

WID oder SQL Server als ADFS-Konfigurationsdatenbank

WISSEN

Alle Grundlagen, Szenarien und Spokes rund um ADFS an einem Ort.

› [Active Directory Federation Services](#)

BERATUNG

WID behalten, nach SQL umziehen oder gleich den Ausstieg planen – mit jemandem, der das schon oft gemacht hat.

› [Consulting zu ADFS](#)

SCHULUNG

Farm aufbauen, Datenbank wechseln, Primärknoten wiederfinden – im Workshop an echten Farmen.

› [ADFS-Schulungen](#)

Irgendwann im Leben jeder ADFS-Farm sitzt jemand in einem Besprechungsraum und sagt den Satz: „Die Windows Internal Database ist doch nur was für kleine Umgebungen, oder?“ Meist ist das der Moment, in dem das Datenbankteam hellhörig wird, eine Lizenzfrage im Raum steht und drei Wochen später eine SQL-Instanz existiert, die niemand wirklich wollte. Oder, genauso beliebt, die Gegenrichtung: Eine Farm läuft seit zehn Jahren auf WID, niemand hat je darüber nachgedacht, und als der Primärknoten stirbt, stellt sich heraus, dass die Verwaltung gleich mit gestorben ist.

Beides ist vermeidbar. Die Frage „ADFS WID oder SQL Server“ hat nämlich eine erfreulich nüchterne Antwort, und sie hängt an sehr wenigen, klar belegbaren Kriterien: wie viele Knoten und Vertrauensstellungen du hast, ob du zwei ganz bestimmte Funktionen brauchst und wie viel Hochverfügbarkeit du für das Verwalten verlangst – nicht nur für das Anmelden. Die Grundlagen zu Farm, Tokens und Vertrauensstellungen setzen wir hier voraus; die stehen auf der Pillar-Seite [Active Directory Federation Services](#). Dieser Beitrag ist die Entscheidungshilfe, plus die Anleitung für den Fall, dass die Entscheidung „SQL“ lautet und die Farm schon auf WID läuft.

FAKTEN — Die Kurzfassung für Eilige

Die ADFS-Konfigurationsdatenbank liegt entweder komplett in der Windows Internal Database (WID) oder komplett in SQL Server – ein Mischbetrieb innerhalb einer Farm ist nicht vorgesehen.

Eine WID-Farm unterstützt laut Microsoft bis zu 30 Federation-Server, sofern du 100 oder weniger Vertrauensstellungen konfiguriert hast. Darüber empfiehlt Microsoft SQL Server.

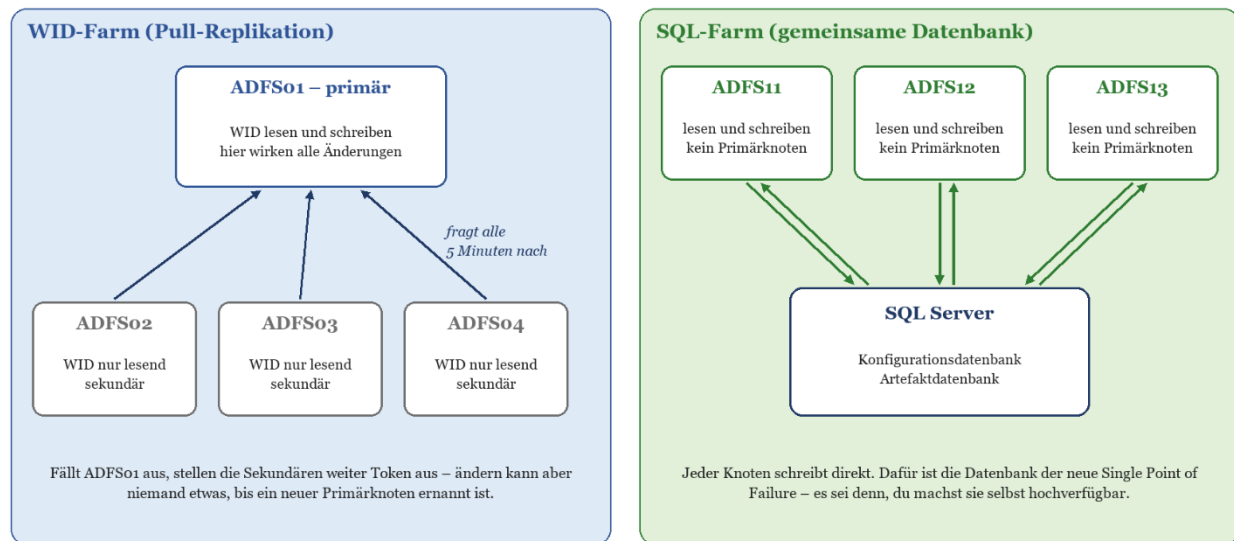
SAML-Artefaktauflösung und SAML/WS-Federation-Token-Replay-Erkennung gibt es nur mit SQL Server.

In der WID-Farm schreibt nur der Primärknoten; die Sekundären fragen ihn standardmäßig alle fünf Minuten nach Änderungen. In der SQL-Farm schreiben alle Knoten in dieselbe Datenbank.

Der Umzug von WID nach SQL ist von Microsoft unterstützt – der Rückweg ist kein Standardverfahren.

Skizze 1: Wer darf schreiben? WID-Farm und SQL-Farm nebeneinander

Links ein Chef und viele Abschreiber, rechts alle gleichberechtigt an einer gemeinsamen Datenbank



Beides ist eine Farm. Der Unterschied liegt nicht im Token, sondern darin, wo die Wahrheit wohnt und wer sie ändern darf.

Skizze 1: Die beiden Modelle. In der WID-Farm gibt es einen Chef und mehrere Abschreiber, in der SQL-Farm sind alle Knoten gleichberechtigt – und hängen gemeinsam an einer Datenbank.

Was WID kann – und wo die Grenzen wirklich liegen

Die Windows Internal Database ist ein abgespeckter SQL-Server-Kern, der mit Windows Server mitkommt, keine eigene Verwaltungsoberfläche hat und vom Betriebssystem mitgepatcht wird. Für ADFS bedeutet das: keine Lizenz, kein Datenbankteam, keine zusätzliche Abhängigkeit. Du verwaltest die Inhalte ausschließlich über die ADFS-Verwaltungskonsolle und die PowerShell-Cmdlets. Das ist kein Notbehelf, sondern der Standard – der Assistent und Install-AdfsFarm schlagen WID vor, wenn du keine SQL-Verbindungszeichenfolge angibst. Wie die Installation im Einzelnen läuft, steht in [ADFS-Farm installieren – Schritt für Schritt auf Windows Server 2022](#).

