# Modul 7: toDos

# Labname

*Ziel:*

*Systemvoraussetzungen:*

*Tools:*

*Dauer:*

*Autor: Andreas Rauch*

*Letzte Änderung: 01.09.2021*

*Falls nötig notwendige Informationen:*

1. Fehlende Logins übertragen

Viele der Hochverfügbarkeitslösungen erstellen einen Kopie einer DB auf einen anderen Server, nehmen aber die Inhalte der Systemdatenbanken nicht mit. Im Folgenden wollen wir einige Probleme beseitigen.

Um das Problem der verwaisten User zu verdeutlichen legen wir einen Benutzer in einer beliebigen Datenbank an- bspw. Northwind. Führen Sie das folgende Script dazu aus.  
  
USE [master]

GO

CREATE LOGIN [Susi]

WITH PASSWORD=N'123'

, CHECK\_POLICY=OFF

GO

USE [Northwind]

GO

CREATE USER [Susi] FOR LOGIN [Susi]

GO

USE [MASTER]

GO

DROP LOGIN [Susi]

* 1. Methode 1 *sp\_users\_change\_login*

Führen Sie folgende Befehle der Reihen nach auf der Datenbank aus

sp\_change\_users\_login 'Report'

exec sp\_change\_users\_login 'Auto\_fix','Susi',NULL,'ppedv2020!'

Das Ergebnis sollte sein, dass Susi als Orphaned User erkannt wurde inkl der SID.

Der zweite Befehl sollte das Login angelegt haben. Die SIDs zwischen Login und User sollten identisch sein.

Das lässt sich sehr leicht mit einer einfachen Abfrage prüfen:  
  
select SID from syslogins where name like 'SUSI'

select SID from sysusers where name like 'SUSI'

* 1. Methode 2: *sp\_revoke\_login*Hier wird es ein wenig komplizierter. Die Prozedur gibt’s nicht ab Werk ☹. Wir müssen Sie uns besorgen.
     1. Verwenden Sie eine beliebige Suchmaschine und geben sp\_help\_revlogin ein. Sie sollten dabei auf eine Microsoft Seite stoßen. https://docs.microsoft.com....  
        genauer gesagt diese hier: https://docs.microsoft.com/de-de/troubleshoot/sql/security/transfer-logins-passwords-between-instances
     2. Kopieren Sie unter Punkt 2 der Website das rel. lange Script und führen sie es auf dem Server in der master Datenbank aus, welche das Login noch besitzt.
     3. Führen Sie nun die Prozedur sp\_help\_revlogin auf der master DB aus.  
        Sie werden im Ergebnis erkennen, dass die Ausgabe ein Sammlung von SQL Statements enthält, welche Logins – Windows und SQL Authentifizierug – anlegen kann.

Tipp: Windows Konten innerhalb einer Domäne zu übertragen ist nicht notwendig, da SQL Server die Windows SID verwendet. Also werden die auf anderen Server konsequenterweise immer identisch sein. Folglich macht sp\_hepl\_revlogin zunächst nur für SQL authentifizierte Konten Sinn.

Übertragen Sie nun ein SQL Konto von HV-SQL1 auf HV-SQL2, das noch nicht auf HV-SQL2 existiert. Zum Beispiel das Konto [Susi] aus der vorherigen Übung.

* + 1. Prüfen Sie, ob die SID auf dem HV-SQL2 identisch ist mit der SID auf dem HV-SQL1

1. Fehlende Jobs übertragen

Auch Jobs werden nicht übertragen, da diese - inklusive Zeitplänen und Verlauf - in der msdb gespeichert werden. Hier gibt es Workarounds, wie man das auf einfache Weise lösen könnte.

* 1. Skript erstellen

Klicken Sie mit der rechten Maustaste, auf einen Job und lassen Sie sich ein Skript erstellen. Führen Sie nun dieses Skript auf dem HV-SQL2 Server aus.  
  
Im Falle eines Fehlers: Korrigieren sie diesen 😉  
  
Tipp: Gelinde gesagt ist das Erstellen eines neuen Wartungsplanes schneller bewerkstelligt als das Übertragen eines bestehenden. Der Grund liegt darin, dass Wartungspläne als SSIS Paket abgespeichert werden, was das ganze deutlich verkompliziert.

1. Contained Databases

..sind keine Wundermittel, da Sie nicht alles beinhalten, was eine Datenbank benötig. Aber das Problem der fehlenden Logins ist lösbar. Um auf einem Server Contained Databases zu betrieben sind nur wenige Schritte notwendig:

**Aktivieren des Features🡪 Aktivieren der Eigenständigkeit der Datenbank🡪 kein Neustart**

* 1. Um auf einem Server Contained Database zu aktivieren, öffnen Sie die Eigenschaften des Server (HV-SQL1).
  2. Unter dem Punkt **Erweitert** setzen Sie den Wert für ***Eigenständige Datenbank*** auf ***true***
  3. Legen Sie mit Hilfe des SSMS eine neue Datenbank mit dem Namen *ContDB* an. Im Dialog können Sie gleich unter **Optionen** den Menüpunkt **Eigenständigkeitstyp** auf ***Teilweise*** festlegen.  
       
