Netz-Weise
Lernen von den Besten.

SQL Server Always On Availability Groups einrichten

von Holger Voges

Version 2.0

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

Inhalt

Grundlagen von SQL-Server Always On

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

Always On Failover Cluster (FCI)

Der Always On 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.

Always On Availability Groups (AG)

Always On Availability Groups sind eine in SQL-Server 2012 komplett neu eingeführte Technologie, die die Fähigkeiten der klassischen SQL-Server Datenbankspiegelung 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 (Kopien) einer Datenbank erstellt werden, ab SQL-Server 2014 sogar 8.

Der Hauptnachteil von Always On AG ist, dass die Technologie nur in der Enterprise-Edition des SQL-Servers zur Verfügung steht. Ab SQL-Server 2016 hat Microsoft die Datenbankspiegelung aber endgültig gegen die Basic AGs ersetzt, die wie die "richtigen" Availability Groups in Verbindung mit dem Failover-Cluster funktionieren, aber pro AG nur 1 Datenbank und eine Offline-Replika unterstützen.

Funktionsweise und Grundbegriffe

Always On 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 Always On 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 ein automatisches 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 Always On können gleichzeitig sowohl synchrone wie auch asynchrone Spiegel existieren.

Einrichten von Always On

Voraussetzungen

- Windows Server mit Failover-Cluster Funktionalität (bis Windows Server 2008R2 mind. Enterprise Edition, ab Server 2012 reicht die Standard-Edition)
- SQL-Server 2012 oder neuer (Enterprise-Edition) oder SQL-Server 2016 oder höher (Standard-Edition) für Basic-AGs
- Eine Windows Domäne
- Für einen produktiven Cluster sollte auf jeden Fall auf jedem Server (=Knoten) eine Netzwerkkarte installiert sein, die nur für den Clusterinternen Netzwerkverkehr genutzt wird.

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 Offline-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 auf allen Server bereits SQL-Server (Enterprise-Edition oder Developer) bereitgestellt wurde. Die Installation von SQL-Server wird unter https://www.netz-weise-it.training/images/dokus/SQL-Server%20installieren.pdf beschrieben.

Die Umgebung

Die Testumgebung umfasst 4 Server – einen Domänencontroller (Domäne: netz-weise.de) sowie 3 Servern mit Windows Server 2019, die Mitglied der Domäne sind und auf denen bereits SQL-Server 2019 Developer Edition installiert ist. Die SQL-Server verfügen über eine zusätzliche Netzwerkkarte für den Clusterinternen Netzwerkverkehr.

Einrichten der Netzwerkkarten

Zuerst wird der Netzwerkadapter für die Clusterinterne Kommunikation eingerichtet. Der Adapter stellt sicher, dass die Clusterknoten sich auch dann erreichen können, wenn die Netzwerkkarte für den öffentlichen Datenverkehr stark unter Last seht. Außerdem wird durch die alternative Netzwerkverbindung sichergestellt, dass der Ausfall eines Netzwerks (Clusterintern oder Öffentlich) nicht die gesamte Clusterkommunikation lahm legt.

Am schnellsten gelangt man in die Netzwerkkonfiguration über die Tastenkombination *Strg+R*, die das Ausführen-Fenster öffnet. Der Befehl *ncpa.cpl* öffnet dann die Netzwerkverbindungen. Benennen Sie zuerst die Netzwerkverbindungen um, um die Karten später besser zuordnen zu können.

Anschließend vergeben Sie für die Clusterinterne Kommunikation für alle Cluster-Knoten eine IP-Adresse aus einem privaten Bereich für die Cluster-Knoten. Lassen Sie DNS und Standard-Gateway leer. Anhand dieser Konfiguration erkennt das Cluster-Setup automatisch, welche Netzwerkkarte für den Clusterinternen Verkehr verwendet werden soll.

Achten Sie außerdem darauf, dass die Netzwerkkarte, auf dem der öffentliche Zugriff ein Standard-Gateway konfiguriert hat. Ohne Standardgateway bleibt der Installationsassistent beim Erstellen des Clusters hängen und die Installation bricht nach mehreren Minuten einfach ab.

Zur besseren Unterscheidung ist es sinnvoll, die Netzwerkkarten zu benennen wie in Abbildung 1, also z.B. Cluster für die Netzwerkkarte, über die nur der Clusterinterne Verkehr abläuft, und Public für die öffentlich erreichbare Netzwerkkarte.

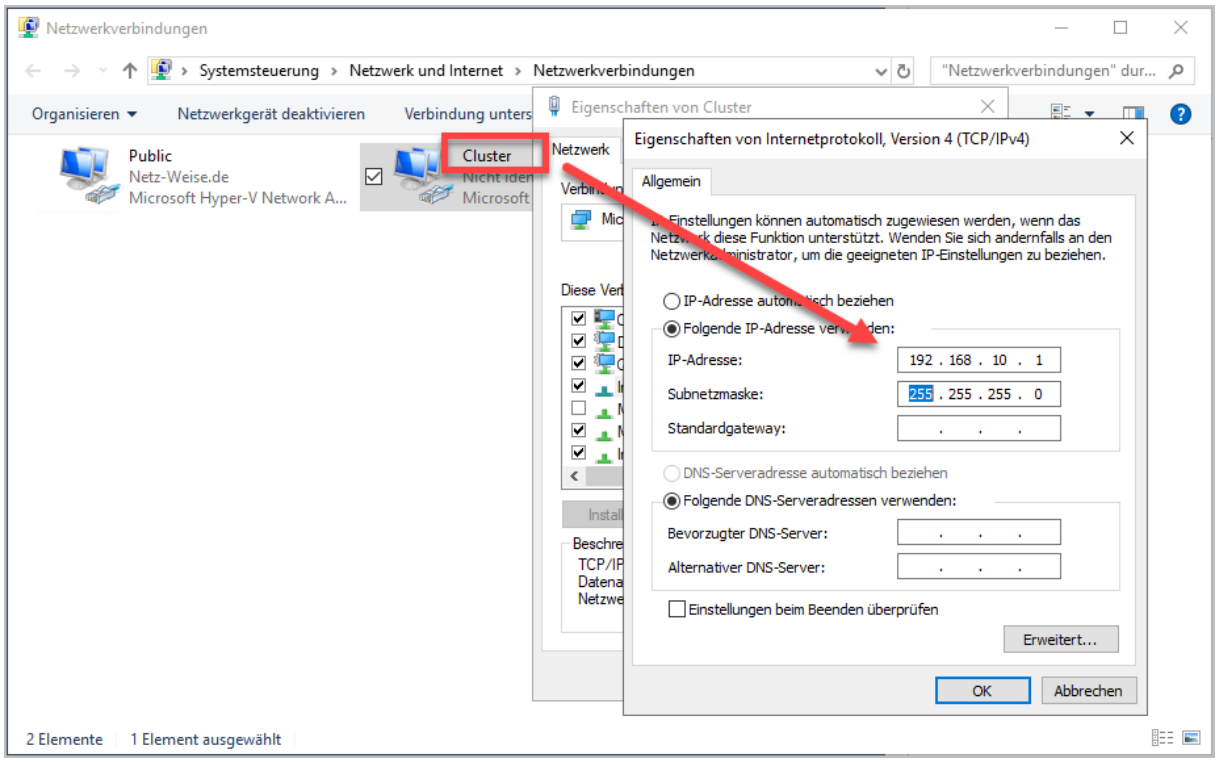


Abbildung 1 - Einrichten des Cluster-Netzwerks

Installieren des Windows Failover-Cluster-Features

Always On nutzt den Windows Failover Cluster 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 SQL-Servern*** zu installieren. Im Server Manager wählen Sie hierzu "Rollen und Features hinzufügen" und aktivieren das Failover-Cluster Feature:

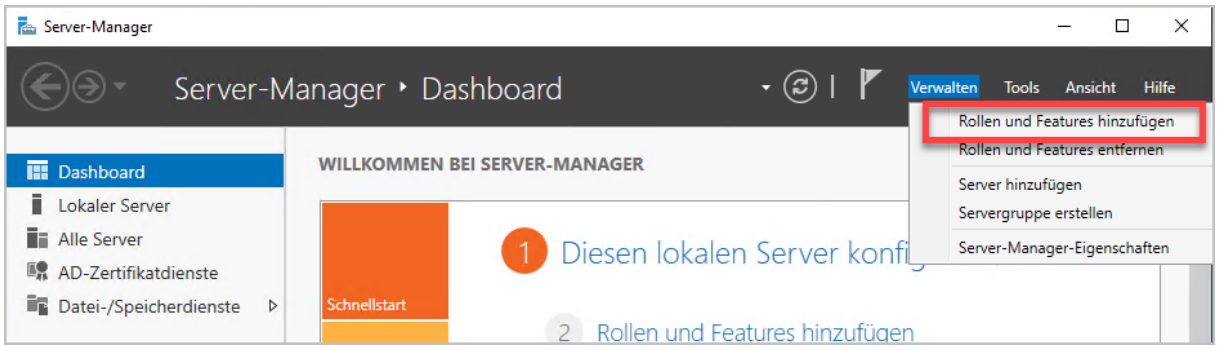


Abbildung 2 - Failover Cluster ist ein Feature

Klicken Sie sich durch den Assistenten bis zum Eintrag "Features" und wählen Sie "Failoverclustering" aus.

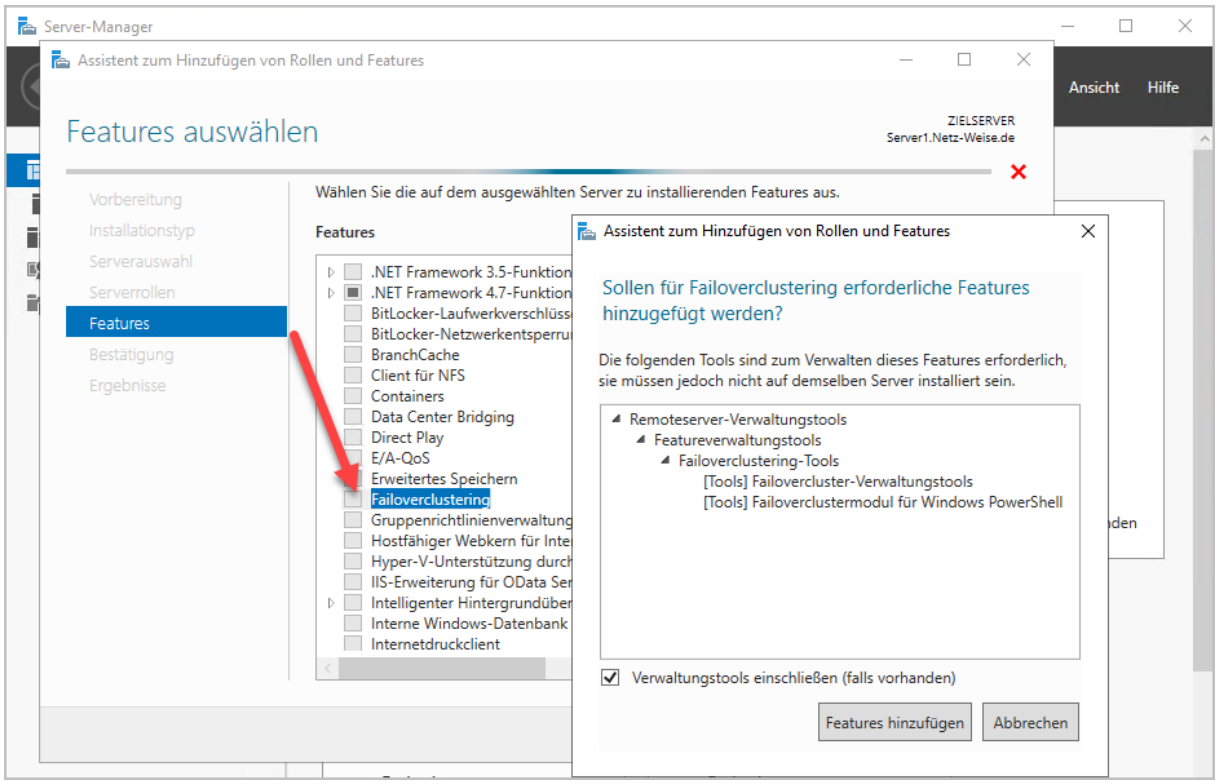


Abbildung 3 - wählen Sie das Feature aus und installieren Sie die Verwaltungstools mit

Beenden Sie den Assistenten, indem Sie "Weiter" und anschließend "Installieren" auswählen.

Alternativ können Sie auch Windows Powershell verwenden. Starten Sie dazu eine Powershell-Konsole mit administrativen Rechen und geben Sie folgendes Kommando (in einer Zeile) ein:

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

Mit Powershell können Sie das Feature sogar auf allen Servern in einem Rutsch installieren, soweit Powershell Remoting verfügbar ist und Ihr Benutzerkonto auf allen Servern administrative Rechte hat. Achten Sie darauf, dass Zeile 3 bis 5 (die geschweifte Klammer samt Inhalt) in einer Zeile eingegeben werden müssen.

```
$SqlServer = "SQL1","SQL2","SQL3"
foreach ($Server in $SqlServer)
{ Install-WindowsFeature -Name
Failover-Clustering -IncludeAllSubFeature -IncludeManagementTools -ComputerName
$Server }
```

Einrichten des Failover-Clusters

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 "Cluster erstellen".

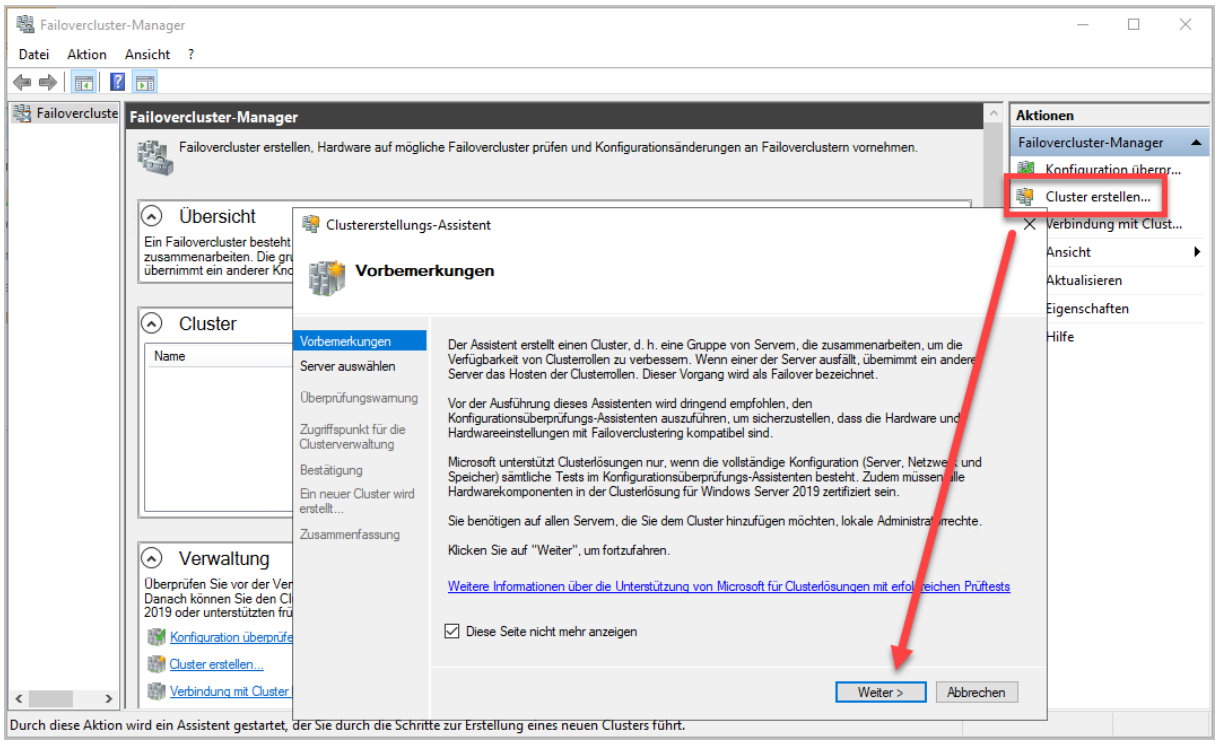


Abbildung 4 - Erstellen eines neuen Clusters über den Failovercluster-Manager

Wählen Sie im Menüpunkt "Server auswählen" die Server aus, auf denen Sie die Failover-Funktion installiert haben.

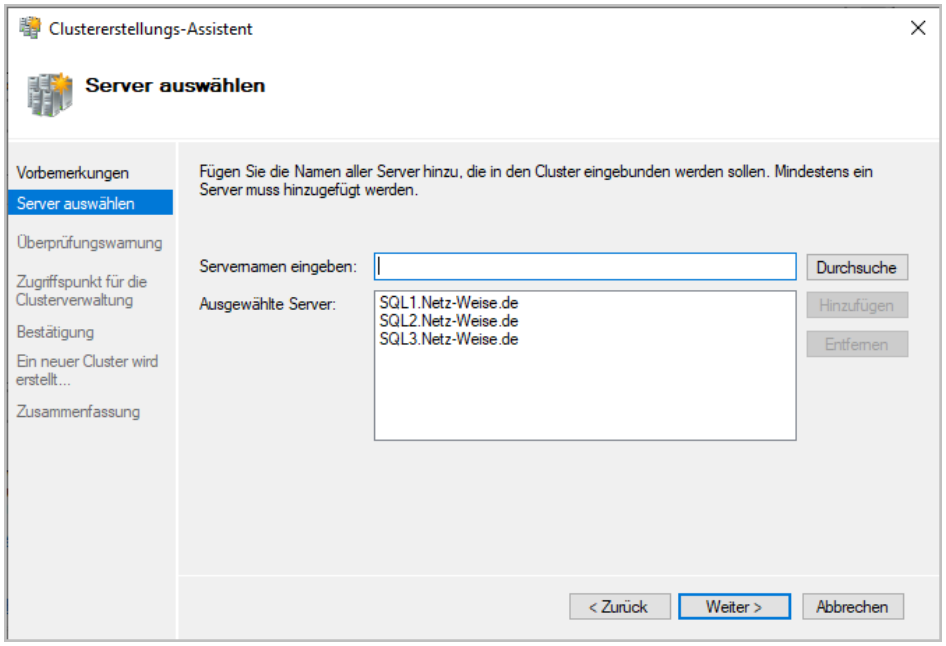


Abbildung 5 - Clusterknoten bestimmen

Seit Windows Server 2008 stellt der Clustererstellungs-Assistent sicher, dass die Server korrekt konfiguriert sind. Lassen Sie hier alle Tests durchlaufen. Dies stellt nicht nur sicher, dass alles richtig konfiguriert ist, sondern ist auch Voraussetzung dafür, dass Sie im Fehlerfall von Microsoft Support erhalten.

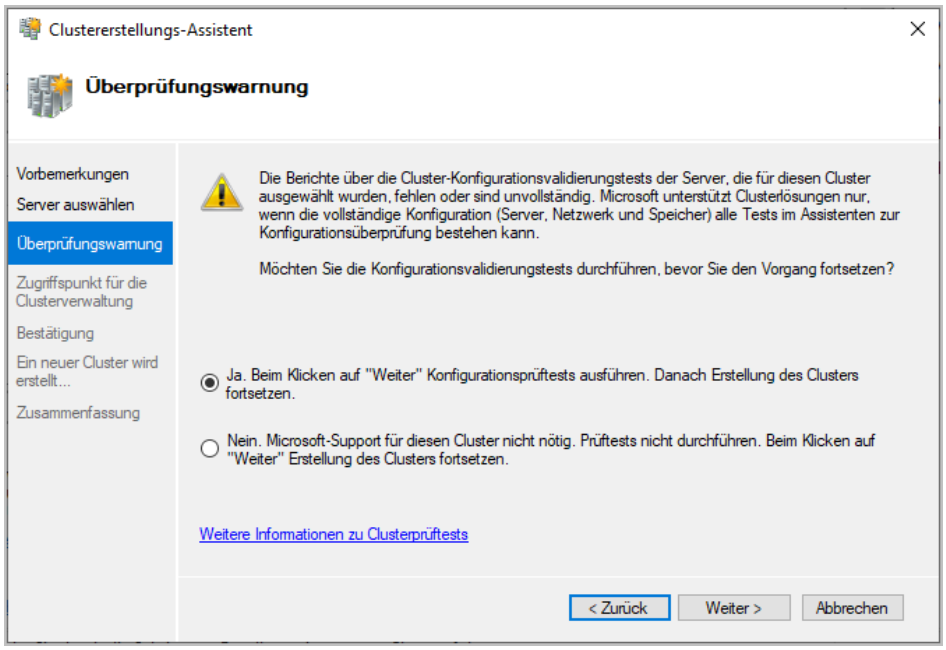


Abbildung 6 - Die Konfigurationsprüfung stellt sich, dass alles korrekt vorbereitet ist.

Der Konfigurationsprüfungs-Assistent ist ein eigenständiges Tool. Wählen Sie "Alle Tests ausführen" aus und starten den Test.

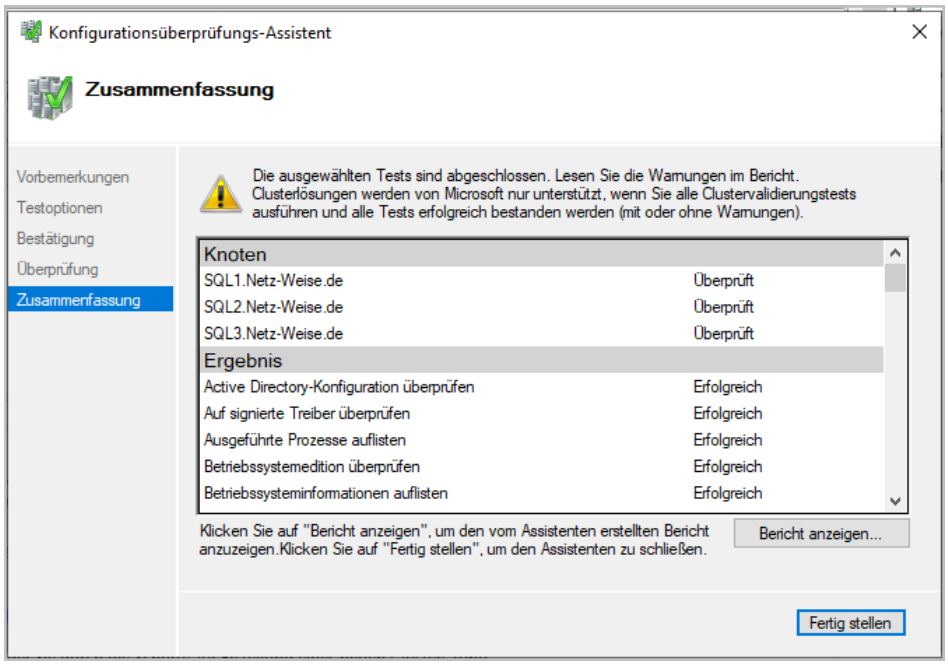


Abbildung 7 - Der Test wurde mit Warnungen abgeschlossen

Der Test wurde mit Warnungen beendet. Den vollständigen Report können Sie über "Bericht anzeigen" einsehen. Im Beispiel hat der Assistent festgestellt, dass die Server nicht auf dem gleichen Patch-Stand sind. Sie können das Problem beheben und die Server einmal durchstarten. Tun Sie das nicht, wird das Setup anschließend wegen eines ausstehenden Reboots abbrechen.

Der Konfigurationsprüfungs-Assistent wird nur beim ersten Durchlauf gestartet, es muss also keine weitere Prüfung ausgeführt werden.

Softwareupdates fehlen auf "SQL2.Netz-Weise.de":

KB-Artikel	Titel	Support	Sicherheitsbulletins
925673	MSXML 6.0 RTM Sicherheitsupdate (925673)	http://support.microsoft.com/kb/925673	MS06-061

Softwareupdates fehlen auf "SQL3.Netz-Weise.de":

KB-Artikel	Titel	Support	Sicherheitsbulletins
4598230	2021-01 Kumulatives Update für Windows Server 2019 (1809) für x64-basierte Systeme (KB4598230)	https://support.microsoft.com/help/4598230	
4598499	2021-01 Kumulatives Update für .NET Framework 3.5, 4.7.2 und 4.8 für Windows Server 2019 für x64 (KB4598499)	http://support.microsoft.com	
925673	MSXML 6.0 RTM Sicherheitsupdate (925673)	http://support.microsoft.com/kb/925673	MS06-061
4535680	Sicherheitsupdate für Windows Server 2019 für x64-basierte Systeme (KB4535680)	http://support.microsoft.com	

Der Clusterüberprüfungstest hat ergeben, dass nicht auf allen Knoten die gleichen Softwareupdates installiert sind. Es wird empfohlen, dass auf allen Knoten die gleiche Betriebssystemversion und die gleichen Softwareupdates installiert sind. Falls Sie überprüft haben, ob alle Knoten konsistent sind, können Sie die Warnung ignorieren.

Abbildung 8 - Der Test hat Warnungen ausgegeben

Die Beispiel-Server verfügen außerdem nicht über einen für alle Server erreichbaren Datenträger. Da wir einen sogenannten Hauptknoten-Cluster (Majority-Node Cluster) einrichten, ist dies auch nicht unbedingt notwendig. Beenden Sie die Validierung mit "Fertig stellen".

Unter "Zugriffspunkt für die Clusterverwaltung" geben Sie einen Namen für den Cluster ein. Sie verwenden diesen Namen später, um sich mit dem Cluster zu Konfigurationszwecken zu verbinden. Er entspricht nicht dem Namen, über den Sie später auf die Always On Availability Group zugreifen. Außerdem benötigen Sie eine eindeutige IP-Adresse.

Für den Zugriffspunkt wird bei der Installation ein Computerkonto im Active Directory eingerichtet.

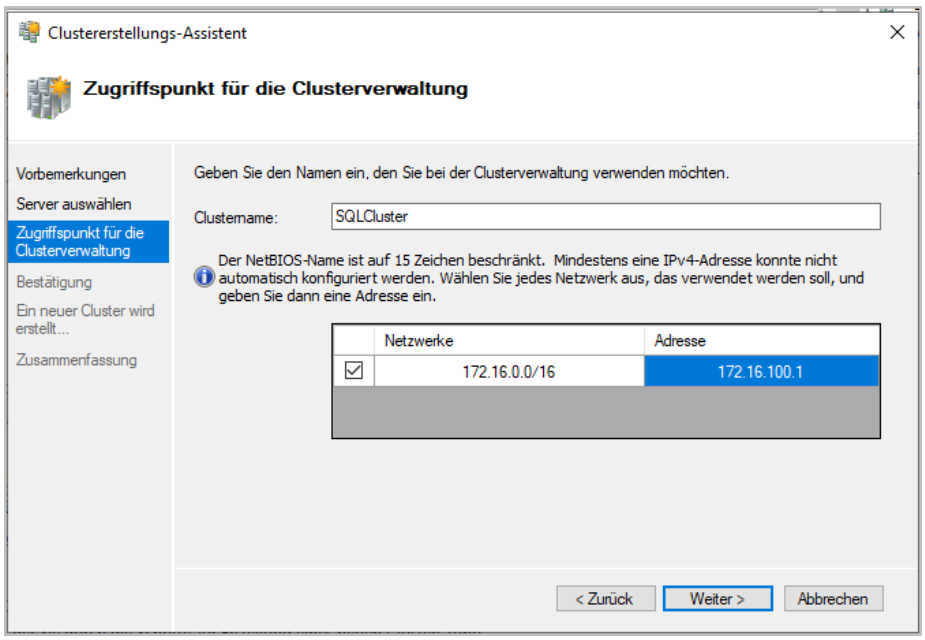
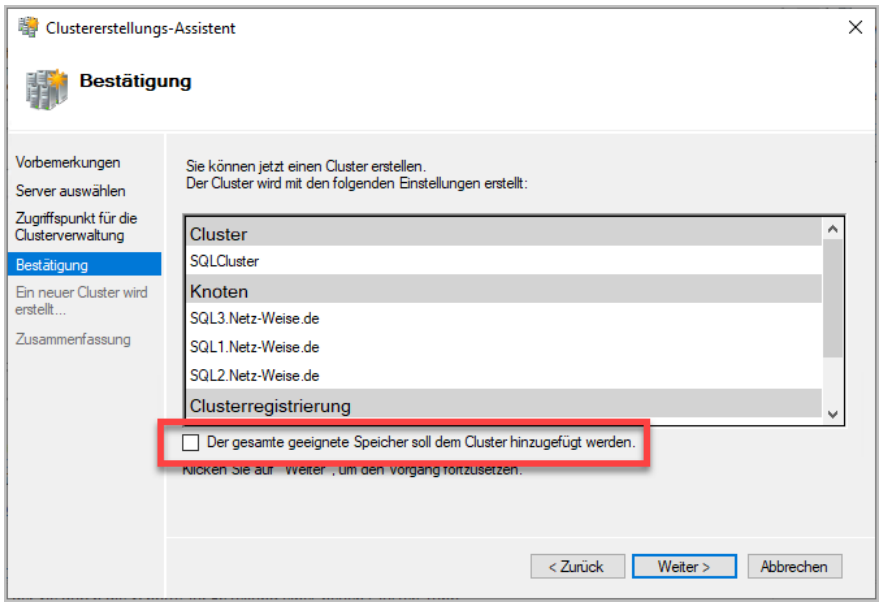


Abbildung 9 - die IP, über die der Cluster später zur Konfigurationszwecken angesprochen wird.