Das Primär-Sekundär-Modell

Der erste Knoten einer WID-Farm wird zum Primärknoten, im PowerShell-Jargon PrimaryComputer. Nur er hält eine beschreibbare Kopie der Konfigurationsdatenbank. Jeder weitere Knoten ist SecondaryComputer, hält eine lesende Kopie und fragt den Primärknoten in festen Abständen, ob sich etwas geändert hat. Microsoft nennt als Standard fünf Minuten; die Übertragung erfolgt inkrementell, also nur mit den Änderungen seit dem letzten Abgleich.

Daraus folgen zwei Eigenschaften, die du kennen solltest, bevor der Ernstfall sie dir erklärt:

- Die Anmeldung ist hochverfügbar. Fällt der Primärknoten aus, stellen die Sekundären weiter Token aus – mit dem Konfigurationsstand ihres letzten Abgleichs.
- Die Verwaltung ist es nicht. Solange kein Primärknoten erreichbar ist, kannst du nichts ändern: keine neue Relying Party, keine Claim Rule, kein Zertifikatswechsel.
- Du kannst einen Sekundärknoten per PowerShell zum Primärknoten befördern. Danach müssen alle übrigen Knoten auf den neuen Primärknoten zeigen – sonst replizieren sie ins Leere.
- Zwei Primärknoten gleichzeitig sind keine Redundanz, sondern ein Problem: Microsoft warnt ausdrücklich vor instabilen Farmen und möglichem Datenverlust.

```
# Auf einem beliebigen Knoten: Wer bin ich, wer ist mein Chef, wann lief der letzte Abgleich?
Get-AdfsSyncProperties

# Abgleichsintervall ändern (Sekunden) – auf dem Primärknoten ausführen
Set-AdfsSyncProperties -PollDuration 600

# Sekundärknoten zum Primärknoten befördern (auf dem künftigen Primärknoten)
Set-AdfsSyncProperties -Role PrimaryComputer

# Alle anderen Knoten auf den neuen Primärknoten zeigen lassen
Set-AdfsSyncProperties -Role SecondaryComputer -PrimaryComputerName adfs02.contoso.local
```

Listing 1: Die Synchronisationseigenschaften einer WID-Farm. Für den Rollenwechsel muss der Primärknoten vom Sekundärknoten aus per HTTP auf Port 80 erreichbar sein.

Eigenschaft	Was sie dir sagt	Worauf du achtest
Role	PrimaryComputer oder SecondaryComputer	Genau ein Knoten in der Farm darf PrimaryComputer sein
PrimaryComputerName	Wen der Sekundärknoten für den Chef hält	Muss auf allen Sekundären identisch sein und auf einen lebenden Server zeigen
LastSyncStatus	Ergebnis des letzten Abgleichs	Alles außer Erfolg ist ein Ticket wert – nicht erst beim nächsten Change
LastSyncTime	Zeitpunkt des letzten Abgleichs	Liegt er Tage zurück, ist der Knoten ein Museum mit Tokenausgabe
PollDuration	Abgleichsintervall in Sekunden	Standard fünf Minuten, selten ein Grund zum Ändern

Tabelle 1: Die wichtigsten Felder von Get-AdfsSyncProperties. Auf dem Primärknoten zeigt das Cmdlet im Wesentlichen nur die Rolle an.

TIPP – Einmal im Monat reicht – aber einmal im Monat muss sein

Lass dir die Ausgabe von Get-AdfsSyncProperties aller Sekundärknoten regelmäßig per Skript einsammeln und vergleiche PrimaryComputerName und LastSyncTime. Fehlgeschlagene Abgleiche landen zusätzlich im ADFS-Admin-Protokoll. Wie du das in dein Monitoring einbaust, zeigt [ADFS überwachen – Health Checks, Diagnose-Toolbox und Entra Connect Health](#).

Die belegten Grenzen der WID-Farm

Hier kursieren erstaunlich viele Halbwahrheiten, deshalb nur das, was Microsoft selbst dokumentiert. Eine WID-Farm unterstützt bis zu 30 Federation-Server, wenn du 100 oder weniger Vertrauensstellungen konfiguriert hast. In älteren Dokumentationsständen aus der Frühzeit von ADFS ist noch von fünf Servern die Rede – wer eine alte Anleitung liest, wundert sich also zu Recht. Hast du mehr als 100 Vertrauensstellungen und musst internen wie externen Anwendern Single Sign-on bieten, empfiehlt Microsoft SQL Server.

Dazu kommen zwei Funktionen, die es mit WID schlicht nicht gibt:

- SAML-Artefaktauflösung. Dabei holt sich die Gegenseite das Token nicht über den Browser, sondern direkt beim ADFS-Server ab. Microsoft merkt selbst an, dass das bei SAML-Anwendungen nicht verbreitet ist. Wenn eine Anwendung es aber verlangt, ist es nicht verhandelbar.
- SAML/WS-Federation-Token-Replay-Erkennung. ADFS merkt sich dabei jedes angenommene Token und verwirft einen zweiten Versuch mit demselben Token – etwa, wenn jemand am Kiosk-PC über den Verlauf die Anmeldeseite des Vorgängers erneut abschickt. Laut Microsoft ist das nur relevant, wenn ADFS Token externer Identitätsanbieter annimmt, also bei Claims Provider Trusts.

Wenn du Partnerorganisationen per Claims Provider Trust anbindest und die Replay-Erkennung dort eine echte Anforderung ist, lohnt der Blick in [Claims Provider Trust – Partnerorganisationen anbinden](#). Für eine typische Farm, die Microsoft 365 und ein paar Dutzend SAML-Anwendungen bedient, spielt keine der beiden Funktionen eine Rolle. Die OAuth-Autorisierungscode-Szenarien funktionieren übrigens mit beiden Datenbanken.

```
# Wie viele Vertrauensstellungen hat die Farm eigentlich?
$rp = (Get-AdfsRelyingPartyTrust).Count
$cp = (Get-AdfsClaimsProviderTrust).Count
"Relying Party Trusts: $rp | Claims Provider Trusts: $cp | Summe: $($rp + $cp)"

# Welche Knoten gehören zur Farm (ab ADFS unter Windows Server 2016)?
(Get-AdfsFarmInformation).FarmNodes
```

Listing 2: Die Zahlen, mit denen du die 30-Knoten- und 100-Vertrauensstellungen-Grenze prüfst. Ehrlich zählen – auch die Test-Relying-Party von 2017 zählt mit.

