

SQL Server AlwaysOn Availability Groups einrichten

von Holger Voges



© 2015 by Holger Voges, Netz-Weise IT Training

Version 1.0

Freundallee 13 a
30173 Hannover
www.netz-weise.de

Inhalt

Grundlagen von SQL-Server AlwaysOn	4
Einrichten von AlwaysOn.....	5
Einrichten des Windows Failover-Cluster-Features	5
Konfigurieren des SQL-Server Dienstes für die Nutzung von AlwaysOn	10
Einrichten der Availability Group	11
Wissenwertes zu Availability Groups	18
Hinzufügen einer Datenbank zu einer Availability Group	19
Read Only Routing konfigurieren	21
Überwachung der AlwaysOn Availability Group	23
System-Views für AlwaysOn.....	25
Weiterführende Links.....	26
Anhang A	27
Anhang B	33
Über den Autor.....	37

Grundlagen von SQL-Server AlwaysOn

SQL-Server AlwaysOn ist der Oberbegriff für zwei Hochverfügbarkeitslösungen im Microsoft SQL-Server Umfeld. Microsoft unterscheidet dabei 2 Typen von AlwaysOn Lösungen.

AlwaysOn Failover Cluster (FCI)

Der AlwaysOn Failover Cluster ist nur eine Bezeichnung für SQL-Server auf Basis der Microsoft Failover-Cluster Technologie. Mit SQL-Server haben sich hier bis auf die Namensänderung keine wesentlichen Neuerungen ergeben.

AlwaysOn Availability Groups (AG)

AlwaysOn Availability Groups sind eine in SQL-Server 2012 komplett neu eingeführte Technologie, die die Fähigkeiten des klassischen SQL-Server Mirrorings mit der Failover-Cluster-Technologie erweitert. Neuerungen sind dabei z.B. lesende Replikas und die Möglichkeit, mit einem Computernamen auf alle SQL-Server Datenbankspiegel zuzugreifen. Dadurch muss für einen Failover zwischen den am Spiegel beteiligten Instanzen die Clientsoftware nicht mehr umgestellt werden. Außerdem können bei SQL-Server 2012 bis zu 4 Replikas einer Datenbank erstellt werden, bei SQL-Server 2012 R2 sogar 8.

Der Hauptnachteil von AlwaysOn ist, dass die Technologie sowohl in SQL-Server 2012 wie auch in SQL-Server 2012 R2 die Enterprise Edition benötigt. Für die Standard-Edition steht also nach wie vor nur die klassische Spiegelung sowie Failover-Clustering zur Verfügung.

Funktionsweise und Grundbegriffe

AlwaysOn Availability Groups sind eine Mischung aus den beiden eingeführten Technologien Failover Clustering und Datenbankspiegelung. Für die Replikation der Daten wird dabei auf die klassische Datenbankspiegelung zurückgegriffen. Im Gegensatz zur Spiegelung stellt AlwaysOn aber auch lesende Replikas zur Verfügung, also Kopien der Datenbank, die online sind und Lesezugriffe zulassen. Für den Failover im Fehlerfall und die Steuerung des Clientzugriffs nutzt die Availability Group den Cluster-Dienst, der ein zuverlässiges System bereitstellt, um die Verfügbarkeit eines Servers zu prüfen und im Fehlerfall einen automatischen Failover zu veranlassen. Der Cluster stellt außerdem einen alternativen Computernamen für den Clientzugriff zur Verfügung. Greift der Client auf den Clusternamen zu, wird er automatisch auf den jeweils aktiven Server weitergeleitet. Dadurch ist, anders als bei der Datenbankspiegelung, keine weitere Witness-Instanz notwendig, und der Client muss auch nichts von der Spiegelung wissen.

Die Server der Availability Group können dabei auf 2 Arten Ihre Daten synchronisieren: synchron und asynchron. Bei der synchronen Spiegelung werden Datenänderungen (Transaktionen) vom Prinzipal, also der einzigen schreibenden Replika der Daten, immer erst dann als abgeschlossen übernommen, wenn der Spiegel den Erhalt der Daten bestätigt hat. Dadurch ist ein Datenverlust ausgeschlossen und Prinzipal und Spiegel sind immer synchron. Bei der asynchronen Spiegelung werden die Daten immer so übertragen, wie der Spiegel sie annehmen kann. Der Prinzipal kümmert sich nicht um den Stand der Daten auf dem Spiegel. Dadurch ist Datenverlust möglich. Bei AlwaysOn können gleichzeitig sowohl synchrone wie auch asynchrone Spiegel existieren.

Einrichten von AlwaysOn

Voraussetzungen

- Windows Server mit Failover-Cluster Funktionalität (bis Windows Server 2008R2 mind. Enterprise Edition, ab Server 2012 reicht die Standard-Edition)
- SQL-Server 2012 / 2014 Enterprise Edition.
- Eine Windows Domäne

Im folgenden Beispiel wird eine Availability Group mit 3 Servern eingerichtet, wobei 1 Server als Prinzipal arbeitet, 1 Server eine lesende Replika zur Verfügung stellt, und 1 Server eine nicht-lesende Replika. Da der Server nicht wie ein klassischer Failovercluster eingerichtet wird, kann der SQL-Server bereits installiert sein. Wir gehen in diesem Szenario davon aus, dass alle 3 Server bereits einen SQL-Server 2012 (Enterprise-Edition oder Developer) installiert haben.

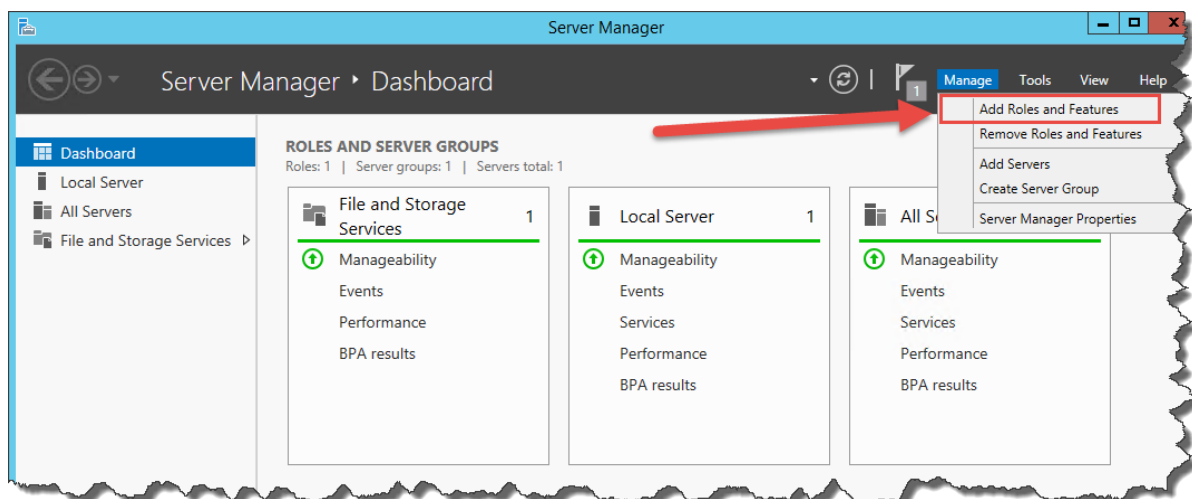
Die Umgebung

Die Testumgebung umfasst 4 Server – einen Domänencontroller (Domäne: Contoso.com) sowie 3 Server mit Windows Server 2012 R2, die Mitglieder der Domäne sind und auf denen bereits SQL-Server 2012 installiert ist.

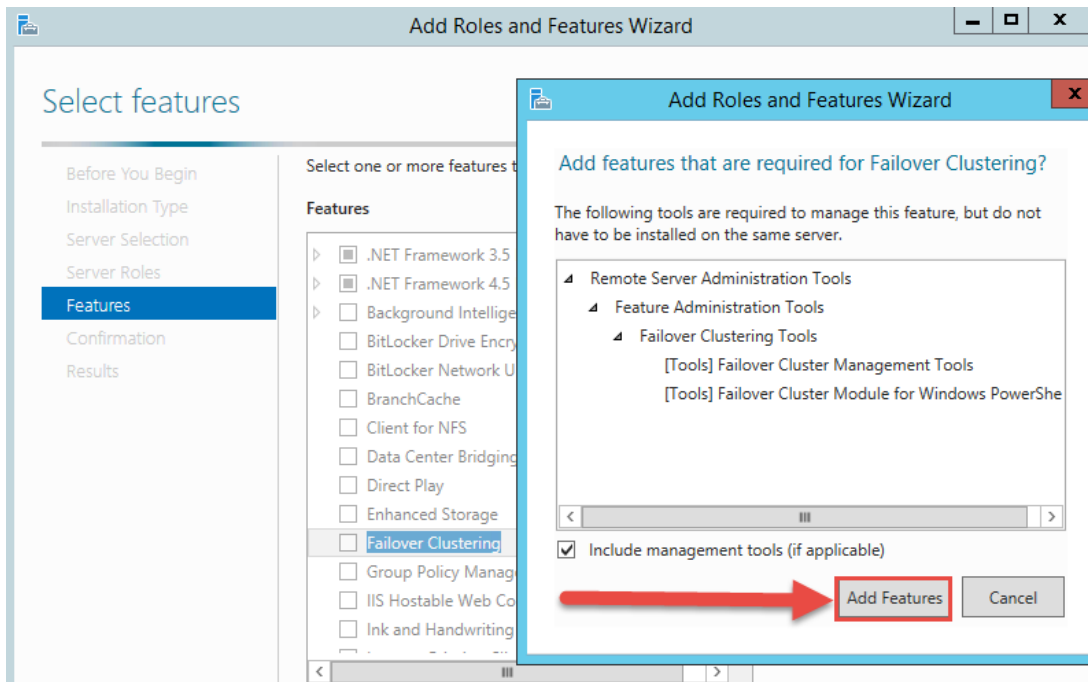
Einrichten des Windows Failover-Cluster-Features

AlwaysOn nutzt den Windows Failovercluster für den automatischen Failover und den Client-Zugriff. Daher muss auf allen Servern, die die Availability Group zur Verfügung stellen sollen, das Failover Cluster Feature installiert sein. Das folgende Beispiel zeigt die Installation auf einem Server.

Starten Sie zuerst den Windows Server Manager oder nutzen Sie die Windows Powershell, um das Failover-Cluster-Feature auf allen Servern zu installieren. Installieren Sie über „Add Roles and Features“ das Failover-Cluster Feature:



Klicken Sie sich durch den Assistenten bis zum Eintrag „Features“ und wählen Sie „Failover Clustering“ aus.

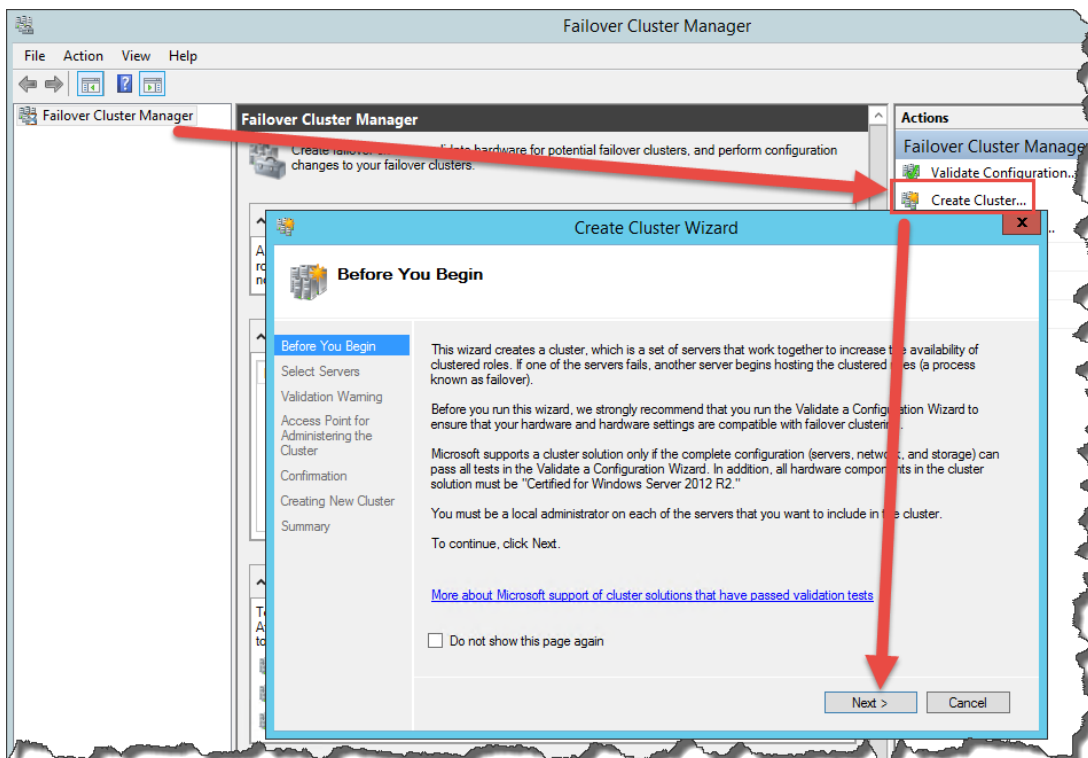


Beenden Sie den Assistenten, indem Sie „Next“ und „Install“ auswählen.

