



Technical Report

Best Practices Guide for Microsoft SQL Server with ONTAP

Pat Sinthusan, NetApp
March 2019 | TR-4590

Abstract

This best practices guide enables storage administrators and database administrators to successfully deploy Microsoft SQL Server on NetApp® storage.

TABLE OF CONTENTS

1	Introduction	4
1.1	Purpose and Scope	4
1.2	Intended Audience	4
2	SQL Server Workload Types	4
2.1	Testing Benchmark	5
3	SQL Server Configuration	5
3.1	Shared Instance Versus Dedicated Instance	5
3.2	CPU Configuration	6
3.3	Memory Configuration	7
4	Database Files and Filegroups	12
5	Database and Storage	16
5.1	Aggregates	16
5.2	Volumes	16
5.3	LUNs	17
5.4	SMB Shares	19
5.5	Data Storage Design	20
5.6	Virtualization	23
6	Storage Efficiency and Manageability	27
6.1	Snapshot Copies	27
6.2	Thin Provisioning	28
6.3	Space Guarantee	28
6.4	Space Reclamation	29
6.5	Fractional Reserve	29
6.6	NetApp FlexClone	30
6.7	Compression, Compaction, and Deduplication	30
6.8	NetApp SnapMirror	32
7	Data Fabric and SQL Server	33
7.1	Introduction	33
7.2	Cloud Volumes ONTAP	34
7.3	Cloud Volumes Service	37
8	Conclusion	43

Appendix.....	43
----------------------	-----------

Where to Find Additional Information	50
---	-----------

LIST OF TABLES

Table 1) Typical SQL Server workloads.	5
Table 2) Volume guarantee set to none.	29
Table 3) Setting up volume with autodelete and autogrow.	30
Table 4) Example of quotas and service level for workload with 16,000 IOPS.	41

LIST OF FIGURES

Figure 1) Log entries indicate number of cores being used after SQL Server startup.	6
Figure 2) Adjusting minimum and maximum server memory using SQL Server Management Studio.	8
Figure 3) Configuring max worker threads using SQL Server Management Studio.	9
Figure 4) Configuring index create memory and min memory per query using SQL Server Management Studio.....	11
Figure 5) Filegroups.	13
Figure 6) Option for granting perform volume maintenance task privilege during SQL Server installation.	15
Figure 7) Basic SQL Server database design for NetApp storage for SMSQL or SnapCenter.....	21
Figure 8) SQL Server database design for NetApp storage using SMSQL or SnapCenter.	22
Figure 9) Server database design for NetApp storage using SMSQL or SnapCenter.	23
Figure 10) Example of simple database layout on VMDKs with VMFS or NFS datastores.	26
Figure 11) Example of database layout on VMDKs with VMFS or NFS datastores.....	27
Figure 12) NetApp Data Fabric – helping you to transform seamlessly to a next-generation data center.	34
Figure 13) Assign _SQLAdmin account, which is SQL Server Service Account, to have full access to the volumes...38	
Figure 14) Example of deploying SQL Server data and log files to Cloud Volumes Service with SMB protocol.	39
Figure 15) SQL Server Always On with Cloud Volumes Service.	42

1 Introduction

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. 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.

1.1 Purpose and Scope

This document 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
 - SnapManager® for SQL Server as backup software, which includes:
 - SnapDrive for Windows
 - SnapManager for SQL Server
- Microsoft SQL Server architecture and administration

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

1.2 Intended Audience

This technical report is intended for NetApp customers, partners, employees, and field personnel who are responsible for deploying a SQL Server database solution in a customer environment. NetApp assumes that the reader is familiar with the various components of the listed solution previously.

2 SQL Server Workload Types

The SQL Server database platform can support several 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 workload 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 clustering with Windows failover cluster 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 month, quarter, and year end. This workload typically has a 100% read I/O mix.

2.1 Testing Benchmark

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 where a population of users executes transactions against databases. Table 1 summarizes the typical SQL Server testing workloads.

Table 1) Typical SQL Server workloads.

Workload Type	TPC Test	Read/Write Ratio (Percentages)
OLTP	TPC-C	~75/25
	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 use the TPC-E tools from Microsoft or TPC-H using HammerDB (HammerDB.com). The detailed instructions on how to use these specific benchmarks are beyond the scope of this document.

3 SQL Server Configuration

This section provides general guidance about how to configure SQL Server settings that should be considered before and after installing SQL Server.

3.1 Shared Instance Versus Dedicated Instance

If an application has many schemas/stored procedures, it could potentially affect other apps that share SQL Server instance. Instance resources could potentially become divided/locked, which in turn causes performance issues for any other apps with 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 and SQL Server licenses. If application performance is paramount, then a dedicated instance is highly recommended.

Microsoft licenses SQL Server at the server level per core 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.

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

3.2 CPU Configuration

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 compute power it can use. For more information, consult [Compute Capacity Limits by Edition of SQL Server](#).

There are essentially 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's 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 1 shows the SQL Server log message after startup indicating the enforcement of the core limit.

Figure 1) 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\FUJIA2R30S'. 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:
    -s "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 2017 Editions](#).

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.

Processor 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 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 impact, 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 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 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.

For DSSs such as data warehouses, NetApp recommends starting 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.

3.3 Memory Configuration

Max Server Memory

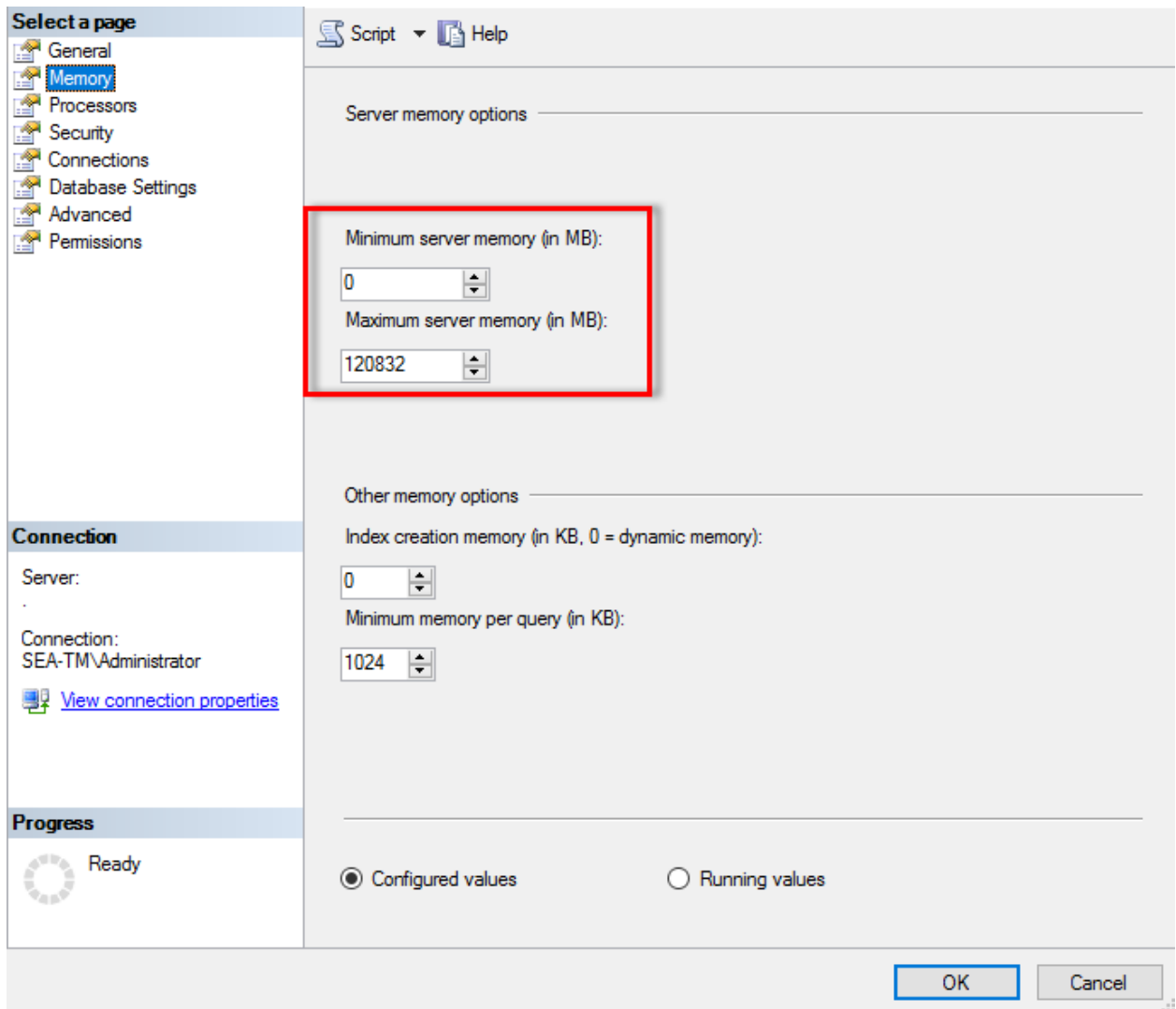
The max server memory option sets the maximum amount of memory that the SQL Server instance can use. It is generally used if multiple applications are running on the same server where SQL Server is running and you want to guarantee that these applications have sufficient memory to function properly.

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. Figure 2 displays how to set up minimum and maximum server memory.

Figure 2) 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
```

Max Worker Threads

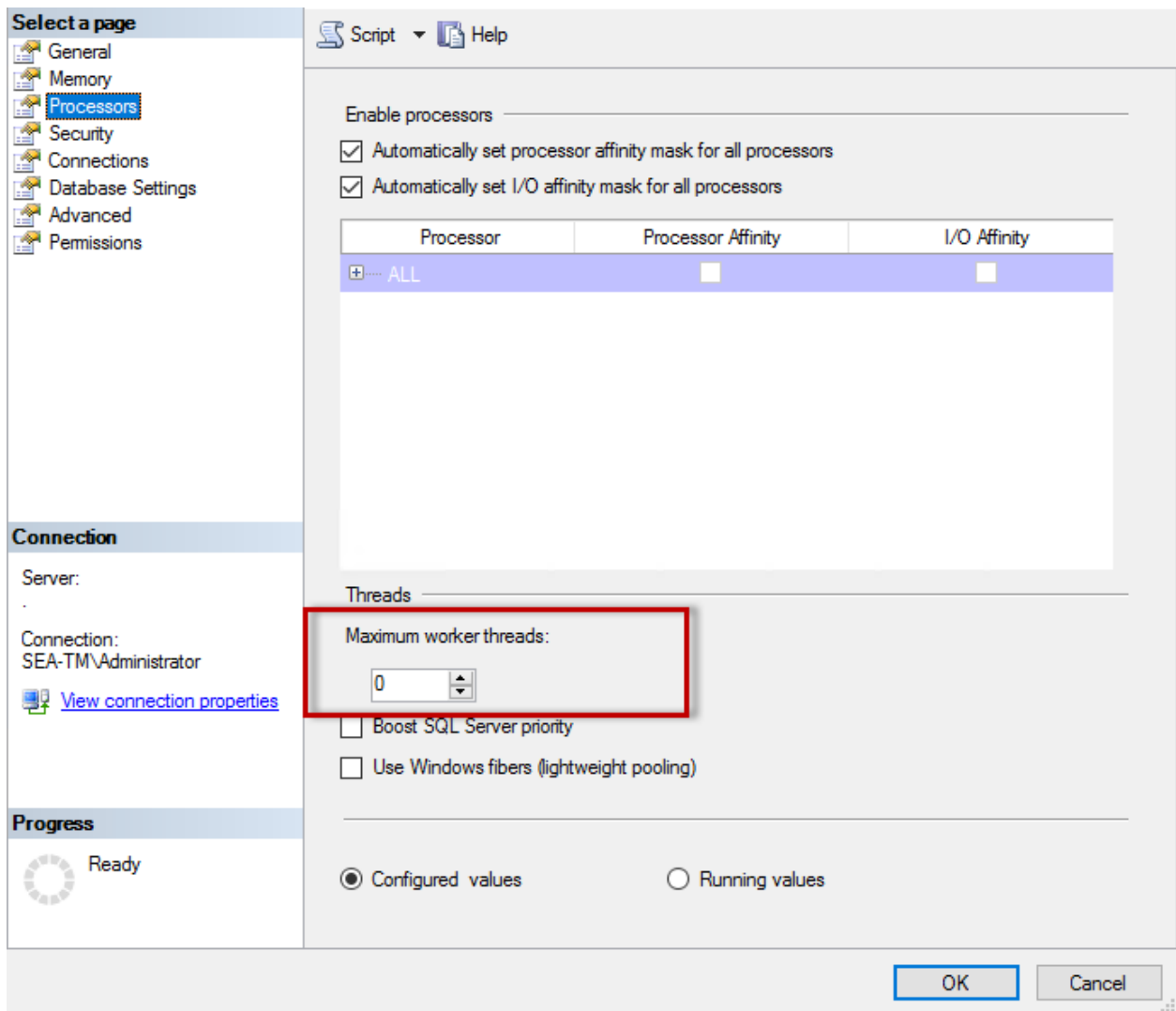
The max worker threads option helps to optimize performance when large numbers of clients are connected to SQL Server. Normally, a separate operating system thread is created for each query request. If hundreds of simultaneous connections are made to SQL Server, then one thread per query

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 will 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](#). Figure 3 indicates how to adjust maximum worker threads.

Figure 3) Configuring max worker threads using SQL Server Management Studio.



The following 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
```

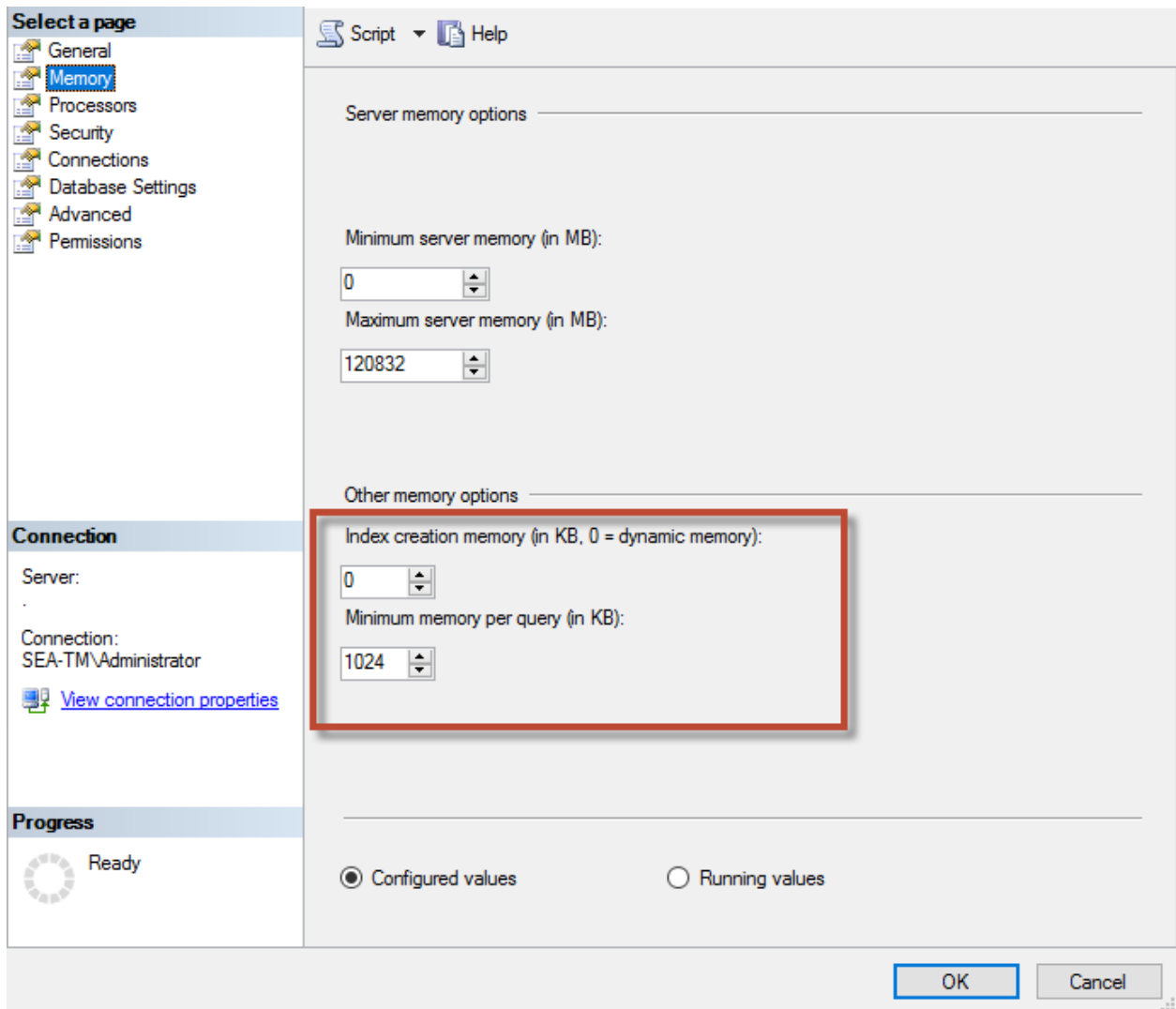
Index Create Memory

The index create memory option is another advanced option that usually should not be changed. 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 ≥ 1024 KB for each query to run. 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 likely hurts overall performance rather than helps it.

