



Der Weg zu neuer Produktivität

Mit Powershell die DBA oder DevOps-Tätigkeiten vereinfachen oder automatisieren

About Me



Björn Peters

SQL Server Lead DBA, Atos IT Information Technologies GmbH
PASS Deutschland e.V. Member, Volunteer, Speaker
Azure Meetup Hamburg Leader
Father, Husband, Snowboarder, Cyclist, Geek



www.sql-aus-hamburg.de



info@sql-aus-hamburg.de



[@SQL_aus_HH](https://twitter.com/SQL_aus_HH)



[SQL_aus_HH](https://www.instagram.com/SQL_aus_HH)



Agenda

Deploy Ola Hallengren Maintenance Scripts with Powershell

- Config OH MS with DCS

Powershell Modul-Set dbatools.io

Powershell Modul-Set dbareport.io

Powershell Script „*Config after Installation*“

Desired State Configuration

Deploy Ola Hallengren via Powershell

Ola Hallengren

Die SQL Server Maintenance Solution um Backups, Integritäts-Checks sowie Index und Statistiken aktuell zu halten.

Lauffähig mit den SQL Server Versionen / Editionen

- SQL Server 2005
- SQL Server 2008
- SQL Server 2008 R2
- SQL Server 2012
- SQL Server 2014
- SQL Server 2016

Diese Skripte wurden für nahezu alle mission-critical-Umgebungen entwickelt und wird in einer Vielzahl von Unternehmen weltweit genutzt.



Ein Überblick

Basiert auf Stored Procedures und Funktionen

Supports Microsoft SQL Server 2005, SQL Server 2008, SQL Server 2008 R2, SQL Server 2012, SQL Server 2014 and SQL Server 2016

DatabaseBackup – Backup

DatabaseIntegrityCheck – Integrity Check

IndexOptimize – Index and Statistics Maintenance

Modular – Einzelnen Funktionen können auch einzeln installiert werden

kostenlos!

Auswahl der Ziel-Datenbanken

Einheitliche Parameter für die einzelnen Prozeduren - DatabaseBackup, DatabaseIntegrityCheck und IndexOptimize

“All Databases”, “All User Databases” oder “All System Databases”

Auch nur einzelne oder eine Liste von Datenbanken möglich

ebenso ein “Exclude” von bestimmten Datenbanken

genauso wie die Auswahl von Datenbanken mittels Wildcards

Defaults

Out of the box

Keine Schedules, nichts läuft von alleine

Man muss also erst Schedules erstellen

Manuelle Pläne erstellen und diese zur späteren Verwendung speichern

Challenges with larger environments

Deployments und Updates müssen auf zahlreichen Maschinen erfolgen

Einheitliche Laufpläne

Zentralisiertes Fehlermanagement bzw Verwaltung

Defaults

SQL Server Agent

-- Specify whether jobs should be created.

SET @CreateJobs = 'Y'

-- Specify the backup root directory.

SET @BackupDirectory = N'C:\Backup'

-- Time in hours, after which backup files are deleted. If no time is specified, then no backup files are deleted.


SET @CleanupTime = NULL

-- Specify the output file directory. If no directory is specified, then the SQL Server error log directory is used.

SET @OutputFileDirectory = NULL

-- Log commands to a table.

SET @LogToTable = 'Y'

 IndexOptimize - USER_DATABASES

 Output File Cleanup

 sp_delete_backuphistory

 sp_purge_jobhistory

 Job Activity Monitor

DatabaseBackup - Beispiele

```
EXECUTE dbo.DatabaseBackup  
@Databases = 'USER_DATABASES',  
@Directory = 'C:\Backup',  
@BackupType = 'FULL',  
@Verify = 'Y',  
@Compress = 'Y',  
@Checksum = 'Y',  
@CleanupTime = 24
```

```
EXECUTE dbo.DatabaseBackup  
@Databases = 'USER_DATABASES',  
@Directory = 'C:\Backup',  
@BackupType = 'FULL',  
@Compress = 'Y',  
@Checksum = 'Y',  
@BufferCount = 50,  
@MaxTransferSize = 4194304,  
@NumberOfFiles = 64
```