WARNUNG – Die Grenze ist eine Planungsgröße, kein Abgrund

Niemand bei Microsoft schaltet deine Farm ab, wenn die 101. Relying Party dazukommt. Die Grenzen beschreiben, wofür WID ausgelegt und getestet ist. Wer deutlich darüber liegt, betreibt eine Konfiguration außerhalb der Empfehlung – und diskutiert im Supportfall zuerst darüber statt über das eigentliche Problem.

Die Entscheidung: WID oder SQL Server

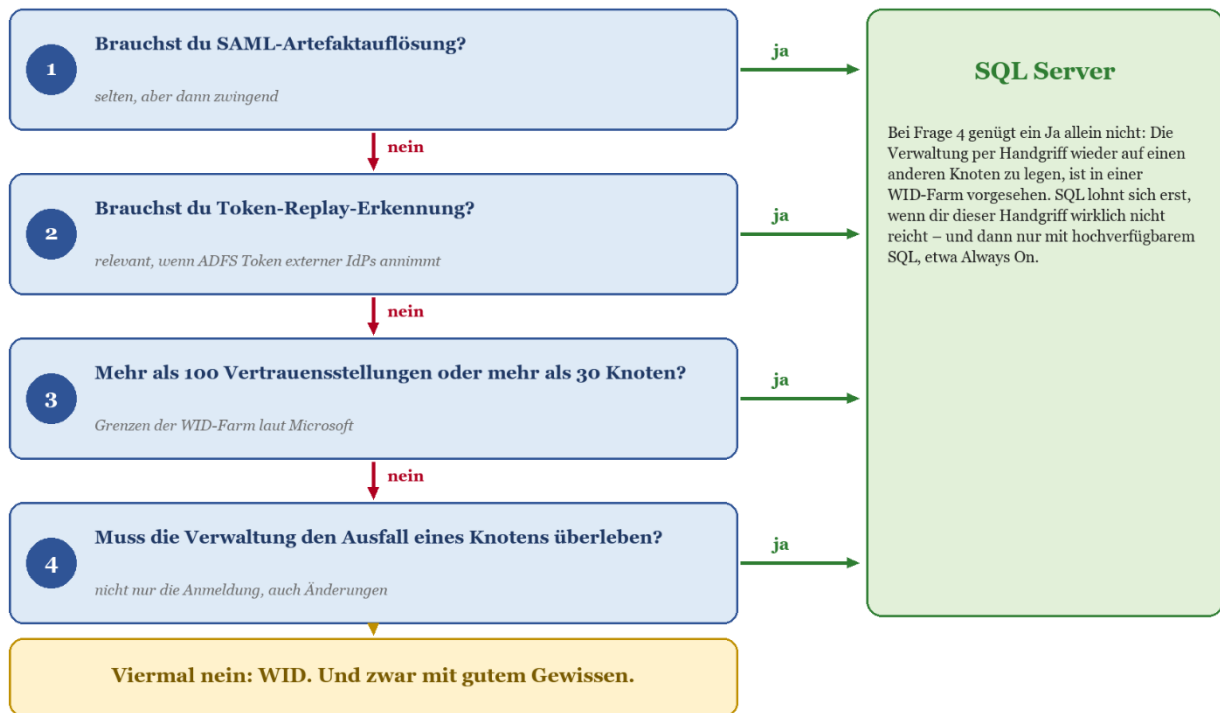
Der häufigste Fehler bei dieser Entscheidung ist nicht die falsche Wahl, sondern die falsche Frage. „Was ist besser?“ ist keine Frage, sondern eine Einladung zum Glaubenskrieg. Die richtige Frage lautet: Welche Anforderung habe ich, die WID nicht erfüllt? Gibt es keine, ist WID die bessere Wahl – weil jede Komponente, die du nicht betreibst, auch nicht ausfallen kann.

Kriterium	Windows Internal Database	SQL Server
Lizenz und Kosten	Im Betriebssystem enthalten	SQL-Lizenzen, bei Always On mehrere Instanzen
Knoten in der Farm	Bis zu 30 (bei höchstens 100 Vertrauensstellungen)	Keine WID-Grenze; Planung nach Last
Vertrauensstellungen	Empfohlen bis 100	Von Microsoft empfohlen für mehr als 100
Schreibzugriff	Nur der Primärknoten	Alle Knoten gleichberechtigt
Primärknoten fällt aus	Anmeldung läuft, Verwaltung steht bis zur Beförderung eines Sekundären	Kein Primärknoten – solange SQL läuft, verwaltet jeder Knoten
Datenbank fällt aus	Jeder Knoten hat seine eigene Kopie	Alle Knoten hängen an derselben Datenbank – ohne SQL-Hochverfügbarkeit ein Single Point of Failure
Redundanz der Datenbank	Pull-Replikation zwischen den ADFS-Knoten	Clustering, Spiegelung oder Always On Availability Groups
SAML-Artefaktauflösung	Nicht unterstützt	Unterstützt
Token-Replay-Erkennung	Nicht unterstützt	Unterstützt
OAuth-Autorisierungscode	Unterstützt	Unterstützt
Patchen und Pflege	Mit Windows Update erledigt	Eigener Patch- und Wartungszyklus, Abstimmung mit dem DBA-Team
Wer hat Zugriff?	Administratoren der ADFS-Server	Zusätzlich alle, die auf der SQL-Instanz administrieren dürfen

Tabelle 2: WID und SQL Server im direkten Vergleich. Die Funktionszeilen entsprechen der Microsoft-Dokumentation, der Rest ist Betriebserfahrung.

Skizze 2: Entscheidungsbaum – WID oder SQL Server?

Vier Fragen, in dieser Reihenfolge. Die erste ehrliche Antwort „ja“ entscheidet.



Skizze 2: Vier Fragen in fester Reihenfolge. Die ersten drei sind harte Kriterien, die vierte ist eine Frage deines Betriebsmodells.

Die unterschätzte Zeile: Wer hat Zugriff?

Die Konfigurationsdatenbank ist nicht irgendeine Datenbank. Sie enthält die komplette Vertrauensarchitektur deines Unternehmens, und bei den von ADFS selbst verwalteten Zertifikaten liegen darin auch die privaten Schlüssel des Token-Signing-Zertifikats – verschlüsselt, mit dem zugehörigen Schlüsselmateriale im Active Directory. Wer die Datenbank und dieses Schlüsselmateriale in die Finger bekommt, kann Token für jede angebundene Anwendung fälschen. Das ist der Kern dessen, was unter dem Namen Golden SAML bekannt geworden ist.