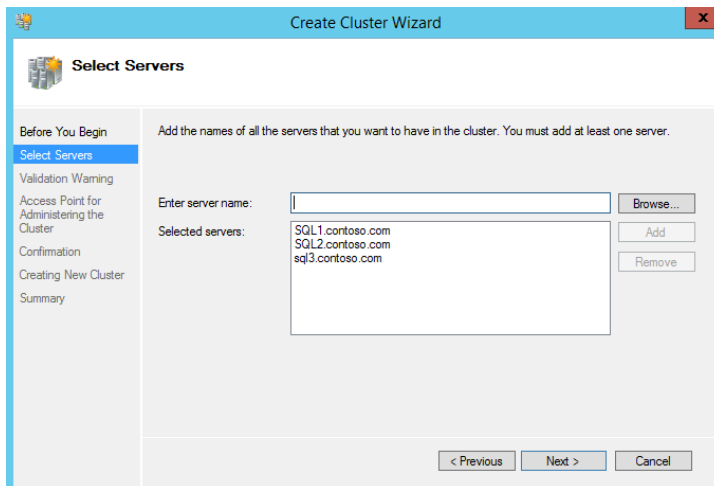
Alternativ können Sie auch Windows Powershell verwenden. Starten Sie dazu eine Powershell-Konsole mit administrativen Rechten und geben Sie folgendes Kommando ein:

```
Install-WindowsFeature -Name "Failover-Clustering" -IncludeAllSubFeature
```

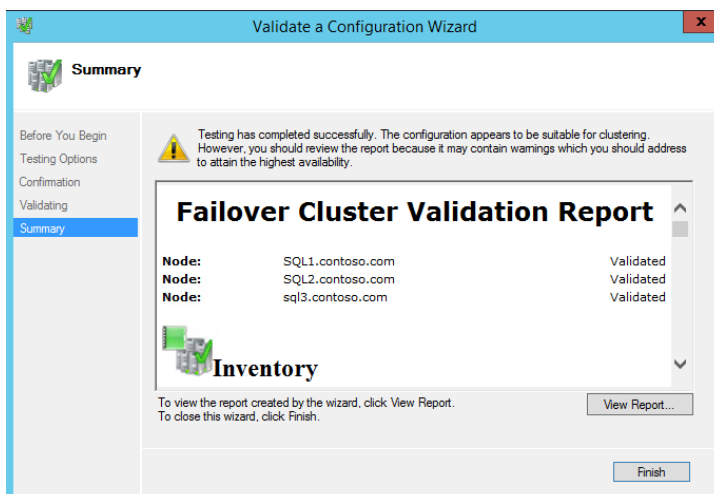
Starten Sie nun auf einem der Server den Windows Failover-Manager und richten Sie einen neuen Failover-Cluster ein. Wählen Sie hierzu aus dem Actions-Menü rechts die Option „Create Cluster“.



Zum Einrichten des Clusters wählen Sie zuerst im Menüpunkt Select Servers die Server aus, die Teil des Clusters werden sollen.



Seit Windows Server 2008 stellt ein Validierungstool sicher, dass der Cluster korrekt konfiguriert ist. Lassen Sie hier alle Tests durchlaufen. Dies stellt nicht nur sicher, dass alles richtig konfiguriert ist, sondern ist auch Voraussetzung dafür, dass der Cluster von Microsoft supported wird.



Der Test wurde mit Warnings beendet. Dies liegt daran, dass wir kein dezidiertes Netzwerk für die Cluster-Kommunikation verwendet haben. In einem produktiven Cluster sollte immer eine zweite Netzwerkkarte für die Cluster-Kommunikation verwendet werden, da der Cluster über den sogenannten Heartbeat, der über das Clusternetzwerk ausgetauscht wird, sicherstellt, dass die Clusterknoten online sind. In unserer Testumgebung haben wir darauf verzichtet. Dies stellt keine Einschränkung der Funktionalität dar, kann aber bei einem Netzwerkausfall oder hoher Netzwerklast zu Problemen führen. Den vollständigen Report können Sie über View Report einsehen.

Description: Validate that servers can communicate, with acceptable latency, on all networks.

Analyzing connectivity results ...

Node SQL2.contoso.com is reachable from Node SQL1.contoso.com by only one pair of network interfaces. It is possible that this network path is a single point of failure for communication within the cluster.

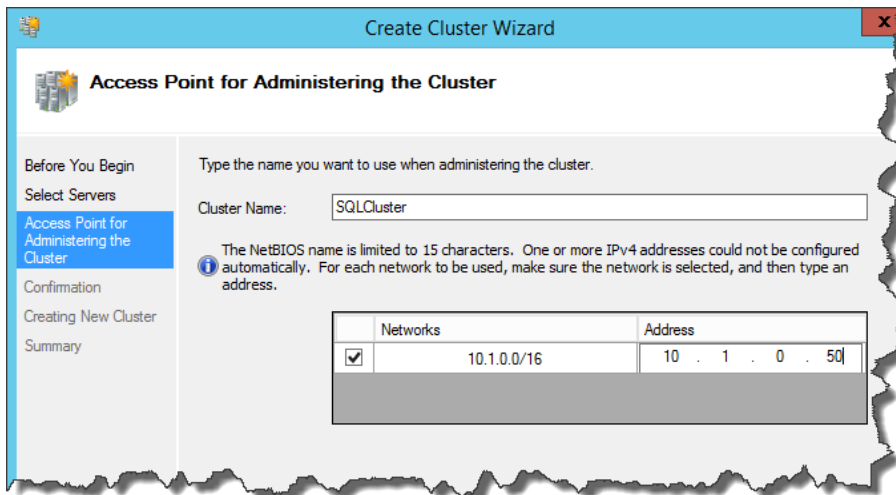
Following are the connectivity checks made using UDP on port 3343 from network interfaces on node SQL1.contoso.com to network interfaces on node SQL2.contoso.com

Result	Source Interface Name	Source IP Address	Destination Interface Name
Success	SQL1.contoso.com - Ethernet	10.1.150.201	SQL2.contoso.com - Ethernet

In unserem Netzwerk verfügen wir außerdem nicht über einen für alle Server erreichbaren Datenträger. Da wir einen Majority-Node Cluster einrichten (eine ungerade Anzahl von Servern) ist dies auch nicht notwendig. Beenden Sie die Validierung mit „Finish“.

Unter „Access Point for Administering the Cluster“ geben Sie einen Namen für den Cluster ein. Dieser Name wird zu Verwaltungs-Zwecken benötigt. Er entspricht nicht dem Namen, über den Sie später

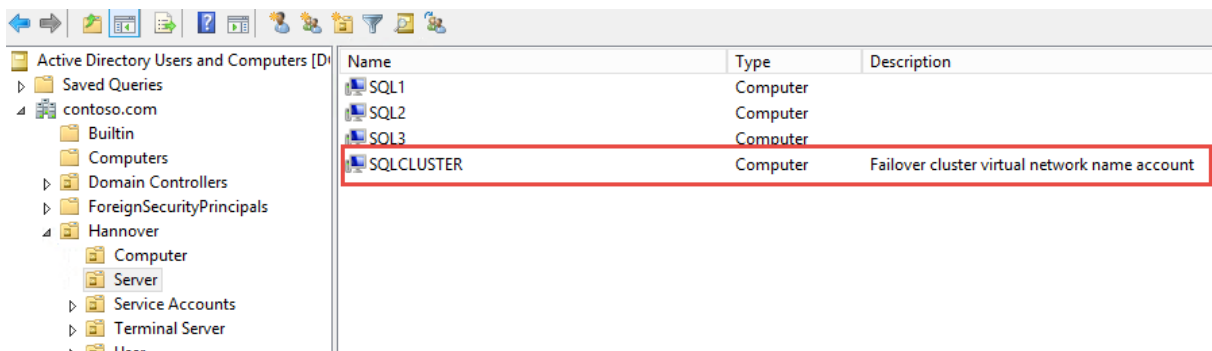
auf die AlwaysOn Availability Group zugreifen! Für diesen Namen wird außerdem ein Computerkonto im Active Directory eingerichtet. Außerdem benötigen Sie eine eindeutige IP-Adresse.



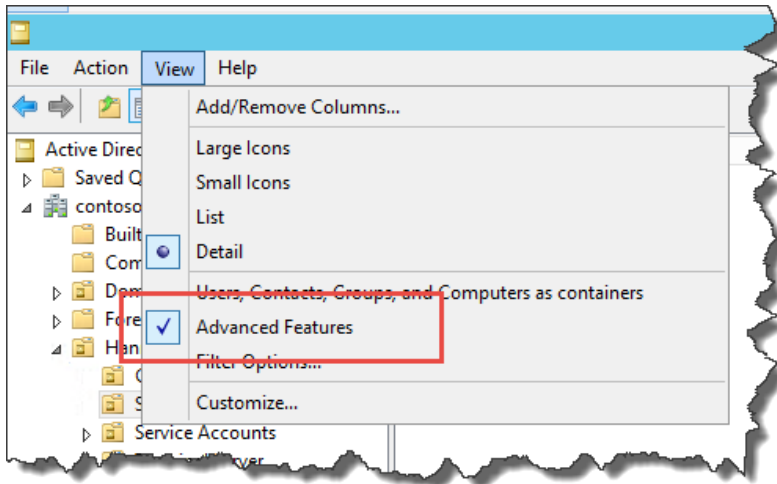
Klicken Sie auf Next, überprüfen Sie die Einstellungen, und klicken Sie ein weiteres Mal auf Next, um den Cluster einzurichten.

Nach der Erstellung des Clusters können Sie unter „Summary“ noch einmal die Konfiguration einsehen. Speichern Sie den Report für Ihre Dokumentation und beenden Sie die Cluster-Konfiguration.

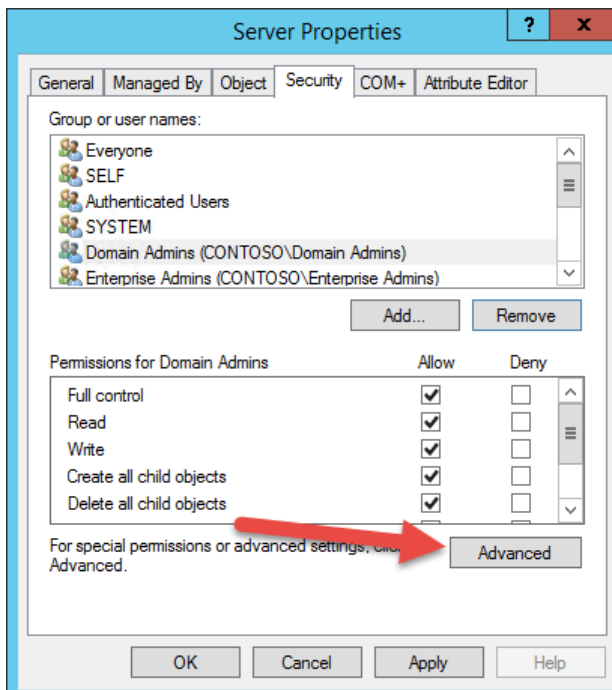
Mit der Erstellung des Clusters können Sie den Cluster nun über die IP-Adresse und den Cluster-Namen im Netzwerk erreichen. Außerdem wurde ein AD-Konto erstellt.



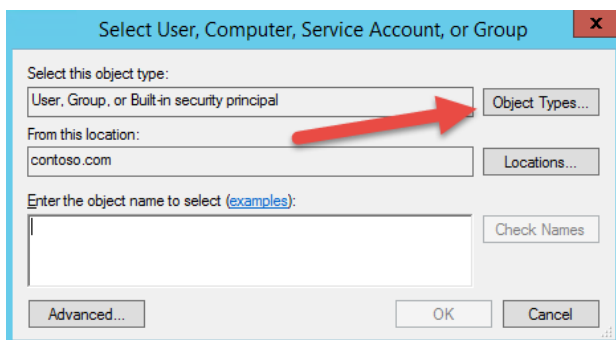
Um sicher zu stellen, dass der Cluster später für die Availability Groups weitere Computer-Konten erstellen kann, berechnen Sie den Cluster außerdem zum Anlegen neuer Computer-Konten. In unserem Beispiel tue ich das auf der Organizational Unit Server (in der sich das Computer-Konto des Clusters befindet). Sollten Ihr Cluster später verschoben werden ist es eventuell sinnvoll, die Berechtigungen direkt auf der Domäne zu vergeben. Starten Sie hierfür Active Directory Benutzer und Computer (z.B. auf einem Domänen-Controller), überprüfen Sie im Menü View, ob die erweiterte Ansicht aktiviert ist, und wählen Sie das Kontextmenü der OU:

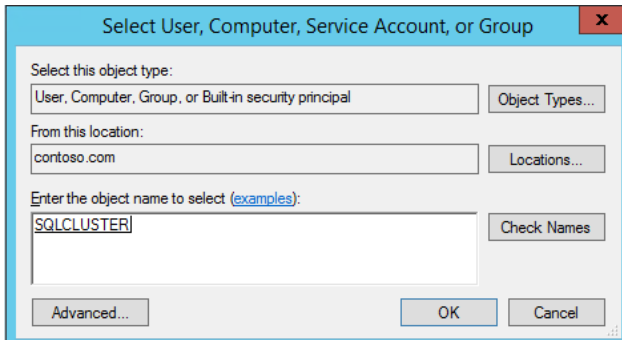
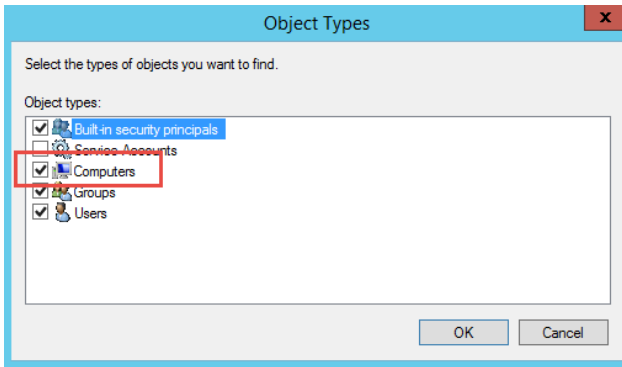


Wählen Sie den Reiter „Security“ aus und wählen Sie das erweiterte Menü:

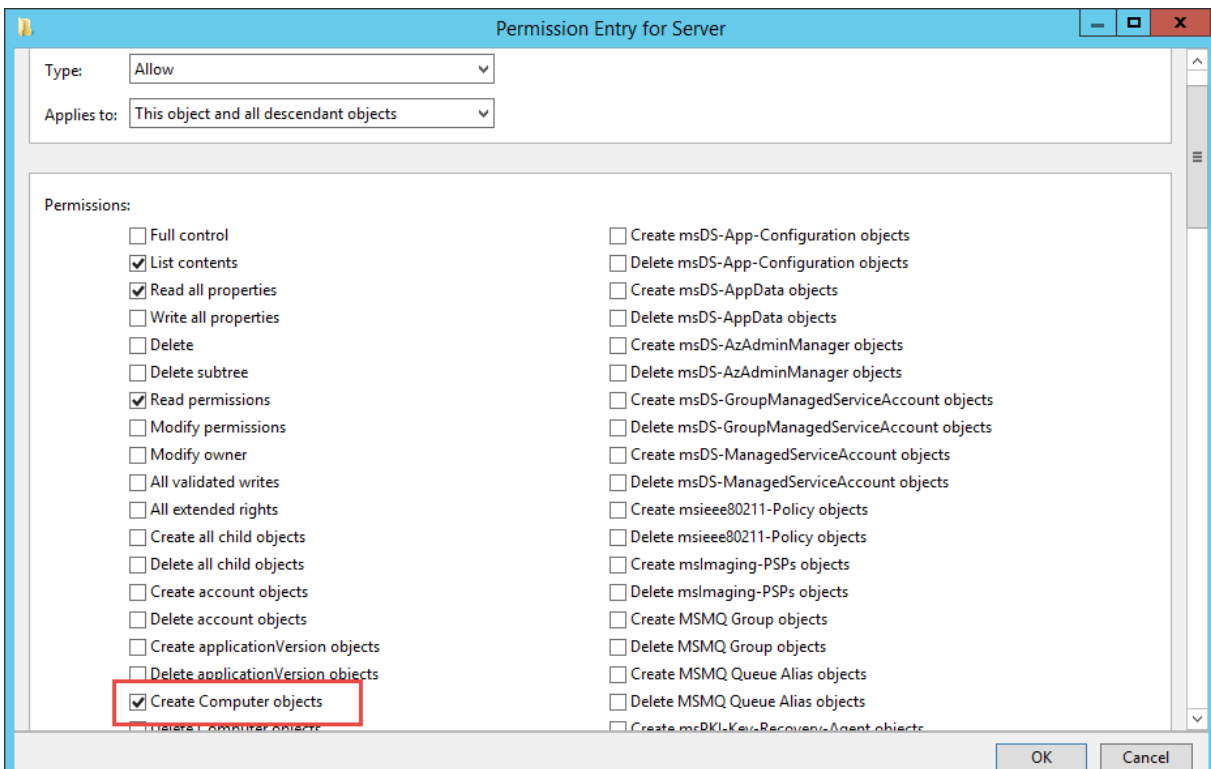


Fügen Sie nun das Computer-Konto des Clusters hinzu. Aktivieren Sie hierzu das Anzeigen von Computernamen unter Object-Types:





