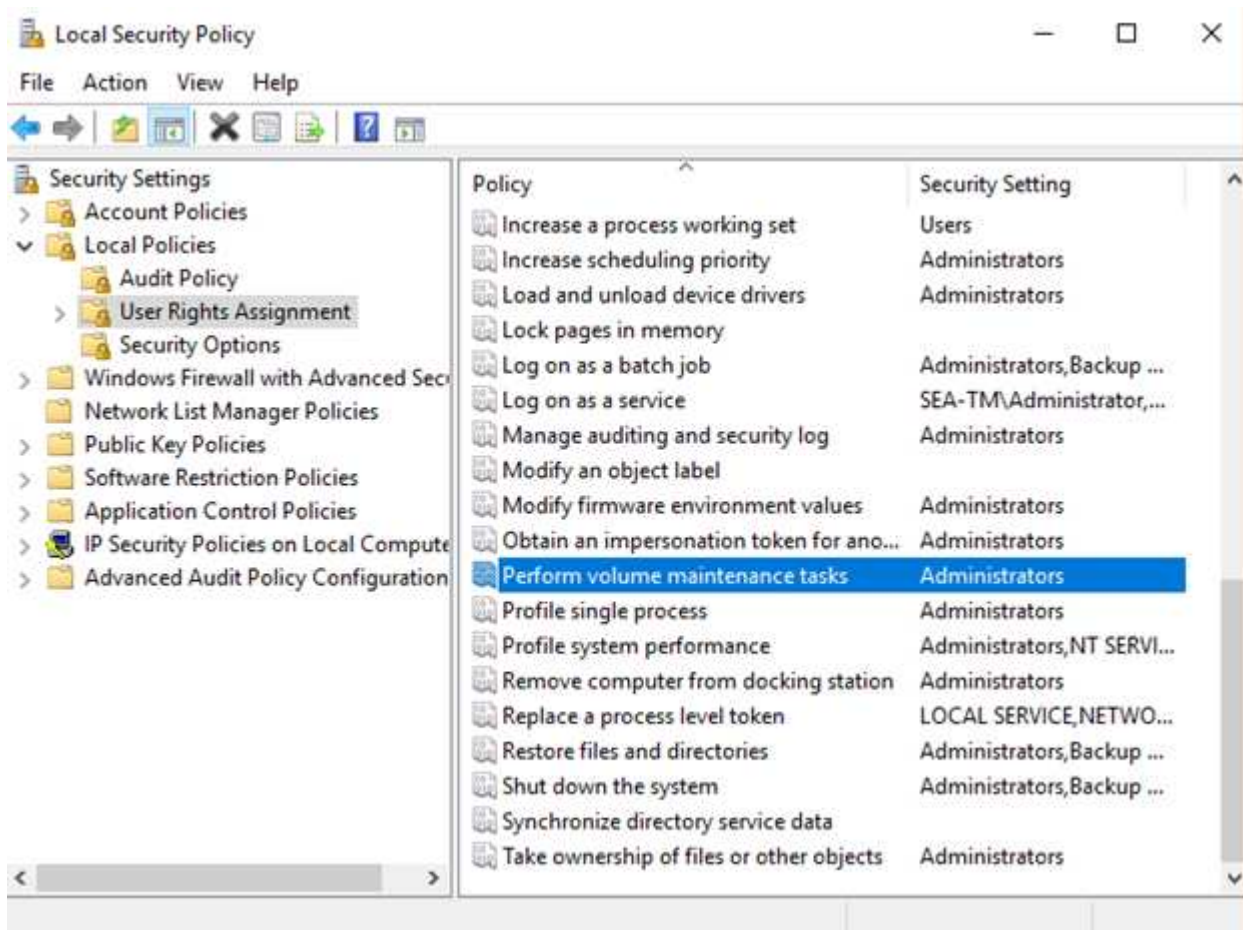
Every time SQL Server grows the files, it fills newly allocated space in the files with zeros. That process blocks all sessions that need to write to the corresponding file or, in case of transaction log growth, generate transaction log records.

SQL Server always zeroes out the transaction log, and that behavior cannot be changed. However, you can control whether data files are zeroing out by enabling or disabling instant file initialization. Enabling instant file initialization helps to speed up data file growth and reduces the time required to create or restore the database.

A small security risk is associated with instant file initialization. When this option is enabled, unallocated parts

of the data file can contain information from previously deleted OS files. Database administrators can examine such data.

