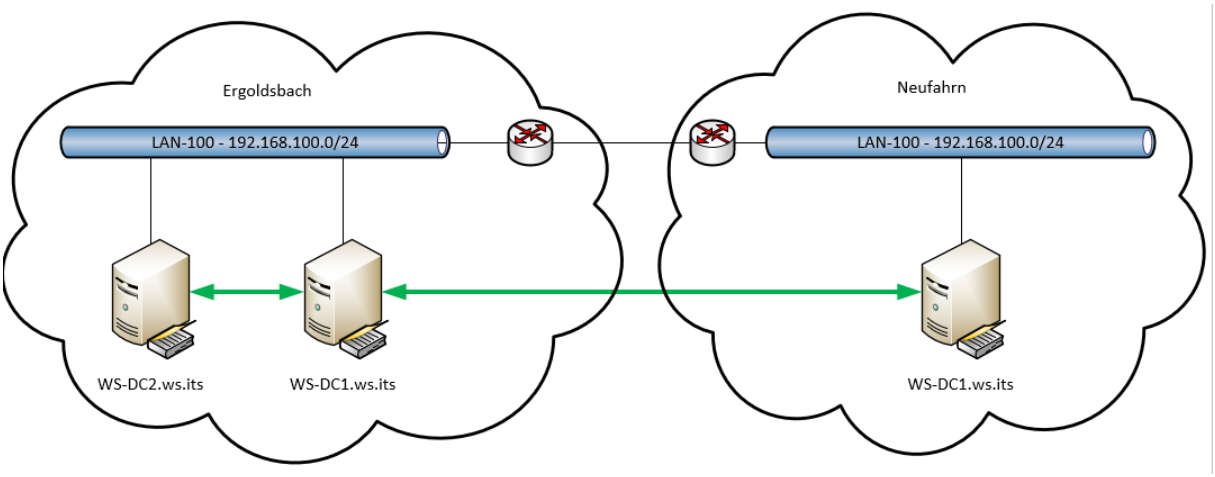
Inhalt

Zielsetzung

IST-Situation

Ich aktualisiere meine gesamte Infrastruktur auf Windows Server 2019. Heute beginne ich mit dem ersten von drei Domain Controllern. Diese stellen die Infrastruktur mit der Active Directory Domain ws.its bereit. Sie arbeiten zusätzlich als DNS-Server und als DHCP-Server.

Mein Active Directory arbeitet über zwei Standorte. Die Domain Controller haben dabei ein festes Replikations-Schema:



Alle drei Domain Controller laufen aktuell mit Windows Server 2016. Die beiden DCs in Ergoldsbach haben eine grafische Oberfläche. Der WS-DC3 ist als Server Core installiert worden. Meine Gesamtstruktur arbeitet mit der Funktionsebene Windows Server 2016.

Im Hauptstandort Ergoldsbach sind alle Server und Services so konfiguriert, dass ich einen der beiden DCs ausschalten kann. Beide Server wurden also überall als DNS-Server angegeben. Der DHCP-Service ist über DHCP-Failover ausfallsicher. Der IPHelper in meiner PFSense spricht beide DHCP-Server an.

Alle Domain Controller laufen als virtuelle Maschine – jede hat dabei ihren eigenen Hyper-V-Host darunter.

Alle zusätzlichen Services wurden im Vorfeld entfernt: Beide DCs in Ergoldsbach stellten einmal eine ADFS-Farm bereit. Die beiden Server sind aber Teil meiner Privileged Access Management Lösung und stellen deren Kernfunktion durch ein Just-Enough-Administration-Enpunkt (JEA) zur Verfügung.

Soll-Situation

Alle Domain Controller sollen mit Windows Server 2019 laufen. Die aktuelle Konfiguration hat sich bestens bewährt. Daher möchte ich am Modell meines Active Directory nichts verändern.

Die Namen und die IP-Adressen der Domain Controller möchte ich wiederverwenden. So spare ich mir den Aufwand, jeden (!) Service und jedes Gerät zu rekonfigurieren. Mal ehrlich: wo geben wir überall die speziellen IPs der Domain Controller an und wo ist der FQDN hinterlegt?

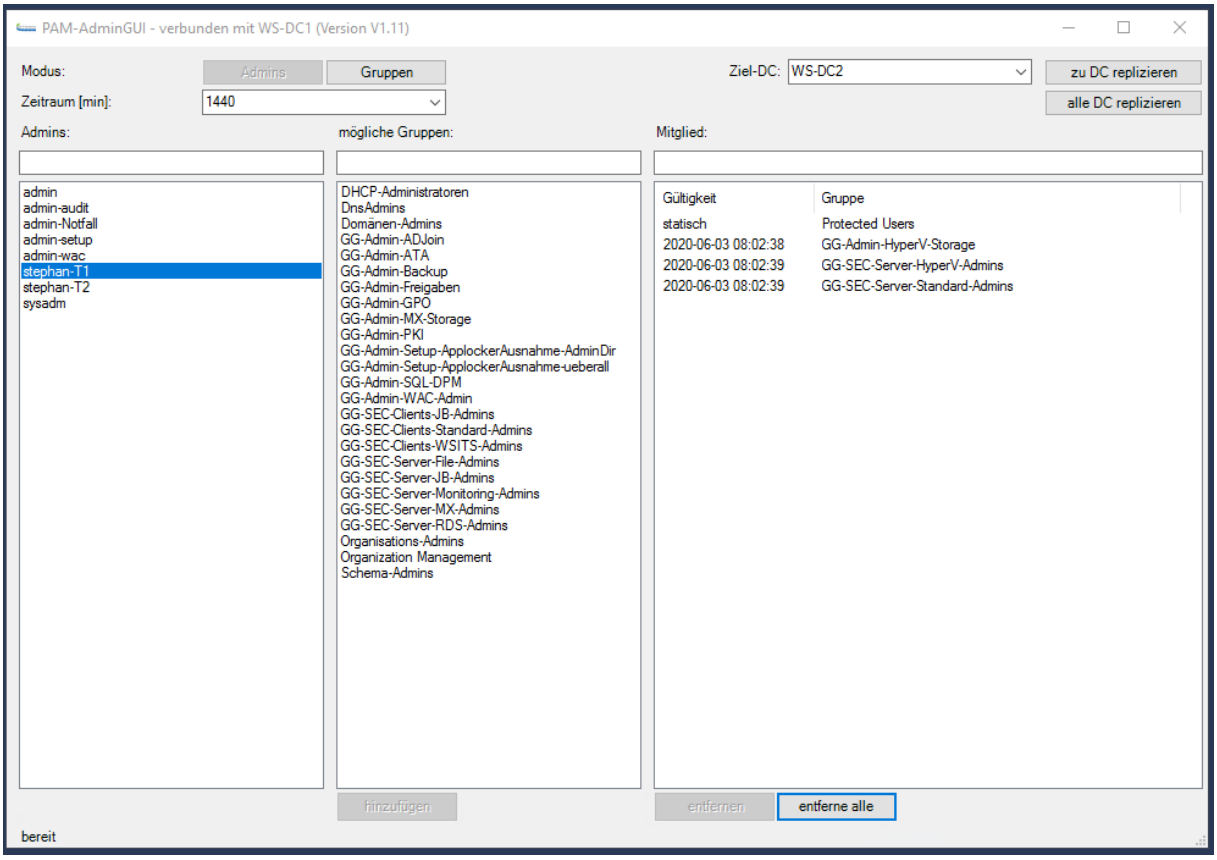
Migrationsplan

Ein Inplace-Upgrade schließe ich aus. Da gibt es einfach zu viele Probleme. Daher kommt für das Recycling der Namen und IPs nur ein Wipe & Load infrage. Bei den Domain Controllern WS-DC1 und WS-DC2 sollte eine Verfügbarkeit der Services AD, DNS und DHCP gegeben sein. Während der Migration kann auf die höhere Verfügbarkeit verzichtet werden. Die Umstellung wird während der normalen Betriebszeit durchgeführt.

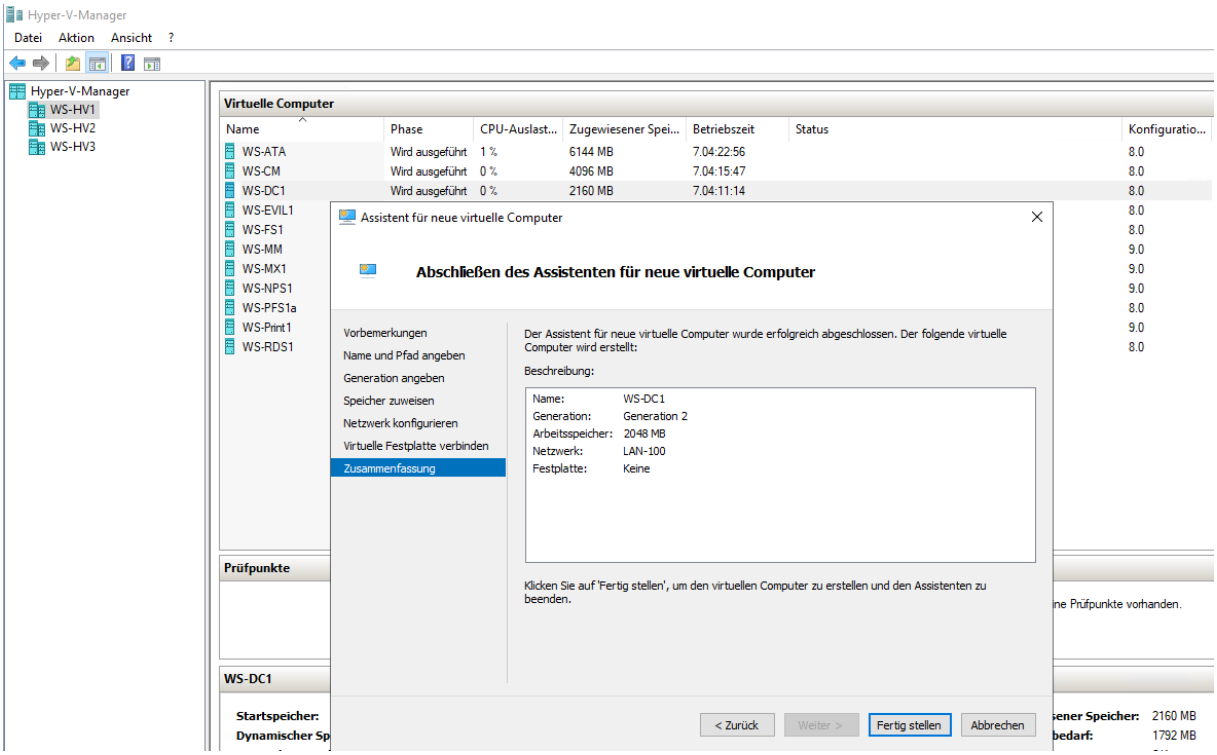
Vorbereitung

Aufbau der neuen VM

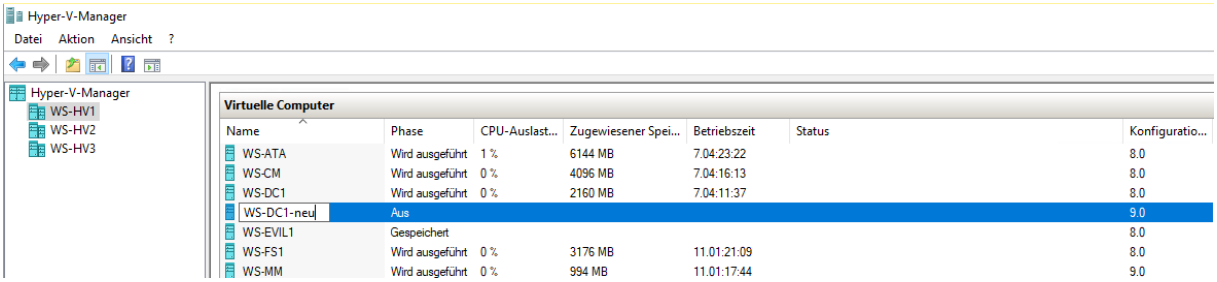
Zuerst erstelle ich mir eine neue virtuelle Maschine. Dazu bereite ich meinen Server-Account mit den passenden, temporären Gruppenmitgliedschaften vor:



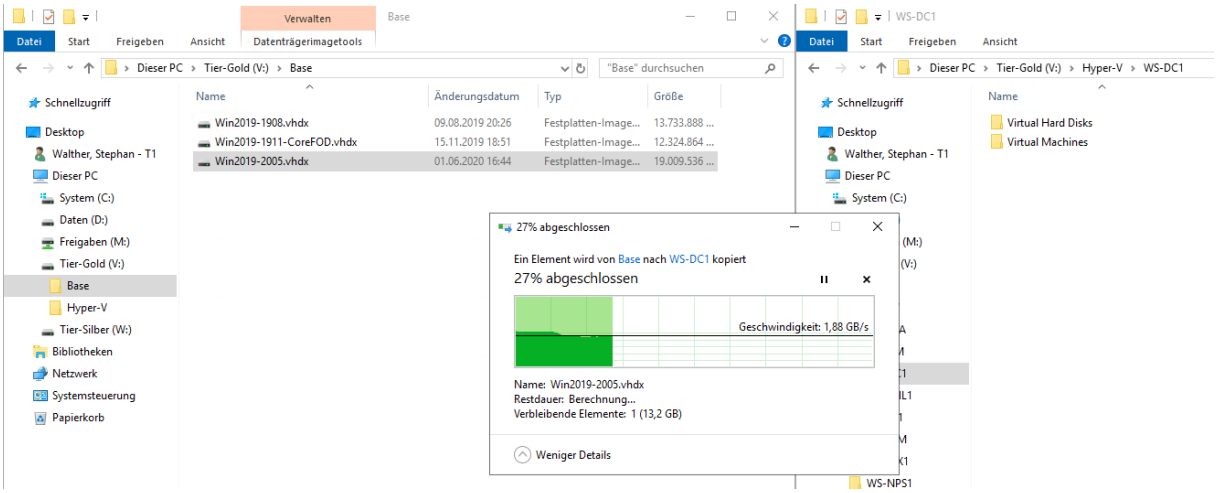
Die VM selber erstelle ich ohne große Besonderheiten. Der Name entspricht im Hyper-V schon dem der alten VM:



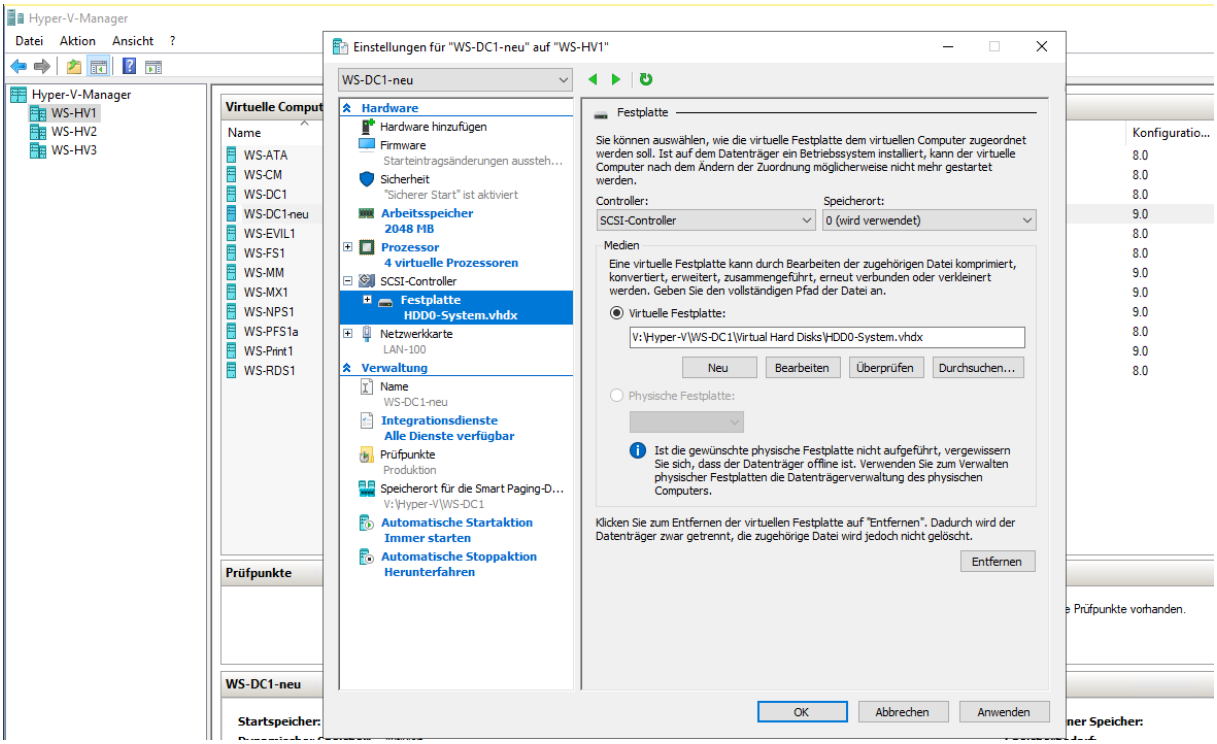
Damit ich zwischenzeitlich nicht durcheinander komme, benenne ich die neue VM um:



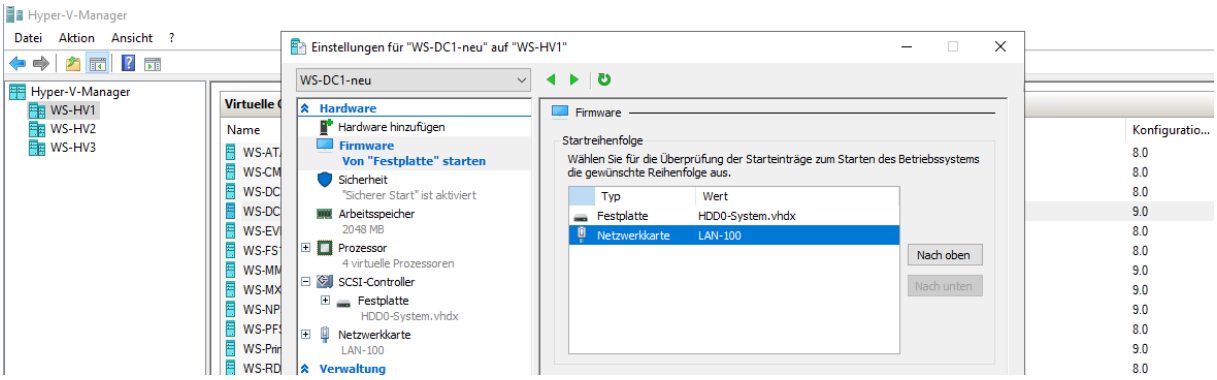
Jetzt kopiere ich mir meine neue Basefile in das Verzeichnis der VM. Darin ist ein vollwertiger Windows Server 2019 mit grafischer Oberfläche enthalten:



Jetzt kann ich die VM fertig konfigurieren. Mehr RAM, mehr CPU und die neue VHDX werden „eingebaut“. Zudem passe ich einige Optionen an:



Abschließend passe ich die Startreihenfolge des UEFI-Boots an:



Die neue VM ist fertig. Aber bevor ich weitermachen kann, muss ich mich um den alten Server kümmern.

Sichtung von Informationen auf dem alten Server

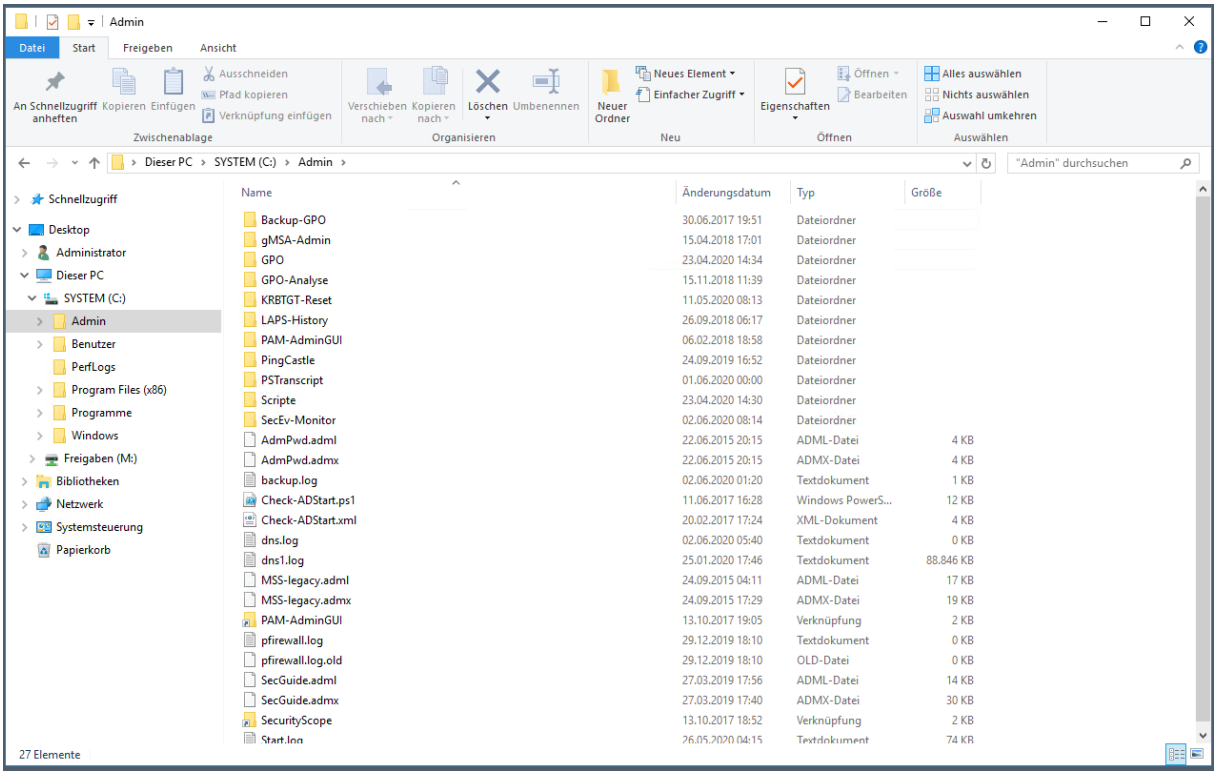
Wie bei jeder Migration sollte das alte System genau untersucht werden. Wenn man etwas übersieht, kann das nach dem Abschalten übel ausgehen. Diese Rollen und Features sind noch aktiv. Das alles gehört irgendwie zum Domain Controller:

```
PS C:\Windows\system32> Get-WindowsFeature | where installed

Display Name                                            Name                       Install State
------------                                            ----                       -------------
[X] Active Directory-Domänendienste                     AD-Domain-Services             Installed
[X] Datei-/Speicherdienste                              FileAndStorage-Services        Installed
    [X] Datei- und iSCSI-Dienste                        File-Services                  Installed
        [X] Dateiserver                                 FS-FileServer                  Installed
    [X] Speicherdienste                                 Storage-Services               Installed
[X] DHCP-Server                                         DHCP                           Installed
[X] DNS-Server                                          DNS                            Installed
[X] .NET Framework 4.6-Funktionen                       NET-Framework-45-Fea...        Installed
    [X] .NET Framework 4.6                              NET-Framework-45-Core          Installed
    [X] WCF-Dienste                                     NET-WCF-Services45             Installed
        [X] TCP-Portfreigabe                            NET-WCF-TCP-PortShar...        Installed
[X] Gruppenrichtlinienverwaltung                        GPMC                           Installed
[X] Remoteserver-Verwaltungstools                       RSAT                           Installed
    [X] Featureverwaltungstools                         RSAT-Feature-Tools             Installed
        [X] Verwaltungshilfsprogramme für die BitLoc... RSAT-Feature-Tools-B...        Installed
            [X] BitLocker-Wiederherstellungskennwort... RSAT-Feature-Tools-B...        Installed
    [X] Rollenverwaltungstools                          RSAT-Role-Tools                Installed
        [X] AD DS- und AD LDS-Tools                     RSAT-AD-Tools                  Installed
            [X] Active Directory-Modul für Windows P... RSAT-AD-PowerShell             Installed
            [X] AD DS-Tools                             RSAT-ADDS                      Installed
                [X] Active Directory-Verwaltungscenter  RSAT-AD-AdminCenter            Installed
                [X] AD DS-Snap-Ins und -Befehlszeile... RSAT-ADDS-Tools                Installed
        [X] DHCP-Servertools                            RSAT-DHCP                      Installed
        [X] DNS-Servertools                             RSAT-DNS-Server                Installed
        [X] Tools für Dateidienste                      RSAT-File-Services             Installed
            [X] DFS-Verwaltungstools                    RSAT-DFS-Mgmt-Con              Installed
[X] Unterstützung für die SMB 1.0/CIFS-Dateifreigabe    FS-SMB1                        Installed
[X] Windows Defender-Features                           Windows-Defender-Fea...        Installed
    [X] Windows Defender                                Windows-Defender               Installed
    [X] GUI für Windows Defender                        Windows-Defender-Gui           Installed
[X] Windows PowerShell                                  PowerShellRoot                 Installed
    [X] Windows PowerShell 5.1                          PowerShell                     Installed
    [X] Windows PowerShell ISE                          PowerShell-ISE                 Installed
[X] Windows Server-Sicherung                            Windows-Server-Backup          Installed
[X] WoW64-Unterstützung                                 WoW64-Support                  Installed
```

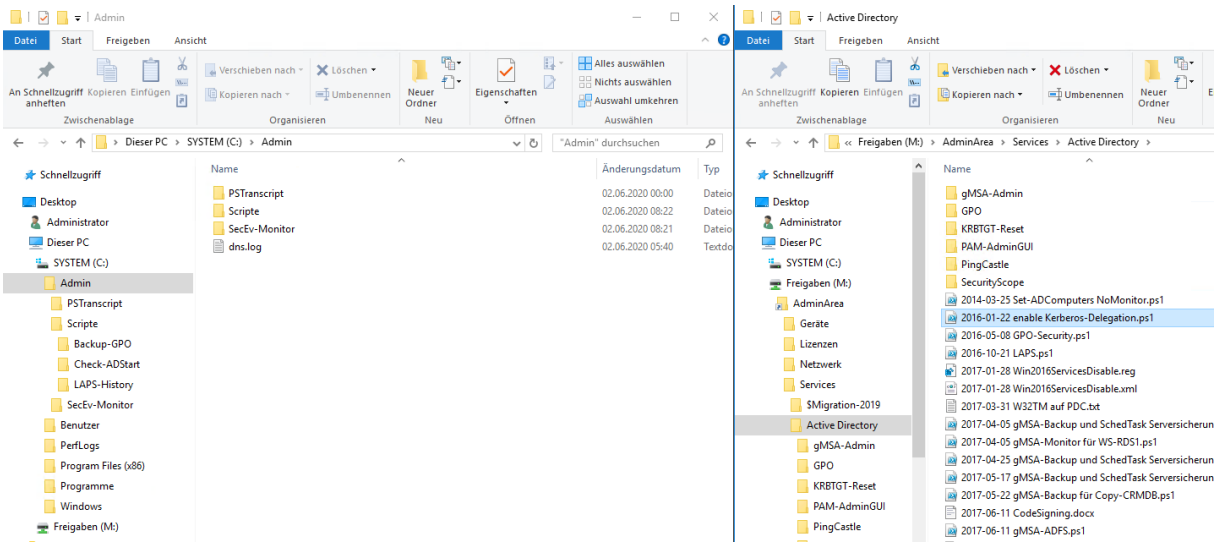
Der Server hat nur eine sichtbare Partition. Wie bei meinen anderen Servern habe ich hier meine Ablage im Verzeichnis C:\Admin aufgebaut.

Praxistipp:
Die Administratoren einer Infrastruktur sollten auf allen Servern eine gleichartige Form der Datenablage aufbauen. So kann man sich einfacher zurechtfinden und auch Migrationen wie diese hier werden vereinfacht.

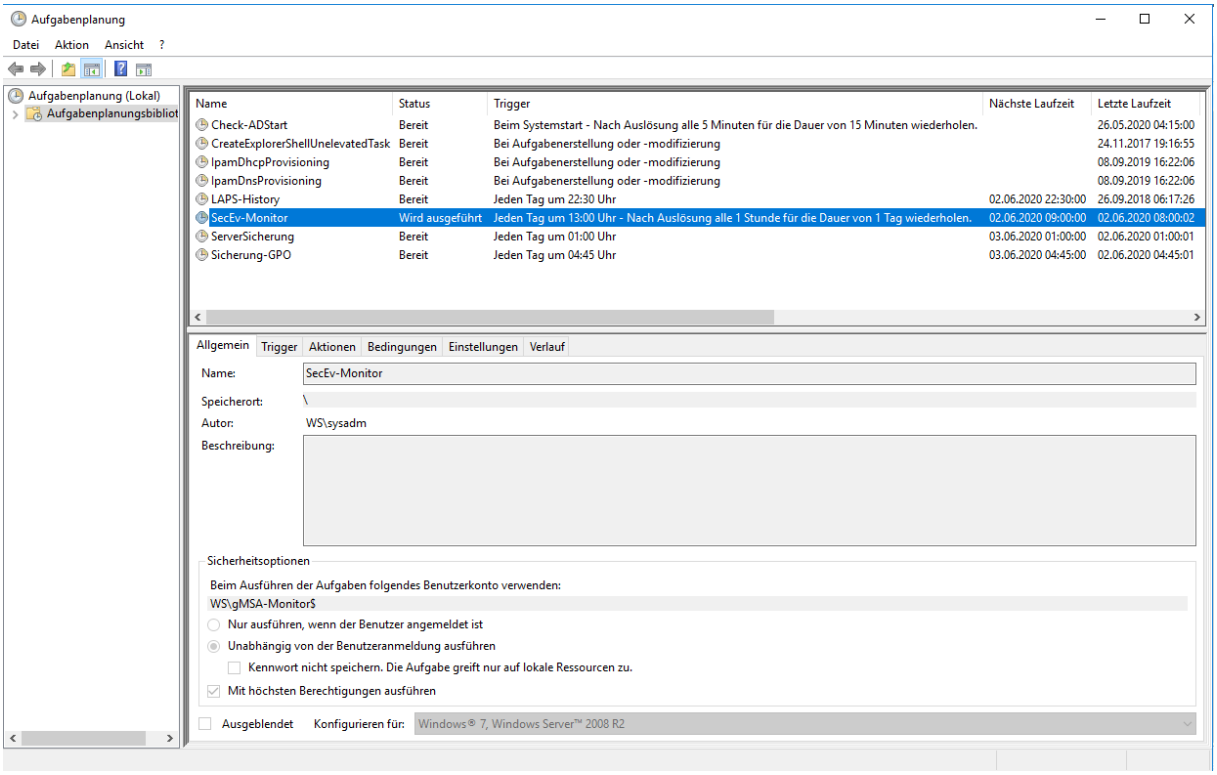
In diesem Verzeichnis sind einige Scripte und Logfiles vorhanden. Da muss ich vorab aufräumen:



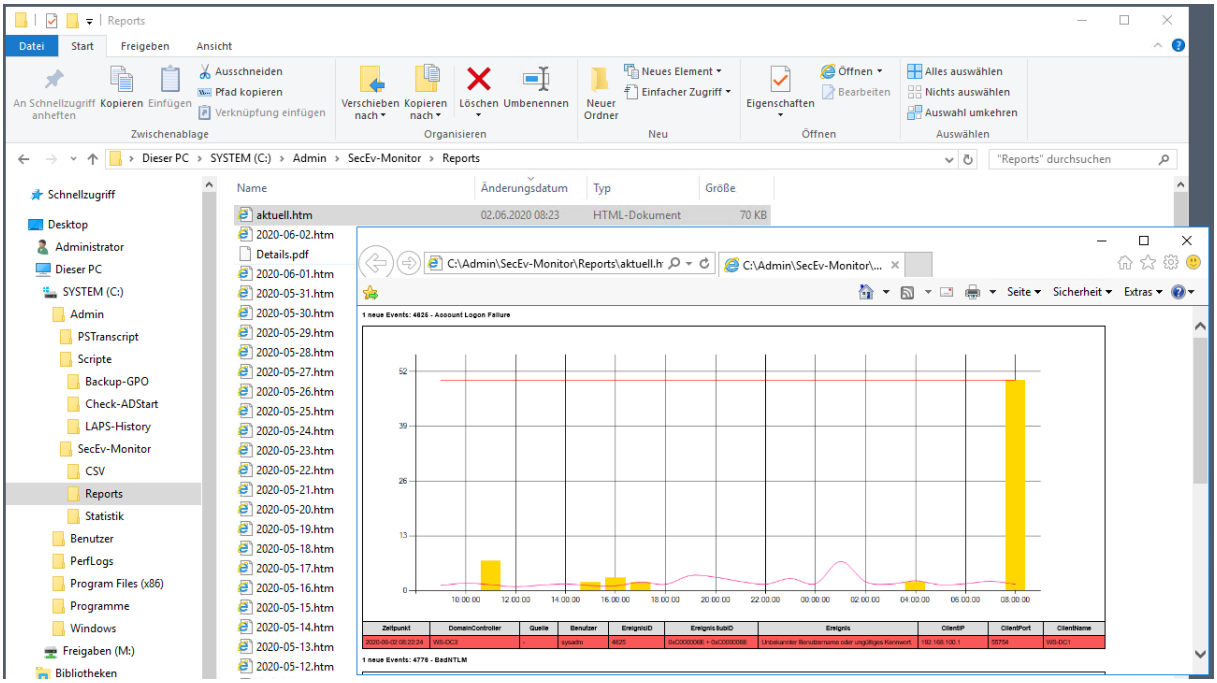
Ich verschiebe etliche Scripte auf mein zentrales Admin-Share auf den Fileservern. Die anderen Scripte verschiebe ich in einen neuen Ordner C:\Admin\Scripte:



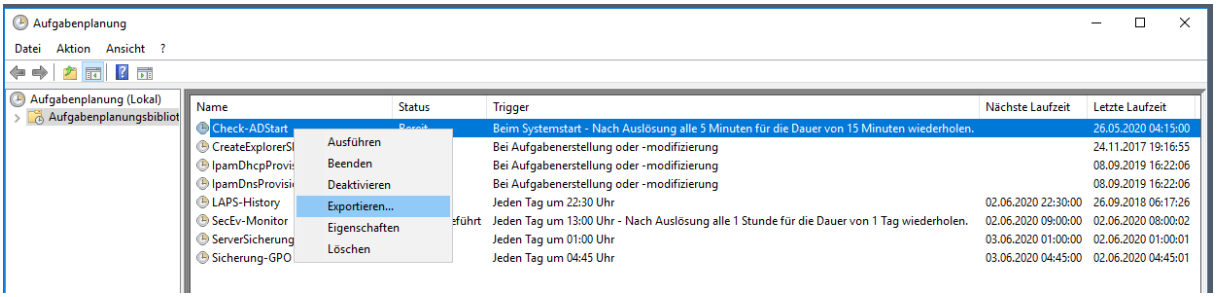
Diese lokalen Scripte werden automatisch durch geplante Aufgaben gestartet. Eine davon ist mein SecEv-Monitor:

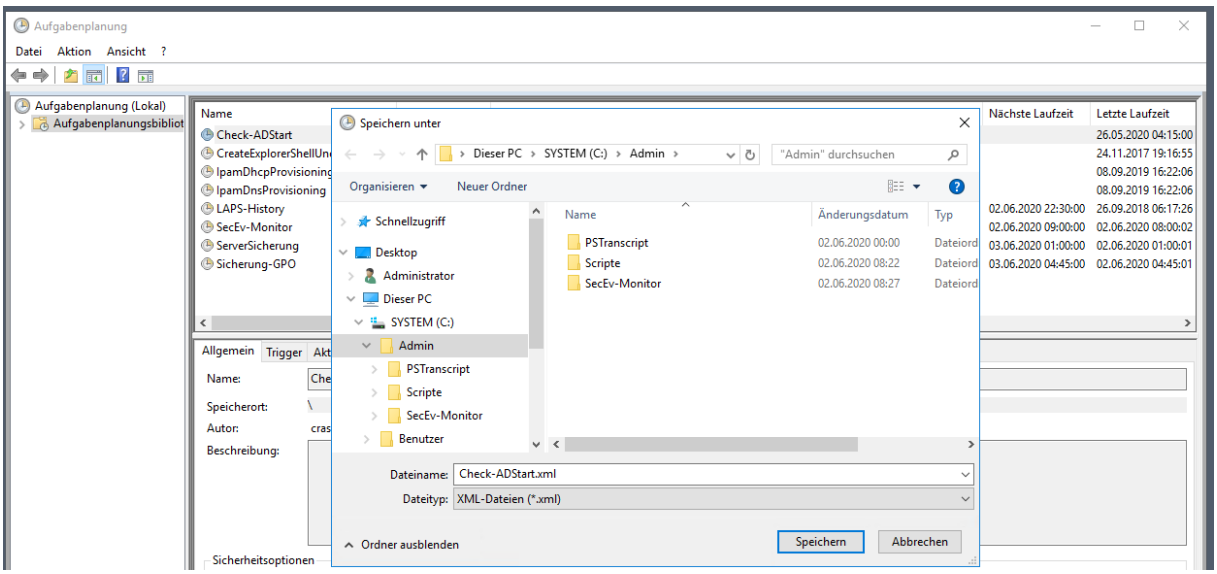


SecEv ist dabei meine Abkürzung für SecurityEvent. Das Script analysiert die Sicherheits-Eventlogs aller Domain Controller und filtert dabei interessante, sicherheitsrelevante Events heraus. Diese werden protokolliert und analysiert. Bei Bedarf kann das Script Warnmeldungen ausgeben:

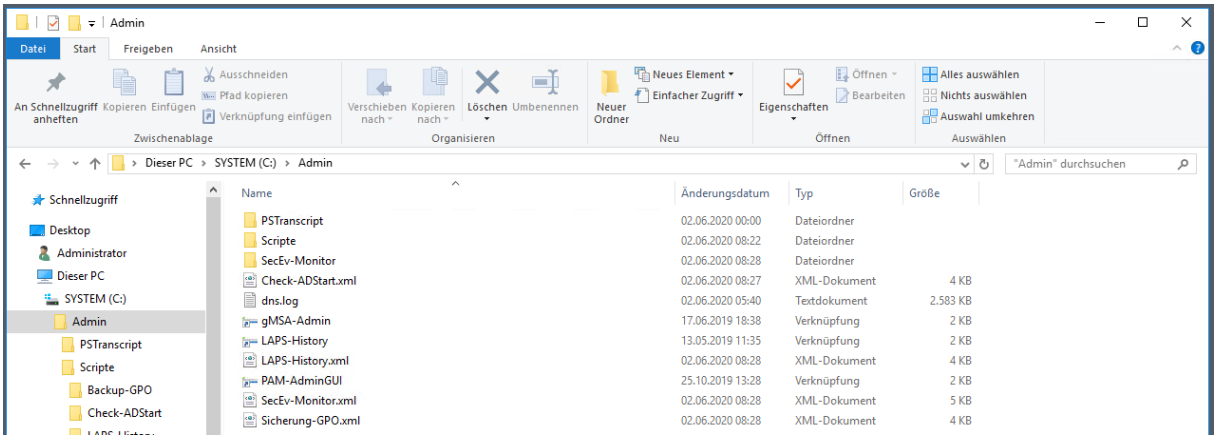


Das Script muss ich vom Server exportieren. Das mache ich nach der Datensichtung. An dieser Stelle kann ich aber die vielen Aufgaben als XML-Datei exportieren:

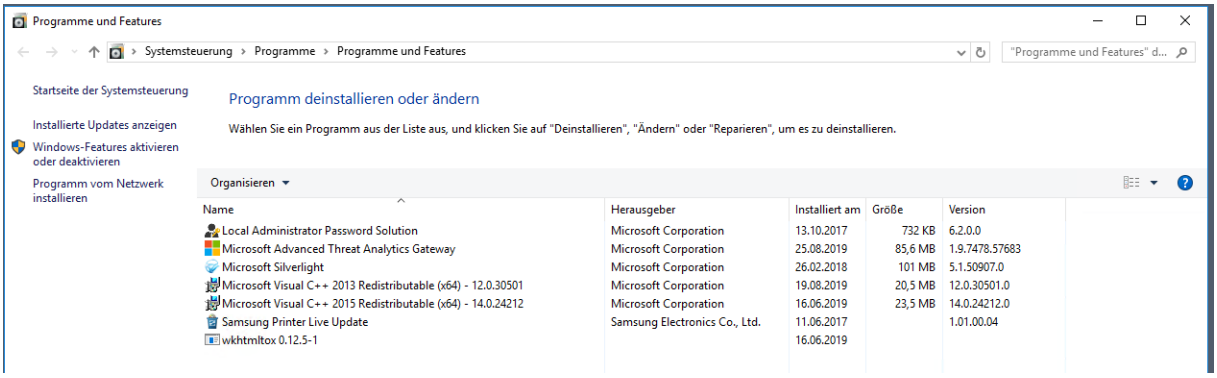




Diese Dateien kann ich dann auf dem neuen Server wieder importieren. Das spart viel Zeit:



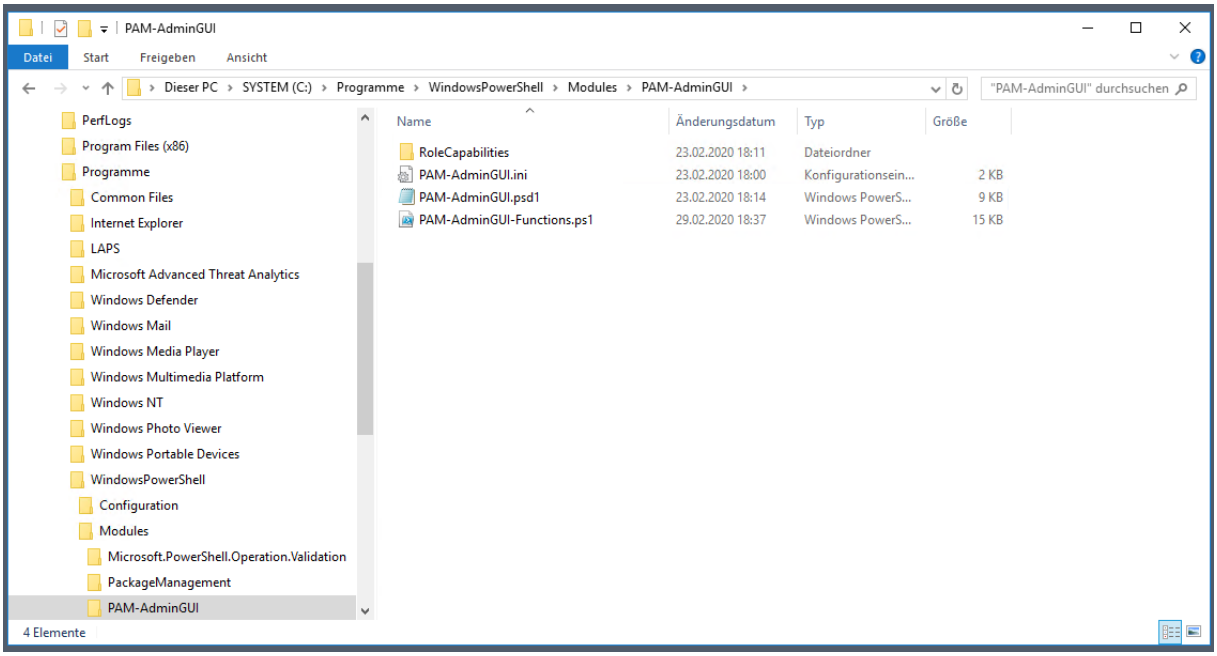
Dann durchsuche ich die lokal installierten Anwendungen. Hier ist ein Microsoft ATA Gateway installiert. Damit kann mein Monitor-Server weitere Analysen durchführen. Auf diesem Server ist das aber nicht mehr erforderlich. Dann sehe ich noch LAPS, mit dem ich die lokalen Adminkennwörter meiner Memberserver und Clients aus der AD-Datenbank auslesen kann. Und ein html-to-pdf-Converter ist vorhanden. Der gehört zu meinem SecEv-Monitor:



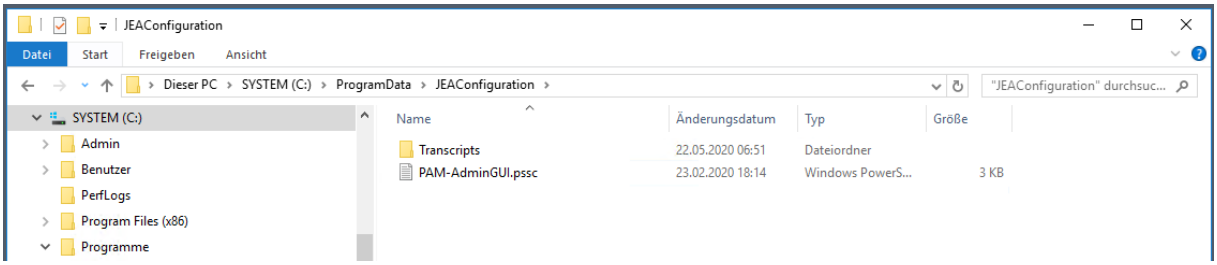
Die IP-Konfiguration ist sehr wichtig. Die muss ich später 1:1 übertragen:

```
PS C:\> ipconfig /all

Windows-IP-Konfiguration

   Hostname  . . . . . . . . . . . . : WS-DC1
   Prim„res DNS-Suffix . . . . . . . : ws.its
   Knotentyp . . . . . . . . . . . . : Peer-Peer
   IP-Routing aktiviert  . . . . . . : Nein
   WINS-Proxy aktiviert  . . . . . . : Nein
   DNS-Suffixsuchliste . . . . . . . : ws.its

Ethernet-Adapter Ethernet:

   Verbindungsspezifisches DNS-Suffix:
   Beschreibung. . . . . . . . . . . : Microsoft Hyper-V Network Adapter
   Physische Adresse . . . . . . . . : 00-15-5D-F9-A7-0B
   DHCP aktiviert. . . . . . . . . . : Nein
   Autokonfiguration aktiviert . . . : Ja
   Verbindungslokale IPv6-Adresse  . : fe80::6d07:88ab:4243:7fa%4(Bevorzugt)
   IPv4-Adresse  . . . . . . . . . . : 192.168.100.1(Bevorzugt)
   Subnetzmaske  . . . . . . . . . . : 255.255.255.0
   Standardgateway . . . . . . . . . : 192.168.100.252
   DHCPv6-IAID . . . . . . . . . . . : 50337117
   DHCPv6-Client-DUID. . . . . . . . : 00-01-00-01-26-18-12-3C-00-15-5D-F9-A7-0B
   DNS-Server  . . . . . . . . . . . : ::1
                                       192.168.100.2
                                       192.168.100.1
                                       127.0.0.1
   NetBIOS ber TCP/IP . . . . . . . : Aktiviert

Tunneladapter isatap.{A6174A8D-92B9-4E3C-B338-4740B4C720C6}:

   Medienstatus. . . . . . . . . . . : Medium getrennt
   Verbindungsspezifisches DNS-Suffix:
   Beschreibung. . . . . . . . . . . : Microsoft ISATAP Adapter #2
   Physische Adresse . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP aktiviert. . . . . . . . . . : Nein
   Autokonfiguration aktiviert . . . : Ja
```

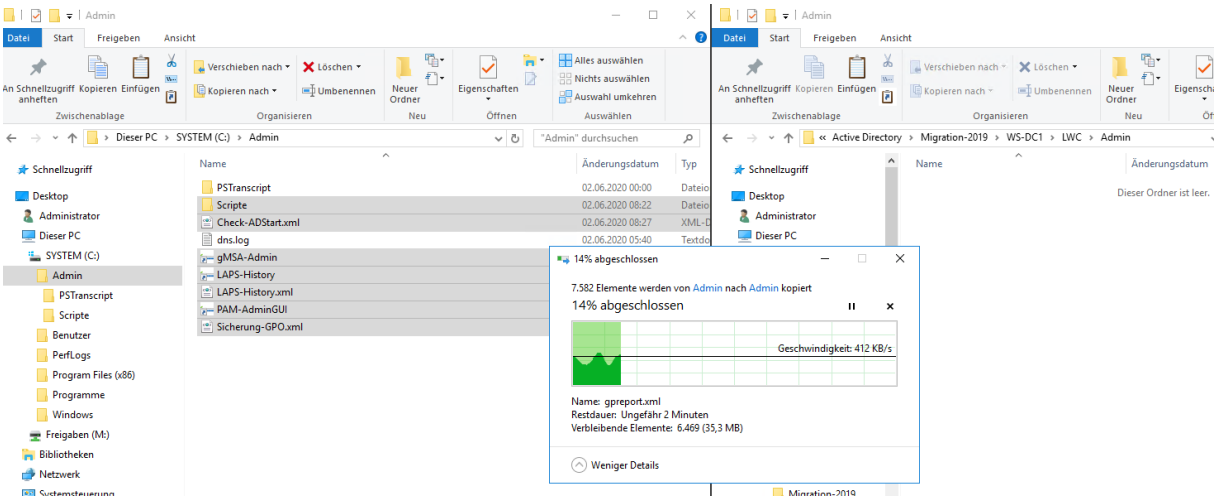
Wie bereits erwähnt, ist mein WS-DC1 für mein PAM-Script das Remote-Ziel. Das dazugehörige Script mit den Proxyfunktionen finde ich im Programmverzeichnis:



Der JEA-Endpunkt ist unter C:\Programdata registriert:



Ich kopiere alle Scripte und Aufgaben-Exporte auf meinen Fileserver:

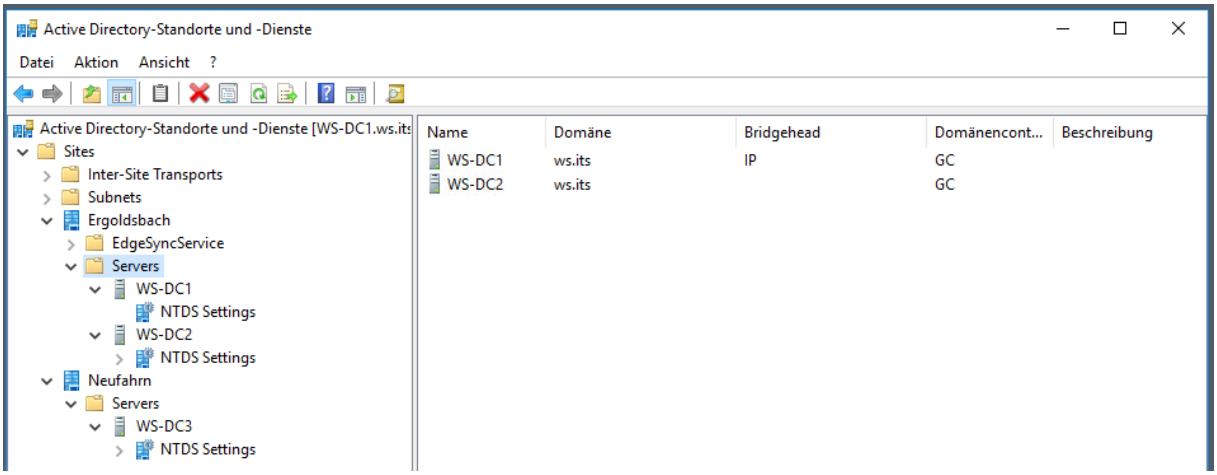


aktuelle Konfiguration des Active Directory

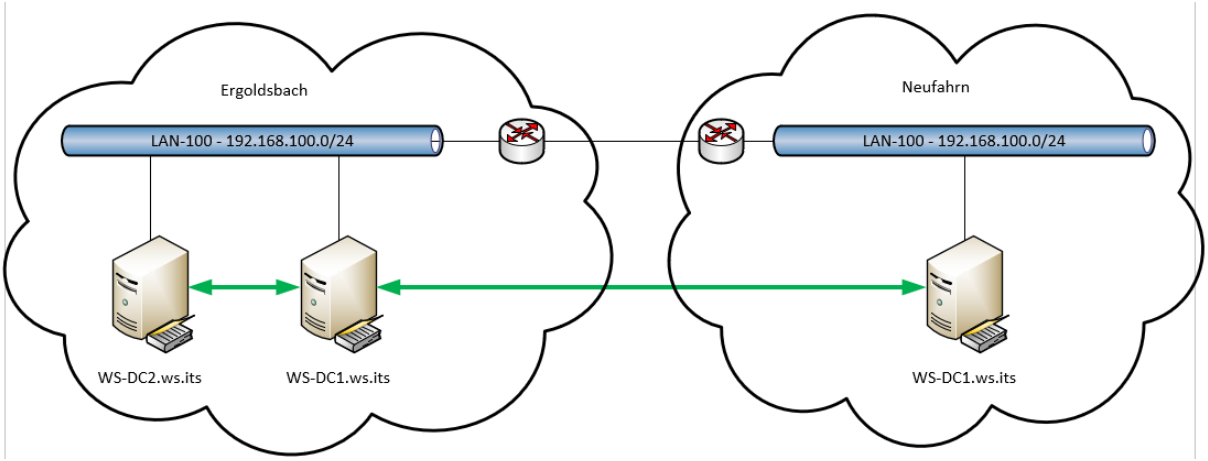
Der Server stellt einen zentralen Teil meiner Infrastruktur dar. Damit ich bei der Migration ohne Probleme durchkomme, analysiere ich nun mein Active Directory. Der Server WS-DC1 ist aktuell mein Flexible Single Master Operator für alle 5 Rollen:

```
PS C:\> netdom /query fsmo
Schemamaster               WS-DC1.ws.its
Dom„nennamen-Master        WS-DC1.ws.its
PDC                        WS-DC1.ws.its
RID-Pool-Manager           WS-DC1.ws.its
Infrastrukturmaster        WS-DC1.ws.its
Der Befehl wurde ausgefhrt.
```

Dazu agiert er als IP-Bridgehead-Server. Mit dieser Funktion ist er der primäre Replikationspartner für standortübergreifende Replikationen:



Das bedeutet aber auch, dass dieser Server die Relikation aller Domain Controller ausführt. Wenn er fehlt, dann sind die beiden anderen DCs zumindest für eine Zeit isoliert:



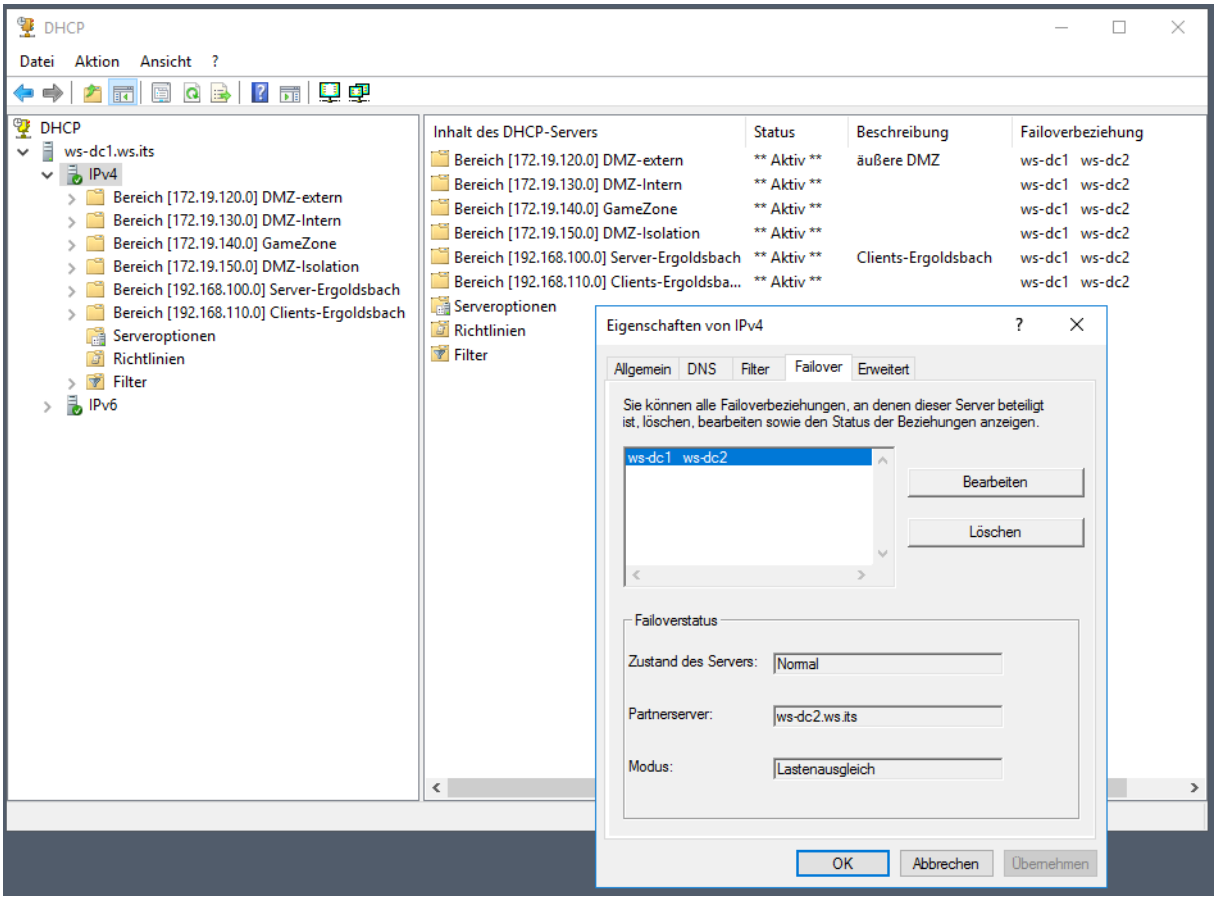
Der Server WS-DC1 ist als PDC-Emulator auch der primäre Zeitserver meiner Infrastruktur. Er selber kann sich seine Zeit von einem öffentlichen NTP-Server holen:

```
PS C:\> w32tm /query /configuration
[Konfiguration]

EventLogFlags: 2 (Lokal)
AnnounceFlags: 10 (Lokal)
TimeJumpAuditOffset: 28800 (Lokal)
MinPollInterval: 6 (Lokal)
MaxPollInterval: 10 (Lokal)
MaxNegPhaseCorrection: 172800 (Lokal)
MaxPosPhaseCorrection: 172800 (Lokal)
MaxAllowedPhaseOffset: 300 (Lokal)

FrequencyCorrectRate: 4 (Lokal)
PollAdjustFactor: 5 (Lokal)
LargePhaseOffset: 50000000 (Lokal)
SpikeWatchPeriod: 900 (Lokal)
LocalClockDispersion: 10 (Lokal)
HoldPeriod: 5 (Lokal)
PhaseCorrectRate: 7 (Lokal)
UpdateInterval: 100 (Lokal)


[Zeitanbieter]

NtpClient (Lokal)
DllName: C:\Windows\system32\w32time.dll (Lokal)
Enabled: 1 (Lokal)
InputProvider: 1 (Lokal)
AllowNonstandardModeCombinations: 1 (Lokal)
ResolvePeerBackoffMinutes: 15 (Lokal)
ResolvePeerBackoffMaxTimes: 7 (Lokal)
CompatibilityFlags: 2147483648 (Lokal)
EventLogFlags: 1 (Lokal)
LargeSampleSkew: 3 (Lokal)
SpecialPollInterval: 3600 (Lokal)
Type: NTP (Lokal)
NtpServer: de.pool.ntp.org (Lokal)

NtpServer (Lokal)
DllName: C:\Windows\system32\w32time.dll (Lokal)
Enabled: 1 (Lokal)
InputProvider: 0 (Lokal)
AllowNonstandardModeCombinations: 1 (Lokal)

VMICTimeProvider (Lokal)
DllName: C:\Windows\System32\vmictimeprovider.dll (Lokal)
Enabled: 1 (Lokal)
InputProvider: 1 (Lokal)
```

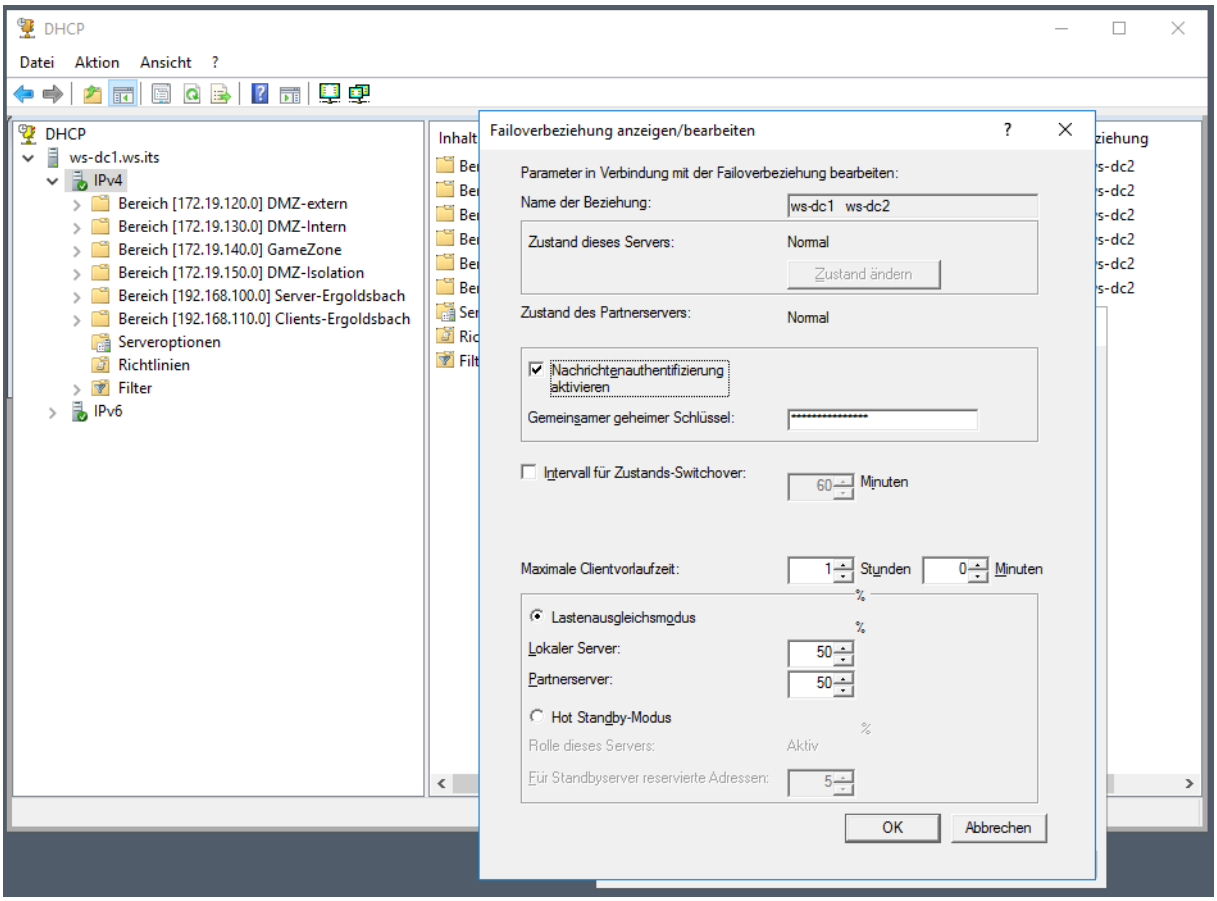
Die Domänenfunktionsebene und die Gesamtstrukturfunktionsebene habe ich bereits auf Windows Server 2016 angehoben. Das Active Directory Schema läuft mit der Version Windows Server 2016. Das sollte für eine Koexistenz mit Windows Server 2019 genügen.

aktuelle Konfiguration des DHCP

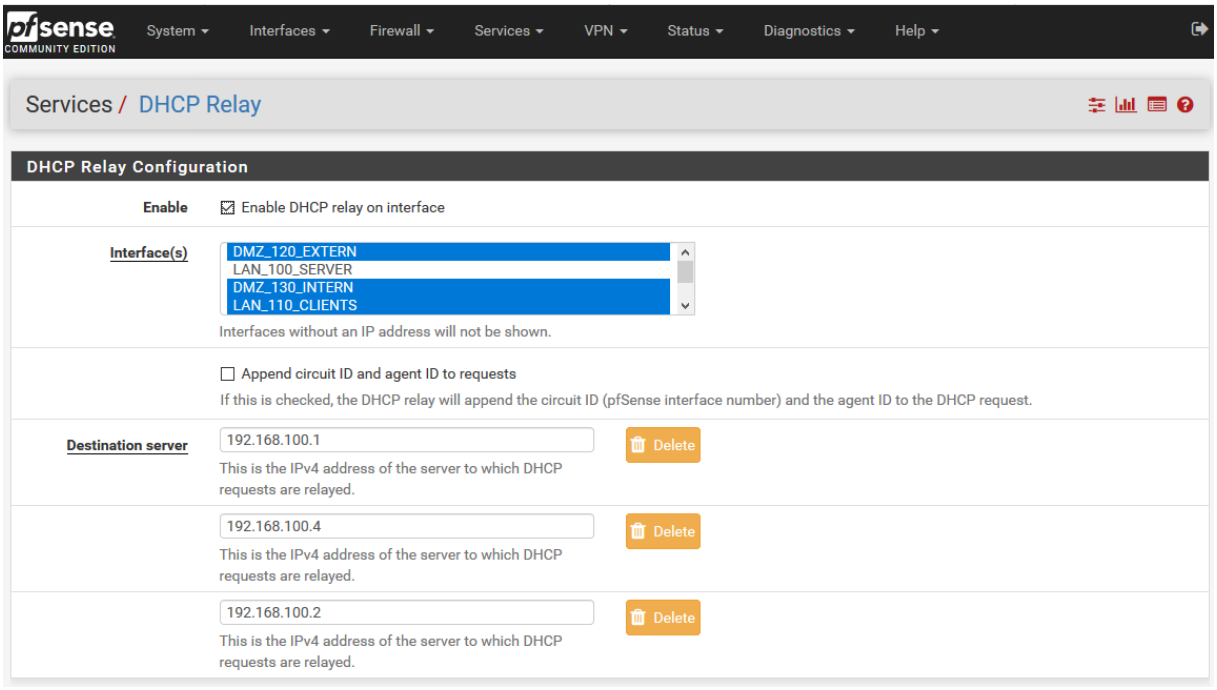
Der Server WS-DC1 ist ebenfalls als DHCP-Server aktiv. Seine Scopes synchronisiert er mit dem WS-DC2 über ein DHCP-Failover:



Die Konfiguration ist Standard:



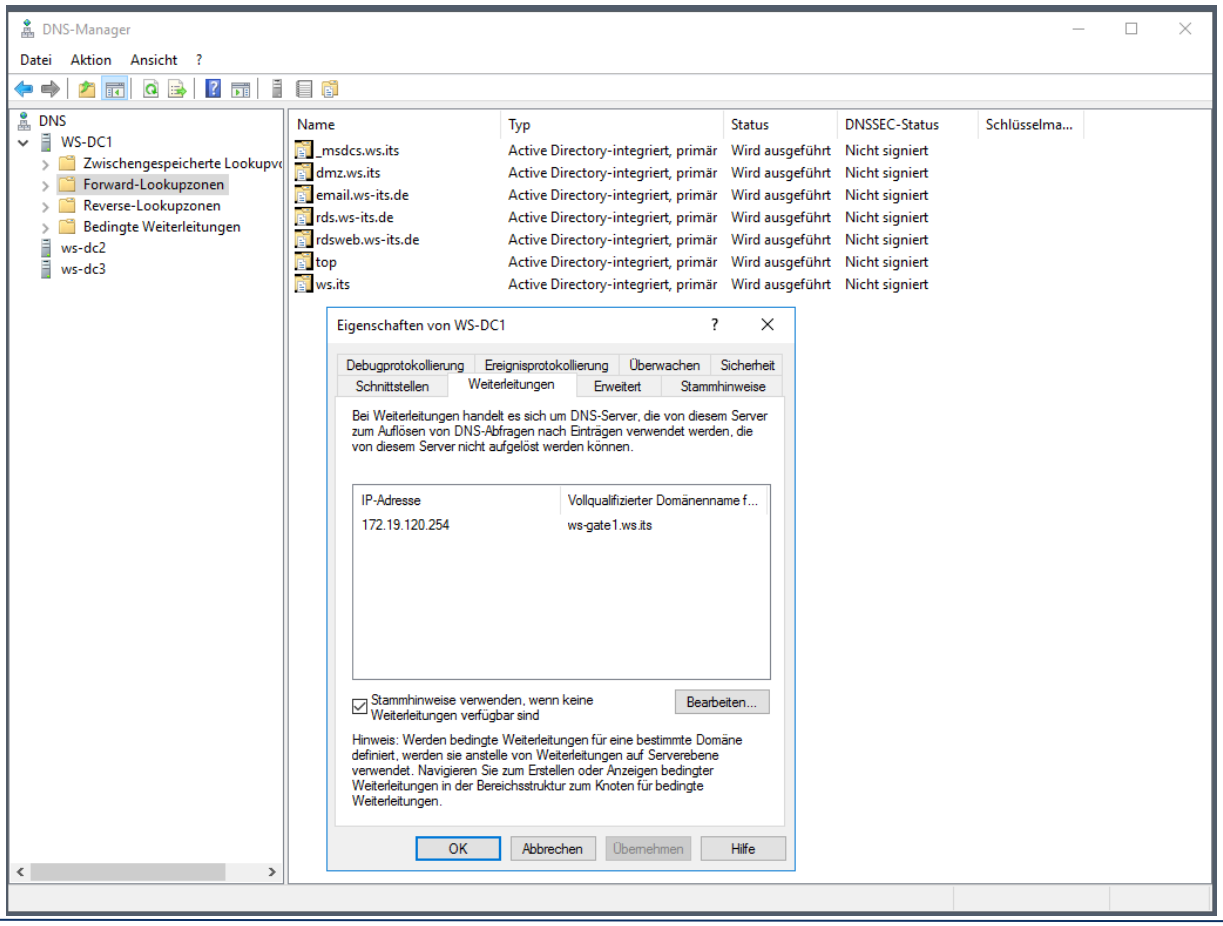
Meine PFSense verbindet alle internen Netzwerke miteinander. Sie fungiert daher als DHCP-Relay (IPHelper) und leitet Anfragen an beide DHCP-Server weiter:



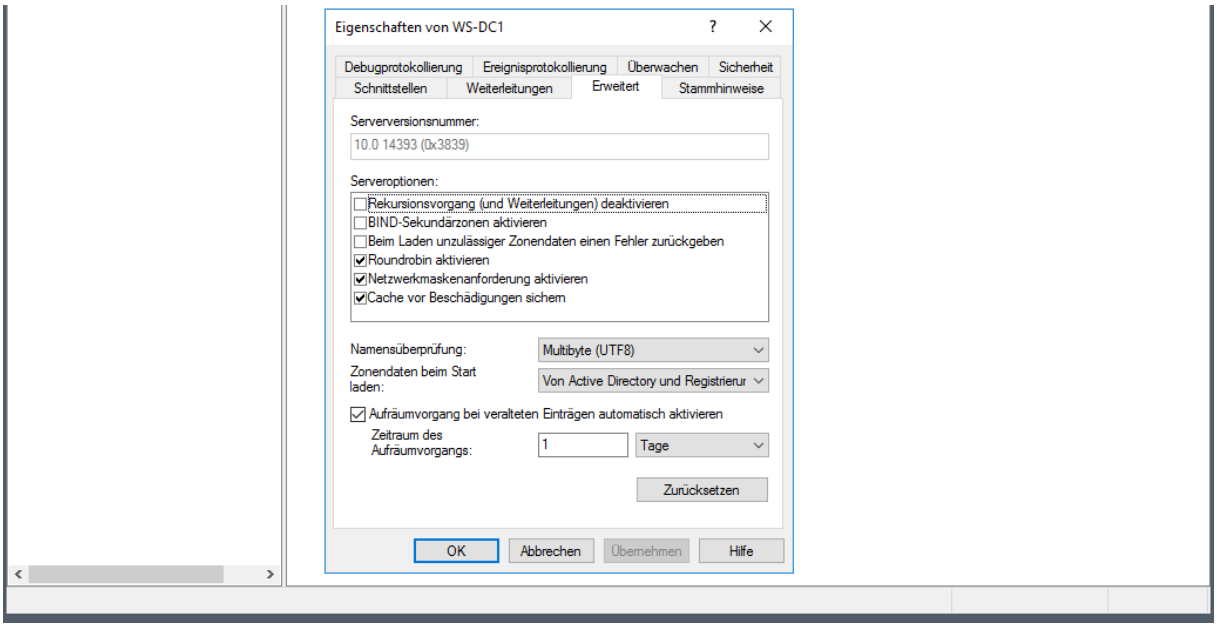
Damit sollte ich die DHCP-Funktion vom alten WS-DC1 herunternehmen können, ohne dass es zum Service-Ausfall kommt.

aktuelle Konfiguration des DNS

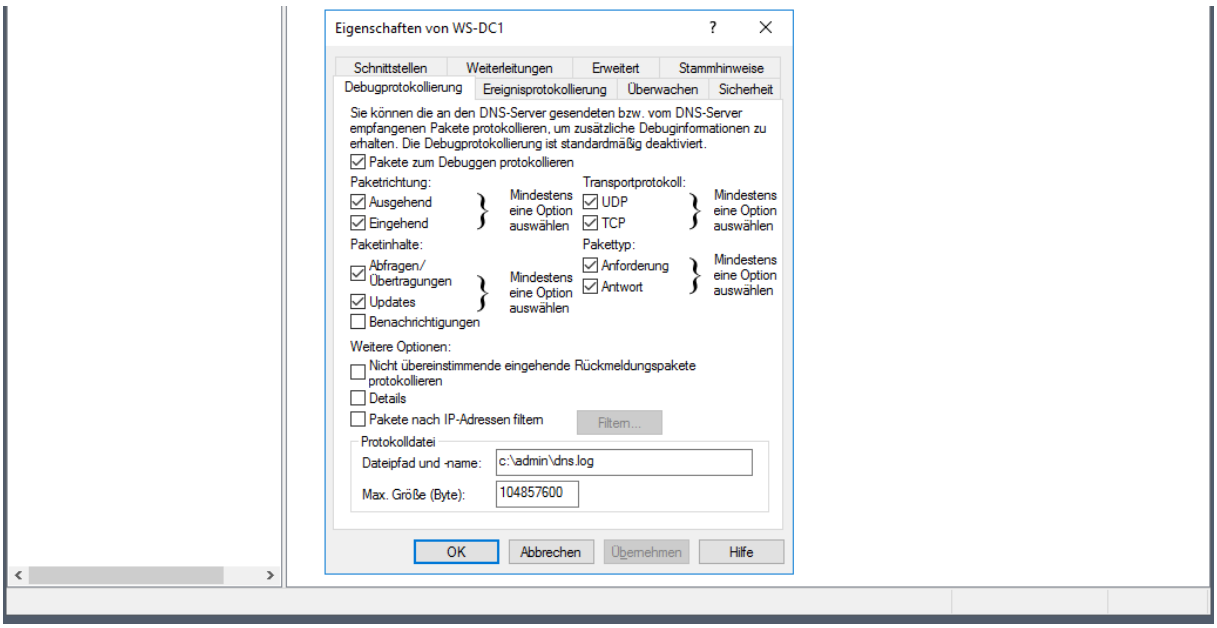
Jeder Domain Controller ist auch als schreibbarer Namensserver unterwegs. Ich sichte nun die lokale Konfiguration des DNS-Services. Der Forwarder ist meine Fritzbox in der äußeren DMZ:



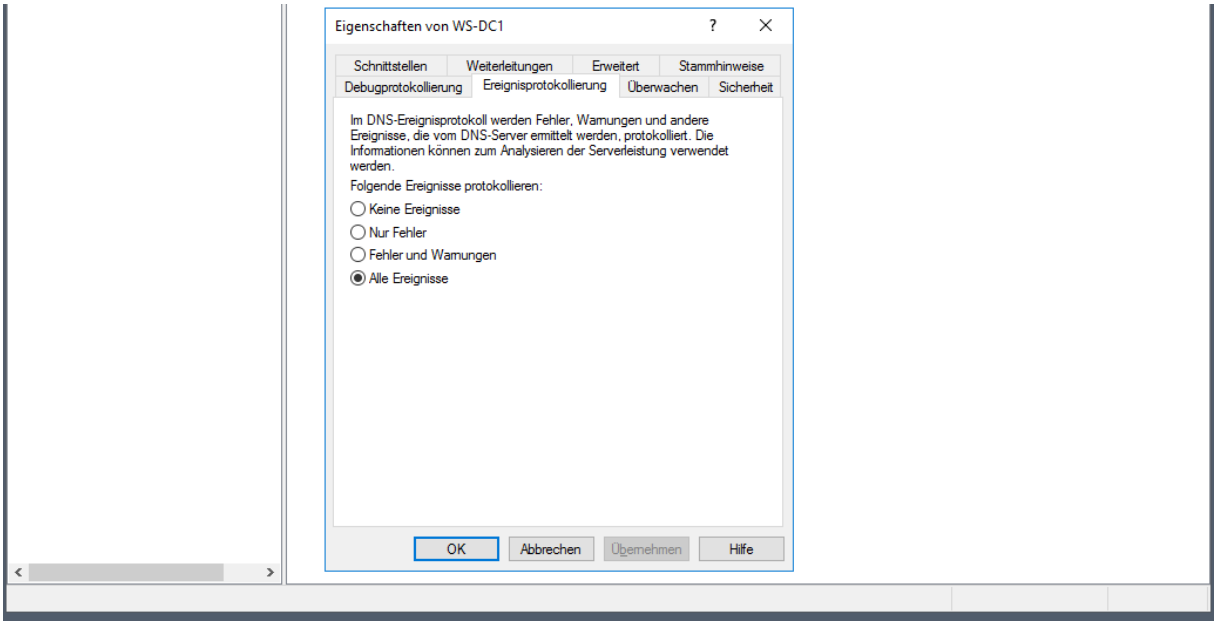
Der Server darf aufräumen:



Ebenso habe ich das Logging aktiv. So kann ich im Nachgang verdächtige Namensanfragen an die Clients zuweisen:



Das Eventlogging ist default:

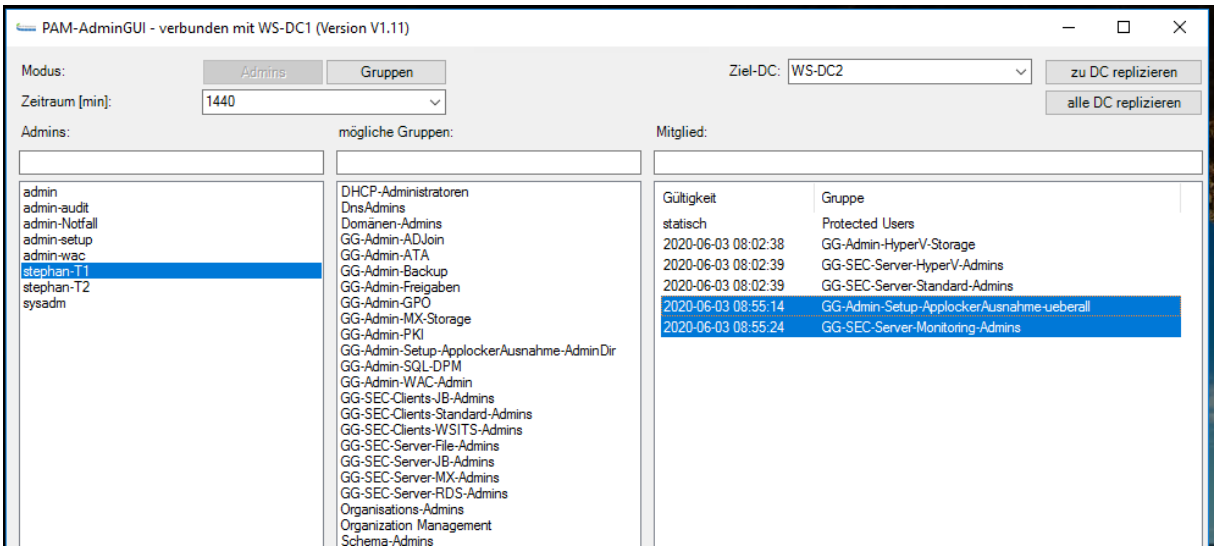


Diese Rolle sollte nicht schwer zu migrieren sein.

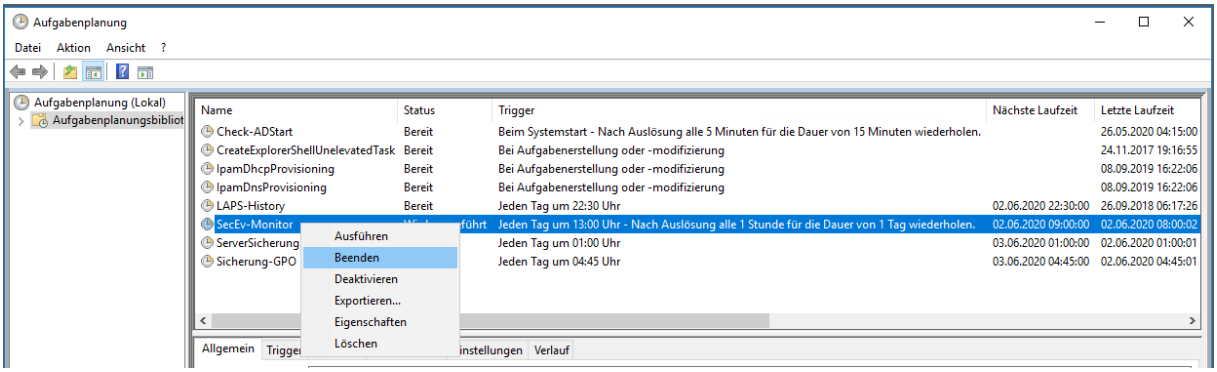
Verschiebung des SecEv-Monitors auf WS-MON

Dieses Script kann auf meinen Domain Controllern nach fehlerhaften Anmeldungen suchen und mit per Mail darüber berichten. Das Script läuft 24/7 auf meinem WS-DC1. Es würde aber besser auf meinem Monitoring-Server WS-MON platziert sein. Daher verschiebe ich diese Funktion.

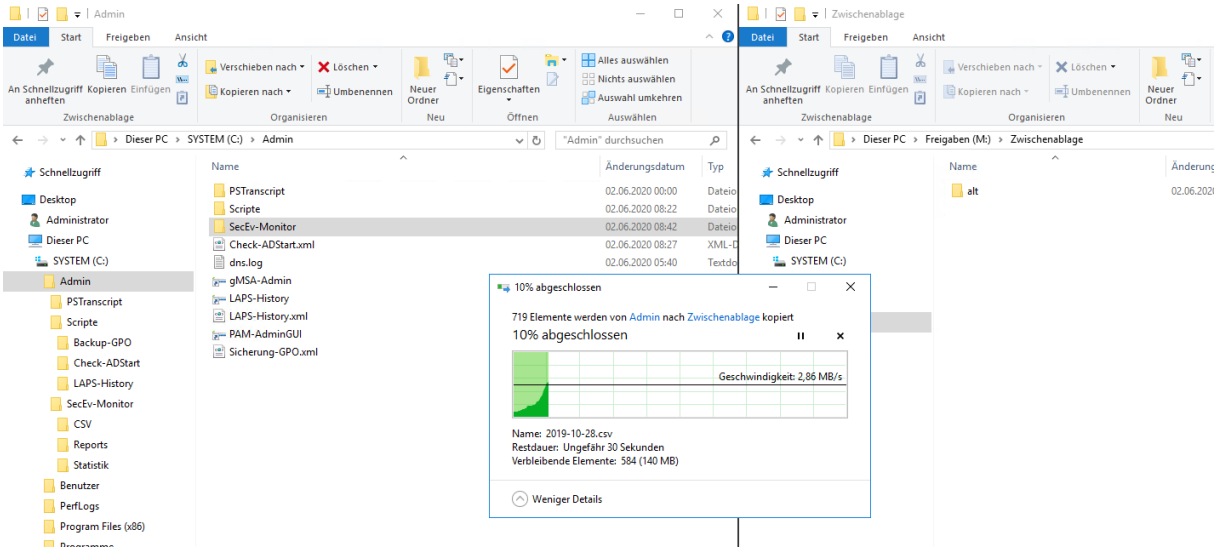
Zuerst bereite ich meinen AdminAccount mit meinem PAM-Tool vor, damit ich mich mit dem WS-MON verbinden kann. Zusätzlich muss ich dort noch eine Anwendung installieren. Das würde mein Applocker auch für den Admin verhindern – wenn er nicht in der Whitelist-Gruppe Mitglied ist:



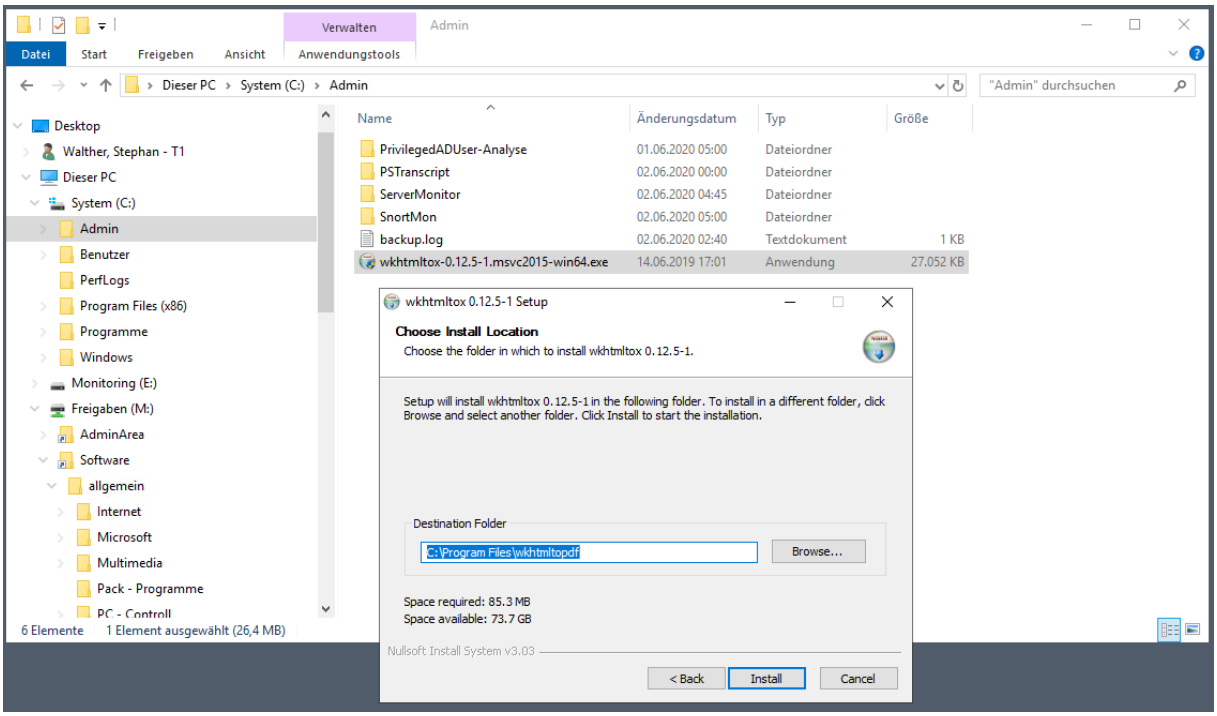
Ich beende den Task auf dem Server. Das Script ist beim Wiederanlauf so clever und holt auf den Domain Controllern die verpassten Events einfach nach, solange diese nicht durch die Umlaufprotokollierung der Eventlogs gelöscht wurden:



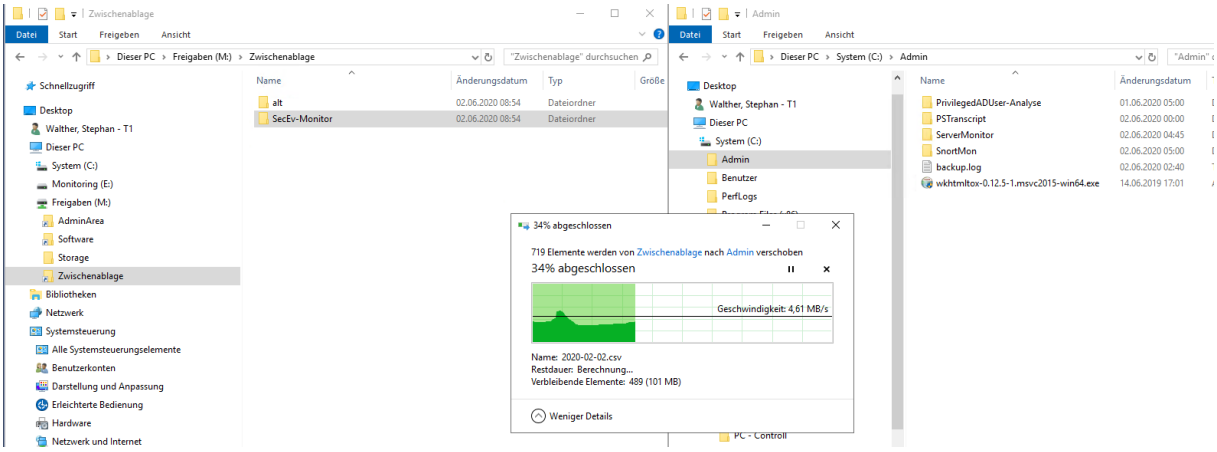
Jetzt verschiebe ich das Scriptverzeichnis in meine Zwischenablage. Mein Domain Controller kann den Server WS-MON nicht direkt erreichen:



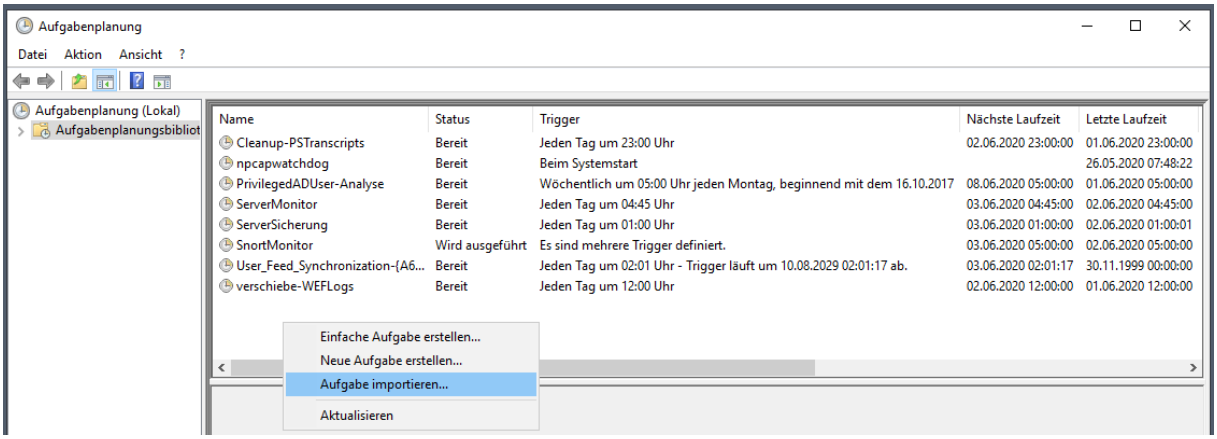
Da hat sich schon einiges an Dateien angesammelt. Während der Verschiebung installiere ich auf dem Server WS-MON einen HTML-to-PDF-Converter:

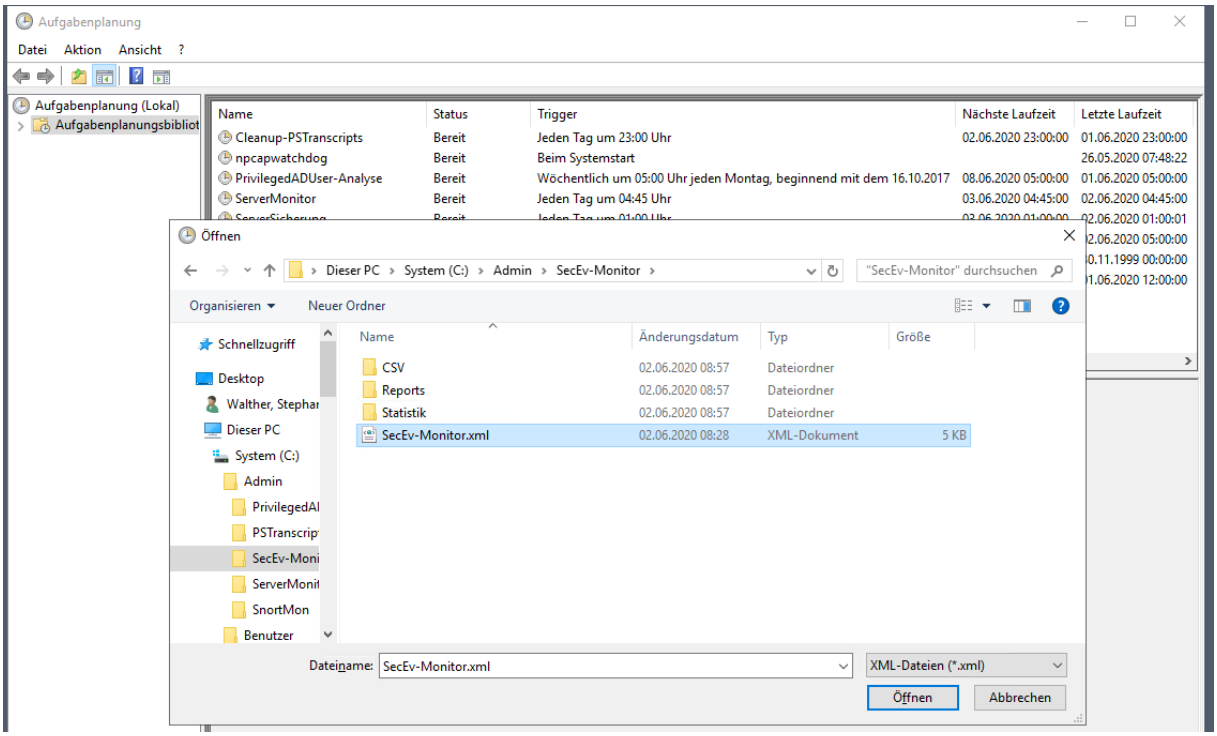


Dann verschiebe ich die Scriptdateien und die alten Logfiles von der Zwischenablage auf den Monitor-Server:

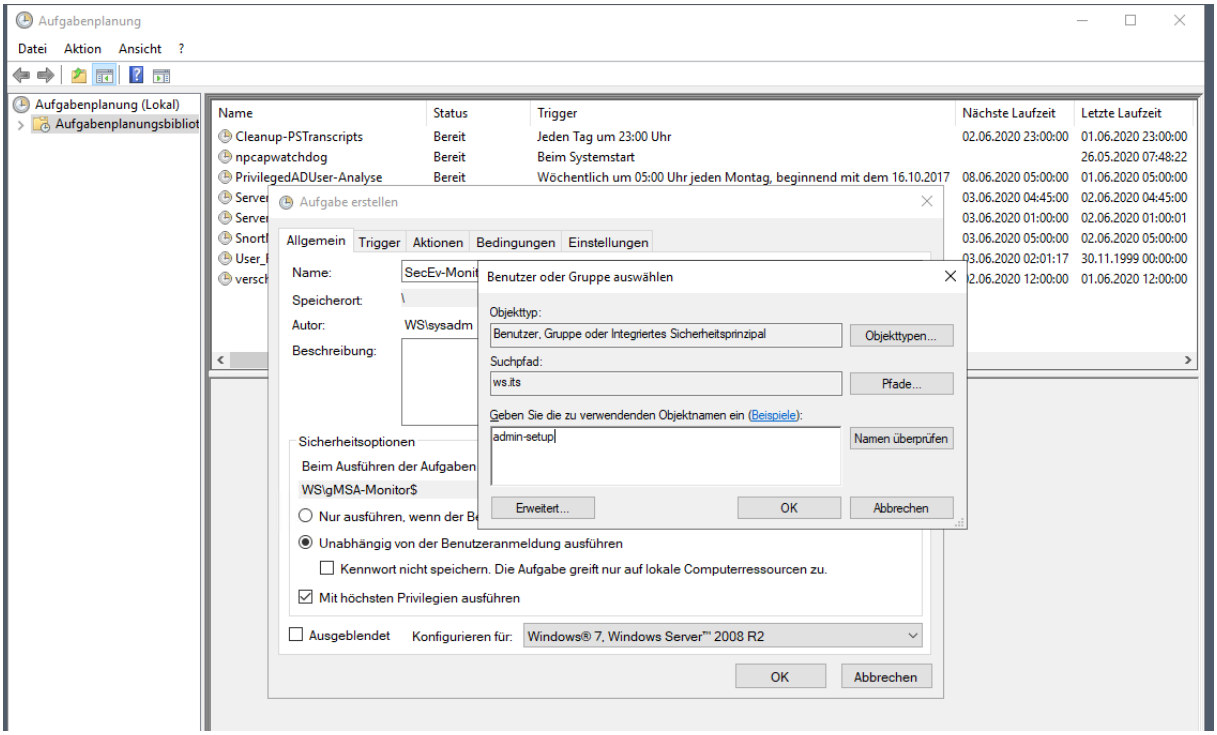


Die exportiere Aufgabe kann ich im Server WS-MON einfach importieren:

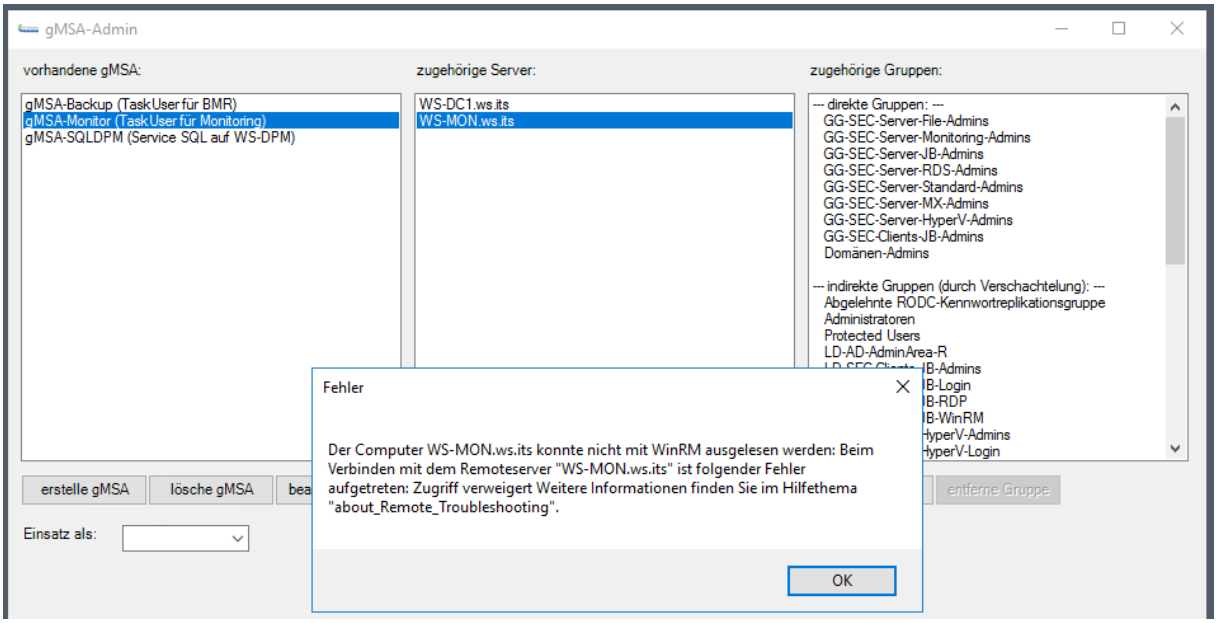




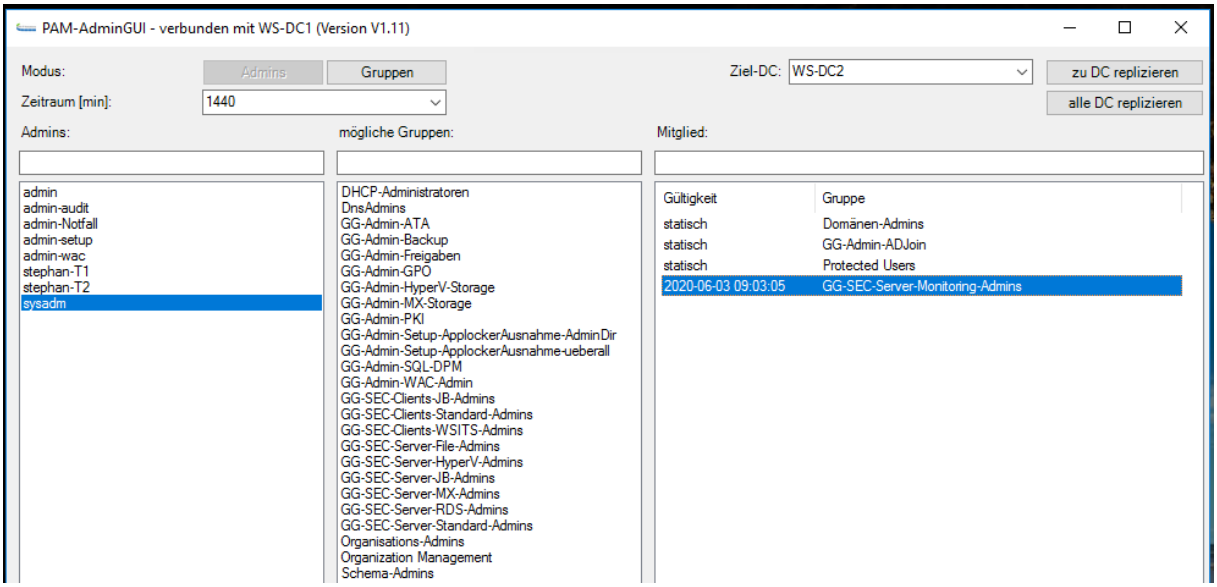
Der ausführende Account ist ein Group-Managed-Service-Account. Den kann ich nicht direkt angeben, da ich sein Passwort nicht kenne. Ich verwende stattdessen einen Dummy-Account:



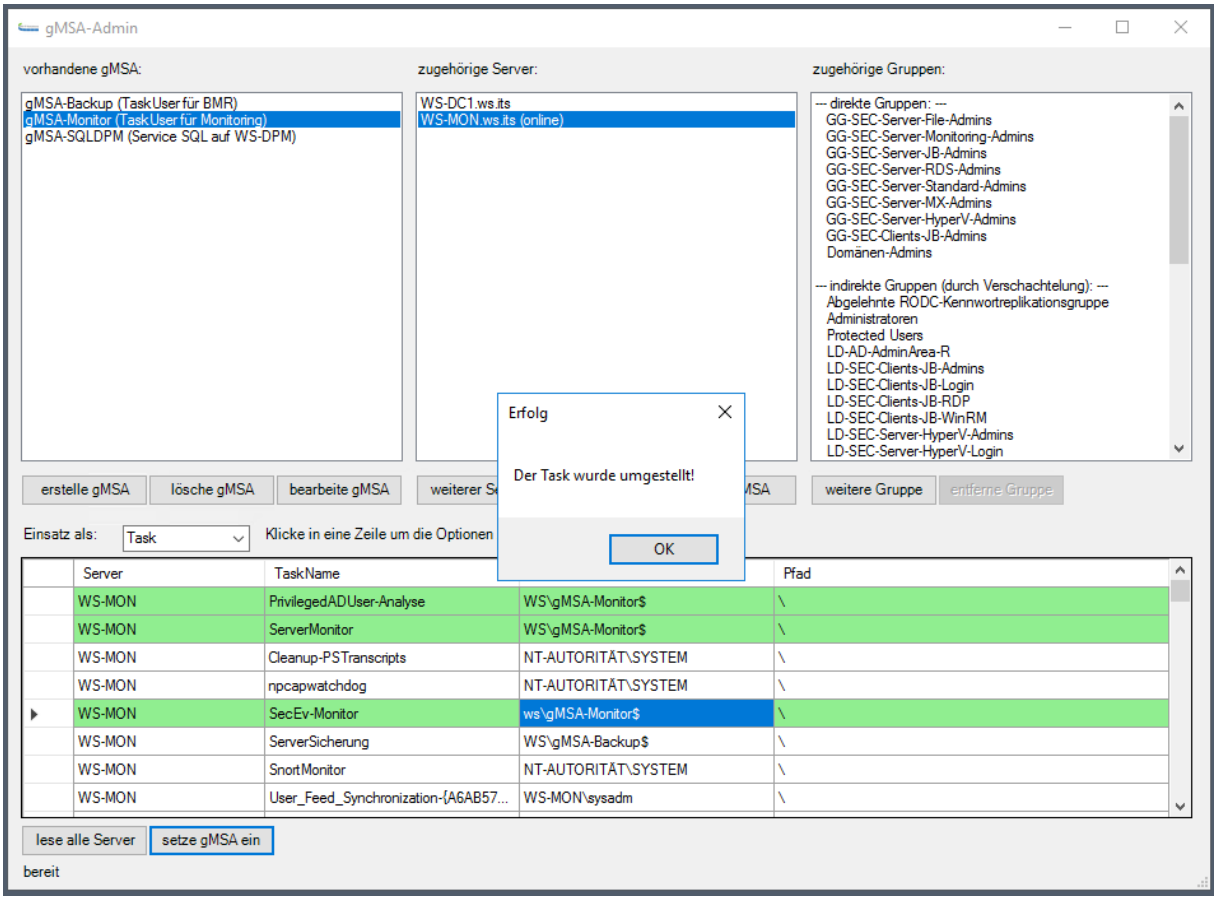
Vom alten DC aus möchte ich nun mit meinem PowerShell-Script gGMSA-Admin den Task auf den gMSA-Account umstellen. Aber meine Rechte reichen nicht aus:



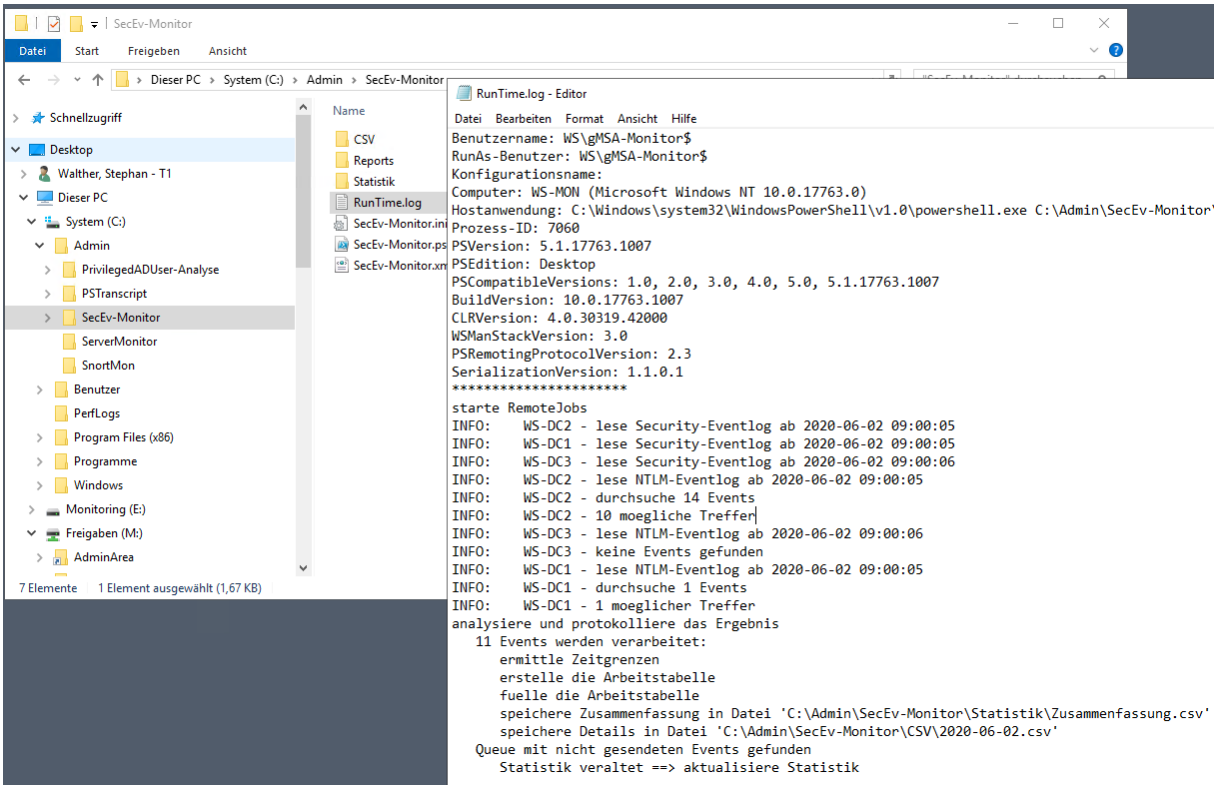
Die Ursache ist einfach – wenn auch leider noch sehr unüblich in den meisten Infrastrukturen: meine Domain Admins haben auf den Memberservern und auf den Clients keine (!) Rechte mehr. Die meiste Zeit sind diese auch nicht erforderlich. Aber jetzt benötige ich eine kurze Brücke. Da hilft mir wieder mein Script PAM-AdminGUI. Ich gebe meinem Domain Admin die Adminrechte auf dem Monitoring-Server durch eine zeitlich begrenzte Gruppenmitgliedschaft:



Nach einer Neuanmeldung am Domain Controller ist die Gruppenmitgliedschaft aktiv. Jetzt kann ich den gMSA in den Script-Task eintragen:



Das Script erstellt ein Runtime-Logfile. So kann ich den Laufzeitstatus überprüfen. Ich starte den Script-Task. Der gMSA wird korrekt verwendet und die Events werden analysiert:

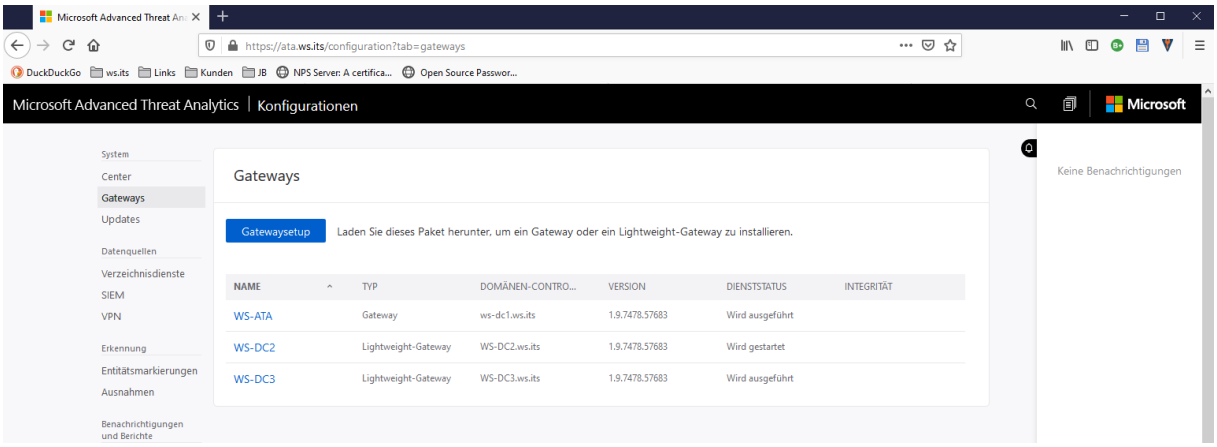


Damit ist diese Funktion erfolgreich verschoben.

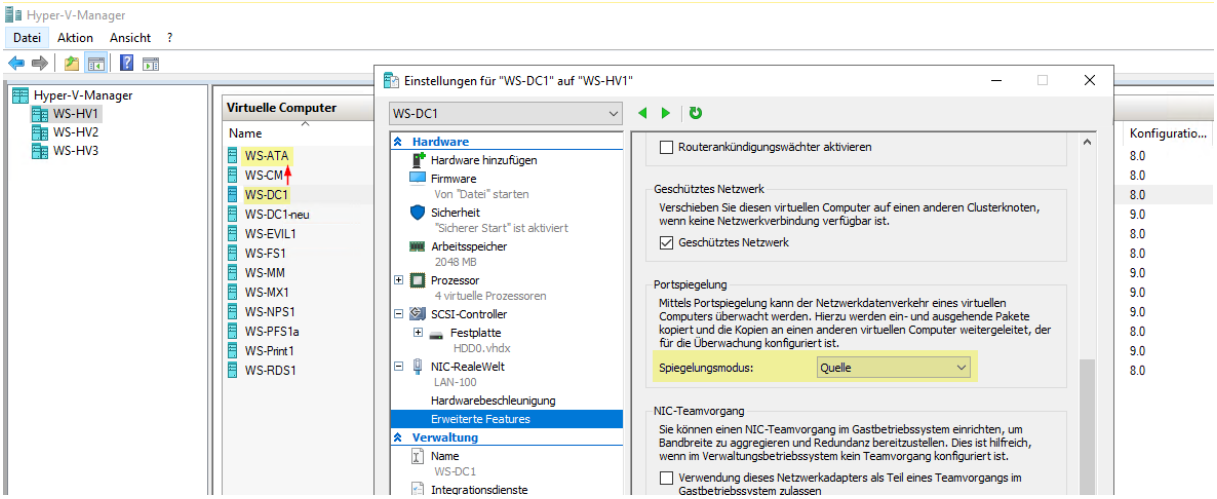
aktuelle ATA-Konfiguration

Meine Sicherheitsanalyse stützt sich auch auf einen Microsoft ATA – ein Advanced Threat Analytics System, das Anomalien erkennt und danach alarmieren kann. Dieser Service klinkt sich in alle Domain Controller ein. Dabei kann entweder ein lokaler Agent installiert werden oder man arbeitet mit einer Netzwerk-Traffic-Weiterleitung.

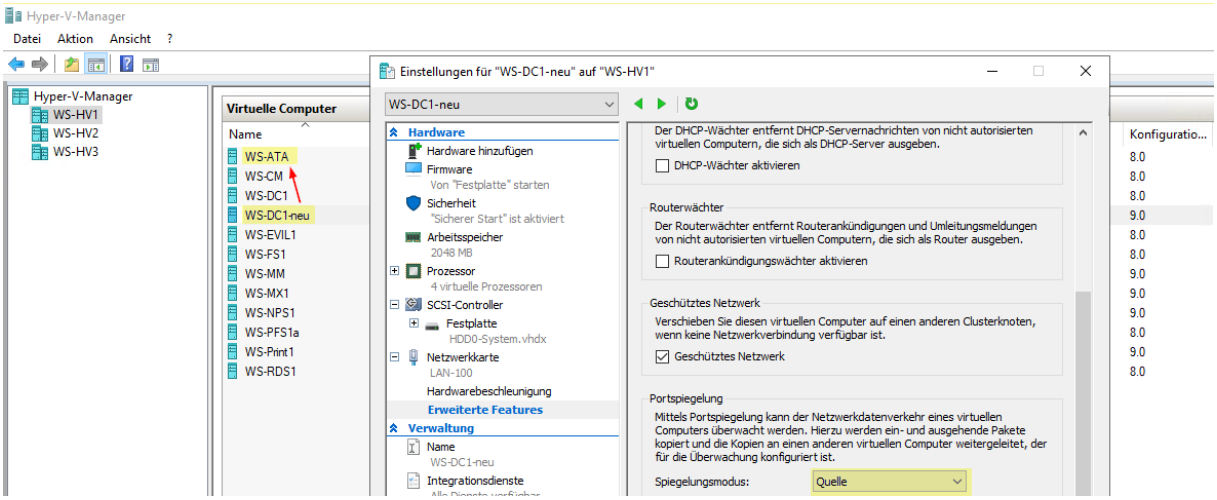
Der Domain Controller WS-DC1 kommt ohne einen Agent aus:



Das funktioniert, weil mein Server WS-ATA als virtuelle Maschine auf dem gleichen Hyper-V-Host läuft, wie der virtuelle Server WS-DC1. Dessen Netzwerk-Traffic lasse ich mit Hyper-V-Mitten spiegeln. Das Interface wird als Quelle angezapft:



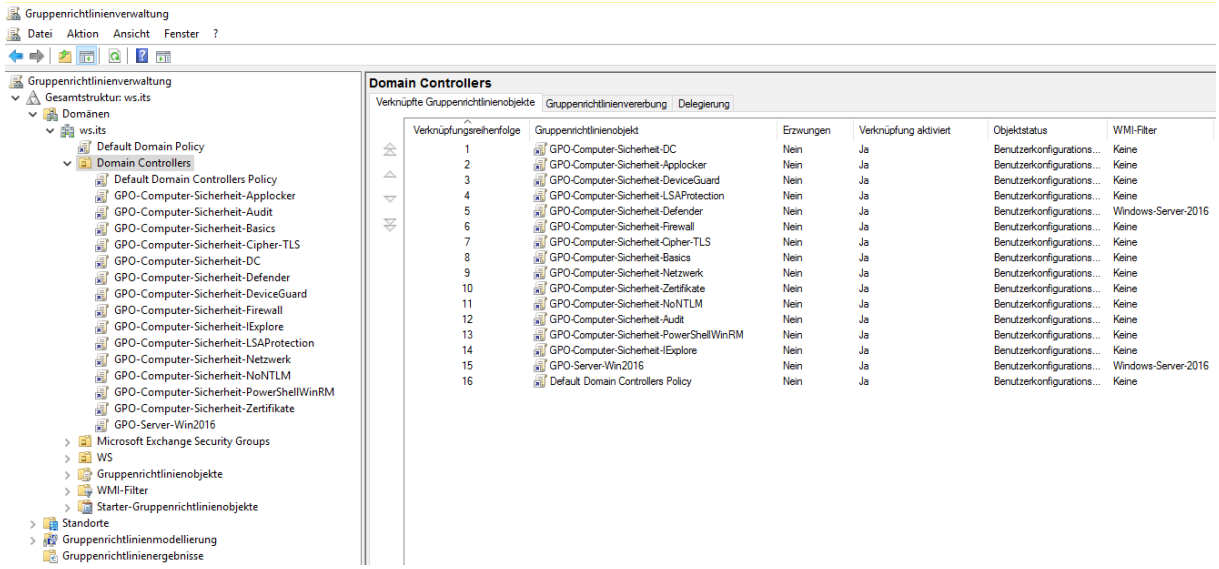
Das Ziel ist die Netzwerkkarte des Servers WS-ATA. Diese Funktion darf beim neuen DC nicht fehlen. Daher stelle ich sie in seiner Netzwerkkarte mit ein:



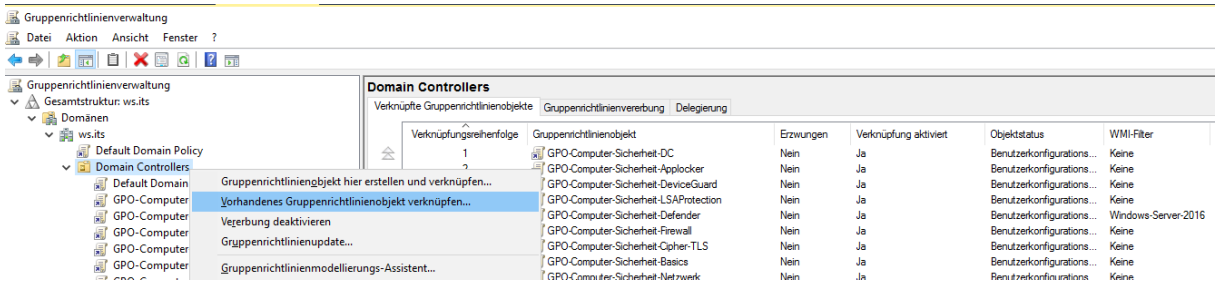
Damit ist im ATA keine weitere Aktion erforderlich.

Prüfung der Gruppenrichtlinien

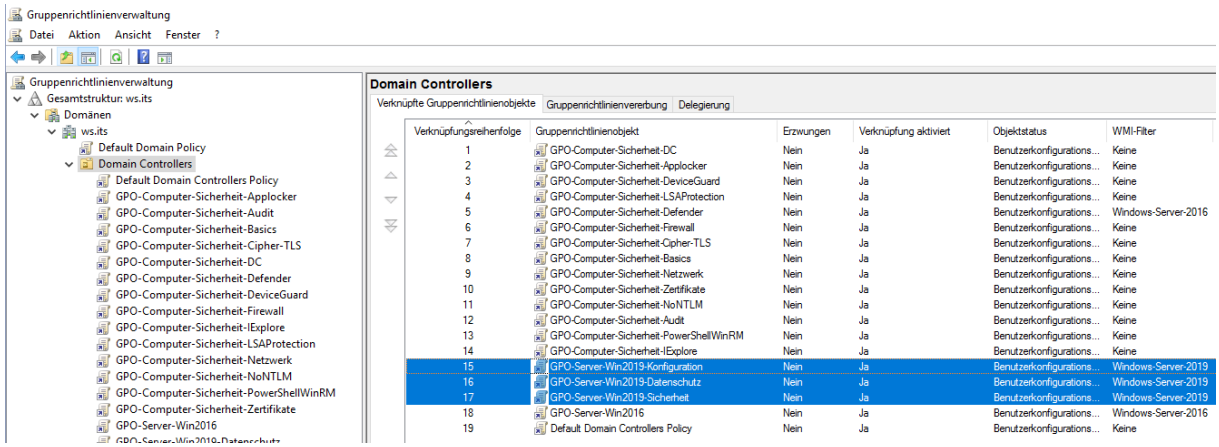
Das wird gerne vergessen: Auch die Domain Controller verarbeiten Gruppenrichtlinien. Und diese müssen auf deren Betriebssysteme abgestimmt sein. Ich habe natürlich schon fertige GPO für Windows Server 2019 im Einsatz – aber diese sind noch nicht auf der OU „Domain Controllers“ verbunden!



Das habe ich bisher nicht benötigt. Aber der neue Server soll vom ersten Moment an gut abgesichert sein. Daher verbinde ich die drei Gruppenrichtlinien:

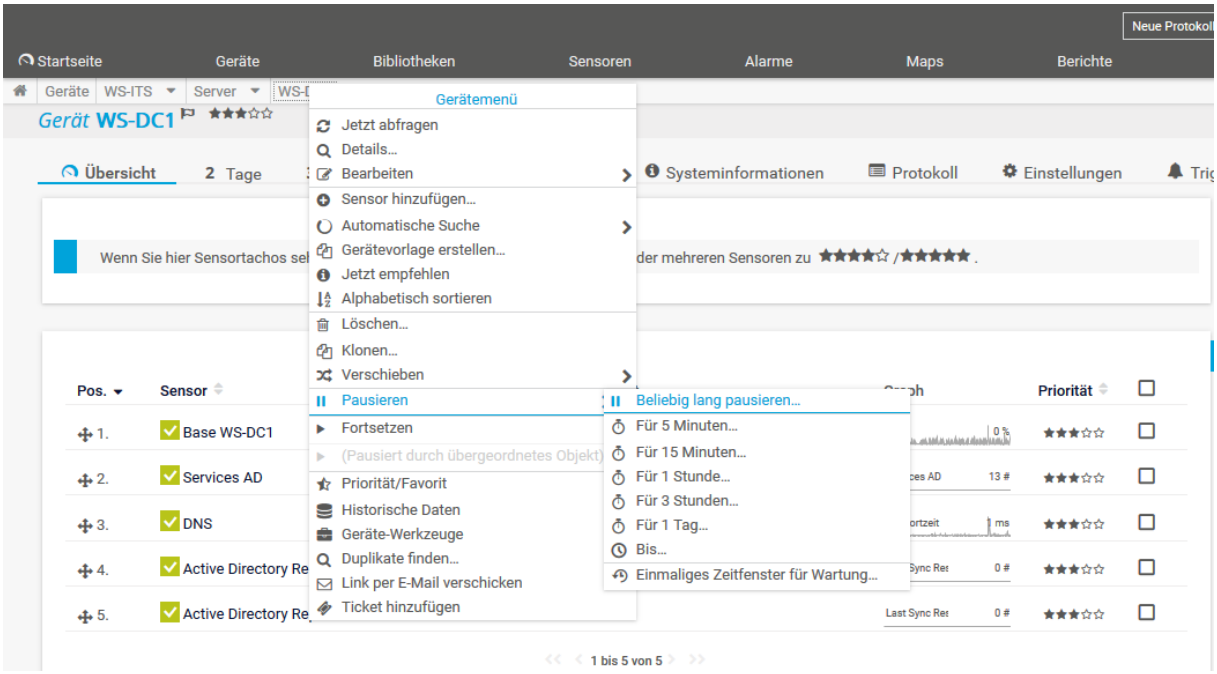


Wichtig ist die Verarbeitungsreihenfolge. Der WMI-Filter sorgt dafür, dass sich die Richtlinien der unterschiedlichen Betriebssysteme nicht in die Quere kommen:



Maintenance

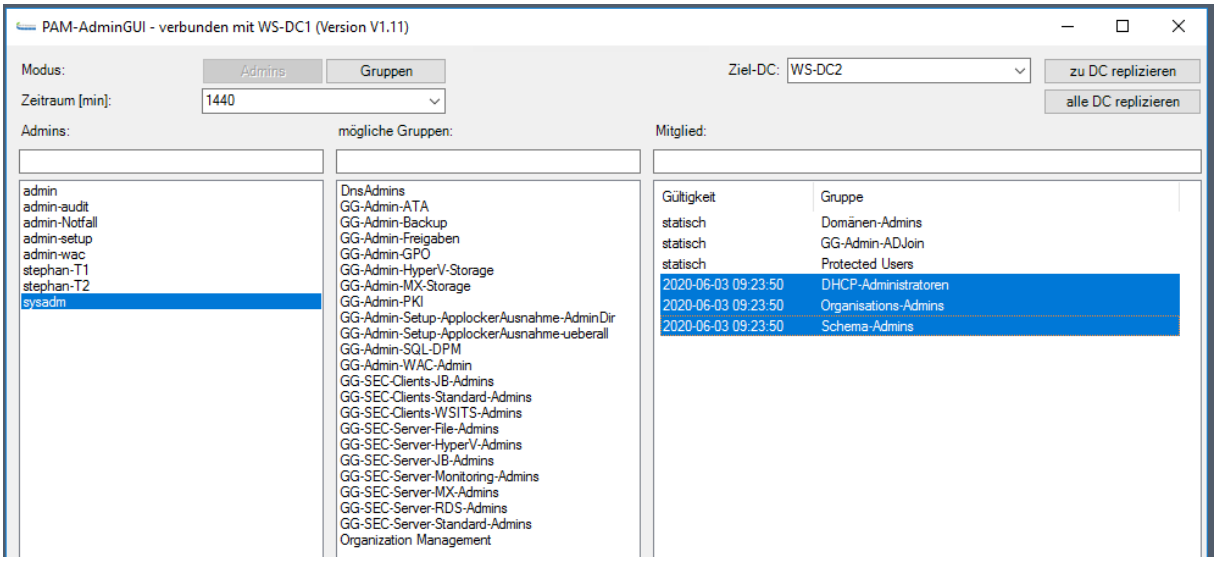
Mein Monitoring hat einige Sensoren, die mein Active Directory überwachen. Mit der Deinstallation würde es hier einige Alarme geben. Daher pausiere ich die entsprechenden Sensoren:



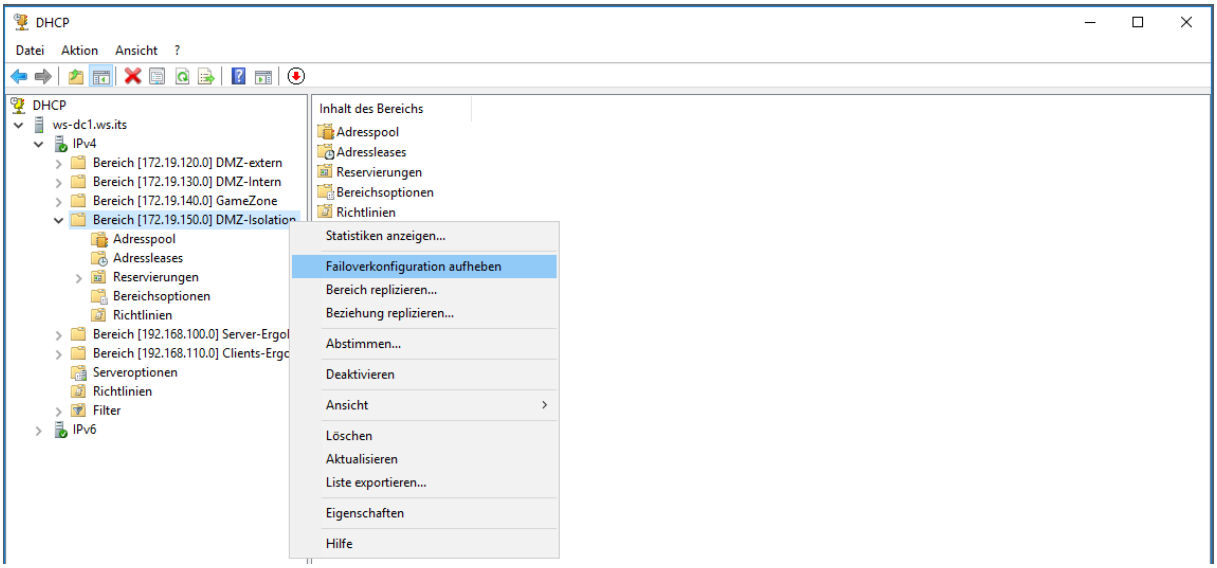
Deinstallation

Entfernen der Rolle DHCP

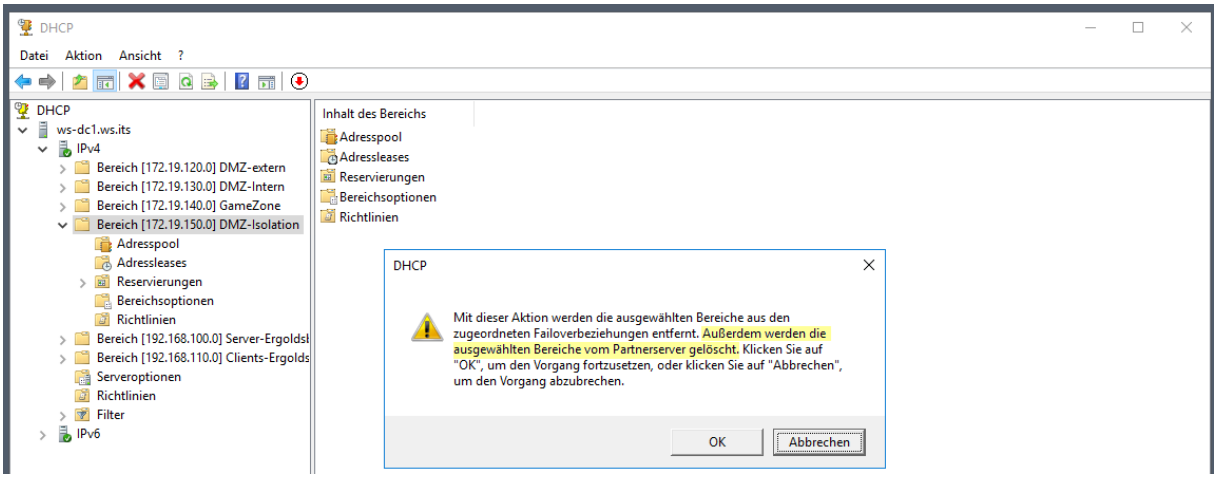
Jetzt kann der Rückbau des alten Servers starten. Nur so kann ich den Namen und die IP-Adresse freibekommen. Ich statte meinen AdminAccount mit weiteren Rechten aus. Die Mitgliedschaft als Enterprise-Admin und als Schema-Admin wird die Neuinstallation ermöglichen:



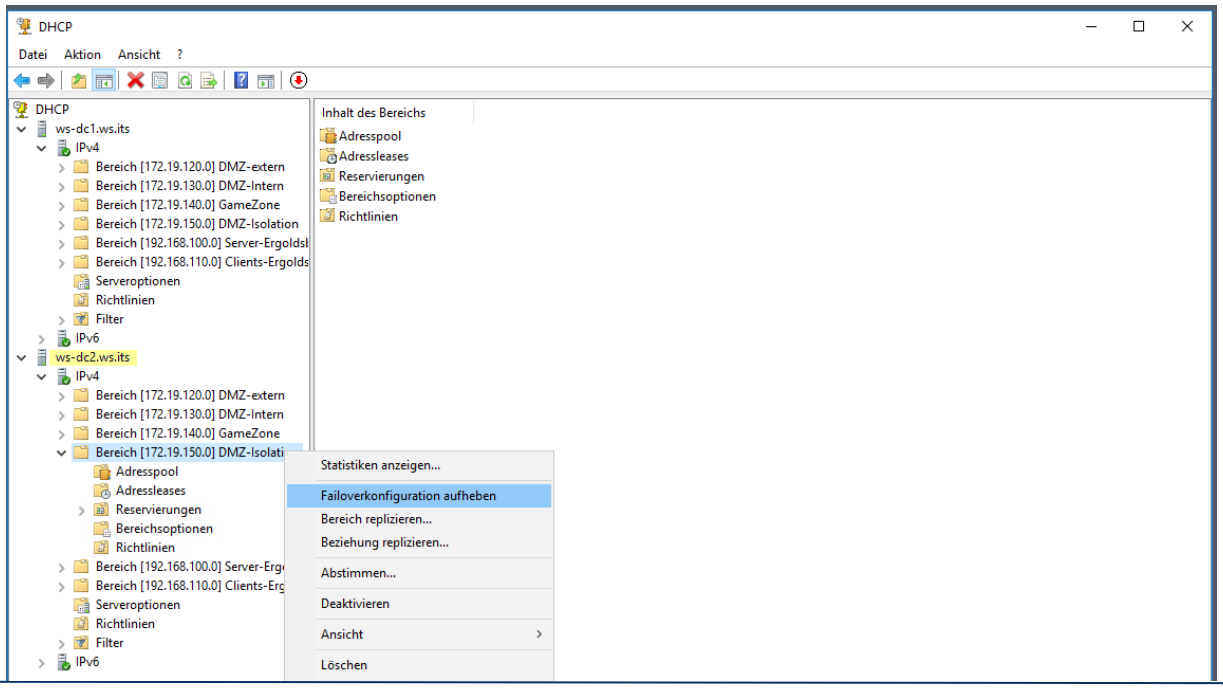
Nach der Neuanmeldung möchte ich den Server aus dem DHCP-Failover entfernen. Dann übernimmt diese Funktion mein Server WS-DC2:



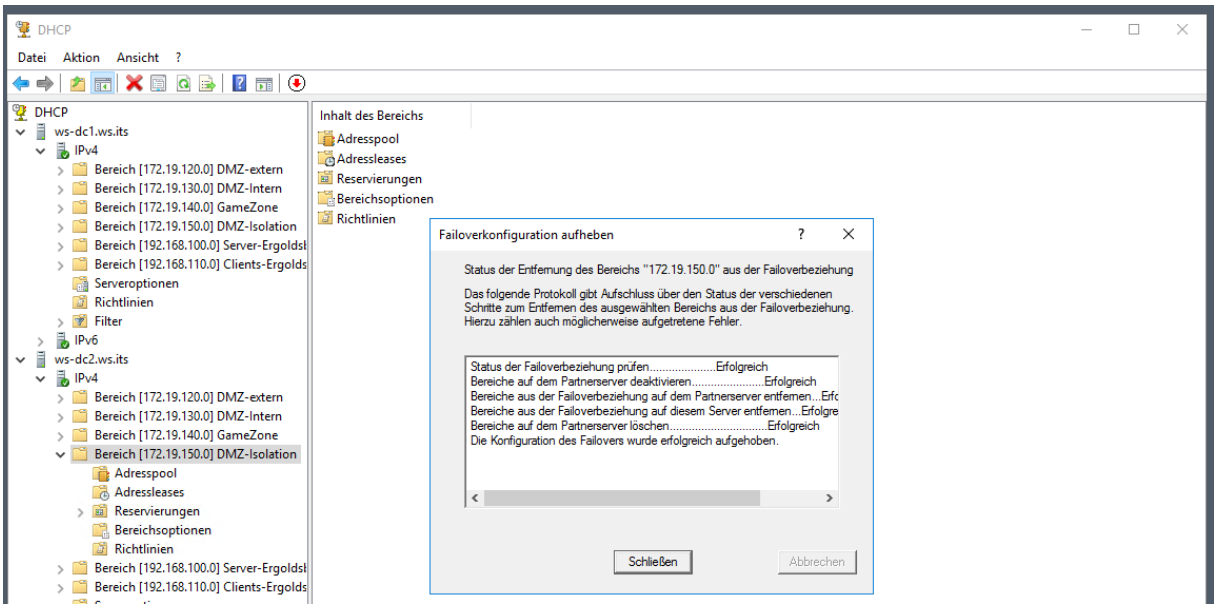
Aber Achtung: Die Meldung besagt, dass ich vom WS-DC1 aus die Entfernung des WS-DC2 einleite! Das ist genau das Gegenteil meines Plans:



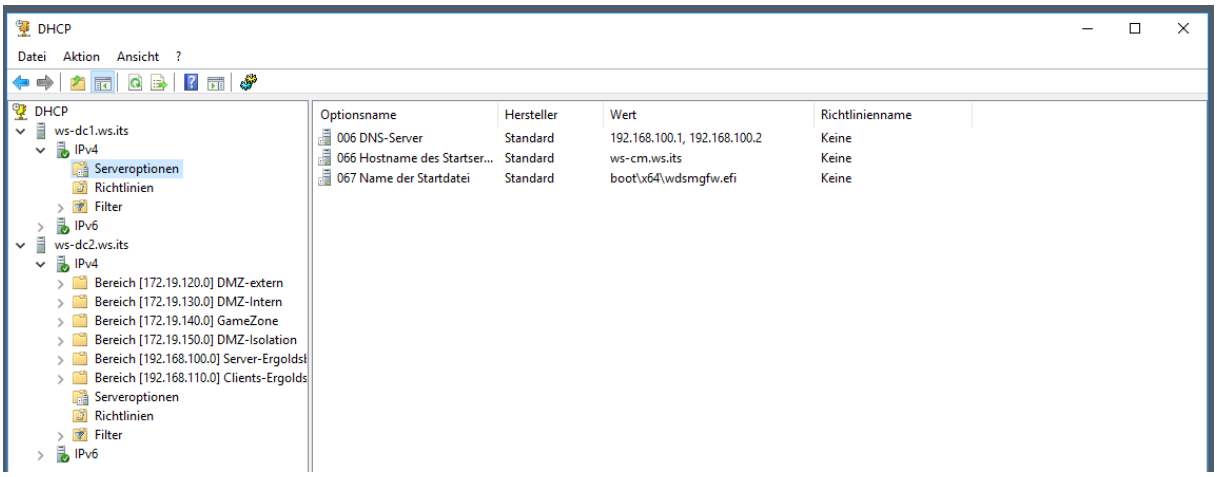
Also starte ich den Assistenten vom WS-DC2 aus:



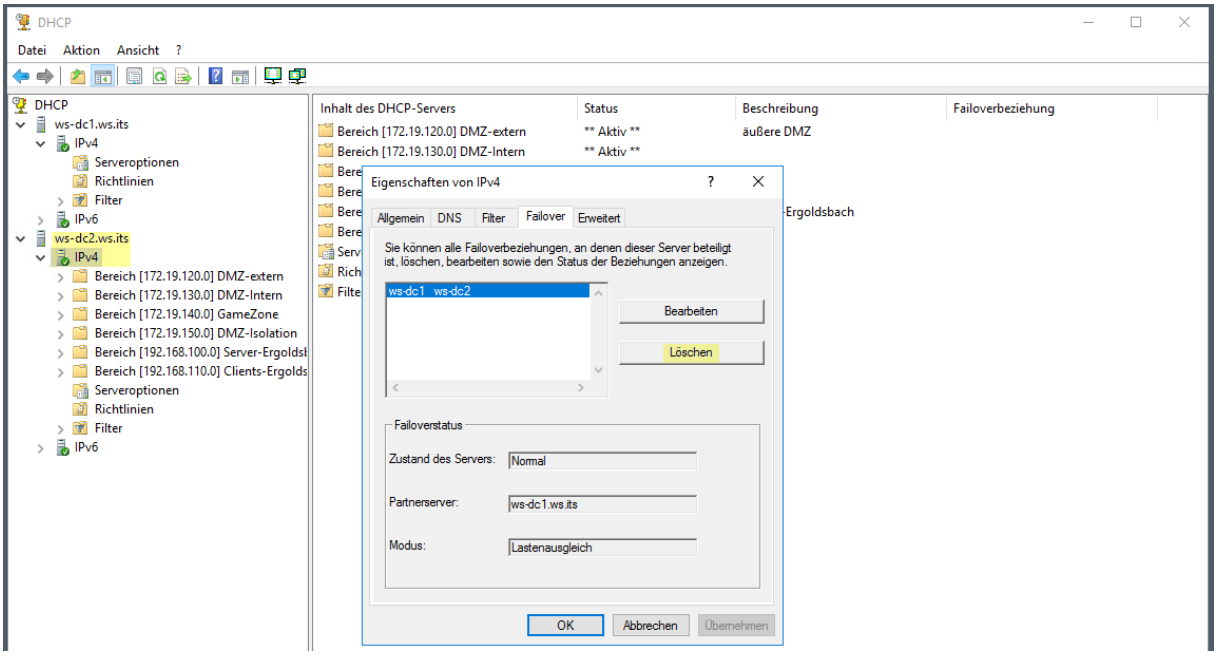
Jetzt ist es richtig. Ich wiederhole die Aktion für jeden einzelnen Scope:



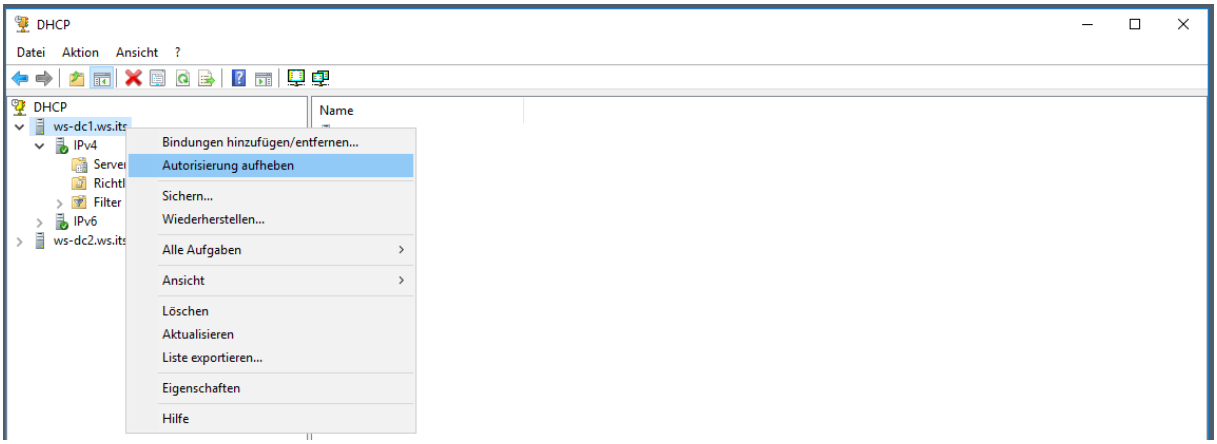
Nach einer Aktualisierung sind alle Scopes des alten Servers entfernt. Nur die Serveroptionen bleiben über:

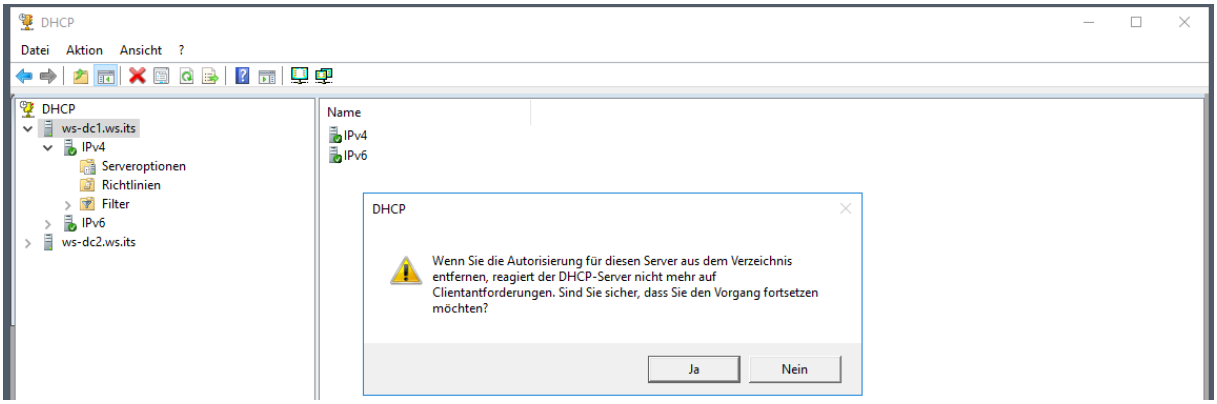


Jetzt entferne ich noch die Einstellung des DHCP-Failovers:



Der Server ist im Active Directoy autorisiert. Das hebe ich für die bevorstehende Deinstallation auf:

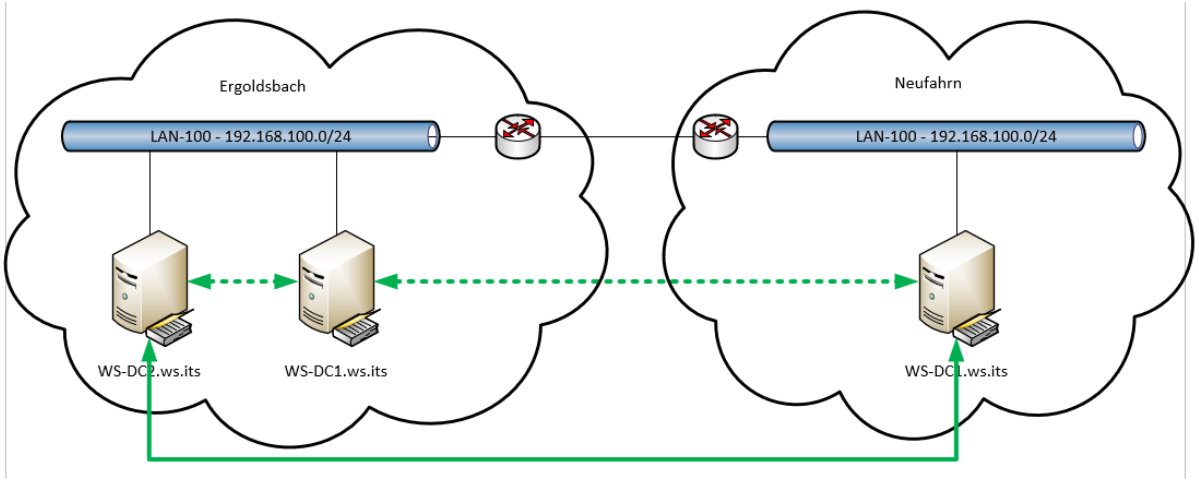




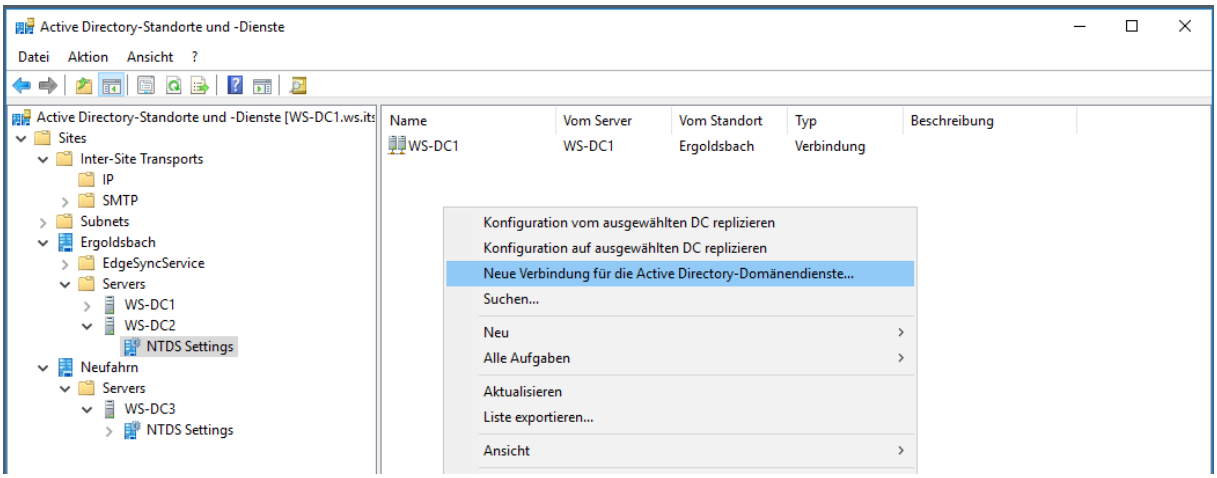
Danach habe ich nur noch einen aktiven DHCP-Server. Mit dem DHCP-Failover ist ein Wechsel so einfach.

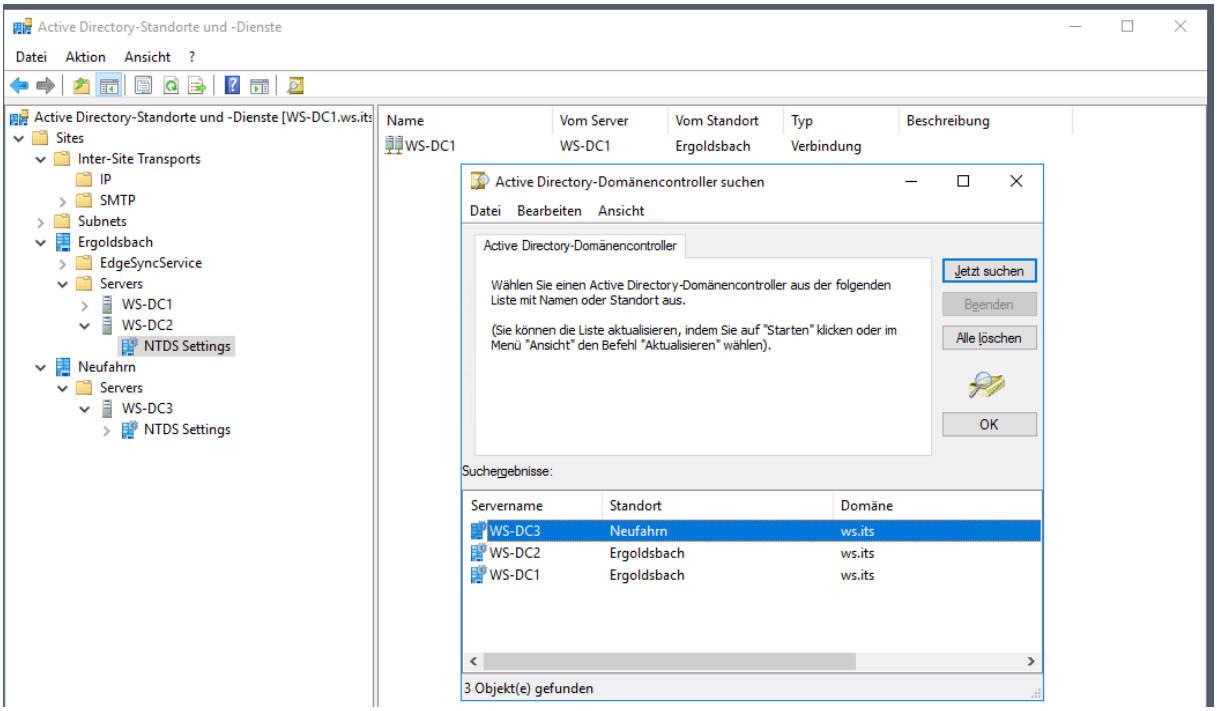
Entfernen der Rolle Active Directory

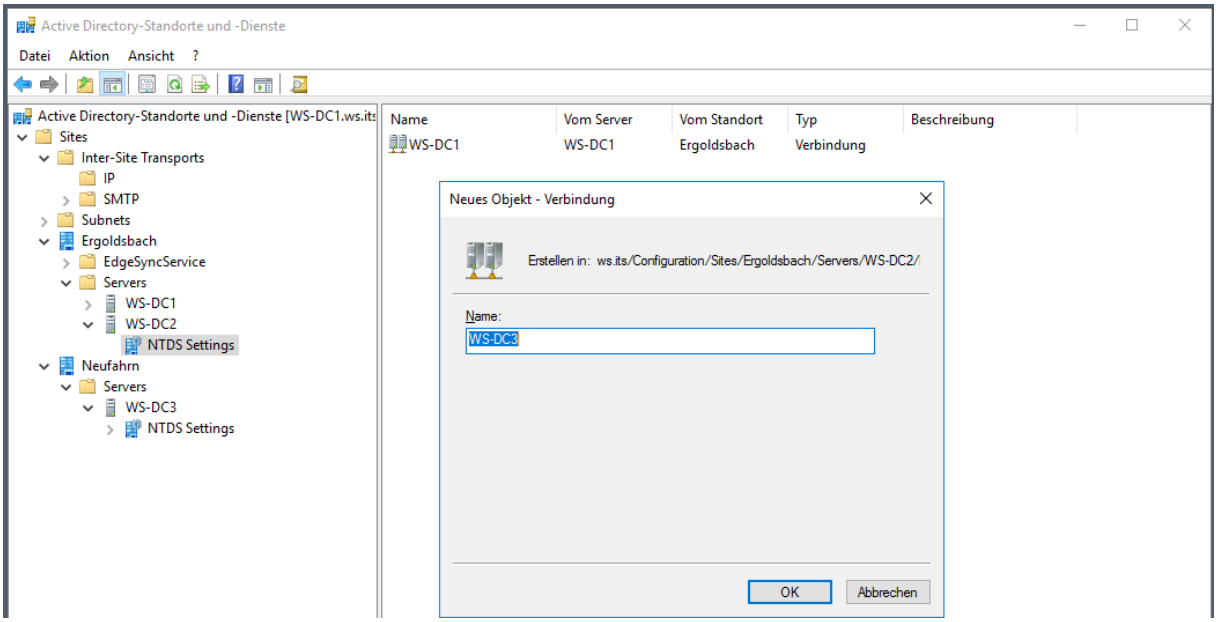
Weiter geht es mit dem Active Directory. Hier baue ich temporär eine alternative Replikationstopologie auf. Der Server WS-DC2 soll sich direkt mit WS-DC3 austauschen können:

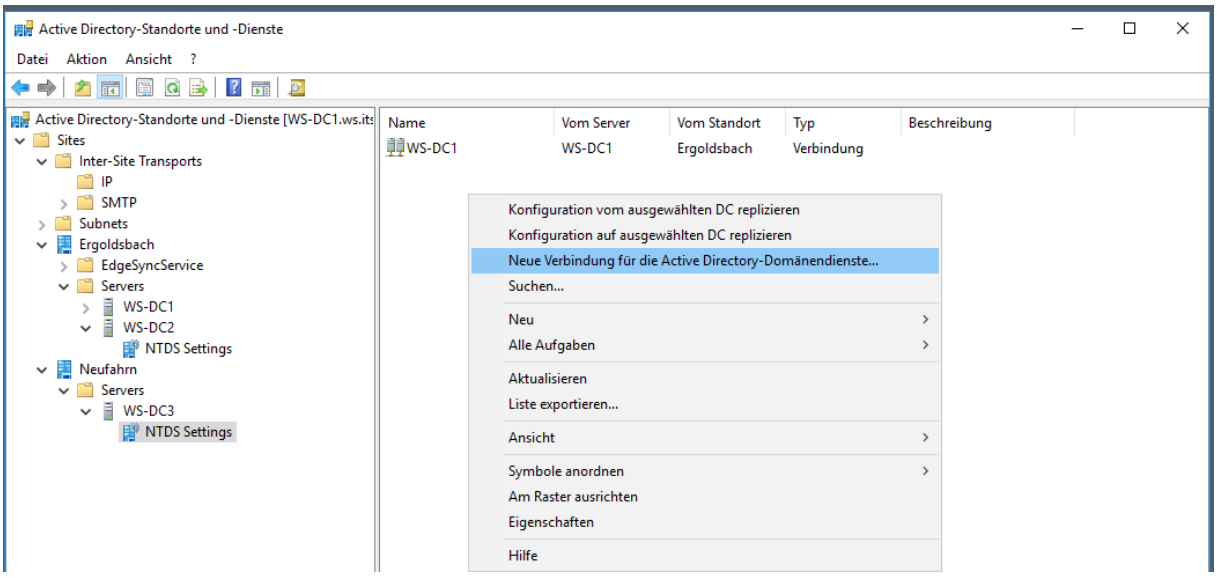


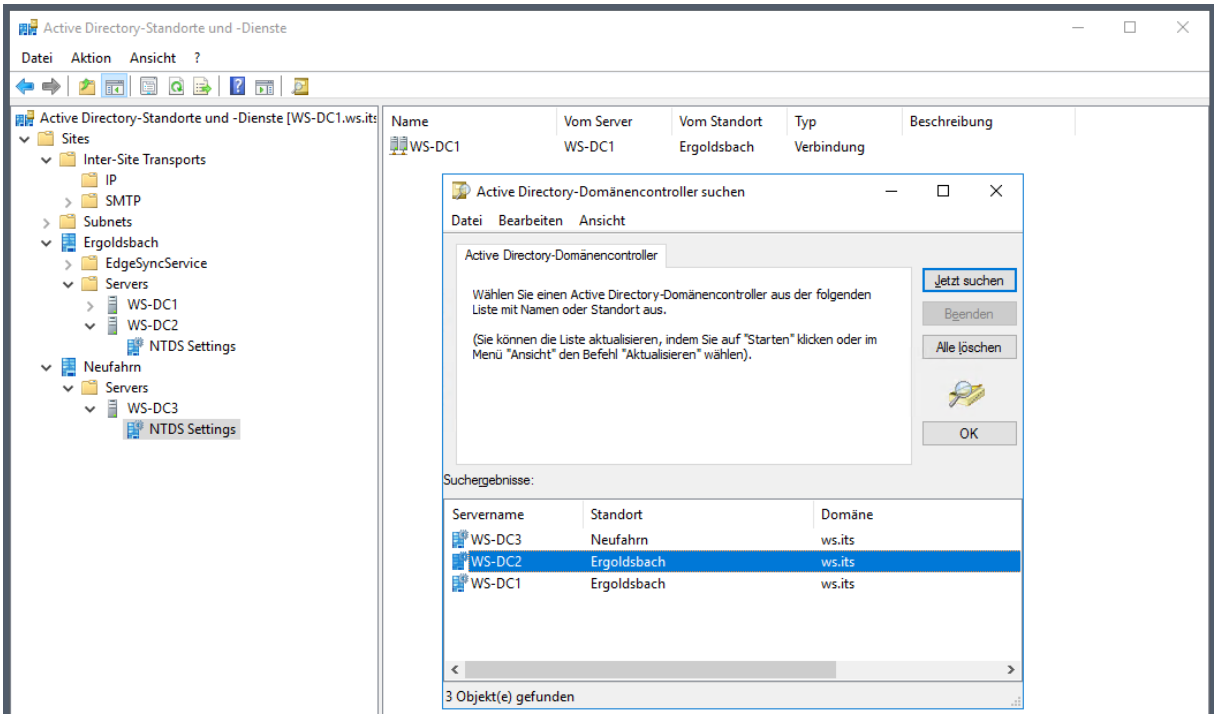
Die Verbindungen können leicht im Active Directory Sites and Services erstellt werden:



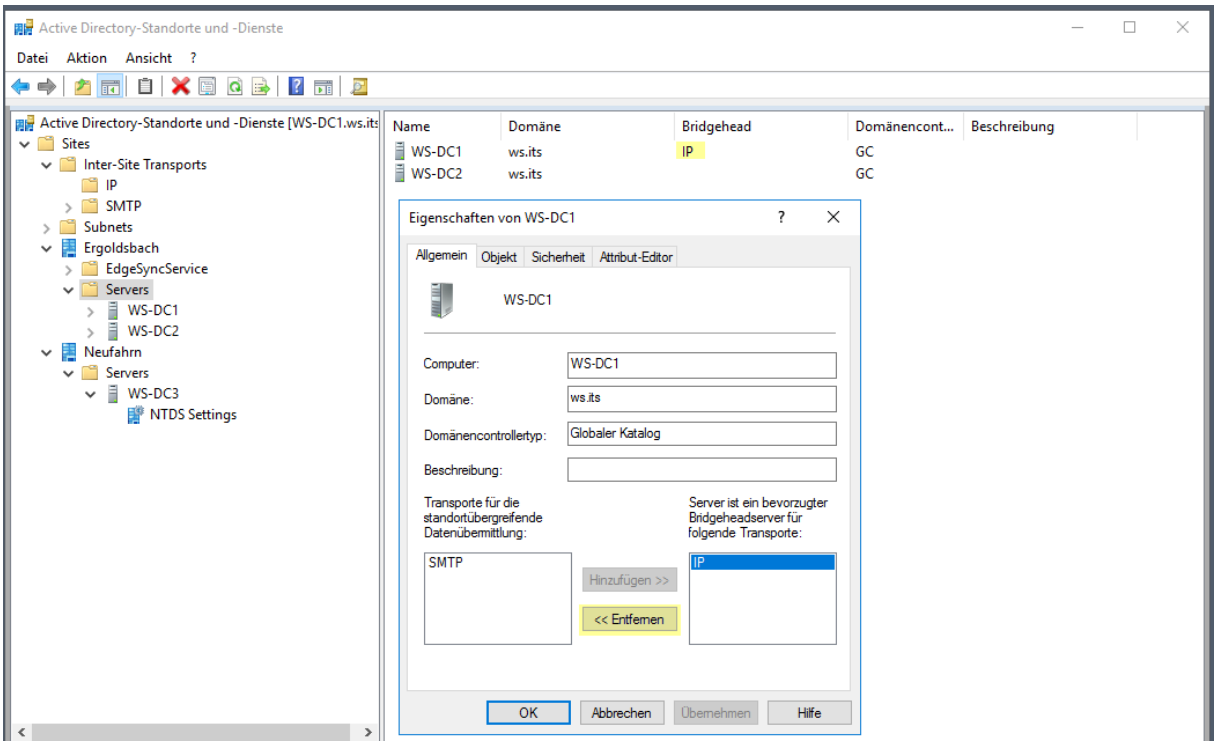



Active Directory-Standorte und -Dienste
Datei Aktion Ansicht ?
Active Directory-Standorte und -Dienste [WS-DC1.ws.its
Sites
Inter-Site Transports
IP
SMTP
Subnets
Ergoldsbach
EdgeSyncService
Servers
WS-DC1
WS-DC2
NTDS Settings
Neufahrn
Servers
WS-DC3
NTDS Settings
Name
Vom Server
Vom Standort
Typ
Beschreibung
WS-DC1
WS-DC1
Ergoldsbach
Verbindung
Neues Objekt - Verbindung
Erstellen in: ws.its/Configuration/Sites/Ergoldsbach/Servers/WS-DC2/
Name:
WS-DC3
OK
Abbrechen

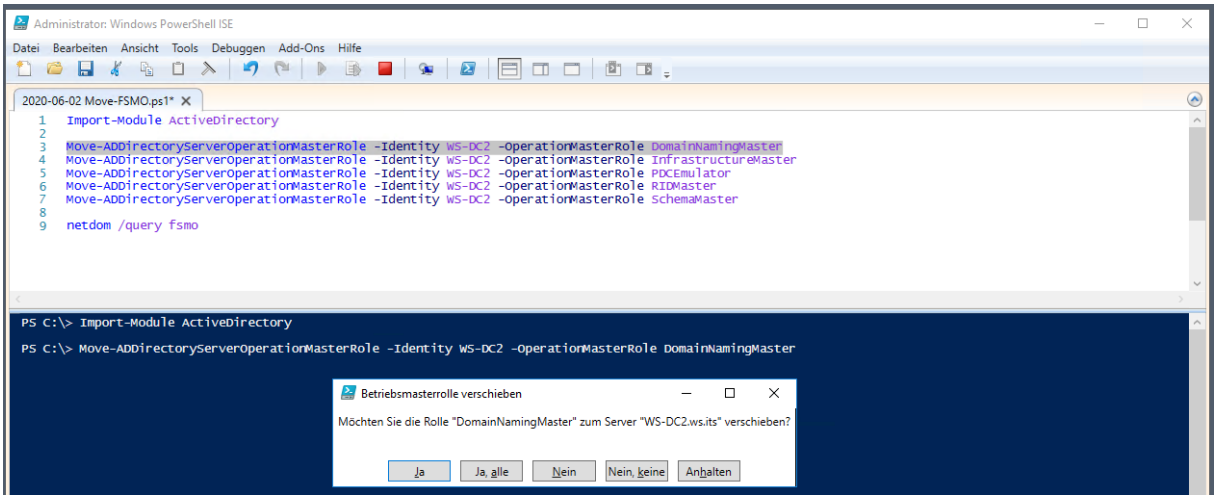

Active Directory-Standorte und -Dienste
Datei Aktion Ansicht ?
Active Directory-Standorte und -Dienste [WS-DC1.ws.its
Sites
Inter-Site Transports
IP
SMTP
Subnets
Ergoldsbach
EdgeSyncService
Servers
WS-DC1
WS-DC2
NTDS Settings
Neufahrn
Servers
WS-DC3
NTDS Settings
Name
Vom Server
Vom Standort
Typ
Beschreibung
WS-DC1
WS-DC1
Ergoldsbach
Verbindung
Konfiguration vom ausgewählten DC replizieren
Konfiguration auf ausgewählten DC replizieren
Neue Verbindung für die Active Directory-Domänendienste...
Suchen...
Neu
Alle Aufgaben
Aktualisieren
Liste exportieren...
Ansicht
Symbole anordnen
Am Raster ausrichten
Eigenschaften
Hilfe



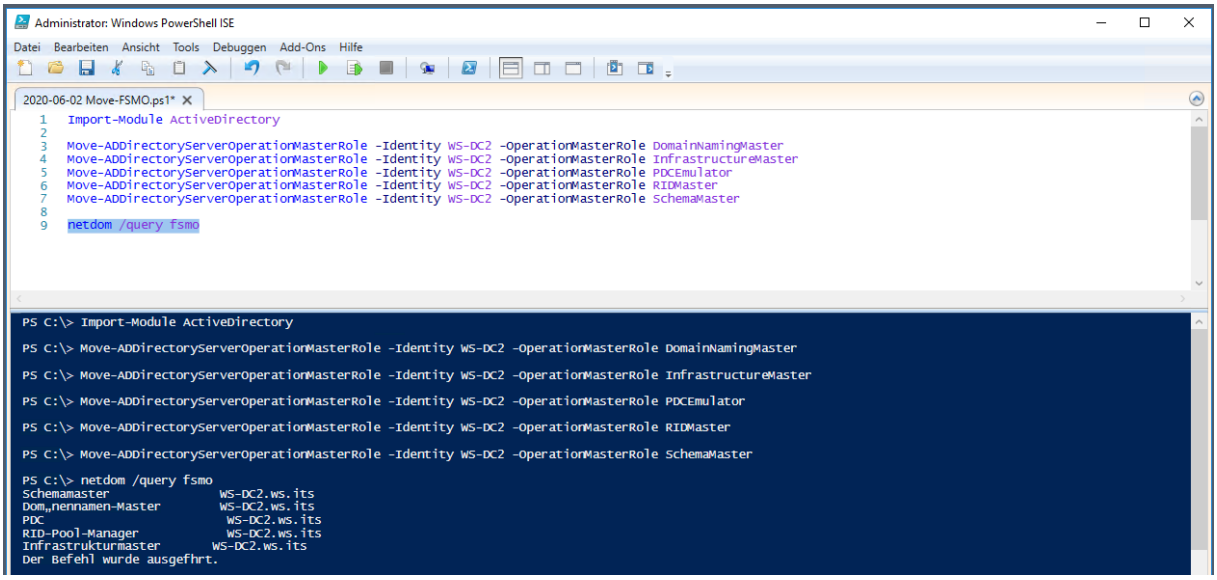
Danach entferne ich die IP-Bridgehead-Funktion vom Server WS-DC1:



Die Anpassung der Replikation wird ein paar Minuten dauern. Währendessen verschiebe ich die FSMO-Rollen. Da hilft die PowerShell. Der neue Träger wird mein Server WS-DC2:



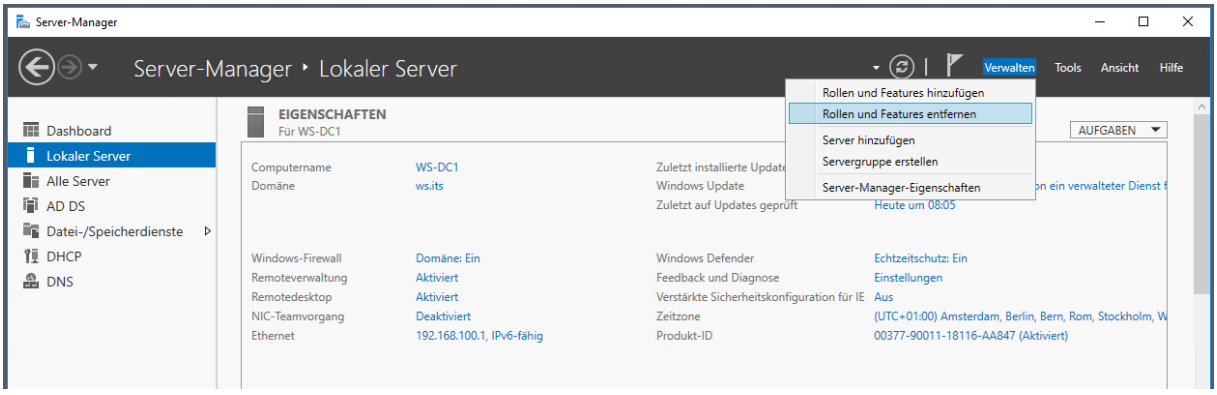
Die Rollen werden live verschoben. Eine Kontrolle bestätigt den neuen Owner:



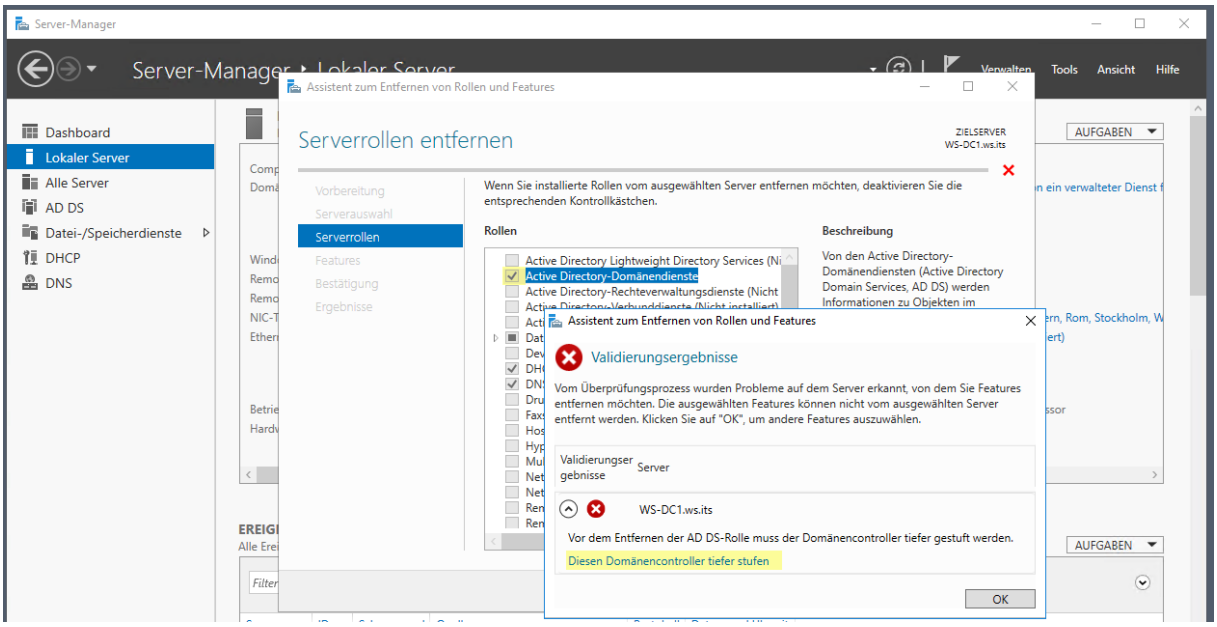
Auch nach einer solchen Aktion sollte man sich einen Kaffee holen. Die DCs benötigen unter Umständen einige Minuten für die Anpassung.

