Always On ohne Active Directory

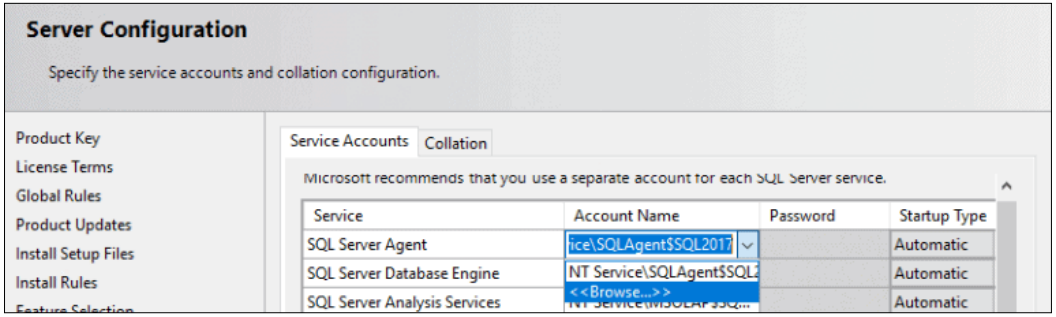
In diesem Artikel erklären wir Ihnen, wie Sie die Konfiguration einer SQL Server Always On Verfügbarkeitsgruppe in einem domainunabhängigen Failover Cluster vornehmen können.

Kategorie
SQL Server

Anna
Mussenbrock

Schritt 1:
Installation eines SQL Servers 2019 unter SQL Availability Group1 (SQLAG1) und SQL Availability Group2 (SQLAG2) mithilfe des integrierten Dienstkontos.

Zunächst installieren wir das SQL Server Datenbankmodul auf allen am Failovercluster beteiligten Schnittstellen. Da wir keine Active Directory konfiguriert haben, ist es hilfreich das integrierte Dienstkonto dafür zu verwenden. Hier haben wir dem Datenbankmodul-Dienst die Berechtigung für die Volumenwartung erteilt.



Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Schritt 2:
Aktivieren Sie die Always On Availability Gruppe (SQLAG1 + SQLAG2)

Im nächsten Schritt öffnen wir den SQL-Server Konfigurationsmanager in SQLAG1 und aktivieren mit dem Anklicken des Kontrollkästchens die Always On Availability Gruppe. Damit die Änderungen wirksam werden, müssen Sie den SQL Dienst neu starten. Das Gleiche führen Sie nun auch auf der SQLAG2 durch. Hier können wir zudem den Namen des domainunabhängigen Failover Clusters sehen.

Schritt 3:
Erstellen eines Datenbank Masterkeys auf der primären Replikation der SQLAG1.

Für die Replikation der SQLAG1 sollte ein verschlüsselter Datenbank Masterkey erstellt werden. Diesen schützen wir mit dem nachfolgenden Befehl und einem selbst festgelegten Passwort:

```
USE [master]
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'P@ssw0rd123!';
```

Wir haben zur Veranschaulichung des Beispiels das Passwort ‚P@assw0rd123!' verwendet.

Schritt 4:
Erstellen Sie ein Zertifikat, um die Endpunkte in Always On zu verschlüsseln.

Um den Endpunkt unserer Availability Gruppe zu sichern, erstellen wir nun ein Sicherheitszertifikat. Dieses dient zusätzlich zur Authentifizierung in einer domainunabhängigen Umgebung. Um das Zertifikat zu erstellen, führen wir dafür folgenden Befehl auf der primären Replikation der SQLAG1 aus:

```
USE [master]
GO
CREATE CERTIFICATE SQLAG1_Certificate_private
WITH SUBJECT = 'Certificate for the Domainless_SQL_AG SQL Availability Group'
GO
```