Figure 4) Configuring index create memory and min memory per query using SQL Server Management Studio.



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/Os 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

For buffer pool extensions, NetApp recommends:

- 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 is 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
```

4 Database Files and Filegroups

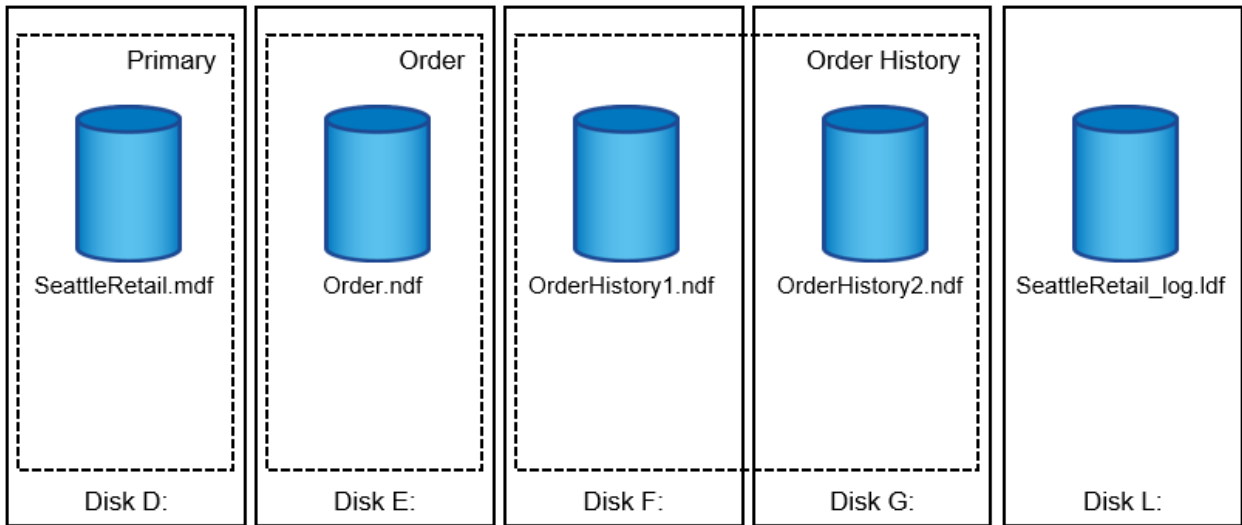
A SQL Server database is a collection of objects that allow one to store and manipulate data. In theory, SQL Server (64-bit) supports 32,767 databases per instance and 524,272TB of database size, although the typical installation usually has several databases. However, the number of the databases SQL Server can handle depends on the load and hardware. It is not unusual to see SQL Server instances hosting dozens or even hundreds or 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. SQL Server's 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.

Figure 5) Filegroups.



The ability to put multiple data files inside the filegroup allows us to spread the load across different storage devices, which helps to improve 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 us 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 underlying I/O configuration and 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.

It is generally recommended to avoid using 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 cases 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 into what data file it should write data. 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.

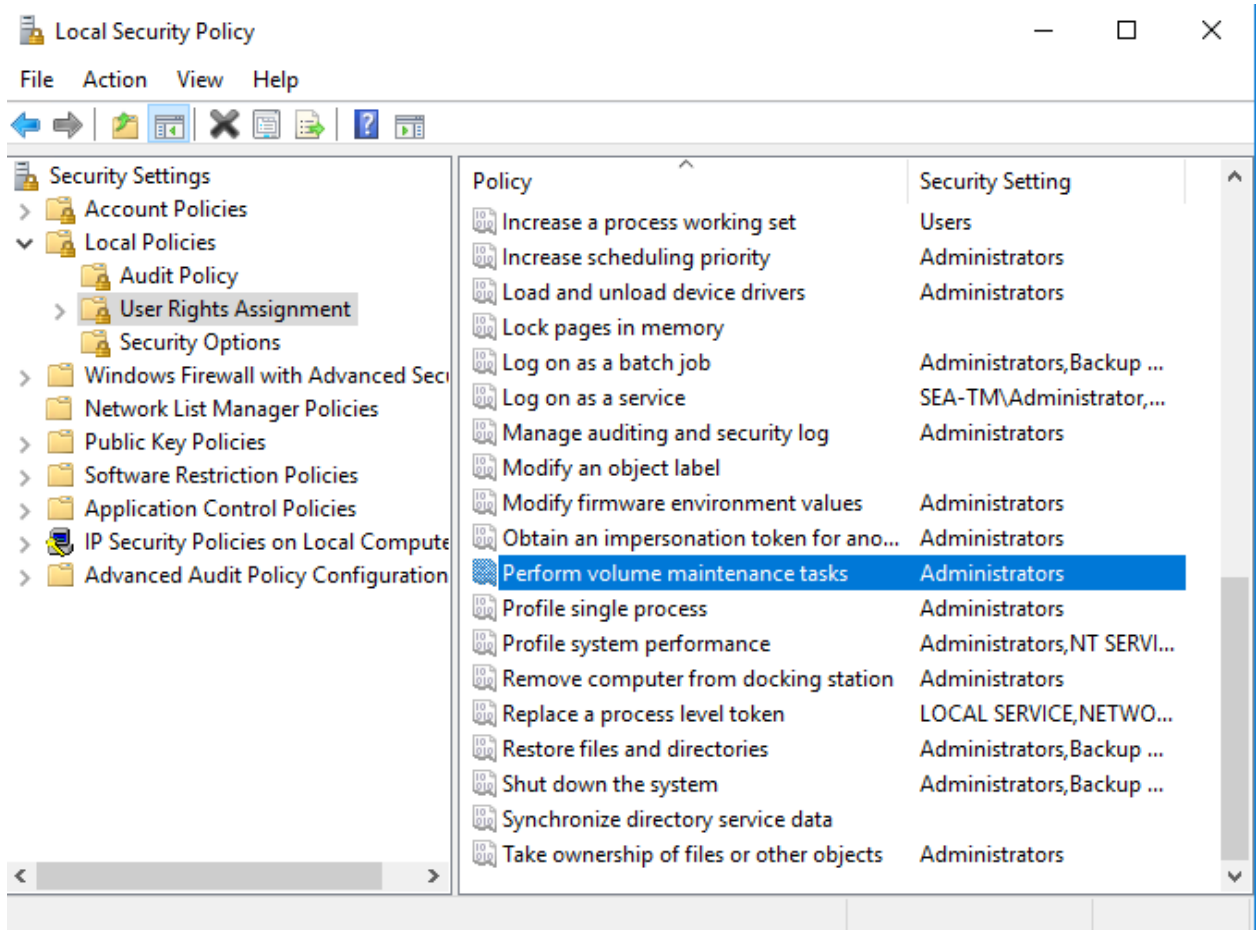
NetApp recommends that all files in the single filegroup have the same initial size and autogrowth parameters, with grow size defined in megabytes rather than percentages. This helps the proportional fill algorithm evenly balance write activities across data files.

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 if data files are zeroing out or not 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 the information from the previously deleted OS files. Database administrators are able to examine such data.

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



To check if 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

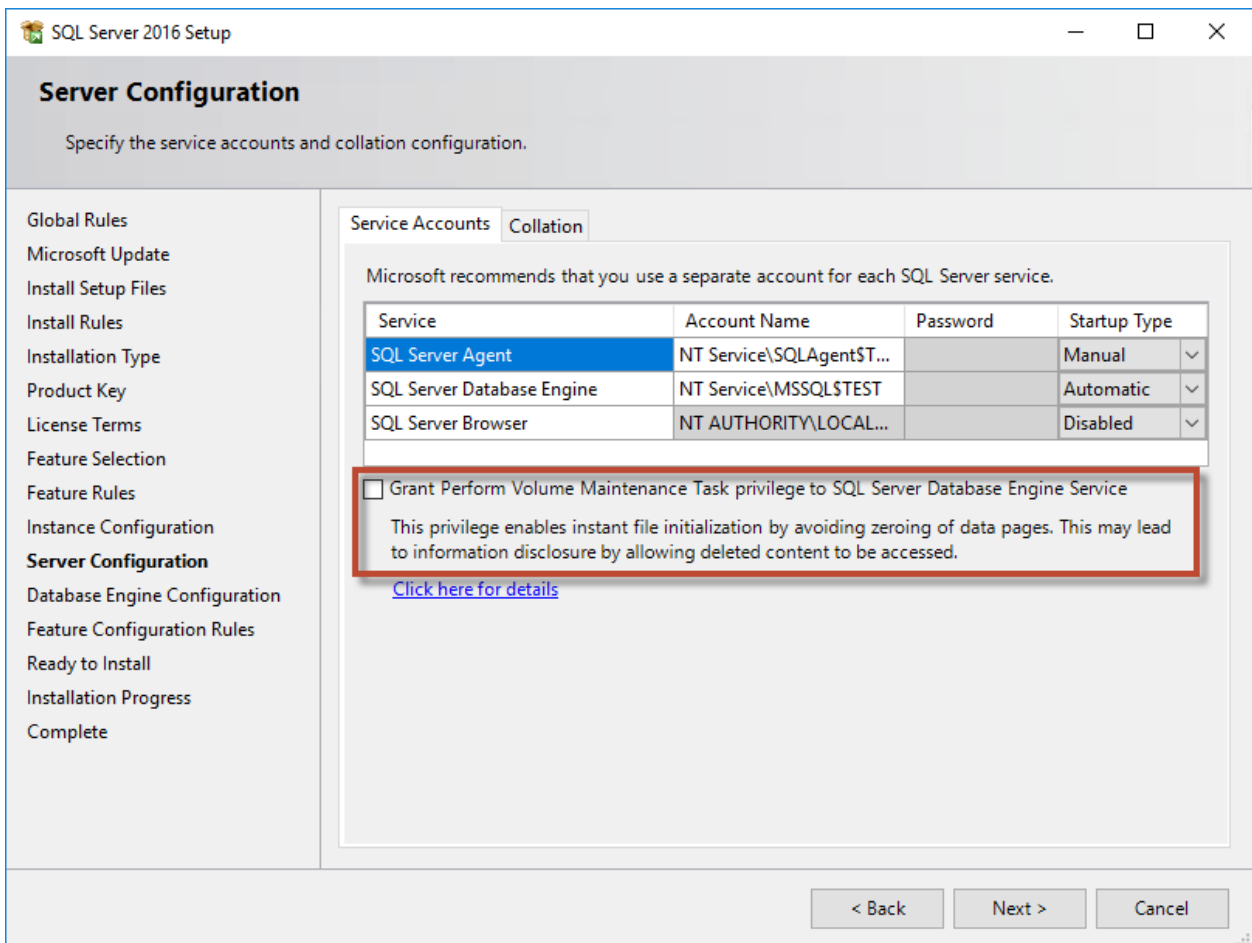
```

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	FixupLogTail(progress) zeroing C:\Program Files\Micros...
372	2017-02-09 08:10:08.160	spid53	Zeroing C:\Program Files\Microsoft SQL Server\MSSQ...
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 later as an option is provided during the installation process. Figure 6 displays the option to grant the SQL Server database engine service the privilege to perform the volume maintenance task.

Figure 6) Option for granting perform volume maintenance task privilege during SQL Server installation.



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 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.

5 Database and Storage

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.

5.1 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 abilities 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.

5.2 Volumes

NetApp FlexVol 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. The following are NetApp recommendation 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, a general recommendation is to give the volume label the same name as the mount point.
- Configure a volume autosize policy, when appropriate, to help prevent out-of-space conditions.

- 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 or SnapManager for SQL Server to coordinate Snapshot copies of the SQL Server data volumes.
- For SnapManager for SQL Server, place the SQL Server system databases on a dedicated volume or VMDK, because colocating system databases with user databases, including availability group databases, prevents Snapshot backups of the user databases. Backups of system databases are streamed into the SnapInfo LUN. This LUN is typically the same volume or VMDK that hosts the Windows operating system files and SQL Server binaries, which are random read/write workloads.
- `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.
- Create the host log directory (for SnapCenter) or SnapManager share (for SMSQL) on the dedicated FlexVol volume in which SnapManager or SnapCenter copies transaction logs.

x
Configure Plug-in for SQL Server

1 Log Directory

2 Verification Server

3 Summary

Configure the log backup directory for SC2ProdCluster.sea-tm.netapp.com

Configure host log directory

Host

Host log directory

Configure FCI instance log directory

FCI instance

FCI log directory

5.3 LUNs

tempdb Files

NetApp recommends proactively inflating `tempdb` files to their full size to avoid disk fragmentation. Page contention can occur on global allocation map (GAM), shared global allocation map (SGAM), or page free space (PFS) pages when SQL Server has to 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. A good rule of thumb for creating tempdb data files:

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

The following is the script to modify tempdb by creating 8 tempdb files and move tempdb to 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.

SnapInfo Directory and Host Log Directory

SnapManager for SQL Server (SMSQL) uses the SnapInfo directory as the main repository to store all metadata related to the SMSQL instance installed on a host (such as SnapManager backup metadata, stream-based backups of system databases, and transaction log backups).