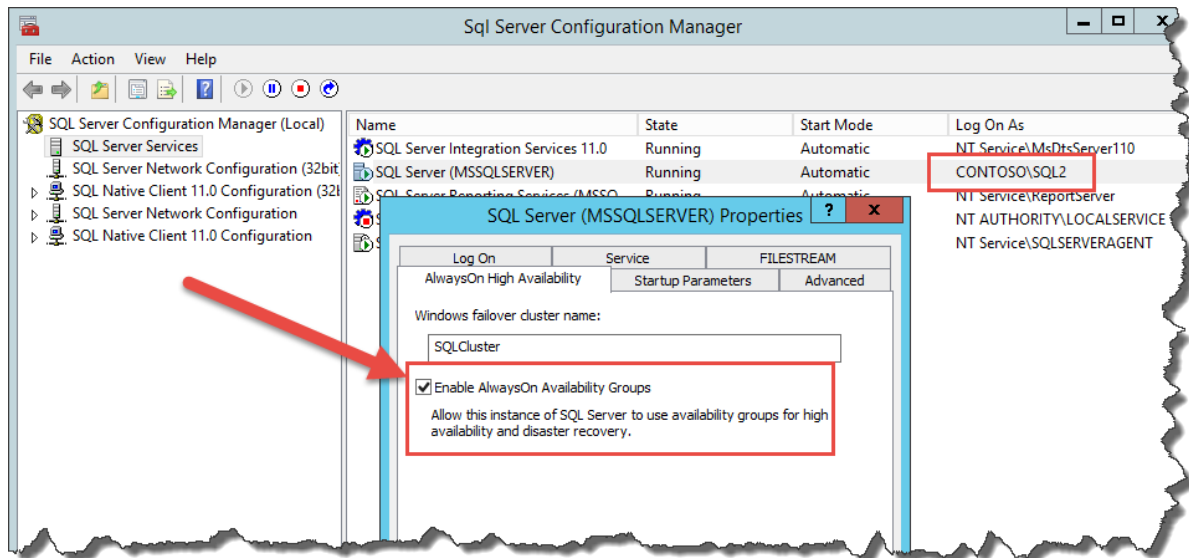
In der Berechtigungs-Liste vergeben Sie jetzt die Berechtigung „Create Computer objects“ und prüfen, ob das Konto auch Read-Permissions hat.



Konfigurieren des SQL-Server Dienstes für die Nutzung von AlwaysOn

Der SQL-Server-Dienst muss für die Nutzung von AlwaysOn aktiviert werden. Starten Sie hierzu den SQL-Server Configuration Manager und aktivieren Sie AlwaysOn, indem Sie unter „SQL Server

Services“ den Dienst auswählen, unter dem der SQL-Server läuft. Vergewissern Sie sich außerdem, dass der Dienst mit einem Domänen-Konto startet.



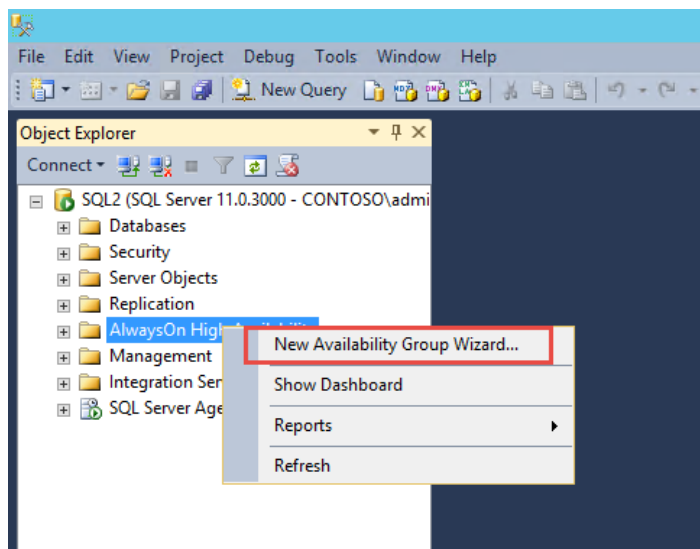
Auch dieses Feature können Sie alternativ über Windows Powershell aktivieren – per Powershell funktioniert das auch remote!

```
Enable-SqlAlwaysOn -Path sqlserver:\sql\sql13\default
```

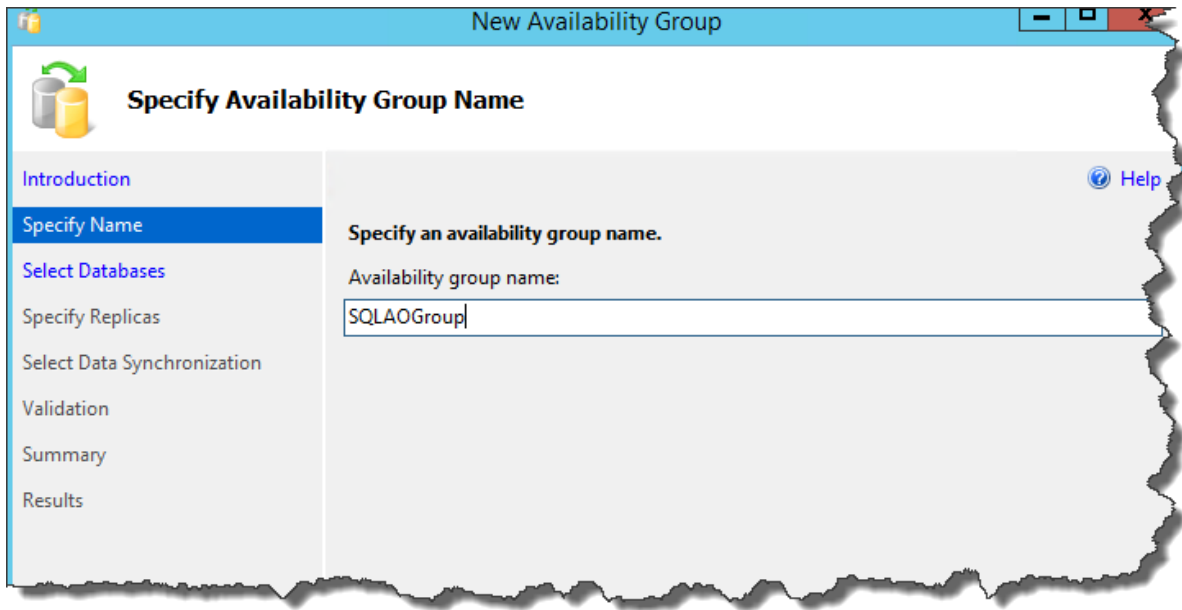
Achten Sie beim Pfad darauf, dass die letzten beiden Einträge den Servernamen und die Instanz darstellen. Die Konfiguration erfordert einen Dienstneustart!

Einrichten der Availability Group

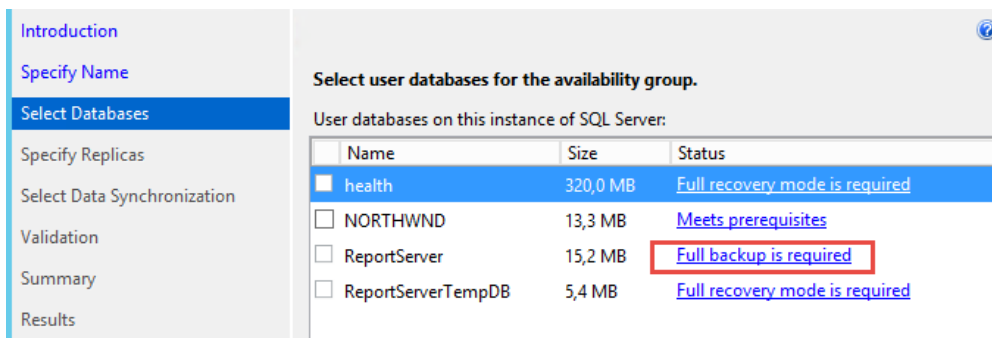
Die Availability Group wird über das SQL-Server Management Studio eingerichtet. Klicken Sie hierzu den Management-Knoten AlwaysOn High Availability im Objekt-Explorer mit der rechten Maustaste an und wählen Sie „New Availability Group“.



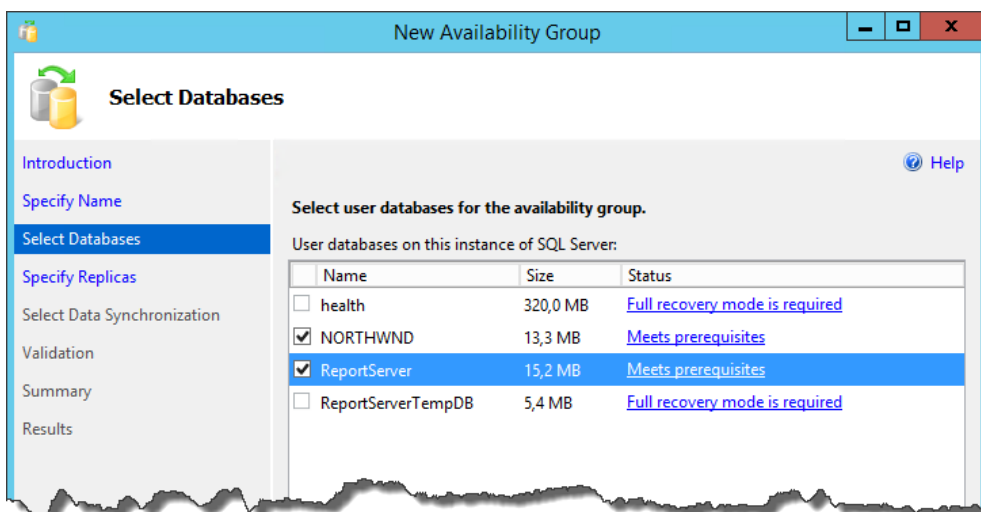
Der Assistent leitet Sie nun durch die Erstellung der Availability Group. Geben Sie zuerst einen Namen an. Dieser Name dient nur der Anzeige im Management-Studio. Der Name für den Clientzugriff wird später eingerichtet.



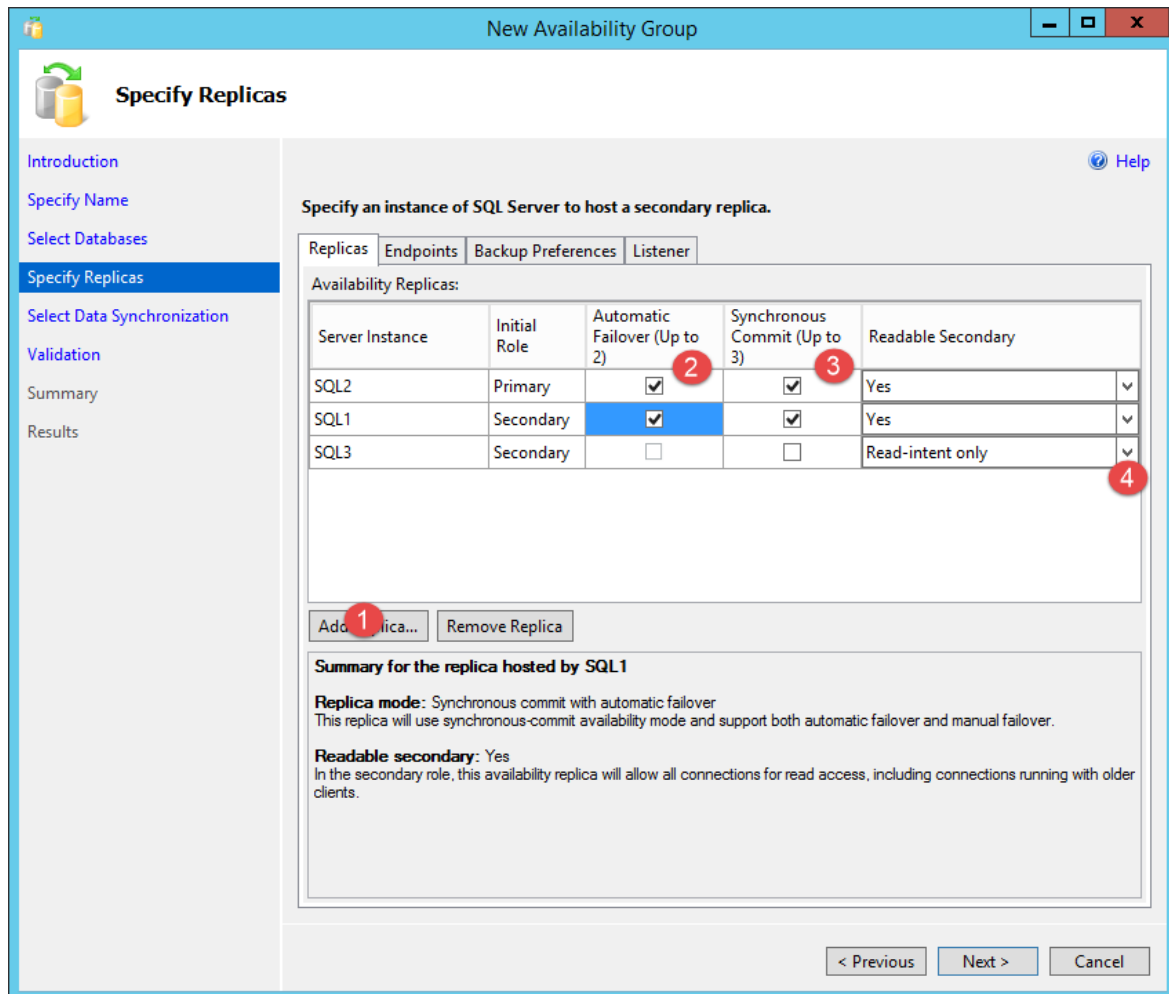
Eine Availability Group kann mehrere Datenbanken zu einer Failover-Gruppe zusammenfassen. Wählen Sie im folgenden Fenster die Datenbanken aus, die der Gruppe hinzugefügt werden sollen. Voraussetzung hierfür ist, dass von der Datenbank bereits einmal ein Full-Backup gemacht wurde und die Datenbank sich im Full Recovery-mode befindet. Der Assistent prüft die Voraussetzungen und informiert Sie über Probleme.