Schritt 5:
Sichern des Zertifikats

Da wir in späteren Schritten die Benutzer auf der sekundären Replikation des SQLAG2 authentifizieren müssen, machen wir nun ein Backup des bereits erstellten Zertifikat aus Schritt 4. Dafür führen wir folgenden Befehl aus:

```
BACKUP CERTIFICATION SQLAG1_Certificate_private
TO FILE = 'c\temp\SQLAG1_Certificate_private.cert'
GO
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Schritt 6:
Erstellen eines Endpunkts für die AG-Kommunikation

Wir erstellen nun mit folgendem Befehl ‚**create endpoint'** einen Endpunkt auf dem primären Replikat der SQLAG1 und erhalten dann folgende Informationen:

- Für die Kommunikation verwendet das SQLAG1 den Standardport 5022
- Das in Schritt 4 verwendete Zertifikat wird zur Authentifizierung verwendet
- Der **AES**-Verschlüsselungsalgorithmus wird verwendet

```
CREATE ENDPOINT SQLAG_Endpoint
STATE = STARTED
AS TCP
(
LISTENER_PORT = 5022
)
FOR DATABASE_MIRRORING
(
AUTHENTICATION = CERTIFICATE SQLAG1_Certificate_private,
ROLE = ALL,
ENCRYPTION = REQUIRED ALGORITHM AES
)
```

Schritt 7:
Wiederholen Sie die Schritte 3 bis 6 auf dem zweiten Knoten

Nun führen wir die Schritte 3-6 auf dem sekundären Replikat SQLAG2 mit t-SQL durch. Um nicht erneut alle Befehle eingeben zu müssen, kopieren wir einfach das bereits ausgeführte Skript wie folgt:

```
USE [master]
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'P@assw0rd123!';
GO
USE [master]
GO
CREATE CERTIFICATE SQLAG2_Certificate_private
WITH SUBJECT = 'Certificate for the Domainless_SQL_AG_SQL
Availability Group'
GO

BACKUP CERTIFICATE SQLAG2_Certificate_private
TO FILE = 'c\temp\SQLAG2_Certificate_private.cert'
GO
CREATE = ENDPOINT SQLAG_Endpoint
STATE = STARTED
AS TCP
(
LISTENER_PORT = 5022
)
FOR DATABASE_MIRRORING
(
AUTHENTICATION = CERTIFICATE SQLAG2_Certificate_private,
ROLL = ALL,
ENCRYPTION = REQUIRED ALGORITHM AES
)
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Schritt 8:
Erstellen eines SQL-Login auf dem SQLNode1

Wir erstellen nun in der Master-Datenbank eine SQL-Anmeldung und einen Benutzer. Der Benutzer wird zur Autorisierung des öffentlichen Schlüssels des SQLAG2 Zertifikats verwendet. Folgende Befehle führen wir aus:

```
CREATE LOGIN SQLAG2Login WITH PASSWORD = 'P@assw0rd123!'
GO
CREATE USER SQLAG2User FOR LOGIN SQLAG2Login
GO
```

Schritt 9:
Importieren des öffentlichen Zertifikatsschlüssels vom SQLAG2 Knoten

Wir kopieren nun das Sicherheitszertifikat vom SQLAG1 Knoten in den SQLAG2 Knoten. Im Anschluss erstellen wir im SQLAG1 ein weiteres Zertifikat und autorisieren in diesem Zuge den SQLAG2 User, um darauf zugreifen zu können.

Sollten Sie mehrere sekundäre Replikationen in Ihrer Umgebung haben, müssen Sie die Zertifikate für alle Weiteren ebenfalls erstellen.

```
CREATE CERTIFICATE SQLAG1_public_cert
AUTHORIZATION SQLAG2User
FROM FILE = 'c\temp\SQLAG2 Certificate\SQLAG2_Certificate_private.cert'
GO
```

Schritt 10:
Erteilen einer Berechtigung zur Herstellung einer Verbindung zum Endpunkt für SQLAG2 Login

