Technical Report

Best Practices Guide for Microsoft SQL Server and SnapManager 7.2 for SQL Server with Clustered Data ONTAP

Cheryl George, NetApp

January 2015 | TR-4369

Abstract

This best practices guide is intended for storage administrators and database administrators to help them successfully deploy Microsoft® SQL Server® 2014, 2012, 2008, and 2005 on NetApp® storage using NetApp SnapManager® 7.2 technology for Microsoft SQL Server.

TABLE OF CONTENTS

1 Executive Summary 4
1.1 Purpose and Scope 4
1.2 Audience 4
1.3 Caveats 5
2 Introduction to SQL Server 2014 5
2.1 In-Memory OLTP Capabilities (Hekaton) 5
2.2 SQL Server Data Files in Windows Azure 5
2.3 AlwaysOn Enhancements 6
2.4 Buffer Pool Extensions 6
2.5 Security Enhancements 6
3 Database and Storage 7
3.1 Aggregates 7
3.2 Volumes 7
3.3 LUNs 8
3.4 SMB Shares 10
4 Storage Efficiency and Manageability 12
4.1 Snapshot Copies 12
4.2 Thin Provisioning 13
4.3 Space Guarantee 13
4.4 Space Reclamation 14
4.5 Fractional Reserve 14
4.6 NetApp FlexClone 15
4.7 NetApp Deduplication 16
4.8 NetApp SnapMirror 16
5 NetApp Backup and Restore Solution for Microsoft SQL Server 17
5.1 SnapManager 7.2 for SQL Server 17
5.2 Backup 22
5.3 Restore 27
5.4 Reseeding Availability Groups 28
6 Virtualization 28
6.1 Hyper-V 28
6.2 VMware 28
7 MetroCluster with Data ONTAP 8.3 30

8 Best Practices for Ongoing Management and Monitoring 30

References 31

LIST OF TABLES

Table 1) Database layout on SMB shares. 10

Table 2) Volume guarantee set to `none`. 14

Table 3) Using autogrow and autodelete. 15

LIST OF FIGURES

Figure 1) Basic SQL Server database design for NetApp storage system. 11

Figure 2) SQL Server database design for NetApp storage systems: tempdb on dedicated volume. 12

Figure 3) SQL Server database design for NetApp storage systems: transaction logs on dedicated volume. 12

Figure 4) Clone of clone operation. 18

Figure 5) SMSQL clone wizard will not display level-2 clones. 18

Figure 6) Split clone database operation. 20

Figure 7) Delete clone database operation. 21

Figure 8) SMSQL clone wizard displaying only level-2 clones for deletion. 22

Figure 9) Backup wizard allowing backup of cloned databases with clone level displayed. 23

Figure 10) Backup management groups settings in the SMSQL GUI. 25

Figure 11) Backup grouping setting in the SMSQL GUI. 26

1 Executive Summary

Many organizations use Microsoft SQL Server as a back-end datastore for mission-critical business applications. The latest release, SQL Server 2014, delivers manageability, increased availability, and performance. Availability is improved through the AlwaysOn Availability Groups feature, which makes SQL Server capable of hosting up to eight secondary replicas; performance is improved through the new SQL Server in-memory capabilities, which are built into the database for online transaction processing (OLTP), and through features for data warehousing workloads and buffer pool extension (BPE). BPE extends the database engine buffer pool cache, usually through the use of solid-state drives (SSDs).

SQL Server implementations have become more complex and require more reliability than ever before. Database service-level agreements (SLAs) require predictable performance, and outages are disruptive and costly for database consumers. The underlying physical storage architecture that supports SQL Server is expected to scale to meet growing capacity needs, but bottlenecks arise frequently and backup processes get slower as databases grow. SQL Server databases are growing significantly large to meet the needs of today's organizations. At the same time, business requirements dictate shorter backup and restore windows.

Using the built-in SQL Server streaming backup and recovery solution causes restore times to require the backup time plus additional time to replay any transaction logs. The larger the database, the harder it is to meet organizational SLAs. Failure to meet the organization's SLAs can have a direct and devastating effect on revenue and customer goodwill. It is clear that conventional backup mechanisms simply do not scale, and many organizations have sought new, innovative solutions to comply with increasingly stringent business requirements.

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 clustered Data 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 topics:

- The NetApp clustered Data ONTAP operating system
- NetApp SnapDrive® for Windows® data management software
- NetApp SnapManager® for SQL Server (SMSQL) technology
- Microsoft SQL Server architecture and administration
- Microsoft SQL Server

For configuration compatibility across the NetApp stack, refer to the NetApp Interoperability Matrix Tool (IMT).

1.2 Audience

This report is intended for experienced SQL Server administrators, IT managers, and storage administrators who reviewed the following product documentation:

- NetApp SnapDrive for Windows
- SnapManager for Microsoft SQL Server
- Clustered Data ONTAP

Readers should ideally have a good understanding of the SQL Server storage architecture and administration as well as of SQL Server backup and restore concepts. For more information about the SQL Server architecture, refer to Books Online for SQL Server 2014.

1.3 Caveats

The following caveats apply to the information in this technical report:

- Applications that use SQL Server as a back end might have specific requirements dictated by the applications' design characteristics and are beyond the scope of this technical report. For application-specific guidelines and recommendations relating to physical and logical database layout, contact your application provider.
- Best practices for managing SQL Server environments focus exclusively on the latest NetApp storage operating system, the Data ONTAP architecture.

2 Introduction to SQL Server 2014

2.1 In-Memory OLTP Capabilities (Hekaton)

New in SQL Server 2014, the In-Memory OLTP engine can significantly improve OLTP database application performance. In-Memory OLTP is a memory-optimized database engine integrated into the SQL Server engine, optimized for OLTP.

The In-Memory OLTP engine is designed for extremely high session concurrency of OLTP transactions that are driven from a highly scaled-out middle tier. To achieve this performance, the engine uses latch-free data structures and optimistic, multiversion concurrency control. The result is predictable, submillisecond low latency and high throughput with linear scaling for database transactions. The actual performance gain depends on many factors, but 5-to-20 times performance improvements are common.

In-Memory OLTP runs only in the 64-bit version of SQL Server 2014 Enterprise Edition. The committed checkpoint file for durable tables must be configured to run on the fastest disks in the storage controller.

2.2 SQL Server Data Files in Windows Azure

SQL Server Data Files in Windows Azure™ provides native support for SQL Server database files stored as Windows Azure blobs. It allows you to create a database in SQL Server that runs on the premises or in a virtual machine (VM) in Windows Azure with a dedicated storage location for your data in Windows Azure blob storage.

SQL Server Data Files in Windows Azure simplifies the task of moving databases between machines through detach and attach operations. In addition, it provides an alternative storage location for your database backup files by allowing you to restore from or to Windows Azure storage. The feature enables several hybrid solutions and provides benefits for data virtualization, data movement, data security, and data availability, in addition to low cost and easy maintenance for high availability and elastic scaling.

Hosting a SQL Server Database in a Windows Azure Virtual Machine

SQL Server 2014 includes a new wizard for deploying SQL Server databases to a Windows Azure VM. The wizard performs a full database backup operation—it always copies the complete database schema and the data from a SQL Server user database. The wizard also automatically configures the Windows Azure VM, so no manual preconfiguration of the VM is required.

2.3 AlwaysOn Enhancements

SQL Server 2014 adds the following enhancements to the AlwaysOn Availability Groups feature:

- An Add Azure Replica wizard simplifies the creation of hybrid solutions for AlwaysOn availability groups.
- The maximum number of secondary replicas is increased from four to eight.
- When disconnected from the primary replica or during cluster quorum loss, readable secondary replicas remain available for read workloads.
- Failover cluster instances (FCIs) can use Cluster Shared Volumes (CSVs) as cluster-shared disks.

Note: SnapManager for SQL Server does not support CSV disks.

2.4 Buffer Pool Extensions

The buffer pool extension provides seamless integration of a nonvolatile RAM 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

Best Practices
• Make sure that an SSD-backed LUN 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 or larger than the buffer pool.

2.5 Security Enhancements

SQL Server 2014 has three new security levels:

- `CONNECT ANY DATABASE` is a new server-level permission. Grant `CONNECT ANY DATABASE` to a login that must connect to all existing databases and to any new databases created in the future. `CONNECT ANY DATABASE` does not grant any permission in any database beyond permission to connect. Combine this permission with `SELECT ALL USER SECURABLES` or `VIEW SERVER STATE` to allow an auditing process to view all data or all database states on the SQL Server instance.
- `IMPERSONATE ANY LOGIN` is a new server-level permission. When granted, it allows a middle-tier process to impersonate the account of clients connecting to it as it connects to databases. When this permission is denied, a high privileged login can be blocked from impersonating other logins. For example, a login with `CONTROL SERVER` permission can be blocked from impersonating other logins.
- `SELECT ALL USER SECURABLES` is a new server-level permission. When this permission is granted, a login such as an auditor can view data in all databases to which the user can connect.

3 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.

3.1 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. Starting with Data ONTAP 8.0, aggregates are either 32-bit or 64-bit format. With large-sized SATA disks, increased spindle count caused by increased disk count can help maximize performance and maintain high storage efficiency due to the large-size thresholds.

NetApp has performed various tests by using shared and dedicated aggregates with data files and transaction log files separated and I/O workload characterization. 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 AlwaysOn 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 using NetApp SnapMirror® technology.