IndexOptimize - Kernfunktionen

Überprüfen der Fragmentation

3 Fragmentation Gruppen: low, medium and high

frei wählbare Fragmentierungsgrenzen für jede Gruppe

frei wählbare Aktionen für jede Gruppe

Failback-Aktionen verfügbar

IndexOptimize - Beispiele

```
EXECUTE dbo.IndexOptimize
@Databases = 'USER_DATABASES',
@FragmentationLow = NULL,
@FragmentationMedium = INDEX_REORGANIZE,INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE',
@FragmentationHigh = 'INDEX_REBUILD_ONLINE,INDEX_REBUILD_OFFLINE',
@FragmentationLevel1 = 5,
@FragmentationLevel2 = 30
```

```
EXECUTE dbo.IndexOptimize
@Databases = 'USER_DATABASES',
@FragmentationLow = NULL,
@FragmentationMedium = NULL,
@FragmentationHigh = NULL,
@UpdateStatistics = 'ALL',
@OnlyModifiedStatistics = 'Y'
```

DatabaseIntegrityCheck

DBCC CHECKDB

PHYSICAL_ONLY and NOINDEX for very large databases

MaxDOP

AvailabilityGroupReplicas

DatabaseIntegrityCheck - Beispiele

```
EXECUTE dbo.DatabaseIntegrityCheck  
@Databases = 'USER_DATABASES',  
@CheckCommands = 'CHECKDB',  
@PhysicalOnly = 'Y'
```

```
EXECUTE dbo.DatabaseIntegrityCheck  
@Databases = 'AdventureWorks',  
@CheckCommands = 'CHECKFILEGROUP',  
@FileGroups = 'AdventureWorks.PRIMARY'
```


schon Fragen ???



GO is not SQL

...

GO



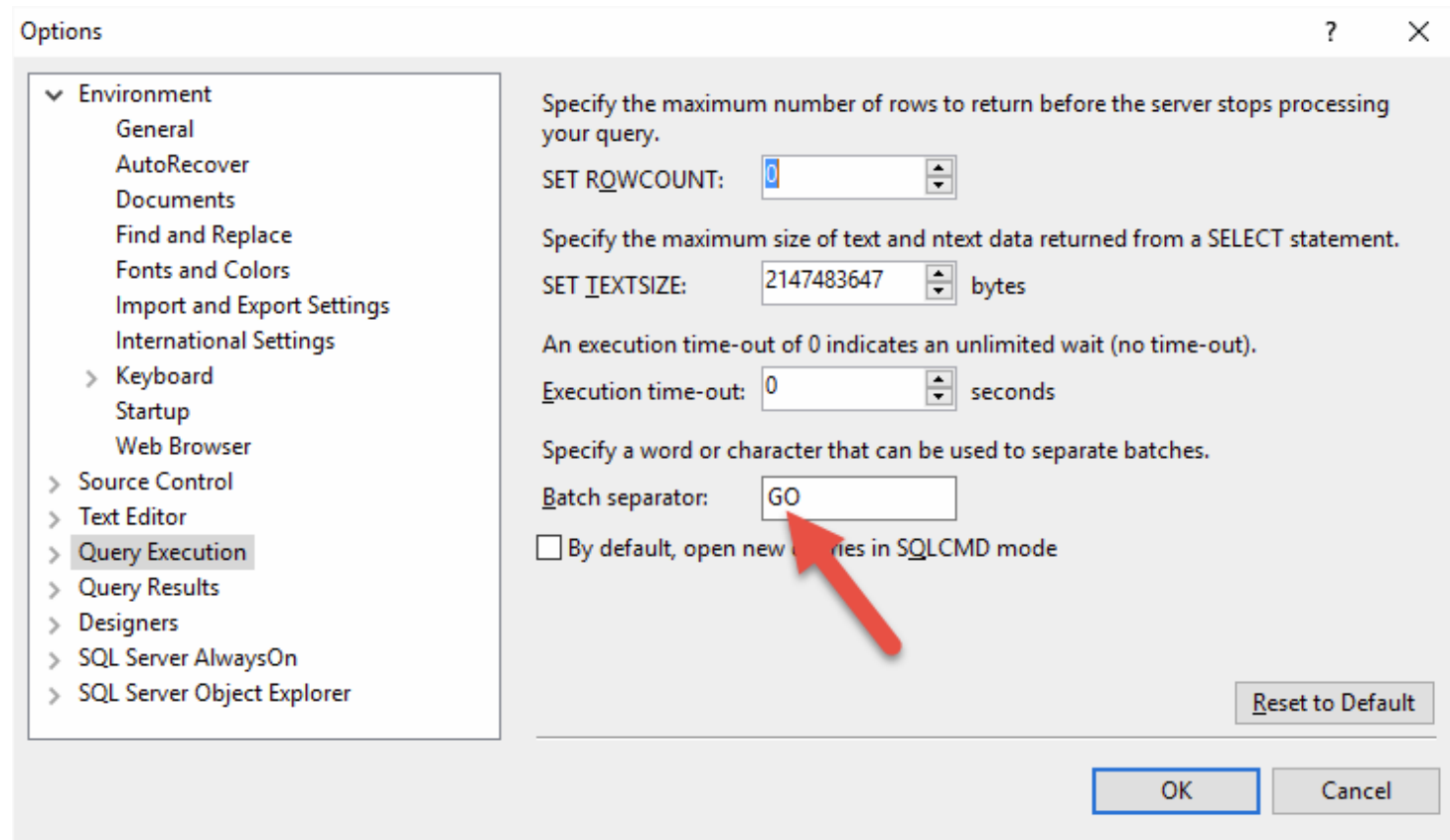
NO ANSI SQL !!!