Wählen Sie Weiter und überprüfen Sie die Einstellungen.

In diesem Dialog finden Sie neben den getroffenen Konfiguration noch eine Checkbox "Den gesamten geeigneten Speicher soll dem Cluster hinzugefügt werden". Lassen Sie die Checkbox aktiviert, werden alle von allen Clusterknoten gleichzeitig sichtbaren Datenträger dem Cluster automatisch hinzugefügt. In einer normalen Serverkonfiguration gibt es solche Datenträger gar nicht. Sie müssen explizit über ein SAN mit iSCSI oder Fibrechannel oder per SAS direkt angebunden werden. Für Always On Availability Groups gibt es nur einen Grund, warum Sie gemeinsame Datenträger verwenden sollten – wenn Sie ein Quorum-Laufwerk für den Cluster verwenden wollen. Wenn Sie kein Quorum-Laufwerk eingebunden haben, deaktivieren Sie die Checkbox. Das ist zwar in den

meisten Fällen nicht unbedingt notwendig, aber Sie vermeiden damit Scherereien, sollten doch Datenträger über alle Clusterknoten hinweg sichtbar sein.



Schließen Sie die Installation ab, indem Sie "Weiter" auswählen. Der Cluster wird nun eingerichtet.

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

Cluster-Quorum

Eine der Funktionen des Clusters ist es, ausgefallene Server zu erkennen und die Dienste dann auf einem anderen Knoten zur Verfügung zu stellen – man spricht von einem Failover. Die Prüfung der Verfügbarkeit findet dabei über das Netzwerk statt, indem ein Heartbeat zwischen den Clusterknoten ausgeführt wird.

Wenn sich in einem Cluster eine gerade Anzahl von Servern befindet, kann es dabei zu einer Situation kommen, die als *Split-Brain* bezeichnet wird. Wird nämlich die Netzwerkkommunikation zwischen zwei gleich großen Gruppen von Cluster-Servern unterbrochen, kann keine der beiden Gruppen entscheiden, ob die jeweils andere Gruppe tatsächlich ausgefallen oder nur gerade nicht erreichbar ist – die beiden Gehirnhälften des Clusters sind getrennt. Damit nicht beide Clustergruppen einfach die "ausgefallenen" Datenbanken der anderen Gruppe schreibend verfügbar machen, müssen beide Clustergruppen alle hochverfügbaren Datenbanken herunterfahren.

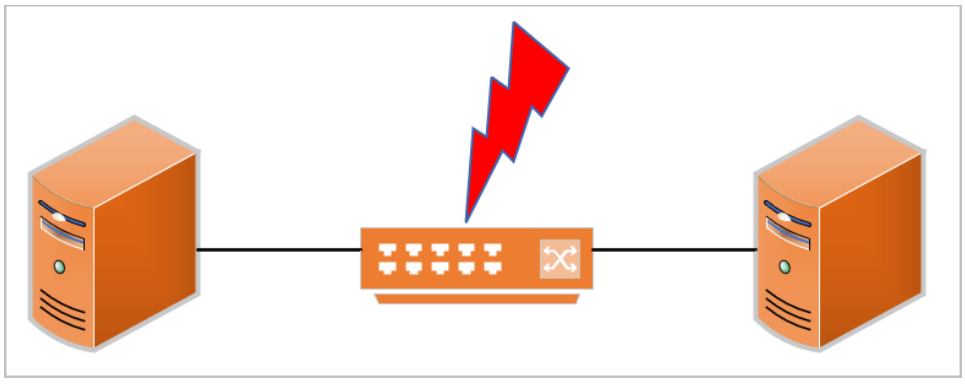


Abbildung 10 - Split Brain Szenario - Die Netzwerkkverbindung zwischen den zwei Knoten eines Cluter fällt aus

Um das zu verhindern, muss man dafür sorgen, dass bei einer geraden Anzahl von Knoten immer eine Mehrheit – ein Quorum – erreicht werden kann. Der Cluster verwendet hierzu einen Zeugen oder Witness. Der Zeuge wird auch als Tie-Breaker bezeichnet. Es gibt 3 Typen von Zeugen:

- Disk-Witness
- Fileshare-Witness
- Cloud-Witness

Außerdem kann man einen Cluster komplett ohne Zeugen betreiben, indem man eine ungerade Anzahl von Knoten verwendet. Man spricht dann von einem Hauptknoten-Cluster.

Die Funktion von allen 3 Zeugen ist prinzipiell gleich. Beim Disk-Witness wird eine Festplatte für alle Clusterknoten verfügbar gemacht. Das geht entweder per SAN (iSCSI oder Fibre-Channel), oder indem man eine physikalische Festplatte per SAS (Serial Attached SCSI) extern an alle Server anschließt. Einer der Server im Cluster hat diese Festplatte im Zugriff, während alle anderen Clusterknoten die Platte nicht verwenden können.

Der Fileshare-Witness funktioniert ähnlich, nur wird hier keine gemeinsame Festplatte verwendet, sondern eine Datei in einer Freigabe. Der Fileserver erlaubt den Zugriff immer nur für den Server, der die Datei geöffnet hat. Achten Sie darauf, dass auch der Fileserver selbst hochverfügbar sein sollte, weil sonst bei jeder Wartung des Servers der Cluster kein Quorum mehr hat.

Ein Cloud-Witness schließlich ist einfach nur ein Fileshare-Witness, bei dem die Freigabe durch einen Storage-Account in einem Azure-Rechenzentrum bereitgestellt wird – man benötigt also keinen eigenen File-Server.

Fällt ein Heartbeat zu einem Server im Cluster mehr als 10-mal aus, wird im Cluster eine Abstimmung gestartet. Dabei hat im Normalfall jeder Clusterknoten eine Stimme, plus einer Stimme für den Witness. Werden bei der Abstimmung mehr als 50% der Stimmen erreicht, wird der nicht mehr erreichbare Clusterknoten aus dem Cluster entfernt. Werden 50% oder weniger der Stimmen erreicht, werden die vom Cluster zur Verfügung gestellten Dienste beendet, bis wieder eine Mehrheit erreicht werden kann.

Bei einer ungeraden Anzahl von Clusterknoten kommt man niemals auf genau 50% der Stimmen. Daher braucht ein Majority-Node Cluster (ungerade Anzahl von Knoten) auch keinen Witness – er ist hier sogar kontraproduktiv. Seit Windows Server 2012 ist das Quorum normalerweise dynamisch – die Zeugeninstanz wird automatisch dem Cluster hinzugefügt oder entfernt, je nachdem, ob die Anzahl der Server gerade (Zeuge ist an) oder ungerade ist (Zeuge ist deaktiviert).

Die Quorum-Konfiguration kann für einen bestehenden Cluster jederzeit mit ein paar Mausklicks angepasst werden. Erweiterte Einstellungsmöglichkeiten stehen Ihnen außerdem per Powershell zur Verfügung.

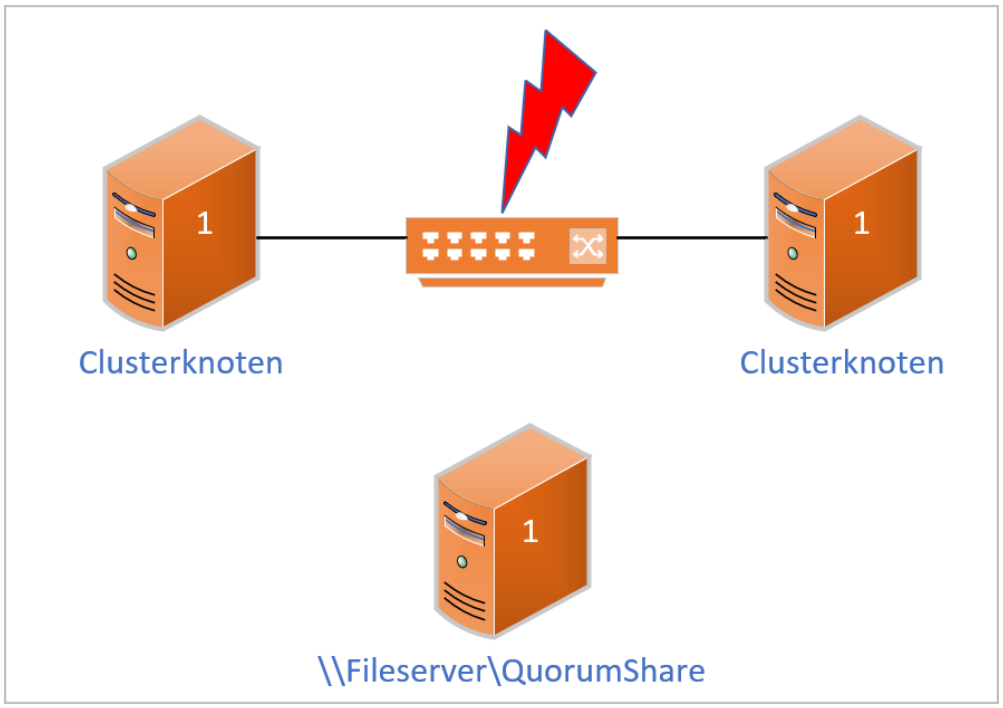


Abbildung 11 - Der Server, der die Datei im Quorum-Share im Zugriff hat, hat 2/3 der Stimmen

Active Directory anpassen

Nach der Erstellung können Sie den Cluster nun über die IP-Adresse und den Cluster-Namen im Netzwerk erreichen. Außerdem wurde ein Computer-Konto im Active Directory erstellt.

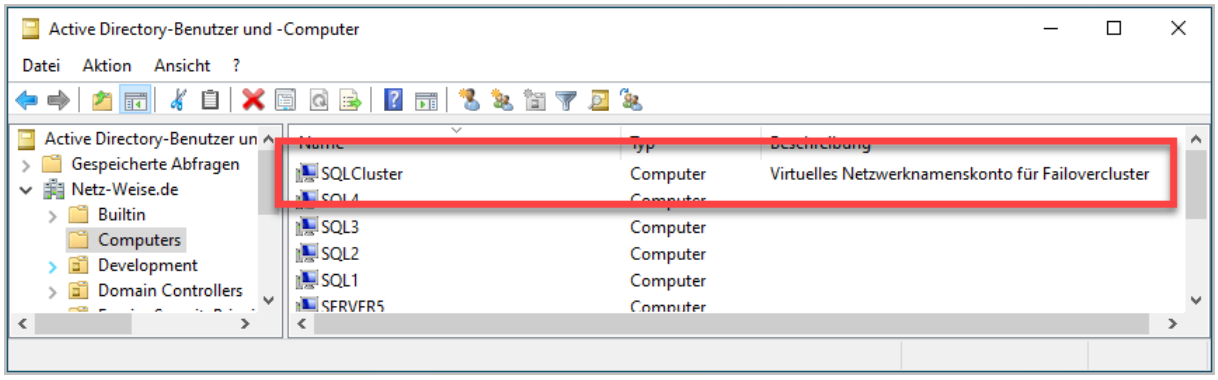


Abbildung 12 - Für den Verwaltungsknoten wurde ein Konto im AD erzeugt

Wenn Sie den Cluster nicht im Container Computers belassen, müssen Sie ihn im AD für das Anlegen neuer Computer-Konten berechtigen, da mit dem Cluster-Konto für jede Availability Group ein neues Computerkonto erstellt wird. Am besten erledigen Sie das auf der Organisationseinheit, in der sich das Computer-Konto des Clusters befindet. Alternativ können Sie die Berechtigungen direkt auf der Domäne vergeben, wenn Sie planen, das Clusterkonto später zu verschieben. Starten Sie hierfür Active Directory Benutzer und Computer und überprüfen im Menü "Ansicht", ob die erweiterte Ansicht aktiviert ist. Anschließen öffnen Sie die Eigenschaften der OU, in der sich das Clusterkonto befindet, aus dem Kontextmenü.

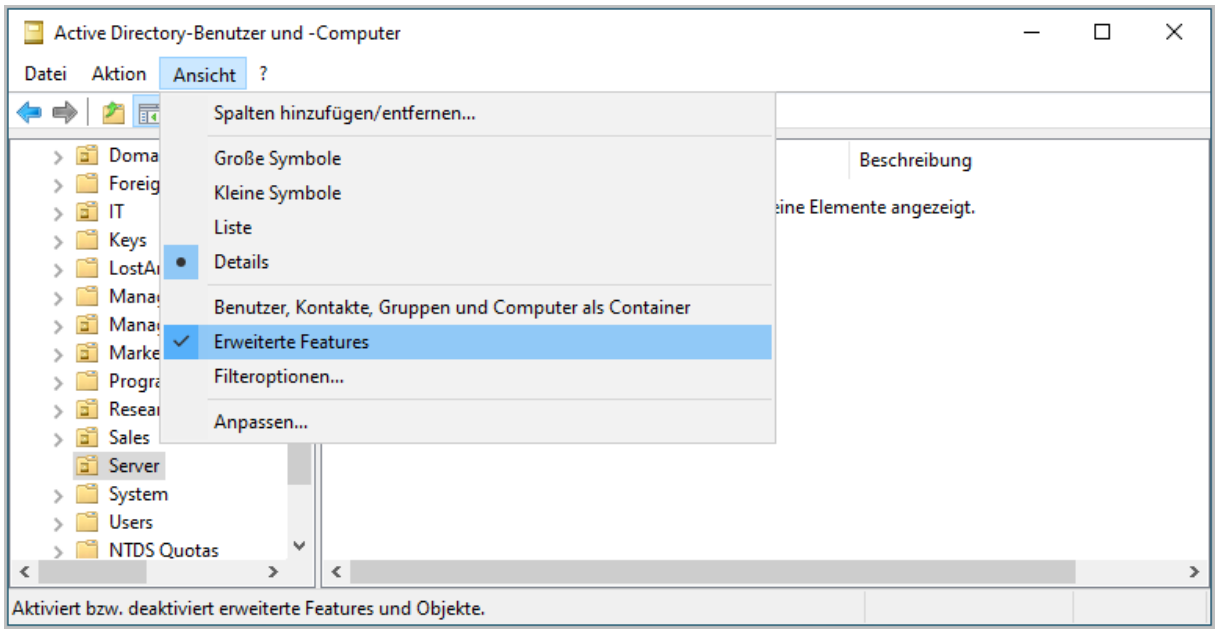


Abbildung 13 - Aktivieren Sie die erweiterten Featues, damit AD Benutzer und Computer Berechtigungen anzeigt

In den Eigenschaften wählen Sie auf der Registerkarte "Security" das erweiterte Menü aus.

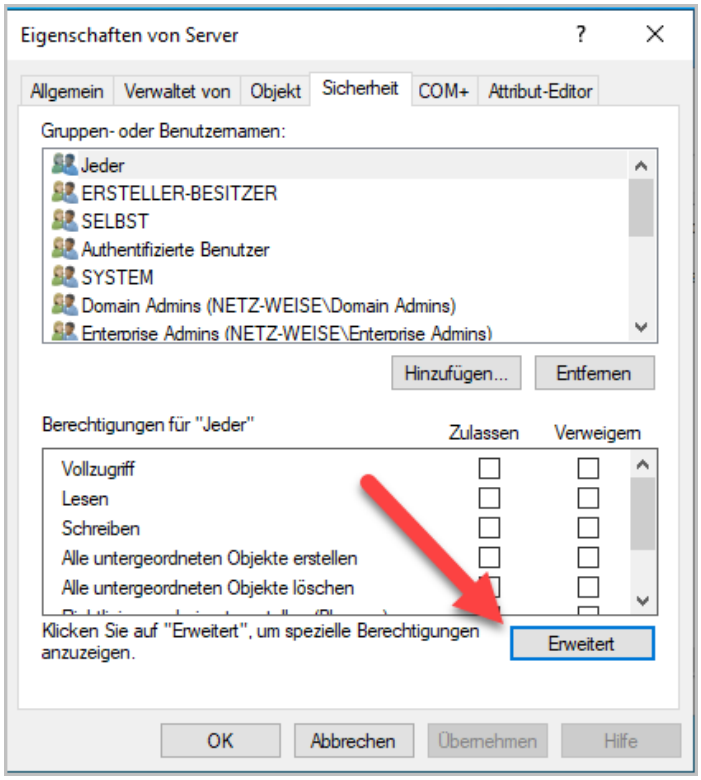


Abbildung 14 - Verwenden Sie die erweiterten Sicherheitseinstellungen zum Hinzufügen des Cluster-Kontos

Fügen Sie in den erweiterten Sicherheitseinstellungen das Computer-Konto des Clusters hinzu, indem Sie "Hinzufügen" auswählen.

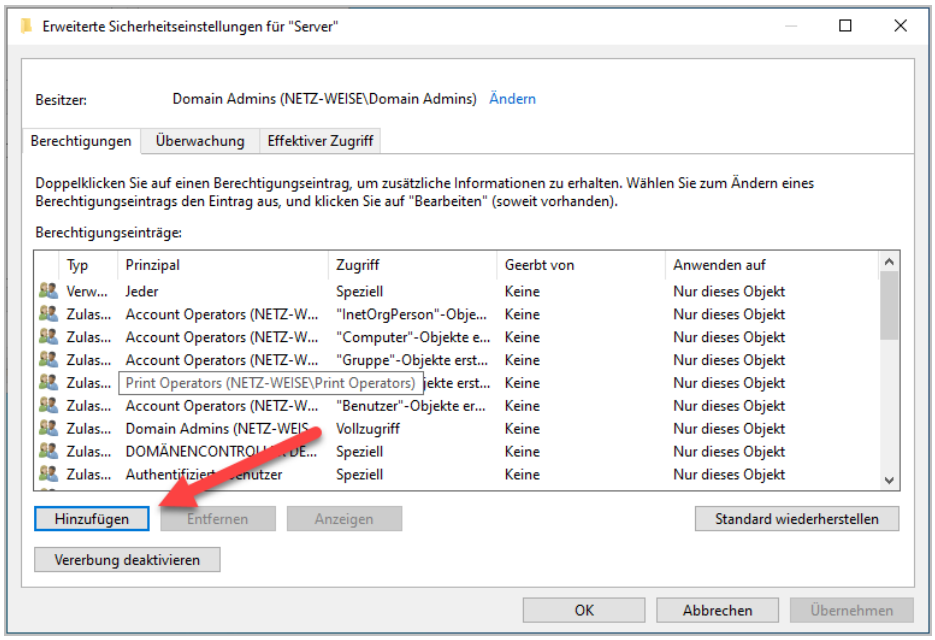


Abbildung 15 - Über "Hinzufügen" berechtigen Sie weitere Benutzer

Im Berechtigungsfenster müssen Sie zuerst das zu berechtigende Konto angeben. Wählen Sie hierfür "Prinzipal auswählen".

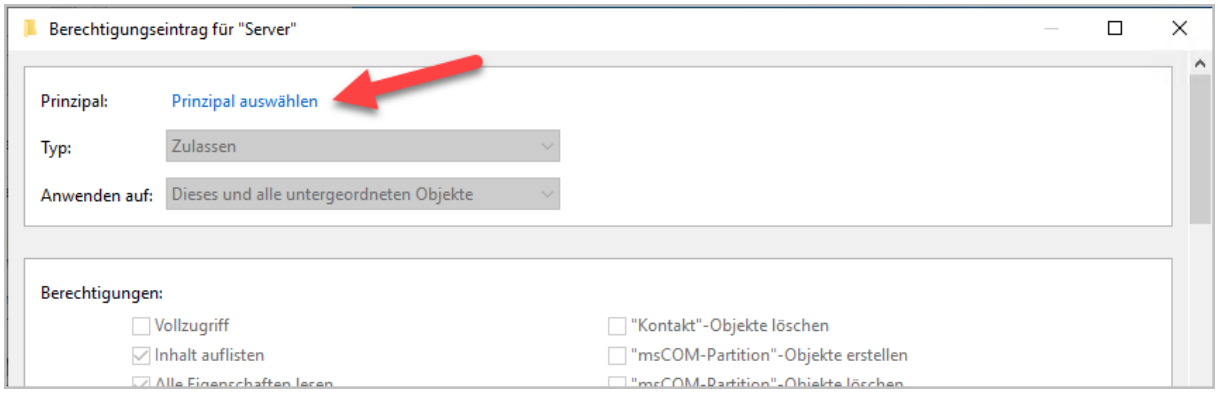


Abbildung 16 - Wählen Sie den Prinzipal aus, also das zu berechtigende Konto

Computerkonten werden standardmäßig nicht angezeigt. Passen Sie den Filter über den Button "Objekttypen" an (s. Abbildung 18 (1)).

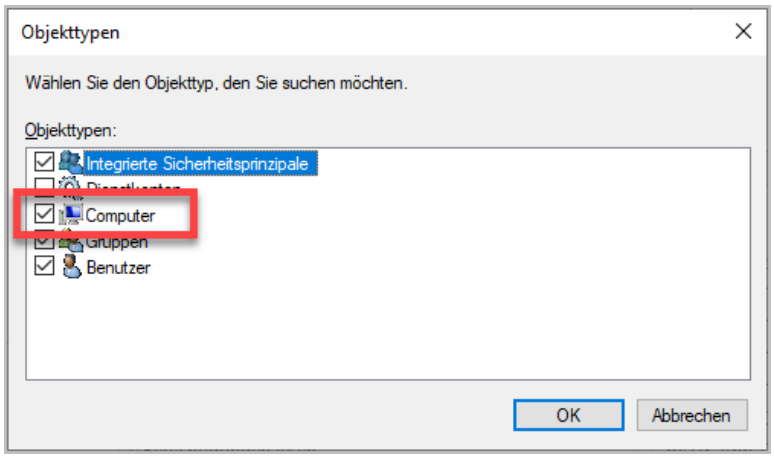


Abbildung 17 - Nehmen Sie Computerkonten in den Anzeigefilter mit auf

Anschließend geben Sie den Namen des Clusterkontos an, prüfen den Namen (2) und übernehmen die Einstellungen.

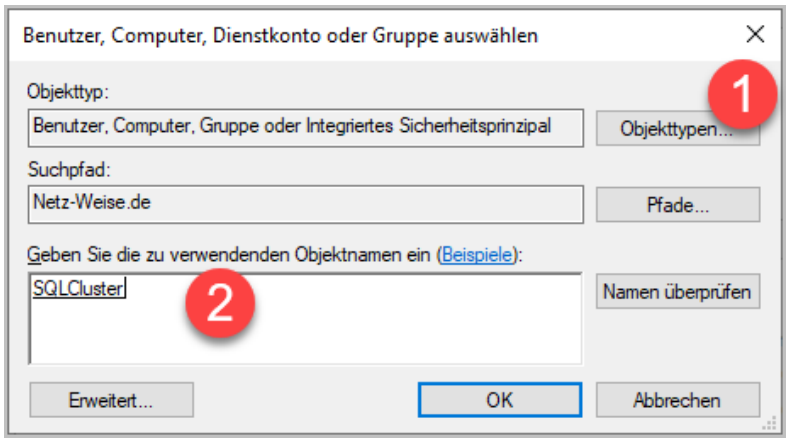


Abbildung 18 - Geben Sie den Namen des Clusterkontos an und prüfen Sie den Namen

In der Berechtigungs-Liste vergeben Sie jetzt die Berechtigungen "Computer-Objekte erstellen" und "Computer-Objekte löschen". Achten Sie darauf, dass Sie unter "Anwenden auf" "Dieses und alle untergeordneten Objekte" aktivieren. Diese Einstellung sorgt dafür, dass die Berechtigungen auch auf untergeordnete Organisationseinheiten vererbt werden.

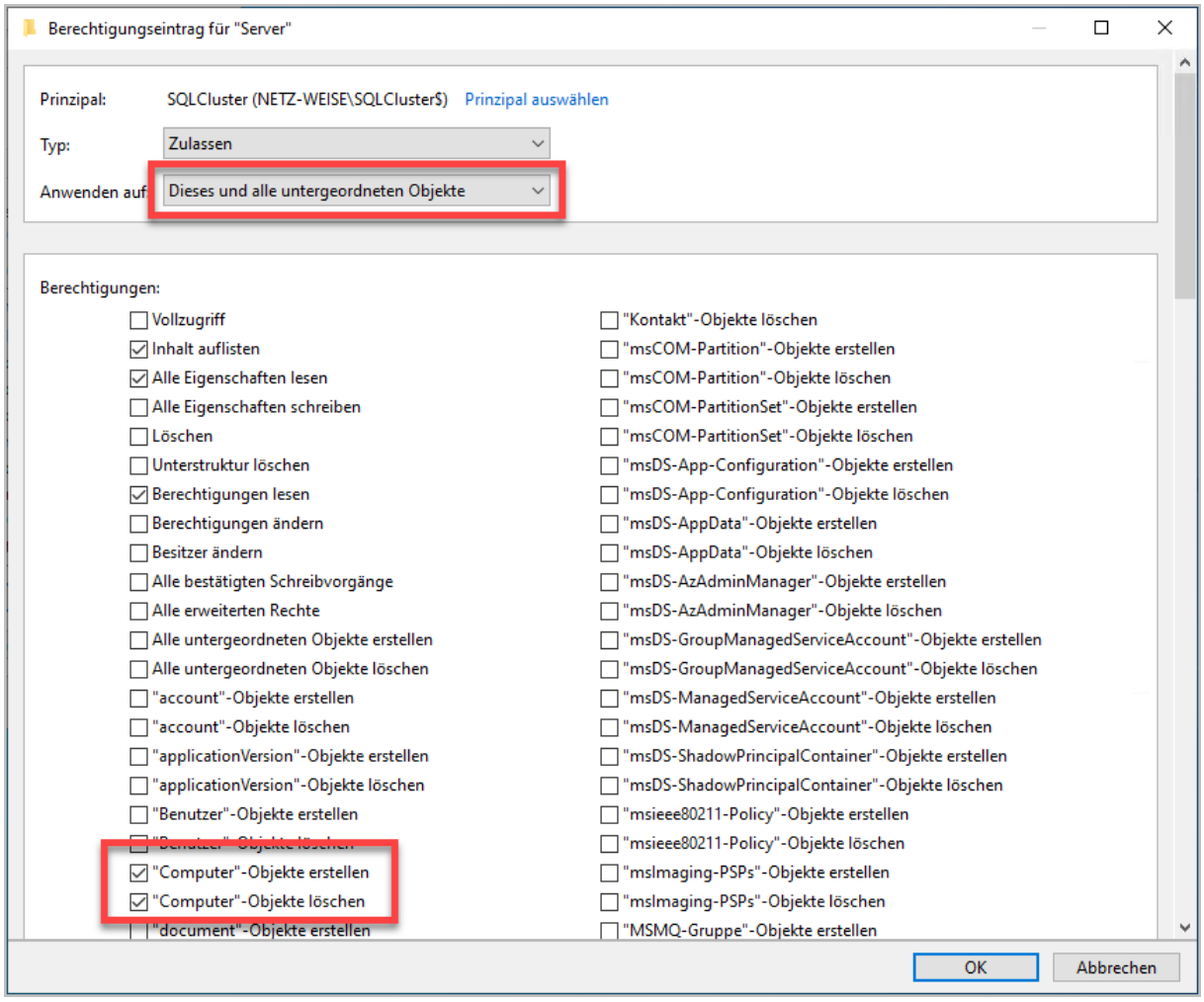


Abbildung 19 - Vergeben Sie das Recht zum Erstellen und Löschen von Computer-Konten

Konfigurieren des SQL-Server Dienstes für die Nutzung von Always On Availability Groups

Die Nutzung von Always On Availability Groups muss auf dem SQL-Server-Dienst aktiviert werden. Starten Sie hierzu den SQL-Server Configuration Manager auf allen Clusterknoten, öffnen Sie unter "SQL Server Services" die Eigenschaften des SQL Server Dienstes und aktivieren Sie die Checkbox Always On Availability Groups auf der Registerkarte "AlwaysOn High Availability".

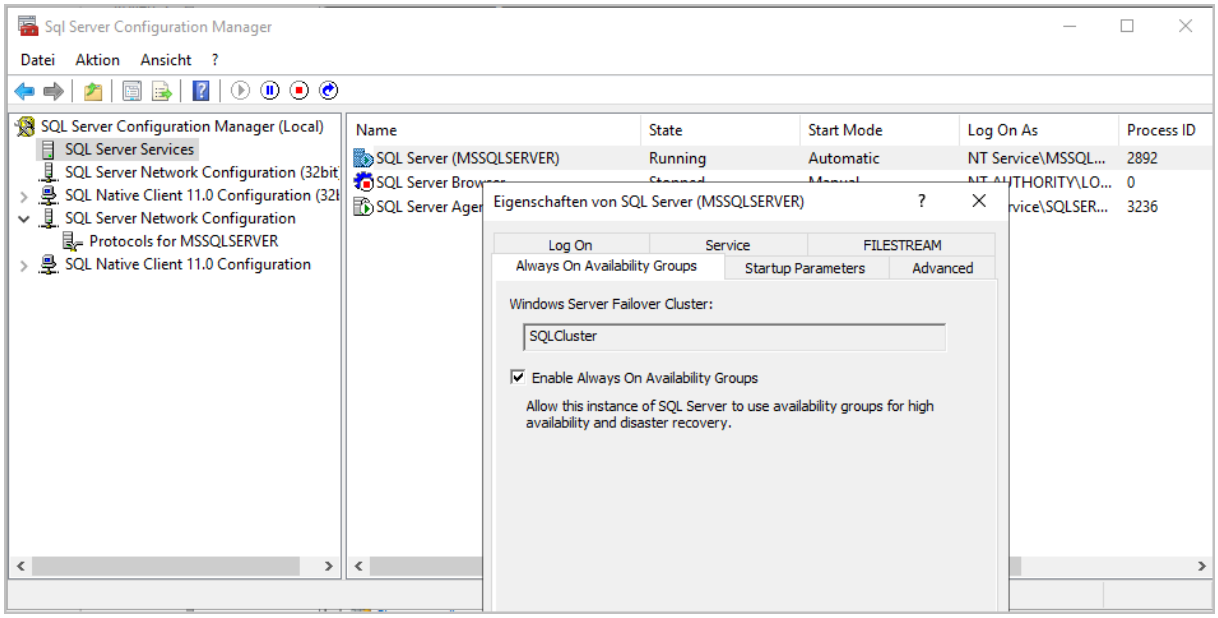


Abbildung 20 - Aktivieren Sie Always On Availability Groups auf dem SQL-Server Dienst

Alternativ können Sie Always On Availability Groups auch per Powershell einschalten.

```
Enable-SqlAlways On -Path sqlserver:\sql\sql3\default
```

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

Aktivieren des TCP-Protokolls auf der SQL-Server Developer-Edition

Wenn Sie eine Developer-Edition für die Installation des SQL-Server verwendet haben, müssen Sie das TCP/IP-Protokoll für den Netzwerkzugriff noch aktivieren. Wechseln Sie dafür im SQL-Server Configuration Manager auf den Knoten "SQL Server Network Configuration" – "Protocols for SQL-Server" und öffnen Sie die Eigenschaften von "TCP/IP".