Wenn Sie die Probleme behoben haben, können Sie die Datenbanken erneut prüfen, indem Sie Refresh auswählen:

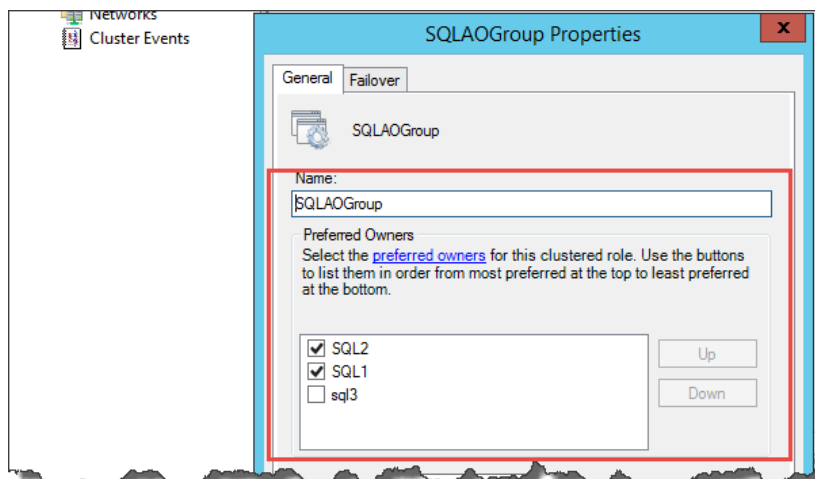


Unter Specify Replicas wird es interessant. Hier fügen Sie zuerst die vorher für AlwaysOn konfigurierten Server über „Add Replica“ (1) aus:



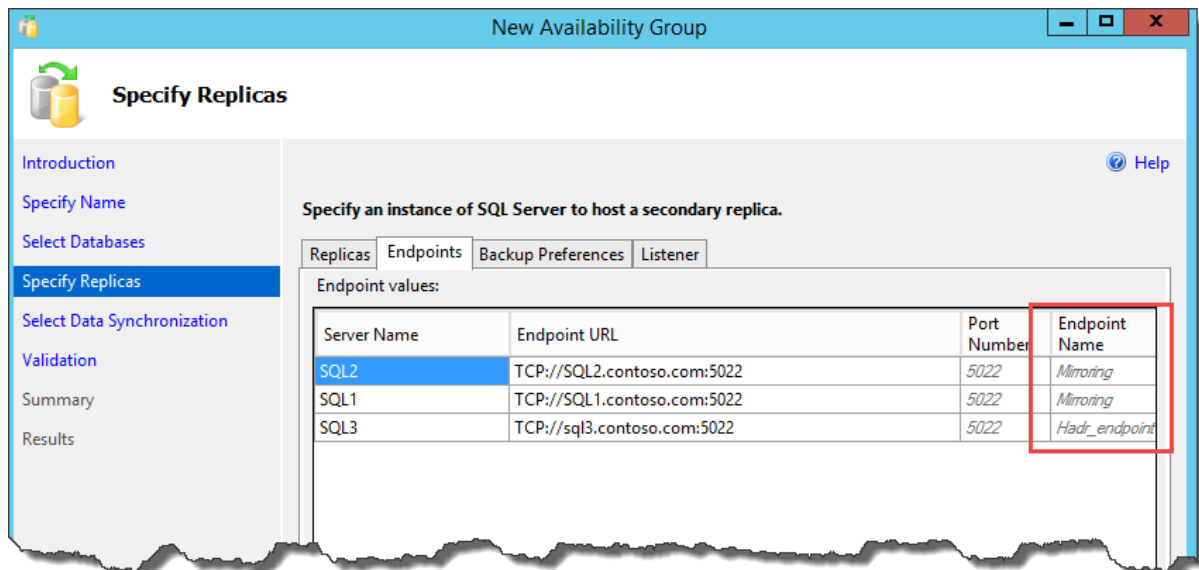
Die Spalte „Initial Role“ zeigt Ihnen an, welcher Server schreibend auf die Daten zugreifen kann. Die Primäre Rolle kann nur ein Server haben. Alle anderen Server sind Secondaries.

Über Automatic Failover (2) legen Sie fest, welche Server im Failover-Fall die Primäre Rolle übernehmen können. Der Automatic Failover legt im Cluster-Server fest, wer Besitzer der Resource werden kann:

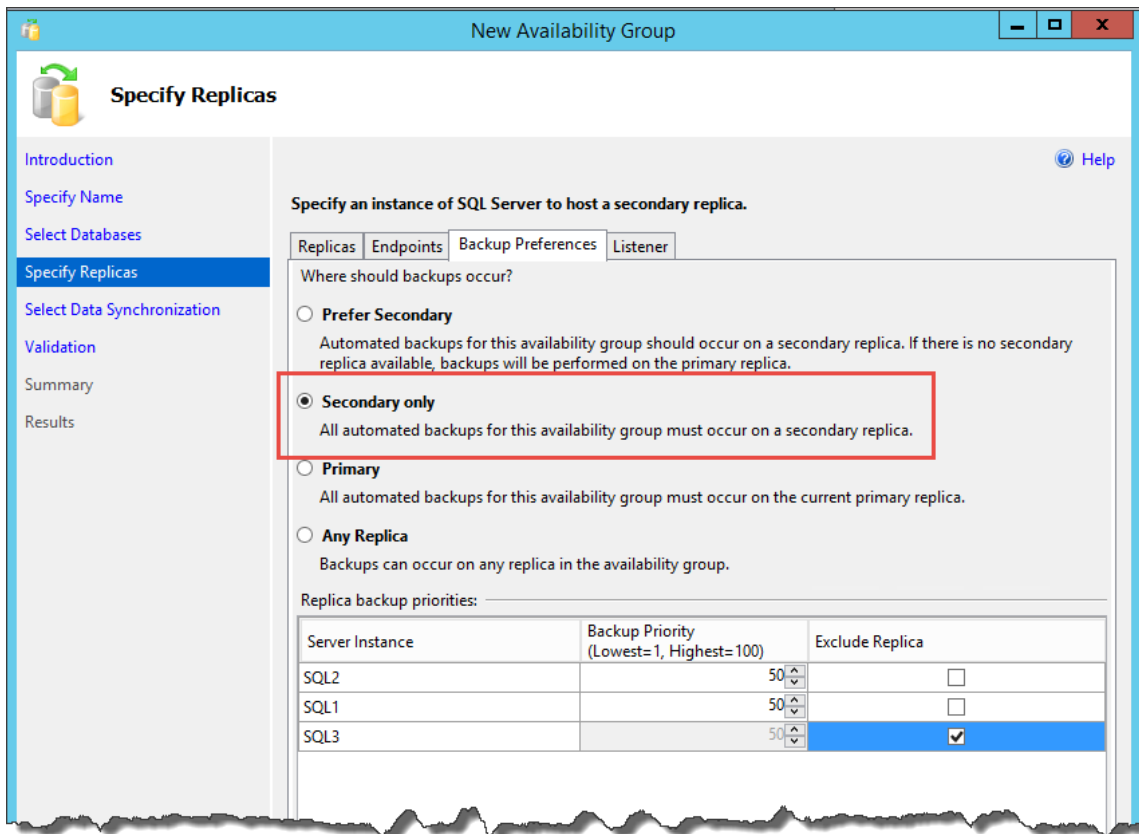


Synchronous Commit legt fest, ob Transaktionen auf dem Zielsystem bestätigt werden müssen, bevor Sie auf dem Primary committed werden können. Ein Synchroner Commit kann eventuell die Performance auf dem Primary negativ beeinflussen, stellt aber sicher, dass es nicht zu einem Datenverlust kommen kann. Ein automatischer Failover kann mit synchronen Partnern stattfinden. In der Spalte „Readable Secondary“ legen Sie fest, ob die Replika lesend verwendet werden kann, oder sich im Recovery-Mode befindet. Read-Only Intent bedeutet in diesem Fall, dass ein Client beim Verbindungsaufbau im Connection-String „Read Only Intent“ angibt. Diese Clients können über Routing-Regeln auf einen Server weitergeleitet werden, der eine „Read Only Intent“-Replika hält. [Mehr über Routing-Regeln erfahren Sie weiter unten.](#)

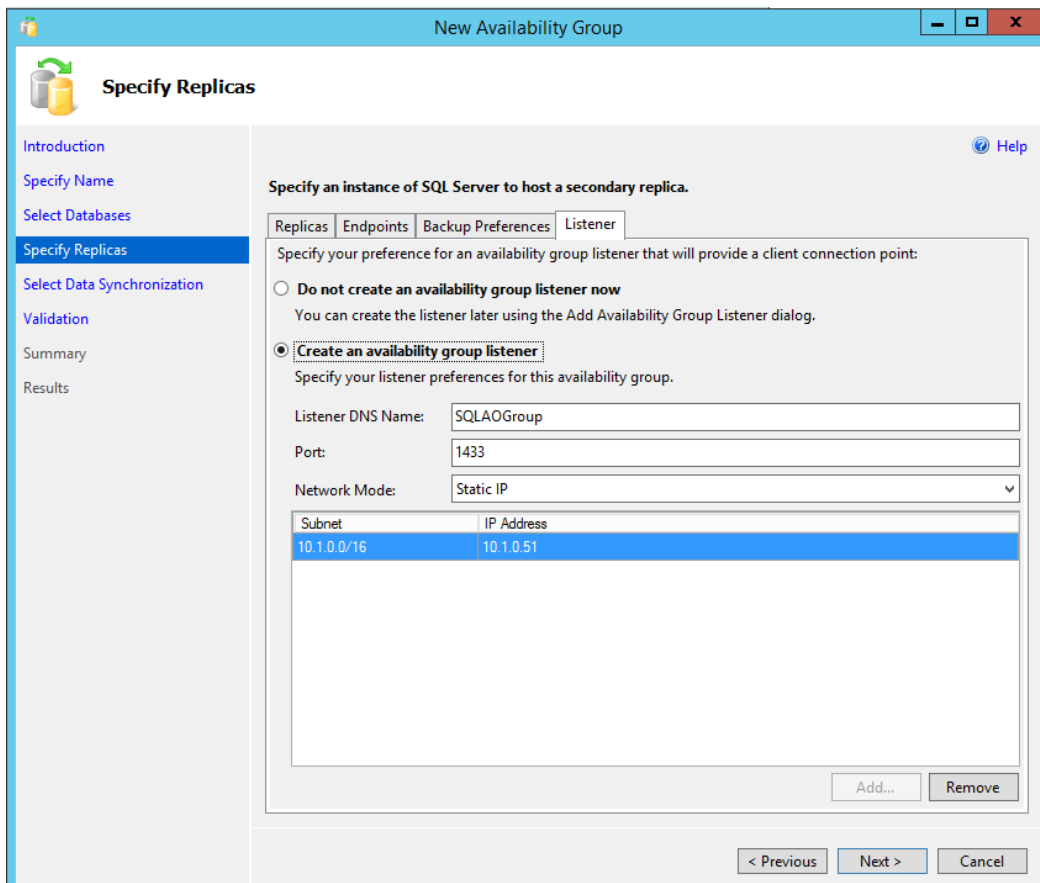
Nachdem Sie Ihre Replikas konfiguriert haben, wählen Sie den Reiter Endpoints. Da AlwaysOn auf Datenbankspiegelung aufbaut, legen Sie hier einen TCP-Port fest, über den die Transaktionen kopiert werden. Standardmäßig wird der Port 5022 verwendet, den auch die Datenbankspiegelung verwendet. Über diesen Port werden die Transaktionen zwischen den Replikas kopiert. Der Name für den AlwaysOn Endpoint ist normalerweise „Hadr_endpoint“. In unserem Beispiel verfügten die beiden Server SQL1 und SQL2 bereits über einen Mirroring-endpoint mit Namen Mirroring. Die vorhandenen Endpoints werden von AlwaysOn in diesem Fall weiterverwendet.



Über die Backup Preference können Sie steuern, welche Server für Backups verwendet werden. So können Sie Ihre Primary Instanz von Backup-Verkehr entlasten. Wählen Sie hier „Secondary Only“, werden nur Secondaries für das Backup genutzt. Über die Backup-Priorität können Sie festlegen, welcher Server als erstes für das Backup angesprochen wird, wenn mehrere alternative Server zur Verfügung stehen, wobei 1 für die niedrigste und 100 für die höchste Priorität steht. Die Standardeinstellung ist „Prefer Secondary“, aber in unserem Beispiel ist die Konfiguration so angepasst, dass Backups nur vom Secondaries durchgeführt werden dürfen.



Der letzte Reiter Listener legt eine neue Cluster-Ressource an, also einen Computernamen samt IP-Adresse und Computerkonto im AD.