CREATE PROCEDURE [dbo].[CommandExecute]

@Command nvarchar(max),

...

GO – Batch-Separator



Powershell - Parameter

```
Param(
```

```
    [parameter(
```

```
        Mandatory = $true,  
        ValueFromPipeline = $true
```

```
    )]
```

```
    [string]$SqlServer,  
    [string]$BackupDir,  
    [int]$CleanupTime,  
    [string]$Schedule
```

```
)
```

Skript-Ansätze / -Ideen

```
$script=@()  
[string]$scriptpart
```

```
$fullscript= Get-Content $MaintenanceSolution
```

```
foreach ($line in $fullscript) {  
    if($line -ne "GO") {  
        $scriptpart += $line + "`n"  
    } else {  
        $properties = @{Scriptpart=$scriptpart}  
        $newscript = New-Object PSObject -Property $properties  
        $script += $newscript  
        $scriptpart=""  
    }  
}
```

```
if($line -ne "GO") {  
    if ( $BackupDir -and $line -match "Specify the backup root directory") {  
        $line = $line.Replace("C:\Backup", $BackupDir)  
    }  
    if( $CleanupTime -and $line -match "Time in hours, after which backup files are deleted") {  
        $line = $line.Replace("NULL", $CleanupTime)  
    }  
    $scriptpart+=$line+"`n"  
}
```


Demo

PREPARE OUR ENVIRONMENT
DEPLOY O.H. MAINTENANCE-SOLUTION

Offene Fragen ???



Links

Download script:

<http://ola.hallengren.com/scripts/MaintenanceSolution.sql>

Documentation:

<http://ola.hallengren.com/Documentation.html>

License:

<http://ola.hallengren.com/License.html>

dbatools - Powershell Module

dbatools is a free PowerShell module with over 180 SQL Server administration, best practice and migration commands included.



dbatools.io

dbatools is a free PowerShell module with over 180 SQL Server best practice, administration and migration commands included.

Introduced by Chrissy LeMaire and enriched by the community

Chrissy LeMaire, Creator, SQL DBA & Cloud and Datacenter Management (PowerShell) MVP