Best Practice
NetApp recommends having at least 10% free space available in an aggregate for optimal storage performance.

3.2 Volumes

NetApp FlexVol® volumes are created and reside inside aggregates. Many volumes can be created in a single aggregate, and each volume can be expanded, shrunk, or moved between aggregates.

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.

Best Practices

- Use FlexVol 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 SnapManager for SQL Server to coordinate Snapshot copies of the SQL Server data volumes.
- 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.
- Reporting and decision support system (DSS)–type workloads use the tempdb extensively as a temporary workspace, especially for I/O-intensive `DBCC CHECKDB` operations against large databases, which causes spooling into the tempdb. Therefore, place this database on a dedicated volume with a separate set of spindles on which Snapshot copies are not created, because the tempdb is recreated every time SQL Server is restarted, thereby saving storage space. 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.
- Place user data files (.mdf) on separate volumes or VMDK 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. This significantly improves SQL Server performance because it allows multiple physical devices to concurrently service requests.
- Place the database and backups on separate volumes to enable recovery of databases in case of device failure. Doing so also allows isolation of read-write I/O against the production database from write I/O during backup operations.
- Create the SnapManager share on the dedicated FlexVol volume in which SnapManager copies transaction logs for the availability group database. These logs are required to perform an up-to-the-minute restore of the availability group databases on another node.

For complete details on volume design considerations for SQL Server environments, refer to Books Online for SQL Server 2014 and SnapManager 7.2 for SQL Server Installation and Setup Guide.

3.3 LUNs

SnapInfo Directory

The SnapInfo directory is the main repository of the SnapManager for SQL Server (SMSQL) software. All the 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) is kept in the SnapInfo directory. The size of the SnapInfo LUN is calculated as follows:

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

The SnapInfo LUN sizing formula assumes the following:

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

The amount of data in the SnapInfo 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.

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.

Note: Direct access is read-only.

Best Practices

- Place the SnapInfo directory on a LUN in a dedicated volume that is not shared by any other type of data that can potentially corrupt the backup Snapshot copies taken by SMSQL.
- Do not place user databases or system databases on a LUN that hosts mount points.
- Always use the SMSQL configuration wizard to migrate databases to NetApp storage so that the databases are stored in valid locations with SnapInfo configured appropriately. This enables successful SMSQL backup and restore operations. Keep in mind that the migration process is disruptive and can cause the databases to go offline during the migration process.

The following conditions must be in place to enable failover cluster instances (FCIs) of SQL Server:

- The SnapInfo LUN must be a cluster disk resource in the same cluster group as the SQL Server instance being backed up by SnapManager.
- 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 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 file group) 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 file groups share the same LUNs, the database layout must be identical for all databases. However, this restriction is not applicable when the Unrestricted Database Layout option is enabled.

3.4 SMB Shares

SnapManager supports SQL 2012/2014 databases on SMB shares residing on clustered Data ONTAP 8.2 and later. You can set up the layout as shown in Table 1.

Table 1) Database layout on SMB shares.

Content	SMB share
UserDB (multiple instances)	\\<CIFS servername>\UserDB
UserLOG (multiple instances)	\\<CIFS servername>\UserLog
SQL Server binaries + SystemDBs	\\<CIFS servername>\SystemDB
SnapInfo	\\<CIFS servername>\SnapInfo

Note: SQL Server 2014 does not support in-memory (Hekaton) databases on x86 platforms and on NAS configuration.

For additional information, refer to Hardware and Software Requirements for Installing SQL Server 2014 and Install SQL Server with SMB Fileshare as a Storage Option.

Best Practices
• Configure the Transport Protocol Setting dialog box in SnapDrive for 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 SnapDrive for 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.

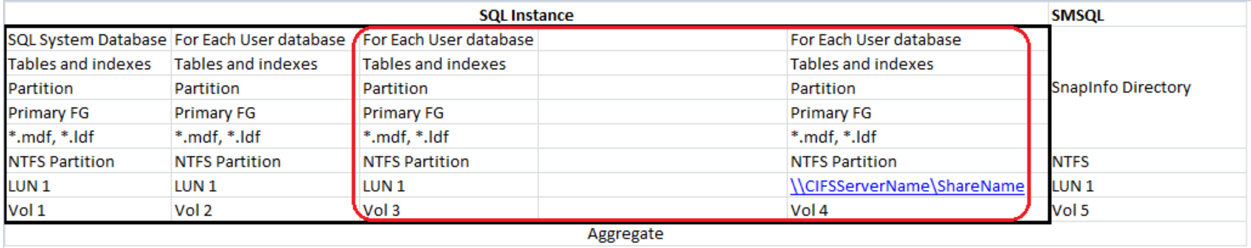
Database Storage Design

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

Design Example 1: Basic Design

Figure 1 shows information about the basic SQL Server database design for NetApp storage systems.

Figure 1) Basic SQL Server database design for NetApp storage system.



SQL Instance					SMSQL
SQL System Database	For Each User database	For Each User database		For Each User database	
Tables and indexes	Tables and indexes	Tables and indexes		Tables and indexes	
Partition	Partition	Partition		Partition	SnapInfo Directory
Primary FG	Primary FG	Primary FG		Primary FG	
*.mdf, *.ldf	*.mdf, *.ldf	*.mdf, *.ldf		*.mdf, *.ldf	
NTFS Partition	NTFS Partition	NTFS Partition		NTFS Partition	NTFS
LUN 1	LUN 1	LUN 1		\\CIFSServerName\ShareName	LUN 1
Vol 1	Vol 2	Vol 3		Vol 4	Vol 5
Aggregate					

The basic database storage design has the following characteristics:

- It does not use SQL Server partitions beyond the default configuration.
- It contains one aggregate for the SQL Server instance.
- It uses a dedicated volume and LUN for the SQL Server system databases, including the tempdb database.
- It uses a dedicated volume and LUN for each user database.
- It uses a dedicated volume for databases that reside on an SMB share.
- It uses a dedicated volume and LUN for the SMSQL SnapInfo directory.

This configuration can be scaled for multiple user-defined databases per instance by replicating the information in the area circled in red in Figure 1. It can also be scaled for multiple SQL Server instances on the same server by replicating the information inside the black box in Figure 1.

The same aggregate can be used as the user databases and the SQL Server instances are scaled out. In fact, the same aggregate can be used for all SQL Server hosts that are connected to the storage system.

For SnapManager for SQL Server, you can use one SnapInfo directory for all databases, which is the configuration implied in this design, or one SnapInfo directory per database. For information that can help you determine what is appropriate for your specific design, refer to the SnapManager 7.2 for Microsoft SQL Server Installation and Setup Guide.

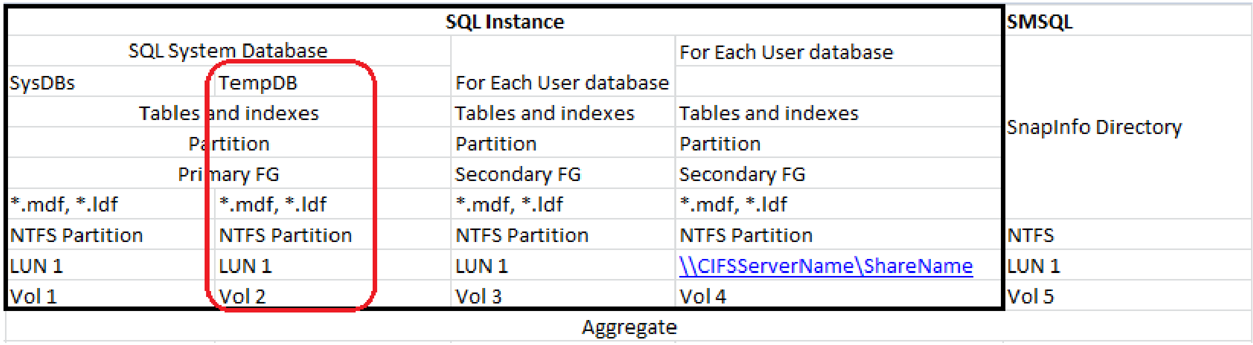
Design Example 2: tempdb on Dedicated Volume

Figure 2 shows information about a SQL Server database design for NetApp storage systems in which the tempdb database is placed on a dedicated volume. The design in Figure 2 is identical to the design in Figure 1, except that tempdb has been placed on its own volume.

Isolating the tempdb database on a separate volume makes it possible to keep it out of Snapshot copies. This design also provides more granular control over the disks in which tempdb resides and over the number of LUNs of which tempdb is composed. Spreading tempdb across multiple LUNs can help improve its ability to handle higher I/O requirements.

After the storage is configured, you can move tempdb to its own LUNs by using the SnapManager for SQL Server (SMSQL) configuration wizard or Microsoft SQL Server Management Studio.

Figure 2) SQL Server database design for NetApp storage systems: tempdb on dedicated volume.