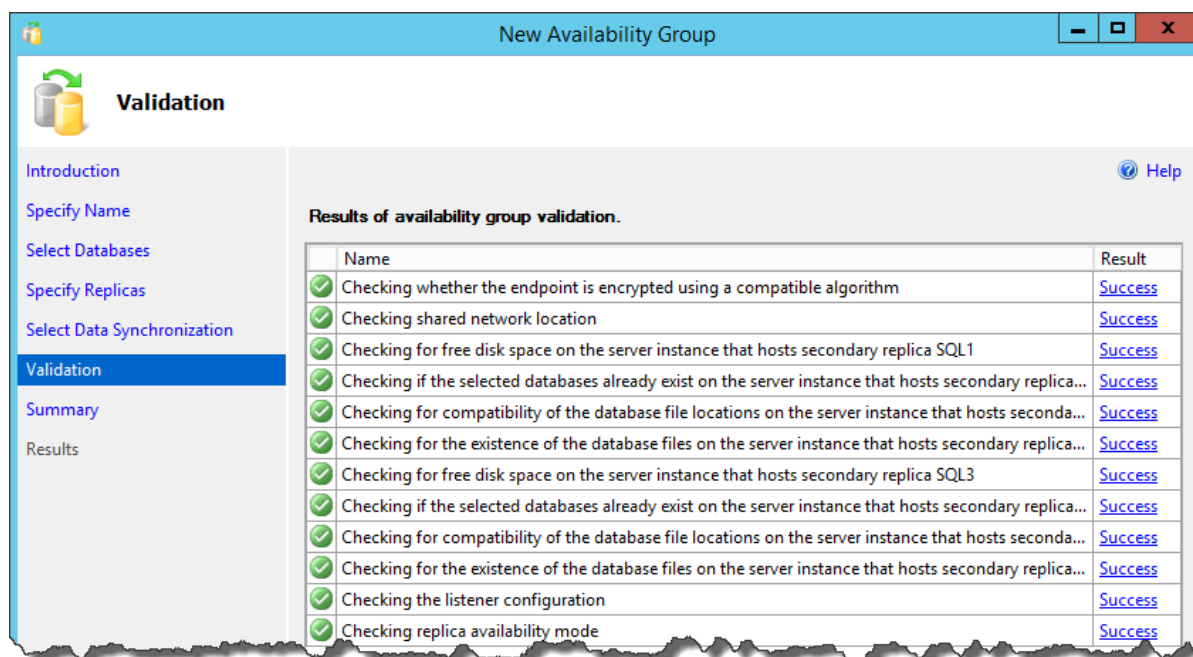


Geben Sie hier den Namen an, unter dem die Availability Group erreichbar sein soll, sowie einen Port und eine IP-Adresse. Wichtig ist, dass das Cluster-Computerkonto das Recht hat, neue Computerkonten im AD anzulegen, da dieser Schritt sonst fehlschlägt. Mehr Informationen hierzu erhalten Sie hier:

<http://www.netz-weise-it.training/weisheiten/tipp/always-on-ag-das-erstellen-eines-listeners-schlaegt-fehl.html>.

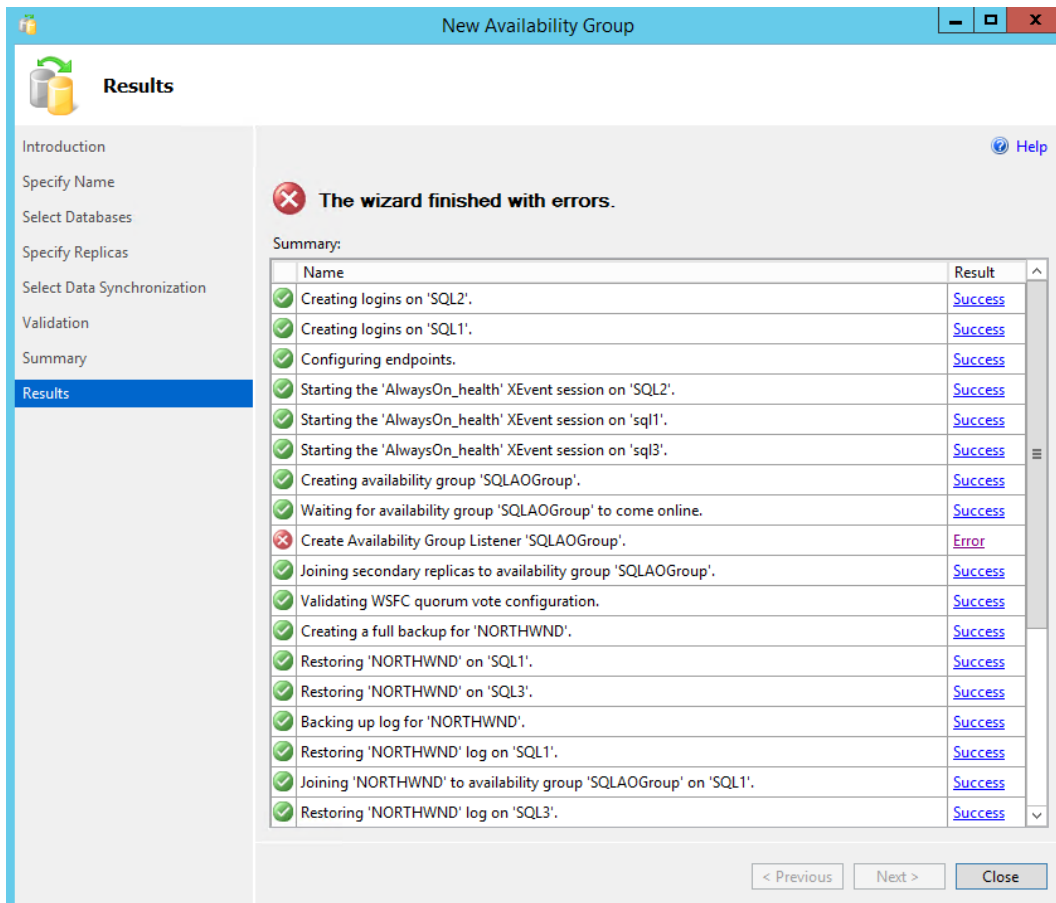
Da der Assistent selber dafür sorgt, dass ein Backup der Datenbanken der AlwaysOn Group gemacht und auf die Replikas übertragen wird, müssen Sie unter „Select Data Synchronization“ eine Freigabe angeben, auf die alle Server Zugriff haben. Im Beispiel habe ich auf dem Server der Primary Instanz eine Freigabe angelegt, auf die alle Benutzerkonten aller teilnehmenden Instanzen Zugriff haben. Stellen Sie sicher, dass Sie sowohl Freigabe- als auch NTFS-Berechtigungen vergeben. Handelt es sich um Produktionsinstanzen, sollten Sie außerdem sicherstellen, dass nur die SQL-Server Zugriff auf die Freigabe haben, um Benutzer daran zu hindern, auf die Backups zuzugreifen.

Der Server prüft jetzt ein letztes Mal, ob alle Voraussetzungen erfüllt sind.

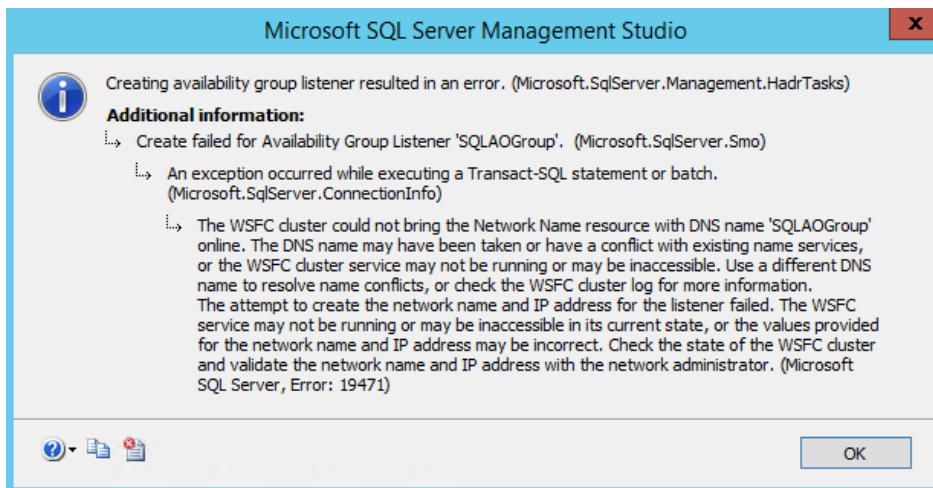


Sind alle Tests durchgelaufen, klicken sehen Sie unter Summary noch einmal die Zusammenfassung Ihrer Konfiguration. Über Script können Sie sich das Script erzeugen lassen, mit dem der Assistent die Einrichtung vornimmt. Das Script aus unserem Beispiel habe ist in Anhang A angefügt.

Starten Sie nun die Einrichtung. Der Assistent sollte die komplette Konfiguration selbständig durchführen. Ist Ihre Einrichtung nicht erfolgreich, zeigt Ihnen der Assistent an, welcher Teil der Konfiguration fehlgeschlagen ist.

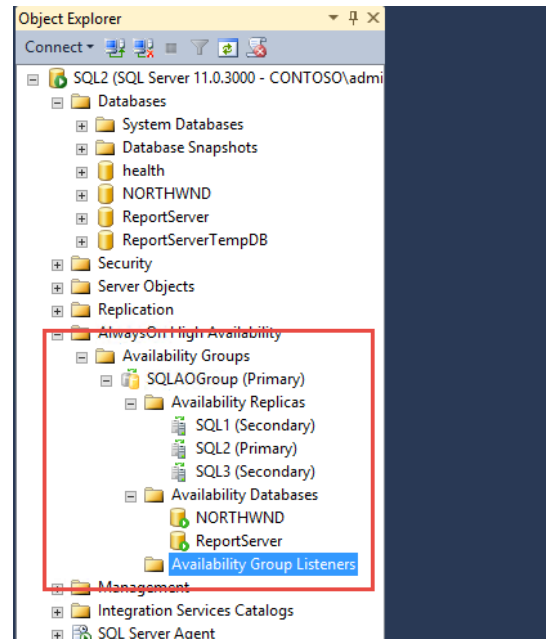
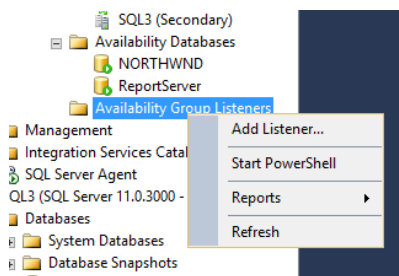


Im diesem Fall ist die Installation bis auf die Einrichtung des Listeners durchgelaufen. Der Listener ist die Failover-Cluster Resource, die den Failover steuert. Klicken Sie auf Error, wird Ihnen eine erweiterte Fehlermeldung angezeigt:

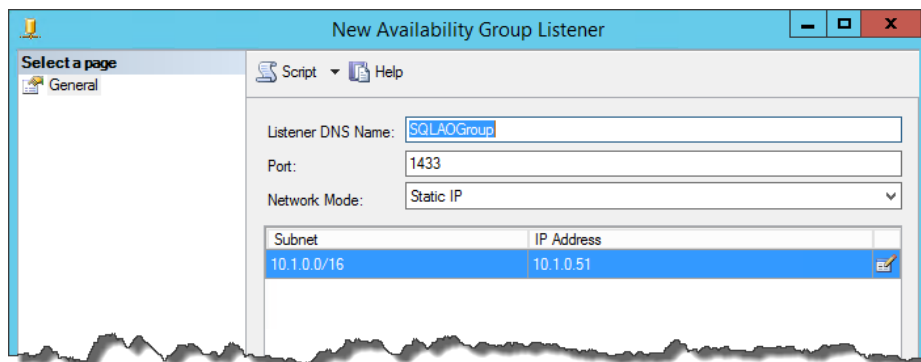


In unserem Fall liegt die Ursache im Cluster-Dienst, der das Computer-Konto nicht anlegen konnte. Den Fehler habe ich bereits weiter oben beschrieben. Es fehlt die Berechtigung zum Anlegen von Computerkonten im AD. Da alle weiteren Einrichtungen erfolgreich abgeschlossen wurden, ist AlwaysOn aber bereits aktiv, und lediglich der Zugriff über den Listener-Namen ist nicht möglich. Wenn Sie im Management-Studio unter AlwaysOn High Availability schauen (und gegebenenfalls einmal F5 zum Aktualisieren drücken), sehen Sie, dass ihre Availability Group eingerichtet ist und läuft.

Tatsächlich können Sie sich auch direkt mit den einzelnen Instanzen verbinden, ohne den Listener zu benutzen. Die Verbindung über Server\Instanz ist weiterhin möglich, und Sie werden hier alle Datenbanken sehen können. Jegliche Failover-Features sind jedoch nicht verfügbar, da hierzu der Listener notwendig ist. Glücklicherweise ist das nachträgliche Einrichten eines Listeners sehr einfach zu bewerkstelligen. Öffnen Sie hierzu das Kontextmenü von „Availability Group Listeners“ und wählen Sie „Add Listener“.



Hierauf öffnet sich der Dialog zum Einrichten eines Listeners. Tragen Sie einfach die Daten, die sie bereits im Einrichtungsassistenten verwendet hatten, ein. Achten Sie darauf, den „Network Mode“ auf Static IP umzustellen, um eine statische IP zu vergeben.



Klicken Sie auf OK, so wird der Listener neu angelegt.



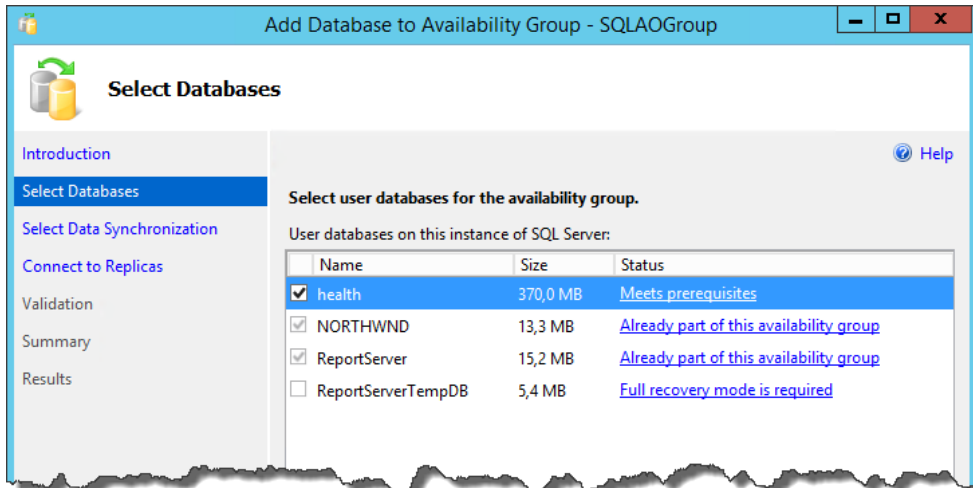
Wissenwertes zu Availability Groups

Availability-Groups unterscheiden sich in einem ganz wesentlichen Punkt von der Datenbankspiegelung. Bei der Spiegelung konfigurieren Sie jeweils einzelne Datenbanken. Eine Availability-Group kann jedoch mehrere Datenbanken zusammenfassen. Die Availability Group wird zu einer Einheit, die gemeinsam verwaltet wird. Kommt es beispielsweise zu einem Failover, so wird jeweils die komplette Group geschwenkt, nicht nur eine einzelne Datenbank. Sie konfigurieren mit der Availability Group also Abhängigkeiten, über die Sie sich bewusst sein sollten! Wollen Sie mehrere Datenbanken unabhängig voneinander schwenken, müssen Sie mehrere Availability-Groups einrichten! Wenn Sie eine neue Datenbank in eine Availability-Group aufnehmen wollen, so können Sie dies über das Management-Studio machen.