Major Contributors 2017



Claudio Silva



Rob Sewell



Constantine Kokkinos



Klaas Vandenberghe



Simone Bizzotto



Shawn Melton



Mitchell Hamann



Stuart Moore



Stephen Bennett



Chris Sommer



Warren Frame



Nic Cain



Garry Bargsley



Iraklis Evangelinos



Brandon Abshire



Drew Furgiuele



Fred Weinmann



Enrico van de Laar

Migration Commands

Everything

Start-SqlMigration

Server

Copy-SqlCustomError

Copy-SqlSpConfigure

Export-SqlSpConfigure

Import-SqlSpConfigure

Databases

Copy-SqlDatabase

Copy-SqlDatabaseAssembly

Copy-SqlSysDbUserObjects

Export-DbaAvailabilityGroup

Test-SqlMigrationConstraint

Security

Copy-SqlLogin

Copy-SqlCredential

Copy-SqlAudit

Copy-SqlAuditSpecification

Export-SqlLogin

Export-SqlUser

Server Objects

Copy-SqlBackupDevice

Copy-SqlEndpoint

Copy-SqlLinkedServer

Copy-SqlServerTrigger

SSIS

Copy-SqlSsisCatalog

Management

Copy-SqlPolicyManagement

Copy-SqlDataCollection

Copy-SqlResourceGovernor

Copy-SqlExtendedEvent

Copy-SqlDatabaseMail

Registered Servers

Copy-SqlCentralManagementServer

SQL Server Agent

Copy-SqlServerAgent

Copy-SqlJob

Copy-SqlSharedSchedule

Copy-SqlAgentCategory

Copy-SqlAlert

Copy-SqlOperator

Copy-SqlProxyAccount

Beispiel - Copy-SqlLogin

This command migrates logins from source to destination SQL Servers. Supports SQL Server versions 2000 and above. Migrates logins with SIDs, passwords, defaultdb, server roles & securables, database permissions & securables, login attributes (enforce password policy, expiration, etc).

By default, all logins with the exception of system (####Example Login## and local (SERVERNAME\administrators) logins are copied. The -Logins parameter is autopopulated for command-line completion and can be used to copy only specific logins.

If the login already exists on the destination, it will be skipped unless -Force is specified. Force drops and recreates the login.

Examples

To copy all logins from sqlserver to sqlcluster, using integrated (Windows) login. If logins with the same name exist on sqlcluster, they will be skipped.

```
Copy-SqlLogin -Source sqlserver -Destination sqlcluster
```

To copy only two SQL logins, netnerds and realcajun, as well as one Windows login 'base\ctrlb'

```
Copy-SqlLogin -Source sqlserver -Destination sqlcluster -Logins netnerds, realcajun, 'base\ctrlb'
```

Demo

SPEED UP THE MIGRATION OF SQL SERVER

Best Practice Commands

Expand-SqlTLogResponsibly
Find-SqlDuplicateIndex
Find-SqlUnusedIndex
Get-DbaLastBackup
Get-DbaLastGoodCheckDb
Get-DbaMaxMemory
Repair-DbaServerName

Set-DbaDatabaseOwner
Set-DbaJobOwner
Set-DbaMaxDop
Set-DbaPowerPlan
Set-DbaMaxMemory
Set-SqlTempDbConfiguration
Test-DbaMaxDop

Test-DbaDatabaseCollation
Test-DbaDatabaseCompatibility
Test-DbaDatabaseOwner
Test-DbaDiskAlignment
Test-DbaDiskAllocation
Test-DbaJobOwner
Test-DbaMaxMemory

Test-DbaOptimizeAdHoc
Test-DbaPowerPlan
Test-DbaServerName
Test-DbaSpn
Test-DbaValidLogin
Test-DbaVirtualLogFile
Test-SqlTempDbConfiguration

Beispiel - Test-SqlTempDbConfiguration

Evaluates TempDB against a set of rules to match best practices. The rules are:

- TF 1118 enabled: Is Trace Flag 1118 enabled (See KB328551).
- File Count: Does the count of data files in TempDB match the number of logical cores, up to 8.
- File Growth: Are any files set to have percentage growth, as best practice is all files have an explicit growth value.
- File Location: Is TempDB located on the C:\? Best practice says to locate it elsewhere.
- File MaxSize Set(optional): Do any files have a max size value? Max size could cause TempDB problems if it isn't allowed to grow.

Other rules can be added at a future date. If any of these rules don't match recommended values, a warning will be thrown.