SnapCenter uses a host log directory to store transaction log backup data. This is at the host and instance level. Each SQL Server host used by SnapCenter needs to 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 SnapInfo LUN and host log directory are calculated as follows:

Size of SnapInfo LUN or host log directory = ((system database size + (maximum DB LDF size × daily log change rate %)) × (Snapshot copy retention) ÷ (1 – LUN overhead space %))

The SnapInfo LUN and host log directory sizing formula assumes the following:

- A system database backup that does not include the tempdb database
- A 10% LUN overhead space

Place the SMSQL SnapInfo directory or host log directory on a dedicated volume or LUN. The amount of data in the SnapInfo directory or host log directory depends on the size of the backups and the number of days that backups are retained. The SnapInfo directory can be configured at any time by running the configuration wizard. Although using a single SnapInfo directory reduces complexity, separate SnapInfo directories offer flexibility for applying varying retention and archiving policies to databases. Host log directories can be configured from SnapCenter > Host > Configure Plug-in.

You can back up SQL Server databases to a NetApp SnapVault® location and perform the following operations:

- Restore all LUNs in a volume.
- Restore a single LUN from the vault.
- Access the latest Snapshot copy of a LUN directly from the backup vault.

The following are NetApp recommendations for SnapInfo directories and host log directories:

- Make sure that the SMSQL SnapInfo LUN or SnapCenter host log directory is not shared by any other type of data that can potentially corrupt the backup Snapshot copies.
- Do not place user databases or system databases on a LUN that hosts mount points.
- Always use the SMSQL or SnapCenter configuration wizards to migrate databases to NetApp storage so that the databases are stored in valid locations, enabling successful SMSQL or 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 using failover cluster instance, the SnapInfo or SnapCenter host log directory LUN must be a cluster disk resource in the same cluster group as the SQL Server instance being backed up by SnapManager.
 - If using 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.
- Make sure that the user database and the SnapInfo or host log directory are on separate volumes to prevent the retention policy from overwriting Snapshot copies when these are used with SnapVault technology.
- Make sure that SQL Server databases reside on LUNs separate from LUNs that have nondatabase 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 any database filegroups share LUNs, the database layout must be identical for all databases. However, this restriction is not applicable when the unrestricted database layout (UDL) option is enabled.
- Create LUNs with SnapCenter Plug-in for Microsoft Windows whenever possible. If you want to create LUNs with DiskManager or other tools, make sure that allocation unit size is set to 64K for partitions during formatting the LUNs.

5.4 SMB Shares

SMSQL supports databases on SMB shares only for SQL Server 2012 and later residing on ONTAP. SnapCenter currently does not support backup, restore, or cloning for databases on SMB shares. NetApp recommends layouts as shown in the following table.

Content	SMB Shares
SQL Server binaries and system databases	\\<CIFS servername>systemdb
User database files	\\<CIFS servername>\userdb_data
User log files	\\<CIFS servername>\userdb_log
SnapInfo	\\<CIFS servername>\SnapInfo

The following are NetApp recommendations for SMSQL and databases on SMB shares:

- Configure the Transport Protocol Setting dialog box in SnapCenter Plug-in for Microsoft Windows with the information for connecting to the SVM (SVM IP address, user name, and password) to view all SMB shares on its CIFS server, which then becomes visible to SnapManager for SQL Server.
- Disable the automatic Snapshot copy scheduling configured by SnapCenter Plug-in for Microsoft Windows.
- For SnapManager to be able to recognize the database file path as a valid file path hosted on NetApp storage, you must use the CIFS server name on the storage system in the SMB share path instead of the IP address of the management LIF or other data LIF. The path format is \\<CIFS server name>\<share name>. If the database uses the IP address in the share name, manually detach and attach the database by using the SMB share path with the CIFS server name in its share name.
- When provisioning volumes for SMB environments, you must create the volumes as NTFS security-style volumes.
- Make sure that all database files (the data files, in addition to the transaction log files) reside on SMB shares instead of placing them across LUNs and SMB shares.
- Make sure that no antivirus scanning is performed on the SMB/CIFS shares in which SQL Server data is stored to avoid failed transactions due to latency.
- Make sure that Windows host caching is disabled on the SMB/CIFS share in which SQL Server data is stored to avoid corruption caused by caching.
- When you use availability groups, the transaction logs are streamed to a shared network location that is a SnapManager SMB share accessible by all the replicas. Therefore, verify that this CIFS share is sized to accommodate the transaction logs.

5.5 Data Storage Design

This section contains a few examples of SQL Server designs for NetApp storage and considerations for environments that use SnapManager for SQL Server and SnapCenter.

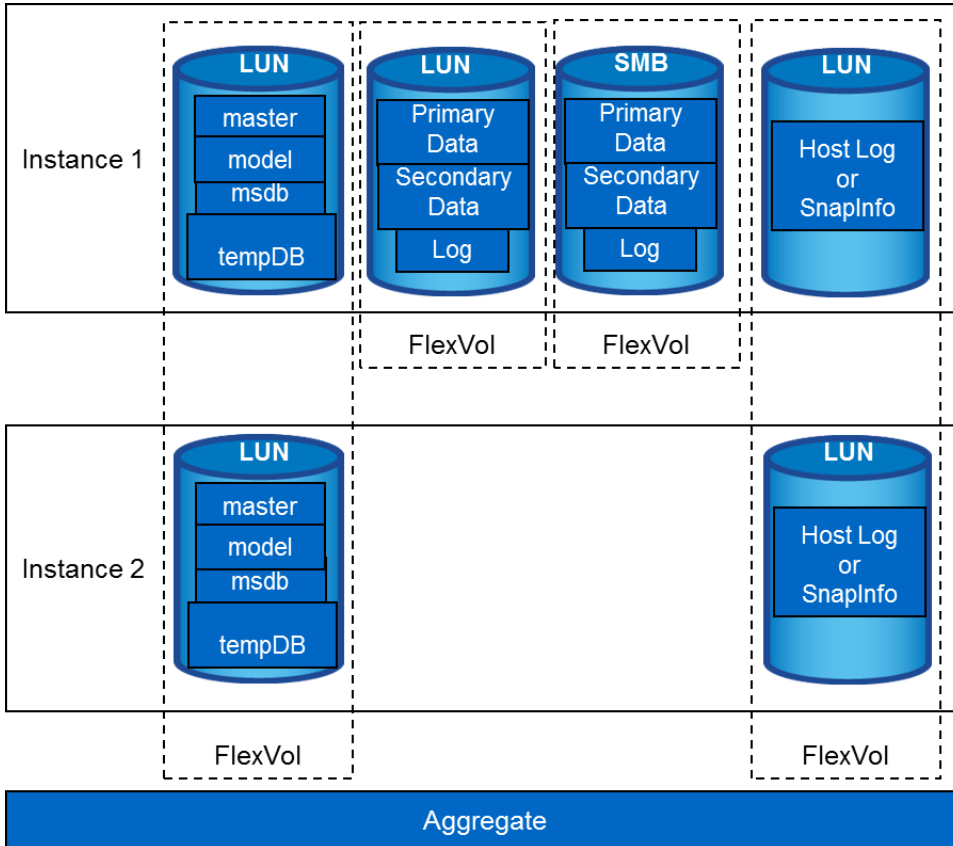
Design Example 1

This configuration can be used for SQL Server instances that require basic performance and contain multiple small databases. The database storage design has the following characteristics:

- Contains one aggregate for SQL Server instances.
- Uses a dedicated volume and LUN for the SQL Server system databases, including the tempdb database.
- Uses a dedicated LUN for each database.
- Uses a single volume for both data and log.
- Uses dedicated SMB shares for both data and log (if using SMSQL for backup).
- Has a relatively lower recovery time objective (RTO) because data and log reside in the same volume.

- Is suited for small-size databases in which medium to low performance is sufficient.

Figure 7) Basic SQL Server database design for NetApp storage for SMSQL or SnapCenter.



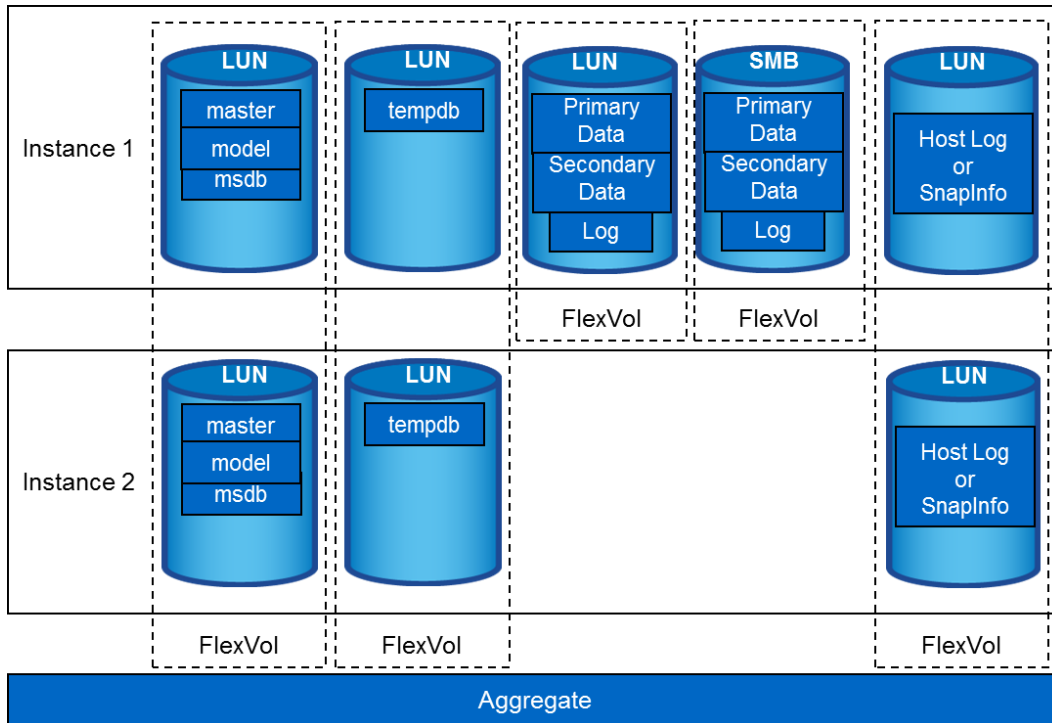
Because system databases, including tempdb databases, reside in the same volume, the database backup is performed using native SQL Server but not SMSQL or SnapCenter.

Design Example 2

This configuration is designed to be used for SQL Server instances that require basic performance and contain multiple databases that are backed up using either SMSQL or SnapCenter. The database storage design has the following characteristics:

- Contains one aggregate for SQL Server instances.
- Uses a dedicated volume and LUN for the SQL Server system databases.
- Uses a dedicated volume and LUN for tempdb database.
- Uses a dedicated LUN for each database.
- Uses a single volume for both data and log.
- Uses dedicated SMB shares for both data and log (if using SMSQL for backup).
- Has a low to medium RTO because the data and log reside in the same volume.
- Is suitable for medium-size databases where medium performance required.

Figure 8) SQL Server database design for NetApp storage using SMSQL or SnapCenter.

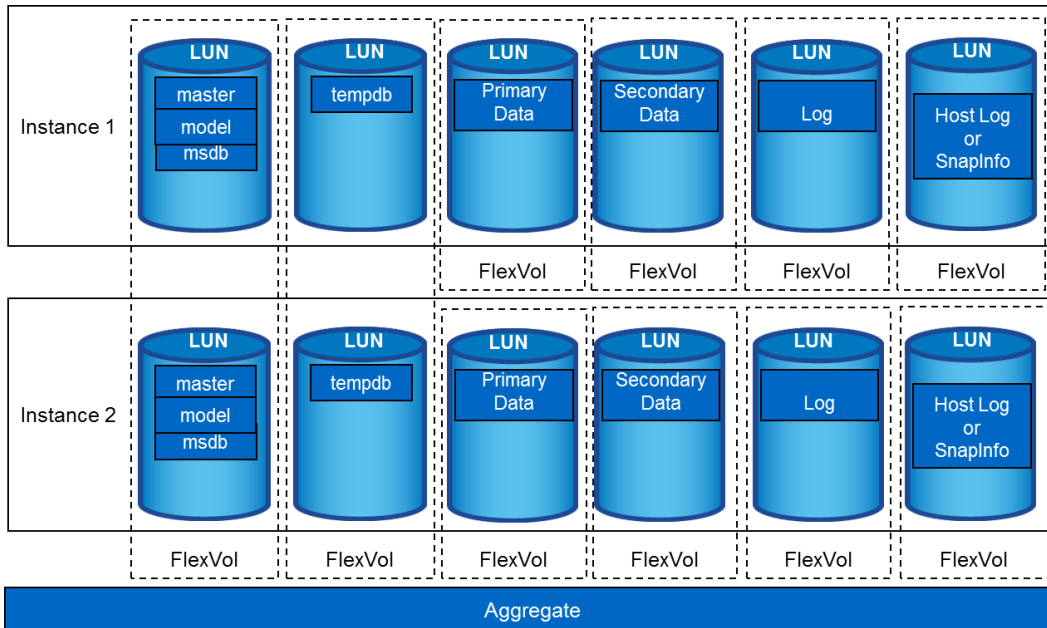


Design Example 3

This configuration is designed to be used for SQL Server instances that require high performance and contain a few databases that are backed up using either SMSQL or SnapCenter. The database storage design has the following characteristics:

- Contains one aggregate for SQL Server instances.
- Uses a dedicated volume and LUN for the SQL Server system databases.
- Uses a dedicated volume and LUN for tempdb database.
- Uses a dedicated LUN for each user database.
- Uses dedicated volumes for primary and secondary data and log files.
- Has a relatively high RTO because data and log reside in separate volumes.
- Is suitable for medium to large size databases where high performance required.

Figure 9) Server database design for NetApp storage using SMSQL or SnapCenter.



5.6 Virtualization

SQL Server virtualization enables consolidation, which boosts efficiency. Recent developments have brought about very powerful virtual machines that can manage larger capacities and provide faster provisioning of the database environment. In addition, SQL Server virtualization is cost-effective and convenient. There are three things to consider when virtualizing SQL Server: disk throughput, memory, and processors. This document focuses on disk only. The two major hypervisors that are commonly used to virtualize SQL Server are Hyper-V and VMware. Information about each hypervisor is covered in the following section.

Hyper-V

The improvements in Windows Server 2012 Hyper-V and its expanded virtual machine capabilities eliminated most of the limits related to performance in a virtualized environment. Windows Server 2016 Hyper-V provides better consolidation of workloads that are traditionally more complex and that tend to saturate resources and contend for other system resources and storage.

SQL Server 2016 and Windows Server 2016 provide a host of new features that can be used to effectively virtualize demanding complex database workloads such as online transaction processing/analysis (OLTP/OLTA), data warehousing, and business intelligence, which were not previously considered for virtualization.