You can enable instant file initialization by adding the SA_MANAGE_VOLUME_NAME permission, also known as “perform volume maintenance task,” to the SQL Server startup account. You can do this under the local security policy management application (secpol.msc), as shown in the following figure. Open the properties for the “perform volume maintenance task” permission and add the SQL Server startup account to the list of users there.



To check if the permission is enabled, you can use the code from the following example. This code sets two trace flags that force SQL Server to write additional information to the error log, create a small database, and read the content of the log.

```

DBCC TRACEON(3004,3605,-1)

GO

CREATE DATABASE DelMe

GO

EXECUTE sp_readerrorlog

GO

DROP DATABASE DelMe

GO

DBCC TRACEOFF(3004,3605,-1)

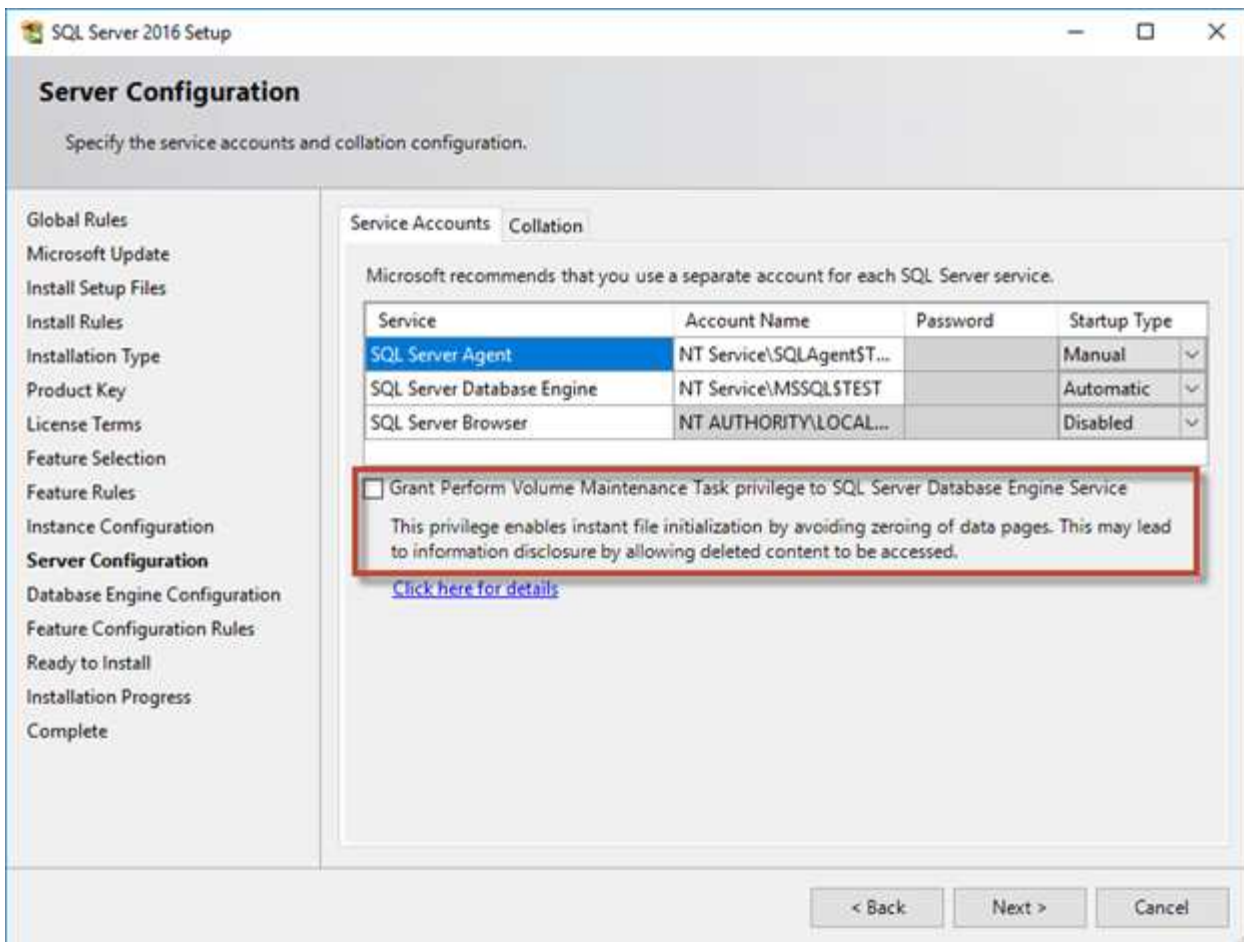
GO

```

When instant file initialization is not enabled, the SQL Server error log shows that SQL Server is zeroing the mdf data file in addition to zeroing the ldf log file, as shown in the following example. When instant file initialization is enabled, it displays only zeroing of the log file.