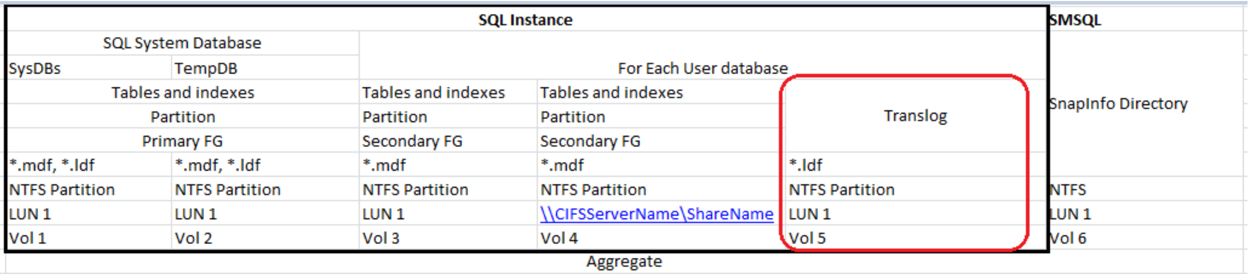
SQL Instance				SMSQL
SQL System Database			For Each User database	SnapInfo Directory
SysDBs	TempDB	For Each User database		
Tables and indexes		Tables and indexes	Tables and indexes	
Partition		Partition	Partition	
Primary FG		Secondary FG	Secondary FG	
*.mdf, *.ldf	*.mdf, *.ldf	*.mdf, *.ldf	*.mdf, *.ldf	
NTFS Partition	NTFS Partition	NTFS Partition	NTFS Partition	NTFS
LUN 1	LUN 1	LUN 1	\\CIFSServerName\ShareName	LUN 1
Vol 1	Vol 2	Vol 3	Vol 4	Vol 5
Aggregate				

Design Example 3: Separate Transaction Log

The design shown in Figure 3 builds on the previous two designs. In this design, the user database log is placed on its own volume. This design provides more granular control over the disks in which the log resides and over how many LUNs are included in it. Spreading the log across multiple LUNs can help improve its ability to handle higher I/O requirements.

This design also allows the log file to be managed independently of the data files. Snapshot copies are created at the volume level; using SnapManager for SQL Server, you can, for example, schedule database Snapshot backups every hour and create a schedule for log backups every 10 minutes.

Figure 3) SQL Server database design for NetApp storage systems: transaction logs on dedicated volume.



SQL Instance					SMSQL
SQL System Database					SnapInfo Directory
SysDBs	TempDB	For Each User database			
Tables and indexes		Tables and indexes	Tables and indexes	Translog	
Partition		Partition	Partition		
Primary FG		Secondary FG	Secondary FG		
*.mdf, *.ldf	*.mdf, *.ldf	*.mdf	*.mdf	*.ldf	
NTFS Partition	NTFS Partition	NTFS Partition	NTFS Partition	NTFS Partition	NTFS
LUN 1	LUN 1	LUN 1	\\CIFSServerName\ShareName	LUN 1	LUN 1
Vol 1	Vol 2	Vol 3	Vol 4	Vol 5	Vol 6
Aggregate					

For best practices applicable to the configuration of the NetApp storage subsystem, refer to TR-3838: Storage Subsystem Configuration Guide.

4 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 these technologies are used in conjunction, the larger the savings.

4.1 Snapshot Copies

NetApp Snapshot technology provides low-cost, fast-backup, point-in-time copies of the file system (volume) or LUN by preserving Data ONTAP architecture WAFL® (Write Anywhere File Layout) consistency points. NetApp SnapManager integrates with 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 a number of 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, an SMSQL 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 verification process.

4.2 Thin Provisioning

Thin provisioning is a method for optimizing the utilization 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 Data 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 actually 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 very 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.

4.3 Space Guarantee

Space guarantee enables thin provisioning. The space guarantee option can be set at the volume or LUN level. If 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 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 actually written to the volume. Write operations to space-reserved LUNs in a volume with the setting `guarantee=none` will 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 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.

4.4 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.

Note: When using Windows Server® 2012 with clustered Data ONTAP 8.2 and Host Utilities Kit 6.2, you do not need to schedule postspace reclamation because it happens inline.

Best Practice
Run space reclamation during periods of low activity because it initially consumes cycles on the host.

4.5 Fractional Reserve

Fractional reserve is a volume option that determines how much space Data ONTAP reserves to be used for Snapshot copy overwrite for LUNs after all other space in the volume has been used. NetApp recommends setting the fractional reserve to 0 in SQL Server environments.

Best Practice Configurations When Using Thin Provisioning in SQL Server Environments

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.

Option 1: Volume Guarantee Set to none

Table 2 shows the settings for thin-provisioned volumes with the volume guarantee set to `none`.

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

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.

Option 2: Using Autogrow and Autodelete

Table 3 shows the settings for thin-provisioned volumes that use autogrow and autodelete.

Table 3) Using autogrow and autodelete.

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

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 will 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.

Best Practice
NetApp recommends using autogrow for most common deployment configurations.

4.6 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.

For detailed information about how FlexClone works and about command-line references, refer to the Clustered Data ONTAP 8.2 Logical Storage Management Guide.

Best Practice
Make sure that FlexClone is licensed on the storage system for successful SMSQL cloning operations.

4.7 NetApp Deduplication

NetApp deduplication is a data compression technique for eliminating coarse-grained redundant data, typically to improve storage utilization. When deduplication runs for the first time on a FlexVol volume with existing data, it scans the blocks in the volume and creates a fingerprint database, which contains a sorted list of all fingerprints for used blocks in the volume. Each 4kB block in the storage system has a digital fingerprint, which is compared to other fingerprints in the volume. If two fingerprints are found to be the same, a byte-for-byte comparison is done of all bytes in the block. If they are an exact match, the duplicate block is discarded, and the space is reclaimed. The core enabling technology of deduplication is fingerprints.

Deduplication consumes system resources and can alter the data layout on the disk. Due to the application I/O pattern and the effect of deduplication on the data layout, the read/write I/O performance can vary.

Note: Deduplication is transparent to SQL Server, which does not recognize the block changes, so the SQL Server database remains unchanged in size from the host even though there are capacity savings at the volume level.

4.8 NetApp SnapMirror

NetApp SnapMirror technology offers a fast and flexible enterprise solution for mirroring or replicating data over local area networks (LANs) and wide area networks (WANs). SnapMirror technology transfers only modified 4kB data blocks to the destination after the initial base transfer, thereby significantly reducing network bandwidth requirements. SnapMirror in clustered Data ONTAP provides asynchronous volume-level replication that is based on a configured replication update interval.

Best Practices

- 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, make sure that SnapMirror operations are not scheduled to occur on the source volume at the same time as SMSQL backup Snapshot copies are being created.
- 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, refer to the following resources:

- TR-4015: SnapMirror Configuration and Best Practices Guide for Clustered Data ONTAP
- SnapMirror Sizing Guide for Clustered Data ONTAP 8.2

- SnapMirror Schedule Advisor (helps model transfer times and provides guidance for SnapMirror schedules)

5 NetApp Backup and Restore Solution for Microsoft SQL Server

When planning the backup and restore process for recovery purposes, the following objectives must be clearly defined according to customer SLAs:

- **Recovery point objective (RPO).** To what point in time must the data be recovered?
- **Recovery time objective (RTO).** How long will it take to get the database back online and rolled forward or backward to the RPO?

5.1 SnapManager 7.2 for SQL Server

SnapManager for SQL Server (SMSQL) provides an integrated data management solution for Microsoft SQL Server that dramatically boosts the availability and reliability of SQL Server databases. SnapManager provides rapid online backup and near-instantaneous restore of databases by using the online Snapshot technology that is part of Data ONTAP software. SnapManager can also leverage the SnapMirror capabilities of storage systems to provide on-site or off-site SnapManager backup set mirroring for disaster recovery. In addition, SMSQL allows database cloning, which is the process of creating a point-in-time copy of a production database or its backup set. Some of the new features available with SnapManager 7.2 for SQL Server, using SnapDrive 7.1 for Windows, include:

- Clone databases from existing backup set
- Clone active production databases
- Split cloned database
- Delete cloned database

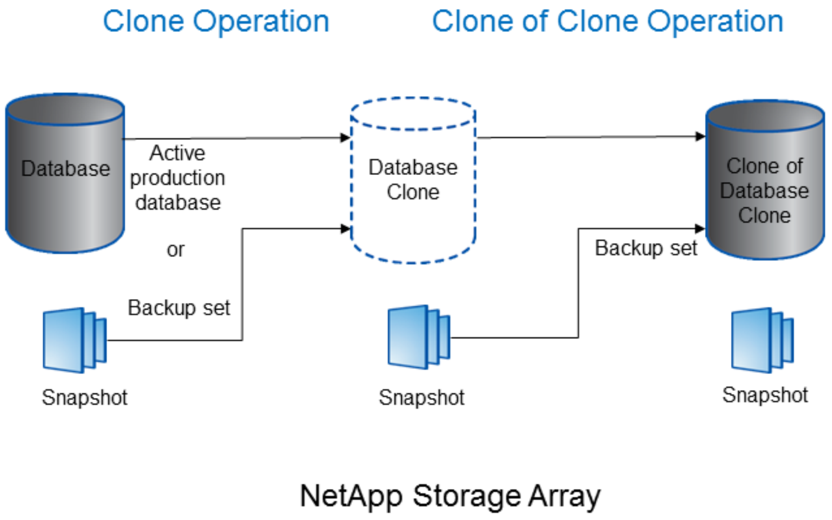
The sections that follow provide more details about these features.

Clone of Clone Database

This feature provides the ability to clone an existing clone of the production database. You can run tests such as updating or upgrading service packs on this clone of clone database to avoid risking corruption of the clone database.