All virtual hard disks (VHDs) are just files. They have a specialized format that Hyper-V and other operating systems can mount to store or retrieve data, but they are nothing more than files. There are three basic types of VHDs that you can use with VMs. These are fixed VHDs, dynamic VHDs, and pass-through disks:

- **Fixed VHDs.** When creating fixed VHDs, operating systems allocate 100% of the indicated space on the underlying physical media. There is no block allocation table to be concerned with, so the extra I/O load above a pass-through disk is only what occurs within the virtualization stack. Fixed VHD benefits are:
 - Fastest VHD mechanism
 - No potential for causing overcommitment collisions

- Always same fragmentation level as at creation time

Fixed VHD drawbacks include:

- VM-level backups capture all space, even unused.
- Larger size inhibits portability.

Realistically, the biggest problems with this system appear when you need to copy or move the VHD. The VHD itself doesn't know which of its blocks are empty, so 100% of them have to be copied. Even if your backup or copy software employs compression, empty blocks aren't necessarily devoid of data. In the guest's file system, "deleting" a file just removes its reference in the file allocation table. The blocks themselves and the data within them remain unchanged until another file's data is saved over them.

- **Dynamic VHDs.** A dynamic VHD has a maximum size but is initially small. The guest operating system believes that it just a regular hard disk of the size indicated by the maximum. As it adds data to that disk, the underlying VHD file grows. This technique of only allocating a small portion of what could eventually become much larger resource usage is also commonly referred to as thin provisioning. The following are some benefits of dynamic VHD:
 - Quick allocation of space. Because a new dynamic VHD contains only header and footer information, it is extremely small and can be created almost immediately.
 - Minimal space usage for VM-level backups. Backup utilities that capture a VM operate at the VHD level and back it up in its entirety no matter what. The smaller the VHD, the smaller (and faster) the backup.
 - Overcommitment of hard drive resources. You can create 20 virtual machines with 40GB boot VHDs on a hard drive array that only has 500GB of free space.
- **Pass-through disks.** These are not virtualized at all, but hand I/O from a virtual machine directly to a disk or disk array on the host machine. This could be a disk that is internal to the host machine or a LUN on an external system connected by FC or iSCSI. This mechanism provides the fastest possible disk performance but has some restrictive drawbacks. The following are some benefits of pass-through disks:
 - Fastest disk system for Hyper-V guests.
 - If the underlying disk storage system grows (such as by adding a drive to the array) and the virtual machine's operating system allows dynamic disk growth, the drive can be expanded within the guest without downtime.

The following are some drawbacks of pass-through disks:

- Live migrations of VMs that use pass-through disks are noticeably slower and often include an interruption of service. Because pass-through disks are not cluster resources, they must be temporarily taken offline during transfer of ownership.
- Hyper-V's VSS writer cannot process a pass-through disk. That means that any VM-level backup software has to take the virtual machine offline while backing it up.
- Volumes on pass-through disks are nonportable. This is most easily understood by its contrast to a VHD. You can copy a VHD from one location to another, and it works the same way. Data on pass-through volumes is not encapsulated in any fashion.

The following are NetApp recommendations regarding disk storage for SQL Server on Hyper-V guests:

- Always use a fixed VHD for high-I/O applications such as Exchange and SQL Server. Even if you won't be placing as much burden on these applications as they were designed to handle, these applications always behave as though they need a lot of I/O and are liberal in their usage of drive space. If space is a concern, start with a small fixed VHD; you can expand it if necessary.
- If you aren't certain, try to come up with a specific reason to use fixed. If you can't, then use dynamic. Even if you determine afterward you made the wrong choice, you can always convert the drive later. It takes some time to do so (dependent upon your hardware and the size of the VHD), but you probably won't need to make the conversion.

- A single virtual machine can have more than one VHD, and it is acceptable to mix and match VHD types. Your virtualized SQL Server can have a dynamic VHD for its C: drive to hold Windows and a fixed VHD for its D: drive to hold SQL Server data.
- If using dynamic drives in an overcommit situation, set up a monitoring system or a schedule to keep an eye on physical space usage. If the drive dips below 20% available free space, the VSS writer might not activate, which can cause problems for VM-level backups. If the space is ever completely consumed, Hyper-V pauses all virtual machines with active VHDs on that drive.
- For a high-performance production virtual SQL Server instance, it's important that you put your OS files, data files, and log files on different VHDs or pass-through disks. If you're using a shared storage solution, it's also important that you be aware of the physical disk implementation and make sure that the disks used for the SQL Server log files are separate from the disks used for the SQL Server data files.

VMware

VMware virtual machines include a set of files in typically one of two given formats: virtual machine file system (VMFS) or raw device mapping (RDM). Both formats enable you to access the virtual machine's disk (VMDK), but they differ in approach to storage, and VMware recommends VMFS for most VMs. With VMFS, the VMDK files also hold the data, while with RDM, the data is stored on an external disk system similar to Hyper-V pass through disks. VMFS holds disk data from multiple VMs; RDM does not.

VMFS was designed specifically to support virtualization. Although RDM is sometimes recommended for I/O-intensive operations, with VMFS, a storage volume can support one or many VMs. This volume can change without affecting network operations. Because they share storage volumes, VMs are easier to manage, and resource utilization remains high. Various ESXi servers can read and write to the file system simultaneously, because it stores information at the block level.

The following are NetApp recommendations for VMDK:

- Use separate VMDKs for primary (.mdf) and log (.ldf) files for user databases. Make sure that these VMDKs reside in a datastore placed on a separate volume from the volume containing system databases and the operating system VMDKs.
- Use separate VMDKs for system databases (master, model, and msdb). Make sure that these VMDKs reside in a datastore placed on a separate volume from the volume containing user databases and the operating system VMDKs.
- Use separate VMDKs for the tempdb database.
- Use NetApp Virtual Storage Console (VSC) for VMware vSphere to provision VMDKs.
- Create user databases directly on the VSC-provisioned VMDKs.
- Data files (tables and indexes) are the primary files that are used by the SQL Server storage engine. Each database might have multiple files and be spread across multiple VMDKs.
- If possible, avoid sharing volumes and datastores between different Windows Server machines.

ESXi can access a designated NFS volume on a NAS server, mount the volume, and use it for its storage needs. You can use NFS volumes to store and boot virtual machines in the same way that you use VMFS datastores. ESXi supports the following shared storage capabilities on NFS volumes:

- vMotion
- VMware DRS and VMware HA
- ISO images, which are presented as CD-ROMs to virtual machines
- Virtual machine snapshots

ESX 5.0 and later support up to 256 NFS datastores. The default value is 8, but this value can be increased to the maximum number that is specific to the version of ESX or ESXi being used. The following are NetApp recommendations for NFS datastores:

- Make sure that each NFS export that is used as an NFS datastore resides on its own volume.
- Use one NFS datastore for multiple system databases from multiple instances.
- Use one NFS datastore per user database and user log; alternatively, separate the user database and the user log to different NFS datastores.
- Do not define a default gateway for the NFS storage network.
- Make sure that each NFS datastore is connected only once from each ESX or ESXi server using the same NetApp target IP address on each ESX or ESXi server.

Figure 10 is a simple layout for small databases (~=<200GB) that do not require very low (submillisecond) latency. This layout makes it easy to manager datastores and flexible volumes.

Figure 10) Example of simple database layout on VMDKs with VMFS or NFS datastores.

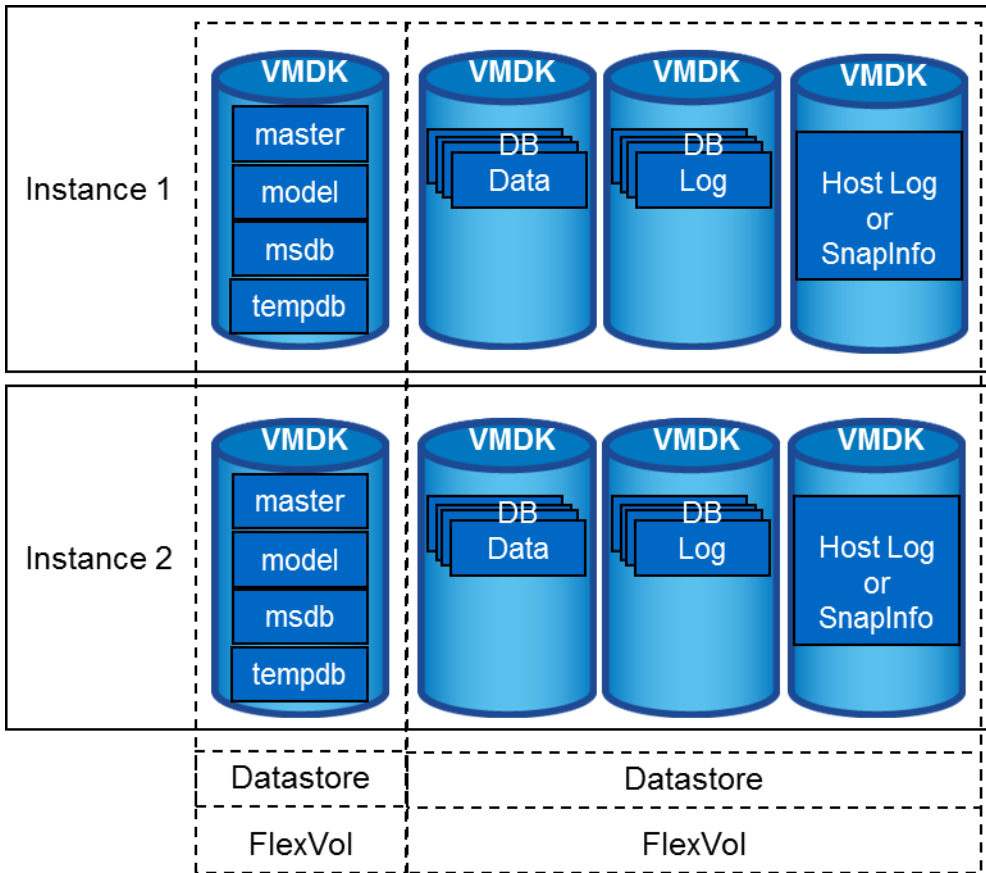
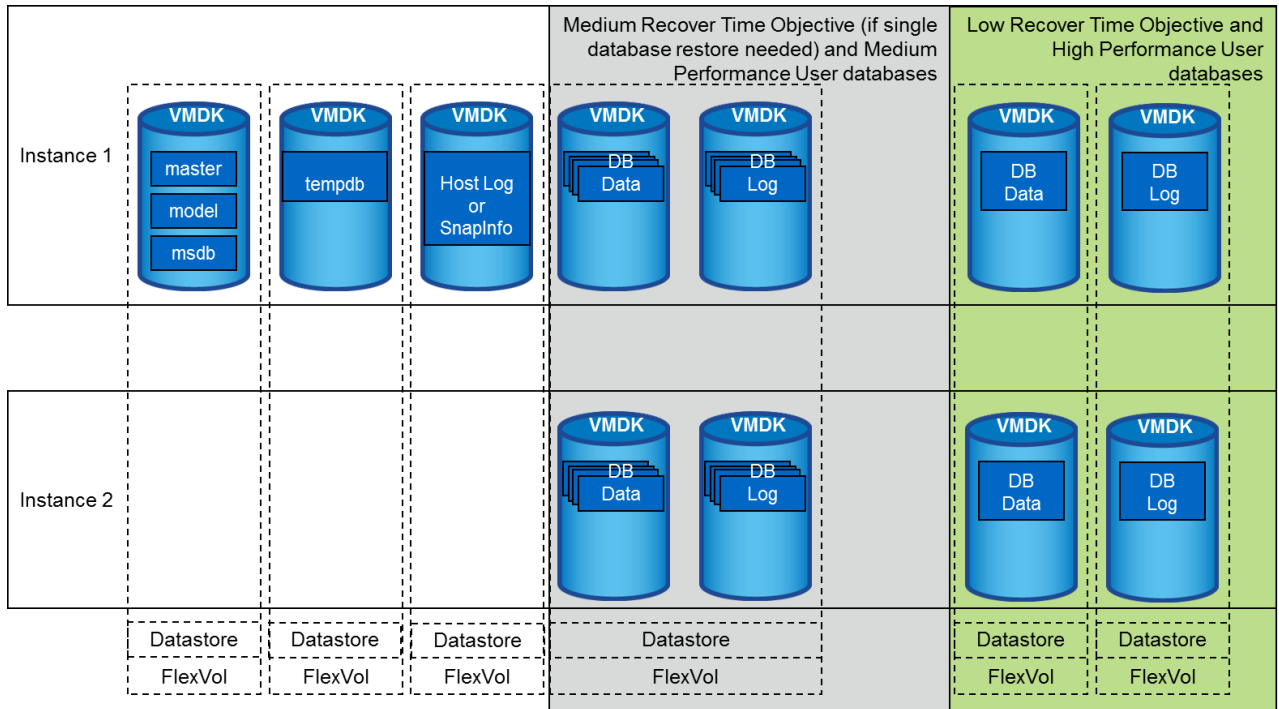


Figure 11 is a database layout example for VMware using VMDK with VMFS or NFS datastores. The database storage design has the following characteristics:

- If the user wants to restore a single database in the VMDK that resides on NFS datastore, SMSQL or SnapCenter creates a flexible volume from selected Snapshot copies and performs single-file SnapRestore® (SFSR) to recover the data file and log. Therefore, the recovery time is slightly longer than when using volume SnapRestore.
- When placing multiple database files and logs in the same VMDK, there is a chance that I/O might compete, and performance might not be as high as dedicated VMDK.

Figure 11) Example of database layout on VMDKs with VMFS or NFS datastores.



With RDM, the VM directly connects to the SAN using a dedicated storage LUN. The total number of LUNs visible to an ESXi host is capped at 256, with the same LUNs visible across a whole cluster of up to 32 ESXi servers. RDM is recommended in a few specific situations, such as when a virtual machine is SAN-aware. NetApp recommends using SnapCenter Plug-in for Microsoft Windows when using RDM LUNs on the virtual machine. NetApp recommends using RDM for one of the following reasons:

- If you need files larger than 2TB in size, because VMDK files are limited to 2TB in size
- If using clustered data and quorum disks is required (applies for both virtual-to-virtual and physical-to-virtual clusters)

6 Storage Efficiency and Manageability

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 impact on the overall performance of the system. Storage efficiency goes beyond data deduplication; it is a combination of RAID, provisioning (overall layout and utilization), mirroring, and other data protection technologies.

The following NetApp technologies implement storage efficiency and reap its cost savings benefits by optimizing existing storage in the infrastructure and deferring or avoiding future storage expenditures. The more of these technologies that are used together, the larger the savings.

6.1 Snapshot Copies

NetApp Snapshot technology provides low-cost, fast backup, point-in-time copies of the file system (volume) or LUN by preserving ONTAP architecture WAFL[®] consistency points. NetApp SnapCenter or SnapManager integrates with the SQL Server virtual device interface (VDI) for the creation of application-consistent Snapshot copies of production SQL Server databases with no downtime for the production database.

There is no performance penalty for creating Snapshot copies, because data is never moved as it is with other copy-out technologies. The cost for Snapshot copies is only at the rate of block-level changes, not at 100% for each backup, as is the case with mirror copies. Snapshot technology can result in savings in storage costs for backup and restore purposes and opens up several efficient data management possibilities.