	LogDate	ProcessInfo	Text
365	2017-02-09 08:10:07.660	spid53	Ckpt dbid 3 flush delta counts.
366	2017-02-09 08:10:07.660	spid53	Ckpt dbid 3 logging active xact info.
367	2017-02-09 08:10:07.750	spid53	Ckpt dbid 3 phase 1 ended (8)
368	2017-02-09 08:10:07.750	spid53	About to log Checkpoint end.
369	2017-02-09 08:10:07.880	spid53	Ckpt dbid 3 complete
370	2017-02-09 08:10:08.130	spid53	Starting up database 'DelMe'.
371	2017-02-09 08:10:08.150	spid53	FixupLog Tail(progress) zeroing C:\Program Files\Microsoft SQL Server\MSSQL
372	2017-02-09 08:10:08.160	spid53	Zeroing C:\Program Files\Microsoft SQL Server\MSSQL
373	2017-02-09 08:10:08.170	spid53	Zeroing completed on C:\Program Files\Microsoft SQL
374	2017-02-09 08:10:08.710	spid53	Ckpt dbid 6 started
375	2017-02-09 08:10:08.710	spid53	About to log Checkpoint begin.

The perform volume maintenance task is simplified in SQL Server 2016 and is later provided as an option during the installation process. This figure displays the option to grant the SQL Server database engine service the privilege to perform the volume maintenance task.



Another important database option that controls the database file sizes is autoshrink. When this option is enabled, SQL Server regularly shrinks the database files, reduces their size, and releases space to the operating system. This operation is resource intensive and is rarely useful because the database files grow again after some time when new data comes into the system. Autoshrink must never be enabled on the database.

Microsoft SQL Server tempdb files

NetApp recommends proactively inflating tempdb files to their full size to avoid disk fragmentation.

Page contention can occur on lobal allocation map (GAM), shared global allocation map (SGAM), or page free space (PFS) pages when SQL Server must write to special system pages to allocate new objects. Latches protect (lock) these pages in memory. On a busy SQL Server instance, it can take a long time to get a latch on a system page in tempdb. This results in slower query run times and is known as latch contention. See the following best practices for creating tempdb data files:

- For ≤ 8 cores: tempdb data files = number of cores
- For > 8 cores: 8 tempdb data files

The following example script modifies tempdb by creating eight tempdb files and moving tempdb to the mount point C:\MSSQL\tempdb for SQL Server 2012 and later.

```
use master
```

```

go

-- Change logical tempdb file name first since SQL Server shipped with
logical file name called tempdev

alter database tempdb modify file (name = 'tempdev', newname =
'tempdev01');

-- Change location of tempdev01 and log file

alter database tempdb modify file (name = 'tempdev01', filename =
'C:\MSSQL\tempdb\tempdev01.mdf');

alter database tempdb modify file (name = 'templog', filename =
'C:\MSSQL\tempdb\templog.ldf');

GO

-- Assign proper size for tempdev01

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'tempdev01', SIZE = 10GB );

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'templog', SIZE = 10GB );

GO

-- Add more tempdb files

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev02', FILENAME =
N'C:\MSSQL\tempdb\tempdev02.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev03', FILENAME =
N'C:\MSSQL\tempdb\tempdev03.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev04', FILENAME =
N'C:\MSSQL\tempdb\tempdev04.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev05', FILENAME =
N'C:\MSSQL\tempdb\tempdev05.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev06', FILENAME =
N'C:\MSSQL\tempdb\tempdev06.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev07', FILENAME =

```

```
N'C:\MSSQL\tempdb\tempdev07.ndf' , SIZE = 10GB , FILEGROWTH = 10%);  
  
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev08', FILENAME =  
N'C:\MSSQL\tempdb\tempdev08.ndf' , SIZE = 10GB , FILEGROWTH = 10%);  
  
GO
```

Beginning with SQL Server 2016, the number of CPU cores visible to the operating system is automatically detected during installation and, based on that number, SQL Server calculates and configures the number of tempdb files required for optimum performance.

