Technical Report

SQL Server Database Quiesce Tool (ScSqlApi)

Pat Sinthusan
Geert van Teylingen
March 2022 | TR-4833

In partnership with

Microsoft

Abstract

This document describes the functionality, installation, and operation of the NetApp® SQL Server Database Quiesce Tool (ScSqlApi) to provide Microsoft SQL Server application consistency when used in combination with the Azure NetApp Files snapshot functionality.

TABLE OF CONTENTS

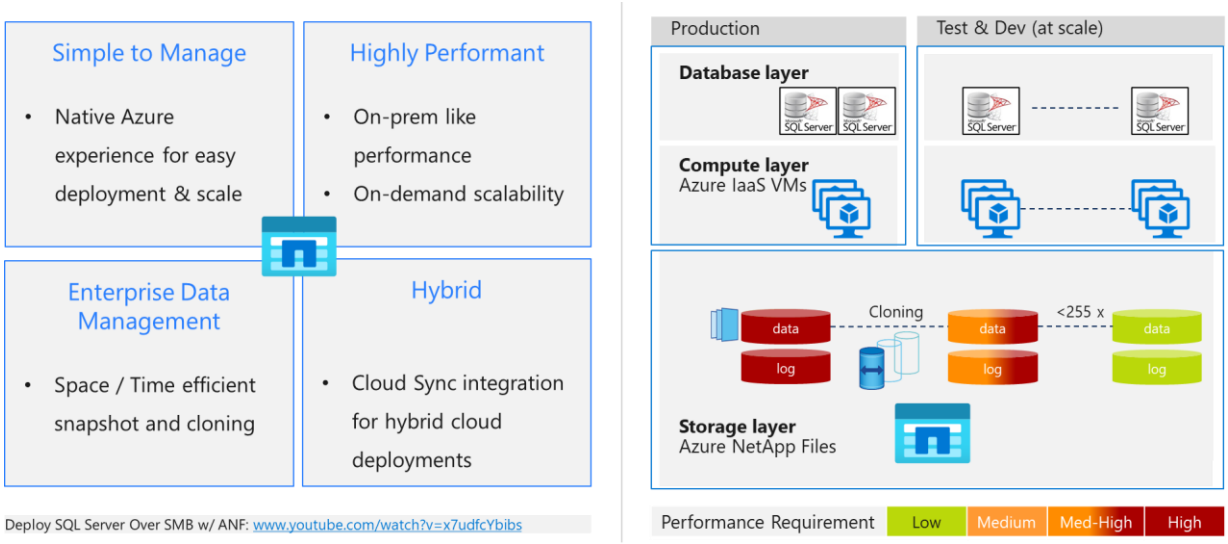
LIST OF FIGURES

Introduction

Managing a very large database (VLDB) is a demanding task for a database (DB) administrator. Although there is no precise definition for a VLDB, a multi-terabyte DB can currently add additional complexity for high availability and backup and restore operations. We are seeing SQL Server installations with DB sizes of over 50TB up to around 200TB for some of our customers. Azure NetApp Files dramatically simplifies the management of SQL Server VLDBs in Azure environments. DB-consistent storage snapshots offer great advantages over the traditional built-in SQL Server backup functionality in terms of speed and added business value. Snapshots can, for example, be utilized for quickly syncing dev/test-systems or a fast recovery from failed operations or DB corruptions.

Businesses have been increasingly migrating on-premises workloads to Azure for a number of reasons, including datacenter consolidation and cost effectiveness. For VLDB lift-and-shift scenarios with DB sizes in the double-digit TB range, virtual machine (VM)-based, Infrastructure as a Service (IaaS) architectures are still a viable option. Memory-optimized Azure VMs like the E-series and M-series offer excellent compute capabilities for demanding DB workloads such as SQL VLDBs or SAP HANA. Azure NetApp Files is an Azure-native NAS storage solution for running high performance SQL workloads in combination with Azure VMs (Figure 1).

Figure 1) Azure NetApp Files overview.



SQL Server has supported SMB since version 2012[1]. Azure NetApp Files offers access to storage through the multichannel-enabled SMB protocol, providing low-latency file storage to SQL Server combined with snapshot capabilities. The ScSqlApi tool uses existing NetApp technology to quiesce a SQL Server DB and take an application-consistent storage snapshot.