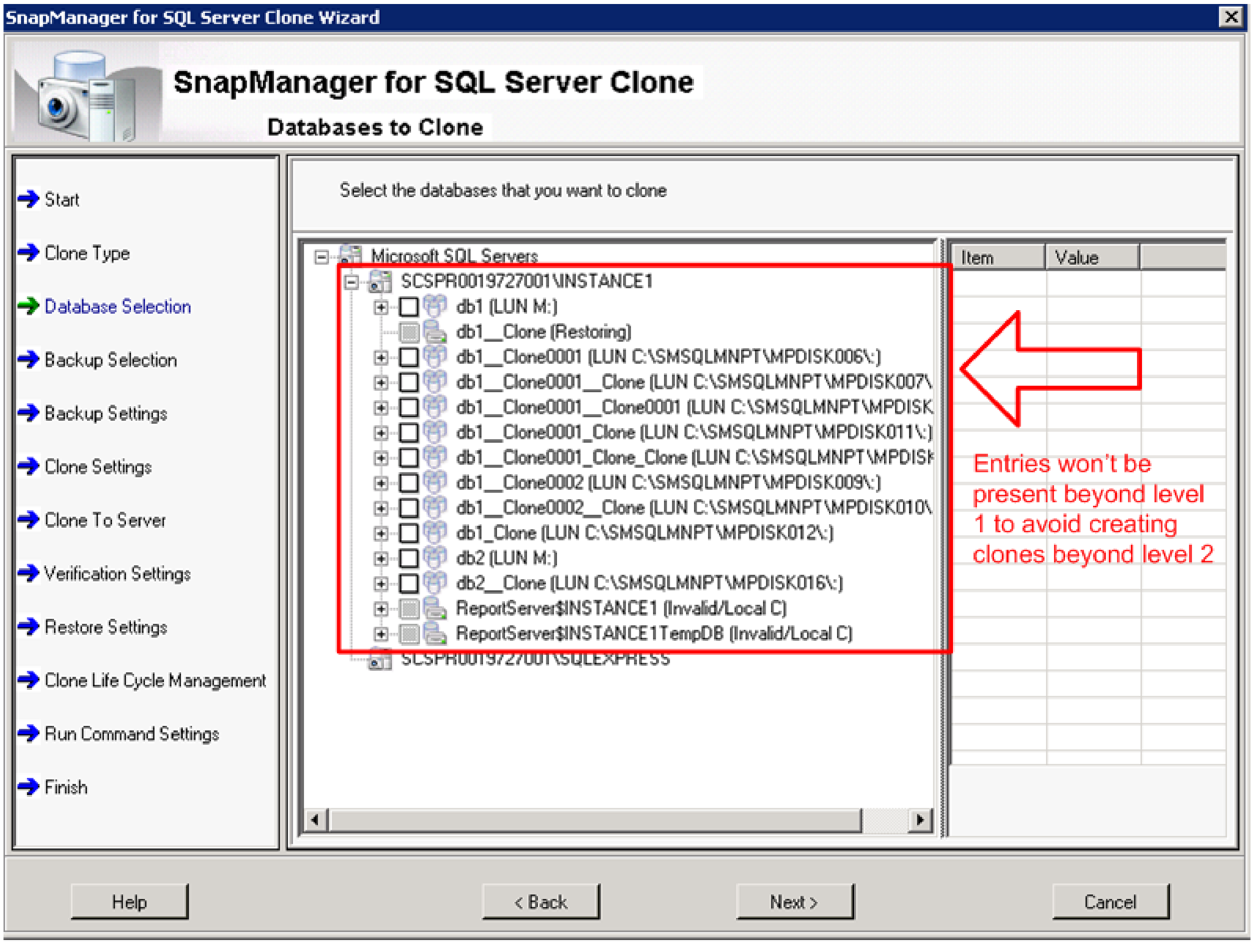
Creation of the clone of clone database operation can be performed against active production databases or from existing backup Snapshot copies of cloned databases. The SMSQL backup wizard can then be used to back up this clone of clone databases.

Figure 4) Clone of clone operation.



However, this clone operation is restricted to two levels, which means that a level-2 clone database cannot be further cloned. Therefore, the SMSQL clone wizard will not display these level-2 clone databases, as shown in Figure 5.

Figure 5) SMSQL clone wizard will not display level-2 clones.

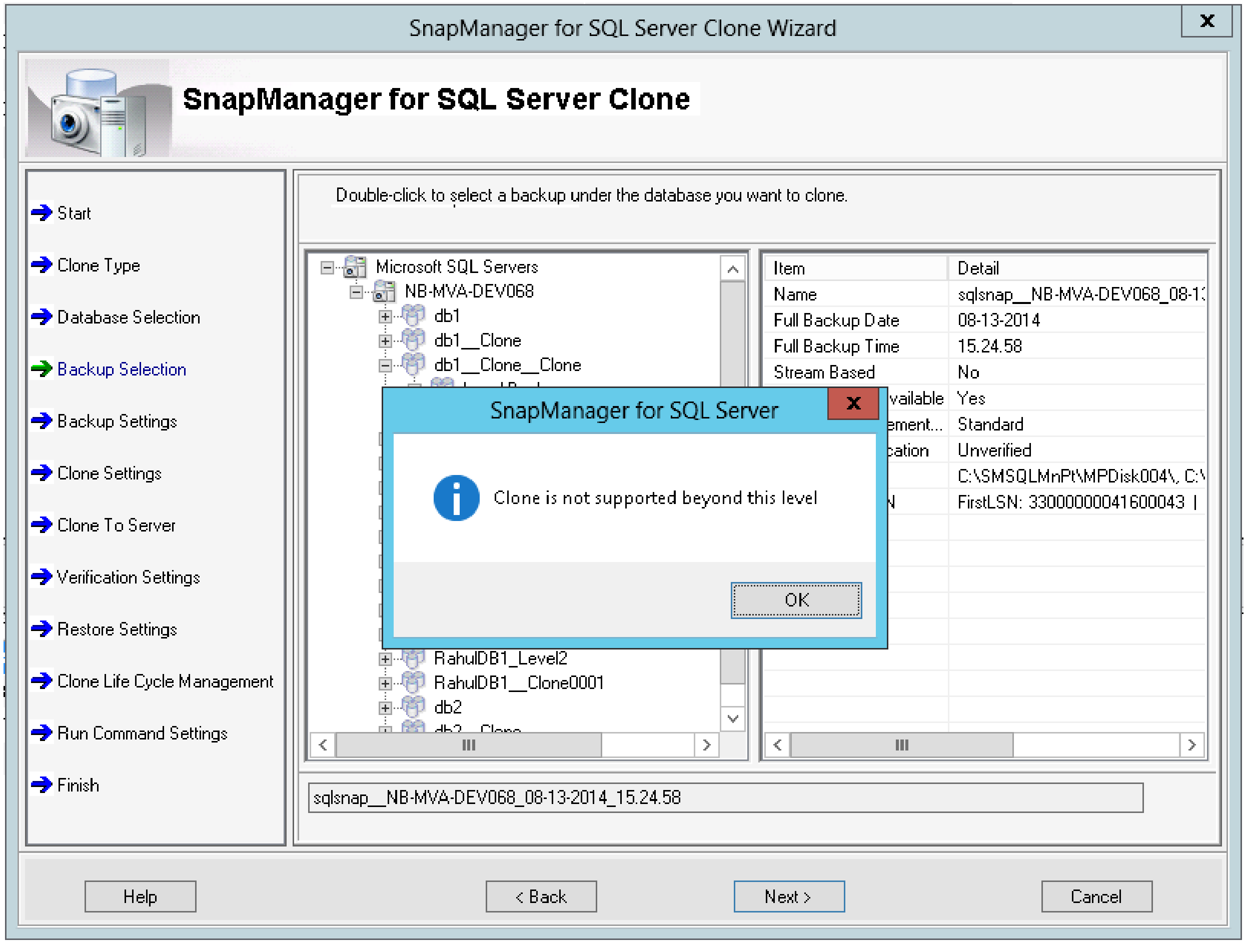


Select the databases that you want to clone

```
Microsoft SQL Servers
  SCSPR0019727001\INSTANCE1
    □ db1 (LUN M:)
    db1__Clone (Restoring)
    □ db1__Clone0001 (LUN C:\SMSQLMNPT\MPDISK006\:)
    □ db1__Clone0001__Clone (LUN C:\SMSQLMNPT\MPDISK007\
    □ db1__Clone0001__Clone0001 (LUN C:\SMSQLMNPT\MPDISK
    □ db1__Clone0001_Clone (LUN C:\SMSQLMNPT\MPDISK011\:)
    □ db1__Clone0001_Clone_Clone (LUN C:\SMSQLMNPT\MPDISK
    □ db1__Clone0002 (LUN C:\SMSQLMNPT\MPDISK009\:)
    □ db1__Clone0002__Clone (LUN C:\SMSQLMNPT\MPDISK010\
    □ db1_Clone (LUN C:\SMSQLMNPT\MPDISK012\:)
    □ db2 (LUN M:)
    □ db2__Clone (LUN C:\SMSQLMNPT\MPDISK016\:)
    ReportServer$INSTANCE1 (Invalid/Local C)
    ReportServer$INSTANCE1TempDB (Invalid/Local C)
  SCSPR0019727001\SQLEXPRESS
```

Entries won't be present beyond level 1 to avoid creating clones beyond level 2

When a backup Snapshot copy containing a cloned database at level 2 is selected in the SMSQL clone wizard, the message below appears. This message also implies that verification cannot be performed against a level-2 database clone.