Microsoft SQL Server storage considerations

The combination of NetApp storage solutions and Microsoft SQL Server enables the creation of enterprise-level database storage designs that can meet today's most demanding application requirements.

To optimize both technologies, it is vital to understand the SQL Server I/O pattern and characteristics. A well-designed storage layout for a SQL Server database supports the performance of SQL Server and the management of the SQL Server infrastructure. A good storage layout also allows the initial deployment to be successful and the environment to grow smoothly over time as the business grows.

Data storage design

For SQL Server databases that do not use SnapCenter to perform backups, Microsoft recommends placing the data and log files on separate drives. For applications that simultaneously update and request data, the log file is write intensive, and the data file (depending on your application) is read/write intensive. For data retrieval, the log file is not needed. Therefore, requests for data can be satisfied from the data file placed on its own drive.

When you create a new database, Microsoft recommends specifying separate drives for the data and logs. To move files after the database is created, the database must be taken offline. For more Microsoft recommendations, see [Place Data and Log Files on Separate Drives](#).

Aggregates

Aggregates are the primary storage containers for NetApp storage configurations and contain one or more RAID groups consisting of both data disks and parity disks. NetApp has performed various I/O workload characterization tests using shared and dedicated aggregates with data files and transaction log files separated. The tests show that one large aggregate with more RAID groups and spindles optimizes and improves storage performance and is easier for administrators to manage for two reasons:

- One large aggregate makes the I/O capabilities of all spindles available to all files.
- One large aggregate enables the most efficient use of disk space.

For high availability (HA), place the SQL Server Always On Availability Group secondary synchronous replica on a separate storage virtual machine (SVM) in the aggregate. For disaster recovery purposes, place the asynchronous replica on an aggregate that is part of a separate storage cluster in the DR site, with content replicated by using NetApp SnapMirror technology. NetApp recommends having at least 10% free space available in an aggregate for optimal storage performance.

Volumes

NetApp FlexVol volumes are created and reside inside aggregates. Many volumes can be created in a single aggregate, and each volume can be expanded, shrunk, or moved between aggregates with no user downtime.

Volume design considerations

Before you create a database volume design, it is important to understand how the SQL Server I/O pattern and characteristics vary depending on the workload and on the backup and recovery requirements. See the following NetApp recommendations for flexible volumes:

- Use flexible volumes to store SQL Server database files and avoid sharing volumes between hosts.
- Use NTFS mount points instead of drive letters to surpass the 26-drive-letter limitation in Windows. When using volume mount points, it is a general recommendation to give the volume label the same name as the mount point.
- When appropriate, configure a volume autosize policy to help prevent out-of-space conditions. 17 Best practice guide for Microsoft SQL Server with ONTAP © 2022 NetApp, Inc. All rights reserved.
- Enable read reallocation on the volume when the SQL Server database I/O profile consists of mostly large sequential reads, such as with decision support system workloads. Read reallocation optimizes the blocks to provide better performance.
- If you install SQL Server on an SMB share, make sure that Unicode is enabled on the SMB/CIFS volumes for creating folders.
- Set the NetApp snapshot copy reserve value in the volume to zero for ease of monitoring from an operational perspective.
- Disable storage Snapshot™ copy schedules and retention policies. Instead, use SnapCenter to coordinate Snapshot copies of the SQL Server data volumes.
- Place the SQL Server system databases on a dedicated volume or VMDK.
- tempdb is a system database used by SQL Server as a temporary workspace, especially for I/O intensive DBCC CHECKDB operations. Therefore, place this database on a dedicated volume with a separate set of spindles. In large environments in which volume count is a challenge, you can consolidate tempdb into fewer volumes and store it in the same volume as other system databases after careful planning. Data protection for tempdb is not a high priority because this database is recreated every time SQL Server is restarted.
- Place user data files (.mdf) on separate volumes because they are random read/write workloads. It is common to create transaction log backups more frequently than database backups. For this reason, place transaction log files (.ldf) on a separate volume or VMDK from the data files so that independent backup schedules can be created for each. This separation also isolates the sequential write I/O of the log files from the random read/write I/O of data files and significantly improves SQL Server performance.