The challenge with crash-consistency versus application-consistency with DB backups is that crash-consistent backups can cause unexpected and unpredictable backup results and corrupted data. There are definite differences in how crash-consistent and application-consistent backups handle application data such as Microsoft SQL Server DBs. Crash-consistent backups are unaware of the in-memory data and pending I/O operations from these types of applications. In contrast, application-aware backups are

[1] https://docs.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server-with-smb-fileshare-as-a-storage-option?view=sql-server-2017

aware of these types of transient data and are equipped to handle them. Application-aware backups use Volume Shadow Copy Service (VSS) writers in the Windows VSS service to correctly quiesce applications by flushing memory and pending I/O to disk, and thus back applications up properly with transactional consistency.

NetApp provides a tool that can be executed from within PowerShell that quiesces a SQL Server DB, which in turn allows you to take the application-consistent storage snapshot for backup.

After snapshot creation, the snapshots can be used for data recovery or test and development purposes by using the Azure NetApp Files Restore to New Volume feature.

Requirements

Currently, the tool works with Microsoft SQL Server 2016 through 2019 on all supported versions of the Windows operating system. The Microsoft .NET Framework 4.5.2 is also required.

Limitations

Consider the following limitations when using the ScSqlApi tool:

1. Allows to quiesce one more DBs at the time. The default maximum value of quiesce DBs is 100. This can be changed in the `ScSqlApiServiceHost.exe.config` file.
2. Only supports full backup of the DB. Log backups are not supported.
3. Can only be used with user DBs.
4. Will not take a storage snapshot. Customers must use an external tool to do this. The purpose of the tool is to enable a typical, often scripted, three-step (quiesce, snapshot, unquiesce) application-consistent snapshot workflow.
5. SQL-instance-level quiesce and unquiesce operations are not supported.
6. The tool does not provide restore functionality. Restores should be performed using external tools or the T-SQL command. Virtual-desktop-infrastructure-based restore is not supported.
7. Log backup and log recovery is not supported.
8. Always On availability groups are supported on secondary nodes with the Copy-Only option.
9. Always On failover cluster DBs are supported. The ScSqlApi tool must be installed on all nodes in the cluster.
10. For heavily used, highly transactional DBs, the quiesce operation can take more time to complete. This issue is to be expected because of the operations that must be flushed to SMB volumes before quiesce can occur.

Installation

Prerequisites

- Install SharedManagmentObjects and SQLSysClrTypes, which already included in the zip file.

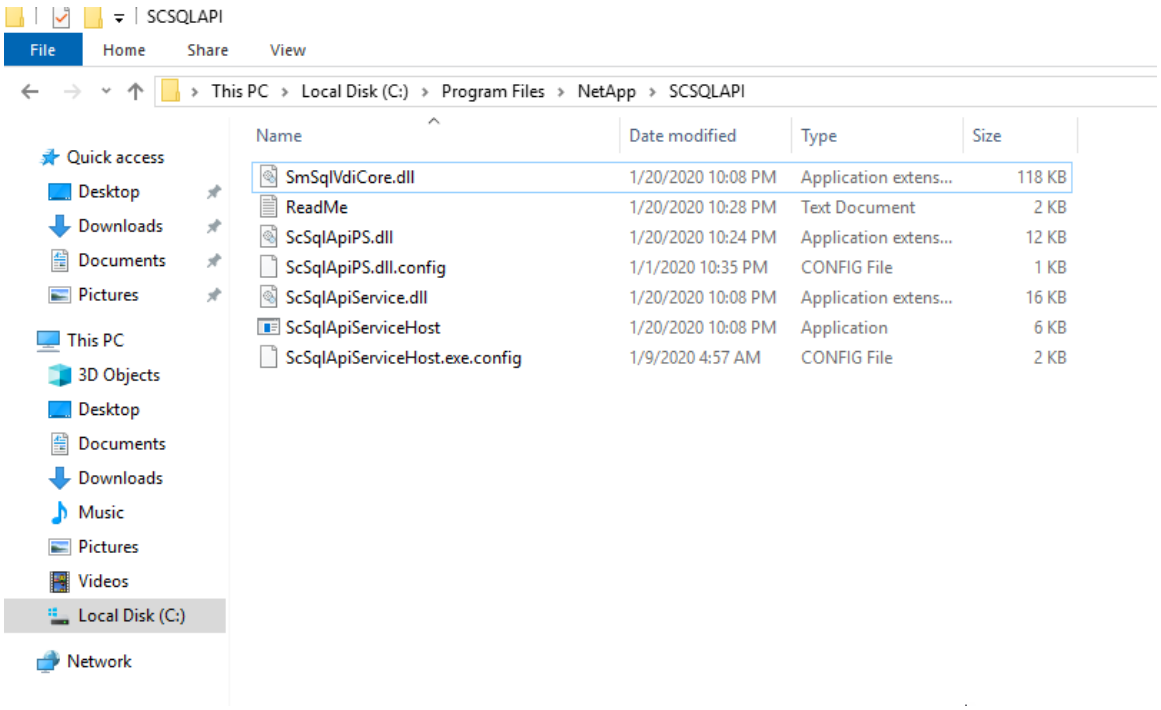
  Note: This is not required if you are using SQL Server 2016 and the client modules are already installed.