Double-click to select a backup under the database you want to clone.

Item	Detail
Name	sqlsnap__NB-MVA-DEV068_08-1:
Full Backup Date	08-13-2014
Full Backup Time	15.24.58
Stream Based	No
vailable	Yes
ement...	Standard
cation	Unverified
	C:\SMSQLMnPt\MPDisk004\, C:\
N	FirstLSN: 33000000041600043 |

sqlsnap__NB-MVA-DEV068_08-13-2014_15.24.58

SnapManager for SQL Server

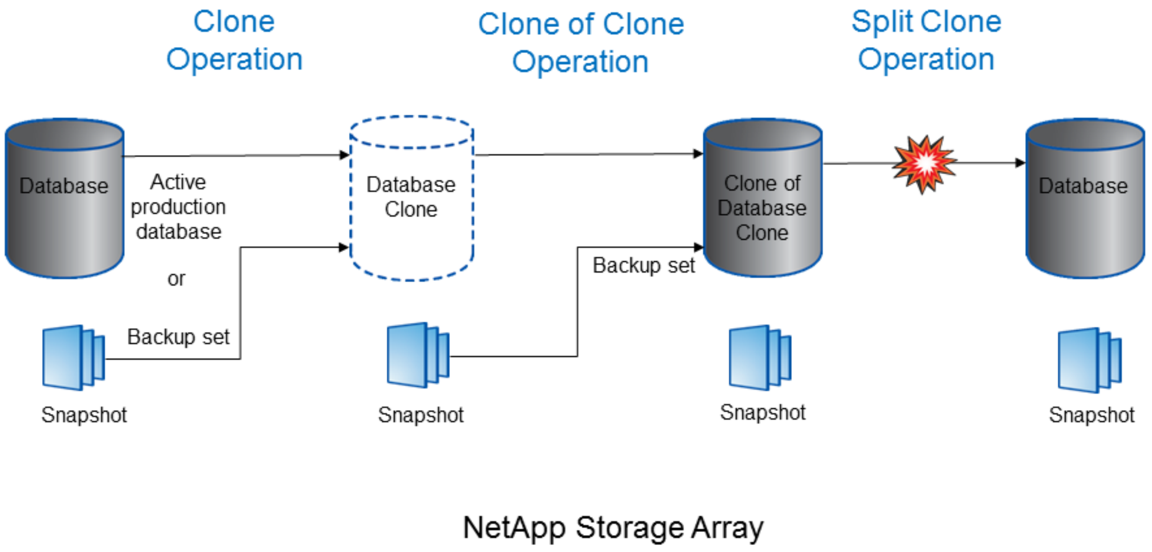
Clone is not supported beyond this level

OK

Split a Cloned Database

After the functional testing of a recently upgraded cloned database in a dev/test environment is complete, the split clone operation allows splitting this cloned database from its parent for further use as an independent database for production application usage. This means, however, that Data ONTAP requires additional space for the new volume to accommodate this new database from the successful split operation—the same amount of storage space as that of the database source volume. This action, however, will not carry forward the Snapshot copies from the source volume to the destination because the newly split volume is an independent entity.

Figure 6) Split clone database operation.



The SMSQL clone wizard will first perform the split operation on level-2 clones; the parent clones will be dimmed. After the level-2 clone is successfully split, the parent clone database is available for the split operation.

SMSQL provides a new PowerShell® cmdlet to perform the split operation "`split-clone`." This cmdlet enables the splitting of the level-2 cloned volume from its parent volume.

Syntax

```
split-clone [-Server <String>] [-Database <string[]>] [-GetStatus]
```

Where:

- `-Server <String>`

  Short form: `-svr`

  This parameter specifies the SQL Server to which the database is attached as a clone database. SMSQL adopts the local computer name as the default server instance.

- `-Database <String[]>`

  Short form: `–d`

  Use this option to specify the databases that need to be cloned. Use a comma-separated list of strings; for example:-d Database 1, Database 2, Database 3, Database 4, and so on.

- `GetStatus <switch parameter>`

  Short form: `–gtst`

  This parameter provides the status of each volume being split. Each mount point that has unique volumes will be displayed in the status. Because the split operation occurs at the volume level, other mount points or disks from the same volume are ignored.

Note: There is no equivalent option to obtain the status in the GUI.

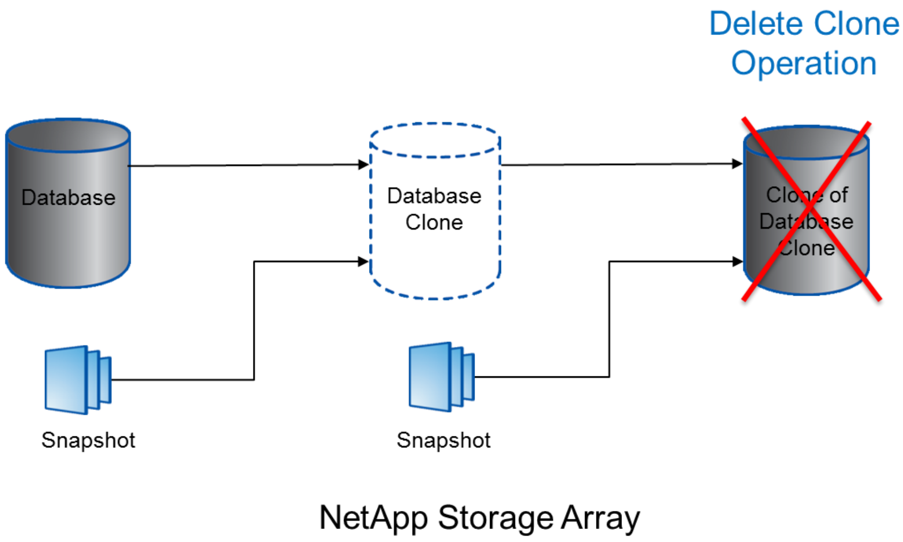
For example:

```
split-clone -Server SCSPR0019727001\INSTANCE1 -Database db1__Clone_Clone -GetStatus
```

Delete Cloned Database

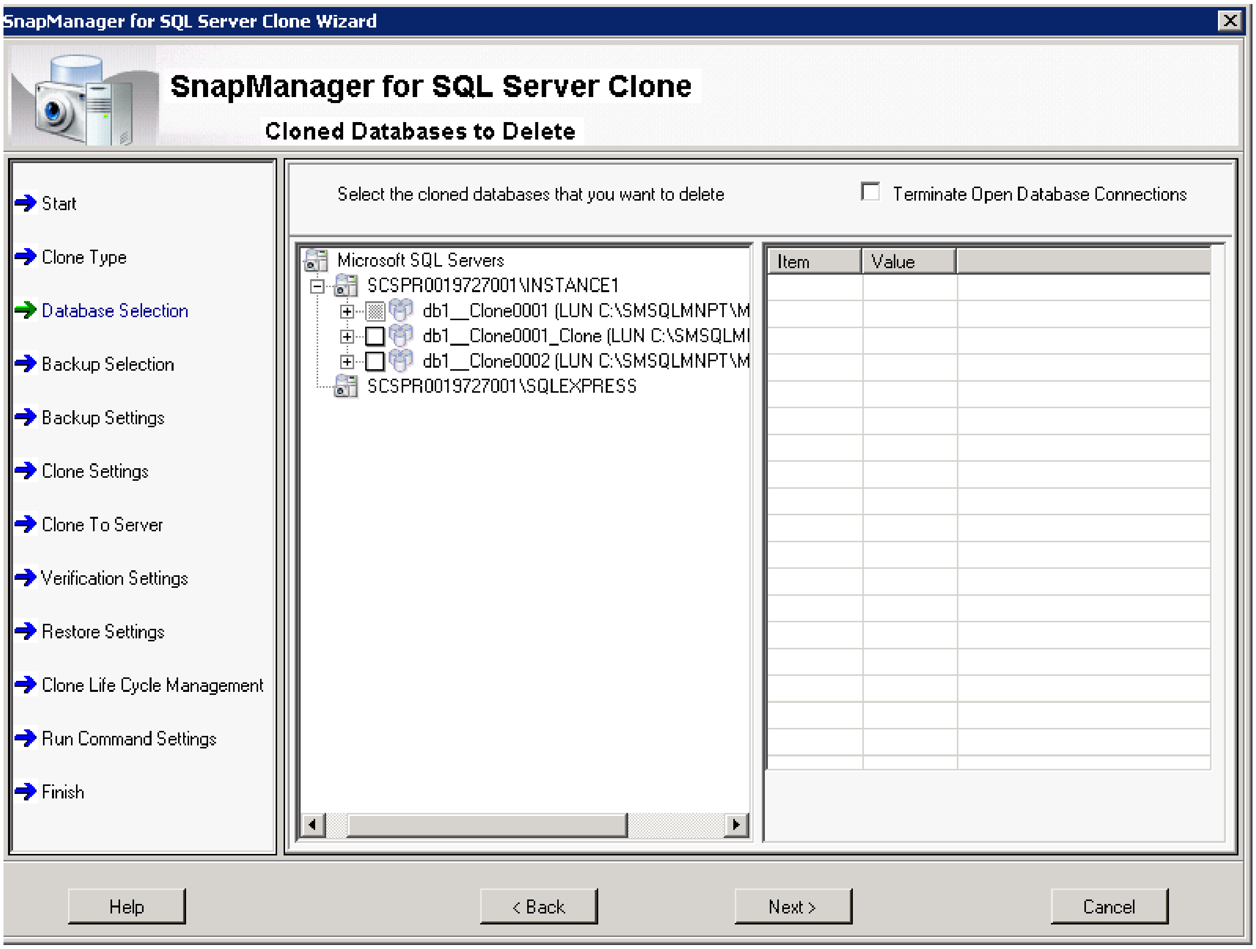
After the tests are completed, you can delete the cloned databases by using the SMSQL clone wizard.