LUN

- Make sure that the user database files and the log directory to store log backup are on separate volumes to prevent the retention policy from overwriting snapshots when these are used with SnapVault technology.
- Make sure that SQL Server databases reside on LUNs separate from LUNs that have non-database files, such as full-text search-related files.
- Placing database secondary files (as part of a filegroup) on separate volumes improves the performance of the SQL Server database. This separation is valid only if the database's .mdf file does not share its LUN with any other .mdf files.
- If you create LUNs with DiskManager or other tools, make sure that the allocation unit size is set to 64K for

partitions when formatting the LUNs.

- See the [Microsoft Windows and native MPIO under ONTAP best practices for modern SAN](#) to apply multipathing support on Windows to iSCSI devices in the MPIO properties.

Log directory

Log directory is specified in SQL Server to store transaction log backup data at the host level. If you are using SnapCenter to backup log files then each SQL Server host used by SnapCenter must have a host log directory configured to perform log backups. SnapCenter has a database repository, so metadata related to backup, restore, or cloning operations is stored in a central database repository.

The sizes of the host log directory is calculated as follows:

Size of host log directory = ((maximum DB LDF size x daily log change rate %) x (snapshot retention) ÷ (1 - LUN overhead space %)

The host log directory sizing formula assumes a 10% LUN overhead space

Place the log directory on a dedicated volume or LUN. The amount of data in the host log directory depends on the size of the backups and the number of days that backups are retained. SnapCenter allows only one host log directory per SQL Server host. You can configure the host log directories at SnapCenter → Host → Configure Plug-in.

NetApp recommends the following for a host log directory:

- Make sure that the host log directory is not shared by any other type of data that can potentially corrupt the backup snapshot data.
- Do not place user databases or system databases on a LUN that hosts mount points.
- Create the host log directory on the dedicated FlexVol volume to which SnapCenter copies transaction logs.
- Use SnapCenter wizards to migrate databases to NetApp storage so that the databases are stored in valid locations, enabling successful SnapCenter backup and restore operations. Keep in mind that the migration process is disruptive and can cause the databases to go offline while the migration is in progress.
- The following conditions must be in place for failover cluster instances (FCIs) of SQL Server:
 - If you are using a failover cluster instance, the host log directory LUN must be a cluster disk resource in the same cluster group as the SQL Server instance being backed up SnapCenter.
 - If you are using a failover cluster instance, user databases must be placed on shared LUNs that are physical disk cluster resources assigned to the cluster group associated with the SQL Server instance.



ONTAP Storage efficiency with Microsoft SQL Server

Storage efficiency is the ability to store and manage SQL Server data in a way that consumes the least amount of storage space with little or no effect on the overall performance of the system.

Storage efficiency is a combination of RAID, provisioning (overall layout and utilization), mirroring, and other data protection technologies. NetApp technologies like Snapshot copies, Thin provisioning, FlexClone helps in creating cost benefits by optimizing existing storage in the infrastructure and deferring or avoiding future storage expenditures. The more you use these technologies together, the larger the savings.

Thin provisioning

Thin provisioning comes in many forms and is integral to many features that ONTAP offers to an enterprise application environment. Thin provisioning is also closely related to efficiency technologies for the same reason: efficiency features allow more logical data to be stored than what technically exists on the storage system.

Almost any use of snapshots involves thin provisioning. For example, a typical 10TB database on NetApp storage includes around 30 days of snapshots. This arrangement results in approximately 10TB of data visible in the active file system and 300TB dedicated to snapshots. The total 310TB of storage usually resides on approximately 12TB to 15TB of space. The active database consumes 10TB, and the remaining 300TB of data only requires 2TB to 5TB of space because only the changes to the original data are stored.

Cloning is also an example of thin provisioning. A major NetApp customer created 40 clones of an 80TB database for use by development. If all 40 developers using these clones overwrote every block in every datafile, over 3.2PB of storage would be required. In practice, turnover is low, and the collective space requirement is closer to 40TB because only changes are stored on the drives.

Space management

Some care must be taken with thin provisioning an application environment because data change rates can increase unexpectedly. For example, space consumption due to snapshots can grow rapidly if database tables are reindexed, or wide-scale patching is applied to VMware guests. A misplaced backup can write a large amount of data in a very short time. Finally, it can be difficult to recover some applications if a file system runs out of free space unexpectedly.

Fortunately, these risks can be addressed with careful configuration of `volume-autogrow` and `snapshot-autodelete` policies. As their names imply, these options enable a user to create policies that automatically clear space consumed by snapshots or grow a volume to accommodate additional data. Many options are available and needs vary by customer.

See the [logical storage management documentation](#) for a complete discussion of these features.