- Install the x86 version of Visual C++ Redistributable for Visual Studio 2012 (included in the zip file).

Installation instructions

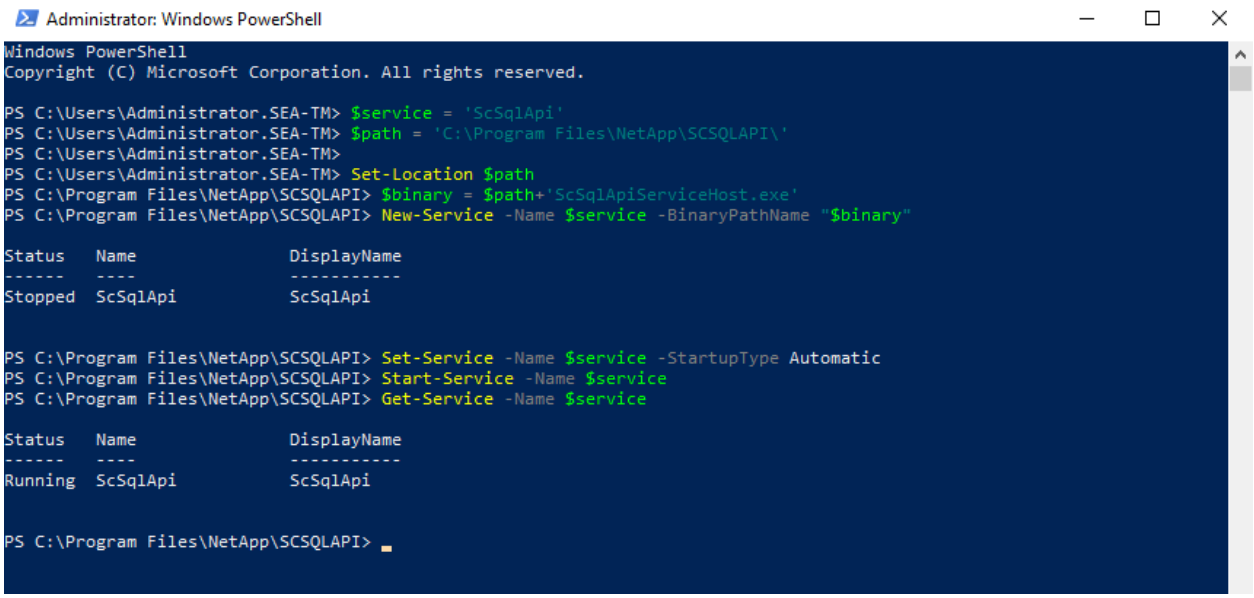
1. Download ScSqlApi.zip.

2. Copy or extract all files in ScSqlApi.zip to the folder `C:\Program Files\NetApp\ScSqlApi`.



3. Start PowerShell as administrator, and run the following commands to register the service with automatic start after reboot:

```
$service = 'ScSqlApi'
$path = 'C:\Program Files\NetApp\SCSQLAPI\'

Set-Location $path
$binary = $path+'ScSqlApiServiceHost.exe'
New-Service -Name $service -BinaryPathName "$binary"
Set-Service -Name $service -StartupType Automatic
Start-Service -Name $serviceGet-Service -Name $service
```



Usage

To use the service, complete the following steps:

1. Open Windows PowerShell, switch to the directory from step 2 in the installation instructions and run the following command to import the PowerShell Module.

```
Import-Module .\ScSqlApiPS.dll
```

2. Run the following command to quiesce and unquiesce the SQL Server DB:

```
New-ScSqlBackup -Database <databaseName1>, <databaseName2>  -SQLInstance <sqlInstance> -Operation
Quiesce -Authentication Windows/SQL -Credential "Windows\Credential"

New-ScSqlBackup -Database <databaseName1>, <databaseName2> -SQLInstance <sqlInstance> -Operation
Unquiesce
```