Figure 7) Delete clone database operation.



The SMSQL clone wizard for the delete operation will display the level-2 clones with its parent clone dimmed if the cloned databases reside on SAN, NAS, and VMDK over NFS volumes. The parent clones are available for the delete operation only after the level-2 clones are deleted. However, when databases reside on VMFS over VMDK, the clone operation allows the deletion of the parent or level-2 clones because a LUN clone is used.

Figure 8) SMSQL clone wizard displaying only level-2 clones for deletion.



SnapManager for SQL Server Clone Wizard

SnapManager for SQL Server Clone

Cloned Databases to Delete

- Start
- Clone Type
- Database Selection
- Backup Selection
- Backup Settings
- Clone Settings
- Clone To Server
- Verification Settings
- Restore Settings
- Clone Life Cycle Management
- Run Command Settings
- Finish

Select the cloned databases that you want to delete

□ Terminate Open Database Connections

```
Microsoft SQL Servers
  SCSPR0019727001\INSTANCE1
    ▨ db1__Clone0001 (LUN C:\SMSQLMNPT\M
    □ db1__Clone0001_Clone (LUN C:\SMSQLMI
    □ db1__Clone0002 (LUN C:\SMSQLMNPT\M
  SCSPR0019727001\SQLEXPRESS
```

Item	Value

Best Practices

- The aggregate used by the SQL Server database should be included in the SVM's list of assigned aggregates for the clone operation to be successful.
- Verify that the remote server has connectivity to the SMB storage performing remote cloning.
- Run the split clone operation during off-peak hours; the split operation may take time based on the size and load on the volume.
- Make sure that the aggregate has sufficient space because the split clone operation creates a new volume that is of the same size as the source volume. Use the `–GetStatus` switch with the Split-Clone cmdlet to gather details for space availability within the aggregate for a successful split clone operation.
- Split clone operations operate at the volume level. Therefore, you should verify the selection of the clone database on which the split operation will be performed because this operation will split the cloned volume, irrespective of the database selection made by the end user.

5.2 Backup

SnapManager leverages NetApp Snapshot technology to protect SQL Server databases through the Virtual Device Interface (VDI). Snapshot technology greatly reduces the time it takes to back up large SQL Server databases because it tracks only changed blocks of data. In this respect, Snapshot backups contrast with traditional backups created through SQL Server `BACKUP` statements.

Use the SMSQL configuration wizard to migrate the SQL Server databases to a NetApp LUN or SMB share so that they can be backed up by SnapManager for SQL Server. SMSQL is now also capable of backing up the cloned and clone of clone databases. SnapDrive for Windows provides the underlying layer of support by making storage available to a Windows Server host. SnapDrive integrates with Windows Volume Manager so that storage systems can serve as virtual storage devices for application data in a Windows Server environment or provision storage for Windows virtual machines that are hosted on ESXi™ hypervisors.

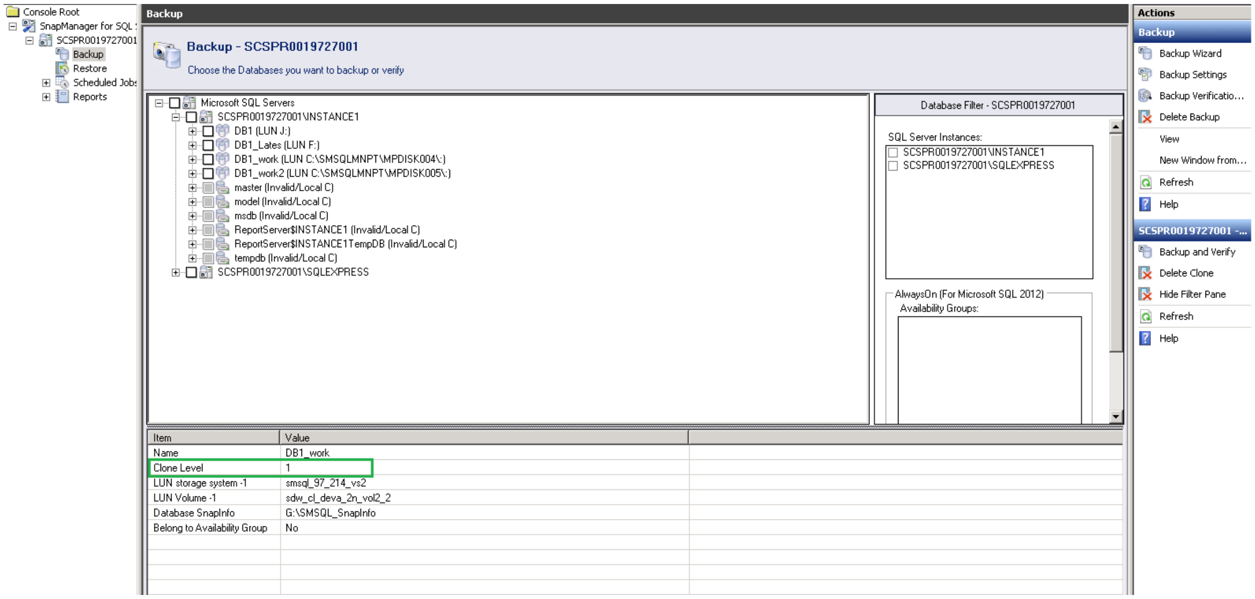
During a backup, SnapManager for SQL Server performs the following actions:

- Creates Snapshot copies of all volumes that are used by the SQL Server databases being backed up
- Backs up user databases placed in LUNs or SMB shares that also contain system databases by using a stream-based backup method to the SnapInfo directory

Note: An SMSQL transaction log backup is functionally the same as a transaction log backup created by using `BACKUP` log statements and is streamed to the SnapInfo directory to provide point-in-time restores. The length of time required to back up a transaction log depends mainly on the active section of the transaction log that has to be extracted and copied to the dump directory. SQL Server 2014 does not support the backup of the in-memory (Hekaton) database, which is read-only.

Error! Reference source not found. shows a screenshot of the backup wizard displaying the cloned atabases and the level of clone.

Figure 9) Backup wizard allowing backup of cloned databases with clone level displayed.



Backup - SCSPR0019727001

Choose the Databases you want to backup or verify

Item	Value
Name	DB1_work
Clone Level	1
LUN storage system -1	smsql_97_214_vs2
LUN Volume -1	sdw_cl_deva_2n_vol2_2
Database SnapInfo	G:\SMSQL_SnapInfo
Belong to Availability Group	No

The RPO and RTO for backups vary from one database to another. A typical backup configuration with an RPO of 15 minutes would look like the example that follows.

An SMSQL full backup can be created every 4 hours and transaction log backups can be scheduled every 15 minutes with the following retention schedule:

- Hourly backup with a retention of 7 days
- Daily Snapshot copies with a retention of 14 days
- Weekly Snapshot copies with a retention of 364 days

For detailed information about how to back up and restore SQL Server databases, refer to the "SnapManager 7.2 for SQL Server Installation and Setup Guide."