Examples

To check a SQL Server instance:

`Test-SqlTempDbConfiguration -SqlServer localhost`

General Administration Commands

Find-DbaAgentJob	Get-DbaRestoreHistory	Get-SqlRegisteredServerName	Set-DbaDatabaseState
Find-DbaDatabase	Get-DbaRoleMember	Get-SqlServerKey	Set-DbaSpn
Find-DbaOrphanedFile	Get-DbaRunningJob	Install-SqlWhoIsActive	Show-SqlDatabaseList
Find-DbaStoredProcedure	Get-DbaClusterActiveNode	New-DbaDatabaseSnapshot	Show-SqlServerFileSystem
Get-DbaAgentAlert	Get-DbaDiskSpace	Read-DbaBackupHeader	Show-SqlWhoIsActive
Get-DbaAgentOperator	Get-DbaMemoryUsage	Remove-DbaBackup	Sync-SqlLoginPermissions
Get-DbaAvailabilityGroup	Get-DbaMsdtc	Remove-DbaDatabaseSnapshot	Test-DbaConnectionAuthScheme
Get-DbaBackupHistory	Get-DbaPageFileSetting	Remove-DbaSpn	Stop-DbaProcess
Get-DbaDatabaseFreespace	Get-DbaSpConfigure	Remove-SqlDatabaseSafely	Test-DbaFullRecoveryModel
Get-DbaDatabaseSnapshot	Get-DbaSpn	Remove-SqlOrphanUser	Test-SqlNetworkLatency
Get-DbaDatabaseState	Get-DbaStartupParameter	Rename-DbaLogin	Test-SqlPath
Get-DbaDetachedDatabaseInfo	Get-DbaTcpPort	Repair-SqlOrphanUser	Update-SqlWhoIsActive
Get-DbaHelpIndex	Get-DbaTrigger	Reset-SqlAdmin	Watch-SqlDbLogin
Get-DbaPermission	Get-DbaUptime	Resolve-DbaNetworkName	
Get-DbaProcess	Get-DbaXEventsSession	Restore-DbaFromDatabaseSnapshot	

Beispiel - Repair-SqlOrphanUser

Repairing all users within database 'db2'. Some users does not have their matching login.

```
PS C:\Windows\system32> Repair-SqlOrphanUser -SqlServer sql2008 -Databases db2
Attempting to connect to Sql Server..
Validating users on database '[db2]'

User 'TestOrphan1' mapped with their login
WARNING: Orphan user TestOrphan10 does not have matching login.
WARNING: Orphan user TestOrphan2 does not have matching login.

User 'TestOrphan3' mapped with their login
WARNING: Orphan user TestOrphan4 does not have matching login.

User 'TestOrphan5' mapped with their login
WARNING: Orphan user TestOrphan6 does not have matching login.

User 'TestOrphan7' mapped with their login
WARNING: Orphan user TestOrphan8 does not have matching login.

User 'TestOrphan9' mapped with their login
Total Elapsed time: 00:00:00.0744474
```

Repair-SqlOrphanUser -SqlServer sql2008 -Databases db2

Demo

AUTOMATING DATABASE COPY-JOBS

Fragen zu dbatools ???



Links

Webseite:

<https://dbatools.io/>

Download Moduls / Github:

<https://github.com/sqlcollaborative/dbatools>

Documentation:

<https://dbatools.io/functions/>

Videos:

https://www.youtube.com/channel/UC1qrq0zfsW5M_Uf9-vBrsiw

License:

<https://github.com/sqlcollaborative/dbatools/blob/master/LICENSE.txt>

dbareports

Build your own SQL Server Inventory
with powershell



dbareports.io

dbatools is a free PowerShell module with over 180 SQL Server best practice, administration and migration commands included.

Introduced by Chrissy LeMaire and enriched by the community

Chrissy LeMaire, Creator, SQL DBA & Cloud and Datacenter Management (PowerShell) MVP

Major Contributors 2017



Claudio Silva



Rob Sewell



Constantine Kokkinos



Klaas Vandenberghe



Simone Bizzotto



Shawn Melton



Mitchell Hamann



Stuart Moore



Stephen Bennett



Chris Sommer



Warren Frame



Nic Cain



Garry Bargsley



Iraklis Evangelinos



Brandon Abshire



Drew Furguele



Fred Weinmann



Enrico van de Laar

AUTOMATED REPORTS FOR DBAS

Gather, store and display your SQL Server estate data. Automatically, without fuss and free.