     Ein Bild, das Text enthält.

     Automatisch generierte Beschreibung

Nun kommen wir zum Test. Wir legen einen Benutzer ohne Login an und versuchen anschließend ohne Login eine Verbindung mit der DB herstellen zu können.

* 1. Gehen Sie der gerade angelegten Datenbank **ContDB** zum Bereich **Sicherheit🡪Benutzer**.   
     Wenn Sie nun mit der *rechten Maustaste* einen *neuen Benutzer* anlegen, sollte folgender Dialog erscheinen:  
       
     Ein Bild, das Text, Tisch enthält.

     Automatisch generierte Beschreibung
  2. Vergleichen Sie diesen Dialog mit dem Dialog der in der Datenbank northwind erscheint. Was stellen Sie fest?

|  |  |
| --- | --- |
| Name | ContUserX |
| Kennwort | ppedv2020! |

* 1. Legen Sie nun in der ContDB einen Benutzer an:
  2. Gehen Sie nun im SSMS zu **Datei🡪Neu🡪Datenbank-Engine-Abfrage** undversuchenSie eine Verbindung via SQL Authentifizierung herzustellen.   
     Ich bin nicht Nostradamus, aber lassen Sie mich raten: Es klappt nicht!.. Warum?
  3. Probieren wir es ein weiteres Mal. Nur dieses Mal wählen Sie im Dialog   
     **Optionen🡪Verbindungseigenschaften** und geben bei **Verbindung mit Datenbank** herstellen die ***ContDB*** an.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung  
Besser? 😉

1. Monitoring des (nicht nur) SQL Clusters und der AVGs

SQL Server Cluster wird mit Hilfe der ressource.dll überwacht. Genauer gesagt, erledigt das die Prozedur *sp\_server\_diagnostics*. Die Intervalle werden auf des HealthChecks festgelegt - und diesen werden wir nun manuell festlegen:

1. Geben Sie im SSMS einer Instanz folgende Abfrage ein:

ALTER SERVER CONFIGURATION

SET FAILOVER CLUSTER PROPERTY HealthCheckTimeout = 15000;

1. Um mehr Einblick zu bekommen, aus welchen Gründen ein SQL Server einen Failover auslöst, ist es ratsam die Messergebnisse der sp\_server\_diagnostics genauer anzusehen und eine Aufzeichnung der Messergebnisse zu organisieren. Dazu stehen uns zwei Methoden zur Verfügung. Gute alte Handarbeit oder automatisiert mit Extended Events  
   1. XEvents  
      Natürlich lässt sich folgendes auch im SSMS nachstellen. Wir werden das später nachholen.
   2. Legen Sie einen Ordner auf dem Laufwerk C: mit dem Namen ***\_Log*** an
   3. Führen Sie daher folgendes Skript dazu aus.

CREATE EVENT SESSION [CLdiag]

ON SERVER

ADD EVENT [sp\_server\_diagnostics\_component\_result] (set collect\_data=1)

ADD TARGET [event\_file] (set filename='C:\\_Log\CLdiag.xel');

GO

ALTER EVENT SESSION [CLdiag]

ON SERVER STATE = start;

GO

* 1. Gehen Sie in das Verzeichnis C:\\_Logs und suchen sie an einer Datei mit der Endung .xel. Kopieren sie den Dateinamen und verwenden diesen in der nachfolgenden Aufgabe e.
  2. Führen Sie folgendes Skript aus. Korrigieren Sie vorher aber den Dateinamen.

SELECT \*

,CONVERT(xml,event\_data) AS 'xml\_data'

FROM sys.fn\_xe\_file\_target\_read\_file('C:\\_log\CLdiag\_0\_132617556098060000.xel',NULL,NULL,NULL)

* 1. Gehen Sie nun zum SSMS und dort zu **Verwaltung🡪Erweiterte Ereignisse.**Hier sollte eine Sitzung mit dem Namen Cldiag zu finden sein. Öffnen sie diese.
  2. Gehen Sie nun in den Bereich Ereignisse. In der Liste der nicht ausgewählten Ereignisse können Sie im Kanal filtern. Standardmäßig wird aber der Kanal Debug, den wir jetzt benötigen, ausgeblendet. Aktivieren Sie folglich in Kanal den Punkt Debug, in dem Sie auf den Kanal klicken und dort Debug anhaken.
  3. Die Liste der Ereignisse wurde nun länger. Unter anderem finden wir nun auch ***sp\_server\_diagnostic\_result\_set***. Ergänzen Sie die Sitzung um dieses Ereignis und bestätigen dies mit OK.

Klicken Sie auf die Sitzung Cldiag und lassen Sie sich die Live Daten anzeigen.

Andreas Rauch, 01.09.2021

# 