LUN thin provisioning

The efficiency of thin provisioning of active LUNs in a file system environment can be lost over time as data is deleted. Unless that deleted data is either overwritten with zeros or the space is released with TRIM/UNMAP space reclamation, the "erased" data occupies more and more unallocated whitespace in the file system. Furthermore, thin provisioning of active LUNs is of limited use in many database environments because datafiles are initialized to their full size at the time of creation.

Careful planning of LVM configuration can improve efficiency and minimize the need for storage provisioning and LUN resizing. When an LVM such as Veritas VxVM or Oracle ASM is used, the underlying LUNs are divided into extents that are only used when needed. For example, if a dataset begins at 2TB in size but could grow to 10TB over time, this dataset could be placed on 10TB of thin-provisioned LUNs organized in an LVM diskgroup. It would occupy only 2TB of space at the time of creation and would only claim additional space as extents are allocated to accommodate data growth. This process is safe as long as space is monitored.

Fractional reservations

Fractional reserve refers to the behavior of a LUN in a volume with respect to space efficiency. When the option `fractional-reserve` is set to 100%, all data in the volume can experience 100% turnover with any data pattern without exhausting space on the volume.

For example, consider a database on a single 250GB LUN in a 1TB volume. Creating a snapshot would

immediately result in the reservation of an additional 250GB of space in the volume to guarantee that the volume does not run out of space for any reason. Using fractional reserves is generally wasteful because it is extremely unlikely that every byte in the database volume would need to be overwritten. There is no reason to reserve space for an event that never happens. Still, if a customer cannot monitor space consumption in a storage system and must be certain that space never runs out, 100% fractional reservations would be required to use snapshots.

Compression and deduplication

Compression and deduplication are both forms of thin provisioning. For example, a 50TB data footprint might compress to 30TB, resulting in a savings of 20TB. For compression to yield any benefits, some of that 20TB must be used for other data, or the storage system must be purchased with less than 50TB. The result is storing more data than is technically available on the storage system. From the data point of view, there is 50TB of data, even though it occupies only 30TB on the drives.

There is always a possibility that the compressibility of a dataset changes, which would result in increased consumption of real space. This increase in consumption means that compression must be managed as with other forms of thin provisioning in terms of monitoring and using `volume-autogrow` and `snapshot-autodelete`.

Compression and deduplication are discussed in further detail in the section [xref:./mssql/efficiency.html](#)

Compression and fractional reservations

Compression is a form of thin provisioning. Fractional reservations affect the use of compression, with one important note; space is reserved in advance of the snapshot creation. Normally, fractional reserve is only important if a snapshot exists. If there is no snapshot, fractional reserve is not important. This is not the case with compression. If a LUN is created on a volume with compression, ONTAP preserves space to accommodate a snapshot. This behavior can be confusing during configuration, but it is expected.

As an example, consider a 10GB volume with a 5GB LUN that has been compressed down to 2.5GB with no snapshots. Consider these two scenarios:

- Fractional reserve = 100 results in 7.5GB utilization
- Fractional reserve = 0 results in 2.5GB utilization

The first scenario includes 2.5GB of space consumption for current data and 5GB of space to account for 100% turnover of the source in anticipation of snapshot use. The second scenario reserves no extra space.

Although this situation might seem confusing, it is unlikely to be encountered in practice. Compression implies thin provisioning, and thin provisioning in a LUN environment requires fractional reservations. It is always possible for compressed data to be overwritten by something uncompressible, which means a volume must be thin provisioned for compression to result in any savings.

NetApp recommends the following reserve configurations:



- Set `fractional-reserve` to 0 when basic capacity monitoring is in place along with `volume-autogrow` and `snapshot-autodelete`.
- Set `fractional-reserve` to 100 if there is no monitoring ability or if it is impossible to exhaust space under any circumstance.

Efficiency

Space efficiency features, such as compression, compaction, and deduplication are designed to increase the amount of logical data that fits on a given amount of physical storage. The result is lower costs and management overhead.

At a high level, compression is a mathematical process whereby patterns in data are detected and encoded in a way that reduces space requirements. In contrast, deduplication detects actual repeated blocks of data and removes the extraneous copies. Compaction allows multiple logical blocks of data to share the same physical block on media.



See the sections below on thin provisioning for an explanation of the interaction between storage efficiency and fractional reservation.

SQL Server also have feature to compress and efficiently manage data. SQL Server currently supports two types of data compression: row compression and page compression.