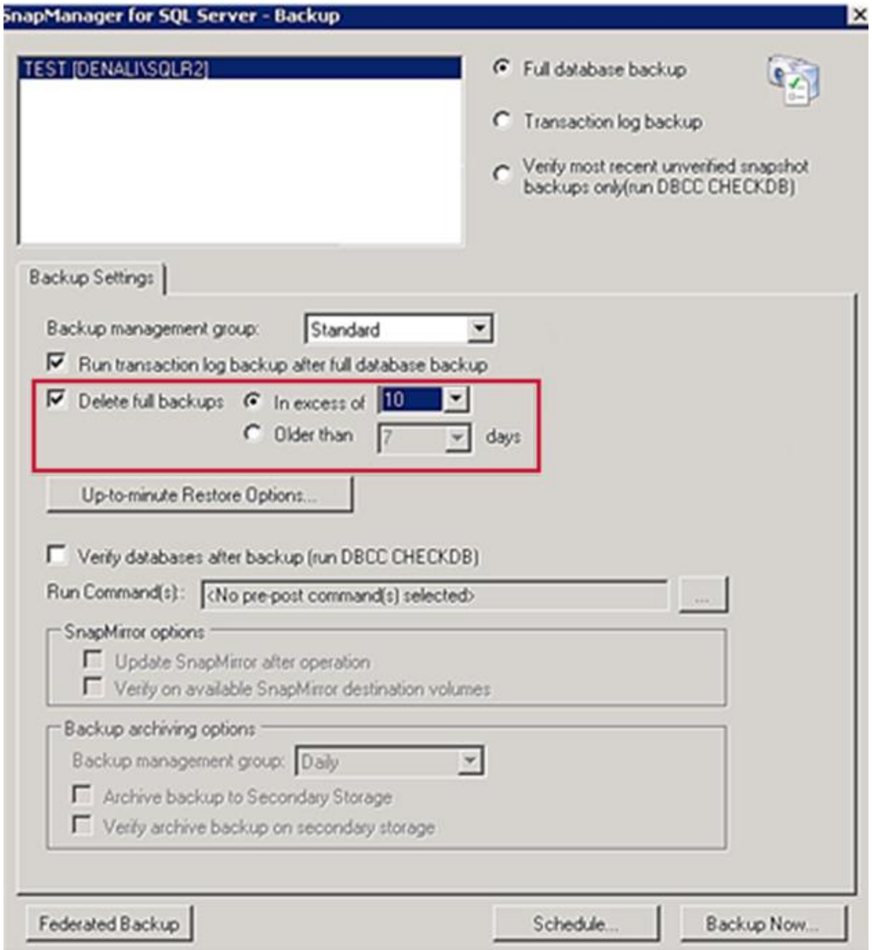
Best Practices

- Synchronize the system clock on the host running SnapManager with the clock on the storage system so that SnapDrive functions correctly.
- Do not include the tempdb database in a backup because the data it contains is temporary. Place tempdb on a LUN or an SMB share that is in a storage system volume in which Snapshot copies will not be created; otherwise, large amounts of valuable Snapshot space could be consumed.
- Always run SMSQL backups during off-peak hours to avoid performance impact on regular user activity.
- If you change the database configuration, especially when enabling the unrestricted database layout option, perform a backup immediately to allow up-to-the-minute restores.
- While doing a federated backup, make sure that both SAN and NAS databases from the same host are not included in a single federated group. However, SAN and NAS databases from different hosts are supported in a single federated group. SAN and NAS databases from the same host but in different federated groups are also supported.
- SnapManager cannot automatically restore transaction log backups created by using SQL Management Studio against an SMSQL full backup of a database. The transaction log files created by SQL Server Management Studio have a default extension (.tnr), which needs to be renamed to .trb and placed into the respective SnapInfo directory. After you place these transaction logs into the correct SnapInfo directory path, you can use SnapManager for SQL to restore them.
- When cloning FlexVol volumes, make sure that you have sufficient space for the clone metadata.
- To successfully back up an availability group, make sure that the replica databases are not in the `Synchronizing` or `Synchronized` states.
- Make sure that secondary replicas have the Readable Secondary option set to `Yes` when these replicas are part of an availability group backup.

Backup Retention Management

Backup management groups are an efficient way to manage the retention of transaction log backups and limit the number of old, saved transaction logs. When a new backup runs, it marks the previous backups as being point-in-time only and then deletes the logs as well as the Snapshot copies to reduce log traffic. Figure 10 shows the settings for backup management groups in the SMSQL GUI.

Figure 10) Backup management groups settings in the SMSQL GUI.



SnapManager for SQL Server - Backup

TEST (DENALI\SQLR2)

● Full database backup

○ Transaction log backup

○ Verify most recent unverified snapshot backups only(run DBCC CHECKDB)

Backup Settings

Backup management group: Standard

☑ Run transaction log backup after full database backup

☑ Delete full backups ● In excess of 10

○ Older than 7 days

☐ Verify databases after backup (run DBCC CHECKDB)

Run Command(s): <No pre-post command(s) selected>

SnapMirror options

☐ Update SnapMirror after operation

☐ Verify on available SnapMirror destination volumes

Backup archiving options

Backup management group: Daily

☐ Archive backup to Secondary Storage

☐ Verify archive backup on secondary storage

Backup Grouping Performance

The backup grouping feature groups databases that share storage into larger groups to be processed as a single backup set (or subjob) by the entire backup job. The most important backup rule to keep in mind is that Snapshot copies are always made at the volume level, and the volume can have many qtrees, LUNs, and, of course, file system objects.

Backup grouping offers the following advantages:

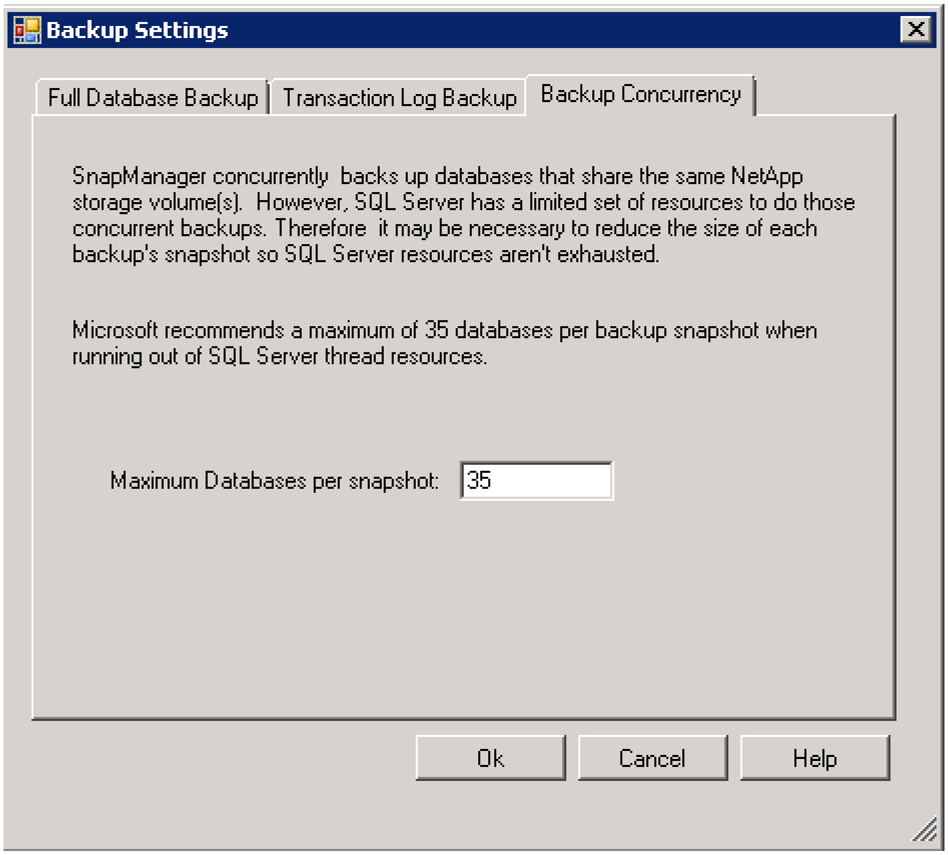
- Grouping, by default, reduces the per-group overhead (of VDI, storage, Windows, and so on) for each backup subjob.
- Grouping allows more Snapshot copies of volumes to be created concurrently, thus increasing parallel processing.

Default Limits

The maximum number of databases per backup set is 35.

The backup grouping feature groups databases into a default of 35 databases in a backup set. If each database is located on a separate volume, then once the 35-database limit is reached, SnapManager commits the backup set to execute the backup. If the combined number of databases exceeds the maximum limit, the backup grouping feature commits the set as a backup set for the overall current backup. Figure 11 shows the settings in the SMSQL GUI for configuring backup grouping.

Figure 11) Backup grouping setting in the SMSQL GUI.



SnapManager concurrently backs up databases that share the same NetApp storage volume(s). However, SQL Server has a limited set of resources to do those concurrent backups. Therefore it may be necessary to reduce the size of each backup's snapshot so SQL Server resources aren't exhausted.

Microsoft recommends a maximum of 35 databases per backup snapshot when running out of SQL Server thread resources.

Maximum Databases per snapshot: 35

Verification with SnapManager for SQL Server

SnapManager for SQL Server has an important feature for verification. It also supports the verification of backup sets at SnapMirror and SnapVault destinations without breaking the relationships.

Note: Verification on level-2 clones during backup and restore is not supported because this operation in turn creates another level of clone to mount.

Best Practices

- The architecture and version of SQL Server on the database verification server must match the architecture and version of the SQL Server host being backed up by SnapManager.
- The verification server must be connected to the NetApp storage system where the backup Snapshot copies are located. The connectivity can be Fibre Channel, Fibre Channel over Ethernet, or iSCSI. The connectivity of the verification server to the NetApp storage does not have to match the connectivity of the production SQL Server host.
- NetApp recommends using a separate nonproduction SQL Server host for the verification server to offload the I/O traffic of the verification process. NetApp recommends a maximum of four servers for this purpose.
- To use remote verification, configure the transport protocol setting in SnapDrive for Windows on the remote server to be able to locate the SVM on which the LUNs or SMB shares reside.
- Make sure that SMSQL backups on VMDK use a verification server that is running on a virtual machine only.
- Offload Snapshot verification to a dedicated server and schedule verification for off-peak hours by using the SnapManager deferred verification option. `DBCC CHECDB` is a CPU-intensive operation that reduces the performance of Snapshot verification on SQL Server workloads.
- Use a shared mount-point LUN to set verification for a SQL Server failover cluster instance.

5.3 Restore

Depending on the database restore option that you select, SnapManager performs a point-in-time restore, an up-to-the-minute restore, or a marked transaction restore by using the SnapManager backup (database and transaction log) that is available in a centralized shared location. In addition, SnapManager can reseed databases, create cloned replicas of availability-group databases, and perform up-to-the-minute restores of the databases in an availability group.