Praxistipp:
Der PDC-Emulator ist immer noch der primäre Zeitserver unter den Domain Controllern. Wird dieser für eine längere Zeit auf einen anderen DC verschoben, dann sollte dort auch die NTP-Konfiguration angepasst werden.

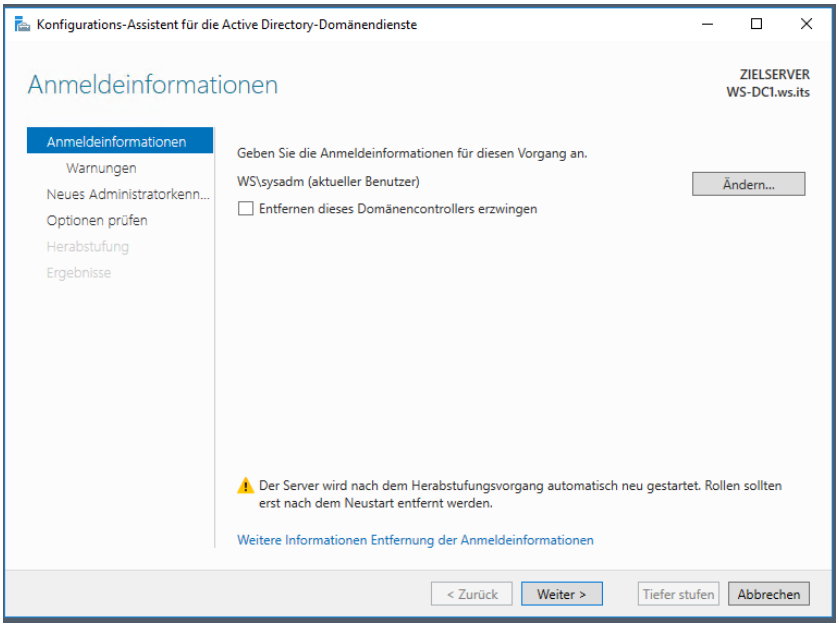
Ich verzichte auf eine Zeitkonfiguration. Der neue Server ist in knapp einer Stunde einsatzbereit. Ich starte die Herabstufung des Domain Controllers durch die Deinstallation der Rolle Active Directory:

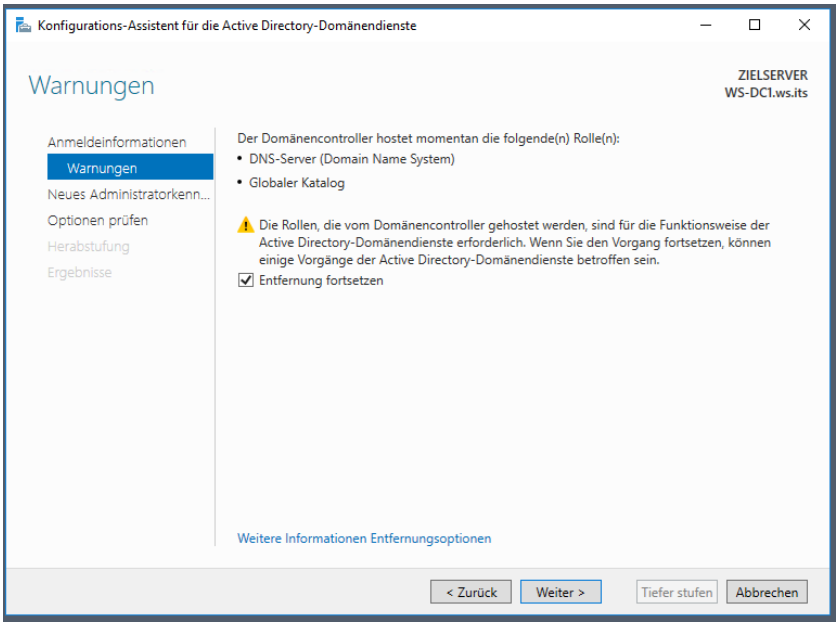


Bei diesem Vorgang hält mich der Server Manager auf. So kann ich das Demoting einleiten:

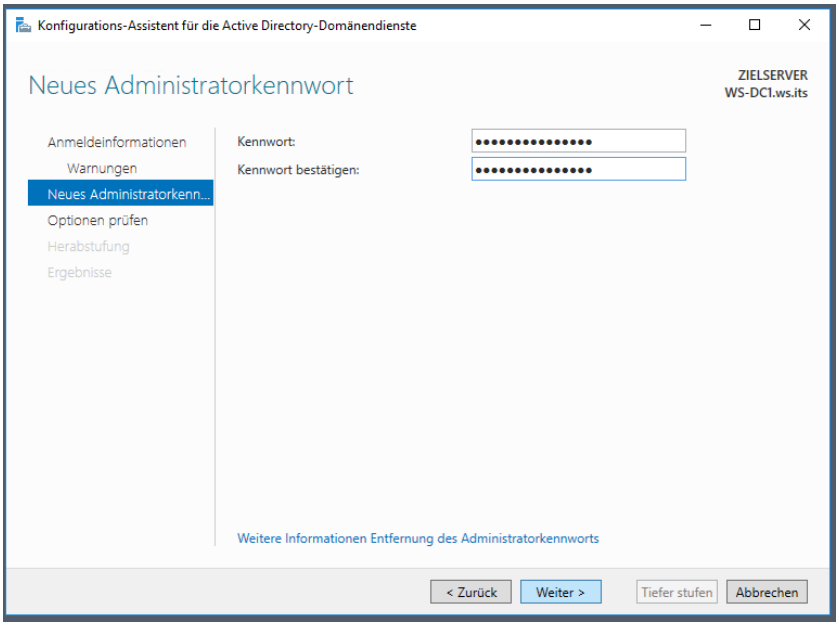


Das Entfernen eines Domain Controllers ist recht einfach:

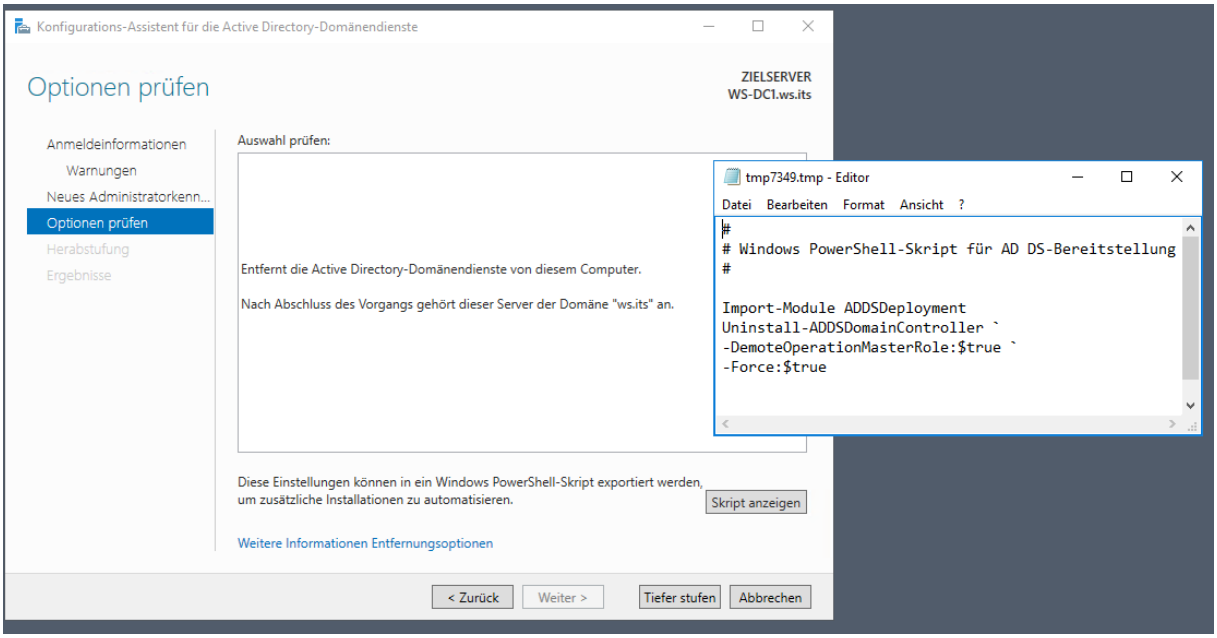




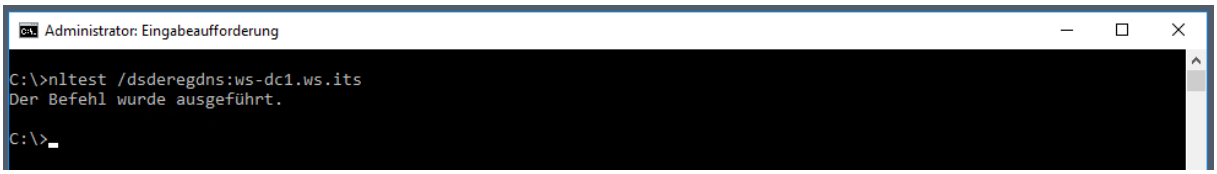
Das Passwort wird für den nach Abschluss wieder aktiven, lokalen Administrator benötigt:



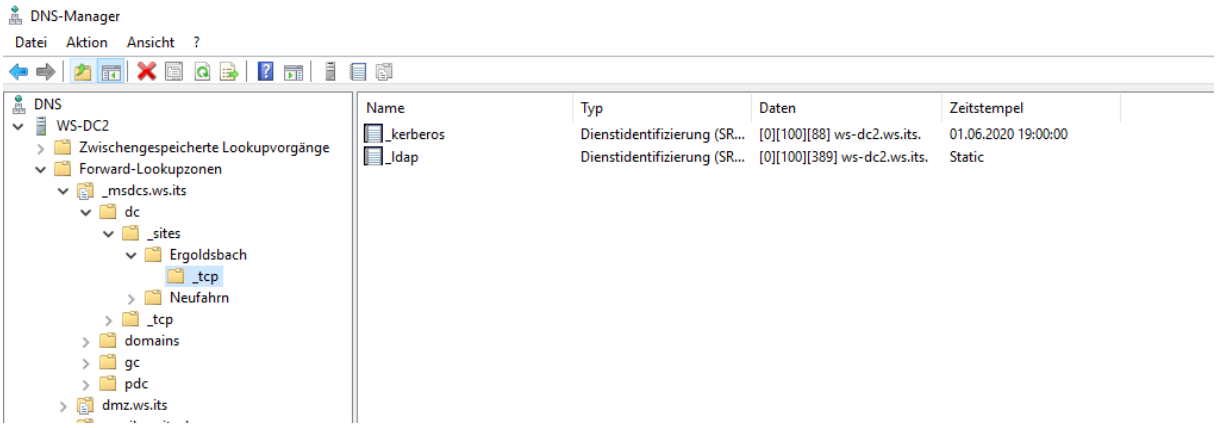
Die Aktion lässt sich auch scripten, aber mit ein paar Mausklicks geht es dann doch schneller:



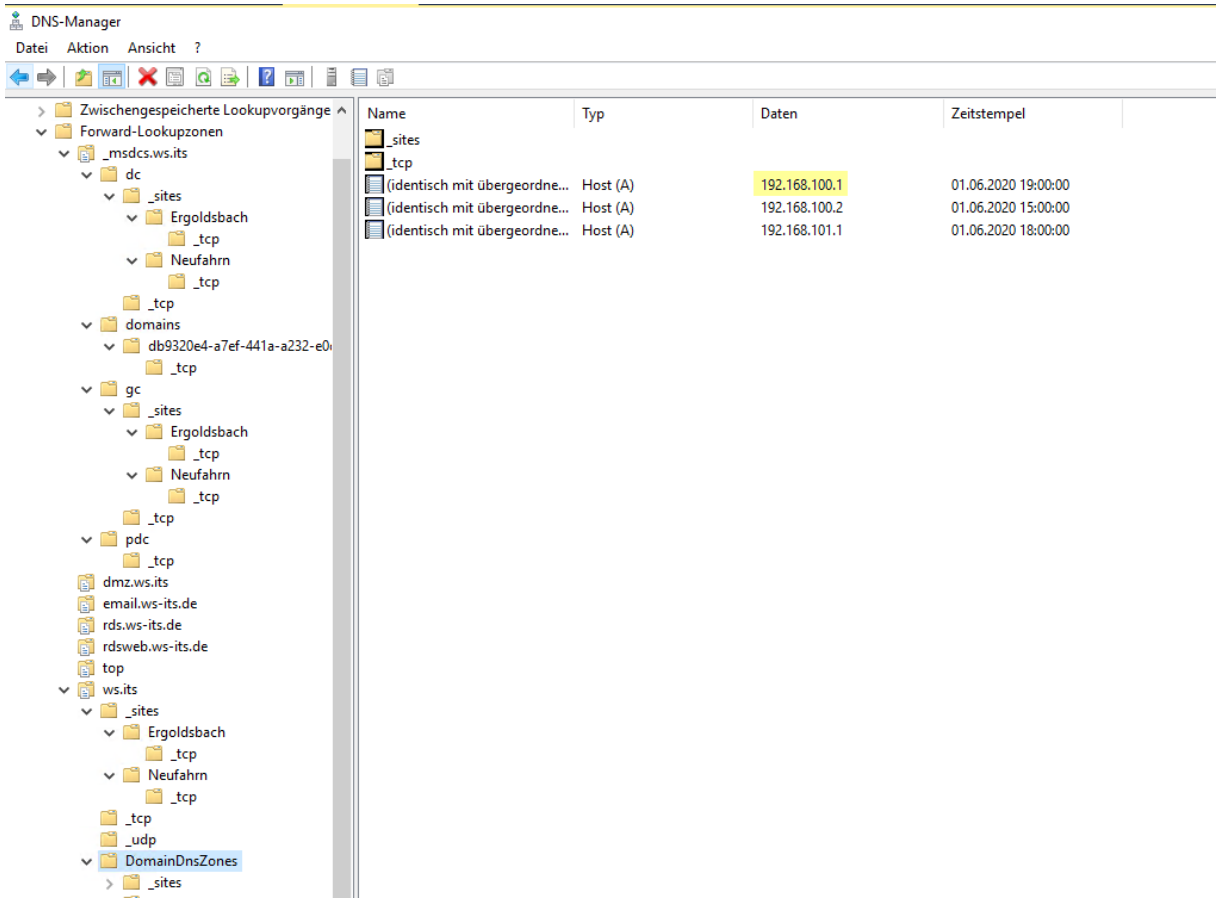
Bevor ich den Prozess starte, leite ich eine Art Active Directory Maintenance ein, indem ich seine DNS-Records aus den DNS-Servern entferne. Der Domain Controller wird üblicherweise über DNS gefunden. Ohne seine Records werden die Clients auf andere DCs gelenkt. Die Records haben eine TTL von 10 Minuten. Diese sollte man dazu noch abwarten. Danach kennt den DC kein Client mehr:

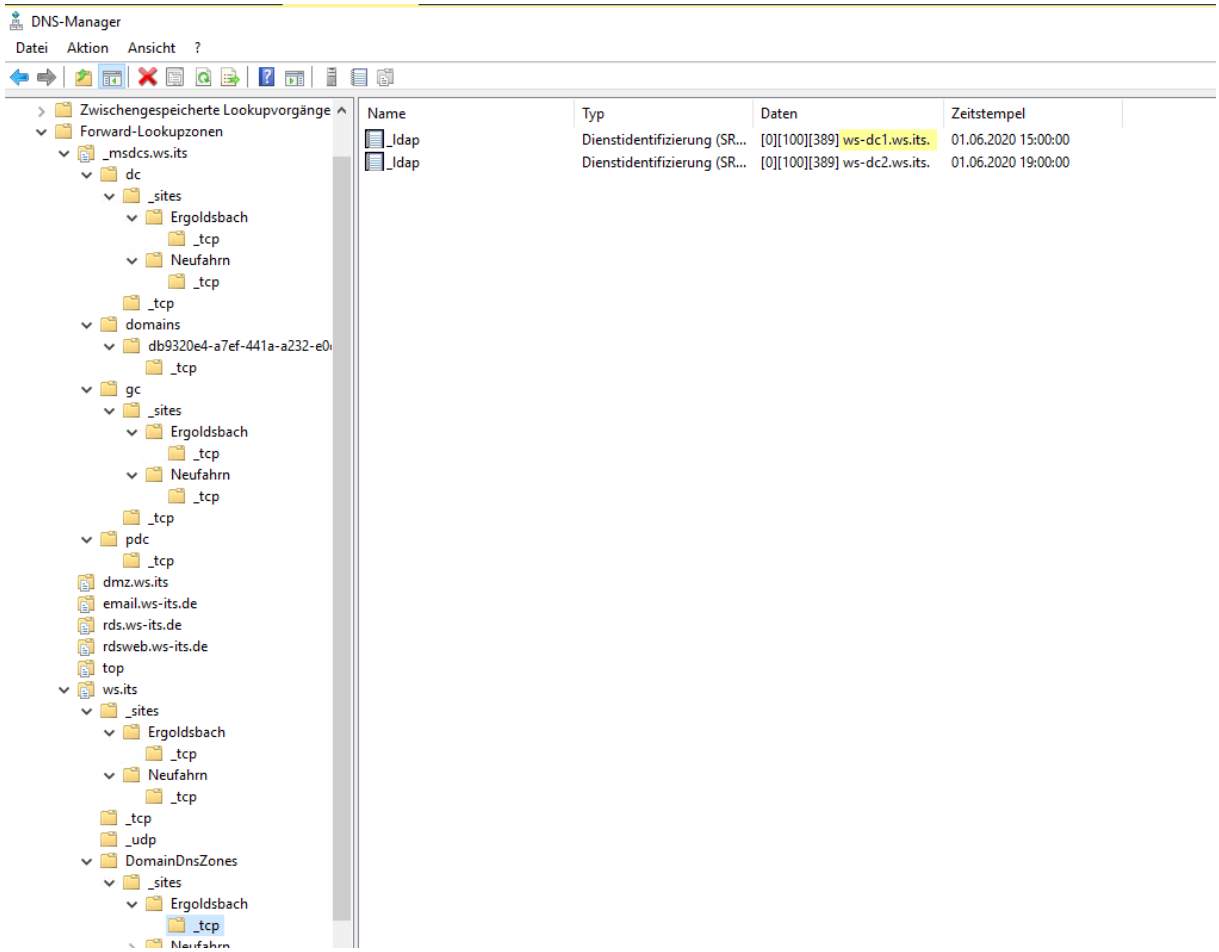


Hier sieht man das Ergebnis in einem Container im DNS. Da stand bis eben auch der WS-DC2 als Kerberos- und LDAP-Server:



Die Wartezeit nutze ich zur Kontrolle im DNS. Nltest kann nicht alle Records entfernen. Die überzähligen lösche ich manuell:

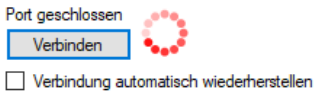

DNS-Manager
Datei Aktion Ansicht ?
Zwischengespeicherte Lookupvorgänge
Forward-Lookupzonen
_msdcs.ws.its
dc
_sites
Ergoldsbach
_tcp
Neufahrn
_tcp
_tcp
domains
db9320e4-a7ef-441a-a232-e0
_tcp
gc
_sites
Ergoldsbach
_tcp
Neufahrn
_tcp
_tcp
pdc
_tcp
dmz.ws.its
email.ws-its.de
rds.ws-its.de
rdsweb.ws-its.de
top
ws.its
_sites
Ergoldsbach
_tcp
Neufahrn
_tcp
_tcp
_udp
DomainDnsZones
_sites
Name
Typ
Daten
Zeitstempel
_sites
_tcp
(identisch mit übergeordne...
Host (A)
192.168.100.1
01.06.2020 19:00:00
(identisch mit übergeordne...
Host (A)
192.168.100.2
01.06.2020 15:00:00
(identisch mit übergeordne...
Host (A)
192.168.101.1
01.06.2020 18:00:00


DNS-Manager
Datei Aktion Ansicht ?
Zwischengespeicherte Lookupvorgänge
Forward-Lookupzonen
_msdcs.ws.its
dc
_sites
Ergoldsbach
_tcp
Neufahrn
_tcp
_tcp
domains
db9320e4-a7ef-441a-a232-e0
_tcp
gc
_sites
Ergoldsbach
_tcp
Neufahrn
_tcp
_tcp
pdc
_tcp
dmz.ws.its
email.ws-its.de
rds.ws-its.de
rdsweb.ws-its.de
top
ws.its
_sites
Ergoldsbach
_tcp
Neufahrn
_tcp
_tcp
_udp
DomainDnsZones
_sites
Ergoldsbach
_tcp
Name
Typ
Daten
Zeitstempel
_ldap
Dienstidentifizierung (SR...
[0][100][389] ws-dc1.ws.its.
01.06.2020 15:00:00
_ldap
Dienstidentifizierung (SR...
[0][100][389] ws-dc2.ws.its.
01.06.2020 19:00:00

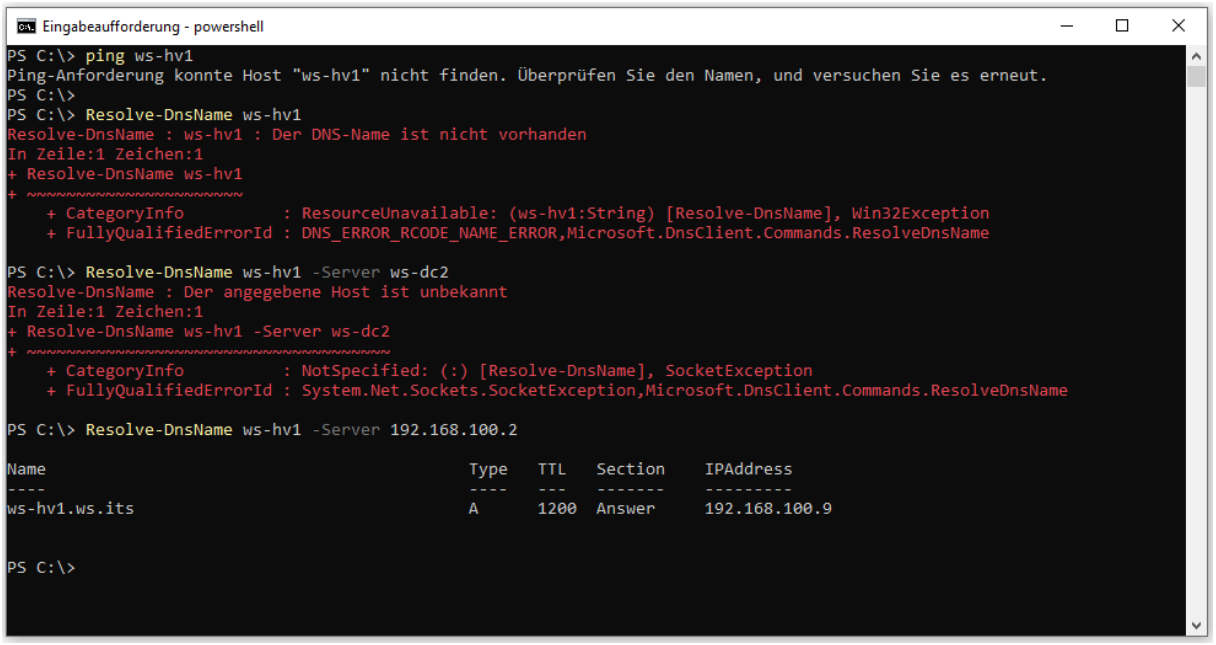
Das Entfernen des Domain Controllers läuft fehlerfrei. Anschließend startet der Server als Memberserver neu.

Troubleshooting nach dem Herabstufen

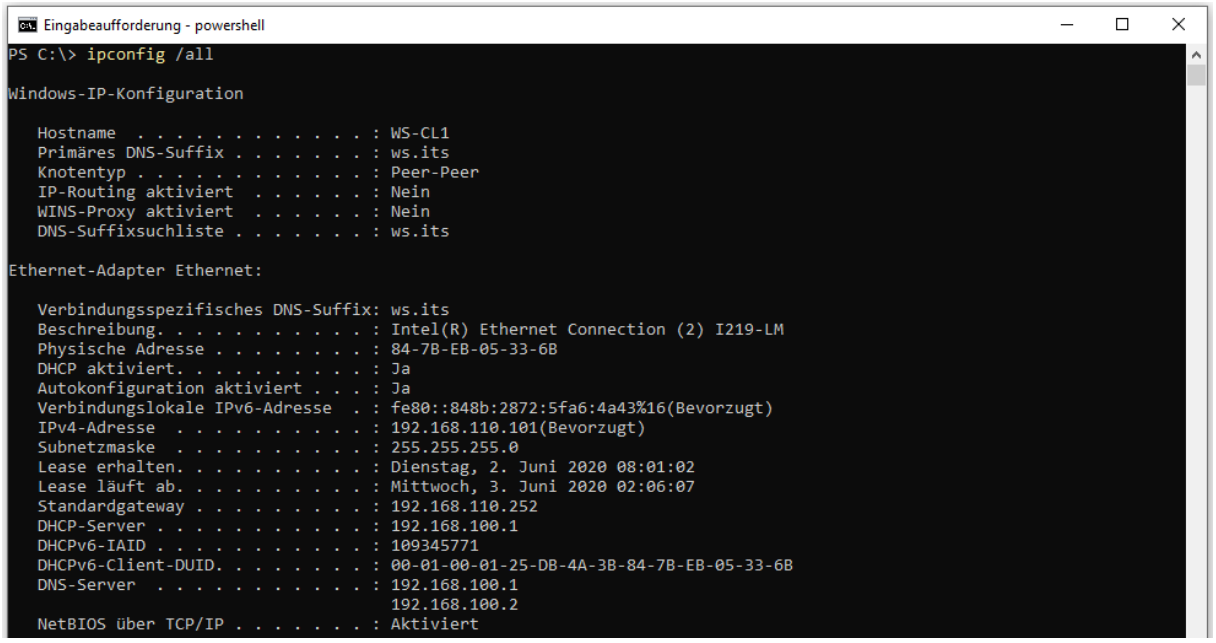
Aber jetzt habe ich ein Problem. Ich kann auf keinen meiner Server mehr zugreifen. Mein Remoting-Tool zeigt dieses Fenster für jeden Server:



Das kann eigentlich nur an der Namensauflösung liegen. Diese teste ich am Client:



Die Ursache ist einfach – wenn zugleich auch fatal: Der Server WS-DC1 ist für etwa die Hälfte meiner Clients und Server der primäre DNS-Server gewesen. Zusätzlich habe ich in meiner Infrastruktur alle anderen Namensauflösungs-Methoden (NBNS, LLMNR) deaktiviert. Eigentlich sollten jetzt alle Systeme den einsatzbereiten WS-DC2 für die DNS-Namensauflösung verwenden:



Aber ein DNS-Client mit mehreren konfigurierten DNS-Servern hat eine feste Arbeitsweise:

1. Alle DNS-Server werden gemäß ihrer konfigurierten Reihenfolge kontaktiert.
2. Reagiert ein DNS-Server nicht auf eine Anfrage, dann wird er aus der Liste temporär entfernt und der Client befragt den nächsten Server der Liste. Der Timeout eines DNS-Servers liegt dabei bei etwa 20 Sekunden.
3. Sind alle DNS-Server der Liste temporär entfernt, dann schlägt die DNS-Anfrage fehl.

Aber der WS-DC1 ist doch kein DNS-Server mehr und so sollten doch alle Systeme den funktionalen Server WS-DC2 befragen. Oder? Eben nicht: der Server WS-DC1 ist immer noch ein DNS-Server. Nur hat er eben keinen Zugriff mehr auf die Active Direcory integrierten Zonen! Er bekommt also eine Anfrage für z.B. ws-hv1.ws.its, hat aber keine Zone, in welcher er nachsehen könnte. Also antwortet er mit „non existent“. Und genau das ist mein Problem: **er antwortet**! Damit ist er als DNS-Server vom Client bestätigt. Warum sollte der Client also einen sekundären DNS-Server konaktieren? Nur, weil ihm die Antwort nicht gefällt?

Jetzt habe ich mehrere Optionen:

1. Ich muss mich mit dem Server WS-DC1 verbinden, um den Service DNS zu beenden. Dann kann ich ihn deinstallieren und alles ist gut.
2. Ich könnte mich auch mit dem Hyper-V-Host verbinden, die Netzwerkkarte des virtuellen DCs deaktivieren und dann die Rolle DNS über die Konsole deinstallieren.

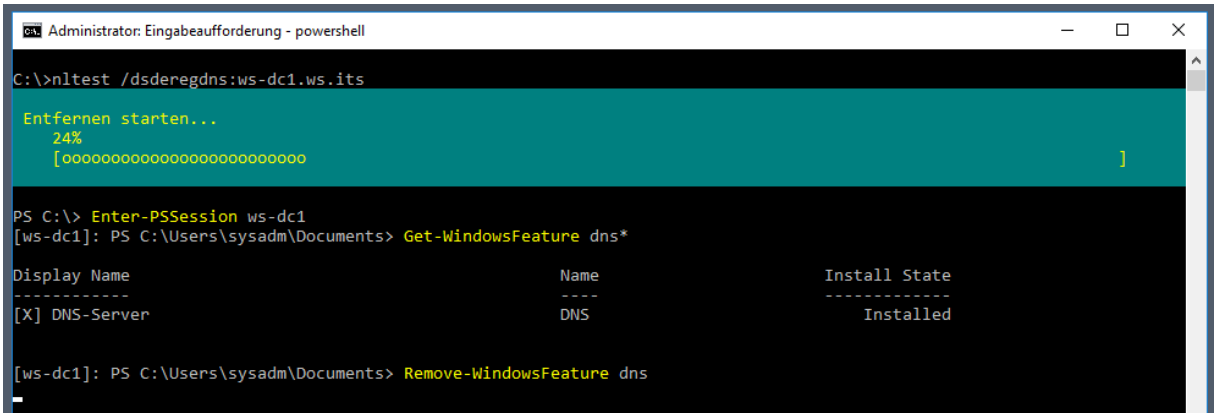
Eigentlich ist das auch kein Problem. Man verbindet sich einfach mit mstsc und der IPv4-Adresse mit dem Zielserver. Aber auch hier grätscht mir meine sichere Infrastruktur rein: Alle meine administrativen Accounts sind Mitglied der Gruppe „Protected Users“. Diese dürfen nur Kerberos für die Authentifizierung verwenden. Und Kerberos kann nur mit dem Namen bei der Verbindung verwendet werden. Die IP-Adresse hat ja keine ServicePrincipalName. Für die IPv4-Verbindung kommt nur NTLM infrage. Dieses Protokoll habe ich aber zusätzlich domänenweit deaktiviert.