Row compression changes the data storage format. For example, it changes integers and decimals to the variable-length format instead of their native fixed-length format. It also changes fixed-length character strings to the variable-length format by eliminating blank spaces. Page compression implements row compression and two other compression strategies (prefix compression and dictionary compression). You can find more details about page compression in [Page Compression Implementation](#).

Data compression is currently supported in the Enterprise, Developer, and Evaluation editions of SQL Server 2008 and later. Although compression can be performed by the database itself, this is rarely observed in a SQL Server environment.

Here are the recommendation for managing space for SQL Server data files

- Use thin provisioning in SQL Server environments to improve space utilization and to reduce the overall storage requirements when the space guarantee functionality is used.
- Use autogrow for most common deployment configurations because the storage admin only needs to monitor space usage in the aggregate.
- Advice not to enable deduplication on any volumes containing SQL Server data files unless the volume is known to contain multiple copies of the same data, such as restoring database from backups to a single volume.

Space reclamation

Space reclamation can be initiated periodically to recover unused space in a LUN. With SnapCenter, you can use the following PowerShell command to start space reclamation.

```
Invoke-SdHostVolumeSpaceReclaim -Path drive_path
```

If you need to run space reclamation, this process should be run during periods of low activity because it initially consumes cycles on the host.

Microsoft SQL Server data protection with ONTAP

Protecting database is critical to any organization. As data size and count of databases have increased, maintaining recovery time objective (RTO) and recovery point objective

(RPO) is very crucial.

The main consideration for database backup is leveraging NetApp Snapshot technology. Application consistent backup and database layout are other key consideration to achieve RTO and RPO which can be orchestrated by NetApp SnapCenter.

SnapCenter

SnapCenter is the NetApp data protection software for enterprise applications. SQL Server databases can be quickly and easily protected using NetApp SnapCenter software with the Plug-in for SQL Server and the Plug-in for Microsoft Windows.

This products enable application-consistent backup, automated cloning, and restore and recovery of SQL Server databases, instances, or availability groups.



NetApp recommends using SnapCenter to create Snapshot copies.

For more details about the SQL Server Plug-in for SnapCenter, see [TR-4714: Best practice guide for SQL Server using NetApp SnapCenter](#).

Protecting database using T-SQL snapshots

In SQL Server 2022, Microsoft introduced T-SQL snapshots that offers built-in advantage than traditional method that was not easily consumed by database administrator. Leveraging ONTAP REST API's, you can call commands to snapshot volumes.

The following is a sample backup workflow:

1. Freeze a database with ALTER command - providing an opportunity to perform a consistent snapshot on the underlying storage. After this you can thaw the database and record the snapshot with BACKUP command.
2. Perform snapshots of multiple databases on the storage volumes simultaneously with the new BACKUP GROUP and BACKUP SERVER commands.
3. Perform FULL backups or COPY_ONLY FULL backups. These backups are recorded in msdb as well.
4. Perform point-in-time recovery using log backups taken with the normal streaming approach after the snapshot FULL backup. Streaming differential backups are also supported if desired.

To learn more, see [Microsoft documentation to know about the T-SQL snapshots](#).

Microsoft SQL Server disaster recovery

NetApp provides various approaches to increase data availability if there is hardware, software or site failure.

NetApp SnapMirror

NetApp SnapMirror technology offers a fast and flexible enterprise solution for mirroring or replicating data over LANs and WANs. SnapMirror technology transfers only modified 4KB data blocks to the destination after the initial base transfer, significantly reducing network bandwidth requirements. SnapMirror provides asynchronous volume-level replication that is based on a configured replication update interval.

The following are recommendations for SnapMirror for SQL Server:

- If CIFS is used, the destination SVM must be a member of the same Active Directory domain of which the source SVM is a member so that the access control lists (ACLs) stored within NAS files are not broken during recovery from a disaster.
- Using destination volume names that are the same as the source volume names is not required but can make the process of mounting destination volumes into the destination simpler to manage. If CIFS is used, you must make the destination NAS namespace identical in paths and directory structure to the source namespace.
- For consistency purposes, do not schedule SnapMirror update from the controllers. However, enable SnapMirror update from SnapCenter to update SnapMirror after either full or log backup is completed.
- Distribute volumes that contain SQL Server data across different nodes in the cluster to allow all cluster nodes to share SnapMirror replication activity. This distribution optimizes the use of node resources.