Mit WID liegt die Datenbank auf den ADFS-Servern, die du ohnehin wie Domänencontroller schützen solltest. Mit SQL Server wandert sie auf eine Instanz, auf der womöglich noch die Zeiterfassung, das Warenwirtschaftssystem und ein halbes Dutzend Datenbankadministratoren mit sysadmin-Rechten wohnen. Das ist kein Argument gegen SQL – aber ein Argument für eine eigene, entsprechend gehärtete Instanz. Mehr zu Angriffswegen und Härtung steht in [ADFS-Sicherheit: Angriffe, Härtung, Golden SAML und MFA](#).

WICHTIG – Keine geteilte Sammel-Instanz

Wenn SQL, dann eine Instanz, die nur ADFS dient und deren Administratoren denselben Schutzstatus haben wie die ADFS-Administratoren. Wer die ADFS-Datenbank auf den Sammel-SQL-Server legt, auf dem auch das Intranet-Wiki läuft, hat den Tresor in die Kantine gestellt.

Und was ist mit dem Ausstieg?

Viele Farmen laufen seit über zehn Jahren, und viele werden noch einige Jahre laufen. Trotzdem gehört die Frage auf den Tisch: Wenn du ohnehin planst, Microsoft 365 und die wichtigsten Anwendungen schrittweise nach Entra ID zu verlagern, ist ein Datenbankumzug nach SQL nur

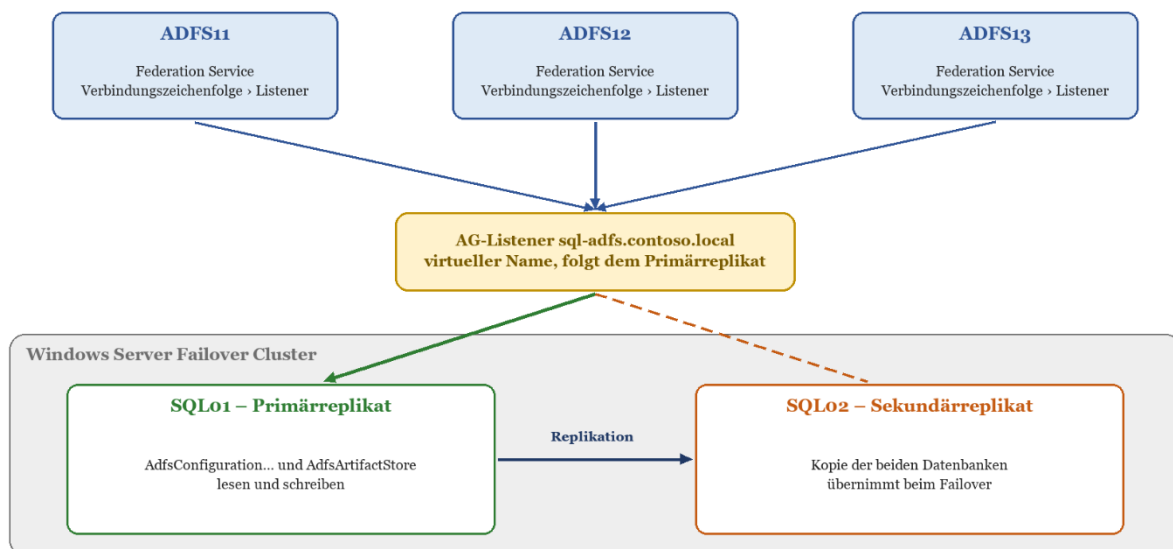
dann sinnvoll, wenn du ihn wirklich brauchst. Weniger Vertrauensstellungen bedeuten weniger Grund für SQL – und eine saubere Bestandsaufnahme ist der Anfang von beidem. Wie die aussieht, beschreibt [Relying Parties inventarisieren – die Bestandsaufnahme vor der Entra-Migration](#). Umgekehrt gilt: Eine Farm, die sich nicht verwalten lässt, weil der Primärknoten verschollen ist, lässt sich auch schlecht migrieren.

SQL Server mit Hochverfügbarkeit: Always On

Wer sich für SQL Server entscheidet, tauscht ein verteiltes Modell gegen ein zentrales. Die ADFS-Knoten werden gleichberechtigt, dafür hängen jetzt alle an einer Datenbank. Eine einzelne SQL-Instanz ohne Hochverfügbarkeit ist deshalb in fast jedem Fall ein Rückschritt gegenüber WID: Du hast eine Farm mit vier Knoten gebaut, die mit einem einzigen Server steht und fällt. Microsoft nennt als Optionen für die Datenbankschicht Failover-Clustering, Spiegelung und die Always On Availability Groups; für Letztere gibt es eine eigene Bereitstellungsanleitung.

Skizze 3: ADFS-Farm auf SQL Server mit Always On Availability Group

ADFS spricht nur mit dem Listener – welche SQL-Instanz gerade primär ist, bleibt SQL-Sache



Beide ADFS-Datenbanken gehören in dieselbe Availability Group. Eine Konfiguration ohne ihr Artefaktlager ist wie ein Schlüsselbund ohne Haustür.

Skizze 3: Die ADFS-Knoten verbinden sich mit dem Listener der Availability Group. Welche SQL-Instanz gerade das Primärreplikat hält, merken sie im Idealfall gar nicht.

Was ADFS von der Availability Group sieht

Aus Sicht der ADFS-Knoten ersetzt die Availability Group einfach die einzelne SQL-Instanz. Die Verbindung läuft über den Listener, einen virtuellen Namen, der immer auf das aktuelle Primärreplikat zeigt. Fällt die primäre SQL-Instanz aus, übernimmt ein Sekundärreplikat, der Listener zieht mit, und ADFS verbindet sich neu. Das Primärreplikat ist das einzige, das schreibt – aber anders als bei WID ist das ein Problem des SQL-Clusters, nicht deines.