Der SQLAG2 Login sollte bestenfalls eine Berechtigung erteilt bekommen haben, eine Verbindung zum HADR-Endpunkt herstellen zu dürfen. Dies ermöglicht die Kommunikation zwischen primären und sekundären Replikationen. Mit dem nachfolgenden Befehl weisen wir die Berechtigung zu:

```
GRANT CONNECT ON ENDPOINT :[SQLAG_Endpoint] TO SQLAG2Login
GO
```

Schritt 11:
Wiederholen Sie die Schritte 8-10 auf dem SQLAG2 Knoten

Nun wiederholen wir die Schritte 8-10 auf dem sekundären SQLAG2 Knoten mit folgendem Skript:

```
CREATE LOGIN SQLAG1Login WITH PASSWORD = 'P@assw0rd123!'
GO

CREATE USER SQLAG1USer FOR LOGIN SQLAG1Login
GO

CREATE CERTIFICATE SQLAG2_public_cert
AUTHORIZATION = 'c\temp\SQLAG1 Certificate\SQLAG1_Certificate_private.cert'
GO

GRANT CONNECT ON ENDPOINT : [SQLAG_Endpoint] TO SQLAG1Login
GO
```

Mit diesem Schritt haben wir nun die Sicherheitskonfiguration für die Always On Availability Gruppe abgeschlossen und können folgend die AG-Gruppe bereitstellen.

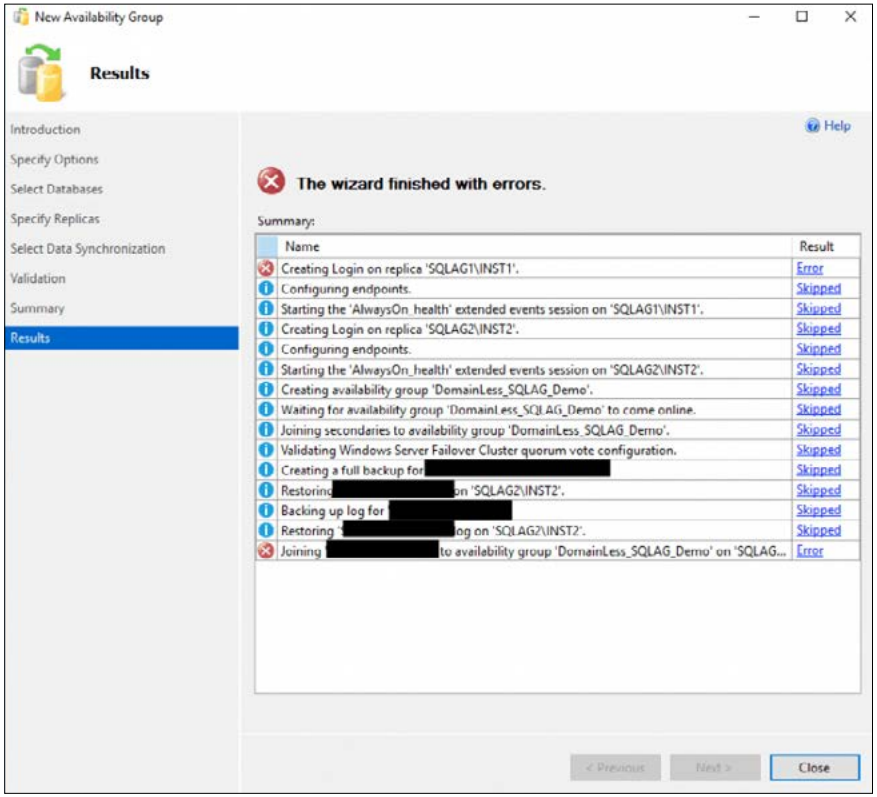
Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Dafür müssen einige Voraussetzungen erfüllt sein:

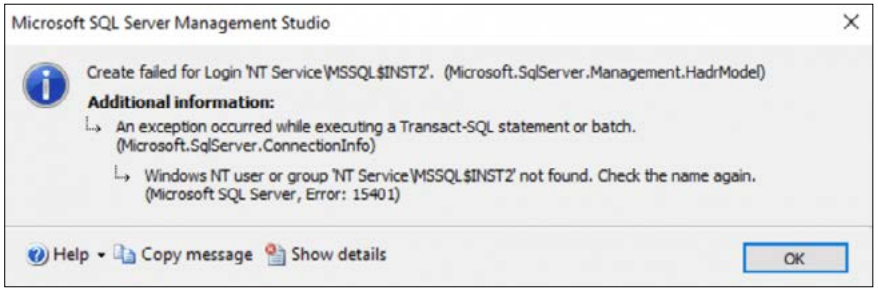
- Erstellen Sie eine Datenbank in der primären Availability Group Replikation
- Sie sollten in beiden AG-Knoten ähnliche Datafiles und Log Files verwenden
- Sichern Sie ihre primäre Replikation vollständig und führen Sie eine Transaktionsprotokollsicherung durch.
- Kopieren Sie die Sicherungsdateien auf den sekundäre Replikationsknoten
- Stellen Sie die Sicherungsdateien auf der sekundären Replikation im NORECOVERY-Modus wieder her.

Sie werden sehen, wenn Sie eine neue Verfügbarkeitsgruppe in SQL Server Management Studio (SSMS) erstellen, dass das integrierte Dienstkonto für die Endpunktbenutzer verwendet wird. Im Normalfall verwendet man eine einer Availability Group ein Active Directory Dienstkonto für die Endpunktautorisierung. Diese funktioniert allerdings nicht in einer domainunabhängigen Umgebung.

Wenn Sie nun versuchen, die domainunabhängige AG mithilfe des SSMS Assistenten zu konfigurieren, sehen Sie folgende Fehlermeldung:



Klicken Sie nun den Hyperlink um in das Fehlerprotokoll zu gelangen, sehen Sie, dass das integrierte Dienstkonto ‚**NT Service \ MSSQL $ INST2**' nicht gefunden werden kann.



Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Um den Endpunktuser als SQL-User im Skript zu ändern, führen Sie in der primären und sekundären Replikation folgende Befehle durch:

× SQLAG1 (primäre Replikation)

```
:CONNECT SQLAG1\INST1

IF (SELECT satat FROM sys.endpoints WHERE name = N'SQLAG_Endpoint')<> 0
BEGIN
    ALTER ENDPOINT [SQLAG_ENDPOINT] STATE = STARTED
END

GO
USE [master]
GO
GRANT CONNECT ON ENDPOINT : [SQLAG_Endpoint] TO SQLAG2LOGIN
GO
```

× SQLAG2 (sekundäre Replikation)

```
:CONNECT SQLAG1\INST2

IF (SELECT satat FROM sys.endpoints WHERE name = N'SQLAG_Endpoint')<> 0
BEGIN
    ALTER ENDPOINT [SQLAG_ENDPOINT] STATE = STARTED
END

GO
USE [master]
GO
GRANT CONNECT ON ENDPOINT : [SQLAG_Endpoint] TO SQLAG1LOGIN
GO
```