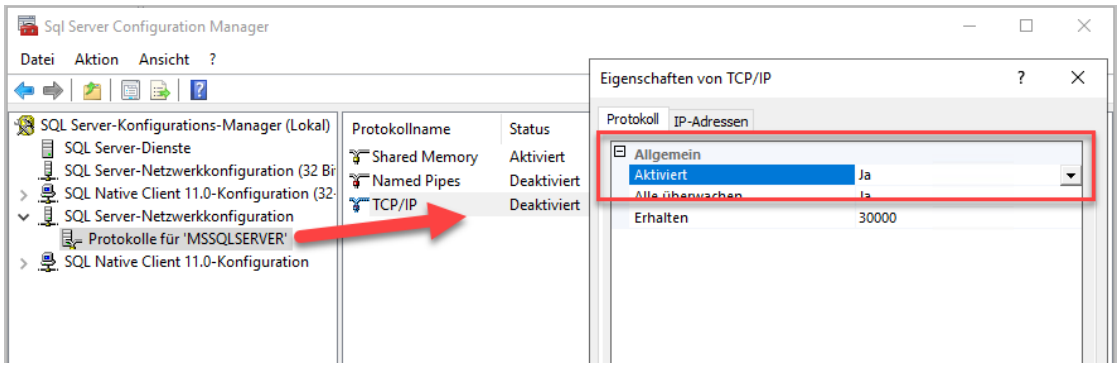


Abbildung 21 - TCP/IP für den Tabular Data Stream (TDS) aktivieren

Hier müssen Sie den Eintrag "Aktiviert" auf "Ja" umstellen. Anschließend starten Sie den Dienst neu.

Die Windows Firewall anpassen

Der SQL-Server verwendet standardmäßig Port TCP 1433, wenn Sie eine Standard-Instanz installiert haben. Den Port der Instanz können Sie in den "Eigenschaften von TCP/IP" (s. Abbildung 21) auf der Registerkarte IP-Adressen anpassen.

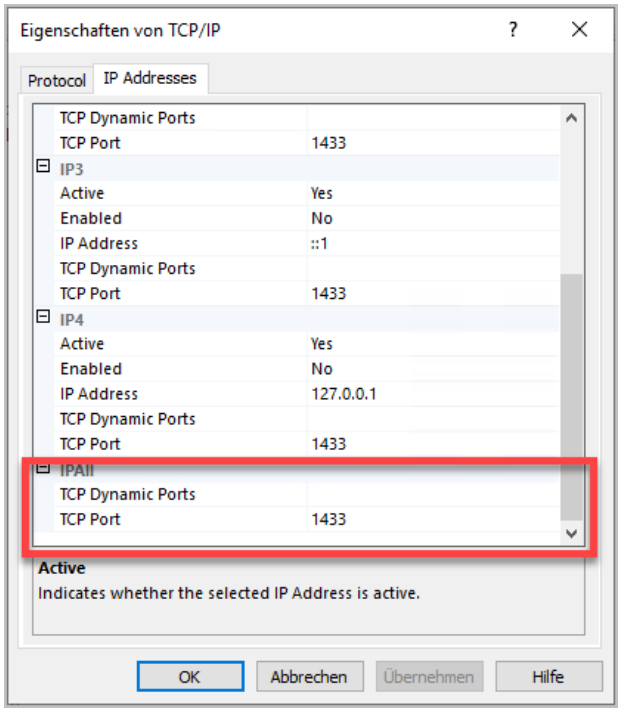


Abbildung 22 - Der Port kann frei festgelegt werden - Standard ist 1433

Für benannte Instanzen ist normalerweise "dynamischer Port" eingestellt. Das bedeutet, dass der SQL-Server bei jedem Start einen freien Port sucht und diesen dann bis zum nächsten Neustart benutzt. Sind dynamische Ports aktiviert, steht der Eintrag "TCP Dynamic Ports" auf 0 und "TCP Port" ist leer. Um eine Instanz auf einen festen Port festzulegen, tragen Sie unter TCP-Port einfach den gewünschten Port ein und entfernen Sie die 0 im Feld "TCP Dynamic Ports".

Die SQL-Server Ports werden in der Windows Firewall nicht automatisch freigegeben. Um eine Ausnahme für eingehenden Datenverkehr zu konfigurieren, öffnen Sie die Windows Defender Firewall und wechseln in die erweiterten Einstellungen.

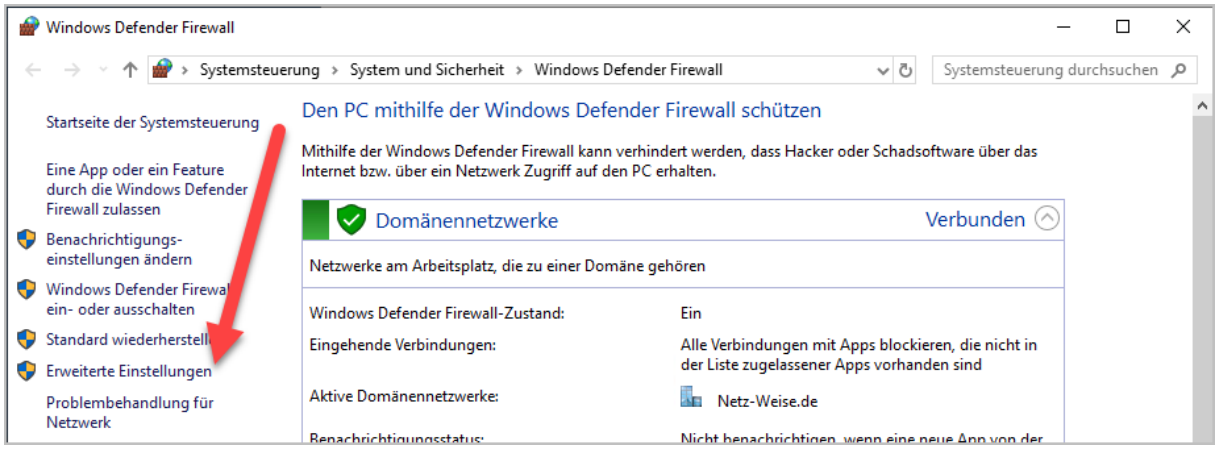


Abbildung 23 - Wechseln Sie in die erweiterten Einstellungen

Anschließend legen Sie neue Firewall-Regeln an. Zuerst definieren Sie, ob die Regel für ein Programm für einen Port oder ein Programm gelten soll. Im folgenden Beispiel wird der Port 1433 verwendet. Wenn Sie eine benannte Instanz mit dynamischen Portregeln freigeben wollen, müssen Sie die sqlserver.exe freigeben. Achten Sie dabei darauf, dass Sie für jede benannte Instanz eine eigene sqlserver.exe haben.

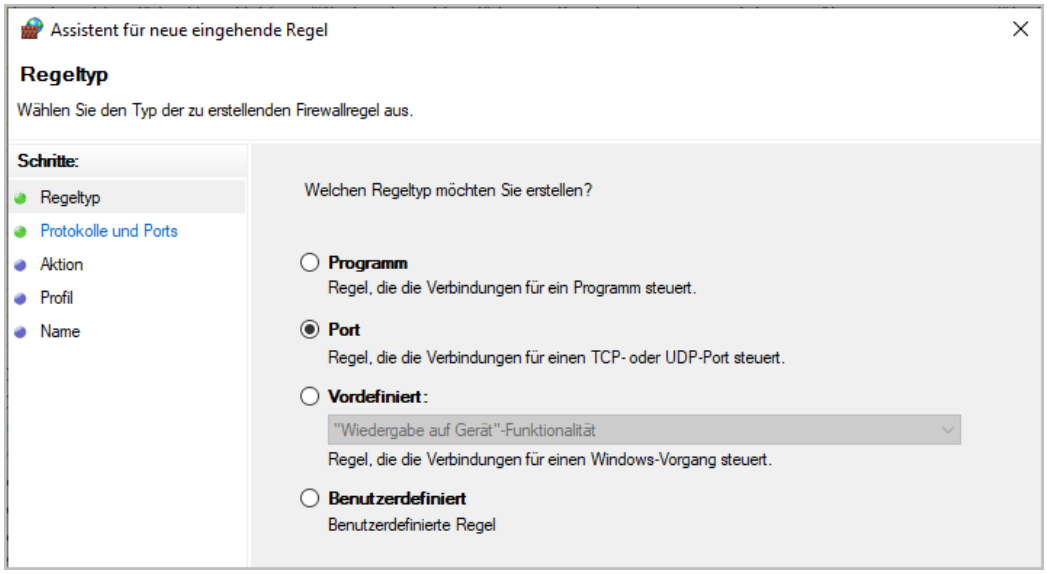


Abbildung 24 - Legen Sie fest, ob die Ausnahme für ein Programm oder einen Port definiert werden soll

Im nächsten Schritt legen Sie den Port und das Protokoll fest.

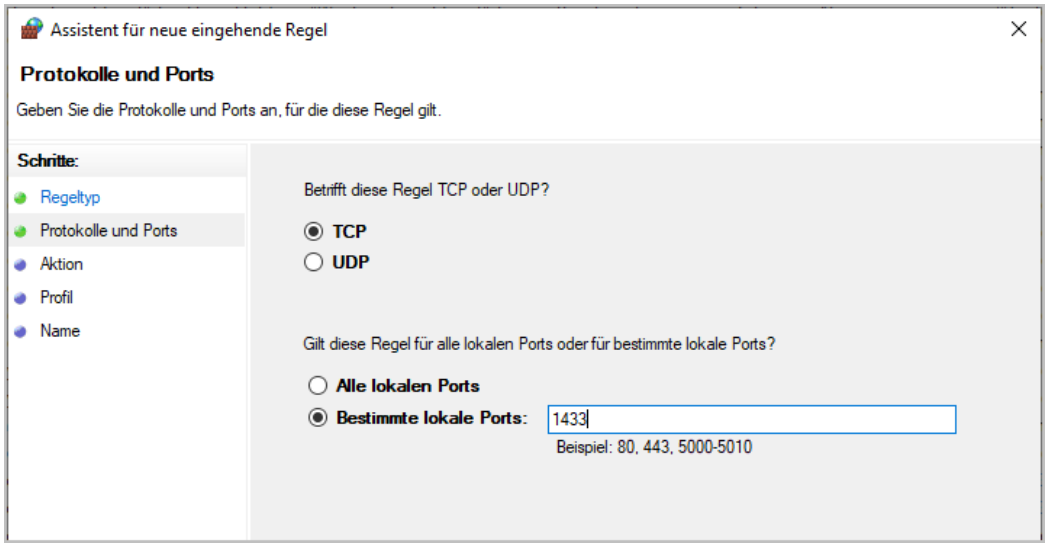


Abbildung 25 - Legen Sie den Port fest, der zugelassen sein soll

Lassen Sie den Zugriff über den Port zu.

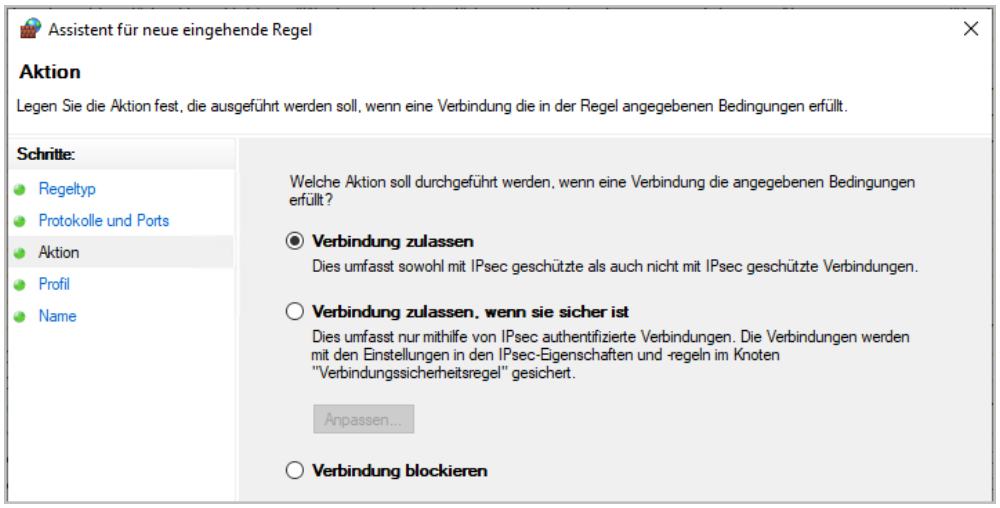


Abbildung 26 - erlauben Sie den Zugriff

Im letzten Schritt geben Sie an, für welche Netzwerkprofile die Regel gelten soll. Da Ihr Server normalerweise in einer geschützten Domänen-Umgebung steht, können Sie alle Profile einschalten.

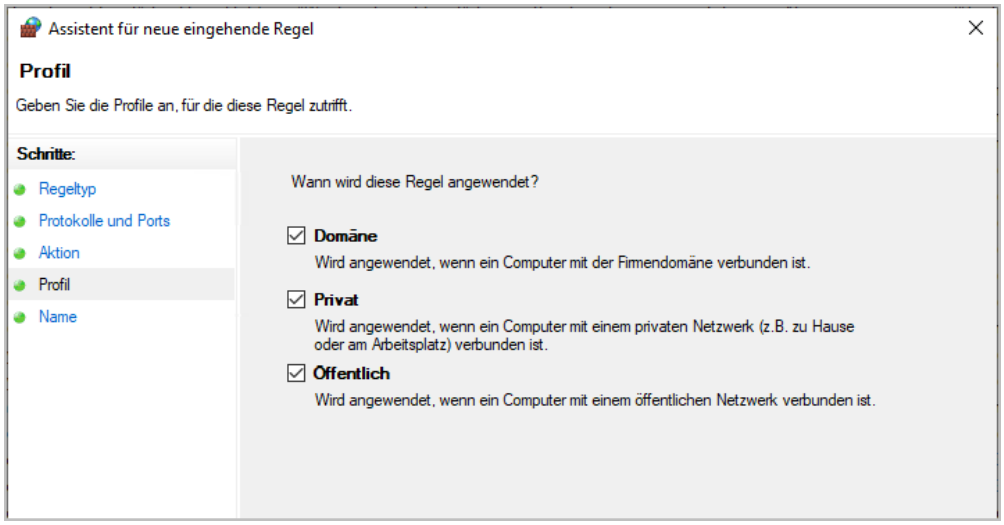


Abbildung 27 - Die Profile entsprechen unterschiedlich restriktiven Regelsätzen

Nun müssen Sie Ihre Regel noch benennen. Sobald die Regel gespeichert ist, ist der Port über das Netzwerk erreichbar.

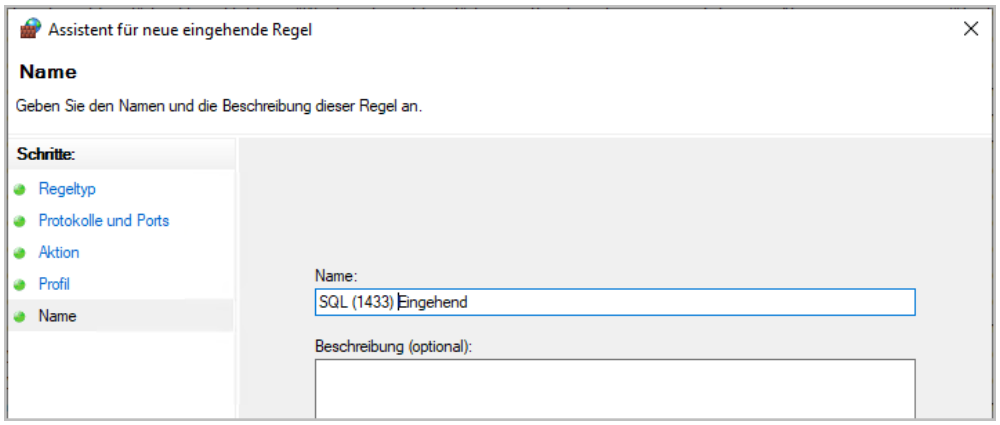


Abbildung 28 - Geben Sie Ihrer Regel einen informativen Namen

Den Spiegelungs-Endpunkt in der Firewall freigeben

Neben Port TCP 1433 benötigen Sie noch eine zweite Firewallregel für den Port TCP 5022. Dieser Port wird für die Datenbankspiegelung, also die Replikation der Datenbankänderungen verwendet.

Legen Sie als eigehenden Port TCP 5022 fest.

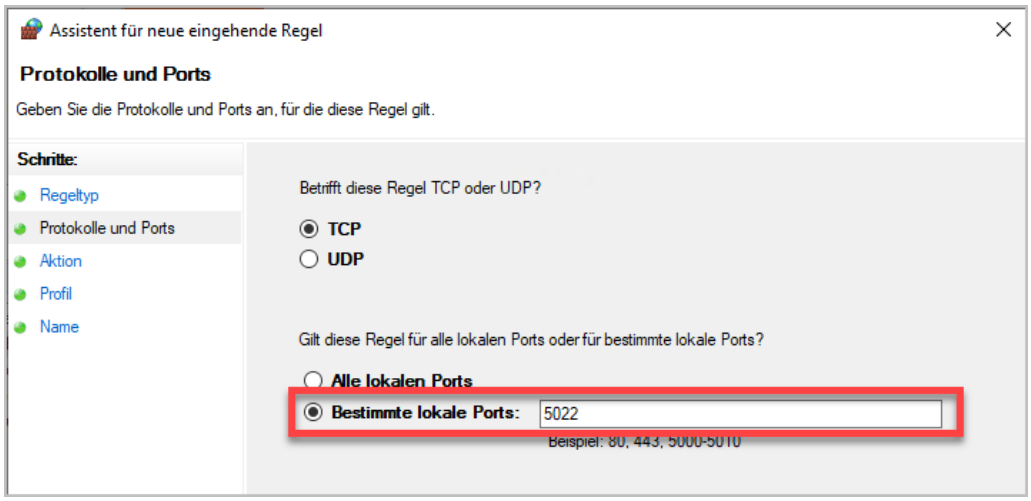
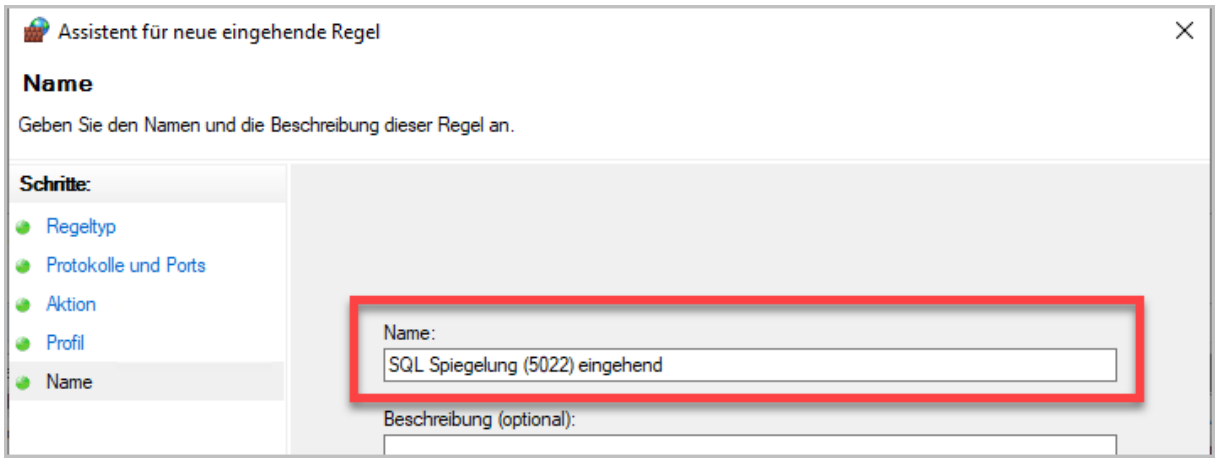


Abbildung 29 - Port 5022 wird für die Datenbankspiegelung verwendet

Erlauben Sie den eingehenden Zugriff auf den Port in allen Firewall-Profilen und vergeben Sie anschließend einen sprechenden Namen.



Group Managed Service Accounts einrichten

Der SQL-Server-Dienst muss im Kontext eines Active Directory Dienstkontos gestartet werden. Das grundsätzliche Problem mit Dienstkonten ist, dass Sie verwaltet werden müssen. Ein Dienstkonto kann gesperrt werden, wenn das Kennwort zu oft falsch eingegeben wurde, und ein Dienstkonto ist ein Sicherheitsproblem, weil das Kennwort niemals geändert wird. Als Lösung hat Microsoft mit Windows Server 2008 die Managed Service Accounts eingeführt. Ein Managed Service Account verhält sich grundsätzlich wie ein Computer-Konto – das Kennwort wird vom PC verwaltet und geändert, ohne dass ein Administrator sich weiter um die Kennwortvergabe kümmern muss. Das sorgt dafür, dass das Konto immer über komplexe Kennwörter verfügt, die sich zudem noch regelmäßig ändern. Der PC ist lediglich für die Kennwortverwaltung zuständig. Aber genau hier liegt auch der Pferdefuß des Managed Service Accounts – wenn Sie mit Clusterdiensten arbeiten, können Sie Managed Service Accounts nicht nutzen, da bei einem Cluster zwei Server dasselbe Dienstkonto

verwenden. Im Fall des Managed Service Accounts heißt das, dass beide Server unabhängig voneinander versuchen, das Kennwort zu ändern, was sehr schnell zu einer Kontosperrung führt. Außerdem kann der Managed Service Accounts nicht unter Exchange, SQL-Server oder für geplante Tasks genutzt werden kann. Daher hat Microsoft mit Windows Server 2012 das Konzept des Managed Service Accounts noch einmal angepasst und den Group Managed Service Account (gmsa) eingeführt.

Beim Group Managed Service Account wird die Kennwortverwaltung des Service Accounts nicht mehr lokal vom PC aus durchgeführt, sondern vom Key Distribution Service des Domänen-Controllers, und die Computer, auf denen der Service Account genutzt wird, werden nur über das neue Kennwort informiert. Voraussetzung für die Nutzung des Group Managed Service Accounts ist ein Domänencontroller mit Windows Server 2012 oder neuer. Außerdem benötigen Sie das Powershell Active Directory Modul.

In diesem Beispiel richten wir für die SQL-Server-Dienste des Clusters einen Group Managed Service Account ein.

Vorbereitung und Anlegen eines Group Managed Service Account

Installieren Sie auf Ihrem Verwaltungsrechner (Ab Windows 8 / Server 2012) das Active-Directory Powershell-Modul. Auf einem Server brauchen Sie das Feature nur im Server-Manager über "Rollen und Features" hinzuzufügen. Alternativ können Sie Powershell verwenden.

```
Install-WindowsFeature -Name RSAT-AD-Powershell
```

Auf einem Windows-Client müssen die RSAT-Tools (Remote-Server Administration Tools) für Active Directory Domain Services nachinstallieren. Das können Sie in den Systemeinstellungen unter "Apps" – "Optionale Features" – "Optionale Features hinzufügen" erledigen.

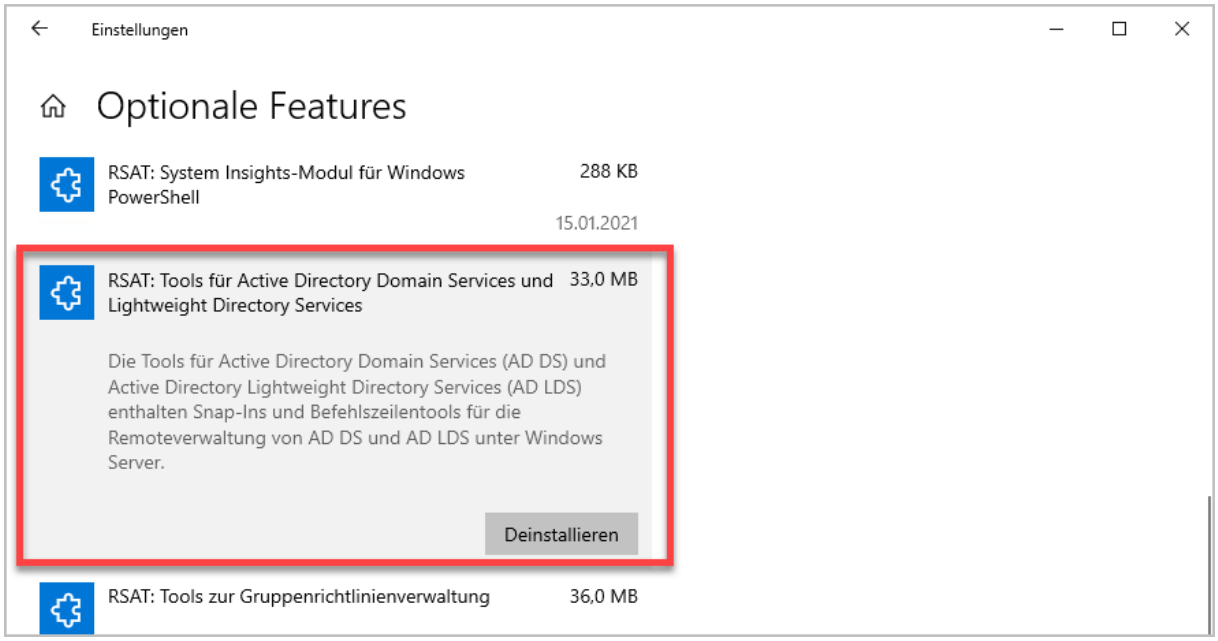


Abbildung 30 - Tools für die Active Directory Verwaltung nachinstallieren

Als erstes erstellen Sie einen KDS Root Key. Der Root-Key ist ein Schlüssel, der über alle Domänen-Controller verteilt wird, und aus dem die später erstellten Kennwörter abgeleitet werden.

```
Add-KDSRootKey -EffectiveImmediately
```

EffectiveImmediately ist übrigens ein schlechter Scherz des Programmierers, denn er bedeutet, dass der KDS-Rootkey in 10 Stunden einsetzbar ist. Dadurch soll sichergestellt werden, dass der KDS-Rootkey erst komplett über alle Domänencontroller repliziert ist, bevor er für das Erstellen von Group Managed Service Accounts genutzt werden kann. Man kann dieses Verhalten umgehen, indem man die Aktivierungszeit des Rootkeys um 10 Stunden zurück setzt – das sollte man aber nur in Laborumgebungen machen, oder wenn alle Domänencontroller an einem AD-Standort stehen!

```
Add-KdsRootKey -EffectiveTime ((get-date).addhours(-10))
```

Das Erstellen des Rootkeys muss nur einmal durchgeführt werden, danach ist er aktiv und kann für das Erstellen beliebiger Group Managed Service Accounts genutzt werden.

Der KDS-Rootkey und der KDS-Dienst

Für die Erstellung der gmsa-Kennwörter ist der Microsoft Key Distribution Service zuständig, der auf allen DC ab Windows Server 2012 läuft. Der KDS erstellt neue Kennwörter für die gmsa. Das Kennwort kann aus dem gmsa-Konto im AD ausgelesen werden, wenn man über die entsprechenden Berechtigungen verfügt. Die Berechtigungen werden direkt beim Erstellen des Kontos mit dem Parameter *-PrincipalsAllowedToRetrieveManagedPassword* vergeben.

Da jeder DC das Kennwort ändern kann, muss von jedem DC das gleiche Kennwort generiert werden. Dafür verwenden die DC alle den gleichen Algorithmus, der Zeitabhängig ist. Daraus ergibt sich das Problem, dass jeder Angreifer, der den Algorithmus kennt, prinzipiell in der Lage ist, das Kennwort zu errechnen. Aus diesem Grund wird ein Zufallwert in den Berechnungsalgorithmus mit aufgenommen und im AD gespeichert, der KDS-Rootkey. Der KDS-Rootkey kann jederzeit neu erstellt werden – er hat nur Auswirkungen auf neu generierte Kennwörter. Bereits erstellte Kennwörter sind im gmsa abgelegt und werden erst beim nächsten Kennwort-Zyklus geändert – dann aber mit dem neuen Rootkey.