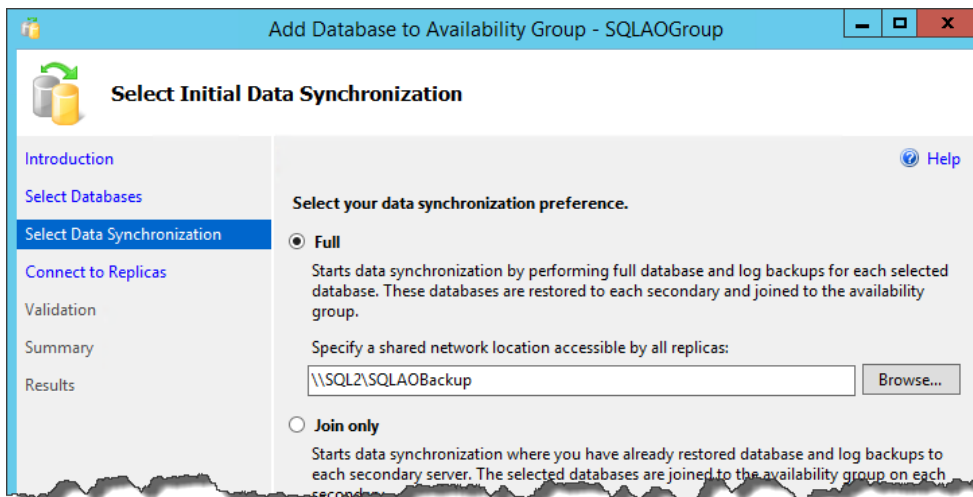
Hinzufügen einer Datenbank zu einer Availability Group

Sie können einer Availability-Group jederzeit weitere Datenbanken hinzufügen. Achten Sie aber darauf, dass Sie mit einer Availability Group auch immer Abhängigkeiten definieren!

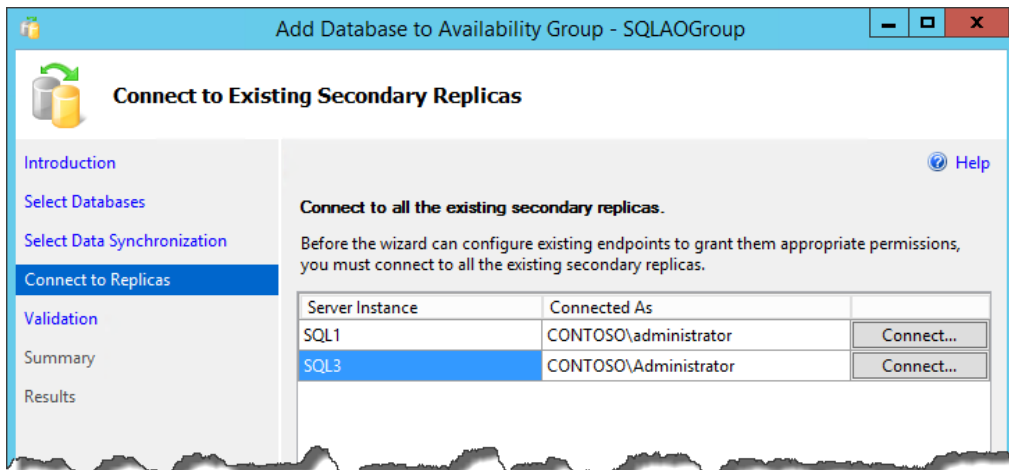
Klicken Sie zum Hinzufügen im Management-Studio unter Ihrer Availability Group auf „Availability databases“ und öffnen Sie das Kontextmenü. Es öffnet sich ein Assistent. Wählen Sie unter „Select Database“ die Datenbank aus, die sie hinzufügen möchten. Es gelten die gleichen Voraussetzungen wie oben definiert, also Datenbank im Full Recovery Mode und ein durchgeführtes Full Backup.



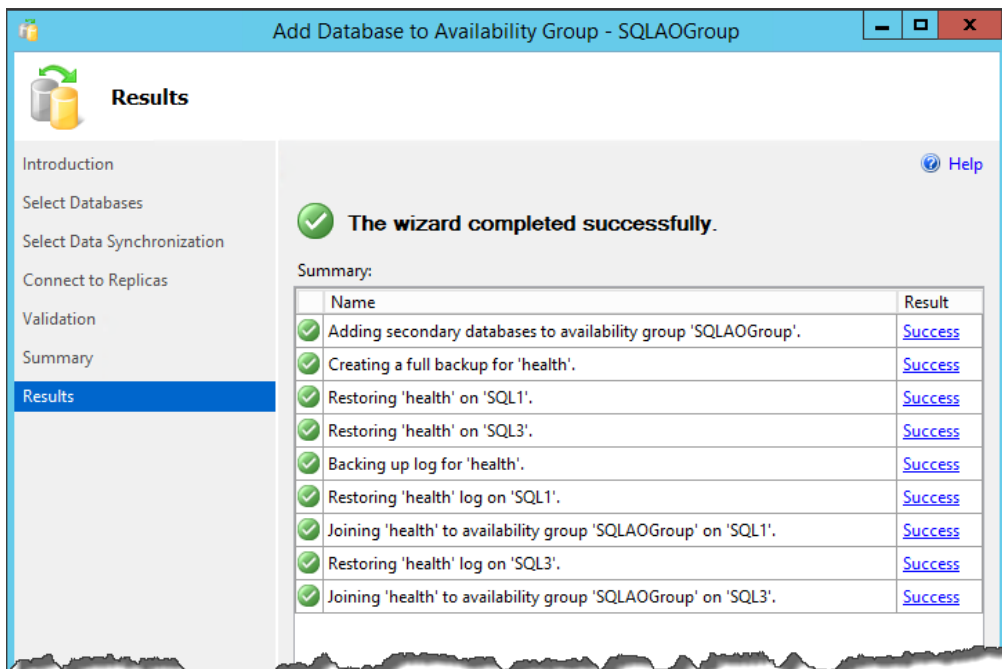
Wählen Sie wieder eine Backup-Freigabe aus. Der Assistent erzeugt automatisch ein Backup und kopiert es auf die Replikas.



Es muss eine Netzwerkverbindung zu den Ziel-Replikas hergestellt werden. Konnektieren Sie hierzu den Zielmaschinen. Standardmässig wird Ihr Benutzerkonto verwendet. Achten Sie darauf, dass Sie die notwendigen Berechtigungen besitzen.



Es findet eine Validierung statt, die von einer Zusammenfassung abgeschlossen wird. Nach erfolgreicher Einrichtung der Datenbank erscheint folgendes Fenster:



Die neue Datenbank ist in die Availability Group aufgenommen worden.

Read Only Routing konfigurieren

Mit dem Read Only Routing können Clients, die nur lesend auf die Datenbanken zugreifen, direkt auf einen Secondary weitergeleitet werden. Voraussetzung hierfür ist, dass der Client in seinem Connection-String „Read Intent“ angegeben hat, dass der Primary über eine Read Only Routing-Tabelle verfügt, und dass es einen oder mehrere Secondaries gibt, die für „Read Intent Only“ konfiguriert worden sind. Außerdem benötigen alle Secondaries eine Read-Only Routing URL. Die URL folgt folgendem Schema:

Tcp://Servername/Instanz:Port

Matt Neerinx hat in seinem Blog ein Script veröffentlicht, das die Read Only Routing URL per Script erzeugt. Das Script finden Sie im Anhang B dieses Dokuments, oder im seinem msdn-Blog:

<http://blogs.msdn.com/b/mattn/archive/2012/04/25/calculating-read-only-routing-url-for-alwayson.aspx>

Wenn Sie die URL bestimmt haben, müssen Sie auf dem Primary die URL auf allen Secondaries einrichten. Verwenden Sie dafür das SQL-Statement „Alter Availability Group“.

```
Alter Availability Group <Groupname>
Modify Replica ON <Server>
With (SECONDARY_ROLE ( READ_ONLY_ROUTING_URL = 'TCP://Servername:Port' ));
```

Führen Sie dieses Script auf allen Servern aus, um für alle Server ein Read-Only Routing URL zu konfigurieren. In unserem Lab konfigurieren Sie daher:

```
Alter Availability Group SQLA0Group
Modify Replica ON 'SQL1'
With (SECONDARY_ROLE ( READ_ONLY_ROUTING_URL = 'TCP://sql1.contoso.com:1433' ));

Alter Availability Group SQLA0Group
Modify Replica ON 'SQL2'
With (SECONDARY_ROLE ( READ_ONLY_ROUTING_URL = 'TCP://sql2.contoso.com:1433' ));

Alter Availability Group SQLA0Group
Modify Replica ON 'SQL3'
With (SECONDARY_ROLE ( READ_ONLY_ROUTING_URL = 'TCP://sql3.contoso.com:1433' ));
```

Als nächstes konfigurieren Sie die Read-Only Routing-Tabelle auf dem Primary mit dem Befehl

```
ALTER AVAILABILITY GROUP <Groupname>
MODIFY REPLICAS ON <PrimaryServer>
WITH (PRIMARY_ROLE
(READ_ONLY_ROUTING_LIST=(<Server1>,<Server2>)));
```

Die Reihenfolge der Server, die Sie angeben, ist wichtig, da die Reihenfolge auch die Priorität der Server bestimmt. Da beim Routing kein Lastausgleich stattfindet, verbinden sich die Clients immer mit dem ersten Server in der Liste, es sei denn, dieser ist nicht verfügbar.

In unserem Lab führen Sie zum Konfigurieren der Routing-Liste folgenden Befehl aus:

```
ALTER AVAILABILITY GROUP [SQLA0Group]
MODIFY REPLICAS ON N'SQL2'
WITH (PRIMARY_ROLE
(READ_ONLY_ROUTING_LIST=('SQL3', 'SQL1')));
```

Wichtig ist, dass die Routing-Liste lokal ist, also nur für den aktuellen Primary gilt. Haben Sie einen Failover konfiguriert, müssen die die Routing-Liste auch auf dem potentiellen Primary ausführen. In unserem Fall ist das SQL1:

```
ALTER AVAILABILITY GROUP [SQLA0Group]
MODIFY REPLICA ON N'SQL2'
WITH (PRIMARY_ROLE
(READ_ONLY_ROUTING_LIST=('SQL3', 'SQL1')));
```

Die Routing-Liste lässt sich mit folgendem Statement anzeigen:

```
SELECT ag.name as "Availability Group", ar.replica_server_name as "When Primary Is",
       r1.routing_priority as "Routing Priority", ar2.replica_server_name as " Routed To",
       ar.secondary_role_allow_connections_desc, ar2.read_only_routing_url
FROM sys.availability_read_only_routing_lists rl
     inner join sys.availability_replicas ar on rl.replica_id = ar.replica_id
     inner join sys.availability_replicas ar2 on rl.read_only_replica_id =
ar2.replica_id
     inner join sys.availability_groups ag on ar.group_id = ag.group_id
ORDER BY ag.name, ar.replica_server_name, rl.routing_priority
```

(Quelle: <http://blogs.msdn.com/b/alwaysonpro/archive/2014/01/22/modifying-alwayson-read-only-routing-lists.aspx>)

Wenn Sie die Routing-Liste ändern möchten, brauchen Sie das Alter Availability Group Statement nur auf dem Primary mit geänderten Servern neu auszuführen.

Überwachung der AlwaysOn Availability Group

Zur Überwachung steht Ihnen das AlwaysOn Dashboard zur Verfügung. Das Dashboard erreichen Sie, indem Sie im Management Studio die Availability Group aufrufen und „Show Dashboard“ auswählen.

SQLAOGroup: hosted by SQL2 (Replica role: Primary) Last updated: 30.03.2015 15:13:03

Availability group state: ✔ Healthy

Primary instance: SQL2

Failover mode: Automatic

Cluster state: SQLCluster (Normal Quorum)

Availability replica:

Name	Role	Failover Mode	Synchronization State	Issues
✔ SQL1	Secon...	Automatic	Synchronized	
✔ SQL2	Primary	Automatic	Synchronized	
✔ SQL3	Secon...	Manual	Synchronizing	

Group by ▾

Name	Replica	Synchronization State	Failover Read...	Issues
SQL1				
✔ health	SQL1	Synchronized	No Data Loss	
✔ NORTHWND	SQL1	Synchronized	No Data Loss	
✔ ReportServer	SQL1	Synchronized	No Data Loss	
SQL2				
✔ health	SQL2	Synchronized	No Data Loss	
✔ NORTHWND	SQL2	Synchronized	No Data Loss	
✔ ReportServer	SQL2	Synchronized	No Data Loss	
SQL3				
✔ health	SQL3	Synchronizing	Data Loss	
✔ NORTHWND	SQL3	Synchronizing	Data Loss	
✔ ReportServer	SQL3	Synchronizing	Data Loss	

Das Dashboard zeigt Ihnen einen Überblick über den Status Ihrer Availability Group. Wichtiger ist aber, dass Sie von hier aus die Availability Group auch steuern können. Oben Rechts im Dashboard findet man die Möglichkeit, einen manuellen Failover zwischen den Instanzen durchzuführen. Darunter können Sie sich den Status der Availability Group anzeigen lassen. SQL Server verwendet eine Reihe von vordefinierten Extended Events, um die Funktionalität Ihre AlwaysOn Gruppe zu überwachen. Klicken Sie auf „View AlwaysOn Health Events“, öffnet sich ein weiteres Fenster. Standardmäßig werden hier in tabellarischer Auflistung die Event-namen und die Aufzeichnungszeit angezeigt. Wählen Sie ein Event aus, sehen Sie unter Details die aufgezeichneten Informationen. Die Spalten, die in der Tabellarischen Ansicht angezeigt werden, können Sie erweitern, indem Sie auf die Kopfzeile der Tabelle mit der rechten Maustaste klicken. Im Kontextmenü finden Sie einen Eintrag „Choose Column“. Alternativ können Sie auch die Spalte unter Details klicken und „Show Column in Table“

SQL2 - AlwaysOn_health: event_file x SQLAOGroup:SQL2 SQLQuery6.sql - SQ...administrator (