If a database uses multiple LUNs on the same volume, then all Snapshot copies of these LUNs are made simultaneously, because Snapshot copies are volume-based. In certain situations, a SnapManager for SQL Server (SMSQL) or SnapCenter clone operation restores a LUN from a Snapshot copy for temporary read/write access to an alternative location by using a writable Snapshot copy during the SMSQL or SnapCenter verification process.

6.2 Thin Provisioning

Thin provisioning is a method for optimizing the use of available storage space. It relies on on-demand allocation of data blocks versus the traditional method of allocating all of the blocks up front. This methodology eliminates almost all white space, thus helping avoid poor utilization rates.

FlexVol volumes are the enabling technology behind NetApp thin provisioning and can be thought of as the virtualization layer of ONTAP. When a LUN is created, it does not dedicate specific data blocks out of the volume for itself or for its Snapshot copies. Instead, it allocates the blocks from the aggregate when the data is written. This allocation method allows the administrator to provision more storage space, as seen from the connected servers, than the space that is physically available in the storage system.

When storage consumption is unpredictable or highly volatile, it is best to reduce the level of storage overcommitment so that storage is available for any growth spikes. Consider limiting storage commitment to 100%—no overcommitment—and using the trending functionality to determine how much overcommitment is acceptable, if any. Overcommitment of storage must be carefully considered and managed for mission-critical applications (such as SQL Server) for which even a minimal outage is intolerable. In such cases, it is best to monitor storage consumption trends to determine how much overcommitment is tolerable.

If the time required to procure new storage is long, storage overcommitment thresholds should be adjusted accordingly. The overcommitment threshold should alert administrators early enough to allow new storage to be procured and installed. The potential risk when configuring the SQL Server environment for thin provisioning is a LUN going offline when not enough space is available to write further data. Use the volume autogrow functionality as a mitigation mechanism to safely allow thin provisioning and higher storage utilization.

6.3 Space Guarantee

Space guarantees enable thin provisioning. The space guarantee option can be set at the volume or LUN level. If the space guarantee at the volume level is set to volume (default setting), the amount of space required by the FlexVol volume is always available from its aggregate during the creation of that volume.

If the space guarantee for the volume is set to none, the volume reserves no space from the aggregate during volume creation. Space is taken from the aggregate when data is written to the volume. Write operations to space-reserved LUNs in a volume with the setting `guarantee=none` fail if the containing aggregate does not have enough available space.

LUN reservation makes sure that the LUN has space in the volume, but setting `guarantee=none` does not guarantee that the volume has space in the aggregate. When the space guarantee for the volume is set to file, the aggregate keeps space available for completely rewriting LUNs that have space reservation enabled.

NetApp recommends using thin provisioning in SQL Server environments to improve space utilization and to reduce the overall storage requirements when the space guarantee functionality is used.

6.4 Space Reclamation

Space reclamation can be initiated periodically to recover unused space in a LUN. Storage space can be reclaimed at the storage level by using the SnapDrive start space reclaimer option, thus reducing utilization in the LUN and in Snapshot copies.