Die Microsoft-Anleitung geht dabei einen etwas überraschenden Weg: Du installierst den ersten ADFS-Knoten zunächst direkt gegen eine SQL-Instanz, weil ADFS die Datenbanken bei der Einrichtung selbst anlegt. Erst danach sicherst du die Datenbanken, baust die Availability Group, stellst die Sicherungen auf dem Sekundärreplikat wieder her und biegst die Verbindungszeichenfolge auf allen ADFS-Knoten auf den Listener um. Anschließend startest du den ADFS-Dienst auf allen Knoten neu.

```
# Erster Knoten einer neuen SQL-Farm (Dienstkonto als gMSA)
Install-AdfsFarm -CertificateThumbprint $thumb `
  -FederationServiceName sts.contoso.de `
  -FederationDisplayName "Contoso Anmeldung" `
  -GroupServiceAccountIdentifier "CONTOSO\gmsa-adfs$" `
  -SQLConnectionString "Data Source=sql01.contoso.local;Integrated Security=True"

# Jeder weitere Knoten
Add-AdfsFarmNode -CertificateThumbprint $thumb `
  -GroupServiceAccountIdentifier "CONTOSO\gmsa-adfs$" `
  -SQLConnectionString "Data Source=sql-adfs.contoso.local;Integrated Security=True"

# Wenn die Datenbanken das DBA-Team anlegen soll: Skripte erzeugen lassen
Export-AdfsDeploymentSQLScript -DestinationFolder C:\Temp\AdfsSql -ServiceAccountName
"CONTOSO\gmsa-adfs$"
```

Listing 3: Neue SQL-Farm. Export-AdfsDeploymentSQLScript liefert die Skripte zum Anlegen der Datenbanken und Berechtigungen – praktisch, wenn ADFS-Admins auf der SQL-Instanz keine sysadmin-Rechte bekommen sollen.

Baustein	Was du klärst	Typischer Stolperstein
Beide Datenbanken	Konfigurations- und Artefaktdatenbank gehören in die Availability Group	Nur die Konfiguration repliziert, das Artefaktlager vergessen
Wiederherstellungsmodell	Availability Groups verlangen das vollständige Wiederherstellungsmodell	Datenbank wird im Assistenten als nicht geeignet angezeigt
SQL-Dienstkonto	Always On setzt einen Domänenaccount für den SQL-Dienst voraus	SQL läuft noch als lokales System
Dienstkonto von ADFS	Login und Datenbankrechte auf allen Replikaten	Nach dem Failover fehlt das Login auf dem neuen Primärreplikat
Listener	DNS-Name und Adresse, Erreichbarkeit von allen ADFS-Knoten	Firewall zwischen ADFS und SQL nur für die alte Instanz geöffnet
Standortübergreifend	Microsoft empfiehlt dann eine Artefaktdatenbank je Rechenzentrum und einen Hintergrund-Cache	Latenz zum entfernten Primärreplikat bremst jede Anmeldung

Tabelle 3: Die Checkliste für ADFS auf Always On. Die ersten drei Zeilen sind SQL-Voraussetzungen, die letzten drei ADFS-spezifisch.

FAKTEN — Was Microsoft zu Always On festhält

Eine Availability Group besteht aus einem Primärreplikat und einem bis vier Sekundärreplikaten, jeweils auf einer eigenen SQL-Instanz auf einem eigenen Knoten des Windows Server Failover Clusters.

ADFS verbindet sich mit dem Listener der Availability Group; die Verbindungszeichenfolge wird auf jedem ADFS-Knoten geändert, danach wird der ADFS-Dienst auf allen Knoten neu gestartet.

Der Name der Konfigurationsdatenbank hängt von der Farmversion ab. Prüfe ihn in deiner Umgebung, statt ihn aus einer Anleitung abzuschreiben.

Wer ADFS mit Always On über mehrere Standorte verteilen will, hat es übrigens nicht mehr nur mit der Datenbank zu tun, sondern auch mit Load Balancern und Health Probes, die im Fehlerfall den richtigen Standort wählen. Das Zusammenspiel ist Thema in [ADFS hinter dem Load Balancer – Health Probe, SNI und Persistenz](#) und im Überblick in [ADFS: Architektur, Einsatzszenarien und Hochverfügbarkeit](#).

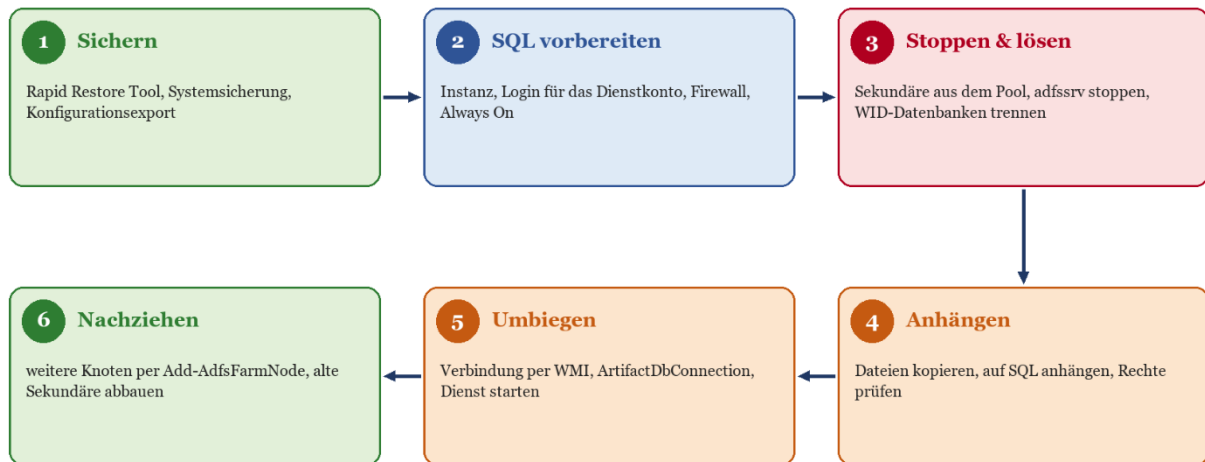
Der Wechsel von WID nach SQL Server

Microsoft unterstützt den Umzug einer bestehenden Konfigurationsdatenbank von WID nach SQL Server. Das Verfahren ist im Kern ein klassischer Datenbankumzug: Dienst stoppen, Datenbanken aus der WID lösen, auf SQL anhängen, ADFS die neue Adresse mitteilen. Die Anleitung dazu stammt noch aus der Zeit von Windows Server 2012 R2, das Prinzip hat sich aber nicht geändert.

Was sich geändert hat, sind die Datenbanknamen: Je nach Farmversion trägt die Konfigurationsdatenbank einen Versionssuffix. Deshalb fragst du die Namen ab, statt sie zu raten.