Anschließend können Sie einen Group Managed Service-Account erstellen. Das geht **ausschließlich** mit Powershell.

```
# Geben Sie das Kommando in einer Zeile ein
New-ADServiceAccount -name <ServiceAccountName> -DNSHostName
<fqdn> -PrincipalsAllowedToRetrieveManagedPassword <Computer>
```

Name gibt den Netbios-Namen des Service-Accounts an, er darf also maximal 15 Zeichen lang sein. *DNSHostname* ist ein vollqualifizierter Domänenname (hier wird der DNS-Name des Dienstes eingetragen, wenn er einen DNS-Namen verwendet), und *PrincipalsAllowedToRetrieveMangedPassword* sind die Computernamen der Maschinen, die Zugriff auf den Account bekommen sollen – in unserem Fall also die Cluster-Knoten. Geben Sie hinter den Computernamen jeweils ein $-Zeichen ein, da die Anmeldenamen der Computer im AD immer mit einem $-Zeichen abschließen.

Um die Verwaltung der Computernamen zu vereinfachen, kann es Sinn ergeben, eine Gruppe zur Berechtigungsvergabe auf den Group Managed Service Account zu nutzen. Denken Sie dann allerdings daran, dass das Aufnehmen eines Computers in eine Gruppe danach auf dem Computer einen Neustart erfordert.

```
# Geben Sie das Kommando in einer Zeile ein
New-ADServiceAccount -Name SvcSQLCluster -DNSHostName
SvcSqlCluster.netz-weise.de -PrincipalsAllowedToRetrieveManagedPassword
SQL1$,SQL2$,SQL3$
```

Der Managed Service Account wird standardmäßig im Container "Managed Service Accounts" direkt im Root Ihrer Domäne angelegt. Mit dem Parameter *Path* können Sie einen alternativen Pfad wählen.

Verwenden des gmsa

Im letzten Schritt müssen Sie den Service-Account noch auf den Computern bekannt machen, die den Service-Account nutzen sollen. Hierfür melden Sie sich an jedem Clusterknoten an und starten eine Powershell mit administrativen Berechtigungen. Das Cmdlet zum Aktivieren des Service-Accounts lautet *Install-ADServiceAccount*. Es ist Bestandteil des Active-Directory Moduls.

```
Install-ADServiceAccount -Identity SvcSQLCluster
```

Mit *Test-ADServiceAccount* können Sie überprüfen, ob der Serviceaccount korrekt angelegt wurde. Das Cmdlet sollte True zurück liefern.

```
Test-ADServiceAccount -Identity SvcSQLCluster
```

Anschließend starten Sie den SQL-Server Configuration Manager, wählen im Knoten "SQL Server Services" den SQL-Server Dienst Ihrer Instanz aus, und öffnen die Eigenschaften aus dem Kontextmenü.

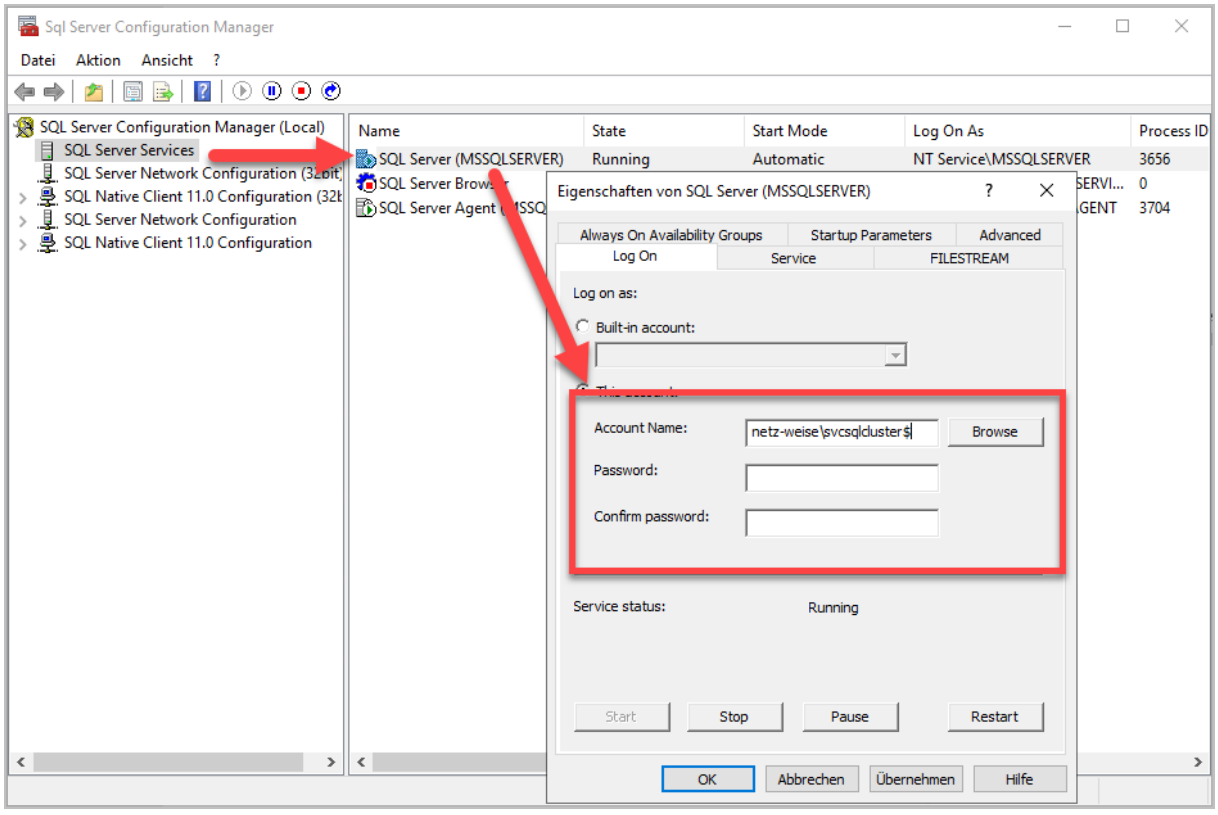


Abbildung 31 - Der gmsa wird ohne Kennwort angegeben.

Statt eines lokalen Dienstkontos geben Sie jetzt den gmsa an, den Sie erstellt haben, und lassen das Kennwort leer. Geben Sie hinter dem Namen wieder ein $-Zeichen an, da der Anmeldenamen eines gmsa wie ein Computerkonto mit einem $-Zeichen abgeschlossen wird. Anschließend starten Sie den Dienst neu, um die neuen Anmeldeinformationen zu übernehmen.

Wie viele Service-Accounts brauche ich?

In der Beispielkonfiguration wird nur ein gmsa für alle SQL-Server verwendet. Aus Sicherheitsgründen ist das aber keine gute Idee. Verwenden Sie in einer Produktiv-Umgebung besser pro SQL-Server einen eigenen Account. Wird einer der Accounts gehackt, hat der Angreifer damit nur auf einen Server Zugriff. Auch wenn dem Account die Anmeldung verweigert wird, sperren Sie nur einen Server, und nicht gleich alle.

Wenn Sie den SQL-Server Agent mit einem Domänenkonto verwenden, können Sie ruhigen Gewissen dasselbe Konto verwenden, da der SQL-Server Agent auch Sysadmin-Rechte benötigt.

Hinweise zu Group Managed Service Accounts

Es gibt eine ganze Reihe von Cmdlets für Group Managed Service Accounts. Um sich per Powershell alle Accounts anzeigen zu lassen, wählen Sie z.B.

```
Get-ADServiceAccount -Filter *
```

Das automatisch generierte Kennwort ist 120 Zeichen lang, also standardmäßig sehr sicher. Schränken Sie die Service-Accounts aber nur auf die Server ein, die den Account wirklich benötigen, da es für einen Benutzer mit Admin-Rechten ein leichtes ist, die Kennwörter aller angemeldeten Benutzer im Klartext anzuzeigen!

Die Kennwörter werden in regelmäßigen Abständen neu gesetzt. Wie oft, kann man aber auch manuell über *New-ADServiceAccount* setzen. Da alle KDS-Server das Kennwort automatisch generieren, beruht der Kennwort-Erzeugungsalgorithmus auf keinem wirklich zufälligen Wert. Die Kennwörter werden generiert aus:

- Dem KDS-Key
- Der aktuellen Zeit
- Der SID des Group Managed Service Accounts

Mehr Informationen dazu finden Sie hier:

http://blogs.technet.com/b/askpfeplat/archive/2012/12/17/windows-server-2012-group-managed-service-accounts.aspx

Eine ausführliche Anleitung zu Group Managed Service Accounts finden Sie bei Microsoft:

https://technet.microsoft.com/en-us/library/jj128431.aspx

Einrichten der Availability Group

Availability Groups werden im SQL-Server Management Studio eingerichtet. Am besten installieren die aktuelle Version des Management-Studio auf einem Admin-Client. Das Management-Studio können Sie unter https://docs.microsoft.com/de-de/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15 oder kurz https://bit.ly/3Cx0PqP herunterladen.

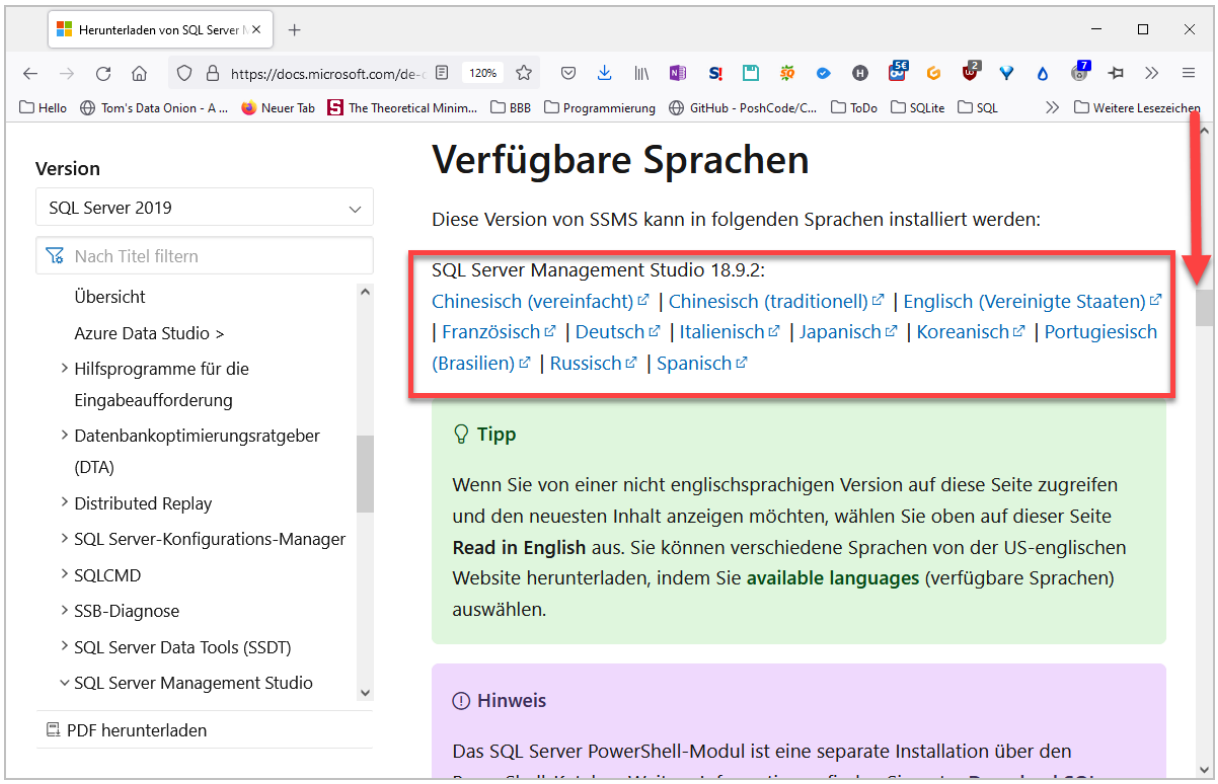


Abbildung 32 - Wenn Sie auf der Website nach untern scrollen, finden Sie dort unterschiedliche Sprachversionen

Verbinden Sie sich im Management Studio mit der Datenbank-Engine des Servers, dessen Datenbank hochverfügbar gemacht werden soll, mit einem Konto, das Mitglied in der Sysadmin-Rolle ist.

Über den Knoten "Hochverfügbarkeit mit Always on" können Sie Availability-Groups einrichten. Da Availability-Groups alle Änderungen in der Datenbank als Transaktionen auf die Zielserver replizieren, müssen sich alle Datenbanken, die Teil der Availability-Group sein sollen, im Vollständigen Wiederherstellungsmodell befinden. Das können Sie im Knoten "Datenbanken" direkt auf der Datenbank prüfen. Öffnen Sie dafür die Eigenschaften der Datenbank aus dem Kontextmenü und wechseln Sie in die Menü-Seite Optionen.

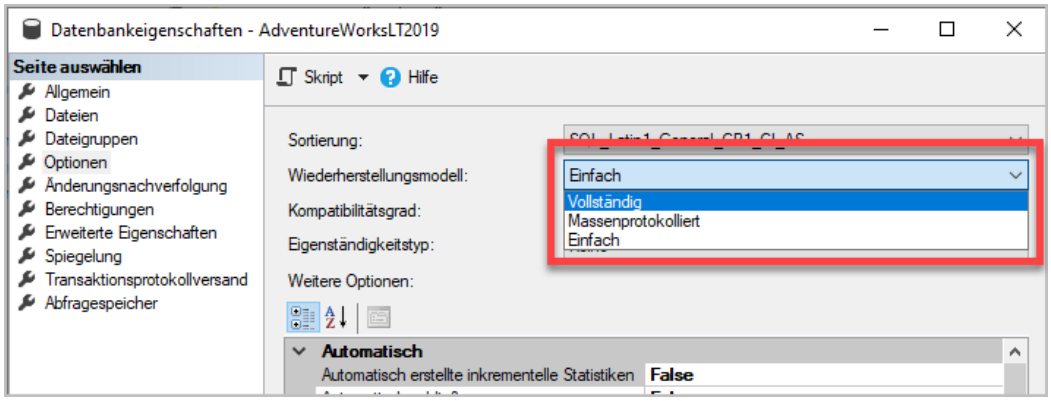


Abbildung 33 - In den Eigenschaften unter Optionen kann das Wiederherstellungsmodel angepasst werden

Unter Wiederherstellungsmodell können Sie zwischen Einfach, Massenprotokolliert und Vollständig auswählen. Nur im Vollständigen Modus speichert SQL-Server jede Datenbankänderung zusätzlich zur eigentlichen Datenbank im Transaktionsprotokoll. Wenn die Datenbank nicht im Vollständigen Wiederherstellungsmodus ist, stellen Sie sie hier um. Anschließend müssen Sie noch einmal ein

vollständiges Backup von der Datenbank machen, bevor das vollständige Wiederherstellungsmodell tatsächlich aktiv ist.

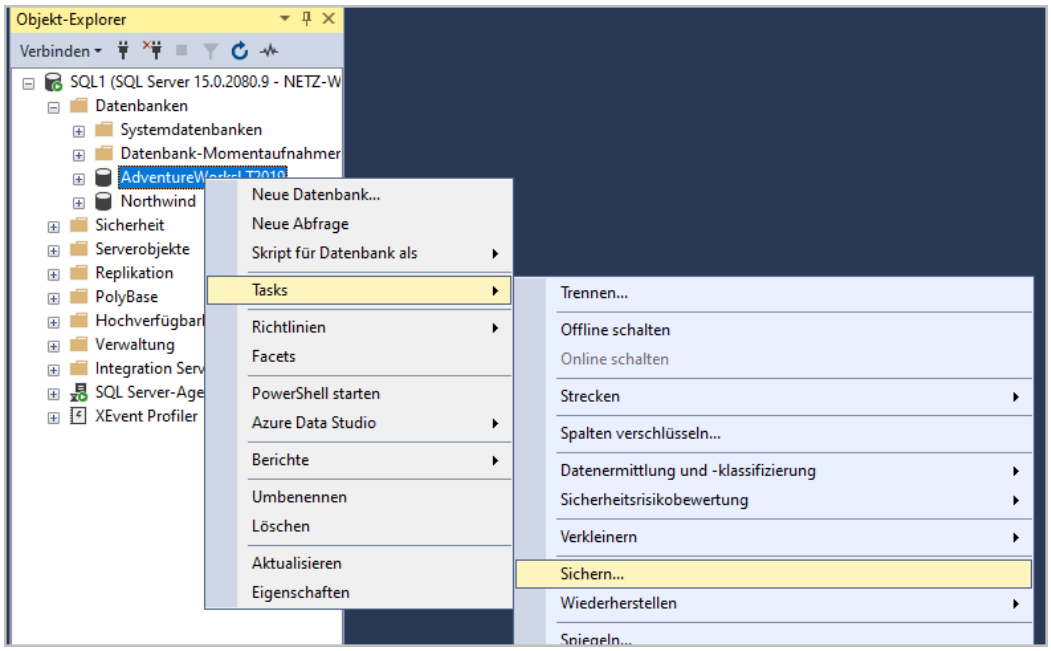


Abbildung 34 - Die Sicherung finden Sie im Kontextmenü der Datenbank im Untermenü Tasks

Um eine Availability Group anzulegen, starten Sie aus dem Kontextmenü des Menü-Knotens "Hochverfügbarkeit mit Always On" jetzt den "Assistent für neue Verfügbarkeitsgruppen".

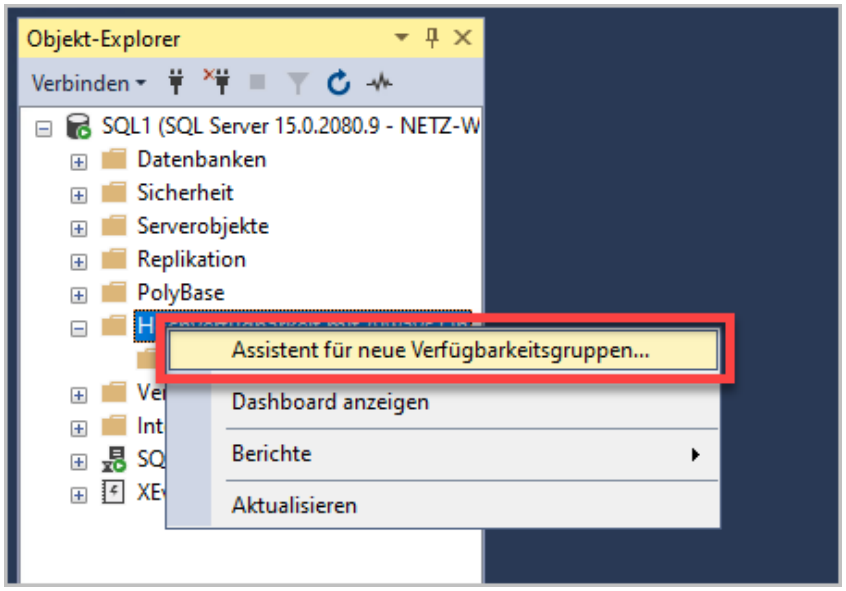


Abbildung 35 - Der Assistent leitet durch die Einrichtung einer Availability Group

Der Assistent leitet Sie nun durch die Erstellung der Availability Group. Überspringen Sie die Einführung und geben Sie einen Namen für die Availability-Group an. Dieser Name dient nur der Anzeige im Management-Studio und repräsentiert die gemeinsam verwalteten Datenbanken. Normalerweise wird dies der Name der Applikation sein, zu der die Datenbanken gehören.

https://www.blackhat.com/html/webcast/09302021-certified-pre-owned-abusing-active-directory-certificate-services.html

Abbildung 36 -Geben Sie einen Namen für die Availability-Group ein

Ab SQL Server 2017 können Sie in dem Fenster auch den Clustertyp auswählen. Neben dem "Windows Server-Failovercluster" steht als Typ noch "Extern" zur Verfügung. Ein Externer Cluster ist eine Availability Group mit einem auf Linux installierten Server. Für Windows-basierte Installationen wählen Sie immer den Typ "Windows Server Failover".

Auch die DTC-Unterstützung pro Datenbank wurde erst mit SQL-Server 2017 eingeführt. Der DTC (Distributed Transaction Coordinator) wird für die Unterstützung von verteilten Transkationen benötigt und ist ein Spezialfall für Anwendungen, die Änderungen über Datenbanken auf mehreren Datenbankservern konsistent halten müssen. Genaueres finden Sie unter https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/transactions-always-on-availability-and-database-mirroring?view=sql-server-ver15 oder kurz https://bit.ly/2Wik9IU.

Seit SQL-Server 2016 gibt es die "Integritätserkennung auf Datenbankebene". Standardmäßig sollte diese Option aktiviert werden. Sie sorgt dafür, dass der Cluster nicht nur prüft, ob der Windows-Server erreichbar ist, sondern auch, ob er sich mit der Datenbank verbinden und einen Datensatz schreiben kann. Die Integritätserkennung auf Datenbankebene kann nur pro Availability-Group aktiviert werden, nicht pro Datenbank! Mehr zur Funktionsweise der Prüfung finden Sie unter https://www.sqlshack.com/database-level-health-detection-in-sql-server-always-on-availability-groups/ oder kurz https://bit.ly/3lV8b0o.

Im nächsten Schritt wählen Sie die Datenbanken aus, die Mitglied der Availability Group sein sollen. Die Datenbanken, die Sie der Availability-Group hinzufügen, laufen immer gemeinsam auf dem gleichen Server. Wenn eine der Datenbanken verschoben werden muss, weil die Datenbankprüfung fehlgeschlagen ist, verschiebt der Server immer die komplette Availability-Group.

Der Assistent zeigt Ihnen auch an, aus welchen Gründen Datenbanken nicht in die Availability-Group aufgenommen werden können. In Abbildung 37 sehen Sie zwei mögliche Beispiele – die Datenbank Demo läuft nicht im vollständigen Wiederherstellungsmodell, während von der Datenbank NoBackup noch kein vollständiges Backup gemacht wurde.

Wenn Sie die Probleme behoben haben, können Sie die Datenbanken erneut prüfen, indem Sie "Aktualisieren" auswählen.

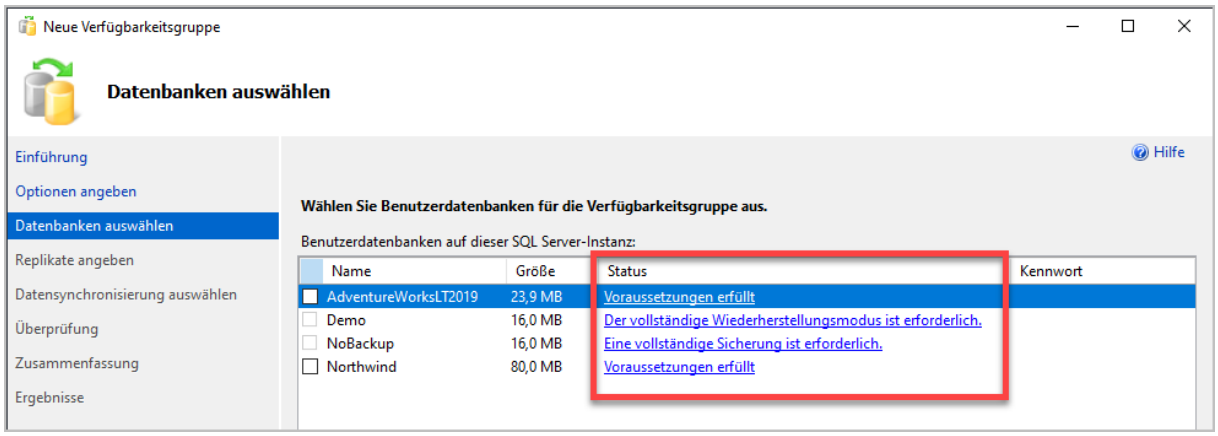


Abbildung 37 - Wählen Sie alle Datenbanken aus, die auf dem gleichen Server laufen müssen

Im Abschnitt "Replikate angeben" wird es interessant. Hier fügen Sie zuerst die vorher für Always On konfigurierten Server über "Replikat hinzufügen" (1) aus. Sollten der Zielserver nicht erreichbar sein, prüfen Sie, ob der SQL-Server Dienst läuft und Sie den Zielserver auf dem SQL-Server Port erreichen können. Bei einer Standardinstanz ist das normalerweise Port 1433. Am einfachsten können Sie die Prüfung remote mit Powershell durchführen. Verwenden Sie dazu das Cmdlet *Test-NetConnection* mit den Parametern *-ComputerName* und *-Port*.

```
Test-NetConnection -Computername SQL1.Netz-Weise.de -Port 1433
```

Die häufigste Fehlerursache ist die Windows-Firewall. Eine weitere Fehlerursache kann sein, dass Sie auf der Developer-Edition das TCP-Protokoll zur Datenübertragung nicht aktiviert haben (s. Aktivieren des TCP-Protokolls auf der SQL-Server Developer-Edition, S. 18). Im Zweifel versuchen Sie, auf dem Zielserver den SQL-Server-Dienst einmal neu zu starten.

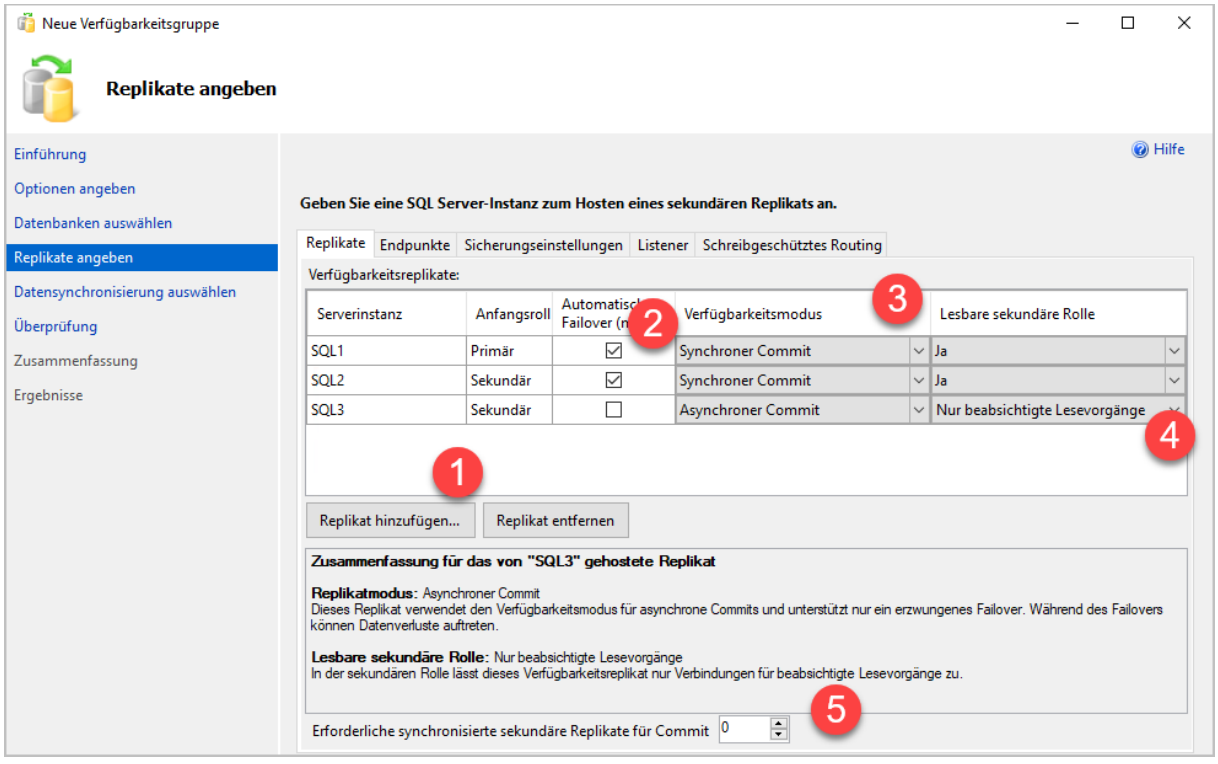


Abbildung 38 - Hier legen Sie die wesentlichen Informationen für den Failover fest

Die Spalte "Anfangsrolle" zeigt Ihnen an, welcher Server schreibend auf die Daten zugreifen kann – die Primäre Rolle. Dies ist auch gleichzeitig der Server, mit dem sich Clients verbinden, wenn Sie eine Verbindung über den Listener herstellen. Alle anderen Server sind Sekundär.

Über "Automatischer Failover" (2) legen Sie fest, welche Server im Failover-Fall die Primäre Rolle übernehmen können. Diese Einstellung wird dann im Failover in die Konfiguration "Bevorzugter Besitzer" übernommen.