For more information about SnapMirror, see [TR-4015: SnapMirror Configuration and Best Practices Guide for ONTAP 9](#).

Database security

Securing a database environment is a multidimensional effort that goes beyond managing the database itself. NetApp offers several unique features designed to secure the storage aspect of your database infrastructure.

Snapshot copies

Storage snapshots are point-in-time replicas of the target data. ONTAP's implementation includes the abilities to set various policies and store up to 1024 snapshots per volume. Snapshots in ONTAP are space-efficient. Space is only consumed as the original dataset changes. They are also read-only. A snapshot can be deleted, but it cannot be changed.

In some cases, snapshots can be scheduled directly on ONTAP. In other cases, software such as SnapCenter may be required to orchestrate application or OS operations before creating snapshots. Whichever approach is best for your workload, an aggressive snapshot strategy can provide data security through frequent, easily-accessible access to backups of everything from boot LUNs to mission-critical databases.

Note: An ONTAP Flexible Volume, or more simply, a volume is not synonymous with a LUN. Volumes are management containers for data such as files or LUNs. For example, a database might be placed on an 8-LUN stripe set, with all LUNs contained in a single volume.

For more information on snapshots, click [here](#).

Tamperproof snapshots

Beginning with ONTAP 9.12.1, snapshots are not just read-only, they can also be protected from accidental or intentional deletion. The feature is called Tamperproof Snapshots. A retention period can be set and enforced via snapshot policy. The resulting snapshots cannot be deleted until they have reached their expiration date. There are no administrative or support center overrides.

This ensures that an intruder, a malicious insider, or even a ransomware attack is unable to compromise the backups even if it resulted in access to the ONTAP system itself. When combined with a frequent snapshot schedule, the result is extremely powerful data protection with a very low RPO.

For more information on Tamperproof Snapshots, click [here](#).

SnapMirror replication

Snapshots can also be replicated to a remote system. This includes Tamperproof Snapshots, where the retention period is applied and enforced on the remote system. The result is the same data protection benefits as local snapshots, but the data is located on a second storage array. This ensures that destruction of the original array does not compromise the backups.

A second system also opens new options for administrative security. For example, some NetApp customers segregate authentication credentials for the primary and secondary storage systems. No single administrative user has access to both systems, which means a malicious administrator cannot delete all copies of data.

For more information on SnapMirror, click [here](#).

Storage Virtual Machines

A newly configured ONTAP storage system is similar to a newly provisioned VMware ESX server because neither of them can support any users until a virtual machine is created. With ONTAP, you create a Storage Virtual Machine (SVM) which becomes the most basic unit of storage management. Each SVM has its own storage resources, protocol configurations, IP addresses, and FCP WWNs. This is the foundation of ONTAP mult-tenancy.

For example, you might configure one SVM for critical production workloads, and a second SVM on a different network segment for development activities. You could then restrict access to the production SVM to certain administrators, while granting developers more expansive control over the storage resources in the development SVM. You might also need to provide a third SVM to your financial and HR teams to store especially critical eyes-only data.

For more information about SVMs, click [here](#).

Administrative RBAC

ONTAP offers powerful role-based access control (RBAC) for administrative logins. Some admins might need full cluster access, while others might only need access to certain SVMs. Advanced helpdesk personnel might need the ability to increase volumes sizes. The result is you can grant administrative users the access required to perform their job responsibilities, and nothing more. Furthermore, you can secure these logins using PKI from various vendors, restrict access to ssh keys only, and enforce failed login attempt lockouts.

For more information on administrative access control, click [here](#).

Multifactor authentication

ONTAP and certain other NetApp products now support multifactor authentication using a variety of methods. The result is a compromised username/password alone is not a security thread without the data from the second factor, such as a FOB or a smartphone app.

For more information, click [here](#).

API RBAC

Automation requires API calls, but not all tools require full administrative access. To help secure automation systems, RBAC is also available at the API level. You can limit the automation user accounts to the API calls required. For example, monitoring software does not need change access, it only requires read access. Workflows that provision storage do not need the ability to delete storage.

To learn more, start https://docs.netapp.com/us-en/ontap-automation/rest/rbac_overview.html[\[here.\]](#)

Multi-admin verification

Multi "factor" authentication can be taken even further by requiring two different administrators, each with their own credentials, to approve certain activities. This includes changing login permissions, running diagnostic commands, and deleting data.

For more information on multi-admin verification (MAV), click [here](#)

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.