With SnapCenter, you can use the following PowerShell command to start space reclaimer.

```
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.

6.5 Fractional Reserve

Fractional reserve is a volume option that determines how much space ONTAP reserves to be used for Snapshot copy overwrite for LUNs after all other space in the volume has been used.

NetApp storage appliances can be configured in many different ways for LUN thin provisioning; each method has advantages and disadvantages. It is possible to have thin provisioned volumes and non-thin-provisioned volumes on the same storage system or even on the same aggregate. The following two options are considered best practice configurations for using thin provisioning with SQL Server.

Volume Guarantee Set to None

The advantage of the configuration in Table 2 is that the free space in the aggregate is used as a shared pool of free space. The disadvantages of this configuration are the high level of dependency between volumes and the fact that the level of thin provisioning cannot be easily tuned on an individual volume basis.

When using the configuration in Table 2, the total size of the volumes is greater than the actual storage available in the host aggregate. With this configuration, storage administrators can generally size the volume so that they only need to manage and monitor the used space in the aggregate. This option does not affect the space for hosting live data but rather allows the backup space to dynamically change.

Table 2) Volume guarantee set to none.

Setting	Value
volume guarantee	none
LUN reservation	enabled
fractional_reserve	0%
snap_reserve	0%
autodelete	volume/oldest_first
autosize	off
try_first	snap_delete

Using Autogrow and Autodelete

The configuration in Table 3 allows the administrator to finely tune the level of thin provisioning for SQL Server environments. With this configuration, the volume size defines or guarantees an amount of space that is available only to LUNs within that volume. The aggregate provides a shared storage pool of available space for all the volumes contained within it.

If the LUNs or Snapshot copies require more space than the space available in the volume, the volumes automatically grow, taking more space from the containing aggregate. Additionally, the advantage of having the LUN space reservation setting disabled is that Snapshot copies can use the space that is not needed by the LUNs. The LUNs themselves are not in danger of running out of space because the autodelete feature removes the Snapshot copies that are consuming space.

Note: Snapshot copies used for creating NetApp FlexClone® volumes are not deleted by the autodelete option.

Table 3) Setting up volume with autodelete and autogrow.

Setting	Value
volume guarantee	volume
LUN reservation	disabled
fractional_reserve	0%
snap_reserve	0%
autodelete	volume/oldest_first
autosize	on
try_first	autogrow

NetApp recommends using autogrow for most common deployment configurations. The reason for this is that the storage admin only needs to monitor space usage in the aggregate.

6.6 NetApp FlexClone

NetApp FlexClone technology can be used to quickly create a writable copy of a FlexVol volume, eliminating the need for additional copies of the data. FlexClone volumes are great for any situation in which testing or development occurs, any situation in which progress is made by locking in incremental improvements, and any situation in which there is a desire to distribute data in changeable form without endangering the integrity of the original. A common scenario for using FlexClone is for test/dev purposes before a rollup patch or hotfix installation.

FlexClone technology can be leveraged both at the primary storage system and at the SnapMirror destination for effective utilization of resources. FlexClone can also be used for disaster recovery testing without affecting the operational continuity of the SQL Server environment.

SQL Server requires application-consistent Snapshot copies to create FlexClone volumes based on Snapshot. 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](#).

6.7 Compression, Compaction, and Deduplication

Compression and deduplication are two storage efficiency options that increase the amount of logical data that fits on a given amount of physical storage. At a high level, compression is a mathematical process whereby patterns in data is detected and encoded in a way that reduces space requirements. In contrast, deduplication detects actual repeated blocks of data and removes the extraneous copies. Although they deliver similar results, they work in significantly different ways and therefore must be managed differently.

Inline data compaction is a technology introduced in ONTAP 9, which improves compression efficiency. Adaptive compression alone can provide at best 2:1 savings because it is limited to storing an 8K I/O in a

4K WAFL block. Compression methods such as secondary compression use a larger block size and deliver better efficiency but are not suitable for data that is subject to small block overwrites. Decompressing 32KB units of data, updating an 8K portion, recompressing, and writing back to disk create overhead.

Compression

There are multiple ways to compress a database. Until recently, compression was of limited value because most databases required a large number of spindles to provide sufficient performance. One side effect of building a storage array with acceptable performance was that the array generally offered more capacity than required. The situation has changed with the rise of solid-state storage. There is no longer a need to vastly overprovision the drive count to obtain good performance.

Even without compression, migrating a database to a partially or fully solid-state storage platform can yield significant cost savings because doing so avoids the need to purchase drives only needed to support I/O. For example, NetApp has examined some storage configurations from recent large database projects and compared the costs with and without the use of solid-state drives (SSDs) using NetApp Flash Cache™ and Flash Pool™ intelligent data caching or all-flash arrays. These flash technologies decreased costs by approximately 50% because IOPS-dense flash media permit a significant reduction in the number of spinning disks and shelves than are otherwise required. See [NetApp AFF8080 EX Performance and Server Consolidation with Microsoft SQL Server 2014](#) for additional information.

As stated earlier, the increased IOPS capability of SSDs almost always yields cost savings, but compression can achieve further savings by increasing the effective capacity of solid-state media.

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.

NetApp Adaptive Compression

Adaptive compression has been thoroughly tested with SQL Server workloads, and the performance effect has been found to be negligible, even in an all-flash environment (where it is enabled by default) in which latency is measured in microseconds. In initial testing, some customers have reported a performance increase with the use of compression. This increase is the result of compression effectively increasing the amount of SSD available to the database.

ONTAP manages physical blocks in 4KB units. Therefore, the maximum possible compression ratio is 2:1 with a typical SQL Server database using an 8KB block. Early testing with real customer data has shown compression ratios approaching this level, but results vary based on the type of data stored.

NetApp Secondary Compression

Secondary compression uses a larger block size that is fixed at 32KB. This feature enables ONTAP to compress data with increased efficiency, but secondary compression is primarily designed for data at rest or data that is written sequentially and requires maximum compression.

NetApp recommends secondary compression for data such as transaction logs and backup files. These types of files are written sequentially and not updated. This point does not mean that adaptive compression is discouraged. However, if the volume of data being stored is large, then secondary compression delivers better savings when compared to adaptive compression.

Consider secondary compression of data files when the amount of data is very large and the data files themselves are either read-only or rarely updated. Data files using a 32KB block size should see more compression under secondary compression that has a matching 32KB block size. However, care must be taken to verify that data using block sizes other than 32KB are not placed on these volumes. Only use this method in cases in which the data is not frequently updated.

Inline Data Compaction

Inline data compaction works by allowing logical WAFL blocks to be stored within physical WAFL blocks. For example, a database with highly compressible data such as text or partially full blocks may compress from 8KB to 1KB. Without compaction, that 1KB of data still occupies an entire 4KB block. Inline data compaction allows that 1KB of compressed data to be stored in just 1KB of physical space alongside other compressed data. It is not a compression technology. It is simply a more efficient way of allocating space on disk and therefore should not create any detectable performance impact.

The degree of savings obtained varies. Data that is already compressed or encrypted cannot generally be further compressed, and therefore such datasets do not benefit from compaction. Newly initialized SQL Server data files that contain little more than block metadata and zeros compress up to 80:1. This creates an extremely wide range of possibilities. The best way to evaluate potential savings is using the NetApp Space Savings Estimation Tool (SSET) available on NetApp Field Portal or through your NetApp representative.

Deduplication

NetApp does not recommend using deduplication with SQL Server database files primarily because this process is almost entirely ineffective. A SQL Server page contains a header that is globally unique to the database and a trailer that is nearly unique. One percent space savings are possible, but this is at the expense of significant overhead caused by data deduplication.

Many competing arrays claim the ability to deduplicate SQL Server databases based on the presumption that a database is copied multiple times. In this respect, NetApp deduplication could also be used, but ONTAP offers a better option: NetApp FlexClone technology. The result is the same; multiple copies of a SQL Server database that share most of the underlying physical blocks are created. Using FlexClone is much more efficient than taking the time to copy data files and then deduplicate them. It is, in effect, nonduplication rather than deduplication, because a duplicate is never created in the first place.

In the unusual case in which multiple copies of the same data files exist, deduplication can be used.

NetApp recommends that you do not 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.

6.8 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 of SnapMirror for SQL Server:

- 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 either SMSQL or 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.
- Mirror the CIFS share used by the availability group to the secondary data center for disaster recovery purposes.

For more information about SnapMirror, see the following resources:

- [TR-4015-0118-SnapMirror-Configuration-Best-Practices-ONTAP-9.1-9.2](#)
- [TR-4733 SnapMirror Synchronous for ONTAP 9.5](#)

7 Data Fabric and SQL Server

7.1 Introduction

The Data Fabric is NetApp's vision for the future of data management. It enables customers to respond and innovate more quickly because their data is free to be accessed where it is needed most. Customers can realize the full potential of their hybrid cloud and make the best decisions for their business.

To fulfill this vision, the Data Fabric defines the NetApp technology architecture for hybrid cloud. NetApp products, services, and partnerships help customers seamlessly manage their data across their diverse IT resources, spanning flash, disk, and cloud. IT has the flexibility to choose the right set of resources to meet the needs of applications and the freedom to change them whenever they want.

A true Data Fabric delivers on five major design principles:

- **Control.** Securely retain control and governance of data regardless of its location: on the premises, near the cloud, or in the cloud.
- **Choice.** Choose cloud, application ecosystem, delivery methods, storage systems, and deployment models, with freedom to change.
- **Integration.** Enable the components in every layer of the architectural stack to operate as one while extracting the full value of each component.
- **Access.** Easily get data to where applications need it, when they need it, in a way they can use it.
- **Consistency.** Manage data across multiple environments using common tools and processes regardless of where it resides.

The NetApp Data Fabric empowers customers to successfully meet the challenge of digital transformation by giving you the ability to:

- Harness the power of the hybrid cloud
- Build a next-generation data center
- Modernize storage through data management

Harness the Hybrid Cloud

Freely and securely manage your data—anywhere, on any cloud. Simplify and integrate your data management across cloud and on-premises to accelerate digital transformation.

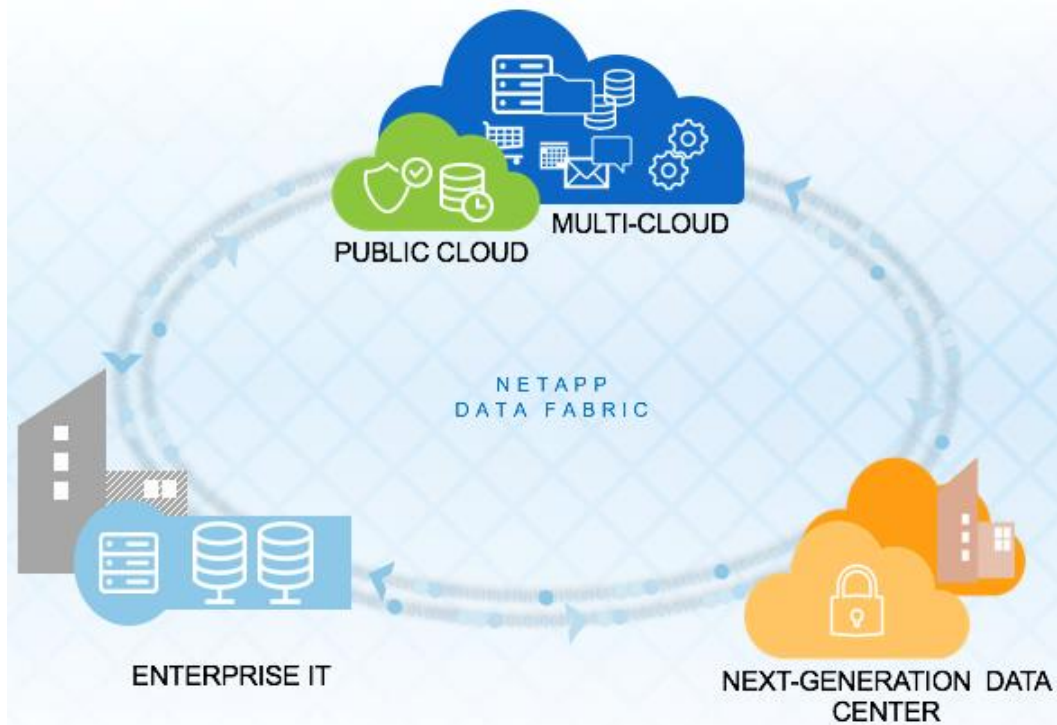
Build a Next-Generation Data Center

Get the scale and quality of service that modern applications require. Deliver consistent and integrated data management services and applications for data visibility and insights, data access and control, and data protection and security.

Modernize Storage Through Data Management

Upgrade infrastructure to bring modern data services to your existing applications. Unleash the power of your data to achieve new competitive advantages.

Figure 12) NetApp Data Fabric – helping you to transform seamlessly to a next-generation data center.



7.2 Cloud Volumes ONTAP

Cloud Volumes ONTAP runs in a hyperscaler cloud environment such as Amazon Web Services (AWS), Azure, and Google Cloud Platform (GCP), bringing intelligence and Data Fabric connectivity to hyperscaler storage volumes. The overall best practices for running SQL Server on Cloud Volumes ONTAP are generally the same as for SQL Server on ONTAP. The primary considerations specific to SQL Server on Cloud Volumes ONTAP surround performance, and to a lesser extent, cost.

Cloud Volumes ONTAP is partially limited by the performance of the underlying volumes managed by the cloud provider. The result is more manageable storage, and, sometimes, the caching capability of Cloud Volumes ONTAP offers a performance improvement. However, there are always some limitations in terms of IOPS and latency due to the reliance on public cloud provider storage.

The primary use cases for Cloud Volumes ONTAP are currently DR, development, and testing efforts. However, some customers have used Cloud Volumes ONTAP for production activity as well.

Benefit of Cloud Volumes ONTAP with SQL Server

Cloud Volumes ONTAP offers advanced data management that enhances service levels, saves time for IT and DevOps, and reduces storage management and associated costs. The following are benefits of using Cloud Volumes ONTAP with SQL Server:

- **Cost savings with storage efficiencies.** Cloud Volumes ONTAP can help you to save up to 90% on storage capacity with space-efficiency technologies such as data deduplication, compression, thin-cloning, and Snapshot copies that don't affect storage footprint.
- **High availability.** Achieve high availability with a two-node solution that supports multiple Availability Zones and enables business continuity for your critical production workloads and databases with no data loss (RPO=0) and short recovery times (RTO < 60 secs).
- **Data protection and disaster recovery.** Recover from data corruption or loss with efficient data Snapshot and disaster recovery copies, which are easily configured, cost effective, and support seamless failover, failback, restore, and recovery processes that meet minute-level SLAs.
- **Hybrid and multicloud environments.** Save time and money by using the same storage and advanced ONTAP data management software across hybrid and multicloud environments, including DR, HA, dev/test and DevOps, sandbox, reporting, data tiering, workload hosting, and training.
- **Data mobility.** Migrate, replicate, and synchronize your data securely, leveraging efficient data Snapshot copies to transfer only incremental changes and recover from any point in time by using NetApp SnapMirror.
- **Cloning technology for developers.** Increase DevOps agility by cloning writable volumes from Snapshot copies so that data can be shared simultaneously across organizations and regions with zero capacity and performance penalties using FlexClone. These processes can be done by using SnapCenter to provide application consistency.
- **Interoperability.** Leverage multiprotocol support (iSCSI and SMB) for your data and file shares and meet the demands of SQL Server workloads.
- **Flexible licensing.** There are multiple Cloud Volumes ONTAP solutions from hourly priced options, longer-term subscriptions, and Bring Your Own License (BYOL) options.
- **Enhanced security.** In addition to security and privacy features offered by the hyperscaler, Cloud Volume ONTAP provides NetApp managed encryption, which gives you the capability to manage encryption keys on your premises.

For more information about Cloud Volume ONTAP, go to <https://cloud.netapp.com/ontap-cloud>.

Planning and Licensing

The most important aspect of Cloud Volumes ONTAP planning is that you must first define the requirements of the database in terms of performance, capacity, and availability. Next, you must reduce to practice the required configuration given the capabilities and limitations of the virtual environment hosting Cloud Volumes ONTAP and the hosts using Cloud Volumes ONTAP. Finally, you must assess the costs of the complete cloud environment.

Considerations include:

- What protocols are you going to use considering that SQL Server supports both block and file protocols?
- How many random IOPS are required?
- How much bandwidth is required between the database server and Cloud Volumes ONTAP?
- What are the license requirements?
- What are the backup, recovery, and availability SLAs?
- How often is the cloud environment active?
- What are the backup and recovery requirements?

Cloud Volumes ONTAP Licensing

Cloud Volumes ONTAP is licensed in one of two ways: on-demand through a hyperscaler cloud provider or as a BYOL model.

The detailed pricing information is available on the [AWS](#) and [Microsoft Azure](#) marketplaces. If the on-demand options are not suitable for your requirements, Cloud Volumes ONTAP licenses can be purchased directly from NetApp or a NetApp partner.

Performance

Cloud Volumes ONTAP uses a collection of hyperscaler volumes to provide advanced capabilities such as backup and recovery, database cloning, and disaster recovery services based on Snapshot copies. Cloud Volumes ONTAP cannot improve performance much beyond the capabilities of the underlying hyperscaler volumes. Cloud Volumes ONTAP provides a cache to assist a highly cacheable workload such as highly repeated read workload. In general, however, database performance of the underlying hyperscaler volumes is neither improved nor reduced by the use of Cloud Volumes ONTAP. Exceptions include unusual workloads or when the overall configuration is not optimal.

Detailed performance information of Cloud Volumes ONTAP is available on For detailed performance information of Cloud Volumes ONTAP, see [Performance Characterization of Cloud Volumes ONTAP in AWS](#) and [Performance Characterization of Cloud Volumes ONTAP in Azure](#).

Backup and Clone for Cloud Volumes ONTAP

SnapCenter software is a unified, scalable platform for data protection. SnapCenter provides centralized control and oversight, and enables users to manage application-specific backup, restore, and clone operations. By using SnapCenter, database, storage, and virtualization administrators learn a single tool to manage backup, restore, and clone operations for various applications, databases, and VMs.

You can use SnapMirror to replicate SQL Server databases between on-premises environments and the cloud, and between private, hybrid, or public clouds.

In a hybrid cloud environment (with FAS or AFF on premise replicating to Cloud Volumes ONTAP), the mirror relationship can be created using OnCommand Cloud Manager. For more information about OnCommand Cloud Manager, see the [Cloud Manager and Cloud Volumes ONTAP documentation](#).

SnapCenter enables centralized application resource management and easy data protection job execution by using resource groups and policy management (including scheduling and retention settings). SnapCenter provides unified reporting by using a dashboard, multiple reporting options, job monitoring, and log and event viewers.

SnapCenter includes the following key features:

- A unified and scalable platform across applications and database environments, and virtual and nonvirtual storage, powered by SnapCenter Server
- Role-based access control (RBAC) for security and centralized role delegation
- Application-consistent Snapshot copy management, restore, clone, and backup verification support from both primary and secondary destinations (SnapMirror and SnapVault)
- Remote plug-in package installation from the SnapCenter UI
- Nondisruptive, remote upgrades
- A dedicated SnapCenter repository that provides fast data retrieval
- Load balancing implemented using Microsoft Windows Network Load Balancing (NLB) and Application Request Routing (ARR), with support for horizontal scaling
- Centralized scheduling and policy management to support backup and clone operations
- Centralized reporting, monitoring, and dashboard views

SnapCenter leverages technologies, including Snapshot, SnapMirror, SnapRestore software, and FlexClone, which allow it to integrate seamlessly with technologies offered by SQL Server across iSCSI protocol for Cloud Volumes ONTAP. More information of SnapCenter Plug-in for Microsoft SQL Server is available on [Best Practice Guide for SQL Server using NetApp SnapCenter](#). For SnapCenter and SQL Server video resources, see the following links:

- [Setup SnapCenter 4.0 for SQL Server plug-in](#)
- [How to clone a database using SnapCenter with SQL Server plug-in](#)
- [How to back up and restore databases using SnapCenter with SQL Server Plug-in](#)
- [Building SQL Server Failover Cluster Instance with SnapCenter for SQL Server Plug-in](#)

7.3 Cloud Volumes Service

NetApp Cloud Volumes Service is a fully managed cloud storage solution that is available for Amazon Web Services (AWS) and Google Cloud Platform (GCP). For Microsoft Azure, NetApp and Microsoft have created a first-party service called Azure NetApp Files that offers a high-performance file-service environment for the Microsoft Azure cloud. These services support NFS v3, NFSv4, and SMB, can grow to 100TB, and run file shares at peak performance while providing the best data protection and security in an organization's preferred cloud provider.

Key benefits are as follows:

- Maintain business continuity with no data loss, fast failover, short recovery times, and nondisruptive upgrade processes.
- Schedule tasks to meet the most demanding file share needs with automation and orchestration capabilities.

The NetApp Cloud Volumes Service is a great fit for developers, line-of-business engineers, database administrators, and application architects who consume storage capacity but don't want to administer it. IT and cloud architects can also use this service to search of file services in the cloud.

NetApp Cloud Volumes Service for AWS

Cloud Volumes Service for AWS is purchased directly through the AWS Marketplace, giving customers a single payment model through Amazon Web Services and an easy-to-use interface for development. IT leaders get transparent reach into premier cloud data services to use NetApp file services expertise. At the same time, cloud architects and development teams can provision everything using REST APIs, automate it, and take it to a larger scale.

NetApp Cloud Volumes Service for AWS is an optimal fit for those who run NFS or SMB workloads and applications that leverage file services for data storage, such as SQL Server, Oracle, or SAP.

Deploy SQL Server with Cloud Volumes Service

SQL Server can be deployed on Cloud Volumes Service with SMB protocol. The advantages of using SQL Server over SMB are as follows:

- Storage can be easily managed and modified.
- DBAs or application owners can manage the storage without going through sysadmin or storage admin.
- Improved data security. Only SQL Server service account can access the volumes and data can be encrypted in-flight.
- Performance comparable with on-premises databases.

When specifying the SMB file share, the following are supported Universal Naming Convention (UNC) Path formats for standalone and Always On Availability Groups databases:

- \\ServerName\ShareName\
- \\ServerName\ShareName

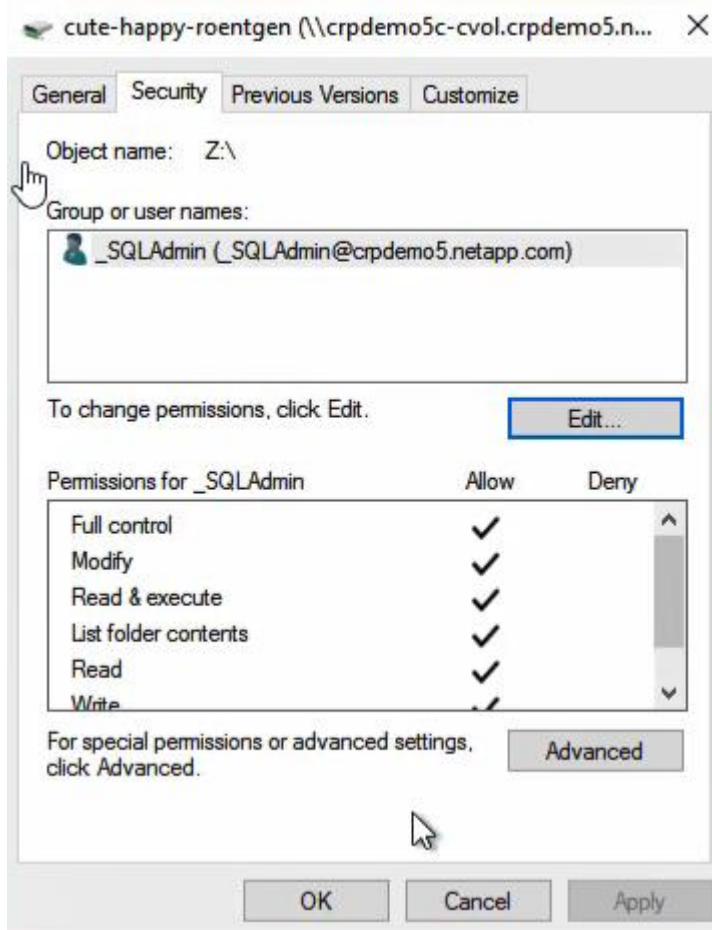
For more information about UNC, see <https://msdn.microsoft.com/library/gg465305.aspx>.

To deploy SQL Server over SMB with Cloud Volumes Service, NetApp recommends the following best practices:

- Managed service accounts or group-managed service accounts
 - DOMAIN\ACCOUNTNAME\$
 - Grant full control for SQL Server service account
- **Do not use** virtual accounts such as NT SERVICE\<SERVICENAME>

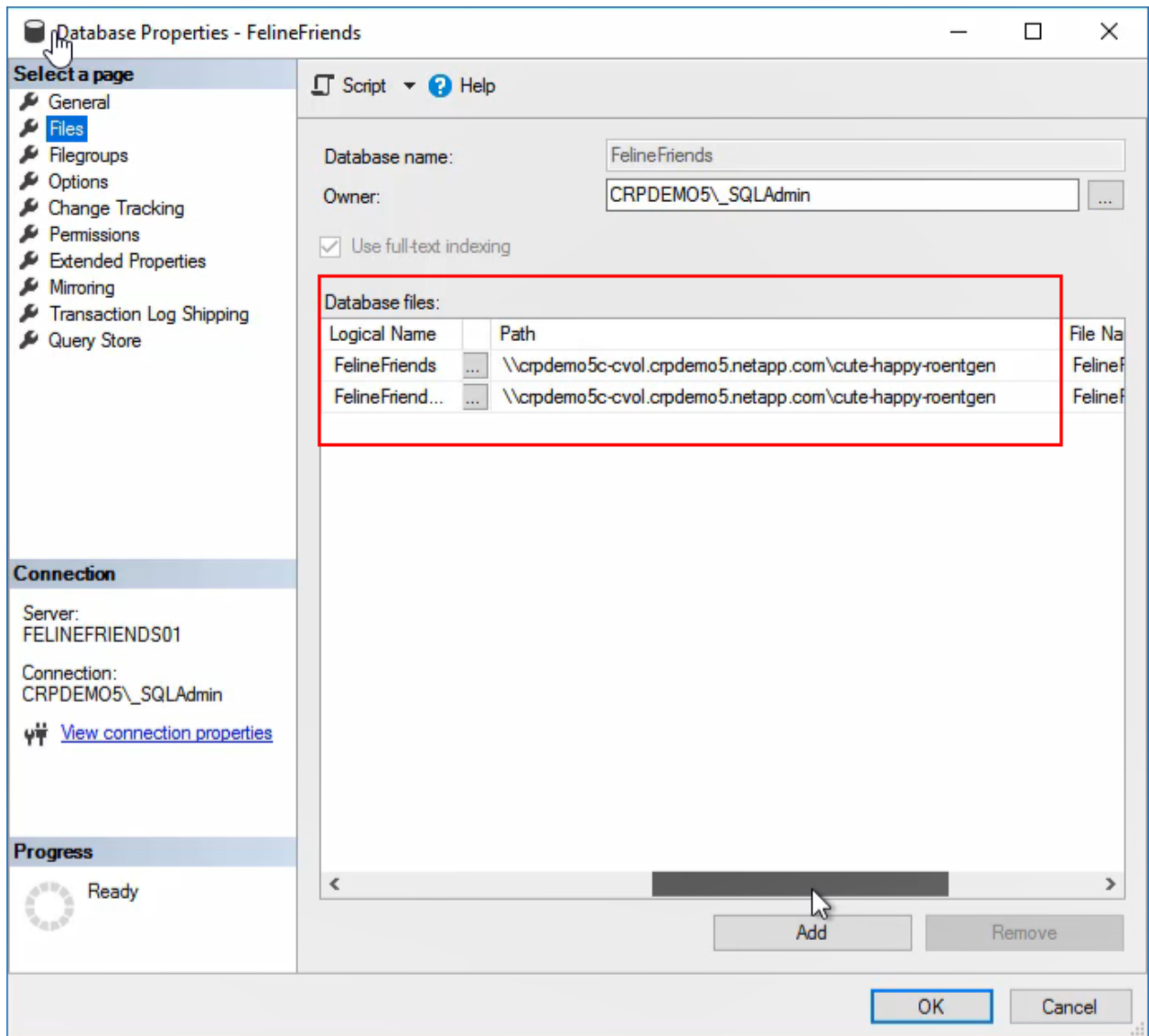
After provisioning Cloud Volumes Service, the Everyone account typically has access to the volumes. NetApp recommends assigning SQL Server service to have full access to the volumes and removing the Everyone account permissions from the volumes.

Figure 13) Assigning full access permissions for the SQL Server Service Account.



After creating the volumes and assigning proper permissions to the volumes, DBAs or application owners can place databases on the volumes using SQL Server Management Studio or T-SQL script.

Figure 14) Example of deploying SQL Server data and log files to Cloud Volumes Service with SMB protocol.



SQL Server known issues for SMB shares:

- After you detach a SQL Server 2017 database that resides on NAS, you might run into database permission issue while trying to reattach the SQL Server database. The issue is defined in <https://go.microsoft.com/fwlink/?LinkId=237321>. To work around this issue, see the More Information section in the KB article.
- Currently Cloud Volumes Service does not support Failover Cluster Instance (FCI) installation.

For the detailed procedures to deploy SQL Server with Cloud Volumes Service, see [Deploy SQL Server Over SMB on Cloud Volumes Service](#).

Selecting the Appropriate Service Level and Allocated Capacity

The cost for Cloud Volumes Service for AWS is based on the service level and the allocated capacity that you select. Selecting the appropriate service level and capacity helps you to meet your storage needs at the lowest cost.

Considerations

When assessing your storage needs, consider two fundamental aspects:

- The storage capacity for holding data
- The storage bandwidth for interacting with data

If you consume more storage space than the capacity you selected for the volume, the following considerations apply:

- You will be billed for the additional storage capacity that you consume at the price defined by your service level.
- The amount of storage bandwidth available to the volume does not increase until you increase the allocated capacity size or change the service level.

Service Levels

NetApp Cloud Volumes Service for AWS supports three service levels. You specify your service level when you create or modify the volume.

The service levels cater to different storage capacity and storage bandwidth needs:

- **Standard (capacity)**
If you want capacity at the lowest cost, and your bandwidth needs are limited, the Standard service level might be most appropriate for you. An example is when you want to use the volume as a backup target.
 - Bandwidth: 16KB of bandwidth per GB provisioned capacity
- **Premium (a balance of capacity and performance)**
If your application has a balanced need for storage capacity and bandwidth, the Premium service level might be most appropriate for you. This level is less expensive per MBps than the Standard service level, and it is also less expensive per GB of storage capacity than the Extreme service level.
 - Bandwidth: 64KB of bandwidth per GB provisioned capacity
- **Extreme (performance)**
The Extreme service level is least expensive in terms of storage bandwidth. If your application demands storage bandwidth without the associated demand for large storage capacity, the Extreme service level might be most appropriate for you.
 - Bandwidth: 128KB of bandwidth per GB provisioned capacity

Allocated Capacity

You must specify your allocated capacity for the volume when you create or modify the volume. Although you would select your service level based on your general, high-level business needs, you should select your allocated capacity size based on the specific needs of applications, for example:

- The storage space required by the applications
- The storage bandwidth per second required by the applications or the users

Allocated capacity is specified in GBps. A volume's allocated capacity can be set within the range of 1GB to 100,000GB (equivalent to 100TBps).

Bandwidth

The combination of the service level and allocated capacity that you select determines the maximum bandwidth for the volume.

If your applications or users need more bandwidth than your selections, you can change the service level or increase the allocated capacity. The changes do not disrupt data access.

Selecting the Service Level and the Allocated Capacity

To select the most appropriate service level and allocated capacity for your needs, you need to know how much capacity and bandwidth you require at the peak or the edge.

Cost Comparison for Service Levels and Allocated Capacity

This [link](#) compares cost for different service levels and allocated capacity sizes. In the table, the first column indicates the capacity; the other columns define the MBps available at each capacity point and its cost.

Translate IOPS to Throughput

As mention previously, Cloud Volumes Service levels are defined by throughput. However, most applications require defining throughput in IOPS. The following formula helps you to convert IOPS to throughput with SQL Server workload.

Required bandwidth = (IOPS * KB per I/O) ÷ 1024

Because SQL Server operates at the minimum of 8KB (one page), the I/O KB that will be used in the previous formula will be 8. The following is the example of how to translate IOPS to throughput:

Assume that the application needs 16,000 IOPS.

Because SQL Server operates minimum of 8K, the application needs 16,000 x 8KB which equal to 128,000KBps

Convert KB to MB by dividing by 1024: 128,000 ÷ 1024 = 125MBps

Therefore, application needs 125MBps

To find out what allocate capacity the application needs, the lookup table from this [link](#) must be created. In the previous example, the capacity that needs to be provisioned will be 8, 2, and 1TB, if the Service Level has been selected as Standard, Premium, and Extreme.

Table 4) Example of quotas and service level for workload with 16,000 IOPS.

Capacity	Standard (MBps)	Premium (MBps)	Extreme (MBps)
1	16	64	128
2	32	128	250
3	48	192	384
4	64	256	512
5	80	320	640
6	96	384	768
7	112	448	896
8	128	5212	1024

Because this task involves calculation and lookup for the Service Level table, a PowerShell script has been developed to assist you with this task and deploy Cloud Volumes Service for your account. The PowerShell script is available in the appendix.

Deploy High Availability with Cloud Volumes Service

SQL Server provides several options for creating high availability for a server or database. The high-availability options include the following:

- Always On Failover Cluster Instances

- As part of the SQL Server Always On offering, Always On FCI uses Windows Server Failover Clustering (WSFC) functionality to provide local high availability through redundancy at the server-instance level—an FCI. An FCI is a single instance of SQL Server that is installed across the WSFC nodes and, possibly, across multiple subnets. On the network, an FCI appears to be an instance of SQL Server running on a single computer, but the FCI provides failover from one WSFC node to another if the current node becomes unavailable.

Note: Currently the Always On FCI is not supported for Cloud Volumes Service.

- Always On Availability Groups

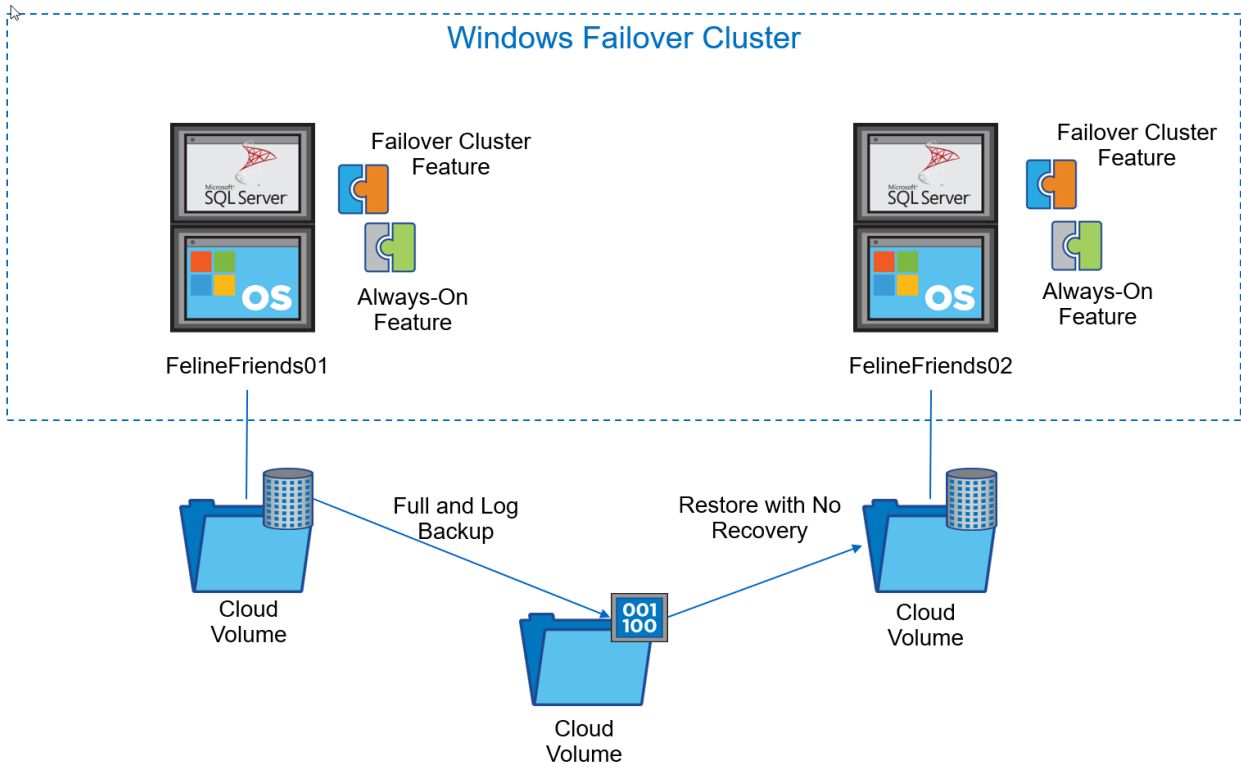
Always On Availability Groups is an enterprise-level high-availability and disaster recovery solution introduced in SQL Server 2012 (11.x) to enable you to maximize availability for one or more user databases. Always On Availability Groups requires that the SQL Server instances reside on WSFC nodes.

For information about Always On Availability Groups for SQL Server, see [Overview of Always On Availability Groups \(SQL Server\)](#).

To set up Always On Availability Groups with Cloud Volumes Service, complete the following steps:

1. Create the Windows Failover Cluster.
2. Enable the Always On feature for SQL Server on all nodes.
3. Create the full and log backup database.
4. Restore the databases to the replica node by using the `norecovery` option.
5. Create Always On Availability Groups and add databases by using the `join only` option.

Figure 15) SQL Server Always On with Cloud Volumes Service.



For instructions to deploy Always On Availability Groups Over SMB on Cloud Volumes Service, see [Deploy Always On Availability Groups Over SMB on Cloud Volumes Service](#).

8 Conclusion

SQL Server users typically face a series of significant challenges in their effort to increase the return on their SQL Server investments and optimize their infrastructure to support business and IT requirements. They must:

- Accelerate new database implementations or migrations and lower the risk of these operations.
- Make sure that the underlying storage infrastructure is fully optimized to support SLAs, including performance, scalability, and availability.
- Consolidate existing databases and infrastructure to lower costs.
- Reduce complexity and simplify IT infrastructure.
- Increase the productivity of IT personnel.

To succeed in these challenges, the architects, sysadmins, or DBAs are looking to deploy their databases and storage infrastructure based on proven best practices and technology.

This document covers NetApp's recommendations for designing, optimizing, and scaling Microsoft SQL Server deployments, which can vary greatly between implementations. Options such as cluster awareness and virtualization introduce further variables. The right solution depends on both the technical details of the implementation and the business requirements driving the project.

This document gives common recommendations in the following areas:

- SQL Server workload type
- SQL Server configuration
- Database storage layout
- Storage efficiency

It also introduces deploying SQL Server in the Data Fabric. Currently, there are two products that available for SQL Server:

- Cloud Volumes ONTAP
- Cloud Volumes Service

SQL Server databases can be quickly and easily protected using NetApp SnapCenter software or a combination of SnapCenter Plug-in for Microsoft Windows and SnapManager for SQL Server. These products enable application-consistent backup, automated cloning, and restore and recovery of SQL Server databases, instances, or availability groups.

NetApp and partner professional services experts are available for assistance in complex projects. Even if assistance is not required during the project, NetApp strongly encourages new customers to use professional services for assistance in developing a high-level approach.

Appendix

The following is an example PowerShell script to convert IOPS to throughput, look up Cloud Volumes Service, and deploy Cloud Volumes Service.