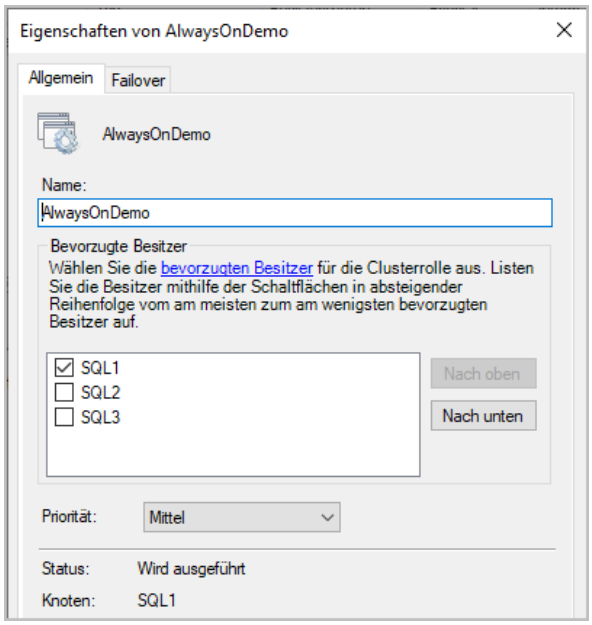


Abbildung 39 - Den bevorzugten Besitzer finden Sie in den Rollen des Cluster-Managers

Der Verfügbarkeitsmodus legt fest, ob Transaktionen auf dem Zielsystem bestätigt werden müssen, bevor Sie auf der primären Rolle in der Datenbank übernommen werden. Ist der Verfügbarkeitsmodus auf "Asynchroner Commit" gestellt, kann der Primäre Server alle Transaktionen sofort abschließen und die Änderungen werden für die Benutzer sofort sichtbar. Beim synchronen Commit muss der sekundäre Knoten die Transaktion erhalten und in seinem Transaktionsprotokoll gespeichert haben, bevor die primäre Rolle die Transkation abschließt.

Ein Synchroner Commit kann die Performance der primären Rolle negativ beeinflussen, stellt aber sicher, dass die Daten zwischen der primären und der synchronen sekundären Rolle immer identisch sind. Daher kann ein manuelles Failover, der durch einen Serverneustart oder im Management-Studio ausgelöst wird, auch immer nur mit Partnern im synchronen Commit-Modus stattfinden.

In der Spalte "Lesbare sekundäre Rolle" legen Sie fest, ob das Replikat lesend verwendet werden kann, oder sich im Wiederherstellungs-Modus befindet, der keinen Zugriff auf die Datenbank zulässt. Nur beabsichtigte Leserolle ist ähnlich der lesbaren sekundären Rolle, allerdings muss die Client-Anwendung explizit angeben, dass sie nur lesend auf die Datenbank zugreifen möchte, um auf diesen Knoten umgeleitet zu werden. Die Lese-Absicht wird beim Herstellen der Verbindung mit der Datenbank im Connection-String als "Read Only Intent" angegeben. Diese Verbindungen können über Routing-Regeln auf einen Server weitergeleitet werden, der eine "Read Only Intent"-Replika hält. Mehr über Routing-Regeln erfahren Sie weiter unten.

Welcher Replika-Typ ist der Richtige?

Eine Primäre Replika ist die einzige Schreibende Instanz der Datenbank. Von Ihr gibt es immer genau eine. Alle Kopien der Replika können entweder im Wiederherstellungszustand (Recovery Mode) verharren, was bedeutet, dass ein Zugriff auf diese Datenbanken nicht möglich ist, oder sie befinden sich im Lesbaren Zustand. Grundsätzlich gibt es keinen guten Grund, eine Replika im Wiederherstellungsmodus zu belassen – außer, man möchte explizit verhindern, dass die Kopie verwendet wird.

Synchrone Replikas haben immer den gleichen Datenstand wie die Primäre Replika. Das wird dadurch sichergestellt, dass jede Änderung immer auf dem Primären Replikat und auf allen Kopien übernommen sein muss, bevor Benutzer die Daten in der Datenbank sehen können. Im Gegensatz dazu läuft eine asynchrone Replika der Primären immer hinterher – jede abgeschlossene Transaktion ist auf der Primären Server immer sofort verfügbar, auch wenn das Replikat noch die Transaktionen vor 20 Minuten abarbeitet.

Grundsätzlich sind synchrone Replikate in den meisten Fällen die bessere Lösung. Wenn die Server aber bei der Replikation aufgrund räumlicher Entfernung sehr hohe Latenzen haben (Serverdistanz ca. > 500km) oder der Server, der das sekundäre Replikat hält, ist signifikant langsamer als der Server, der die Primäre Rolle hält und kommt mit dem Schreiben der Daten oft nicht nach, sollte man auf asynchrone Replikate setzen, da es sonst zu scheinbaren massiven Performance-Problemen kommen kann. Scheinbar deshalb, weil die Primäre Rolle die Daten eigentlich schnell genug schreibt, aber das Warten auf den Commit, also die Bestätigung der sekundären Rolle, zur Verzögerung führt. Da bei einem asynchronen Commit die sekundären Rollen der primären Rolle in den Transaktionen hinterher hängen können, kann es bei einem Failover zu Datenverlust kommen. Wenn man mehrere Replikate hat, kann man aber je Instanz bestimmen, ob die Replikation den synchronen oder asynchronen Commit verwenden soll.

Nachdem Sie Ihre Replikas konfiguriert haben, wechseln Sie auf den Reiter "Endpunkte".

Always On Availability Groups bauen auf der bereits seit Windows Server 2005 verfügbaren Datenbankspiegelung auf. Die Datenbankspiegelung und Availability Groups kopieren alle Transaktionen direkt über das Netzwerk auf die Replikats-Server. Dafür wird ein dedizierter Port verwendet. In diesem Menü legen Sie den Endpunkt fest, der neben dem Port auch noch einen Namen Endpunktnamen definiert. Der Endpunkt wird auch verwendet, um Replikationsberechtigungen zu vergeben.

Standardmäßig wird Port 5022 mit dem Namen "Hadr_endpoint" verwendet. Das Akronym HADR steht dabei für High Availability Disaster Recovery. Über diesen Port werden die Transaktionen zwischen den Replikas kopiert. In unserem Beispiel verfügten die beiden Server SQL1 und SQL2 bereits über einen Mirroring-endpoint mit Namen Mirroring. Die vorhandenen Endpoints werden von Always On in diesem Fall weiterverwendet.

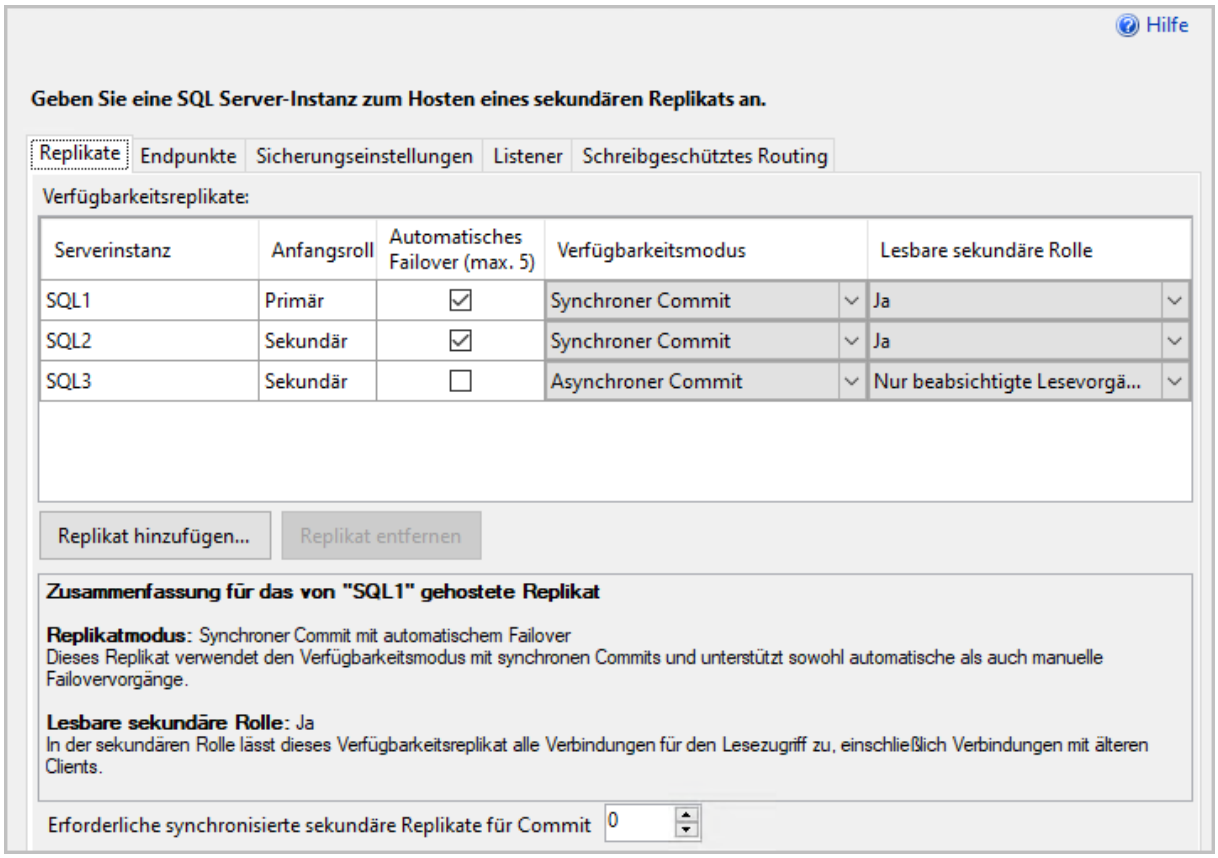


Abbildung 40 - Die Endpunkte anlegen

In der Spalte "Daten verschlüsseln" können Sie festlegen, ob der gesamte Spiegelverkehr verschlüsselt werden soll. Da über die Replikation sämtliche Änderungen, die in der Datenbank vorgenommen werden, übertragen werden, sollte die Verschlüsselung immer aktiv sein.

Auf der Registerkarte "Sicherungseinstellungen" können Sie festlegen, welche Server zur Datensicherung verwendet werden. Dadurch können Sie die Primäre Instanz vom Backup entlasten. Wählen Sie hier "nur Sekundär", werden nur sekundäre, lesbare Replikate für das Backup genutzt. "Sekundäres Replikat bevorzugen" verwendet das primäre Replikat nur dann, wenn ein sekundäres Replikat nicht verfügbar ist. Über die Backup-Priorität können Sie festlegen, welcher Server als erstes für das Backup angesprochen wird, wenn "Beliebiges Replikat" gewählt wurde, wobei 1 für die niedrigste und 100 für die höchste Priorität steht.

Die Bedeutung der Sicherungseinstellungen

Sicherungseinstellungen können schnell missverstanden werden. Tatsächlich haben die Sicherungseinstellungen auf Backups nämlich grundsätzlich keine Auswirkungen. Die Daten für die bevorzugte Instanz zur Sicherung werden auf dem SQL-Server einfach nur in einer Tabelle hinterlegt. Starten Sie ein manuelles Backup, sind dem Server die Vorgaben vollkommen egal. Sie müssen, um die Sicherungspräferenzen zu nutzen, ein Skript verwenden, das die Sicherungseinstellungen abfragt und das Backup nur dann ausführt, wenn die Bedingungen erfüllt sind. Dieses Skript muss dann auf allen Instanzen implementiert werden. Beim Starten des Backup-Jobs prüft das Skript dann, ob es als Instanz für die Sicherung infrage kommt und startet das Backup – oder eben nicht.

Wenn Sie den SQL-Server Wartungsplaner für die Sicherung verwenden, verwendet dieser ein passendes Skript. Eine bessere Alternative sind allerdings die Skripte von Ola Hallengren. Deren korrekte Implementierung wird weiter unten beschrieben.

Wenn Sie die Präferenzen noch genauer durchblicken wollen, bietet der Artikel "Understanding backups on AlwaysOn Availability Groups" unter https://bit.ly/39wYNKH eine gute Zusammenfassung.

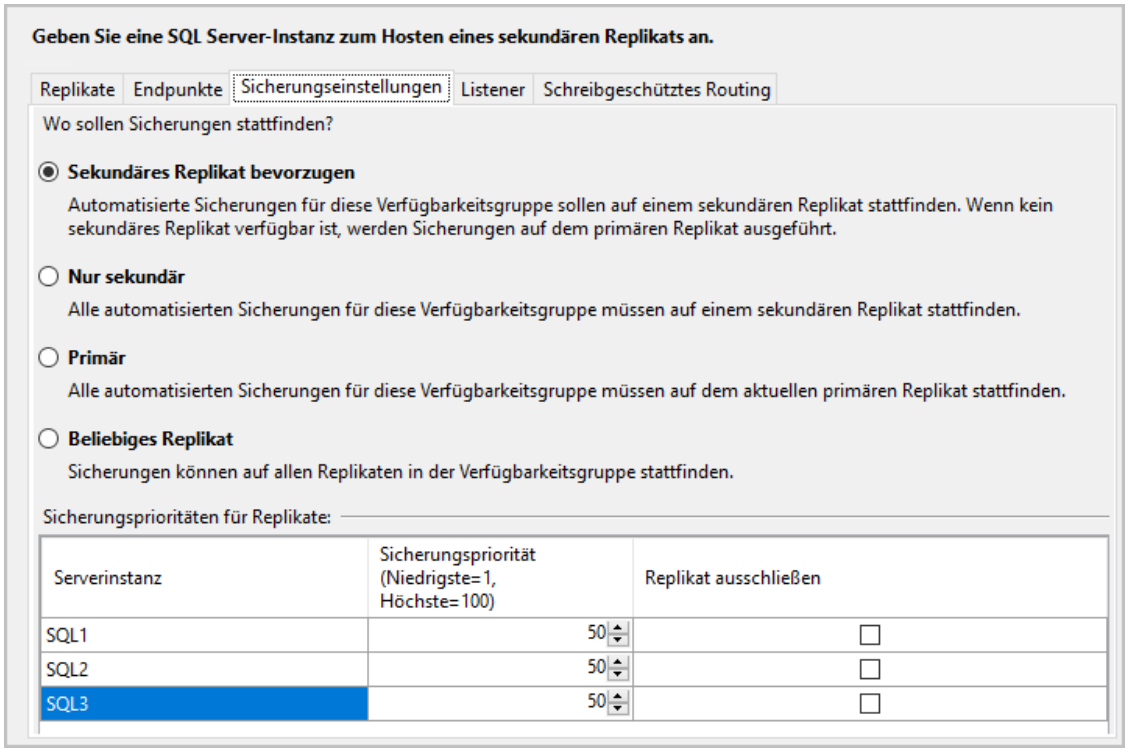


Abbildung 41 - Die Sicherungseinstellungen haben auf manuelle Backups keine Auswirkungen

Im nächsten Schritt legen Sie den Listener an. Der Listener wird zwar im Management-Studio konfiguriert, aber anschließend im Failover-Cluster angelegt. Der Listener legt für die Availability Group einen eigenen Computernamen und eine IP-Adresse sowie einen Computerkonto im Active Directory an. Achten Sie darauf, dass der Cluster-Dienst Schreibrechte auf der OU hat, in der er sich selbst befindet, da das Erstellen der Computernamen für die Listener sonst fehl schlägt (s. Active Directory anpassen, S. 14). Grundsätzlich ist ein Listener nicht notwendig, da Sie die SQL-Server Instanzen immer auch direkt über Ihren Namen erreichen können. Ein automatisches Failover ist ohne Listener aber nicht möglich.

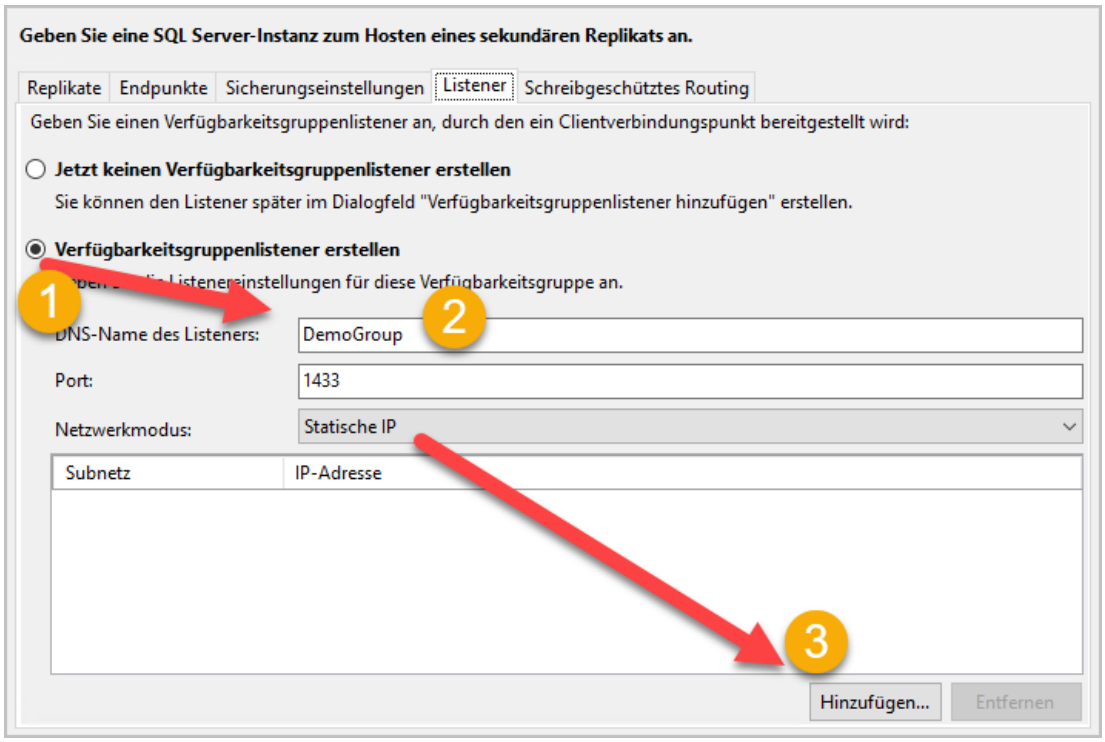


Abbildung 42 - Der Listener verhält sich wie ein neuer SQL-Server

Wählen Sie zuerst "Verfügbarkeitslistener erstellen" (1) aus, da Ihre Availability-Group sonst nicht unter einem gemeinsamen Namen erreichbar ist. Unter "DNS-Name des Listeners" (2) geben Sie den Computernamen an, unter dem die Availability Group erreichbar sein soll (1), sowie einen Port und eine IP-Adresse. Der Port ist sinnvollerweise der Standard-SQL-Port 1433. Das ist unproblematisch, da die Kombination aus Port und IP-Adresse, die im nächsten Schritt festgelegt wird, eindeutig ist.

Um die IP-Adresse festzulegen, wählen Sie nun "Hinzufügen" (3) aus und geben eine vollständige IP-Adresse aus dem Subnetz Ihres Clusters an. Da Ihr Cluster normalerweise in mehreren Subnetzen ist, wählen Sie hier das öffentliche Netzwerk Ihres Clusters.

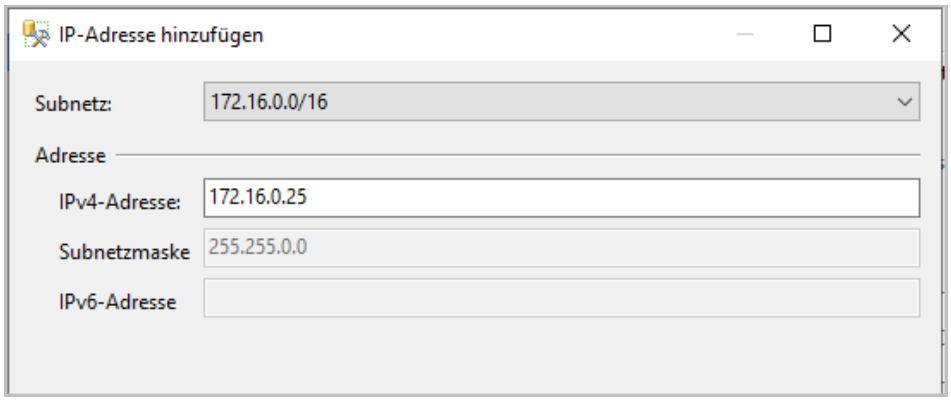


Abbildung 43 - Wählen Sie das Subnetz aus und geben Sie eine eindeutige IP-Adresse an

Im Knoten Schreibgeschütztes Routing können Sie festlegen, mit welchen Replikaten Clients verbunden werden sollen, die nur lesenden Zugriff auf die Datenbank anfordern. Das Read-Only-Routing kann nachträglich angepasst werden (wie alle anderen Einstellungen auch) und wird weiter unten beschrieben.

Bevor Sie die Einrichtung endgültig starten können, muss im letzten Schritt noch festgelegt werden, wie die Kopien von der primären Rolle auf die sekundären Rollen übertragen werden sollen. Hierfür

stehen Ihnen drei Varianten zur Verfügung – "Automatisches Seeding" und "Vollständige Datenbank- und Protokollsicherung". Beim Automatisches Seeding wird die Datenbankkopie direkt über das Netzwerk übertragen. Damit das automatische Seeding funktioniert, müssen die Dateipfade der Original-Dateien auch auf den Replikat-Servern vorhanden sein.

Die vollständige Datenbank- und Protokollsicherung erstellt ein initiales Backup der zu replizierenden Datenbank und legt es in einer Netzwerkfreigabe ab. Die Sekundären Rollen stellen das Backup aus der Freigabe wieder her. Die Primäre Rolle benötigt Schreibrechte auf der Freigabe, die Sekundären Rollen müssen die Freigabe lesen können.

Wenn Sie das Backup von Hand wiederherstellen möchten – z.B. weil die Datenbank über ein manuelles Backup an einem langsam angebundenen Remote-Standort wiederhergestellt wurde – wählen Sie "Nur beitreten".

Wenn Sie "Automatisches Seeding" ausgewählt haben, brauchen Sie nichts weiter tun und können mit der Überprüfung fortfahren. Wenn Sie die Datenbank-Sicherung verwenden, geben Sie den UNC-Pfad zur Freigabe, in der das Backup abgelegt werden soll. Anschließend fahren Sie mit der Überprüfung fort.

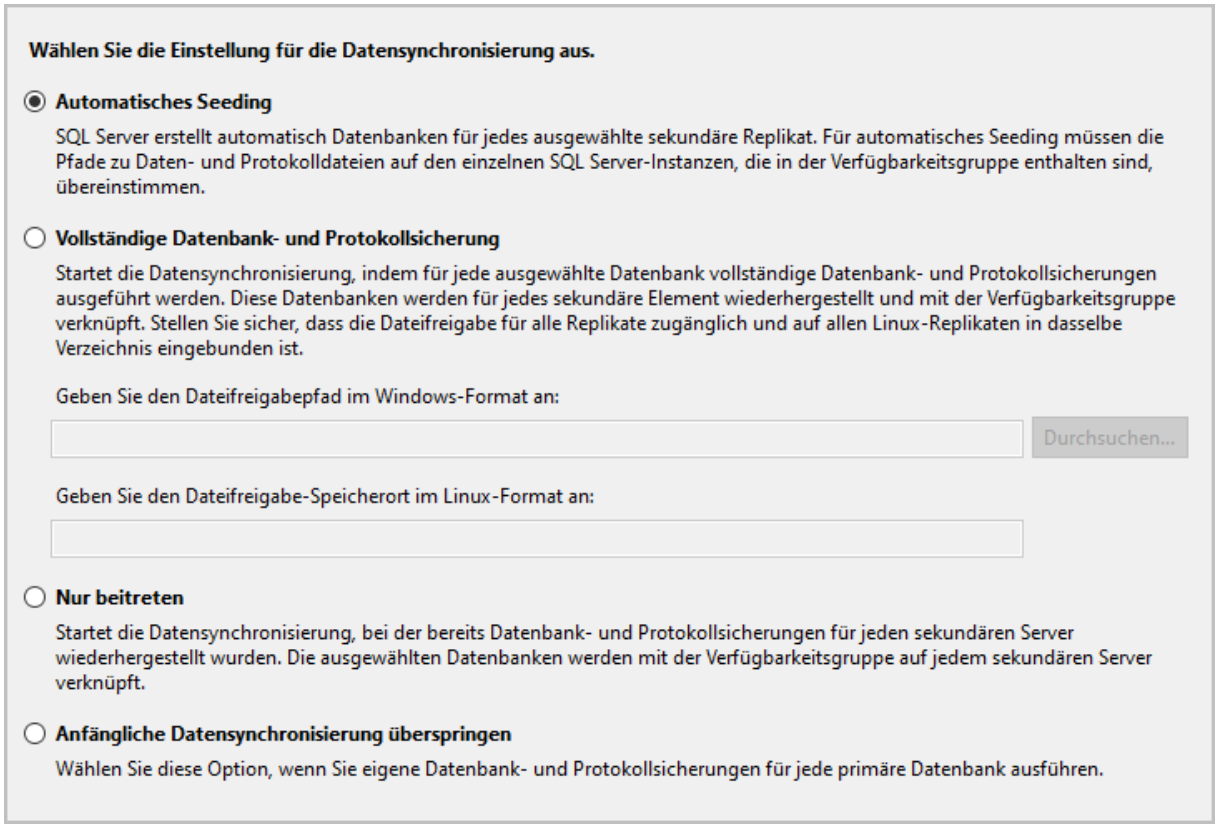


Abbildung 44 - Innerhalb eines Standorts ist automatisches Seeding normalerweise die beste Wahl

Der Server prüft jetzt, ob alle Voraussetzungen für eine erfolgreiche Spiegelung erfüllt sind.

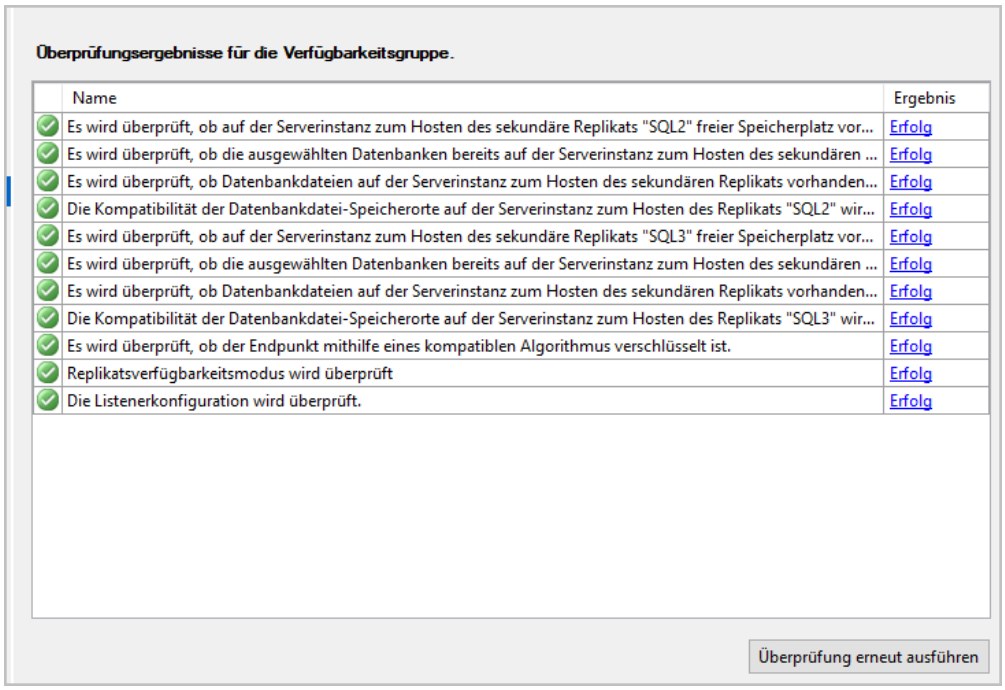


Abbildung 45 - Alle Voraussetzungen sind erfüllt, die Installation kann beginnen

Waren nicht alle Tests erfolgreich, können Sie den Fehler jetzt beseitigen und die Überprüfung erneut ausführen lassen.

Sind alle Tests durchgelaufen, sehen Sie unter Zusammenfassung noch einmal die gewählte Konfiguration. Über den Button "Skript" können Sie sich das "Skript" erzeugen lassen, mit dem der Assistent die Einrichtung vornimmt. Das Skript aus dem Beispiel 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.

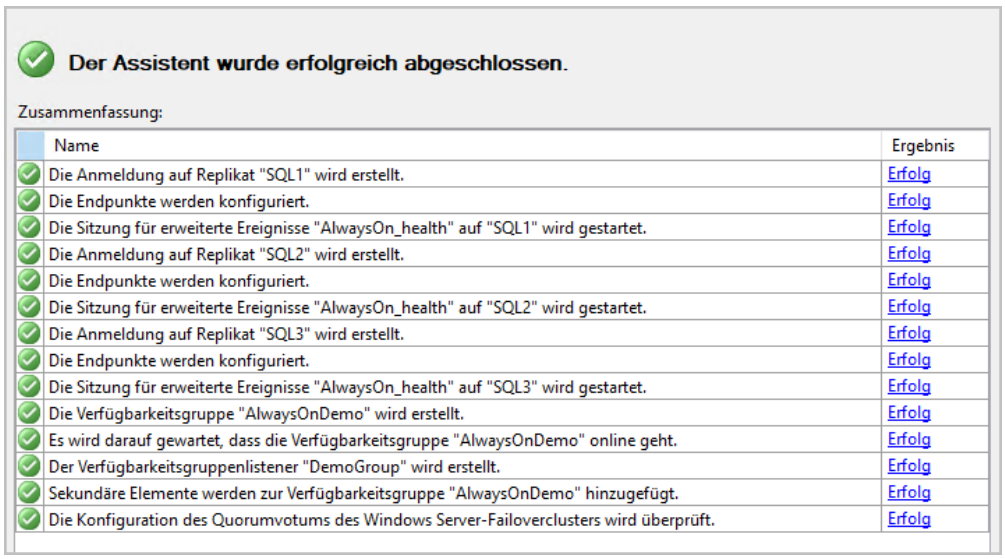


Abbildung 46 - Die Installation war erfolgreich

Am Bericht können Sie sehen, welche Schritte bei der Installation durchgeführt wurden: Das Anlegen von Anmeldungen (Logins) für die Service-Accounts der jeweiligen Instanzen, Erstellen eines Endpunkts, Erstellen einer Rolle im Cluster-Dienst und das Replizieren der Datenbank.