Skizze 4: Der Umzug von WID nach SQL Server in sechs Etappen

Das Wartungsfenster betrifft nur die Etappen 3 bis 5 – vorher und nachher wird geplant und aufgeräumt



Rückweg offen halten: Solange die WID-Dateien noch auf ADFS01 liegen und die Sekundären unangetastet sind, ist der Umzug ein Versuch und kein Schicksal.

Skizze 4: Der Umzug in sechs Etappen. Nur die Etappen 3 bis 5 brauchen ein Wartungsfenster; Etappe 6 kann auch Tage später folgen.

Vorbereitung: Sicherung, SQL-Instanz, Rechte

Bevor du auch nur einen Dienst anhältst, sicherst du die Farm – mit dem ADFS Rapid Restore Tool und einer Systemsicherung des Primärknotens. Wie das Rapid Restore Tool arbeitet und was es abdeckt, steht in [ADFS sichern und wiederherstellen – Rapid Restore Tool in der Praxis](#). Auf der SQL-Seite brauchst du eine Instanz, die vom Primärknoten aus erreichbar ist, ein Login für das ADFS-Dienstkonto und – wenn du Always On willst – die komplette Clustervorbereitung aus dem vorigen Abschnitt. Läuft ADFS noch unter einem klassischen Dienstkonto und soll ohnehin auf ein gMSA wechseln, dann trenne die beiden Umbauten zeitlich. Wie der Kontowechsel funktioniert, steht in [gMSA als ADFS-Dienstkonto einrichten und umstellen](#).

WARNUNG – Ein Umbau pro Wartungsfenster

Datenbankumzug, Dienstkontowechsel, Betriebssystem-Upgrade und Zertifikatstausch in einer Nacht sind eine hervorragende Methode, am nächsten Morgen nicht zu wissen, welcher der vier Schritte die Anmeldung zerlegt hat. Plane jeden Umbau einzeln, mit eigenem Test und eigenem Rückweg.

Schritt für Schritt auf dem Primärknoten

Zuerst nimmst du alle Sekundärknoten aus dem Pool des Load Balancers, sodass nur noch der Primärknoten Anfragen bekommt. Dann stoppst du auf dem Primärknoten den ADFS-Dienst adfssrv. Die Anmeldung ist ab diesem Moment unterbrochen – genau dafür ist das Wartungsfenster da. Anschließend verbindest du dich mit dem SQL Server Management Studio, als Administrator gestartet, über die Named Pipe der WID und ermittelst die Dateipfade der beiden ADFS-Datenbanken.

```
-- Verbindung im SSMS: \\.\pipe\MICROSOFT##WID\tsql\query
```

```
-- Welche ADFS-Datenbanken gibt es, und wo liegen ihre Dateien?
SELECT name, physical_name FROM sys.master_files WHERE name LIKE 'Adfs%';

-- Beide Datenbanken lösen (Namen aus der Abfrage übernehmen)
USE [master];
EXEC master.dbo.sp_detach_db @dbname = N'AdfsArtifactStore';
EXEC master.dbo.sp_detach_db @dbname = N'<Name der Konfigurationsdatenbank>';
```

Listing 4: Die WID-Datenbanken finden und lösen. Standardmäßig liegen die Dateien unter C:\Windows\WID\Data.

Die MDF- und LDF-Dateien kopierst du – kopieren, nicht verschieben – auf den SQL Server und hängst sie dort an. Die Originale bleiben auf dem Primärknoten liegen: Sie sind dein Rückweg. Auf der Konfigurationsdatenbank aktivierst du nach dem Anhängen den Service Broker, und das ADFS-Dienstkonto bekommt auf beiden Datenbanken die Rolle db_genevaservice. Die alte Anleitung erwähnt außerdem, die SQL-Firewall für den Datenbankport zu öffnen; bei einer Standardinstanz ist das TCP 1433.

```
-- Auf dem SQL Server: Dateien anhängen (Pfade anpassen)
CREATE DATABASE [<Name der Konfigurationsdatenbank>]
    ON (FILENAME = N'D:\SQLData\<Konfig>.mdf'), (FILENAME = N'D:\SQLData\<Konfig>_log.ldf')
    FOR ATTACH;
CREATE DATABASE [AdfsArtifactStore]
    ON (FILENAME = N'D:\SQLData\AdfsArtifactStore.mdf'), (FILENAME =
N'D:\SQLData\AdfsArtifactStore_log.ldf')
    FOR ATTACH;
ALTER DATABASE [<Name der Konfigurationsdatenbank>] SET ENABLE_BROKER WITH ROLLBACK IMMEDIATE;
-- Danach: Login für CONTOSO\gmsa-adfs$, Datenbankbenutzer, Rolle db_genevaservice auf beiden
Datenbanken
```

Listing 5: Anhängen auf dem SQL Server. Die Rechte prüfst du, bevor du ADFS wieder startest – nicht danach im Ereignisprotokoll.

Jetzt bekommt ADFS die neue Adresse. Die Verbindungszeichenfolge der Konfigurationsdatenbank steht in einer WMI-Klasse des ADFS-Namespaces, die Artefaktdatenbank setzt du über Set-AdfsProperties. Beides läuft in der Windows PowerShell 5.1, in der auch das ADFS-Modul zu Hause ist.

```
# Konfigurationsdatenbank umbiegen (Primärknoten)
$sts = Get-WmiObject -Namespace root/ADFS -Class SecurityTokenService
$sts.ConfigurationDatabaseConnectionString =
    "Data Source=sql-adfs.contoso.local;Initial Catalog=<Name der
Konfigurationsdatenbank>;Integrated Security=True"
$sts.Put()

# Artefaktdatenbank umbiegen
Set-AdfsProperties -ArtifactDbConnection "Data Source=sql-adfs.contoso.local;Initial
Catalog=AdfsArtifactStore;Integrated Security=True"

# Dienst starten und prüfen
Restart-Service adfssrv
(Get-AdfsProperties).ArtifactDbConnection
(Get-AdfsRelyingPartyTrust).Count
```

Listing 6: Die neue Adresse eintragen. Vorher stand in der Artefaktverbindung die Named Pipe der WID.

Danach testest du über den Primärknoten allein: Anmeldung an Microsoft 365, an einer SAML-Anwendung, an der Testseite. Stimmt die Zahl der Relying Party Trusts mit der Zählung aus Listing 2 überein und laufen die Anmeldungen, ist der schwierigste Teil geschafft.