Displaying 47 Events

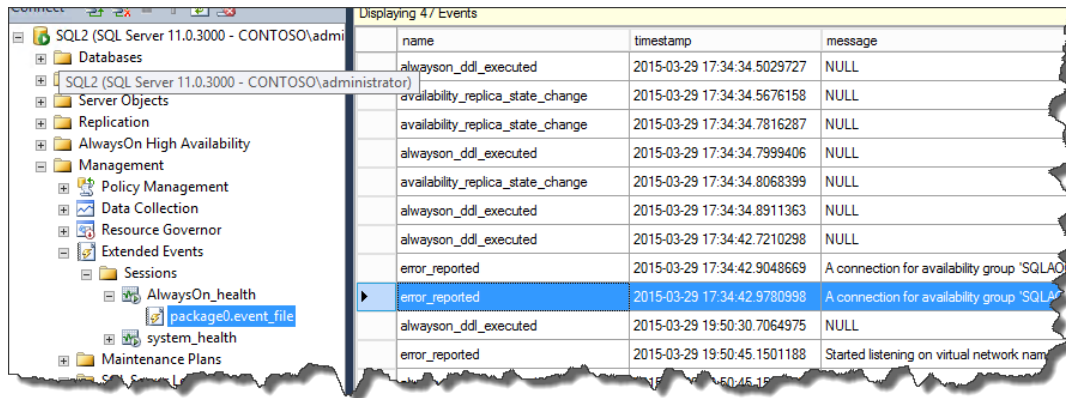
name	timestamp
alwayson_ddl_executed	2015-03-29 17:34:34.5029727
availability_replica_state_change	2015-03-29 17:34:34.5676158
availability_replica_state_change	2015-03-29 17:34:34.7816287
alwayson_ddl_executed	2015-03-29 17:34:34.7999406
availability_replica_state_change	2015-03-29 17:34:34.8068399
alwayson_ddl_executed	2015-03-29 17:34:34.8911363
alwayson_ddl_executed	2015-03-29 17:34:42.7210298
error_reported	2015-03-29 17:34:42.9048669
error_reported	2015-03-29 17:34:42.9780998
alwayson_ddl_executed	2015-03-29 19:50:30.7064975
error_reported	2015-03-29 19:50:45.1501188
alwayson_ddl_executed	2015-03-29 19:50:45.1502037
alwayson_ddl_executed	2015-03-29 20:06:46.1946729
alwayson_ddl_executed	2015-03-29 20:06:46.2154211
error_reported	2015-03-29 20:48:36.6971277
error_reported	2015-03-29 20:48:37.0274879
error_reported	2015-03-30 10:25:23.0583267
error_reported	2015-03-30 11:09:10.6209967

Event: availability_replica_state_change (2015-03-29 17:34:34.7816287)

Details

Field	Value
availability_group_id	8881364F-0FEE-49F9-A544-6E2B9C1427E4
availability_group_n...	SQLAOGroup
availability_replica_id	63B00009-7256-4123-AC35-6B94D82795E7
current_state	PRIMARY_PENDING
previous_state	RESOLVING_NORMAL

auswählen. Die Definition der standardmäßig gewählten Events finden Sie, wenn Sie im Management-Studio „Management -> Extended Events -> Sessions -> AlwaysOn_health“ auswählen.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'AlwaysOn_health' session is selected under 'Extended Events'. The main pane displays a table of events with the following data:

name	timestamp	message
alwayson_ddl_executed	2015-03-29 17:34:34.5029727	NULL
availability_replica_state_change	2015-03-29 17:34:34.5676158	NULL
availability_replica_state_change	2015-03-29 17:34:34.7816287	NULL
alwayson_ddl_executed	2015-03-29 17:34:34.7999406	NULL
availability_replica_state_change	2015-03-29 17:34:34.8068399	NULL
alwayson_ddl_executed	2015-03-29 17:34:34.8911363	NULL
alwayson_ddl_executed	2015-03-29 17:34:42.7210298	NULL
error_reported	2015-03-29 17:34:42.9048669	A connection for availability group 'SQLAO
error_reported	2015-03-29 17:34:42.9780998	A connection for availability group 'SQLAO
alwayson_ddl_executed	2015-03-29 19:50:30.7064975	NULL
error_reported	2015-03-29 19:50:45.1501188	Started listening on virtual network nam

System-Views für AlwaysOn

AlwaysOn stellt eine ganze Reihe von System Views und Dynamic Management Views zur Verfügung. Zum Anzeigen der Routing-Listen und Replikas verwenden Sie folgende Abfragen:

```
select * from sys.availability_read_only_routing_lists
select * from sys.availability_replicas
select * from sys.availability_groups
```

Eine vollständige Auflistung aller Views finden Sie bei Microsoft:

AlwaysOn Availability Groups Catalog Views

<https://msdn.microsoft.com/en-us/library/ff878615.aspx>

AlwaysOn Availability Groups Dynamic Management Views and Functions

<https://msdn.microsoft.com/en-us/library/ff877943.aspx>

Dynamic Management Views and System Catalog Views (AlwaysOn Availability Groups)

[https://technet.microsoft.com/en-us/library/dn135319\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/dn135319(v=sql.110).aspx)

Weiterführende Links

Configure Read-Only Access on an Availability Replica

<https://msdn.microsoft.com/en-us/library/hh213002.aspx>

Configure Read-Only Routing for an Availability Group (SQL Server)

<https://technet.microsoft.com/en-us/library/hh710054.aspx>

Modifying AlwaysOn Read Only Routing Lists

<http://blogs.msdn.com/b/alwaysonpro/archive/2014/01/22/modifying-alwayson-read-only-routing-lists.aspx>

Calculating read_only_routing_url for AlwaysOn

<http://blogs.msdn.com/b/mattn/archive/2012/04/25/calculating-read-only-routing-url-for-alwayson.aspx>

Read-Only Routing with SQL Server 2012 Always On Database Availability Groups

<http://social.technet.microsoft.com/wiki/contents/articles/13503.read-only-routing-with-sql-server-2012-always-on-database-availability-groups.aspx>

Part 2 - Read-Only Routing with SQL Server 2012 Always On Database Availability Groups (Load Balancing Read Intent Connections)

<http://social.technet.microsoft.com/wiki/contents/articles/16083.part-2-read-only-routing-with-sql-server-2012-always-on-database-availability-groups-load-balancing-read-intent-connections.aspx>

AlwaysOn Availability Groups Quiz and FAQ [Video]

<http://www.brentozar.com/archive/category/sql-server-2012-alwayson/>

Monitoring of Availability Groups (SQL Server)

<https://msdn.microsoft.com/en-us/library/ff877954.aspx>

Anhang A

```
--- YOU MUST EXECUTE THE FOLLOWING SCRIPT IN SQLCMD MODE.
:Connect SQL2
USE [master]
GO

CREATE LOGIN [contoso\sql3] FROM WINDOWS
GO

:Connect sql1
USE [master]
GO

CREATE LOGIN [contoso\sql3] FROM WINDOWS
GO

:Connect SQL2
IF (SELECT state FROM sys.endpoints WHERE name = N'Mirroring') <> 0
BEGIN
    ALTER ENDPOINT [Mirroring] STATE = STARTED
END
GO

use [master]
GO

GRANT CONNECT ON ENDPOINT::[Mirroring] TO [contoso\sql1]
GO

use [master]
GO

GRANT CONNECT ON ENDPOINT::[Mirroring] TO [contoso\sql3]
GO

:Connect sql1
IF (SELECT state FROM sys.endpoints WHERE name = N'Mirroring') <> 0
BEGIN
    ALTER ENDPOINT [Mirroring] STATE = STARTED
END
GO

use [master]
GO

GRANT CONNECT ON ENDPOINT::[Mirroring] TO [contoso\sql2]
GO

use [master]
GO

GRANT CONNECT ON ENDPOINT::[Mirroring] TO [contoso\sql3]
GO

:Connect sql3
IF (SELECT state FROM sys.endpoints WHERE name = N'Hadr_endpoint') <> 0
BEGIN
    ALTER ENDPOINT [Hadr_endpoint] STATE = STARTED
END
GO

use [master]
GO

GRANT CONNECT ON ENDPOINT::[Hadr_endpoint] TO [CONTOSO\SQL2]
```

```

GO

use [master]
GO

GRANT CONNECT ON ENDPOINT::[Hadr_endpoint] TO [CONTOSO\SQL1]
GO

:Connect SQL2
IF EXISTS(SELECT * FROM sys.server_event_sessions WHERE name='AlwaysOn_health')
BEGIN
    ALTER EVENT SESSION [AlwaysOn_health] ON SERVER WITH (STARTUP_STATE=ON);
END
IF NOT EXISTS(SELECT * FROM sys.dm_xe_sessions WHERE name='AlwaysOn_health')
BEGIN
    ALTER EVENT SESSION [AlwaysOn_health] ON SERVER STATE=START;
END
GO

:Connect sql1
IF EXISTS(SELECT * FROM sys.server_event_sessions WHERE name='AlwaysOn_health')
BEGIN
    ALTER EVENT SESSION [AlwaysOn_health] ON SERVER WITH (STARTUP_STATE=ON);
END
IF NOT EXISTS(SELECT * FROM sys.dm_xe_sessions WHERE name='AlwaysOn_health')
BEGIN
    ALTER EVENT SESSION [AlwaysOn_health] ON SERVER STATE=START;
END
GO

:Connect sql3
IF EXISTS(SELECT * FROM sys.server_event_sessions WHERE name='AlwaysOn_health')
BEGIN
    ALTER EVENT SESSION [AlwaysOn_health] ON SERVER WITH (STARTUP_STATE=ON);
END
IF NOT EXISTS(SELECT * FROM sys.dm_xe_sessions WHERE name='AlwaysOn_health')
BEGIN
    ALTER EVENT SESSION [AlwaysOn_health] ON SERVER STATE=START;
END
GO

:Connect SQL2
USE [master]
GO

CREATE AVAILABILITY GROUP [SQLAOGroup]
WITH (AUTOMATED_BACKUP_PREFERENCE = SECONDARY_ONLY)
FOR DATABASE [NORTHWND], [ReportServer]
REPLICA ON N'SQL1' WITH (ENDPOINT_URL = N'TCP://SQL1.contoso.com:5022', FAILOVER_MODE =
AUTOMATIC, AVAILABILITY_MODE = SYNCHRONOUS_COMMIT, BACKUP_PRIORITY = 50,
SECONDARY_ROLE(ALLOW_CONNECTIONS = ALL)),
    N'SQL2' WITH (ENDPOINT_URL = N'TCP://SQL2.contoso.com:5022', FAILOVER_MODE =
AUTOMATIC, AVAILABILITY_MODE = SYNCHRONOUS_COMMIT, BACKUP_PRIORITY = 50,
SECONDARY_ROLE(ALLOW_CONNECTIONS = ALL)),
    N'SQL3' WITH (ENDPOINT_URL = N'TCP://sql3.contoso.com:5022', FAILOVER_MODE = MANUAL,
AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT, BACKUP_PRIORITY = 0,
SECONDARY_ROLE(ALLOW_CONNECTIONS = READ_ONLY));
GO

:Connect SQL2
USE [master]
GO

ALTER AVAILABILITY GROUP [SQLAOGroup]
ADD LISTENER N'SQLAOGroup' (
WITH IP
((N'10.1.0.51', N'255.255.0.0')

```

```

)
, PORT=1433);
GO

:Connect sql1
ALTER AVAILABILITY GROUP [SQLAOGroup] JOIN;
GO

:Connect sql3
ALTER AVAILABILITY GROUP [SQLAOGroup] JOIN;
GO

:Connect SQL2
BACKUP DATABASE [NORTHWND] TO DISK = N'\\SQL2\SQLAOLBackup\NORTHWND.bak' WITH COPY_ONLY,
FORMAT, INIT, SKIP, REWIND, NOUNLOAD, COMPRESSION, STATS = 5
GO

:Connect sql1
RESTORE DATABASE [NORTHWND] FROM DISK = N'\\SQL2\SQLAOLBackup\NORTHWND.bak' WITH
NORECOVERY, NOUNLOAD, STATS = 5
GO

:Connect sql3
RESTORE DATABASE [NORTHWND] FROM DISK = N'\\SQL2\SQLAOLBackup\NORTHWND.bak' WITH
NORECOVERY, NOUNLOAD, STATS = 5
GO

:Connect SQL2
BACKUP LOG [NORTHWND] TO DISK = N'\\SQL2\SQLAOLBackup\NORTHWND_20150329153419.trn' WITH
NOFORMAT, NOINIT, NOSKIP, REWIND, NOUNLOAD, COMPRESSION, STATS = 5
GO

:Connect sql1
RESTORE LOG [NORTHWND] FROM DISK = N'\\SQL2\SQLAOLBackup\NORTHWND_20150329153419.trn' WITH
NORECOVERY, NOUNLOAD, STATS = 5
GO

:Connect sql1
-- Wait for the replica to start communicating
begin try
declare @conn bit
declare @count int
declare @replica_id uniqueidentifier
declare @group_id uniqueidentifier
set @conn = 0
set @count = 30 -- wait for 5 minutes

if (serverproperty('IsHadrEnabled') = 1)
    and (isnull((select member_state from master.sys.dm_hadr_cluster_members where
upper(member_name COLLATE Latin1_General_CI_AS) =
upper(cast(serverproperty('ComputerNamePhysicalNetBIOS') as nvarchar(256)) COLLATE
Latin1_General_CI_AS)), 0) <> 0)
    and (isnull((select state from master.sys.database_mirroring_endpoints), 1) = 0)
begin
    select @group_id = ags.group_id from master.sys.availability_groups as ags where name =
N'SQLAOGroup'
        select @replica_id = replicas.replica_id from master.sys.availability_replicas as
replicas where upper(replicas.replica_server_name COLLATE Latin1_General_CI_AS) =
upper(@@SERVERNAME COLLATE Latin1_General_CI_AS) and group_id = @group_id
        while @conn <> 1 and @count > 0
        begin
            set @conn = isnull((select connected_state from
master.sys.dm_hadr_availability_replica_states as states where states.replica_id =
@replica_id), 1)
            if @conn = 1
                begin