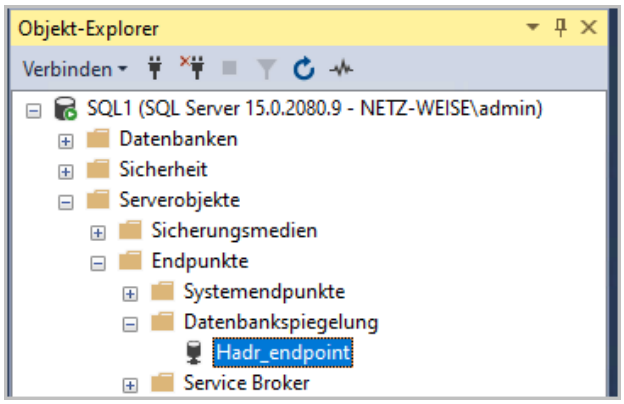


Abbildung 47 - Den Datenbankspiegelungs-Endpunkt finden Sie in den Serverobjekten

Im Failover-Cluster-Manager finden Sie unter "Rollen" die IP-Adresse und den Computernamen, die für die Availability-Group angelegt worden sind.

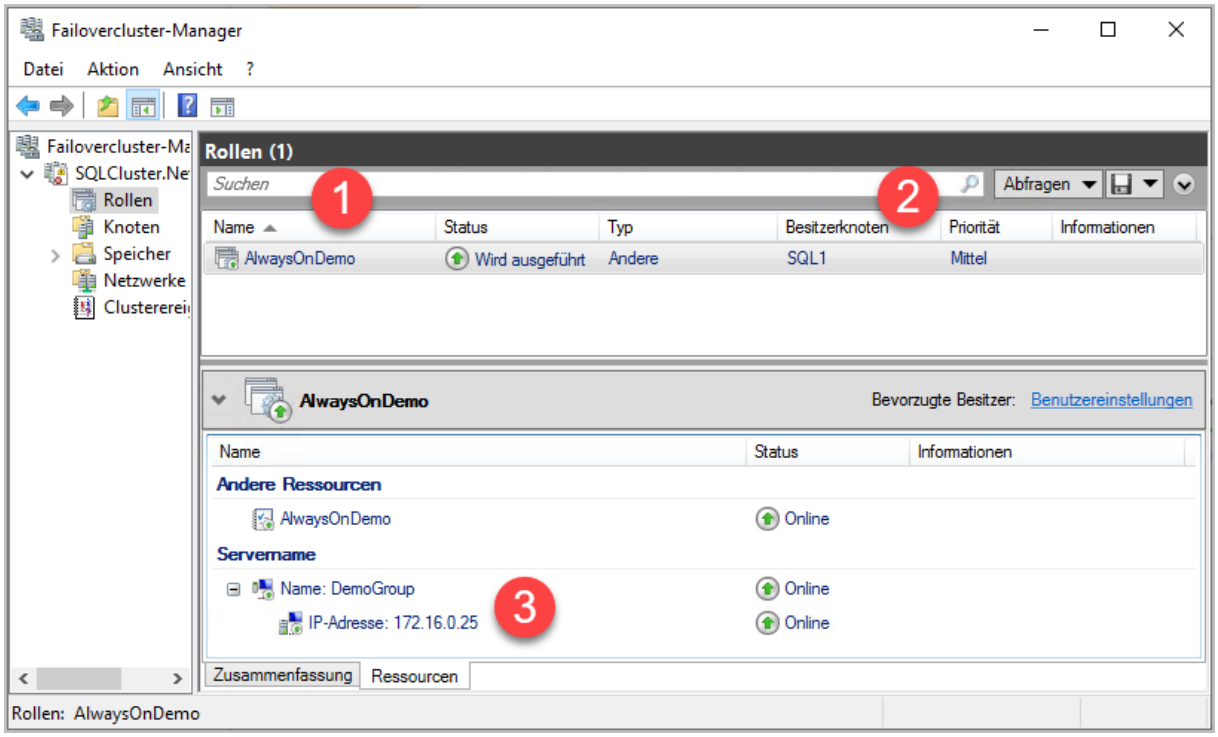


Abbildung 48 - Der Listener, wie er sich im Clustermanager darstellt

In der Spalte "Name" (1) ist der Name der Availability-Group aufgelistet. Er entspricht im Cluster dem Rollen-Namen. Der Besitzerknoten (2) ist derjenige Server, an den anfragende Clients weitergeleitet werden – also der Server mit der primären Rolle. Auf der Registerkarte Ressourcen unten im Fenster finden Sie die IP und den Computernamen, den Sie für den Listener vergeben haben. Der Cluster ist neben der Bereitstellung des Listeners aber auch für die Prüfung der Verfügbarkeit der Ressourcen zuständig. Grundsätzlich sollten Sie die Availability-Group aber nicht im Cluster-Manager, sondern immer im Management-Studio konfigurieren.

Sie können sich auch nach der Cluster-Einrichtung nach wie vor direkt mit einem SQL-Server verbinden. Wenn Sie aber immer mit der primären, schreibenden Rolle verbunden werden wollen (oder mit einer sekundären Rolle, wenn Sie nur lesen wollen), müssen Sie den Listener nutzen.

Einen Listener nachträglich hinzufügen

Wenn Sie die Verfügbarkeitsgruppe ohne Listener angelegt haben, können Sie das jederzeit nachholen. Öffnen Sie hierzu im Management-Studio das Kontextmenü des Eintrags "Verfügbarkeitsgruppenlistener", den Sie unterhalb Ihrer Verfügbarkeitsgruppe finden, und wählen Sie "Listener hinzufügen".

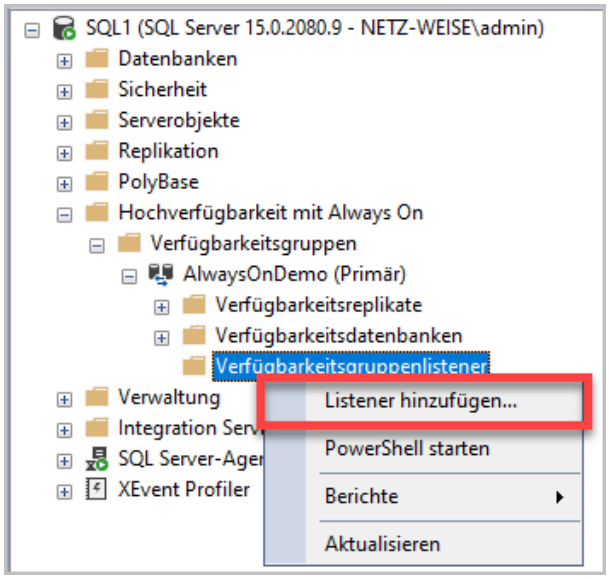


Abbildung 49 - Ein Listener kann auch nachträglich erstellt (oder gelöscht) werden

Die Konfiguration ist dann identisch zum Hinzufügen des Listeners beim Anlegen der Availability-Group.

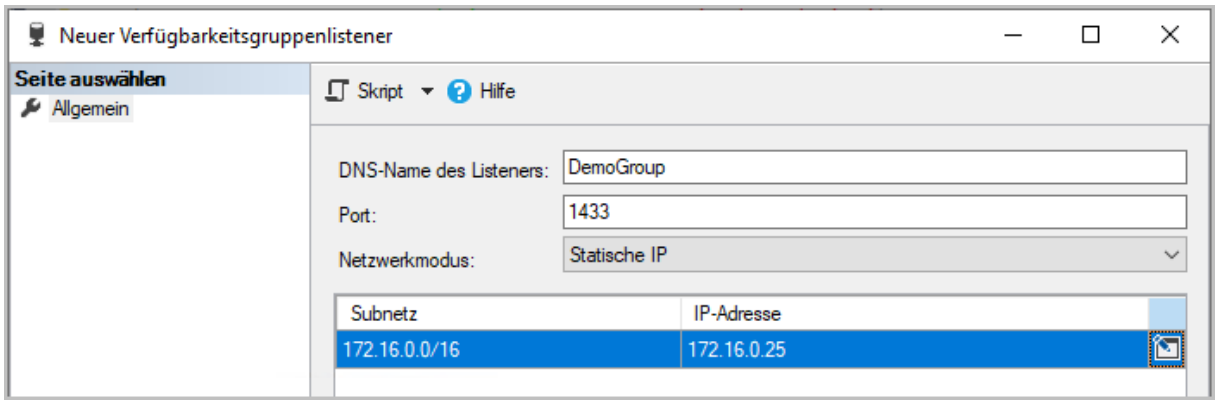


Abbildung 50 - Der Dialog zum Hinzufügen eines LIsteners - kein Unterschied zur Erstellung der AG

Prüfen der Replikation

Nachdem Sie die Replikation eingerichtet haben, können Sie den Status prüfen, indem Sie auf Ihre Verfügbarkeitsgruppe wechseln und aus dem Kontextmenü "Dashboard anzeigen" auswählen.

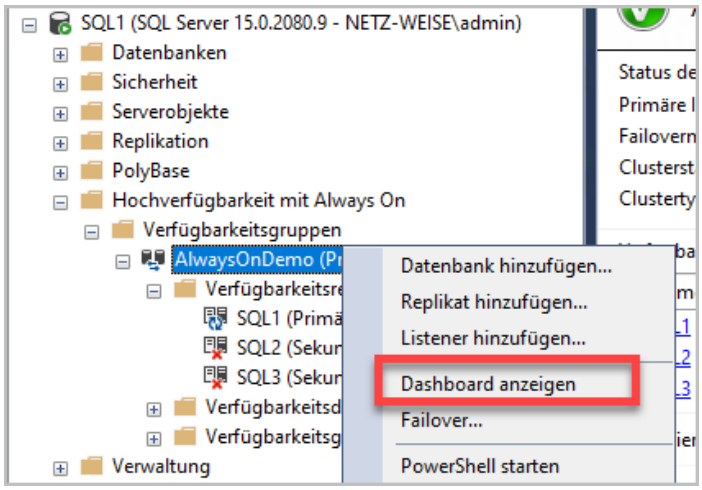


Abbildung 51 - Starten Sie das Dashboard, um den Status der Spiegelung zu prüfen

Das Dashboard zeigt Ihnen auf einer übersichtlichen Seite den Stand Ihrer Replikation.

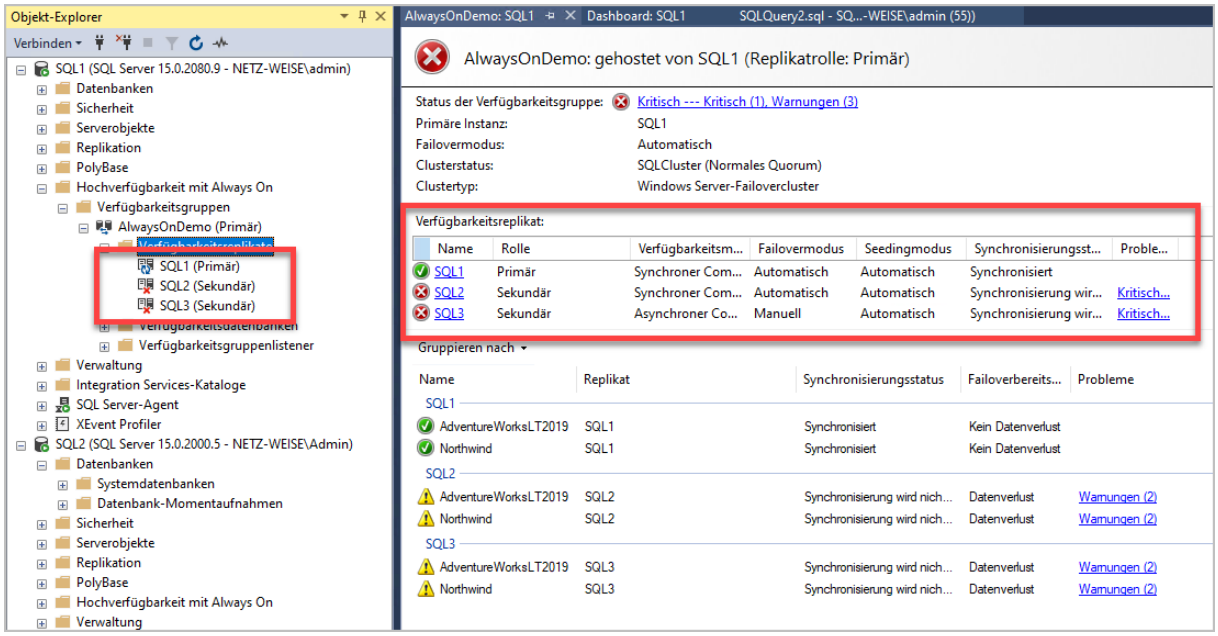


Abbildung 52 - Die Spiegelung läuft offensichtlich nicht

Der Status in Abbildung 52 ist ganz offensichtlich nicht in Ordnung. Das rote Kreuz vor den sekundären Rollen zeigt bereits an, dass etwas nicht stimmt, aber das Dashboard gibt uns Gewissheit – der Synchronisierungstatus sagt "Synchronisierung wird nicht ausgeführt".

Der Fehler in der abgebildeten Situation ist trivial – Port TCP 5022, der für die Spiegelung verwendet wird, ist durch die Firewall geblockt. Das ist sehr irreführend, denn der Assistent hat die Konfiguration ja offensichtlich ohne Fehler abgeschlossen. Der Haken ist, dass der Assistent nur das Anlegen der Server-Objekte bescheinigt, nicht aber der eine funktionierende Spiegelung. Der Seeding-Prozess findet nämlich erst nach der Einrichtung der Server-Objekte statt.

Sobald der Port TCP 5022 freigegen ist, müssen Sie nichts weiter tun – SQL-Server startet das Seeding und danach die Spieglung der Daten. Je nach Größe der Datenbank kann aber auch der Seeding-Prozess eine ganze Weile dauern. Wenn Sie also direkt nach der Einrichtung das Dashboard öffnen und dort sehen, dass die Datenbanken nicht synchron sind, brechen Sie nicht in Panik aus – mit großer Wahrscheinlichkeit hat das System die Daten noch nicht fertig auf die Sekundären Rollen

übertragen. Gut Prüfen kann man das im Taskmanager auf der Registerkarte "Leistung". Solange die Netzwerkverbindung stark ausgelastet ist, ist der SQL-Server vermutlich noch damit beschäftigt, die Daten über die Leitung zu schaufeln. Wenn Sie es genauer wissen wollen, stellt SQL-Server Ihnen zwei Systemsichten zur Verfügung, die Sie per SQL abfragen können. Die folgende Abfrage, auf der primären Replika ausgeführt, gibt Ihnen eine Zeile für jeden Seeding-Prozess aus.

```
SELECT start_time,
    completion_time
    is_source,
    current_state,
    failure_state,
    failure_state_desc
FROM sys.dm_hadr_automatic_seeding
```

	start_time	is_source	current_state	failure_state	failure_state_desc
1	2021-09-24 20:00:26.893	2021-09-24 20:00:27.223	COMPLETED	NULL	NULL
2	2021-09-24 20:00:26.893	2021-09-24 20:00:27.247	COMPLETED	NULL	NULL
3	2021-09-24 20:55:56.467	2021-09-24 20:55:56.770	COMPLETED	NULL	NULL
4	2021-09-24 20:55:56.467	2021-09-24 20:55:57.910	COMPLETED	NULL	NULL

Abbildung 53 - Sie erhalten eine überschaubare Übersicht, ob der Seeding-Prozess noch läuft und ob Fehler aufgetreten sind

Für detailliertere Informationen fragen Sie die Sicht *sys.dm_hadr_physical_seeding_stat* auf der primären Replika ab.

```
SELECT * FROM sys.dm_hadr_physical_seeding_stats;
```

Mehr Informationen zum Seeding erhalten Sie auch unter https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/automatically-initialize-always-on-availability-group?view=sql-server-ver15 oder kurz https://bit.ly/3o2lXBe.

Ein weiterer Punkt, der irritieren kann – in der Availability-Group sind die Verfügbarkeitsreplikate mit einem Symbol gekennzeichnet, dass deren Status anzeigt.

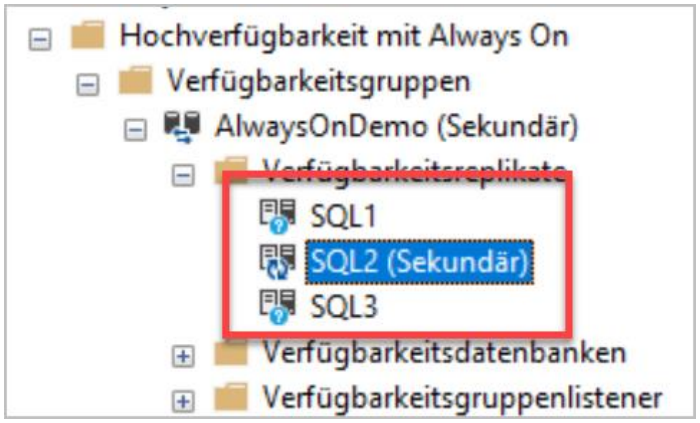


Abbildung 54 - zwei Server sind mit Fragezeichen markiert

In Abbildung 54 sieht man hier auf SQL zwei drehende Pfeile, was anzeigt, dass der Server einwandfrei spiegelt, während SQL1 und SQL3 mit einem Fragezeichen versehen sind. Das bedeutet nicht, dass hier irgendetwas kaputt ist, sondern tatsächlich ist das Management-Studio mit der Instanz SQL2 verbunden, die als sekundäre Rolle konfiguriert ist. Sekundäre Rollen können in der Datenbank nur lesen und haben keine direkte Verbindung mit den anderen Servern. Daher haben Sie auch keine Informationen über den Zustand der anderen Replikate. Nichts anderes bedeuten die ?? – SQL2 weiß einfach nicht, in welchen Zustand die Server sind.

Wenn man im Management-Studio eine Verbindung mit der primären Rolle herstellt und sich dort die Verfügbarkeitsreplikate anzeigen lässt, erhalten wir ein ganz anderes Bild.

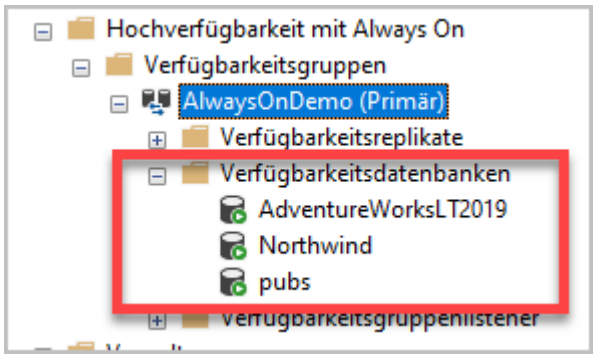


Abbildung 55 - Die primäre Rolle kennt den Status aller Replikate

Verwalten von Availability-Groups

Ein manuelles Failover initiieren

Wenn Sie die Rollen der Server tauschen wollen – z.B. um Updates einzuspielen -, können Sie das über das Kontextmenü der Verfügbarkeitsgruppe oder im Dashboard machen.

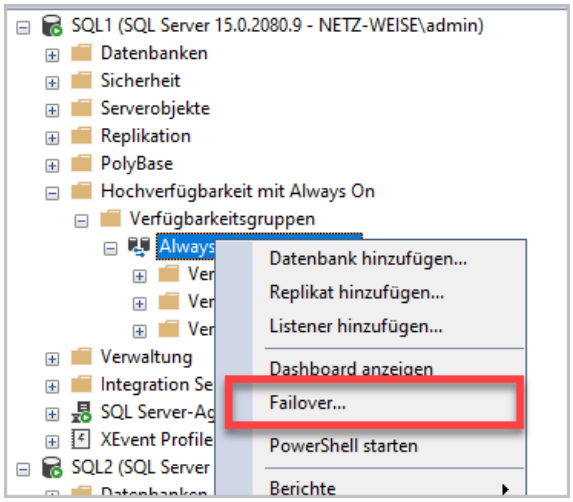
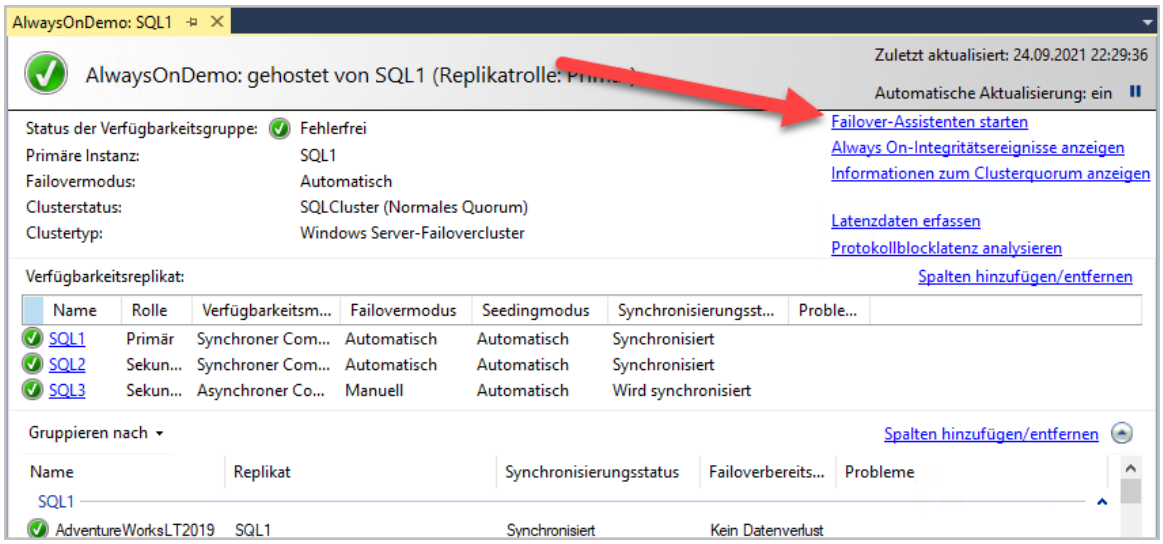


Abbildung 56 - ein manuelles Failover kann im Kontextmenü der AG ausgelöst werden

In beiden Fällen starten Sie den Failover-Assistenten.



Als erstes wählen Sie aus, welche Instanz primäre Rolle sein soll.

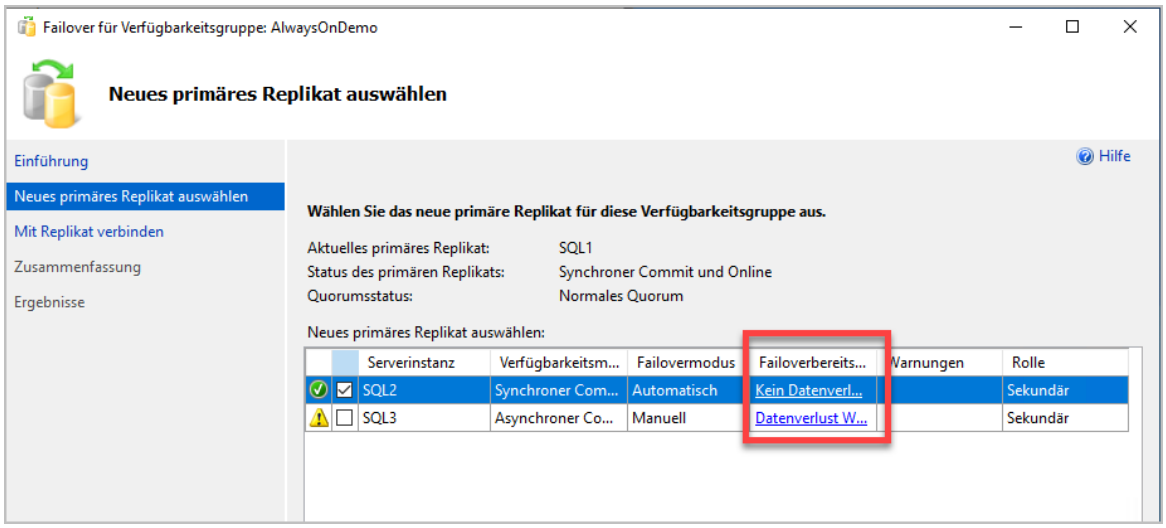


Abbildung 57 - Wählen Sie den Zielserver für das primäre Replikat aus

In der Beispielkonfiguration gibt es zwei Replikate – SQL2 mit synchronem Commit und SQL3 mit asynchronem Commit. Das Failover kann zu beiden Servern durchgeführt werden, aber der Assistent warnt uns, dass es beim Failover zu SQL3 zu einem Datenverlust kommen kann, da der asynchrone Commit nicht garantiert, dass SQL1 und SQL3 auf dem gleichen Stand sind. Das Failover zu SQL2 kann dagegen problemlos ausgeführt werden.

Wenn Sie das Failover zu SQL3 trotzdem ausführen wollen, erhalten Sie eine zusätzliche Warnung. Das Failover kann nur erzwungen werden, indem Sie die Aktion noch einmal bestätigen. Da ein synchroner Knoten existiert, macht es daher mehr Sinn, auf SQL2 zu schwenken.

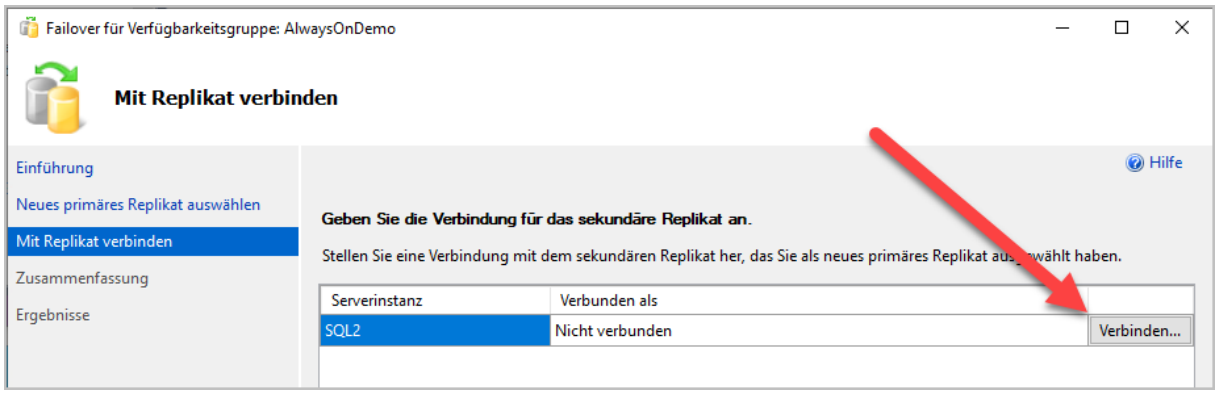


Abbildung 58 - Verbindung mit dem Zielserver herstellen

Vor dem Failover muss das Management-Studio sich am Server anmelden. Wählen Sie hierfür "Verbinden" aus.

Anschließend können Sie die angegebenen Daten noch mal prüfen. Mit "Fertig stellen" wird das Failover eingeleitet. Der ganze Vorgang dauert nur wenige Sekunden, da die Datenbanken auf beiden Servern identisch sind. Im Hintergrund wird nur der Listener auf SQL2 geschwenkt und der Lesemodus der Datenbanken wird umstellt. Anschließend wird die Replikationsrichtung umgedreht.

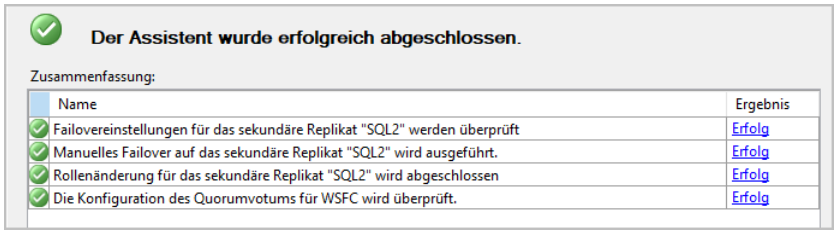


Abbildung 59- Die Datenbank wurde erfolgreich geschwenkt

Wissenwertes zu Availability Groups

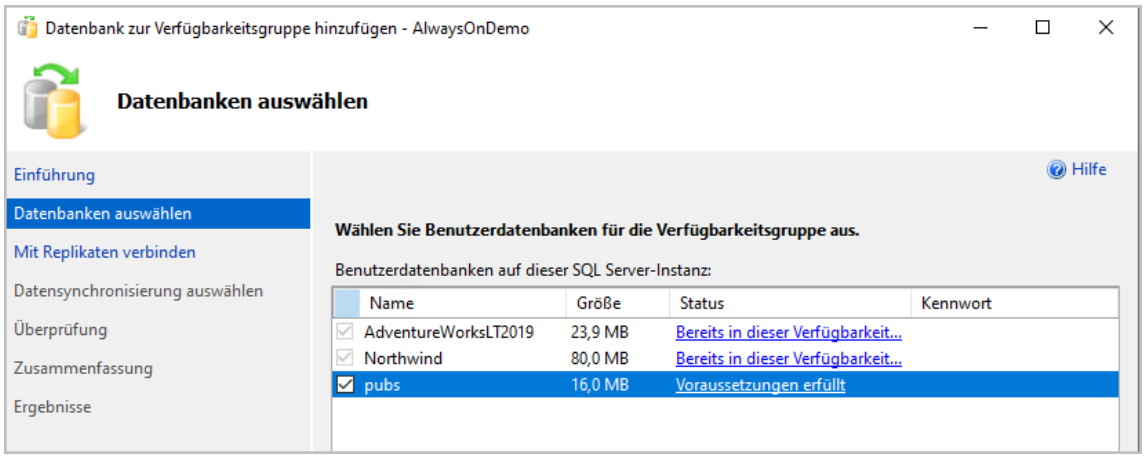
Availability-Groups unterscheiden sich in einem ganz wesentlichen Punkt von der klassischen Datenbankspiegelung. Bei der Spiegelung konfigurieren Sie jeweils einzelne Datenbanken. Eine Availability-Group kann jedoch mehrere Datenbanken zusammenfassen. Die Availability Group wird zu einer Einheit, die 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, da alle Datenbanken in einer Availability-Group zusammengefassten Datenbanken immer eine Einheit darstellen, die nur gemeinsam auf einem Server laufen kann!