```
# +-----+
# | File      : DeployCloudVol.PS1
# | Version  : 1.0
# | Purpose  : Pull Cloud Volume info from web to better assist in volume deployment
# |           based on user input of necessary throughput or IOPS
# | Usage    : .\DeployCloudVol.ps1
# +-----+
# | Maintenance History
# | -----
# | Name           Date [YYYY-MM-DD]      Version      Description
```

```

# | -----
# | Alex Lopez          2018-09-26          1.0          Initial release
# | -----
# *****

# Documentation
# http://nfsaas.runarberg.test/docs#
# https://docs.netapp.com/us-en/cloud_volumes/aws/reference_cloud_volume_apis.html

# *****
# Globals
# *****

# env variables
# # https://cds-aws-bundles.netapp.com x2 <--> http://my.runarberg.test/login
$URI = "http://nfsaas.runarberg.test/v1/"
$apiKey = "cFlqbWVSblg4Q1kyNTJUbtJSWmY4d2ZVY1RzdHhyXXXX" #
$secretKey = "V3B3aHBCMEc1RwtWZF1McjcwelBnNkN1dWZkbm03XXXX" #
$region = "us-east-1"

$headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
$headers.Add("API-KEY", $apiKey)
$headers.Add("SECRET-KEY", $secretKey)

# *****
# GET Functions
# *****

Function GetFileSystems ($search) {
    Invoke-RestMethod -Method Get -Uri ($URI+"FileSystems") -Headers $headers
}

Function GetFileSystemID {
    (Invoke-RestMethod -Method Get -Uri ($URI+"FileSystems") -Headers $headers | Where-Object {
    $_.name -like '*orourke*' }).filesystemId
}

Function GetActiveDirectory {
    Invoke-RestMethod -Method Get -Uri ($URI+"Storage/ActiveDirectory") -Headers $headers
}

Function GetAllJobs {
    Invoke-RestMethod -Method Get -Uri ($URI+"Jobs") -Headers $headers
}

Function GetAllPools {
    Invoke-RestMethod -Method Get -Uri ($URI+"Pools") -Headers $headers
}

Function GetMountTargets {
    $fsID = GetFileSystemID | Select-Object -First 1 # if more than one, change as necessary
    Invoke-RestMethod -Method Get -Uri ($URI+"FileSystems/"+$fsID+"/MountTargets") -Headers
$headers
}