3. For copy-only backup, specify the copy only option.

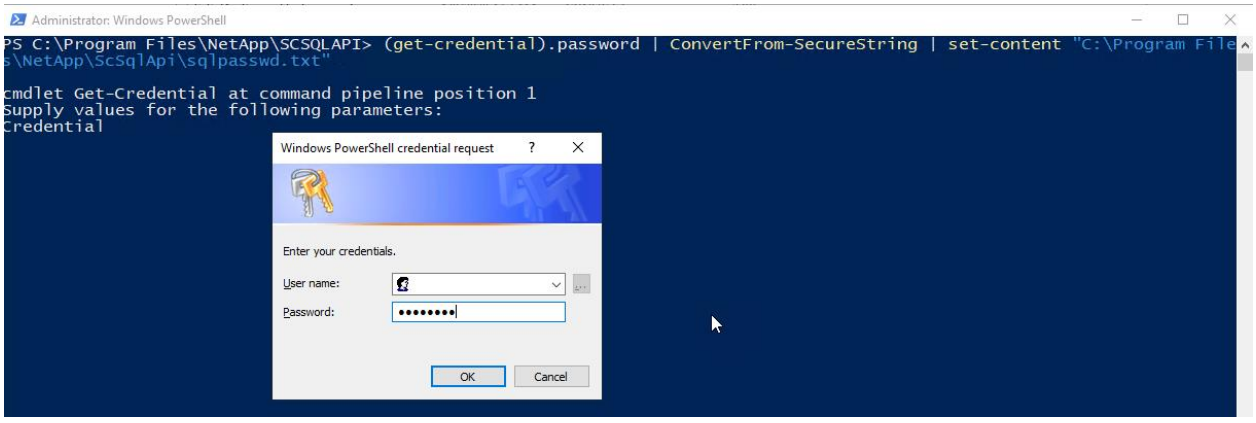
```
New-ScSqlBackup -Database <databaseName> -SQLInstance <sqlInstance> -Operation Quiesce -CopyOnly
-Authentication Windows/SQL -Credential "Windows\Credential"

New-ScSqlBackup -Database <databaseName> -SQLInstance <sqlInstance> -Operation Unquiesce
```

4. The following example scripts show you how to encrypt your password and use a credential object:

```
$password = ConvertTo-SecureString "<SqlUserPassword>" -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential ("<SqlUserName>", $password)
New-ScSqlBackup -Database <databaseName> -SQLInstance <sqlInstance> -Operation Quiesce -
Authentication Windows -Credential $cred
```

5. The following example script shows you how to create encrypted files to conceal SQL Server credentials and read the credentials back to a credential object:

```
(get-credential).password | ConvertFrom-SecureString | set-content "C:\Program
Files\NetApp\ScSqlApi\sqlpasswd.txt"
```



6. Run the following command to get the password credentials that are required to access SQL Server.

```
$passwd = Get-Content " C:\Program Files\NetApp\ScSqlApi\sqlpasswd.txt" | ConvertTo-SecureString
```

Recommendations

If you use the ScSqlApi tool to take a snapshot for the SQL Server DB on Azure NetApp Files, then NetApp makes the following recommendations:

- If the tempdb files have been placed on Azure NetApp Files, the Azure NetApp Files volume should be separated from user DB files.

- Because the ScSqlApi tool quiesces one DB at a time, isolate the files from each DB by placing them into a separate Azure NetApp Files volume.

Support

For support, contact ng-scsqlapi-feedback@netapp.com.

Appendix A: PowerShell example script

The following is an example PowerShell script[2] used to execute the 3-step quiesce, snapshot, and unquiesce process to create an application-consistent snapshot of the DB:

```
#############################################################################################################
#############################
#This section below is to run to generate encrypted Files for password and tenantid
#$Global:apifolder = 'C:\Program Files\NetApp\ScSqlApi\'
#(get-credential).password | ConvertFrom-SecureString | set-content ($apifolder+'azpasswd.txt')
#(get-credential).password | ConvertFrom-SecureString | set-content ($apifolder+'sqlpasswd.txt')
#$tenantid = '1f0daac7-c44b-4ae6-b3b6-407b69b407aa'
#$tenantid | ConvertTo-SecureString -AsPlainText -Force | ConvertFrom-SecureString |  Set-Content
-Path ($apifolder+'tenant.txt')
#############################################################################################################
#############################

$Global:apifolder = 'C:\Program Files\NetApp\ScSqlApi\'
$Global:Poolname = 'DEMOPOOL01'
$Global:RGname = 'ANFdemo.rg'
$Global:Accountname = 'ANF-Demo-Account'
$Global:Location = 'West US 2'
$Global:SQLLogin = 'leejilesoutlook\pats'
$Global:azLogin = 'pats@leejilesoutlook.onmicrosoft.com'
$Global:vols = ('SQLProd01Data01', 'SQLProd01Data02', 'SQLProd01Log')

Function Get-Password ($PasswordType){
    Switch ($PasswordType){
        'az'  {$passwd = Get-Content ($apifolder+'azpasswd.txt') | ConvertTo-SecureString }
        'sql' {$passwd = Get-Content ($apifolder+'sqlpasswd.txt') | ConvertTo-SecureString }
    }
    return $passwd
}

Function New-Snapshot{
    $ss = 'ss-'+ (Get-Date -Format "MM-dd-yyyy-HH-mm-ss")


     $TakeSnapshot = {
        param($vol, $ss)
        New-AzNetAppFilesSnapshot -PoolName $Poolname -ResourceGroupName $RGname -AccountName -
$Accountname VolumeName $vol -Location $location -name $ss
    }

    foreach ($vol in $vols){
        write-host $vol
        Start-job -ScriptBlock $TakeSnapshot -ArgumentList $vol, $ss
    }
    Get-job | Wait-Job
}

Function Set-DB ($action){
    Set-Location $apifolder
```