Öffnen Sie dazu unter Ihrer Availability Group das Kontextmenü des Eintrags „Verfügbarkeitsdatenbanken" und wählen Sie "Datenbank hinzufügen". Unter "Datenbanken auswählen" wählen Sie die Datenbank aus, die sie hinzufügen möchten. Es gelten die gleichen

Voraussetzungen wie oben definiert, die Datenbank muss sich also im vollständigen Wiederherstellungsmodus befinden und einmal vollständig gesichert sein.



Melden Sie sich im nächsten Schritt über "Verbinden" an den Replikat-Servern an. Danach wählen Sie den Datensynchronisationsmodus aus. Hier gelten wieder die gleichen Regeln wie bei der initialen Einrichtung. Sind die Server über ein schnelles Netzwerk verbunden, ist "Automatisches Seeding die einfachste Möglichkeit der Einrichtung.

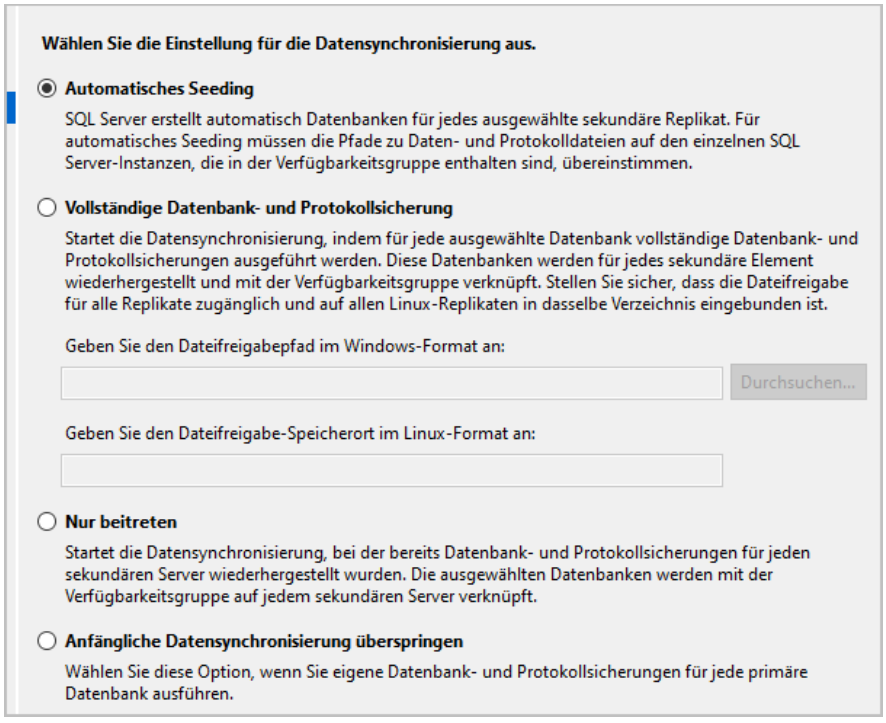


Abbildung 60 - Sie haben die gleichen Optionen wie beim Einrichten der AG

Es findet eine Validierung statt, die von einer Zusammenfassung abgeschlossen wird.

Abbildung 61 - Eine weitere Datenbank wurde erfolgreich hinzugefügt

Die neue Datenbank ist in die Availability Group aufgenommen worden.

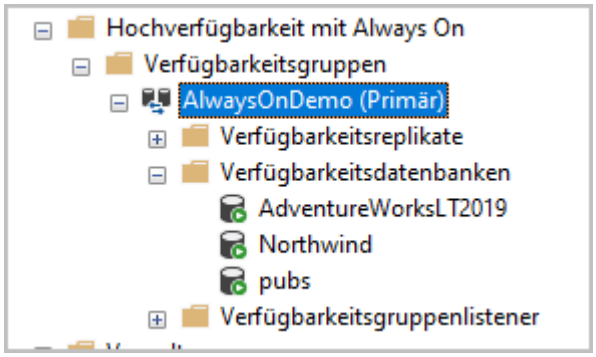


Abbildung 62 - Die Verfügbarkeitsgruppe wurde erweitert

Read Only Routing konfigurieren

Mit dem Read Only oder schreibgeschütztem Routing können Anwendungen, die nur lesend auf die Datenbanken zugreifen, auf ein nur lesendes Replikat weitergeleitet werden. Voraussetzung hierfür ist, dass im Connection-String der Anwendung "Read Intent" angegeben ist, dass die primäre Rolle über eine Read Only Routing-Tabelle verfügt, dass es eine oder mehrere sekundäre Rollen gibt, die für "Read Intent Only" konfiguriert worden sind (s. Abbildung 38) und dass der Client sich über einen Listener verbindet. Außerdem benötigen alle sekundären Rollen eine Read-Only Routing URL. Die URL folgt folgendem Schema:

Tcp://Servername/Instanz:Port

Matt Neerincx hat in einem inzwischen ziemlich alten Blogpost, der noch über das Internet Archive verfügbar ist, genau beschrieben, wie die Verbindung mit der primären Rolle über den Listener und das anschließende Routing vonstattengeht geht. Im gleichen Artikel hat er auch ein SQL-Skript veröffentlicht, dass die Read Only Routing URLs für alle Server einer Availability Group erzeugt. Das Skript finden Sie im Anhang B dieses Dokuments, den Blog-Artikel unter https://web.archive.org/web/20170512023255/https://blogs.msdn.microsoft.com/mattn/2012/04/25/calculating-read_only_routing_url-for-alwayson/ oder kurz https://bit.ly/3oa3rHi.

Hier kurz zusammengefasst der Ablauf:

1. Der Client verbindet sich über einen Listener mit der Availability Group und verbindet sich mit der primären Rolle. Im Connection-String der Client-Anwendung ist die Option ApplicationIntent=ReadWrite gesetzt.
2. Die primäre Rolle prüft, ob es ein Replikat mit der Konfiguration "Read Only Intent" gibt. Danach prüft er in seiner Routing-Liste für den "Read Only Intent" in der angegebenen Reihenfolge, welcher Server verfügbar und im Status "Synchronizing" ist.
3. Der ermittelte Server wird an den Client übermittelt, und der Client stellt eine Verbindung mit der Read-Only Replika her.

Am einfachsten können Sie das Read-Only-Routing über das Management-Studio konfigurieren, indem Sie die Eigenschaften der Availability-Group öffnen und in die Seite "Schreibgeschütztes Routing" wechseln. Ist diese Seite bei Ihnen nicht vorhanden, verwenden Sie vermutlich eine alte Version des Management-Studios. Die aktuelle Version können Sie unter https://bit.ly/3Cx0PqP herunterladen.

Auf der Seite finden Sie eine Tabelle, in der die Instanzen Ihrer Availability-Group aufgelistet sind. Tragen sie hier unter "URL für schreibgeschütztes Routing" zuerst die Routing-URL der jeweiligen Instanz ein (1). Anschließend müssen Sie für jede Instanz eine Routing-Liste erstellen. Die Liste ist einfach eine Auflistung der Routing-URLs aller Server mit Read-Intent. Die Reihenfolge der Server ist wichtig, denn sie gibt die Priorität der Server an. Der primäre Server prüft (s. 2. im Ablauf) die Verfügbarkeit der sekundären Rollen immer in der hier angegebenen Reihenfolge.

Wählen Sie hierzu in der Zusammenfassung die Instanz aus, für die eine Routingliste erstellt werden soll. Unter verfügbare Replikate (2) finden Sie die Liste aller für "Read Only Routing" konfigurieren Replikate, für die eine Routing-URL erstellt wurde. Wählen Sie alle Instanzen aus, auf die weitergeleitet werden soll und wählen Sie auf hinzufügen. "Die Liste für schreibgeschütztes Routing" (3) zeigt die konfigurierte Liste an. Wenn Sie die Priorität der Instanzen ändern wollten, verschieben Sie sie in der Liste einfach mit "Nach oben" und "Nach unten". Achten Sie darauf, dass für jede Instanz eine eigene Liste erstellt werden muss. Sinnvollerweise sollte der erste Server jeder Instanz jeweils nicht auf sich selbst zeigen, da der Client sonst beim Verbinden immer zuerst mit dem

primären Replikat verbunden wird. Vergessen Sie auch nicht, dass primäre Replikat am Ende immer noch mit anzugeben, sonst kann der Client keine Verbindung mit der Availatbility Group herstellen, wenn alle sekundären Replikate mit Read Intent offline sind.

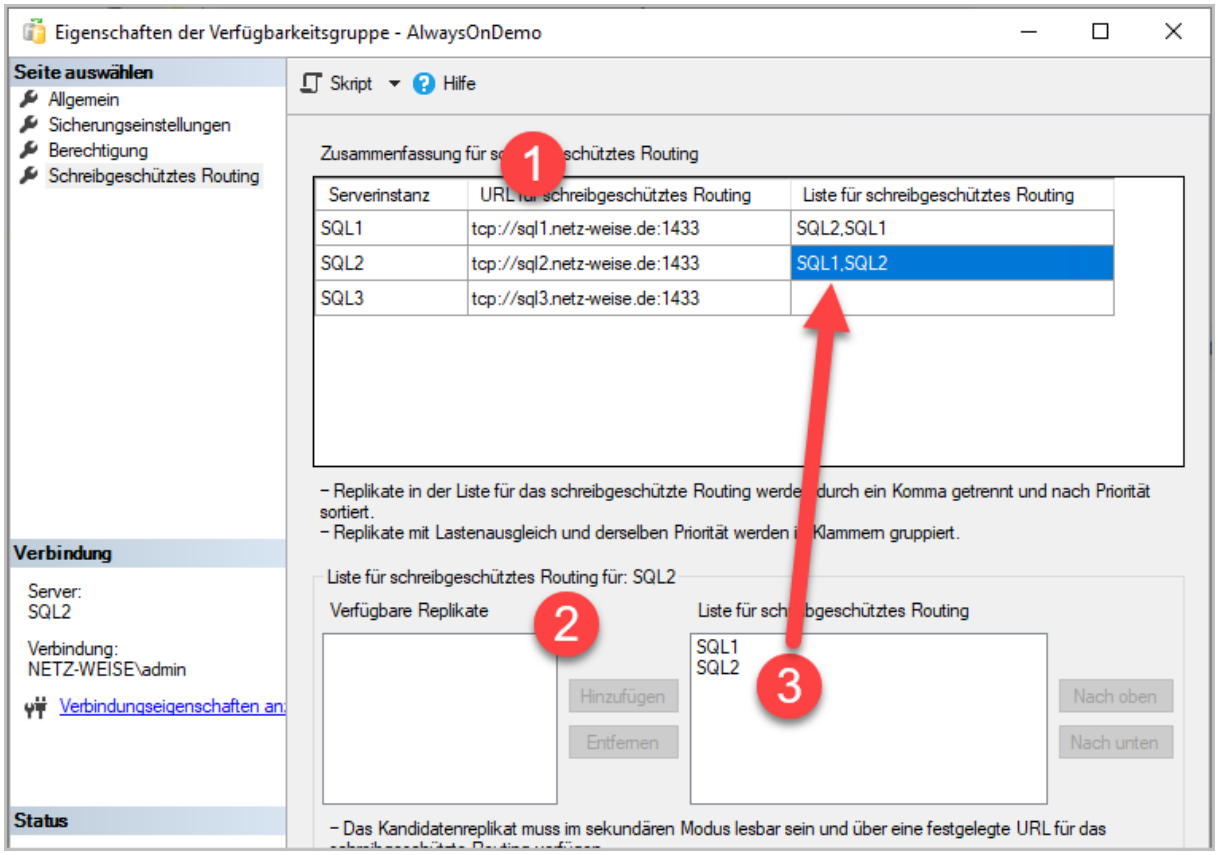


Abbildung 63 - Read Only Routing im Managment-Studio konfigurieren

Im Beispiel in Abbildung 63 ist die Instanz SQL1 so konfiguriert, dass eine Anwendung erst auf SQL2 weitergeleitet wird, wenn er die Option "Read Intent" gesetzt hat. Ist SQL2 nicht verfügbar, wird er mit SQL1 verbunden. Ist stattdessen SQL2 die primäre Instanz, wird die Anwendung zuerst an SQL1 weitergeleitet.

Testen der Routing-Konfiguration

Sie können z.B. mit dem Management-Studio prüfen, ob Sie Ihre Routing-Listen korrekt konfiguriert haben. Verbinden Sie sich hierzu im Objekt-Explorer über "Verbinden" mit einer neuen. Geben Sie als Servernamen den Namen der Availability-Group an, und wählen anschließend "Optionen" aus, um sich die erweitere Verbindungskonfiguration anzeigen zu lassen.

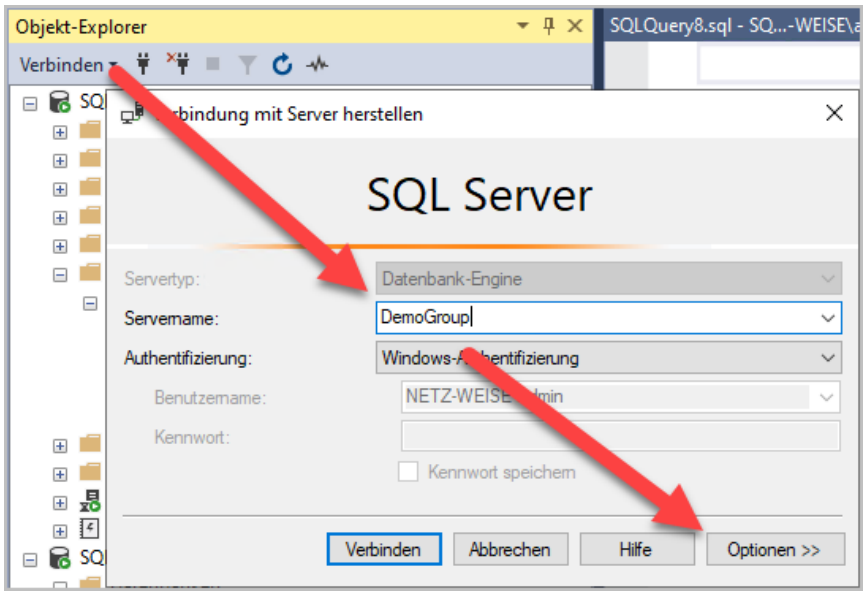


Abbildung 64 - Geben Sie als Servernamen den Listener an und wählen Sie Optionen

Auf der Registerkarte "Zusätzliche Verbindungsparameter" geben Sie ***ApplicationIntent=ReadWrite*** an wählen "Verbinden".

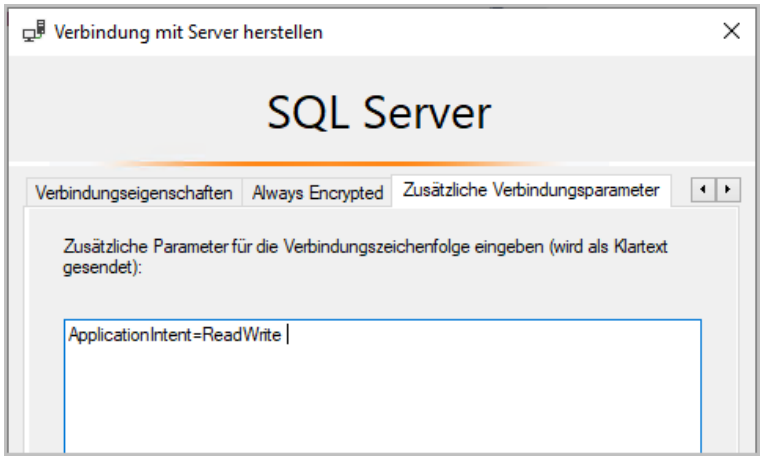


Abbildung 65 - Passen Sie die Verbindungsparameter an.

Sie können prüfen, mit welcher Instanz Sie verbunden sind, indem Sie ein neues Abfragefenster öffnen und die Systemvariable @@Servername abfragen.

```
Select @@servername
```

In der Beispielkonfiguration hält SQL2 nach dem Failover die primäre Rolle, in der Routing-Liste ist SQL1 als bevorzugtes Replikat für "Read-Intent" angegeben. Die Umleitung hat geklappt!

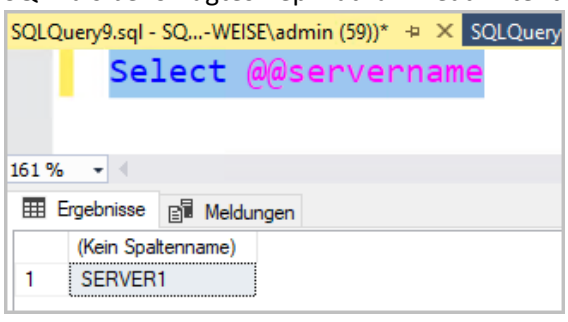


Abbildung 66 - Das Routing hat offensichtlich funktioniert

Read-Only Routing per SQL-Skript einrichten

Sie können Read-Only Routing auch per SQL oder Powershell einrichten.

Wenn Sie die Routing-URLs bestimmt haben, müssen Sie sie in der Availability Group hinterlegen. Verwenden Sie für den Konfiguration die primäre Rolle und das Statement *Alter Availability Group*.

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

Mit dem folgenden Skript fügen Sie die Routing-URLs für die sekundären Replikas hinzu (entspricht der Eingabe der Routing-URLs im Management-Studio).

```
Alter Availability Group SQLAOGroup
Modify Replica ON 'SQL1'
With (SECONDARY_ROLE ( READ_ONLY_ROUTING_URL = 'TCP://sql1.netz-weise.de:1433' ));
GO

Alter Availability Group SQLAOGroup
Modify Replica ON 'SQL2'
With (SECONDARY_ROLE ( READ_ONLY_ROUTING_URL = 'TCP://sql2.netz-weise:1433' ));
GO

Alter Availability Group SQLAOGroup
Modify Replica ON 'SQL3'
With (SECONDARY_ROLE ( READ_ONLY_ROUTING_URL = 'TCP://sql3.netz-weise.de:1433' ));
GO
```

Als nächstes konfigurieren Sie die Read-Only Routing-Tabelle.

```
ALTER AVAILABILITY GROUP <Groupname>
MODIFY REPLICA 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 (s.o).

Nachfolgend das Beispielskript für das Lab.

```
USE [master]
GO
ALTER AVAILABILITY GROUP [AlwaysOnDemo]
MODIFY REPLICA ON N'SQL2' WITH
(PRIMARY_ROLE(READ_ONLY_ROUTING_LIST=(N'SQL1',N'SQL2')))
GO
```

Achten Sie darauf, dass Sie für jede Instanz eine Routing-Liste konfigurieren müssen.

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

Die Routing-Liste lässt sich mit folgendem Statement anzeigen:

```
SELECT ag.name as "Availability Group", ar.replica_server_name as "When Primary Is",
       rl.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
```

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

Lastausgleich mit Read Only Routing Lists

Seit SQL-Sever 2016 können Sie die Last auch zwischen mehreren sekundären Read-Only-Rollen verteilen. Dafür müssen Sie in der Routing-Liste die Server, auf die die Last verteilt werden soll, zusätzlich noch einmal einklammern.

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

Sie können auch weitere Instanzen angeben, die nur verwendet werden sollen, wenn die Lastverteilenden Instanzen nicht verfügbar sind. Dafür müssen die weiteren Instanzen außerhalb der Klammer angeben werden, die die lastverteilenden Instanzen umschließt:

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

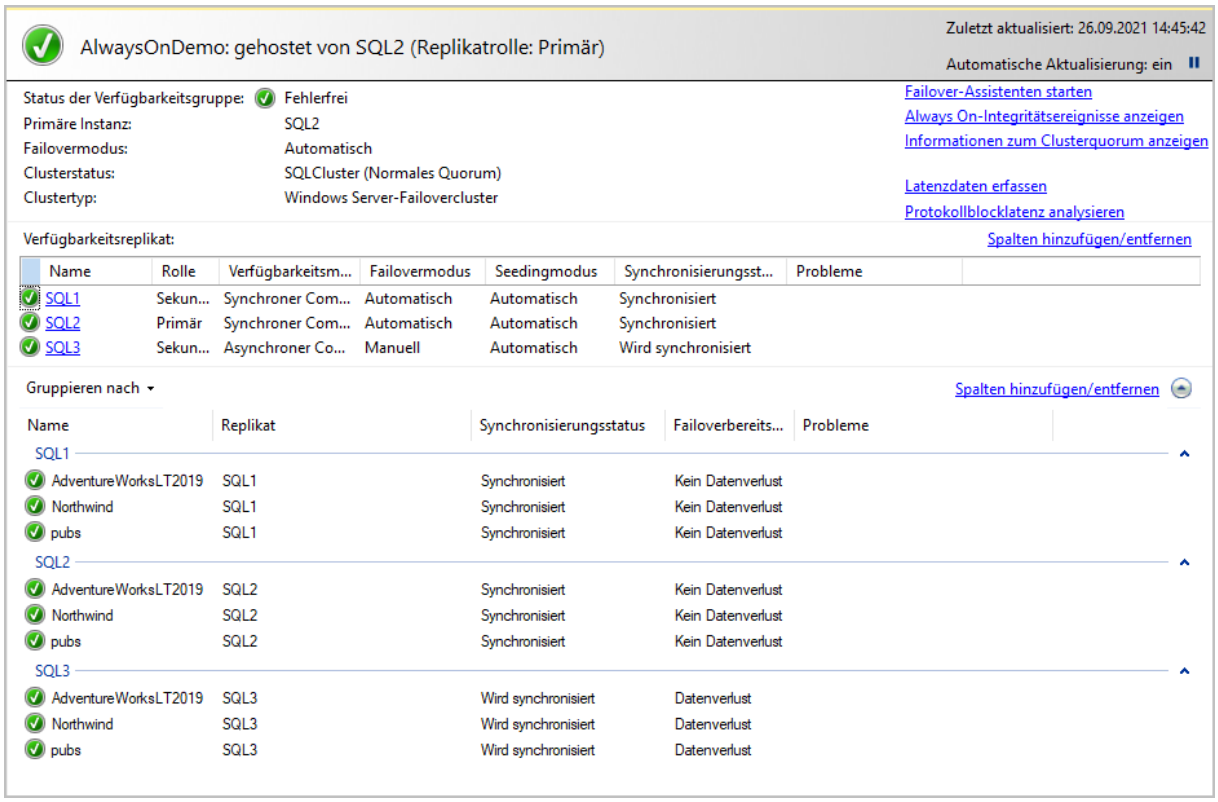
Sie können sogar für die Backup-Server eine Lastverteilung aktivieren, wenn Sie mehr als einen haben.

```
ALTER AVAILABILITY GROUP [AlwaysOnDemo]
MODIFY REPLICA ON N'SQL1' WITH
(PRIMARY_ROLE(READ_ONLY_ROUTING_LIST=((N'SQL2',N'SQL1'),(N'SQL3', ,N'SQL4)')))
GO
```

Mehr zum Thema Read-Only Routing, Lastverteilung beim Read-Only Routing und wie Sie Powershell verwenden können, um die Konfiguration vorzunehmen, finden Sie unter https://docs.microsoft.com/de-de/sql/database-engine/availability-groups/windows/configure-read-only-routing-for-an-availability-group-sql-server?view=sql-server-ver15 oder kurz https://bit.ly/3iaeJHt.

Überwachung der Always On Availability Group

Zur Überwachung steht Ihnen das Always On Dashboard zur Verfügung. Sie finden es im Management Studio im Kontextmenü der Availability Group unter "Dashboard anzeigen". Verwenden Sie immer die primäre Rolle, um alle Informationen zu erhalten.



Das Dashboard zeigt Ihnen einen Überblick über den Status Ihrer Availability Group. Von hier aus können Sie die Availability Group auch steuern. Oben rechts im Dashboard findet man die Möglichkeit, ein manuelles 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 Always On Gruppe zu überwachen. Wählen Sie "Always-On Integritätsereignisse anzeigen", öffnet sich ein neues Fenster. Standardmäßig werden hier in tabellarischer Auflistung die Ereignis-Namen und die Aufzeichnungszeit angezeigt. Die Ereignisse hier ähneln dem Windows-Ereignisprotokoll. Sie finden hier also nicht nur Fehler, sondern hier werden wesentliche Ereignisse der Availability-Groups protokolliert.

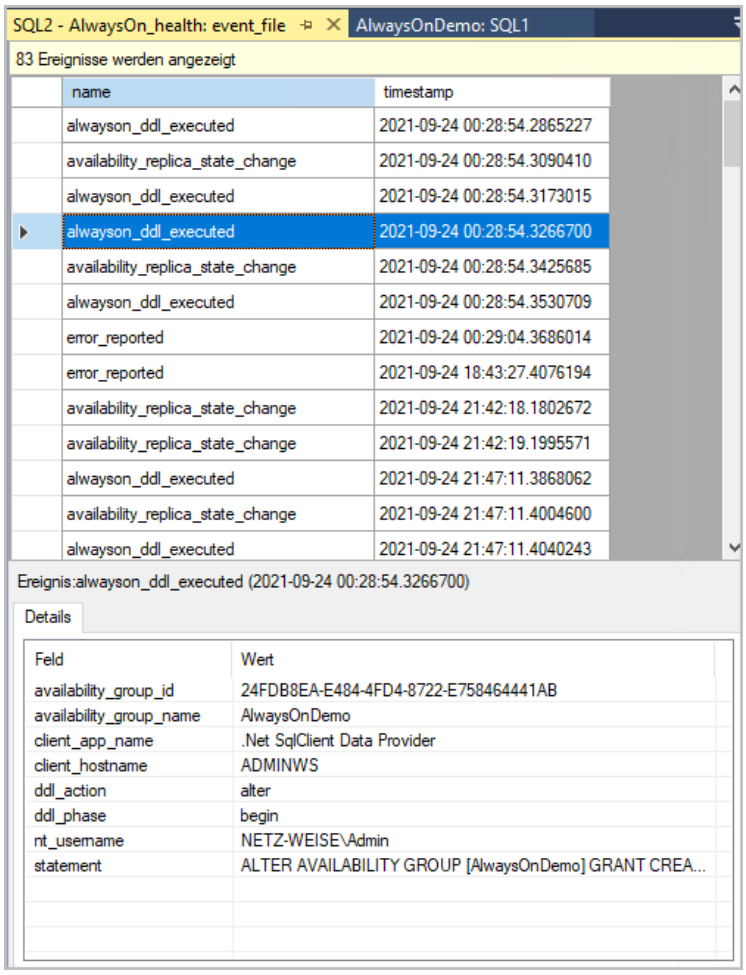


Abbildung 67 - die Ereingisse entsprechen dem Windows Ereignisprotokoll

Wählen Sie ein Ereignis aus, sehen Sie unter Details die aufgezeichneten Informationen. Die Spalten, die in der Tabellarischen Ansicht angezeigt werden, können Sie um die Felder erweitern, die Sie in der Detail-Ansicht eines Events sehen, indem Sie das Kontextmenü der Kopfzeile der Tabelle öffnen. Hier Sie einen Eintrag "Spalten auswählen". Alternativ können Sie auch die Spalte unter Details klicken und "Spalte in Tabelle anzeigen" auswählen.

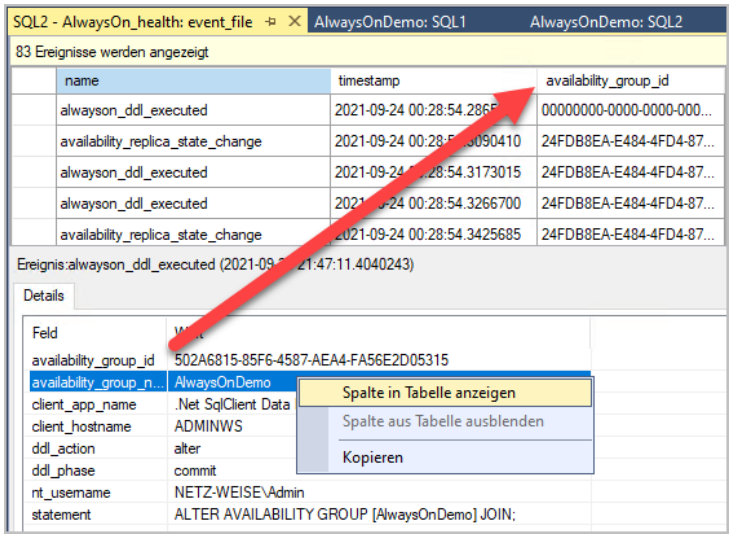


Abbildung 68 - Sie können die Spaltenansicht erweitern

Alternativ können Sie die Ereignisse auch gruppiert anzeigen lassen. Wählen Sie dazu in der Tabelle das Kontextmenü der Spalte aus, nach der gruppiert werden soll, und aktivieren Sie "Nach dieser Spalte gruppieren".

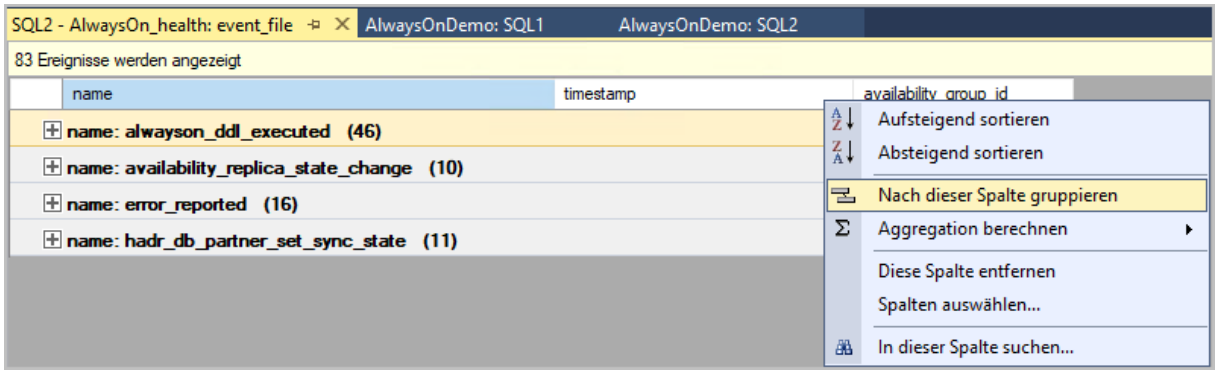


Abbildung 69 - Nach dem Namen gruppiert sieht man schnell, wie oft ein Ereignis aufgetreten ist

Sie können die Events, die aufgezeichnet werden sollen, auch anpassen. Wechseln Sie hierfür im Management-Studio in den Knoten *Verwaltung – Erweiterte Ereignisse – Sitzungen* und passen die die Eigenschaften der Sitzung "AlwaysOn_health" an.

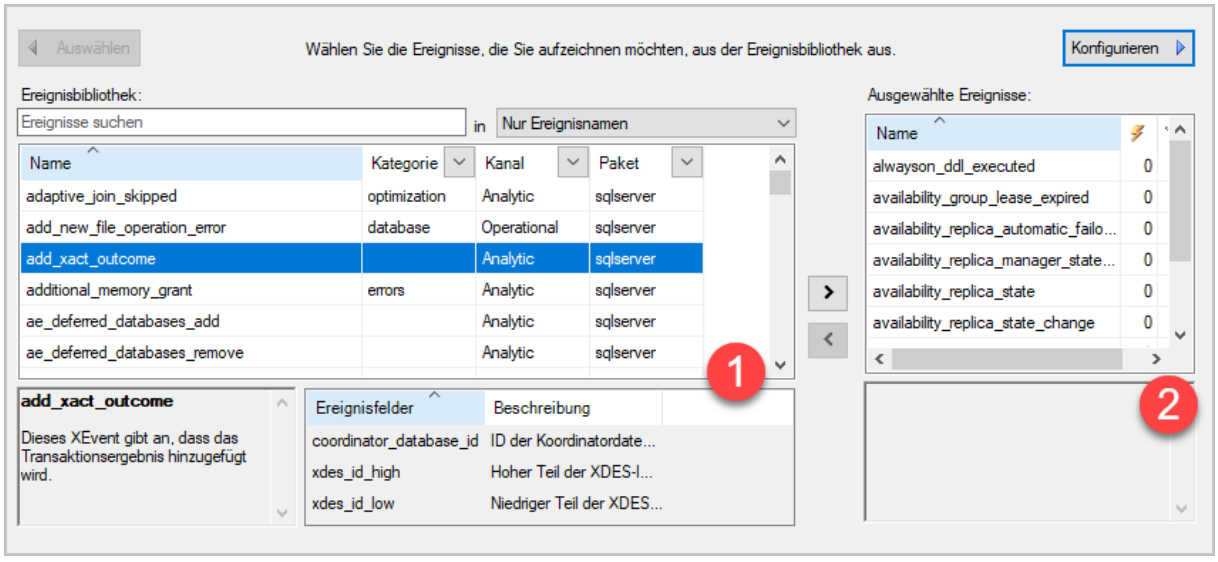


Abbildung 70 - Definition der AlwaysOn Event-Sammlung

Im Wesentlichen sehen Sie hier die Liste der Ereignisse, die überwacht werden können (1), und die Ereignisse, die in der Sitzungskonfiguration tatsächlich überwacht werden. Eine Einführung zu Extended Events im Allgemeinen finden Sie unter https://docs.microsoft.com/en-us/sql/relational-databases/extended-events/quick-start-extended-events-in-sql-server?view=sql-server-ver15 oder kurz https://bit.ly/3m24aaM. Mehr zu Always On Events gibt es z.B. bei https://www.sqlshack.com/deep-dive-into-sql-server-extended-events-the-event-pairing-target/ oder kurz https://bit.ly/39E4vup.

Um sich per SQL-Skript einen Überblick über den Status Ihrer Availability-Group zu verschaffen, können Sie auch folgendes Skript verwenden.