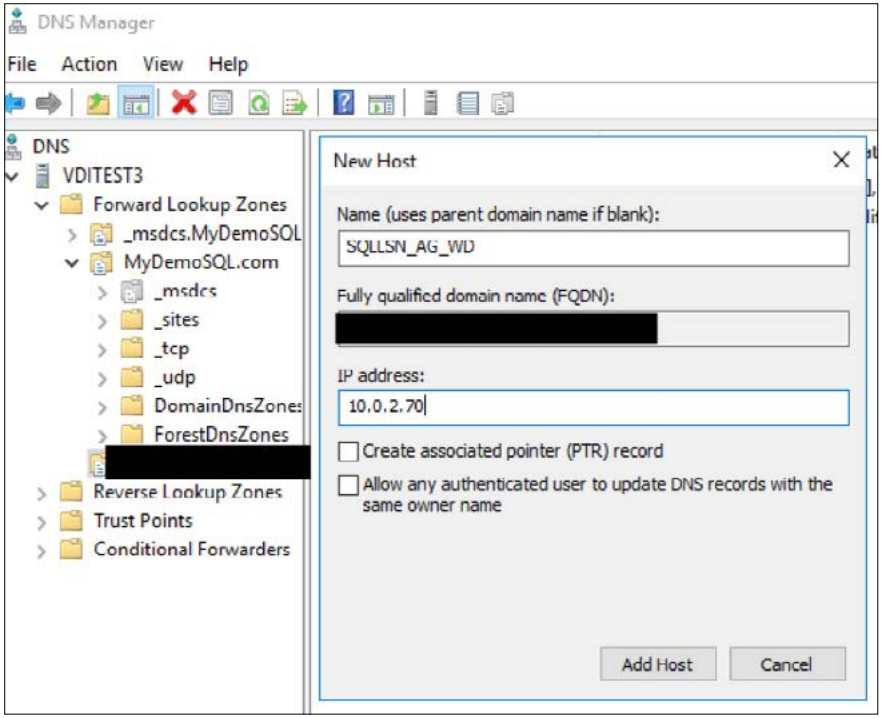
Führen wir anschließend nun das AG Skript im SQLCMD-Modus aus, sehen wir, dass die domainunabhängige Availability Group nun verfügbar ist.

Im nächsten und letzten Schritt zeigen wir Ihnen, wie man einen SQL-Listener für die domainunabhängige SQL Server Always On AG Gruppe erstellt.

Dafür erweitern wir die Verfügbarkeitsgruppe in SSMS auf der primären Replikation und erstellen einen neuen Listener, in dem wir einen DNS-Namen, einen Port und eine statische IP-Adresse eingeben. Sobald die Listener-Konfiguration abgeschlossen ist und Sie versuchen, eine Verbindung in SSMS herzustellen, kann möglicherweise ein netzwerkbezogener Fehler angezeigt werden.

Sobald wir einen herkömmlichen AG konfiguriert haben, erstellen wir ein Computerobjekt in der Active Directory. Im domainunabhängigen Failover Cluster kann der Listener in der AD nicht erstellt werden.

Um dieses Problem zu beheben, stellen wir eine Verbindung zum DNS Manager her und erstellen eine neue Forward-Lookupzone und einen neuen Hostdatensatz für die SQL-Listener.



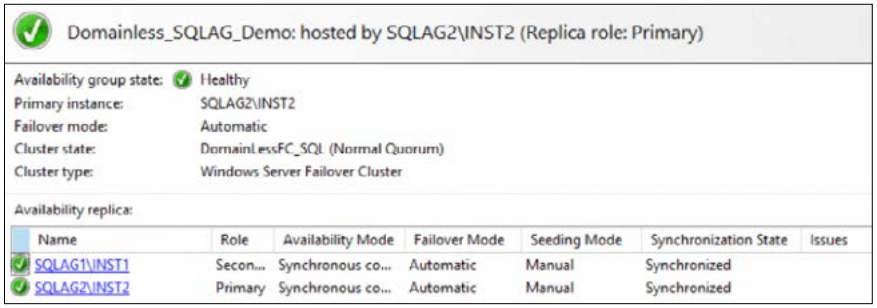
Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Sobald der Host-Datensatz erstellt wurde, sollten wir nun eine Ping-Antwort für den Listener erhalten, der die dazugehörige IP-Adresse zurückgibt.

```
$ ping SQLLSN_AG_WD
```

Jetzt kann auch mit dem Listener eine Verbindung zur domainunabhängigen Availability Group hergestellt werden.

Schlussendlich ist es immer hilfreich, das AG-Failover zu überprüfen, sobald Konfigurationen durchgeführt wurden. Dies können Sie im Failover-Assistent vornehmen. Stellen Sie nach Abschluss des Failover eine Verbindung zur primären Replikation SQLAG2 \ INST2 her und überprüfen Sie den Dashboard-Zustand.



alwayson, domain, domainunabhängig, failovercluster, sql, sqlserver

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612