[2] This script is provided for example purposes only.

```
    Import-Module $apifolder\ScSqlApiPS.dll
    $password = Get-Password 'sql'
    $credential = New-Object System.Management.Automation.PSCredential($SQLLogin, $password)
    New-ScSqlBackup -Database SeattleRetail -SQLInstance 'SQLProd01' -Operation $action -
Authentication Windows -Credential $credential
}

$Connect_Az = {
    Import-Module Az
    Import-Module Az.NetAppFiles
    $secretStuff = Get-Content  -Path ($apifolder+'tenant.txt') | ConvertTo-SecureString
    $tenantid =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStrin
gToBSTR(($secretStuff))))
    $password = Get-Password 'az'
    $credential = New-Object System.Management.Automation.PSCredential($azLogin, $password)
    Connect-AzAccount -Credential $credential -Tenant $tenantid
}

cls
Measure-Command {Start-Job $Connect_Az} | Select TotalSeconds
Measure-Command {Set-DB 'Quiesce'} | Select TotalSeconds
Measure-Command {New-Snapshot} | Select TotalSeconds
Measure-Command {Set-DB 'UnQuiesce'} | Select TotalSeconds
```

Appendix B: Known error messages

See the following known error messages:

1. Incorrect instance name provided.

```
Starting Quiesce operation on database [Test1] ...
New-ScSqlBackup : Failed to connect to server SCSPR177851003\INST1.
```

2. Incorrect username and password provided.

```
Starting Quiesce operation on database [Test2] ...
New-ScSqlBackup : The user name or password is incorrect.
```

3. Quiesce has been repeatedly issued.

```
Starting Quiesce operation on database [Test2] ...
New-ScSqlBackup : Backup already running for database: Test2
```

4. Unquiesce has been repeatedly issued.

```
Starting Unquiesce operation on database [Test2] ...
New-ScSqlBackup : No running backup found to Unquiesce
```

5. SQL Server instance is stopped.

```
Starting Quiesce operation on database [Test2] ...
New-ScSqlBackup : Failed to connect to server SCSPR1778511003\INST1
```

6. Quiesce is issued simultaneously on two databases (DBs).

```
Starting Quiesce operation on database [Test4] ...
New-ScSqlBackup : Backup already running for database: Test3
```

7. Unquiesce is issued simultaneously on two DBs.

```
Starting Unquiesce operation on database [Test1] ...
New-ScSqlBackup : No running backup found to Unquiesce
```

8. Parameters are not completely specified.

```
New-ScSqlBackup : Missing an argument for parameter 'Database'. Specify a parameter of type
'System.String' and try again
```

9. DB does not exist.

```
Starting Quiesce operation on database [Test2ww] ...
New-ScSqlBackup : GetConfiguration timed out
```

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites

- Azure NetApp Files documentation
  https://docs.microsoft.com/en-us/azure/azure-netapp-files/
- NetApp Product Documentation
  https://www.netapp.com/support-and-training/documentation/
- Deploy SQL Server Over SMB with Azure NetApp Files
  https://www.youtube.com/watch?v=x7udfcYbibs&t=2s
- Deploy SQL Server Always On Failover Cluster over SMB with Azure NetApp Files
  https://www.youtube.com/watch?v=zuNJ5E07e8Q&t=120s
- Deploy Always On Availability Groups With Azure NetApp Files
  https://www.youtube.com/watch?v=beWCQi1hNTI
- TR-4888: SQL Server on Azure using Azure NetApp Files
  https://www.netapp.com/pdf.html?item=/media/27154-tr-4888.pdf

TR-4833-0322

NetApp