```
SELECT
	ar.replica_server_name,
	adc.database_name,
	ag.name AS ag_name,
	drs.is_local,
	drs.is_primary_replica,
	drs.synchronization_state_desc,
	drs.is_commit_participant,
	drs.synchronization_health_desc,
	drs.recovery_lsn,
	drs.truncation_lsn,
	drs.last_sent_lsn,
	drs.last_sent_time,
	drs.last_received_lsn,
	drs.last_received_time,
	drs.last_hardened_lsn,
	drs.last_hardened_time,
	drs.last_redone_lsn,
	drs.last_redone_time,
	drs.log_send_queue_size,
	drs.log_send_rate,
	drs.redo_queue_size,
	drs.redo_rate,
	drs.filestream_send_rate,
	drs.end_of_log_lsn,
	drs.last_commit_lsn,
	drs.last_commit_time
FROM sys.dm_hadr_database_replica_states AS drs
INNER JOIN sys.availability_databases_cluster AS adc
	ON drs.group_id = adc.group_id AND
	drs.group_database_id = adc.group_database_id
INNER JOIN sys.availability_groups AS ag
	ON ag.group_id = drs.group_id
INNER JOIN sys.availability_replicas AS ar
	ON drs.group_id = ar.group_id AND
	drs.replica_id = ar.replica_id
ORDER BY
	ag.name,
	ar.replica_server_name,
	adc.database_name;
```

Eine Availability-Group sichern

Grundsätzlich können Sie auf jeder lesenden Instanz einer Availability-Group die Datenbanken der Availability Group sichern. Komplex wird das ganze bei automatisierten Sicherungen, da bei einer Availability-Group die primäre Instanz z.B. durch ein Failover wechseln kann und man die Qual der Wahl hat, auf welchem Server das Backup ausgeführt werden soll. Folgende Dinge sollten Sie über Backups einer Availability-Group wissen:

- Die Primäre Rolle kann von jeder Datenbank der Availability Group jede Form einer Sicherung durchführen.
- Eine sekundäre (nur lesende) Rolle kann Voll- und Transaktionsprotokoll-Backups durchführen. Vollbackups können aber nur als Kopiesicherung durchgeführt werden, Differenz-Backups können gar nicht ausgeführt werden, und Transaktionsprotokoll-Sicherungen können **nicht** als Kopie-Sicherung durchgeführt werden. Das gilt sowohl für Lesende als auch nicht lesbare Replikate(!)

Um das Backup zu vereinfachen, können Sie in der Availability-Group festlegen, welche Server für das Backup verwendet werden sollen und welche nicht. Diese Konfiguration wird auf dem Server als Meta-Information hinterlegt und kann beim Backup explizit abgefragt werden – der SQL-Server erzwingt diese Konfiguration aber nicht! Wie oben erwähnt – Sie können ein Backup manuell oder auch per Skript automatisiert immer auf jeder Instanz durchführen, solange Sie die oben genannten Einschränkungen beachten. Damit die Backup-Konfiguration angewendet wird, brauchen Sie in Backup-Tool oder Skript, dass die von Ihnen festgelegten Präfenzen abfragt und anwendet.

Wenn Sie mit SQL-Server Wartungsplänen arbeiten, beachtet der Wartungsplaner die Konfiguration, ohne dass Sie noch etwas tun müssen. Sie können im Wartungsplan das SQL-Skript abfragen, dass generiert wurde, in dem Sie einen existierenden Plan öffnen, den Sicherungs-Task öffnen und dann "T-SQL anzeigen" auswählen. Das Skript für eine Vollsicherung (Kopie) sieht dann ungefähr aus wie im folgenden Skript.

```
DECLARE @preferredReplica int

SET @preferredReplica = (SELECT
[master].sys.fn_hadr_backup_is_preferred_replica('AdventureWorksLT2019'))

IF (@preferredReplica = 1)
BEGIN
    BACKUP DATABASE [AdventureWorksLT2019] TO  DISK = N'C:\Backup\Adv.bak' WITH
COPY_ONLY, NOFORMAT, NOINIT,  NAME = N'Adv', SKIP, REWIND, NOUNLOAD,  STATS = 10
END
```

Das Skript fragt hier zuerst ab, ob die Instanz das bevorzugte Replikat für die Sicherung ist und sichert nur, wenn das der Fall ist (die Bedingung hinter dem IF). Das bedeutet – das ist der wichtige Schluss der Lektion – dass Sie **auf jeder Instanz Ihrer Availability-Group, die Backups machen soll, einen Backup-Job anlegen müssen**. Richten Sie den Job nur auf dem Server ein, der normalerweise als bevorzugter Server für Backups konfiguriert ist, laufen die Backups nur so lange, wie Sie kein Failover vornehmen. Jede Instanz prüft beim Starten des Backup-Jobs, ob sie die bevorzugte Instanz ist, und sichert nur, solange die Bedingung wahr ist. Wird also Ihre bevorzugte Lese-Replika durch ein Failover zur primären Replika, und auf der primären Replika ist kein Backup-Job eingerichtet, gibt es gar keine Backups mehr!

Besser als der Datenbankwartungsplaner sind die Skripts von Ola Hallengren geeignet, um Backups (und noch einige andere Dinge, die der Wartungsplaner eher schlecht als recht macht) zu automatisieren. Sie können die Skripte unter https://ola.hallengren.com/ kostenlos herunterladen. Es handelt sich um reine SQL-Skripte, die seit über 10 Jahren bei hunderttausenden von

Unternehmen zum Einsatz kommen – sie sind also nicht nur gut getestet, sondern auch frei von Mal- oder Spyware und absolut vertrauenswürdig.

Alles, was Sie zur Installation tun müssen, ist die Datei *MaintenanceSolution.sql* direkt von der Startseite herunterzuladen. Führen Sie das Skript anschließend auf allen SQL-Servern aus, auf denen Sie die Backup-Lösung einsetzen wollen. Das Skript selbst ist völlig harmlos, es hinterlegt auf Ihren Servern nämlich nur ein paar gespeicherte Prozeduren (SQL-Skripte) und legt eine Reihe von geplanten Tasks im SQL-Server Agent an. Sie können das Skript auch gerne selbst prüfen, es ist zu 100% reiner SQL-Code.

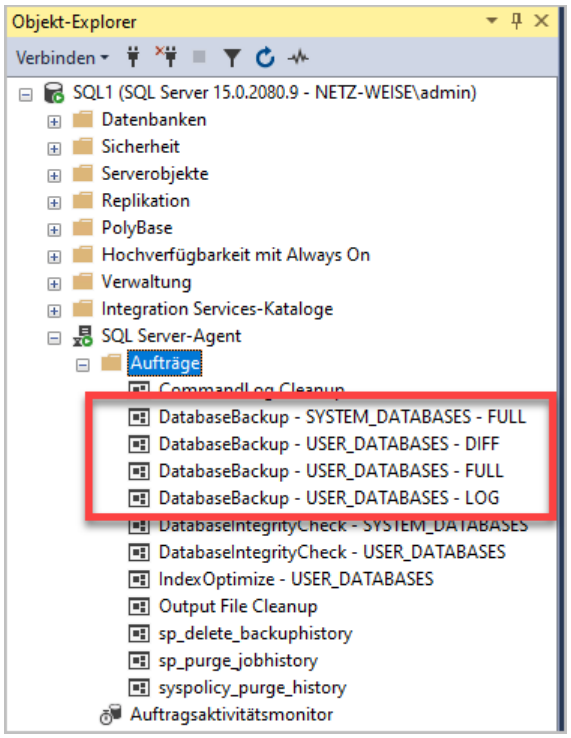


Abbildung 71 - Das Skript legt eine Reihe von geplanten Tasks an

Nachdem Sie das Skript ausgeführt haben, finden Sie im SQL-Server Agent eine Reihe von neuen Aufträgen. Wenn Sie die Aufträge nicht sehen, aktualisieren Sie die Ansicht einmal. Klappt auch das nicht, prüfen Sie, ob Sie das Skript auf der richtigen Instanz ausgeführt haben.

Die Aufträge können Sie jetzt noch anpassen und mit einem Zeitplan versehen. In der derzeitigen Konfiguration passiert nämlich noch gar nichts. Öffnen Sie hierzu den Auftrag "DatabaseBackup – User Databases – Full" mit einem Doppelklick und wechseln Sie auf die Seite Schritte. Hier finden Sie einen gleichnamigen Auftragsschritt, den Sie mit "Bearbeiten" öffnen.

Der Auftrag führt die gespeicherte Prozedur *dbo.Databasebackup* aus. Die Werte, die mit einem @ beginnen, sind Parameter, die der Prozedur übergeben werden. Hinter dem Parameter @Directory können Sie angeben, wohin die Sicherung laufen soll. Standardmäßig wird NULL (=Nichts) übergeben, was dazu führt, dass der Standard-Backup-Pfad des SQL-Servers für das Backup verwendet wird.

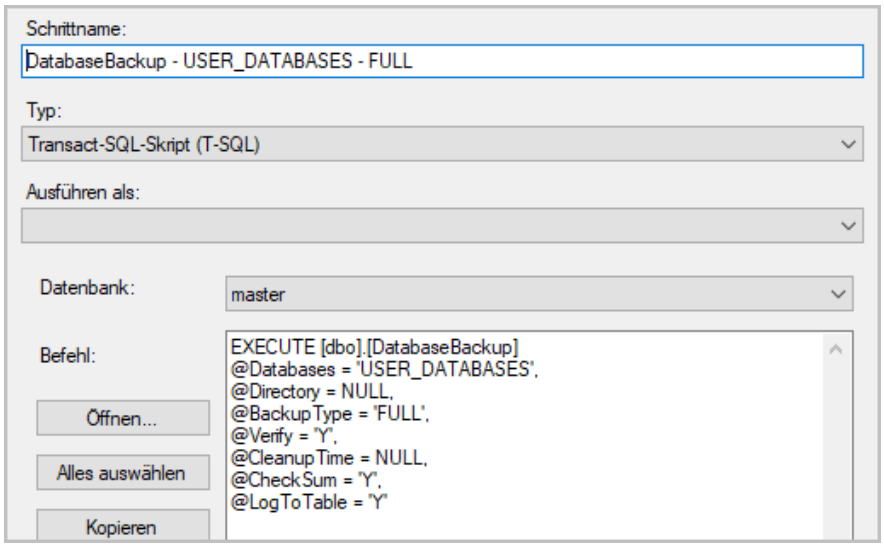


Abbildung 72 - Die Parameter des Full-Backup

Neben den standardmäßig im Auftrag angegebenen Parametern gibt es noch eine ganze Reihe weitere. Die komplette Dokumentation finden Sie unter https://ola.hallengren.com/sql-server-backup.html. Weitere hilfreiche Parameter sind:

Parameter	Bedeutung
@compress = 'Y'	Aktiviert die Backup-Komprimierung. Wird der Parameter nicht aktiviert, wird der Serverstandard verwendet. Die Komprimierung ist auf jeden Fall zu empfehlen, solange die CPU des Servers während des Backups nicht ausgelastet ist.
@Copyonly = 'Y'	Wird das Backup auf einem sekundären Replikat ausgeführt, können Backups nur als Kopie-Backup erstellt werden. Kopie-Backups schreiben keine Sicherungs-Informationen in die Datenbank zurück, die für eine Differenz-Sicherung benötigt werden.
@Encrypt = 'Y'	Das Backup wird verschlüsselt gespeichert

Abschließend müssen Sie noch einen Zeitplan für das Backup erstellen. Das geschieht auf der Seite "Zeitpläne" im Auftrag. Wählen Sie hier den Button neu aus, um einen Zeitplan zu erstellen.

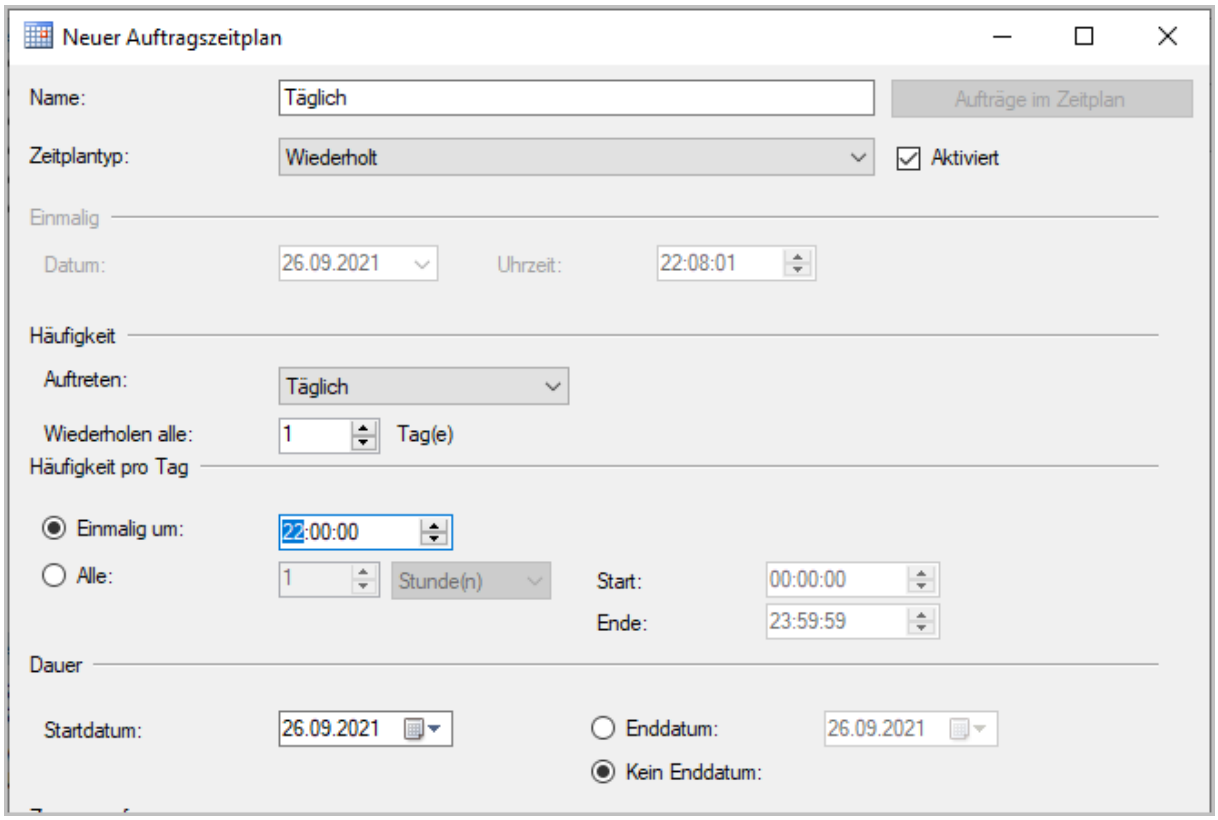
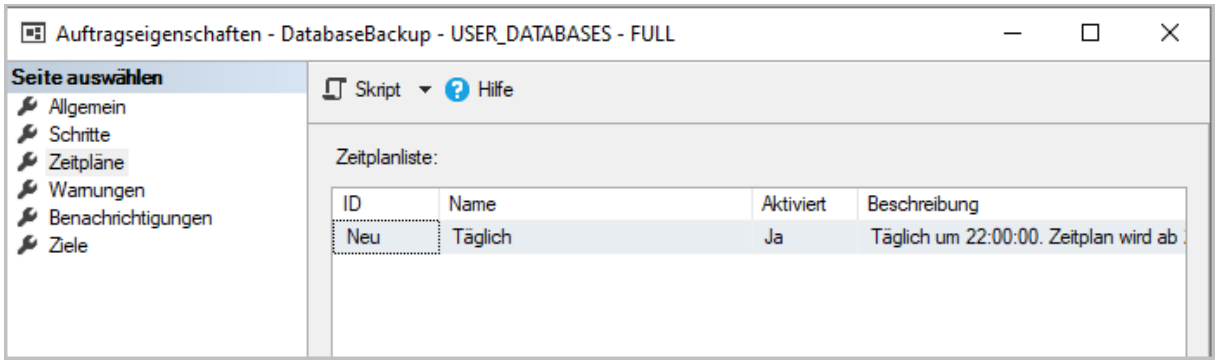


Abbildung 73 – Der Zeitplan kann sehr detailliert festgelegt werden.

Normalerweise legen Sie einen Wiederholten Zeitplan fest, bei dem Sie eine Urzeit und ein Intervall angeben können. Im Beispiel oben ist die Häufigkeit auf täglich angepasst worden. Das Backup soll einmalig um 22 Uhr starten. Anschließend übernehmen Sie den Zeitplan.



Sie können auch mehrere Zeitpläne erstellen. Im Falle des Vollbackups reicht ein Plan aber völlig aus. Den Erstellen Zeitplan sehen Sie in der Zeitplanliste.

Diesen Zeitplan aktivieren Sie jetzt auf allen Instanzen Ihrer Availability-Group. Mehr müssen Sie nicht tun – genau wie beim Wartungsplaner prüft das Skript beim Starten, ob es auf der auszuführenden Instanz läuft. Tut es das nicht, startet es den Sicherungs-Job auch nicht. Sinnvollerweise sollten Sie die Backups in einen zentralen Ordner im Netzwerk laufen lassen, da Sie Ihre Backups sonst über alle Server der Availability-Group verteilen.

Legen Sie jetzt einen Zeitplan für die Transaktionsprotokoll-Sicherung an. Die Transaktionsprotokoll-Sicherung kann auf jeder Instanz der Availability-Group gestartet werden. Die primäre Instanz wird dann über das Backup und die gesicherten Daten informiert, und diese gibt die Transaktionen im Transaktionsprotokoll dann frei. Mehr über die Funktionsweise des Log-Backups finden Sie unter

https://www.sqlshack.com/sql-server-always-on-availability-group-log-backup-on-secondary-replicas/ oder kurz https://bit.ly/3ufaBe5.

Als letztes sollten Sie noch den Datenbank-Integritäts-Check aktivieren und vor oder nach den Voll-Backups durchlaufen lassen. Hintergründe dazu finden Sie bei Brent Ozar unter https://www.brentozar.com/archive/2012/02/where-run-dbcc-on-alwayson-availability-groups/ oder kurz https://bit.ly/3zIgvFN.

System-Views für Always On

Always On 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:

Always On Availability Groups Catalog Views
https://msdn.microsoft.com/en-us/library/ff878615.aspx

Always On Availability Groups Dynamic Management Views and Functions
https://msdn.microsoft.com/en-us/library/ff877943.aspx

Dynamic Management Views and System Catalog Views (Always On Availability Groups)
https://technet.microsoft.com/en-us/library/dn135319(v=sql.110).aspx

Weiterführende Links

Differences between availability modes for an Always On availability group
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/availability-modes-always-on-availability-groups?view=sql-server-ver15

Configure a flexible automatic failover policy for an Always On availability group
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/configure-flexible-automatic-failover-policy?view=sql-server-ver15

Availability group database level health detection failover option
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/sql-server-always-on-database-health-detection-failover-option?view=sql-server-ver15

Tools to monitor Always On availability groups
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/monitoring-of-availability-groups-sql-server?redirectedfrom=MSDN&view=sql-server-ver15

Monitor Availability Groups (Transact-SQL)
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/monitor-availability-groups-transact-sql?view=sql-server-ver15

Quickstart: Extended events in SQL Server
https://docs.microsoft.com/en-us/sql/relational-databases/extended-events/quick-start-extended-events-in-sql-server?view=sql-server-ver15

Monitor SQL Server Always On Availability groups using extended events
https://www.sqlshack.com/monitor-sql-server-always-on-availability-groups-using-extended-events/

simplifying extended events management with dbatools
https://dbatools.io/xevents/

Monitor and troubleshoot availability groups
https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/always-on-availability-groups-troubleshooting-and-monitoring-guide?view=sql-server-ver15

Lease Timeouts and Health Checks in SQL Server Always On Availability Groups
https://www.sqlshack.com/lease-timeouts-and-health-checks-in-sql-server-always-on-availability-groups/

Monitor Performance for AlwaysOn Availability Groups
https://web.archive.org/web/20190114225733/https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2012/dn135338(v=sql.110)

SQL Server Native Client Support for High Availability, Disaster Recovery
https://docs.microsoft.com/en-us/sql/relational-databases/native-client/features/sql-server-native-client-support-for-high-availability-disaster-recovery?view=sql-server-ver15

Anhang A

```
--- YOU MUST EXECUTE THE FOLLOWING SKRIPT IN SQLCMD MODE.
:Connect SQL1

USE [master]
GO
CREATE ENDPOINT [Hadr_endpoint]
	AS TCP (LISTENER_PORT = 5022)
	FOR DATA_MIRRORING (ROLE = ALL, ENCRYPTION = REQUIRED ALGORITHM AES)
GO

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 [netz-weise\svcsqlcluster$]
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 SQL2

USE [master]
GO

CREATE ENDPOINT [Hadr_endpoint]
	AS TCP (LISTENER_PORT = 5022)
	FOR DATA_MIRRORING (ROLE = ALL, ENCRYPTION = REQUIRED ALGORITHM AES)

GO

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 [netz-weise\svcsqlcluster$]
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 SQL3

USE [master]
GO

CREATE ENDPOINT [Hadr_endpoint]
       AS TCP (LISTENER_PORT = 5022)
       FOR DATA_MIRRORING (ROLE = ALL, ENCRYPTION = REQUIRED ALGORITHM AES)

GO

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 [netz-weise\svcsqlcluster$]
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 SQL1

USE [master]
GO

CREATE AVAILABILITY GROUP [AlwaysOnDemo]
WITH (AUTOMATED_BACKUP_PREFERENCE = SECONDARY,
DB_FAILOVER = ON,
DTC_SUPPORT = NONE,
REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT = 0)
FOR DATABASE [AdventureWorksLT2019], [Northwind]
REPLICA ON N'SQL1' WITH (ENDPOINT_URL = N'TCP://SQL1.Netz-Weise.de:5022',
FAILOVER_MODE = AUTOMATIC, AVAILABILITY_MODE = SYNCHRONOUS_COMMIT, BACKUP_PRIORITY =
50, SEEDING_MODE = AUTOMATIC, SECONDARY_ROLE(ALLOW_CONNECTIONS = ALL)),
       N'SQL2' WITH (ENDPOINT_URL = N'TCP://SQL2.Netz-Weise.de:5022', FAILOVER_MODE =
AUTOMATIC, AVAILABILITY_MODE = SYNCHRONOUS_COMMIT, BACKUP_PRIORITY = 50, SEEDING_MODE
= AUTOMATIC, SECONDARY_ROLE(ALLOW_CONNECTIONS = ALL)),
       N'SQL3' WITH (ENDPOINT_URL = N'TCP://SQL3.Netz-Weise.de:5022', FAILOVER_MODE =
MANUAL, AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT, BACKUP_PRIORITY = 50, SEEDING_MODE =
AUTOMATIC, SECONDARY_ROLE(ALLOW_CONNECTIONS = READ_ONLY));
```

```
GO

:Connect SQL1

USE [master]
GO

ALTER AVAILABILITY GROUP [AlwaysOnDemo]
ADD LISTENER N'DemoGroup' (
WITH IP
((N'172.16.0.25', N'255.255.0.0')
)
, PORT=1433);
GO

:Connect SQL2

ALTER AVAILABILITY GROUP [AlwaysOnDemo] JOIN;
GO

ALTER AVAILABILITY GROUP [AlwaysOnDemo] GRANT CREATE ANY DATABASE;
GO

:Connect SQL3

ALTER AVAILABILITY GROUP [AlwaysOnDemo] JOIN;
GO

ALTER AVAILABILITY GROUP [AlwaysOnDemo] GRANT CREATE ANY DATABASE;
GO
```

Anhang B

```
-- Read-only routing url generation Skript.
-- Connect to each replica in your Always On cluster and run this Skript 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 Skript 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 Always On.'
end
else
begin
    print 'This SQL Server instance is NOT enabled for Always On.'
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 Skript.'
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 520 ST Mitte der 80er Jahre. Erste Erfahrungen in großen Netzwerken hat er im Systembetrieb der Volkswagen Financial Services 1999 gesammelt. 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 profesionellen IT-Umfeld spezialisiert und bietet Schulungen u.a. im Bereich Microsoft, VMWare, Linux und Oracle an.