For example, suppose that you want to restore all databases (including level-2 clones) using SnapRestore to perform this task. The restore operation for a database that resides on a LUN or an SMB share leverages SnapDrive for Windows to perform the mount and copy operations. However, the restore of a database on a VMDK or the restore of a database to an alternate path leverages SnapManager to perform the mount and copy operations. NetApp Virtual Storage Console (VSC) provides integration between VMware® vSphere® and NetApp Snapshot copies; to create backups of SQL Server data that is stored in a VMware virtual disk (VMDK) when VSC is used with SnapManager for SQL Server. In such cases, restores are offloaded from SnapDrive for Windows to VSC.

Note: If you have a single database per VMDK, VSC performs an SMSQL mount-and-copy restore. If you have multiple databases on the VMDK, SnapRestore is used to perform a volume-clone operation.

The unrestricted database layout configuration affects the restore operation when databases are scattered on multiple LUNs across multiple volumes and the user wants to restore specific databases. In this scenario, the restore operation is relatively slower because a mount-and-copy restore is performed instead of a SnapRestore operation.

Best Practices

- When databases are located on a dedicated LUN, the restore operation tends to be faster if the entire database is being restored because SnapManager for SQL Server can take advantage of the SIS-clone capability to perform faster restores.
- The Restore to Alternate Location option works only when restoring to the same instance as the instance on which the backup is located, not to a different SQL Server instance. This feature is not supported for availability groups.

5.4 Reseeding Availability Groups

Introduced with SQL Server 2012, availability groups are a logical grouping of databases that can fail over as a single unit to secondary copies (synchronous or asynchronous). SnapManager for SQL Server helps the restore and reseed functionalities to facilitate faster recovery of the availability group replica to a synchronized state.

In an AlwaysOn environment, availability group failover can occur whenever one or more availability group replicas fail or become corrupted. This failover can be automatic or manual, based on the availability group–replica's configuration. In such cases, the primary replica switches to the secondary role, and one of the secondary replicas becomes the primary replica.

Best Practices
• Always run the reseed wizard on the secondary replica when there is a failure to recover the failed or corrupted replica of the availability group and resync with the primary replica. • For the reseed wizard to complete successfully, the SnapManager service account should have read/write permissions to the SnapManager share that contains the backups of the log files.

6 Virtualization

Virtualizing Microsoft SQL Server optimizes compute resources through server consolidation.

Best Practice
Exclude SQL Server data and log files from real-time virus monitoring and scanning on the SQL Server virtual machines to minimize the performance impact on SQL Server.

6.1 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 2012 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 2012 and Windows Server 2012 provide a host of new features that can be used to effectively virtualize demanding complex database workloads such as online transaction processing/analysis (OLTP/OLAP), data warehousing, and business intelligence, which were not previously considered for virtualization. For information about support for SQL Server 2014 in virtualized environments, refer to Microsoft TechNet.

6.2 VMware

VMware ESXi supports three types of configuration when connecting to shared storage arrays: raw device mapping (RDM), NFS datastores, and VMFS datastores. In virtualized SQL Server deployments that use VMDK or NFS configurations, you must have a dedicated management ESX® server with VSC and the appropriate VMware vCenter™ software installed on it.

Each ESX server should have a service console port defined on the vSwitch that transmits public virtual-machine traffic and on the vSwitch that is configured for IP storage traffic.

RDM

Observe the best practice that follows if you use RDM LUNs.

Best Practice

- Enter your ESX or VSC credentials in SnapDrive when you use RDM LUNs on the virtual machine.

NFS Datastore

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

Best Practices

- Make sure that each NFS export that will be used as an NFS datastore resides on its own volume.
- Use the following best-practice layout for the NFS datastore:
  - One NFS datastore per system database VMDK
  - One NFS datastore per user database and user log; alternatively, separate the user database and the user log on 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.

VMFS Datastore

Observe the best practices that follow if you use VMFS datastores.

Best Practices

- 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 (you can use a separate datastore also).
- Do not mix or keep system databases and user databases in the same VMDK. Do not keep system databases and user databases on separate VMDKs in the same datastore and volume.
- Use NetApp VSC to provision VMDKs to install SQL Server binaries.
- 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.
- Avoid sharing volumes and datastores between different Windows Server machines.

Note: When databases hosted on the VMDKs are replicated by SnapMirror, you cannot clone databases from a SnapMirror destination volume to the local SQL Server instance. However, you can clone databases from destination volumes to a SQL Server host running on the remote virtual machine.

To verify VMware configuration compatibility across the NetApp stack, refer to the Interoperability Matrix Tool (IMT).

7 MetroCluster with Data ONTAP 8.3

MetroCluster provides both local failover (HA) capability as well as site failover capabilities with clustered Data ONTAP by providing RAID-level replication of data between sites. MetroCluster is ideally suited for stand-alone or FCI SQL deployments because SQL availability groups essentially result in two times the number of AG replicas, and hence increased storage space requirements to house these duplicate replicas.

Best Practices
• Make sure that Windows Host Utilities or Microsoft DSM is installed on each host to avoid time-out issues, especially during disk enumeration. • In the case of VMFS/NFS VMDK setups, be sure to wait a minimum of 15 minutes after SwitchOver or SwitchBack so that the storage system details get refreshed in the Virtual Storage Console (VSC). • For databases on SMB shares, SQL Server service needs to be restarted after SwitchOver or SwitchBack. • SnapManager Virtual Instance service needs to be restarted after SwitchBack when using VMFS/NFS VMDKs; restarting SnapDrive for Windows service is not required.

For information about MetroCluster, refer to:

Technical FAQ for MetroCluster in Clustered Data ONTAP 8.3.0

MetroCluster 8.3.0 FAQ and How-to Information

8 Best Practices for Ongoing Management and Monitoring

SMSQL gives you a report with the overall status of operations such as backup, verification, and clone resync. When using NetApp efficiency features, the volumes should be appropriately sized so that autosize and/or autodelete policies are not triggered unless there is an abnormal rate of change or a problem with Snapshot copy retention. NetApp OnCommand management software that includes Operations Manager is the recommended tool to monitor SQL Server volumes for these events. NetApp also recommends it to send notifications to the storage administration team to follow up further with the SQL Server administration team. You can also use SNMP to monitor these events.

After a notification for a volume autogrow or Snapshot autodelete event has been received by the storage administration team, the recommended action is for the storage administration team to examine the affected storage controllers and then follow up with the SQL Server administration team for further administrative actions. A typical cause of volume autosize events is that the rate of change greatly surpassed the rate of change assumption used in sizing the volume. Another cause for volume autosize events is that older Snapshot copies created by SnapManager for SQL Server are not being deleted. As Snapshot copies age, they can grow in size and consume more capacity than originally allocated in the volume. A typical cause of SnapManager for SQL Server not deleting backups is that SnapManager for SQL Server backups are failing. By default, SnapManager for SQL Server does not delete Snapshot copies of older SnapManager for SQL Server backup sets if the backup fails. Another cause for SnapManager for SQL Server not deleting backups is that the SnapManager for SQL Server backup retention policies are not being enforced correctly because Snapshot copies were manually removed outside of SnapManager for SQL Server on the controller itself. Monitoring the health of SnapManager for SQL Server can be done by monitoring for SnapManager for SQL Server event IDs and the enhanced monitoring functionality in SMSQL.

SnapManager operational reports are automatically created for SnapManager configuration, backup, restore, backup set deletion, and other miscellaneous operations. Each report is a log file that includes step-by-step details of the operation, the final status of the operation, and any error messages encountered during the operation.

SnapManager also writes to the Windows event application log. Backup and verification operations that are incomplete or that did not run are logged in the Windows event application log as errors. For cloning operations, all events are logged as informational.

Note: The SnapManager report directory can be local or be configured to use a Windows file share network resource.

SnapManager also supports the generation of operational report e-mails for backup, verification, and cloning operations. The e-mail can be configured to contain a brief summary of the report or to include the report as an attachment.

Monitoring the health of SnapManager for SQL Server can be done by monitoring SnapManager event IDs. To monitor the health of SnapManager retention, use Windows PowerShell commands from the Data ONTAP PowerShell Toolkit and native Windows PowerShell commands.

References

This section lists useful resources and references that can assist you in planning and managing your SQL Server storage environment.

NetApp Storage Systems and Clustered Data ONTAP

- NetApp Storage Systems
- Data ONTAP documentation

SnapDrive for Windows and SnapManager for SQL Server

- NetApp SnapDrive for Windows
- SnapManager for Microsoft SQL Server (SMSQL)

Microsoft SQL Server

- Microsoft SQL Server Customer Advisory Team
- SQL Server Storage Engine Blog
- MSDN Product Documentation (includes SQL Server Books Online)
- SQL Server Best Practices
- Hardware and Software Requirements for Installing SQL Server 2014
- Description of Disaster Recovery Options for Microsoft SQL Server
- Disaster Recovery Articles for Microsoft SQL Server
- Disaster Recovery Planning (Database Engine)
- Introduction to Backup and Restore Strategies in SQL Server
- You Cannot Restore System Database Backups to a Different Build of SQL Server

NetApp SnapMirror

- SnapMirror How-to Guide
- SnapMirror Best Practices Guide

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

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.

Trademark Information

NetApp®
www.netapp.com