DOWNLOAD

GET STARTED

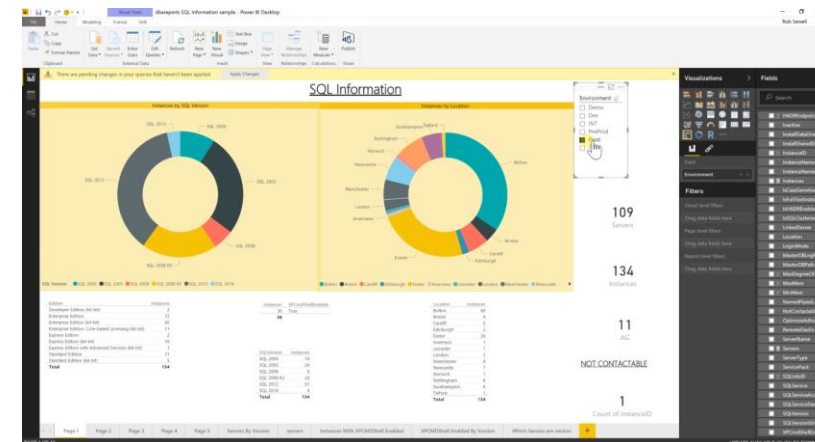
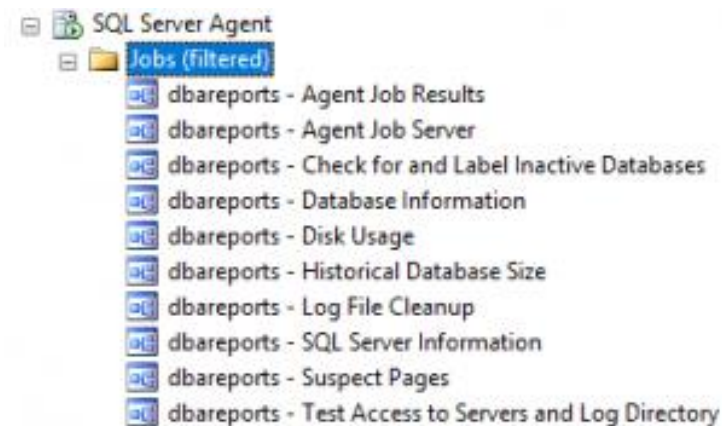
Einblicke

Overall Information: Number of Servers, Instances, Databases, Environments, Clients, Locations

Operating System: HostName, Operating System version, IP Addresses, RAM, CPU

Instance: SQL version, edition, collation, service accounts, memory settings, default locations, configuration

Database: Collation, Compatibility, last backup, owner, space used, space available



Demo

EXEMPLARY : POWER BI REPORTS

Offene Fragen ???



Links

Webseite:

<https://dbareports.io/>

Download Moduls / Github:

<https://github.com/sqlcollaborative/dbareports>

Documentation:

<https://dbareports.io/getting-started/>

Videos:

https://youtu.be/da6S1j32_Ok

License:

<https://github.com/sqlcollaborative/dbareports/blob/Development/LICENSE>

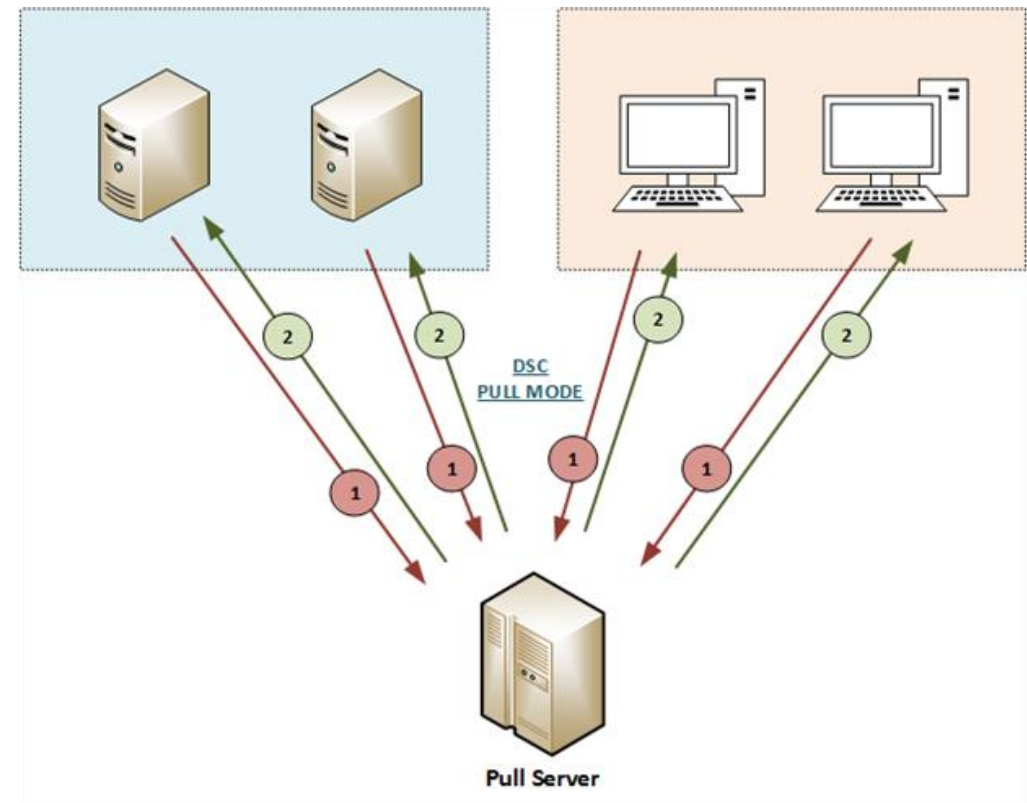
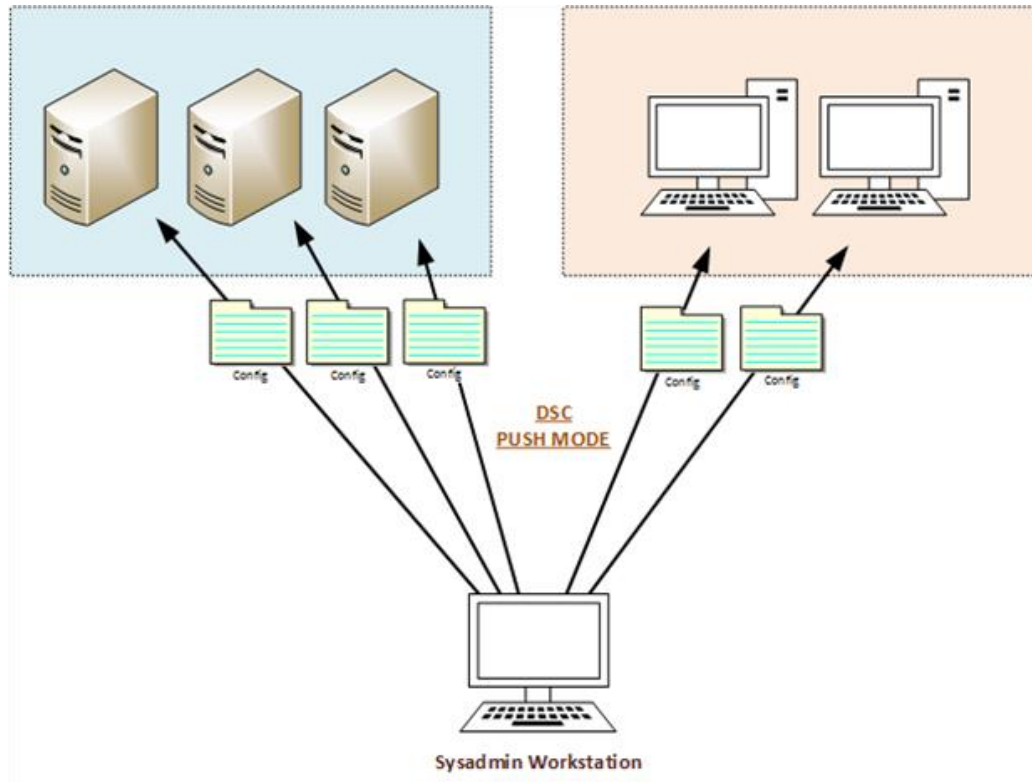
DSC – Desired State Configuration

DSC ist eine Verwaltungsplattform in PowerShell, die es Ihnen ermöglicht, Ihre IT- und Entwicklungsinfrastruktur mit der Konfiguration als Code zu verwalten.

To be continued...



DSC – Pull- und Pull-Verfahren



Special Thanks

Chrissy LeMaire:

Twitter | <https://dbatools.io/>

Ola Hallengren:

Twitter | <http://ola.hallengren.com>

Andre Kamann:

Twitter | <http://clouddba.io/>