Function GetWebTables {
    $tableURL = "https://docs.netapp.com/us-
en/cloud_volumes/aws/reference_selecting_service_level_and_quota.html"
    $page = Invoke-WebRequest $tableURL
    $tables = @($page.ParsedHtml.IHTMLDocument3_getElementsByTagName("TABLE"))
    $table = $tables[2]
    $titles = @('Capacity (TB)', 'Standard (MB/s)', 'Cost1', 'Premium (MB/s)', 'Cost2', 'Extreme
(MB/s)', 'Cost3')
    $rows = @($table.Rows)

    foreach($row in $rows | select -skip 2)
    {
        $cells = @($row.Cells)
        $responseObject = [Ordered] @{}
    }
}

```

```

    for($counter = 0; $counter -lt $cells.Count; $counter++)
    {
        $title = $titles[$counter]
        if(-not $title) { continue }
        if ($cells[$counter].InnerText -contains '*$*') {
            # $responseObject[$title] = (" " + $cells[$counter+1].InnerText).Trim()
        }
        else {
            $responseObject[$title] = (" " + $cells[$counter].InnerText).Trim()
        }
    }

    ## Cast hashtable to a PSCustomObject
    [PSCustomObject] $responseObject
    }
}

# In order to export as CSV:
# GetWebTables | Format-Table -Property 'Capacity (TB)', 'Standard (MB/s)', 'Premium
(MB/s)', 'Extreme (MB/s)' # | Export-Csv -NoTypeInfo -Path 'C:\Your\path\here'

# *****
# POST Functions
# *****

# the following example uses the $filesystem string, formatted as json, which is sent through the
body of the POST request
Function CreateCloudVol ($volName, $creationToken, $quota, $serviceLevel, $protocol) {
    $filesystem = '
    {
        "name": "' + $volName + '",
        "region": "' + $region + '",
        "backupPolicy": {
            "dailyBackupsToKeep": 7,
            "enabled": false,
            "monthlyBackupsToKeep": 12,
            "weeklyBackupsToKeep": 52
        },
        "creationToken": "' + $creationToken + '",
        "jobs": [
            {}
        ],
        "labels": [
            "API"
        ],
        "poolId": "",
        "protocolTypes": [' + $protocol + '],
        "quotaInBytes": ' + $quota + ',
        "serviceLevel": "' + $serviceLevel + '",
        "smbShareSettings": [
            "encrypt_data"
        ],
        "snapReserve": 20,
        "snapshotPolicy": {
            "dailySchedule": {
                "hour": 23,
                "minute": 10,
                "snapshotsToKeep": 7
            },
            "enabled": false,
            "hourlySchedule": {
                "minute": 10,
                "snapshotsToKeep": 24
            },
            "monthlySchedule": {
                "daysOfMonth": "1,15,31",
                "hour": 23,
                "minute": 10,
                "snapshotsToKeep": 12
            }
        }
    }
    '
}

```

```

        },
        "weeklySchedule": {
            "day": "Saturday,Sunday",
            "hour": 23,
            "minute": 10,
            "snapshotsToKeep": 52
        }
    }
}'
# $filesystem
Invoke-RestMethod -Method Post -Uri ($URI+"FileSystems") -Headers $headers -Body $filesystem
-ContentType 'application/json'
}
# CreateCloudVol 'API Test Vol' 'api-nfs-volume' '30000000000' 'extreme' '"NFSv3"'

# protocol param options
# '"NFSv3"'
# '"CIFS"'
# '"NFSv3", "CIFS"'

# *****
# PUT Functions
# *****

# update fs: you can make changes to the name, serviceLevel, quotaInBytes, snapReserve,
snapshotPolicy, backupPolicy, and exportPolicy
# the following example uses a powershell hashtable object and converts to json before sending
the put method
Function UpdateFilesystem ($filesystemID, $SLO, $quota) {
    $filesystem = @{
        name = "FelineFriends01_data"
        quotaInBytes = $quota
        serviceLevel = $SLO
        region = $region
        backupPolicy = @{
            dailyBackupsToKeep = 7
            enabled = $false
            monthlyBackupsToKeep = 12
            weeklyBackupsToKeep = 52
        }
        creationToken = "amazing-backstabbing-mcnulty"
        snapReserve = 20
        snapshotPolicy = @{
            enabled = $false
            dailySchedule = @{
                hour = 23
                minute = 10
                snapshotsToKeep = 7
            }
            hourlySchedule = @{
                minute = 10
                snapshotsToKeep = 7
            }
            monthlySchedule = @{
                daysOfMonth = "1,15,31"
                hour = 23
                minute = 10
                snapshotsToKeep = 7
            }
            weeklySchedule = @{
                day = "Saturday,Sunday"
                hour = 23
                minute = 10
                snapshotsToKeep = 7
            }
        }
    }
    $json = $filesystem | ConvertTo-Json #-Depth 4
    Invoke-RestMethod -Method Put -Uri ($URI+"FileSystems/"+$filesystemID) -Headers $headers -
    Body $json -ContentType 'application/json'
}

```

```

# UpdateFilesystem $global:fsID 'standard' 3000000000

# *****
# User Input Functions
# *****

Function GetInput {
    # store table locally
    $table = @(GetWebTables)
    do {
        do {
            Write-Host "Enter IOPS or Throughput"
            [int]$userIOPS = Read-Host -Prompt 'IOPS (if unknown, hit enter)'
            if ($userIOPS) {
                [ValidateRange(4,128)][int]$userBlock = Read-Host -Prompt 'Block size (in KB,
values 4 - 128)'
                # TP = (IOps * block size [kb]) / 1024
                $userTP = ($userIOPS * $userBlock) / 1024
            }
            else {
                [ValidateRange(0,3500)]$userTP = Read-Host -Prompt 'ThroughPut (in MB/s, values
16 - 3,500)'
            }
            # display offered configs based on table
            if (!$userIOPS -and !$userTP) {
                Write-Host 'Please enter a value' -ForegroundColor Yellow
                $continue = $false
            }
            else {
                $continue = $true
            }
        } while ($continue -eq $false)

        # $table | Where-Object {[int]$_.'Standard (MB/s)' -ge 900} | Select -first 1 | Format-
Table -Property 'Capacity (TB)', 'Standard (MB/s)'
        # $table | Where-Object {[int]$_.'Premium (MB/s)' -ge 900} | Select -first 1 | Format-
Table -Property 'Capacity (TB)', 'Premium (MB/s)'
        # $table | Where-Object {[int]$_.'Extreme (MB/s)' -ge 900} | Select -first 1 | Format-
Table -Property 'Capacity (TB)', 'Extreme (MB/s)'

        # store config options based on input
        $standardTP = $table | Where-Object {[int]$_.'Standard (MB/s)' -ge $userTP} | Select -
ExpandProperty 'Standard (MB/s)' -First 1;
        $standardCapacity = $table | Where-Object {[int]$_.'Standard (MB/s)' -ge $userTP} |
Select -ExpandProperty 'Capacity (TB)' -First 1;

        $premiumTP = $table | Where-Object {[int]$_.'Premium (MB/s)' -ge $userTP} | Select -
ExpandProperty 'Premium (MB/s)' -First 1;
        $premiumCapacity = $table | Where-Object {[int]$_.'Premium (MB/s)' -ge $userTP} | Select
-ExpandProperty 'Capacity (TB)' -First 1;

        $extremeTP = $table | Where-Object {[int]$_.'Extreme (MB/s)' -ge $userTP} | Select -
ExpandProperty 'Extreme (MB/s)' -First 1;
        $extremeCapacity = $table | Where-Object {[int]$_.'Extreme (MB/s)' -ge $userTP} | Select
-ExpandProperty 'Capacity (TB)' -First 1;

        # create a new table, add in above values and format
        $newTable = New-Object system.Data.DataTable "Volume Options"
        $quotaColumn = New-Object system.Data.DataColumn 'Quota (TB)', ([string])
        $tpColumn = New-Object system.Data.DataColumn 'Throughput (MB/s)', ([string])
        $slColumn = New-Object system.Data.DataColumn 'Service Level', ([string])

        $newTable.Columns.Add($slColumn)
        $newTable.Columns.Add($quotaColumn)
        $newTable.Columns.Add($tpColumn)

        $standardRow = $newTable.NewRow()
        $premiumRow = $newTable.NewRow()
        $extremeRow = $newTable.NewRow()

```

```

$standardRow.'Service Level' = '1) Standard'
$standardRow.'Quota (TB)' = $standardCapacity
$standardRow.'Throughput (MB/s)' = $standardTP
$newTable.Rows.Add($standardRow)

$premiumRow.'Service Level' = '2) Premium'
$premiumRow.'Quota (TB)' = $premiumCapacity
$premiumRow.'Throughput (MB/s)' = $premiumTP
$newTable.Rows.Add($premiumRow)

$extremeRow.'Service Level' = '3) Extreme'
$extremeRow.'Quota (TB)' = $extremeCapacity
$extremeRow.'Throughput (MB/s)' = $extremeTP
$newTable.Rows.Add($extremeRow)

# output table
Write-Host 'Available configurations based on input:'
$newTable | Out-Host

# deploy volume loop
[ValidateRange(1,3)]$userSLOnum = Read-Host -Prompt 'Select a Service Level from the
above table [1, 2, 3]'
Switch ($userSLOnum) {
    1 { $userSLO = "basic"; [double]$userQuota = $standardCapacity } # basic --> standard
    2 { $userSLO = "standard"; [double]$userQuota = $premiumCapacity} # standard -->
premium
    3 { $userSLO = "extreme"; [double]$userQuota = $extremeCapacity } # extreme -->
extreme
}

# get desired protocol

[ValidateRange(1,3)]$protocol = Read-Host -Prompt 'Select a protocol [NFSv3 - 1, SMB - 2,
Dual - 3]'
Switch ($protocol) {
    1 { $userProtocol = "NFSv3" }
    2 { $userProtocol = "CIFS" }
    3 { $userProtocol = "NFSv3", "CIFS" }
}

# get name and number of volumes
$userVolName = Read-Host -Prompt 'Give this volume a name'
[ValidateRange(0,25)][int]$numVols = Read-Host -Prompt 'How many of these volumes would
you like to create?'

# convert the quota (TB) to bytes
[double]$quotaInBytes = $userQuota * 1000000000000

if ($numVols -lt [int]'2') {
    # if input is 0 or 1, create 1 volume
    CreateCloudVol $userVolName "api-vol-$userVolName" $quotaInBytes $userSLO
$userProtocol
}
else {
    for ($i=0; $i -lt $numVols; $i++) {
        Write-Host "Creating volume number" ($i+1) -BackgroundColor Green
        CreateCloudVol ($userVolName+$i) ("api-vol-$userVolName"+$i)
$quotaInBytes $userSLO $userProtocol
    }
}
$loopYN = Read-Host "Would you like to create more volumes? (Y/N)"
} while ($loopYN -eq 'y')
}

GetInput

# Notes

# 30000000000 B is 30GB
# 1000000000000 B is 1TB

```



```

# todo
# nicer formatting
# change hardcoded api keys to pull from text file, json, or .csv
# add gui -- may use

<#
$form = New-Object "System.Windows.Forms.Form";
$form.Width = 500;
$form.Height = 150;
$form.Text = $title;
$form.StartPosition = [System.Windows.Forms.FormStartPosition]::CenterScreen;
#>

Add-Type -AssemblyName System.Windows.Forms

$form = New-Object system.Windows.Forms.Form
$form.Text = "Cloud Volumes"
$form.TopMost = $true
$form.Width = 400
$form.Height = 400

# buttons
$okButton = New-Object system.Windows.Forms.Button
$okButton.Text = "OK"
$

$cancelButton = New-Object system.Windows.Forms.Button
$cancelButton.Text = "Cancel"

# labels and textboxes
$iopsTB = New-Object system.Windows.Forms.TextBox
$blocksizeTB = New-Object system.Windows.Forms.TextBox
$throughputTB = New-Object system.Windows.Forms.TextBox

# From here, display tables in the form and allow the user to select from the 3 options
# once selection has been made, clear those tables from view (as well as whatever TB was holding
them
#     and present the name, size, and number of volumes options

$nameTB = New-Object system.Windows.Forms.TextBox
$sizeTB = New-Object system.Windows.Forms.TextBox
$numVolsTB = New-Object system.Windows.Forms.TextBox

# example commands below
$getDate = New-Object system.Windows.Forms.Button
$getDate.Text = "Get Date"
$getDate.Width = 100
$getDate.Height = 30
$getDate.Add_Click({
#add here code triggered by the event
    $results.Text = Get-Date
})
$getDate.Location = new-object system.drawing.point(60,60)
$getDate.Font = "Microsoft Sans Serif,10"
$form.Controls.Add($getDate)

$results = New-Object system.Windows.Forms.TextBox
$results.Text = ""
$results.Width = 100
$results.Height = 20
$results.Location = new-object system.drawing.point(40,23)
$results.Font = "Microsoft Sans Serif,10"
$form.Controls.Add($results)

[void]$form.ShowDialog()
$form.Dispose()

```

Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and/or websites:

- [NetApp Interoperability Matrix Tool](#)
- NetApp Product Documentation:
docs.netapp.com
- TPC
<http://www.tpc.org/>
- HammerDB
<http://www.hammerdb.com/>
- Compute Capacity Limits by Edition of SQL Server
<https://msdn.microsoft.com/en-us/library/ms143760.aspx>
- SQL Server 2017 Editions
<https://www.microsoft.com/en-us/sql-server/sql-server-2017-editions>
- Configure the max worker threads Server Configuration Option
<https://msdn.microsoft.com/en-us/library/ms190219.aspx>
- NetApp AFF8080 EX Performance and Server Consolidation with Microsoft SQL Server 2014
<https://fieldportal.netapp.com/content/248568?assetComponentId=248696>
- Page Compression Implementation
<https://msdn.microsoft.com/en-us/library/cc280464.aspx>
- TR-4015: SnapMirror Configuration Best Practices ONTAP 9.1 and 9.2
<https://fieldportal.netapp.com/content/623586?assetComponentId=624809>
- TR-4733 SnapMirror Synchronous for ONTAP 9.5
<https://fieldportal.netapp.com/content/835406?assetComponentId=837083>
- TR-4383: Performance Characterization of Cloud Volumes ONTAP in AWS
<https://www.netapp.com/us/media/tr-4383.pdf>
- TR-4671: Performance Characterization of Cloud Volumes ONTAP in Azure
<https://www.netapp.com/us/media/tr-4671.pdf>
- Cloud Manager and Cloud Volumes ONTAP documentation
<https://docs.netapp.com/us-en/occm/index.html>
- Best Practice Guide for SQL Server using NetApp SnapCenter
<https://fieldportal.netapp.com/content/783217?assetComponentId=784811>
- Cloud Volumes ONTAP Resources Page
<https://www.netapp.com/us/documentation/cloud-volumes-ontap-and-occm.aspx>
- ONTAP Select Resources Page
<https://www.netapp.com/us/documentation/ontap-select.aspx>

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Copyright © 2019 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.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, 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).

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.