Die Sekundärknoten: nicht umbiegen, neu aufnehmen

Bleiben die alten Sekundärknoten. Sie hängen noch an ihrer lokalen WID-Kopie und am Primärknoten, der jetzt gar keiner mehr ist. Der sauberste Weg: Du nimmst sie nicht mit, sondern baust sie neu. Entweder entfernst du die ADFS-Rolle und nimmst den Server anschließend per Add-AdfsFarmNode mit der SQL-Verbindungszeichenfolge wieder auf – oder du nutzt die

Gelegenheit und stellt gleich neue Knoten auf aktuellem Betriebssystem daneben. Das verbindet sich gut mit einem anstehenden Farm-Upgrade, das [ADFS-Farm upgraden – Farm Behavior Level anheben ohne Ausfall](#) beschreibt. Erst wenn die neuen Knoten laufen und getestet sind, kommen sie in den Pool des Load Balancers.

Knoten	Vor dem Umzug	Während des Umzugs	Danach
ADFS01	PrimaryComputer auf WID	Einziger Knoten im Pool, Dienst kurz gestoppt	SQL-Knoten, WID-Dateien als Rückweg aufbewahrt
ADFS02, ADFS03	SecondaryComputer auf WID	Aus dem Pool genommen, Dienst läuft weiter	Rolle entfernt, per Add-AdfsFarmNode neu aufgenommen
Neue Knoten	–	–	Optional: direkt gegen SQL aufnehmen, alte Knoten abbauen
Load Balancer	Alle Knoten im Pool	Nur ADFS01	Schrittweise alle SQL-Knoten, nach Test

Tabelle 4: Wer wann was ist. Ein Plan in dieser Form gehört vor dem Wartungsfenster ins Change-Ticket.

TIPP – Der Rückweg in zwei Sätzen

Solange die Sekundärknoten unverändert auf WID laufen und die Originaldateien auf ADFS01 liegen, kannst du zurück: Datenbanken in der WID wieder anhängen, Verbindungszeichenfolgen auf die alten Werte setzen, Dienst starten, Sekundäre zurück in den Pool. Genau deshalb baust du die Sekundären erst ab, wenn der SQL-Betrieb einige Tage stabil läuft.

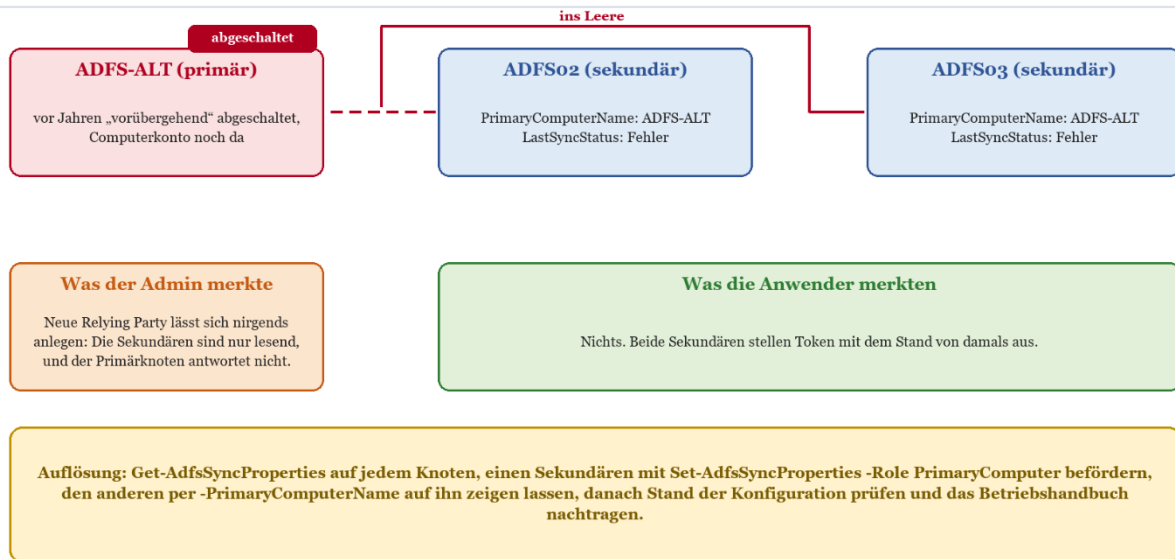
Aus der Praxis: die Farm, deren Primärknoten keiner mehr kannte

Ein mittelständisches Unternehmen, drei ADFS-Knoten, WID, Microsoft 365 und rund vierzig SAML-Anwendungen. Der Auftrag klang harmlos: eine neue Relying Party für eine SaaS-Anwendung anlegen. Der zuständige Administrator hatte es auf ADFS02 versucht, dann auf ADFS03, und beide Male bekam er zu hören, dass hier nur gelesen werde. Also die übliche Frage in die Runde: Welcher ist denn der Primärknoten? Die Antwort war ein kollektives Schulterzucken, das man sonst nur von Fragen nach dem Kennwort des Notfallkontos kennt.

Get-AdfsSyncProperties auf beiden Knoten brachte Klarheit, wenn auch keine angenehme. Beide zeigten brav auf einen Server namens ADFS-ALT. Den gab es noch als Computerkonto im Active Directory und als Eintrag in einer vergilbten Excel-Liste, aber nicht mehr als laufende Maschine. Er war Jahre zuvor im Zuge einer Virtualisierungsbereinigung „vorübergehend“ ausgeschaltet worden. Der letzte erfolgreiche Abgleich lag entsprechend lange zurück. Seitdem hatte die Farm jede Anmeldung klaglos bedient – mit einer Konfiguration, die in der Zeit eingefroren war.

Skizze 5: Die Farm, deren Primärknoten keiner mehr kannte

Anonymisiertes Praxisbeispiel – jeder Knoten hält sich für gut informiert



Skizze 5: Zwei Sekundärknoten, die treu auf einen Chef warten, der längst in Rente ist. Für die Anwender war die Farm gesund; für den Admin war sie unveränderbar.

Dass das so lange niemandem aufgefallen war, ist die eigentliche Pointe. Eine WID-Farm verzeiht einen toten Primärknoten erstaunlich lange, solange man nichts ändern will. Das Token-Signing-Zertifikat war noch einige Monate gültig – bei aktivierter automatischer Erneuerung hätte der Primärknoten den Wechsel angestoßen. Der wäre dann eben nicht passiert, und der Tag, an dem das alte Zertifikat abläuft, hätte das Problem sehr öffentlich gemacht.