```

```

-- exit loop when the replica is connected, or if the query cannot
find the replica status
        break
    end
    waitfor delay '00:00:10'
    set @count = @count - 1
end
end try
begin catch
-- If the wait loop fails, do not stop execution of the alter database statement
end catch
ALTER DATABASE [NORTHWND] SET HADR AVAILABILITY GROUP = [SQLAOGGroup];
GO

:Connect sql3
RESTORE LOG [NORTHWND] FROM DISK = N'\\SQL2\SQLAOLBackup\NORTHWND_20150329153419.trn' WITH
NORECOVERY, NOUNLOAD, STATS = 5
GO

:Connect sql3
-- Wait for the replica to start communicating
begin try
declare @conn bit
declare @count int
declare @replica_id uniqueidentifier
declare @group_id uniqueidentifier
set @conn = 0
set @count = 30 -- wait for 5 minutes

if (serverproperty('IsHadrEnabled') = 1)
    and (isnull((select member_state from master.sys.dm_hadr_cluster_members where
upper(member_name COLLATE Latin1_General_CI_AS) =
upper(cast(serverproperty('ComputerNamePhysicalNetBIOS') as nvarchar(256)) COLLATE
Latin1_General_CI_AS)), 0) <> 0)
    and (isnull((select state from master.sys.database_mirroring_endpoints), 1) = 0)
begin
    select @group_id = ags.group_id from master.sys.availability_groups as ags where name =
N'SQLAOGroup'
    select @replica_id = replicas.replica_id from master.sys.availability_replicas as
replicas where upper(replicas.replica_server_name COLLATE Latin1_General_CI_AS) =
upper(@@SERVERNAME COLLATE Latin1_General_CI_AS) and group_id = @group_id
    while @conn <> 1 and @count > 0
    begin
        set @conn = isnull((select connected_state from
master.sys.dm_hadr_availability_replica_states as states where states.replica_id =
@replica_id), 1)
        if @conn = 1
        begin
            -- exit loop when the replica is connected, or if the query cannot
            find the replica status
                break
            end
            waitfor delay '00:00:10'
            set @count = @count - 1
        end
    end
end try
begin catch
-- If the wait loop fails, do not stop execution of the alter database statement
end catch
ALTER DATABASE [NORTHWND] SET HADR AVAILABILITY GROUP = [SQLAOGGroup];
GO

:Connect SQL2
BACKUP DATABASE [ReportServer] TO DISK = N'\\SQL2\SQLAOLBackup\ReportServer.bak' WITH
COPY_ONLY, FORMAT, INIT, SKIP, REWIND, NOUNLOAD, COMPRESSION, STATS = 5

```

```

GO

:Connect sql1
RESTORE DATABASE [ReportServer] FROM DISK = N'\\SQL2\SQLA0Backup\ReportServer.bak' WITH
NORECOVERY, NOUNLOAD, STATS = 5
GO

:Connect sql3
RESTORE DATABASE [ReportServer] FROM DISK = N'\\SQL2\SQLA0Backup\ReportServer.bak' WITH
NORECOVERY, NOUNLOAD, STATS = 5
GO

:Connect SQL2
BACKUP LOG [ReportServer] TO DISK = N'\\SQL2\SQLA0Backup\ReportServer_20150329153419.trn'
WITH NOFORMAT, NOINIT, NOSKIP, REWIND, NOUNLOAD, COMPRESSION, STATS = 5
GO

:Connect sql1
RESTORE LOG [ReportServer] FROM DISK =
N'\\SQL2\SQLA0Backup\ReportServer_20150329153419.trn' WITH NORECOVERY, NOUNLOAD, STATS =
5
GO

:Connect sql1
-- Wait for the replica to start communicating
begin try
declare @conn bit
declare @count int
declare @replica_id uniqueidentifier
declare @group_id uniqueidentifier
set @conn = 0
set @count = 30 -- wait for 5 minutes

if (serverproperty('IsHadrEnabled') = 1)
    and (isnull((select member_state from master.sys.dm_hadr_cluster_members where
upper(member_name COLLATE Latin1_General_CI_AS) =
upper(cast(serverproperty('ComputerNamePhysicalNetBIOS') as nvarchar(256)) COLLATE
Latin1_General_CI_AS)), 0) <> 0)
    and (isnull((select state from master.sys.database_mirroring_endpoints), 1) = 0)
begin
    select @group_id = ags.group_id from master.sys.availability_groups as ags where name =
N'SQLA0Group'
        select @replica_id = replicas.replica_id from master.sys.availability_replicas as
replicas where upper(replicas.replica_server_name COLLATE Latin1_General_CI_AS) =
upper(@@SERVERNAME COLLATE Latin1_General_CI_AS) and group_id = @group_id
        while @conn <> 1 and @count > 0
        begin
            set @conn = isnull((select connected_state from
master.sys.dm_hadr_availability_replica_states as states where states.replica_id =
@replica_id), 1)
            if @conn = 1
            begin
                -- exit loop when the replica is connected, or if the query cannot
                find the replica status
                break
            end
            end
            waitfor delay '00:00:10'
            set @count = @count - 1
        end
    end
end
end try
begin catch
    -- If the wait loop fails, do not stop execution of the alter database statement
end catch
ALTER DATABASE [ReportServer] SET HADR AVAILABILITY GROUP = [SQLA0Group];
GO

```

```

:Connect sql3
RESTORE LOG [ReportServer] FROM DISK =
N'\\SQL2\SQLAOLBackup\ReportServer_20150329153419.trn' WITH NORECOVERY, NOUNLOAD, STATS =
5
GO

:Connect sql3
-- Wait for the replica to start communicating
begin try
declare @conn bit
declare @count int
declare @replica_id uniqueidentifier
declare @group_id uniqueidentifier
set @conn = 0
set @count = 30 -- wait for 5 minutes

if (serverproperty('IsHadrEnabled') = 1)
    and (isnull((select member_state from master.sys.dm_hadr_cluster_members where
upper(member_name COLLATE Latin1_General_CI_AS) =
upper(cast(serverproperty('ComputerNamePhysicalNetBIOS') as nvarchar(256)) COLLATE
Latin1_General_CI_AS)), 0) <> 0)
    and (isnull((select state from master.sys.database_mirroring_endpoints), 1) = 0)
begin
    select @group_id = ags.group_id from master.sys.availability_groups as ags where name =
N'SQLAOGroup'
        select @replica_id = replicas.replica_id from master.sys.availability_replicas as
replicas where upper(replicas.replica_server_name COLLATE Latin1_General_CI_AS) =
upper(@SERVERNAME COLLATE Latin1_General_CI_AS) and group_id = @group_id
        while @conn <> 1 and @count > 0
        begin
            set @conn = isnull((select connected_state from
master.sys.dm_hadr_availability_replica_states as states where states.replica_id =
@replica_id), 1)
            if @conn = 1
            begin
                -- exit loop when the replica is connected, or if the query cannot
                find the replica status
                break
            end
            end
            waitfor delay '00:00:10'
            set @count = @count - 1
        end
    end
end try
begin catch
    -- If the wait loop fails, do not stop execution of the alter database statement
end catch
ALTER DATABASE [ReportServer] SET HADR AVAILABILITY GROUP = [SQLAOGroup];

GO

```


Anhang B

```
-- Read-only routing url generation script.
-- Connect to each replica in your AlwaysOn cluster and run this script to get the
read_only_routing_url for the replica.
-- Then set this to the read_only_routing_url for the availability group replica =>
-- alter availability group MyAvailability Group modify replica on N'ThisReplica' with
(secondary_role(read_only_routing_url=N'<url>'))
print 'Read-only-routing url script v.2012.1.24.1'

print 'This SQL Server instance version is [' + cast(serverproperty('ProductVersion') as
varchar(256)) + ']'

if (ServerProperty('IsClustered') = 1)
begin
    print 'This SQL Server instance is a clustered SQL Server instance.'
end
else
begin
    print 'This SQL Server instance is a standard (not clustered) SQL Server instance.'
end

if (ServerProperty('IsHadrEnabled') = 1)
begin
    print 'This SQL Server instance is enabled for AlwaysOn.'
end
else
begin
    print 'This SQL Server instance is NOT enabled for AlwaysOn.'
end

-- Detect SQL Azure instance.
declare @is_sql_azure bit
set @is_sql_azure = 0

begin try
    set @is_sql_azure = 1
    exec('declare @i int set @i = sql_connection_mode()')
    print 'This SQL Server instance is a Sql Azure instance.'
end try
begin catch
    set @is_sql_azure = 0
    print 'This SQL Server instance is NOT a Sql Azure instance.'
end catch

-- Check that this is SQL 11 or later, otherwise fail fast.
if (@@microsoftversion / 0x01000000 < 11 or @is_sql_azure > 0)
begin
    print 'This SQL Server instance does not support read-only routing, exiting script.'
end
else
begin -- if server supports read-only routing

    -- Fetch the dedicated admin connection (dac) port.
    -- Normally it's always port 1434, but to be safe here we fetch it from the instance.
    -- We use this later to exclude the admin port from read_only_routing_url.
    declare @dac_port int
    declare @reg_value varchar(255)
    exec xp_instance_regread
        N'HKEY_LOCAL_MACHINE',
        N'SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQLServer\SuperSocketNetLib\AdminConnection\Tcp',
        N'TcpDynamicPorts',
        @reg_value output

    set @dac_port = cast(@reg_value as int)
```

```

print 'This SQL Server instance DAC (dedicated admin) port is ' + cast(@dac_port as
varchar(255))
if (@dac_port = 0)
begin
print 'Note a DAC port of zero means the dedicated admin port is not enabled.'
end

-- Fetch ListenOnAllIPs value.
-- If set to 1, this means the instance is listening to all IP addresses.
-- If set to 0, this means the instance is listening to specific IP addresses.
declare @listen_all int
exec xp_instance_regread
    N'HKEY_LOCAL_MACHINE',
    N'SOFTWARE\Microsoft\Microsoft SQL Server\MSSQLServer\SuperSocketNetLib\Tcp',
    N'ListenOnAllIPs',
    @listen_all output

if (@listen_all = 1)
begin
print 'This SQL Server instance is listening to all IP addresses (default mode).'
end
else
begin
print 'This SQL Server instance is listening to specific IP addresses
(ListenOnAllIPs is disabled).'
end

-- Check for dynamic port configuration, not recommended with read-only routing.
declare @tcp_dynamic_ports varchar(255)
exec xp_instance_regread
    N'HKEY_LOCAL_MACHINE',
    N'SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQLServer\SuperSocketNetLib\Tcp\IPAll',
    N'TcpDynamicPorts',
    @tcp_dynamic_ports output

if (@tcp_dynamic_ports = '0')
begin
print 'This SQL Server instance is listening on a dynamic tcp port, this is NOT A
RECOMMENDED CONFIGURATION when using read-only routing, because the instance port can
change each time the instance is restarted.'
end
else
begin
print 'This SQL Server instance is listening on fixed tcp port(s) (it is not
configured for dynamic ports), this is a recommended configuration when using read-only
routing.'
end

-- Calculate the server domain and instance FQDN.
-- We use @server_domain later to build the FQDN to the clustered instance.
declare @instance_fqdn varchar(255)
declare @server_domain varchar(255)

-- Get the instance FQDN using the xp_getnetname API
-- Note all cluster nodes must be in same domain, so this works for calculating cluster
FQDN.
set @instance_fqdn = ''
exec xp_getnetname @instance_fqdn output, 1

-- Remove embedded null character at end if found.
declare @terminator int
set @terminator = charindex(char(0), @instance_fqdn) - 1
if (@terminator > 0)
begin
set @instance_fqdn = substring(@instance_fqdn, 1, @terminator)
end

```

```

-- Build @server_domain using @instance_fqdn.
set @server_domain = @instance_fqdn

-- Remove trailing portion to extract domain name.
set @terminator = charindex('.', @server_domain)
if (@terminator > 0)
begin
    set @server_domain = substring(@server_domain, @terminator+1,
datalength(@server_domain))
end
print 'This SQL Server instance resides in domain ''' + @server_domain + ''''

if (ServerProperty('IsClustered') = 1)
begin
    -- Fetch machine name, which for a clustered SQL instance returns the network name
of the virtual server.
    -- Append @server_domain to build the FQDN.
    set @instance_fqdn = cast(serverproperty('MachineName') as varchar(255)) + '.' +
@server_domain
end

declare @ror_url varchar(255)
declare @instance_port int

set @ror_url = ''

-- Get first available port for instance.
select
top 1    -- Select first matching port
@instance_port = port
from sys.dm_tcp_listener_states
where
type=0 -- Type 0 = TSQL (to avoid mirroring endpoint)
and
state=0 -- State 0 is online
and
port <> @dac_port -- Avoid DAC port (admin port)
and
-- Avoid availability group listeners
ip_address not in (select ip_address from sys.availability_group_listener_ip_addresses
agls)
group by port
order by port asc -- Pick first port in ascending order

-- Check if there are multiple ports and warn if this is the case.
declare @list_of_ports varchar(max)
set @list_of_ports = ''

select
@list_of_ports = @list_of_ports +
    case datalength(@list_of_ports)
    when 0 then cast(port as varchar(max))
    else ',' + cast(port as varchar(max))
    end
from sys.dm_tcp_listener_states
where
type=0 -- Type 0 = TSQL (to avoid mirroring endpoint)
and
state=0 -- State 0 is online
and
port <> @dac_port -- Avoid DAC port (admin port)
and
-- Avoid availability group listeners
ip_address not in (select ip_address from sys.availability_group_listener_ip_addresses
agls)
group by port

```

```

order by port asc

print 'This SQL Server instance FQDN (Fully Qualified Domain Name) is '' +
@instance_fqdn + ''''
print 'This SQL Server instance port is ' + cast(@instance_port as varchar(10))

set @ror_url = 'tcp://' + @instance_fqdn + ':' + cast(@instance_port as varchar(10))

print
'*****
*****'
print 'The read_only_routing_url for this SQL Server instance is '' + @ror_url + ''''
print
'*****
*****'

-- If there is more than one instance port (unusual) list them out just in case.
if (charindex(',', @list_of_ports) > 0)
begin
print 'Note there is more than one instance port, the list of available instance
ports for read_only_routing_url is (' + @list_of_ports + ')'
print 'The above URL just uses the first port in the list, but you can use any of
these available ports.'
end

end -- if server supports read-only routing
go

```



Über den Autor

Holger Voges ist IT-Trainer und Consultant. Seine IT-Karriere begann mit einem Atari ST 512 Mitte der 80er Jahre. Seine ersten Erfahrungen mit großen Netzwerken hat er im Systembetrieb der Volkswagen Financial Services 1999 gewonnen. Ab dem Jahr 2000 war er dann als freiberuflicher IT-Trainer für verschiedene Schulungsunternehmen im Bereich Braunschweig und Hannover tätig, bevor er 2002 mit 2 Mitstreitern sein erstes Schulungsunternehmen LayerDrei in Braunschweig gegründet hat. Nach seinem Ausstieg bei LayerDrei war er dann mehrere Jahre als freiberuflicher Consultant vor allem im SQL-Server Umfeld u.a. für T-Home Entertain, e.on und

Hewlett-Packard unterwegs, bevor er 2012 das Schulungsunternehmen Netz-Weise gegründet hat.

Netz-Weise hat sich auf Firmenschulungen im professionellen IT-Umfeld spezialisiert und bietet Schulungen u.a. im Bereich Microsoft, VMWare, Linux und Oracle an.