Die Lösung war unspektakulär, weil der Fall glimpflich lag: ADFS02 per Set-AdfsSyncProperties -Role PrimaryComputer befördert, ADFS03 per -PrimaryComputerName auf ADFS02 umgestellt, Abgleich abgewartet, Stand geprüft. Danach die eigentliche Arbeit: Relying Parties, Claim Rules und Zertifikate gegen die Erinnerung der Fachabteilungen abgleichen, das verwaiste Computerkonto entfernen und die Rolle PrimaryComputer in das Betriebshandbuch und das Monitoring aufnehmen. Die Diskussion „Hätten wir mit SQL nicht ...?“ kam natürlich auch. Die ehrliche Antwort: Mit SQL hätte es keinen Primärknoten gegeben, der verschwinden kann – dafür eine Datenbank, die man genauso hätte vergessen können. Das Problem war nicht die Datenbank, sondern die fehlende Überwachung.

FAKTEN – Was aus dem Fall bleibt

Eine WID-Farm meldet einen fehlenden Primärknoten nicht mit Blaulicht, sondern mit fehlgeschlagenen Abgleichen im Admin-Protokoll und veralteten LastSyncTime-Werten.

Get-AdfsSyncProperties auf jedem Sekundärknoten beantwortet die Frage „Wer ist primär?“ in Sekunden – vorausgesetzt, jemand stellt sie.

Die Beförderung eines Sekundärknotens ist ein dokumentierter Handgriff, kein Notfallverfahren. Übe ihn, bevor du ihn brauchst.

Fazit

Die Frage „ADFS WID oder SQL Server“ ist in den meisten Umgebungen schnell beantwortet: WID. Sie kostet nichts, patcht sich mit Windows, trägt bis zu 30 Knoten und 100 Vertrauensstellungen und bringt die Anmeldung auch dann durch, wenn der Primärknoten ausfällt. SQL Server brauchst du, wenn du SAML-Artefaktauflösung oder Token-Replay-

Erkennung benötigst, deutlich über den WID-Grenzen liegst oder die Verwaltung ohne manuellen Rollenwechsel hochverfügbar sein muss. Dann aber richtig: mit einer eigenen, gehärteten Instanz und Always On, nicht mit einem einzelnen Sammelserver, der die ganze Farm zur Geisel nimmt.

Der Umzug von WID nach SQL ist machbar und unterstützt, braucht aber ein Wartungsfenster, einen offenen Rückweg und die Disziplin, die Sekundärknoten erst abzubauen, wenn der neue Betrieb steht. Und ganz gleich, wofür du dich entscheidest: Kenne deinen Primärknoten beziehungsweise deine Datenbank, überwache den Abgleich, und schreib es auf. Farmen sterben selten an der falschen Datenbank, aber oft am Vergessen.

Wenn du die Entscheidung oder den Umzug nicht allein treffen willst, ist das ein typischer Fall für [Consulting zu ADFS \(Active Directory Federation Services\)](#). Den Umgang mit WID-Farm, Rollenwechsel und SQL-Anbindung kannst du in den [ADFS-Schulungen](#) an echten Farmen üben. Und wer Architektur, Datenbankmodelle und Betrieb einer ADFS-Farm im Zusammenhang nachlesen möchte, findet das in [ADFS in der Praxis – das Buch im Detail](#). Steht am Ende nicht der Umzug, sondern der Ausstieg an, hilft [ADFS abschalten – die Farm sauber außer Betrieb nehmen](#).

WEITER — Passend zu diesem Beitrag

- › [ADFS-Farm installieren – Schritt für Schritt auf Windows Server 2022](#)
- › [ADFS sichern und wiederherstellen – Rapid Restore Tool in der Praxis](#)
- › [ADFS-Farm upgraden – Farm Behavior Level anheben ohne Ausfall](#)
- › [ADFS überwachen – Health Checks, Diagnose-Toolbox und Entra Connect Health](#)
- › [ADFS: Architektur, Einsatzszenarien und Hochverfügbarkeit](#)

FAQ

Wie viele ADFS-Server unterstützt eine WID-Farm?

Laut Microsoft bis zu 30 Federation-Server, sofern die Farm 100 oder weniger Vertrauensstellungen hat. Ältere Dokumente aus der Frühzeit von ADFS nennen noch fünf Server – das ist überholt.

Welche ADFS-Funktionen gehen mit WID nicht?

SAML-Artefaktauflösung und SAML/WS-Federation-Token-Replay-Erkennung. Beide setzen SQL Server voraus. Die OAuth-Autorisierungscode-Szenarien funktionieren dagegen mit beiden Datenbanken.

Was passiert, wenn der Primärknoten einer WID-Farm ausfällt?

Die Sekundärknoten stellen weiter Token aus, du kannst aber nichts mehr ändern. Mit `Set-AdfsSyncProperties -Role PrimaryComputer` beförderst du einen Sekundärknoten, anschließend stellst du alle anderen per `-PrimaryComputerName` auf ihn um.

Wie finde ich heraus, welcher Knoten primär ist?

Führe `Get-AdfsSyncProperties` auf einem Sekundärknoten aus. Die Eigenschaft `PrimaryComputerName` zeigt, wen er für den Primärknoten hält, `LastSyncStatus` und `LastSyncTime` zeigen, ob dieser Primärknoten auch antwortet.

Kann ich WID und SQL Server in derselben Farm mischen?

Nein. Die Konfigurationsdatenbank liegt vollständig entweder in der WID oder in SQL Server. Einzelne Knoten auf WID und andere auf SQL sind nicht vorgesehen.

Kann ich später von WID nach SQL Server wechseln?

Ja, Microsoft unterstützt den Umzug. Du löst die Datenbanken aus der WID, hängst sie auf dem SQL Server an, änderst die Verbindungszeichenfolgen per WMI und Set-AdfsProperties und nimmst die übrigen Knoten per Add-AdfsFarmNode neu auf.

Reicht eine einzelne SQL-Instanz für ADFS?

Technisch ja, betrieblich selten. Ohne Hochverfügbarkeit der Datenbank hängen alle ADFS-Knoten an einem Server – das ist weniger ausfallsicher als eine WID-Farm. Für produktive SQL-Farmen gehören Always On Availability Groups oder ein vergleichbares Hochverfügbarkeitsverfahren dazu.

Muss der Primärknoten für den Rollenwechsel erreichbar sein?

Für die Umstellung eines Knotens auf SecondaryComputer muss der neue Primärknoten vom Sekundärknoten aus per HTTP auf Port 80 erreichbar sein. Den toten alten Primärknoten brauchst du dafür nicht – du darfst ihn aber auf keinen Fall später wieder als zweiten Primärknoten starten.