Ich kann mich also weder via IPv4 mit dem Server WS-DC1 verbinden, noch kann ich eine Verbindung zum Hyper-V-Host aufbauen. Also bleiben 2 weitere Optionen:

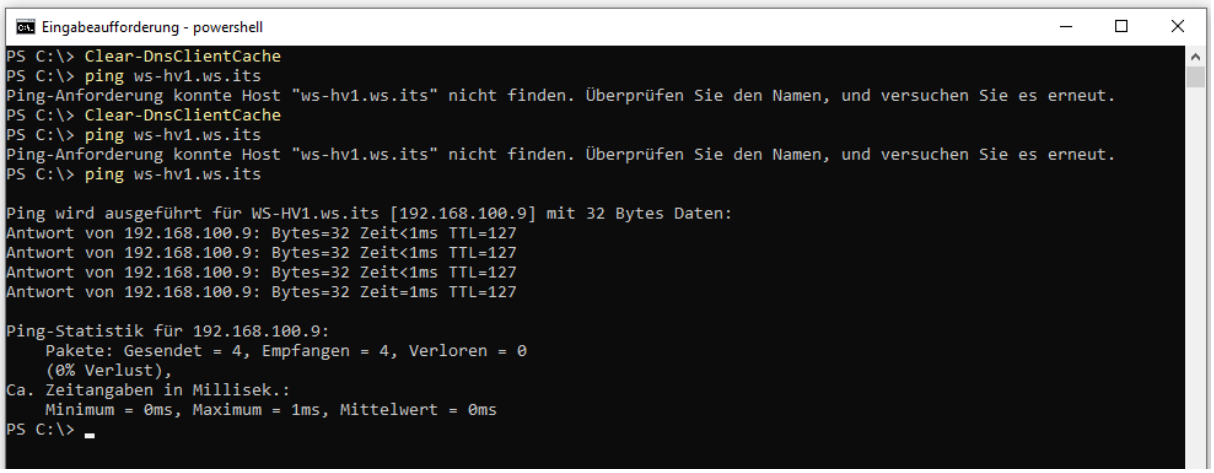
1. Ich verwende einen Notfall-Account für die lokale Anmeldung am Hyper-V-Server und deaktiviere so den alten WS-DC1
2. Oder ich verwende die noch offene Verbindung zum WS-DC2.

Ha, die habe ich ganz übersehen. Eigentlich würde das auch keinen Sinn ergeben, da mein Domain Admin auf Memberservern ja eh keine Rechte hat. Aber der „neue“ Memberserver WS-DC1 ist in CN=Computers,DC=ws,DC=its gelandet. Da greifen meine Gruppenrichtlinien für das Tiermanagement nicht. Also hat der Server in seiner lokalen Gruppe „Administratoren“ die Gruppe „Domain Admins“ aufgenommen. Der Default eben. Also kann ich eine Verbindung über die PowerShell aufbauen. Auf dem WS-DC2 funktioniert die Namensauflösung lokal sehr gut, da er ::1 an der ersten Stelle der

DNS-Serverliste verwendet – das ist seine Loopback-IPv6. Die Verbindung kommt also zustande. So kann ich die Rolle DNS deinstallieren:



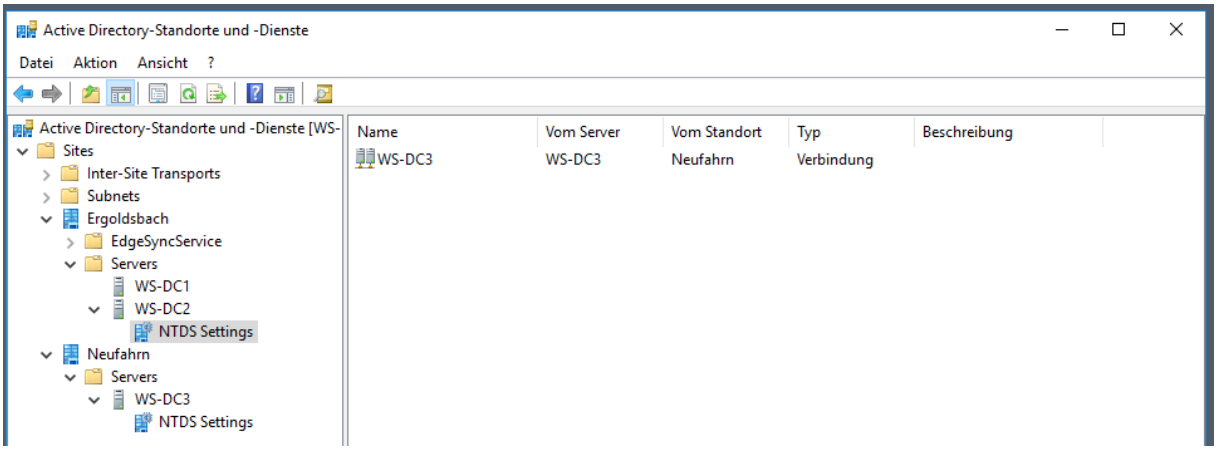
Nach einigen Cache-Bereinigungen schwenkt mein Client nach etwa 20 Sekunden auf den WS-DC2 für die Namensauflösung. Klar, denn ohne die Rolle DNS kann der Server WS-DC1 nicht länger auf Anfragen reagieren und darauf antworten. Er wird also vom Client verworfen:



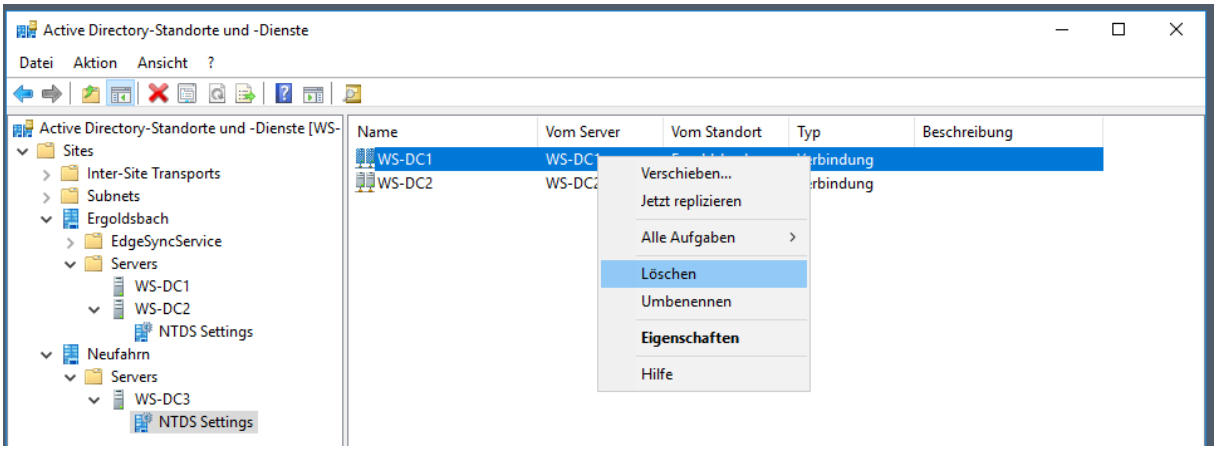
Die Aktion hat etwa 5 Minuten Zeit gekostet und einen Teil meiner Infrastruktur gestört. Nach weiteren 5 Minuten war im Monitoring wieder alles grün. Das zeigt, dass man seine Infrastruktur kennen sollte! Und auch ein wenig Hintergrundwissen ist von Vorteil. Probleme können bei Migrationen entstehen. Man muss nur gezielt darauf reagieren können!

Nacharbeiten im Active Directory

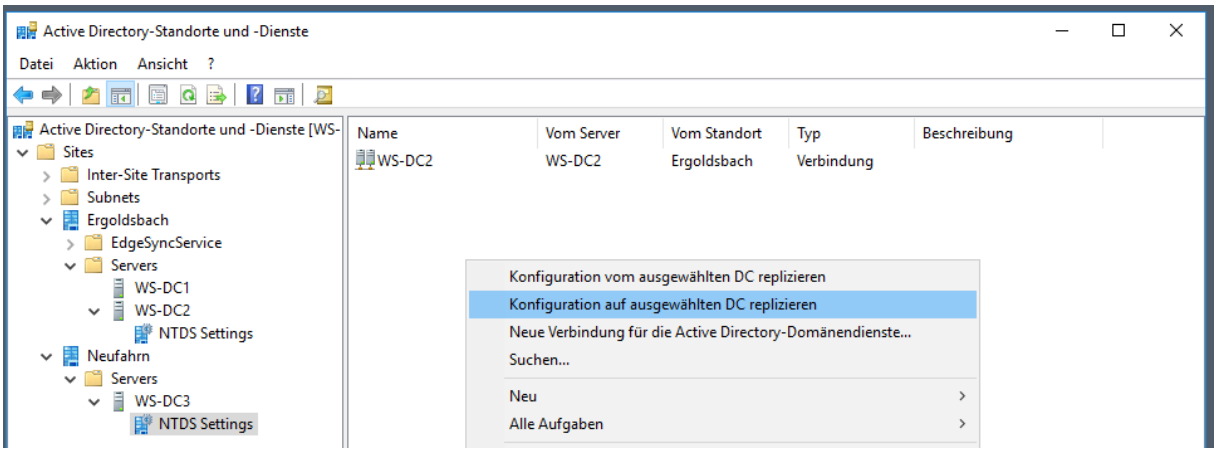
Weiter geht es im Active Directory. Ich muss kontrollieren, ob sich der Server korrekt ausgetragen hat. Dazu kontrolliere ich die Replikationsverbindungen. Der Server WS-DC2 hat den WS-DC1 bereits vergessen:



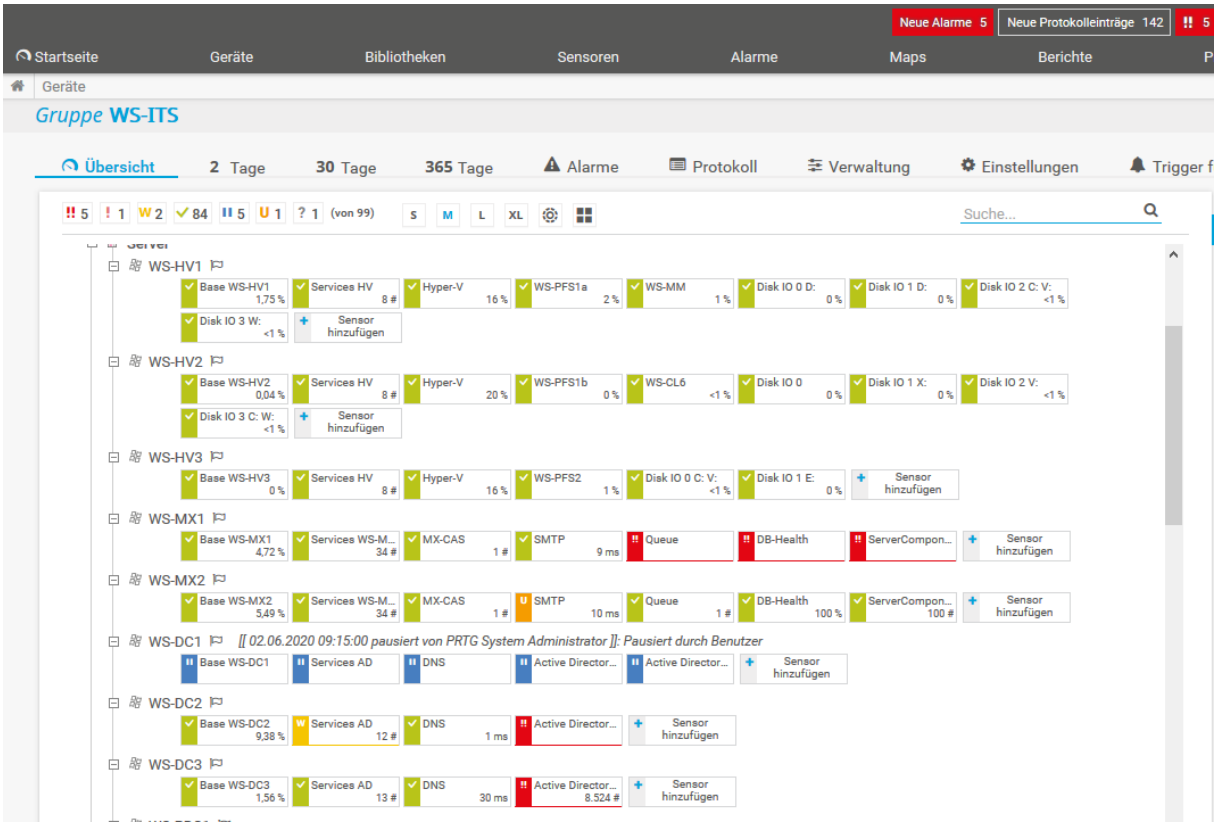
WS-DC1 hat noch eine Verbindung gespeichert. Diese entferne ich manuell. Damit beschleunige ich den Prozess:



Achtung: Das Entfernen habe ich in der Konfigurationspartition vom Server WS-DC2 vorgenommen. Der aktuelle Server kann in der Verbindung oben links im Fenster angezeigt werden. Jetzt muss ich die Veränderung aber noch auf den WS-DC3 replizieren. Natürlich kann ich auch warten. Aber ich will fertig werden:



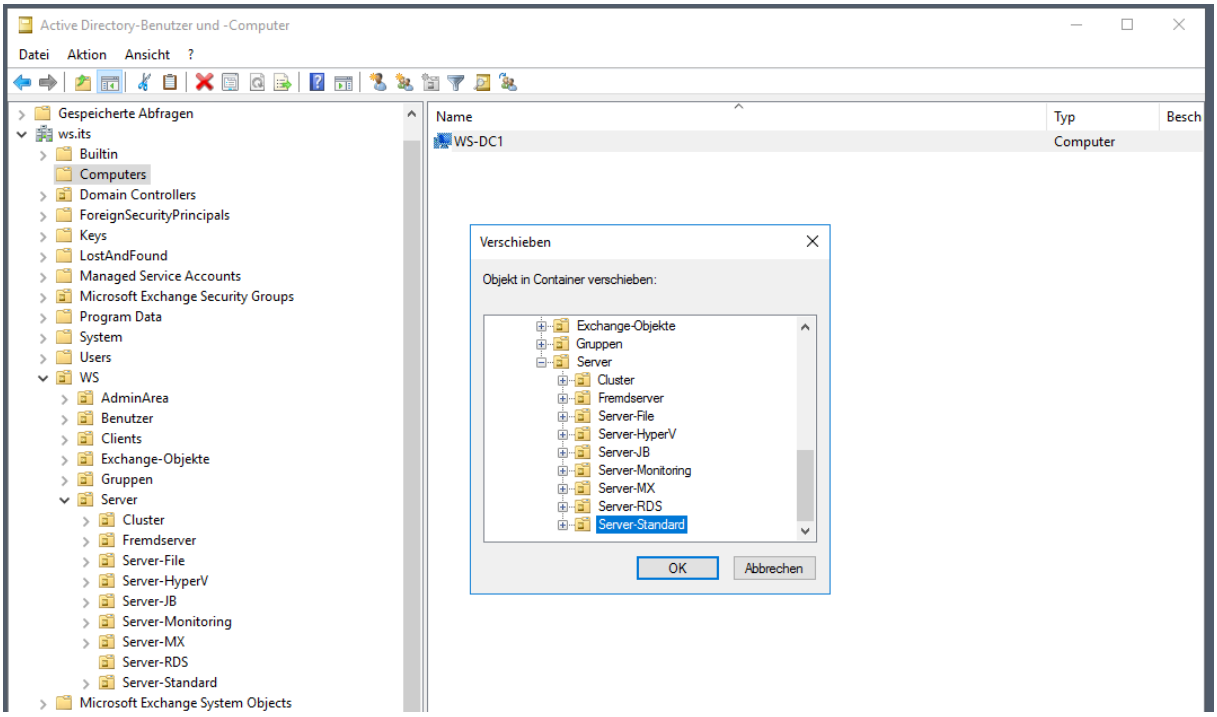
Im Monitoring werden einige Replikationen als ausgefallen gezeigt. Und auch einer meiner Exchange Server ist noch wegen dem DNS-Problem beleidigt:



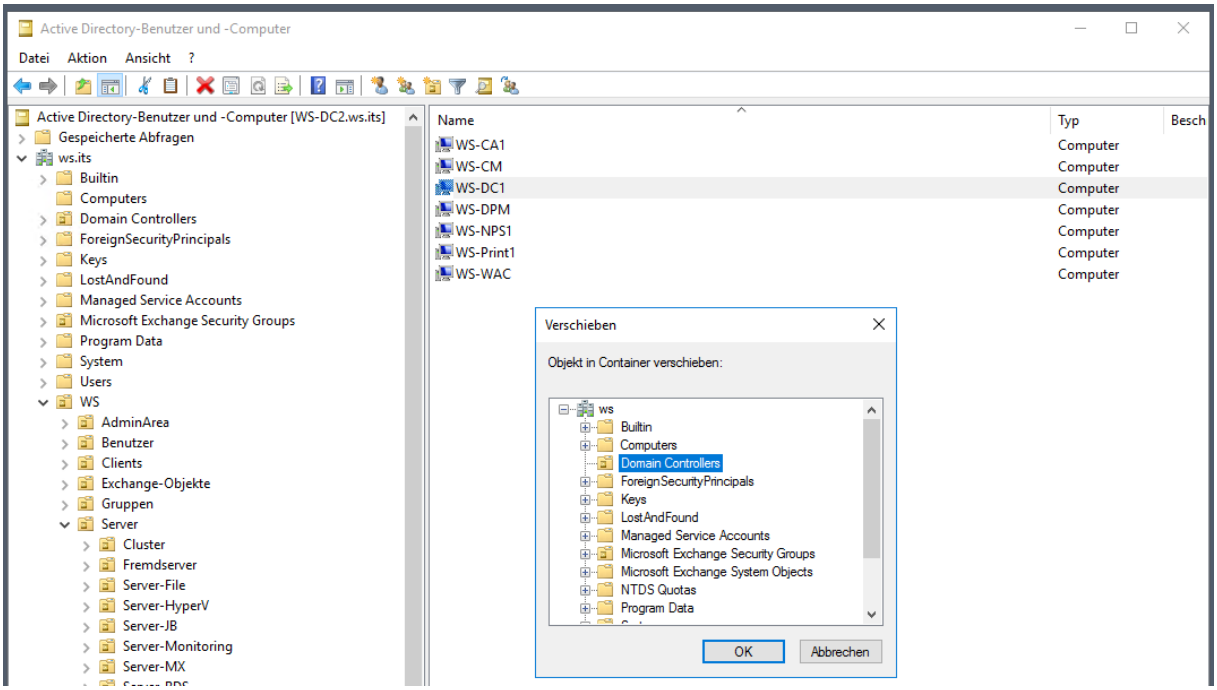
Aber insgesamt ist der alte Domain Controller erfolgreich entfernt worden.

Entfernen des Servers

Ich verschiebe das Computerkonto im Active Directory in eine andere OU:



Aber das war die Falsche. Der Server gehört nachher natürlich in die OU Domain Controllers:

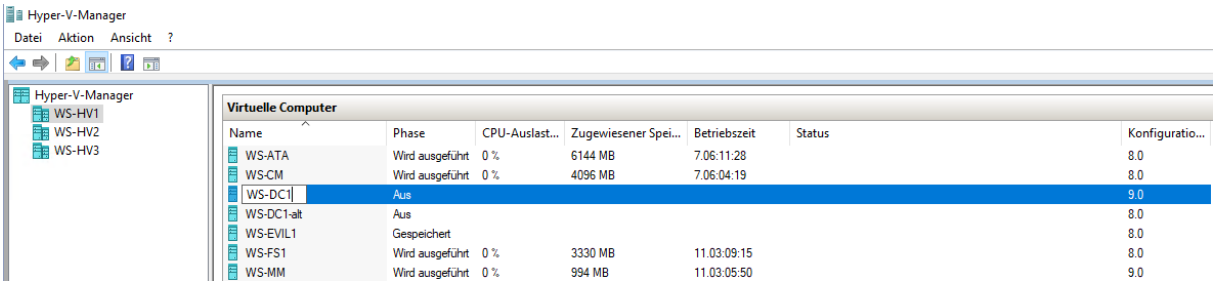


Wenn ich also gleich den neuen Windows Server 2019 in das Active Directory aufnehme, dann wird er von der ersten Sekunde an die Gruppenrichtlinien der Domain Controller verarbeiten.

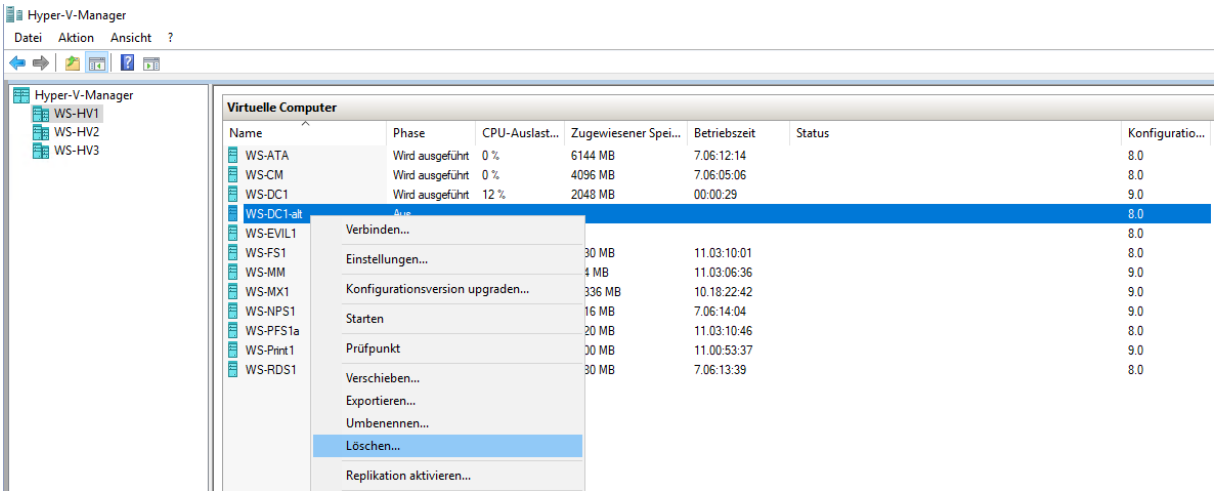
Bereitstellung des neuen Servers

Austausch der VM

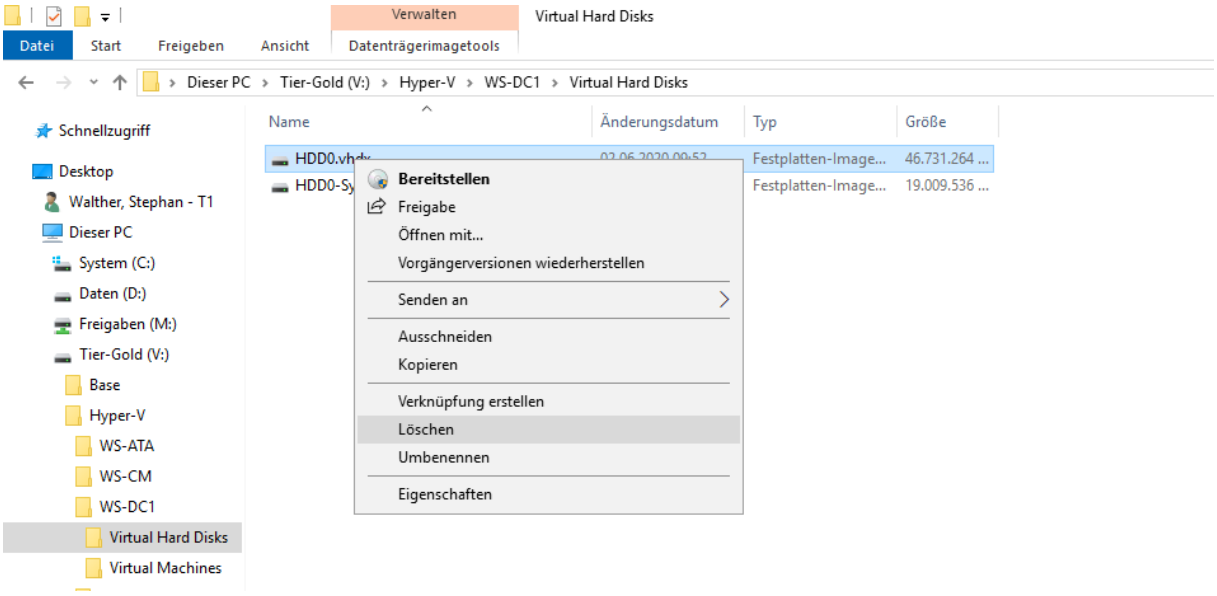
Ich schalte die alte VM aus. Dann ändere ich ihren Namen in „WS-DC1-alt“. Damit ist der Name „WS-DC1“ frei für das neue System:



Den alten Server benötige ich nicht mehr:

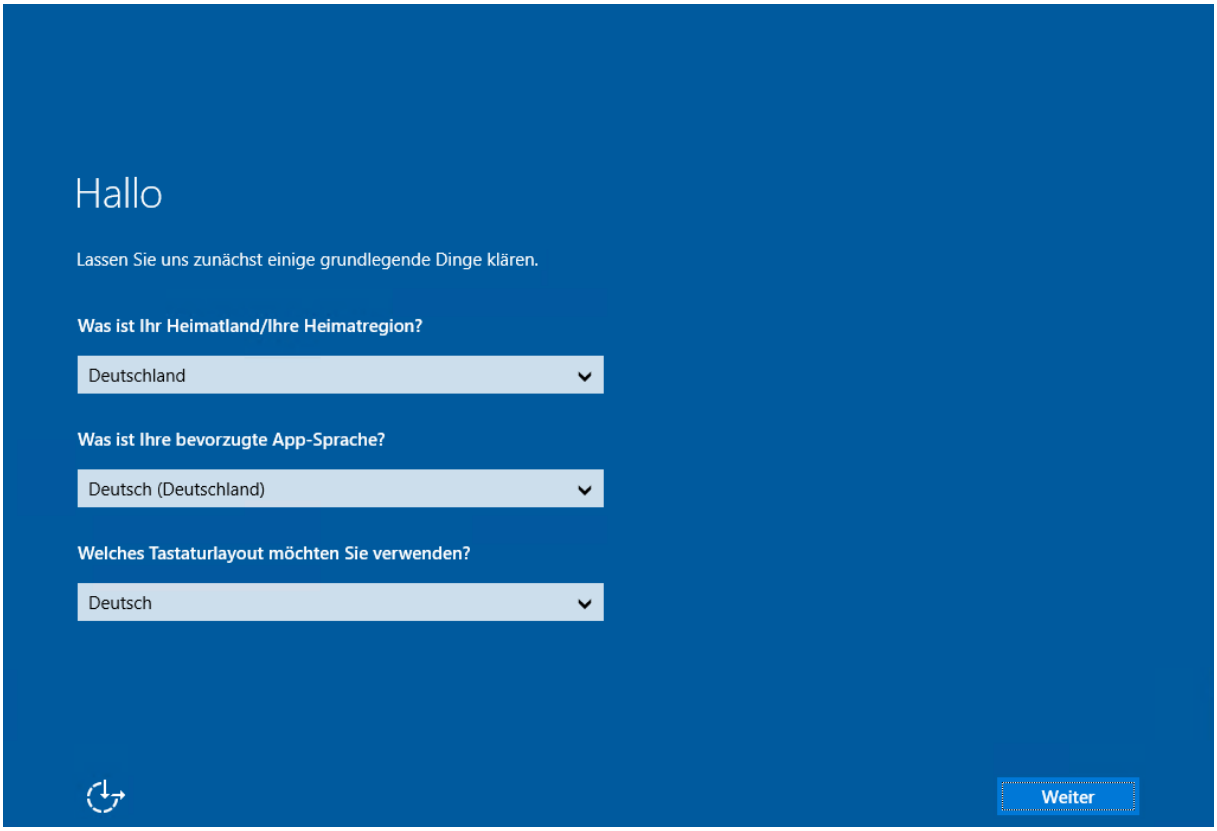


Ebenso lösche ich die alte VHDX-Festplattendatei:

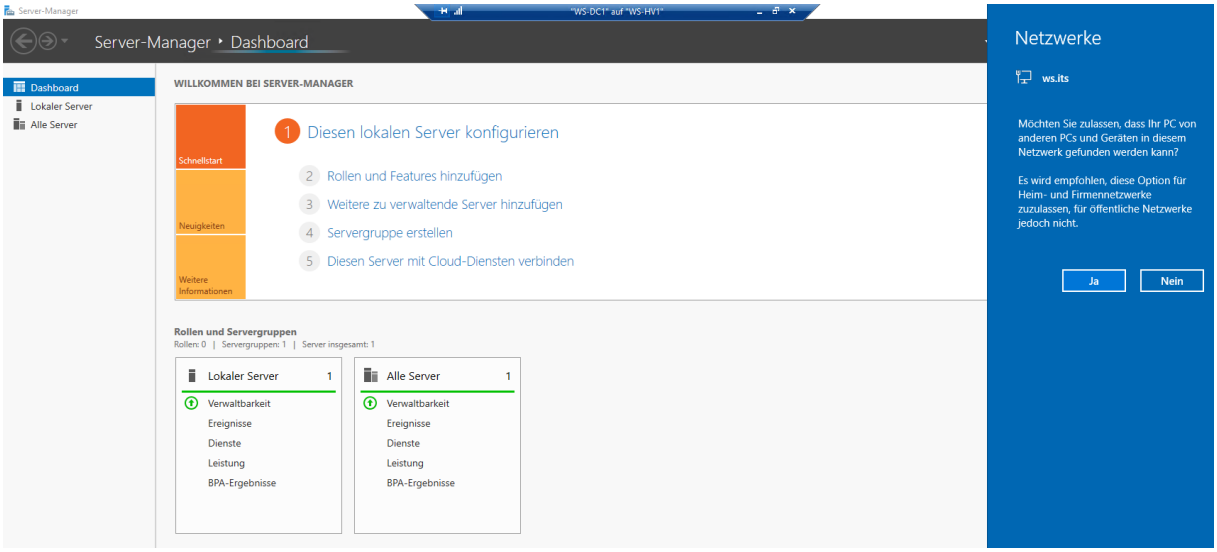


Betriebssystemvorbereitung

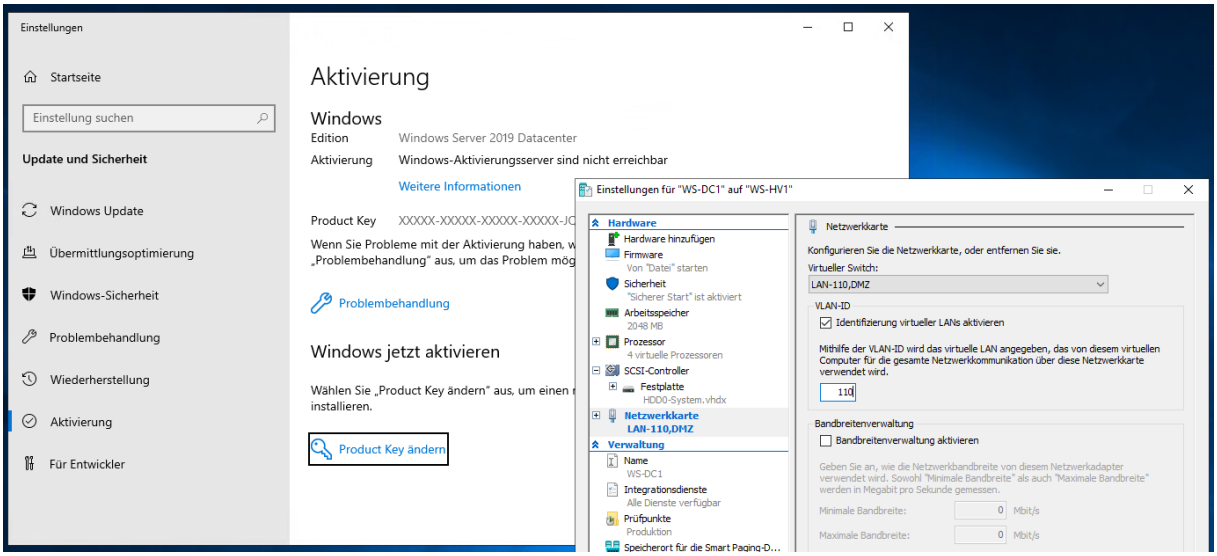
Der neue Server kann gestartet werden. Er durchläuft den Einrichtungsprozess:

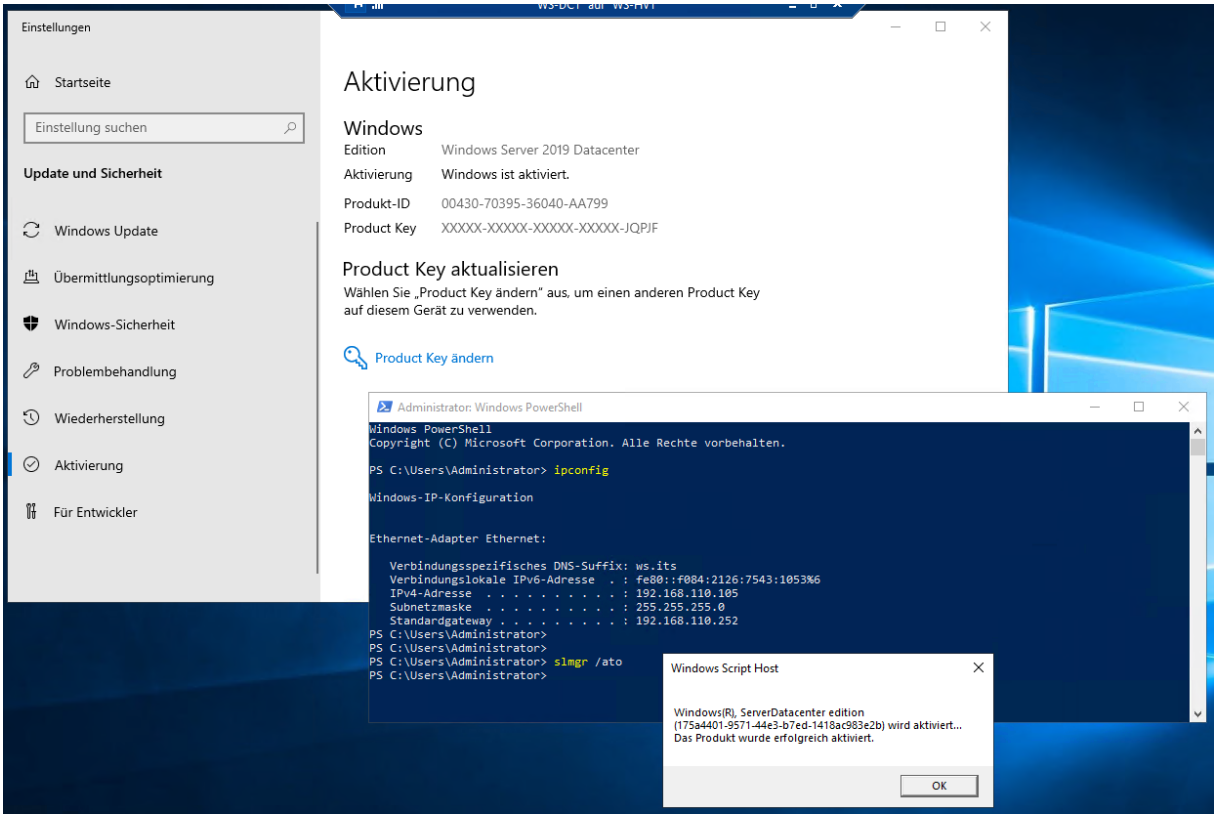


Nach wenigen Eingaben kann ich mich anmelden:

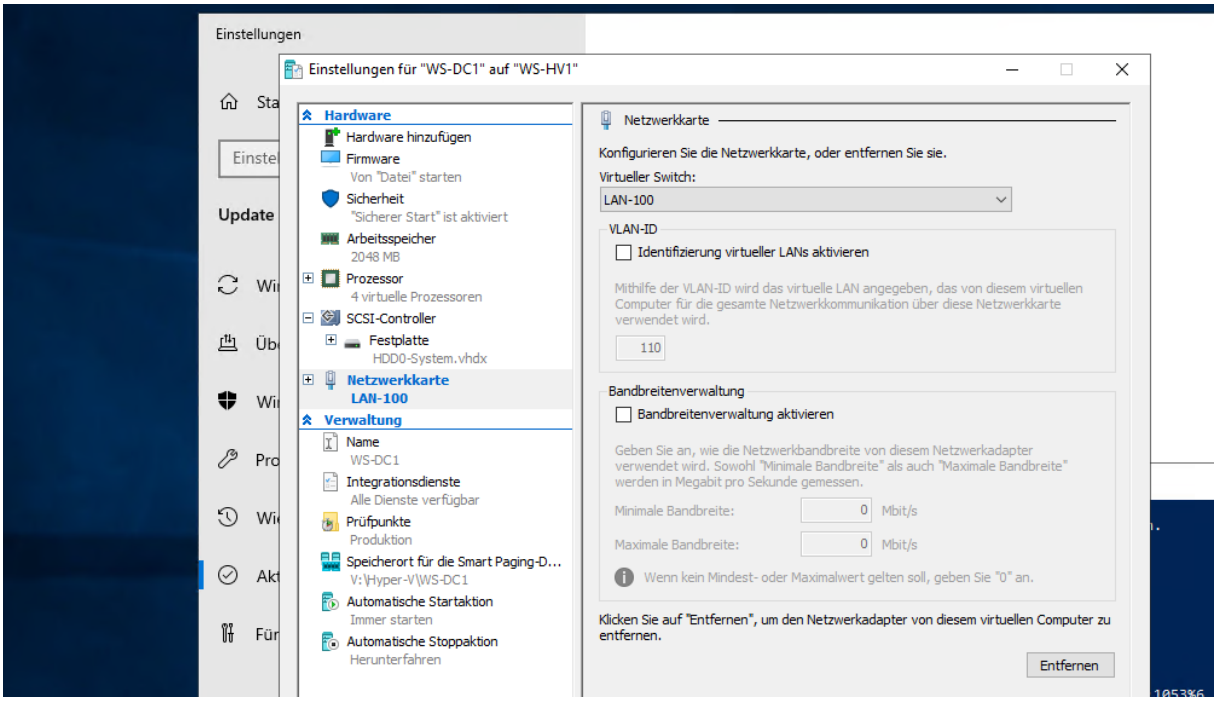


Für die Aktivierung trage ich den Produkt Key ein. Die Online-Aktivierung gelingt aber nur im Client-Netzwerk. Daher patche ich den Server fix um:

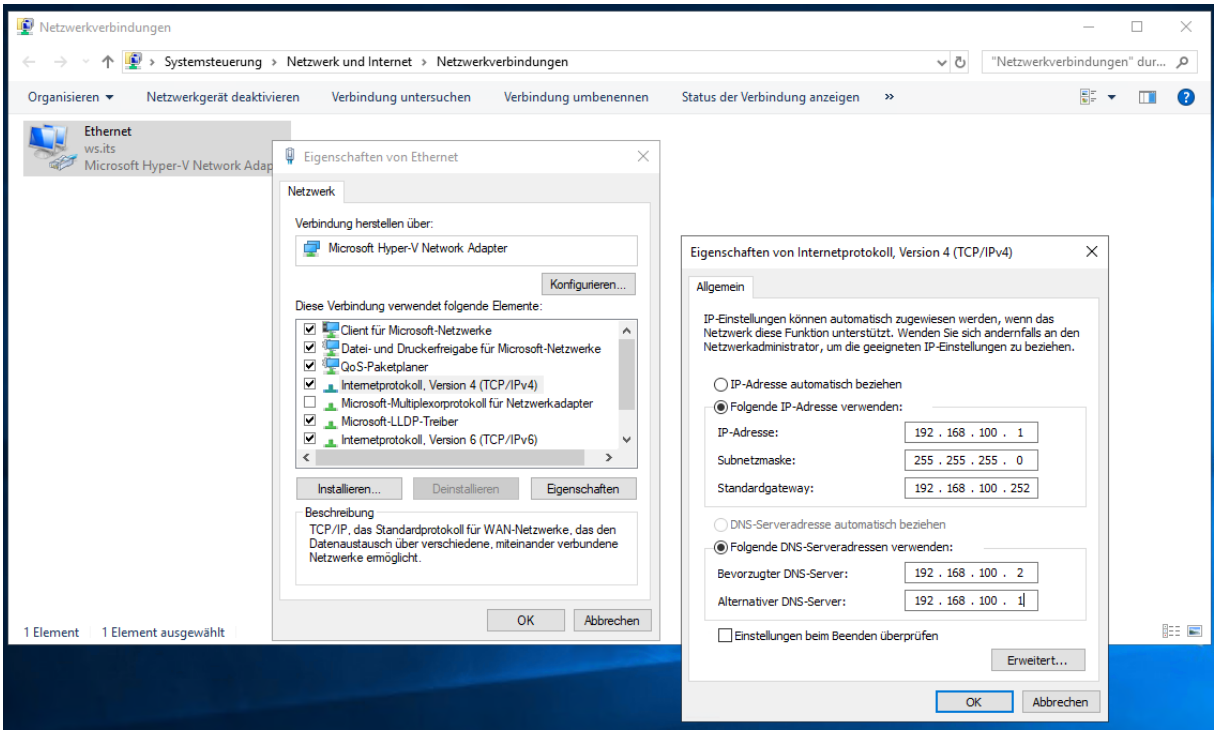




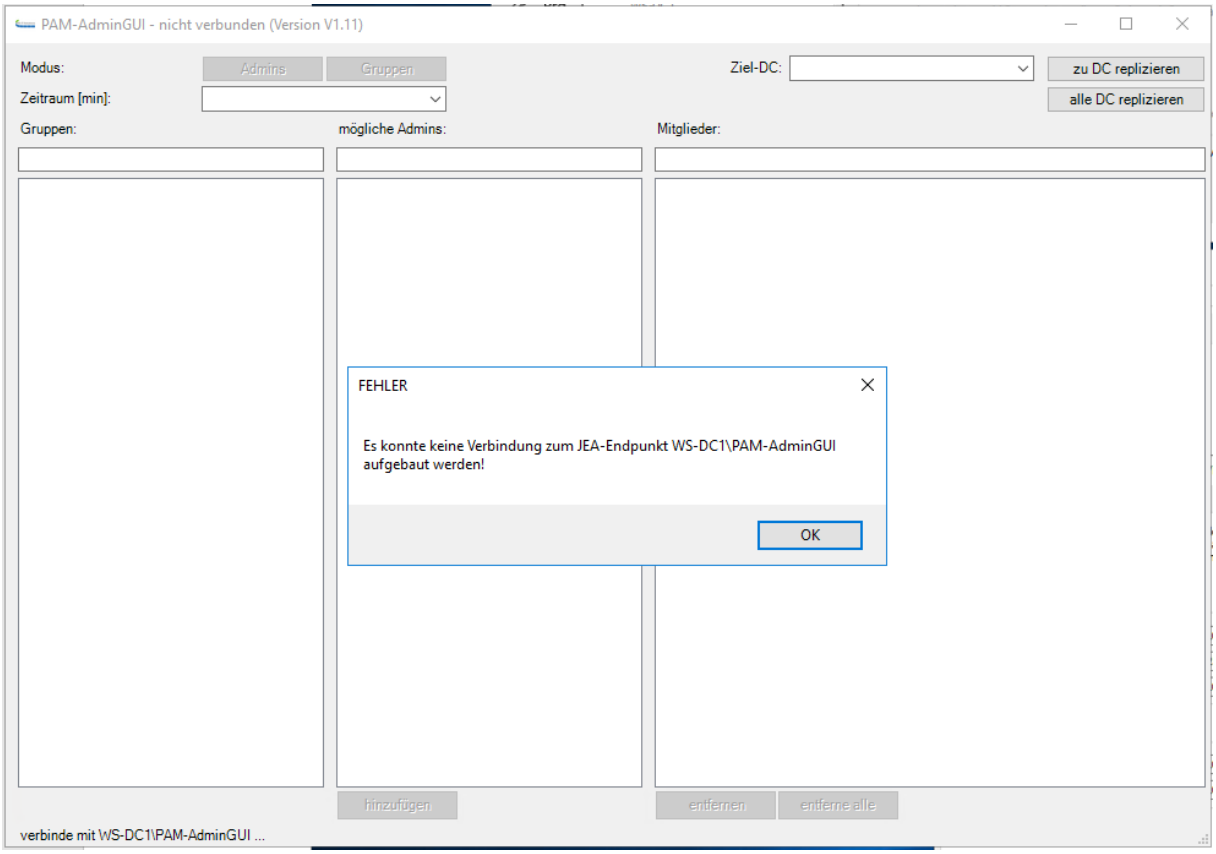
Damit ist das System dauerhaft aktiviert. Er kann zurück in das abgesicherte Servernetzwerk:



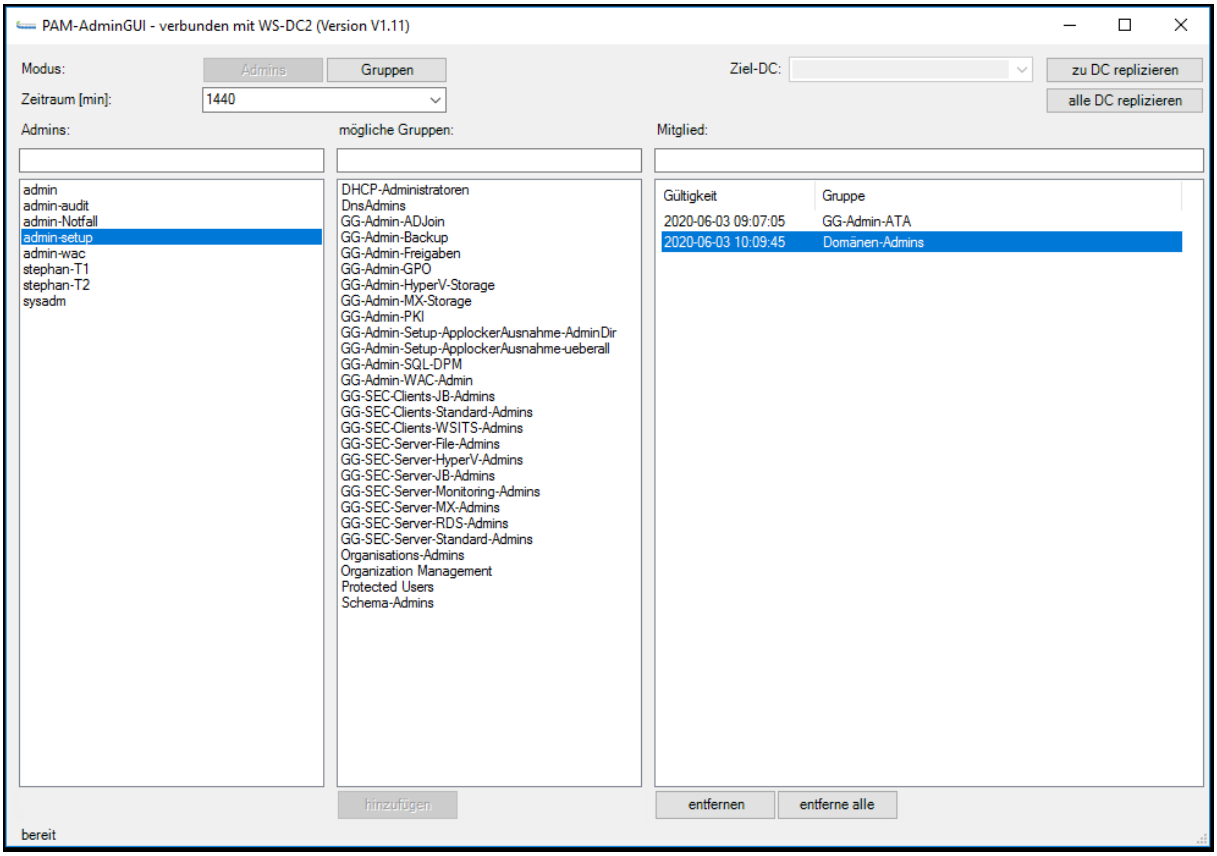
Hier bekommt er jetzt die statische Konfiguration mit der alten IPv4-Adresse:



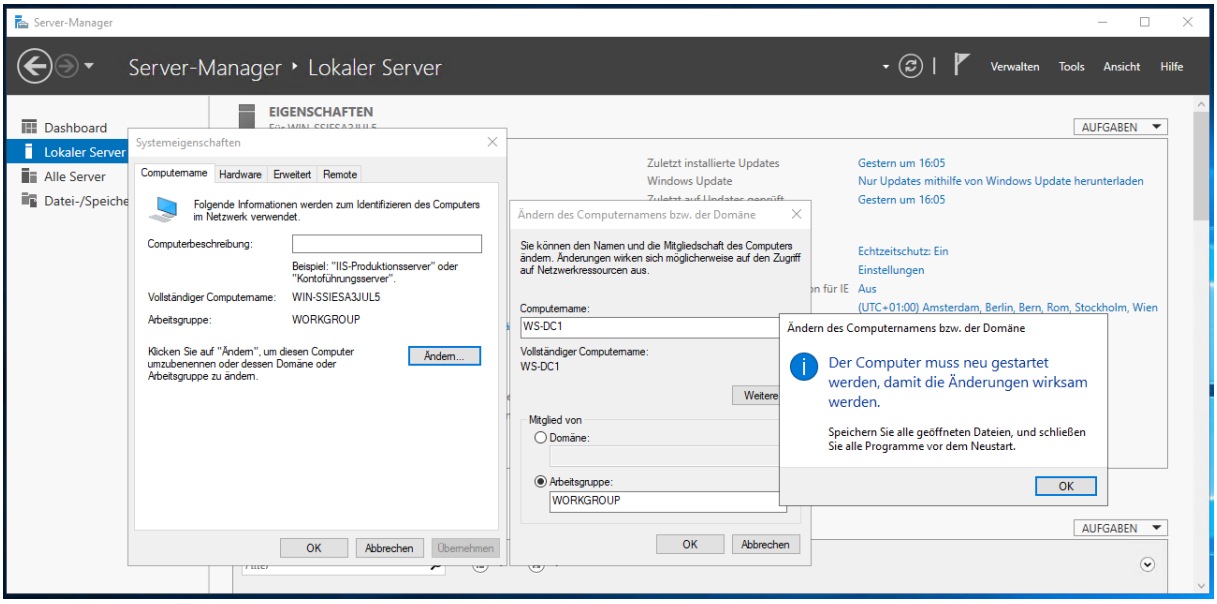
Für den Domain Join muss ich einen NTLM-fähigen AdminAccount in eine spezielle Gruppe aufnehmen. Aber mein PAM-Tool kann keine Verbindung zum Domain Controller WS-DC1 herstellen:



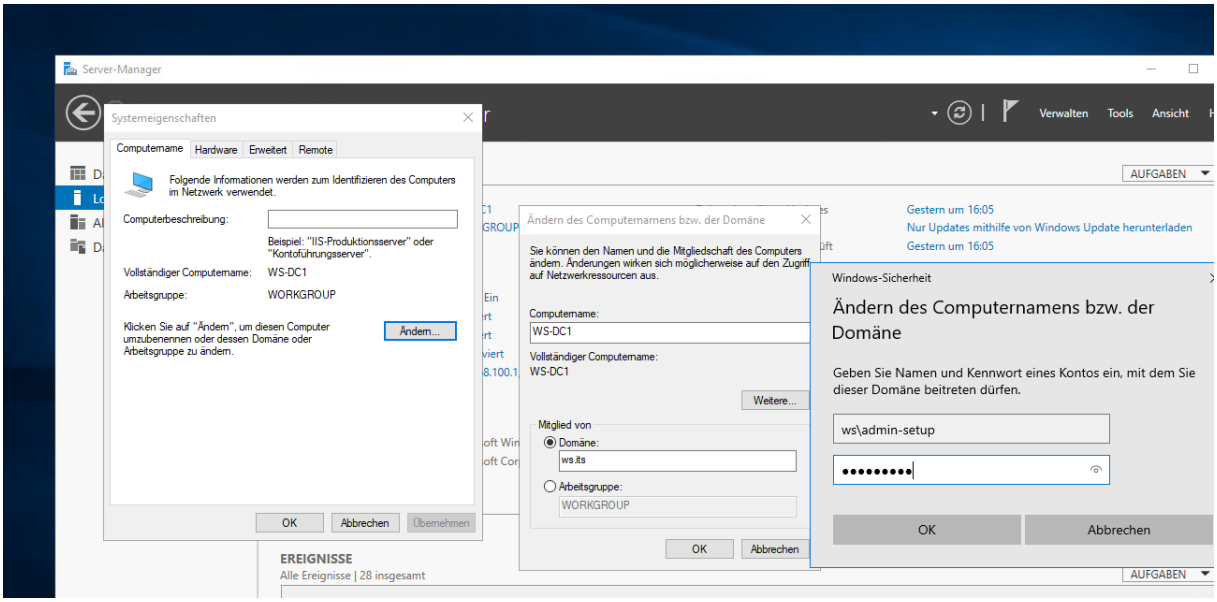
Klar, denn den gibt es aktuell nicht. Aber dafür habe ich in der letzten Version eine hochverfügbare JEA-Konfiguration eingebaut. Das Script verbindet sich einfach mit einem anderen DC:

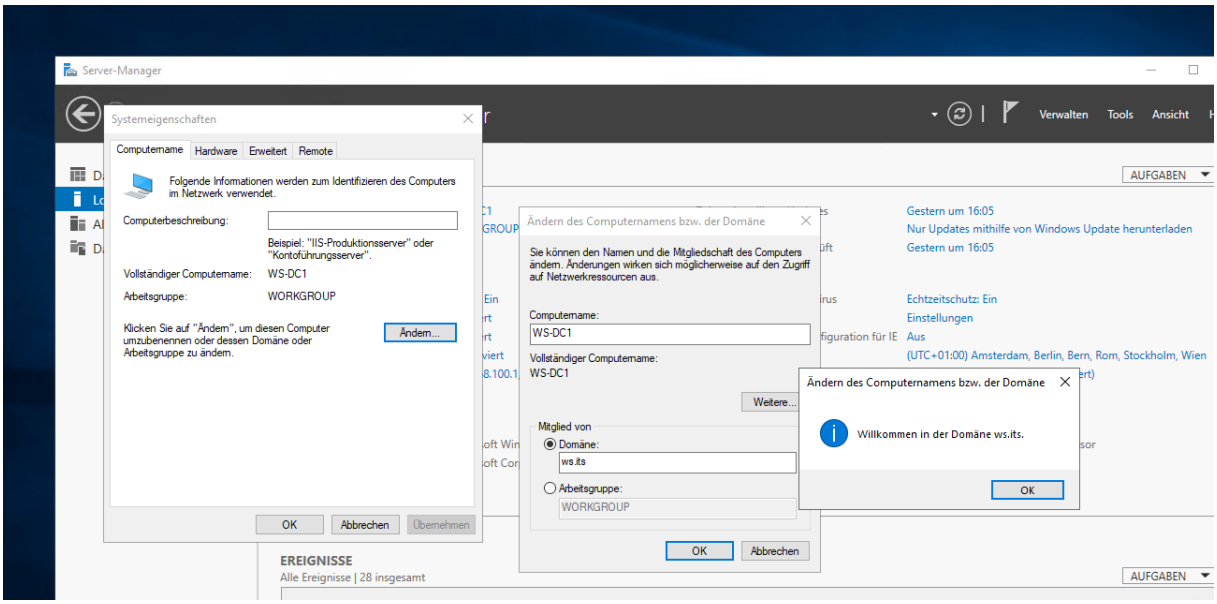


Auf dem neuen Server passe ich den Namen an und starte einmal neu. So kann der Server nachher die Identität des alten DCs übernehmen:

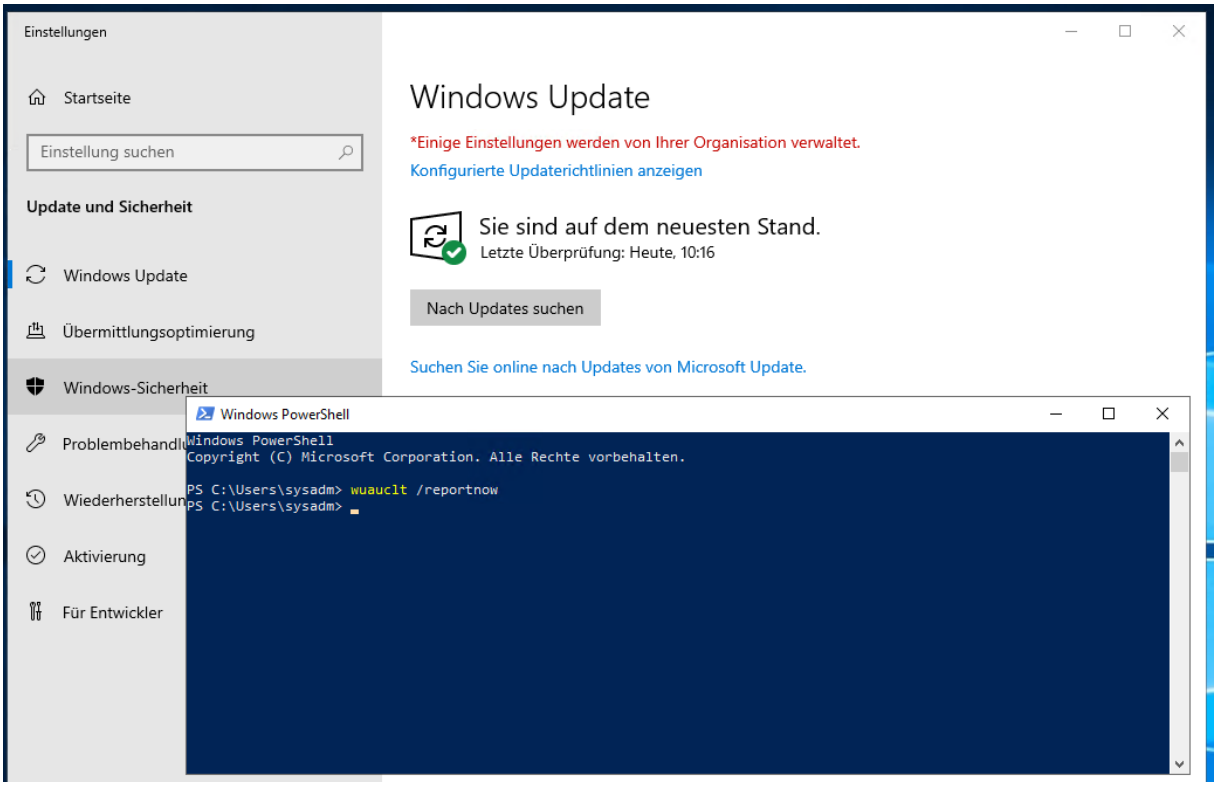


Es folgt der Domain Join:

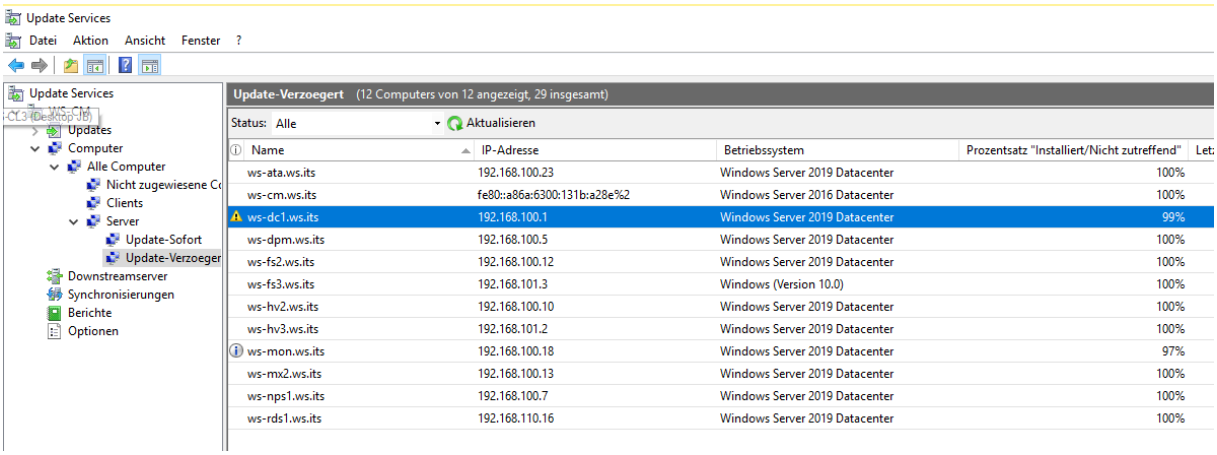




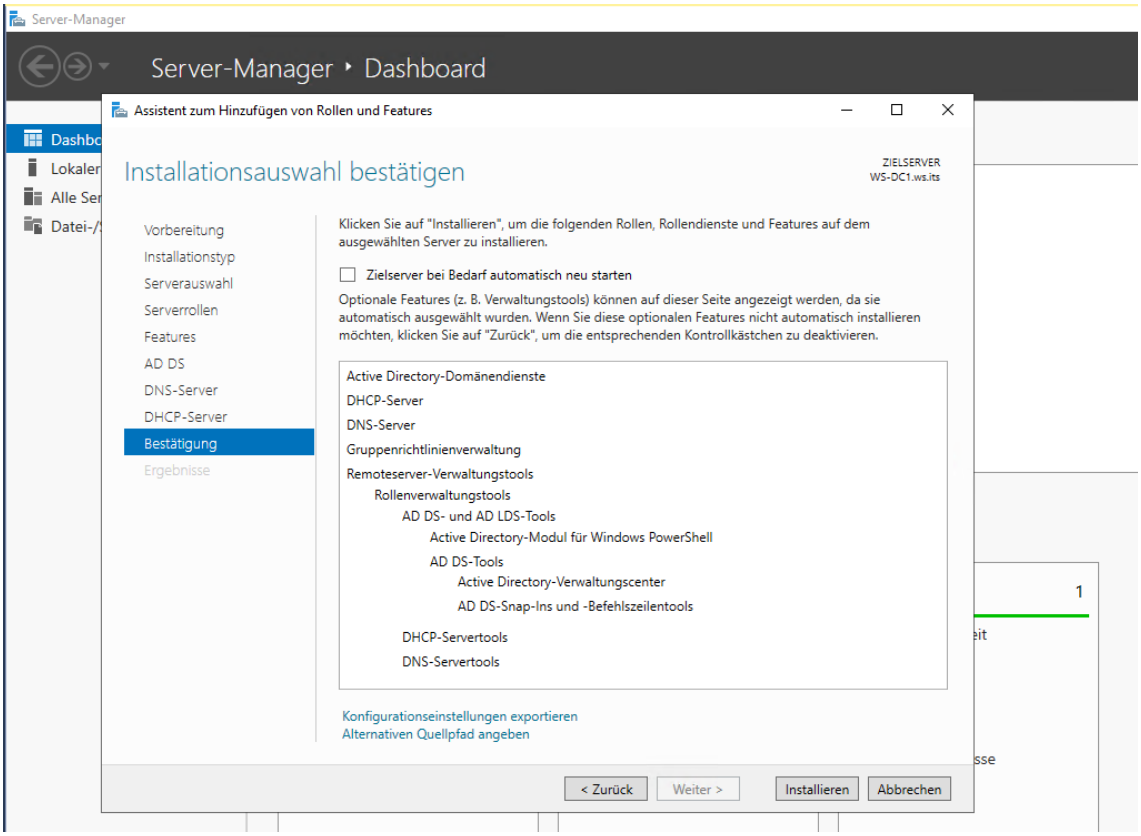
Der Server ist nach dem Neustart als Domain Member aktiv. Normalerweise würde ich jetzt die Windows Updates nachinstallieren. Das Image ist aber erst einen Tag alt. Der Server ist also Up-to-date. Mit wuauclt initialissiere ich den Kontakt zum WSUS-Server:



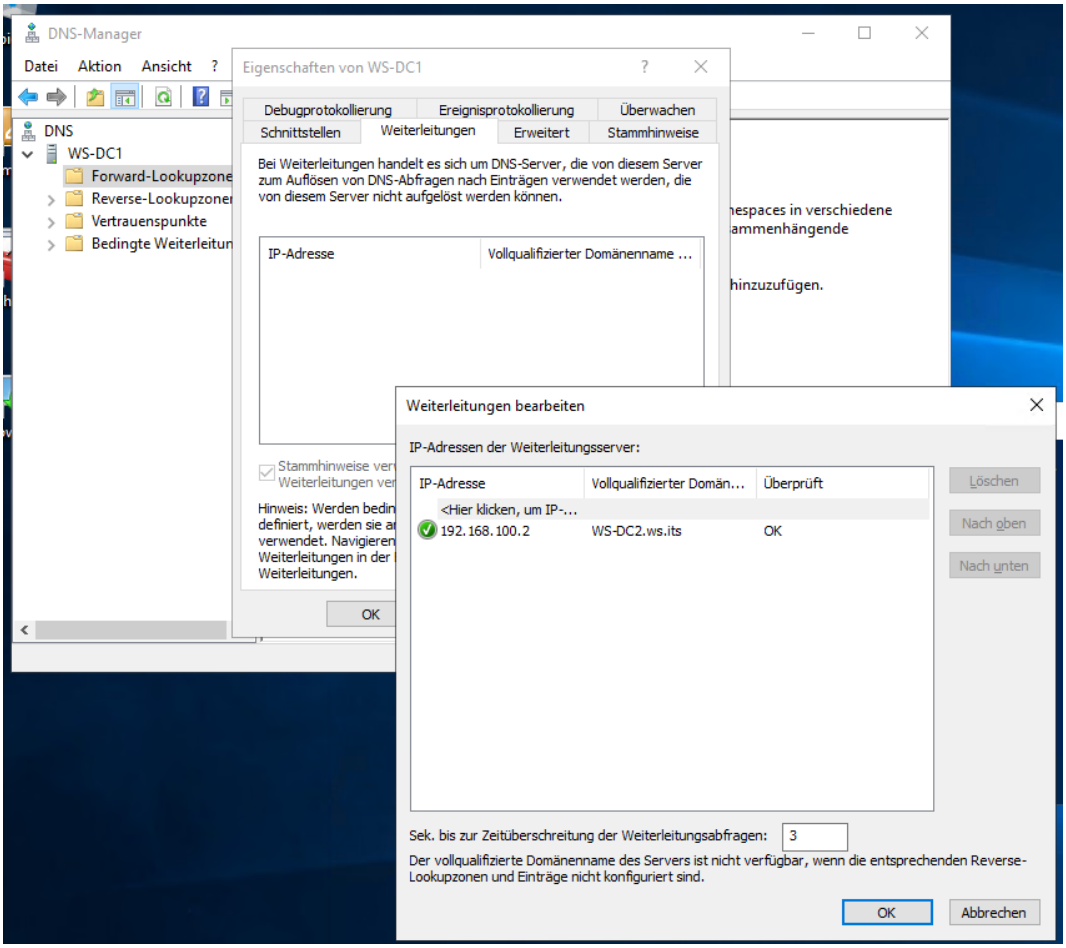
Wuauclt war erfolgreich. Der Server hat sein WSUS-Computerobjekt übernommen:



Ich installiere die Rollen und Features für den neuen Domain Controller. Da sind keine Überraschungen dabei:

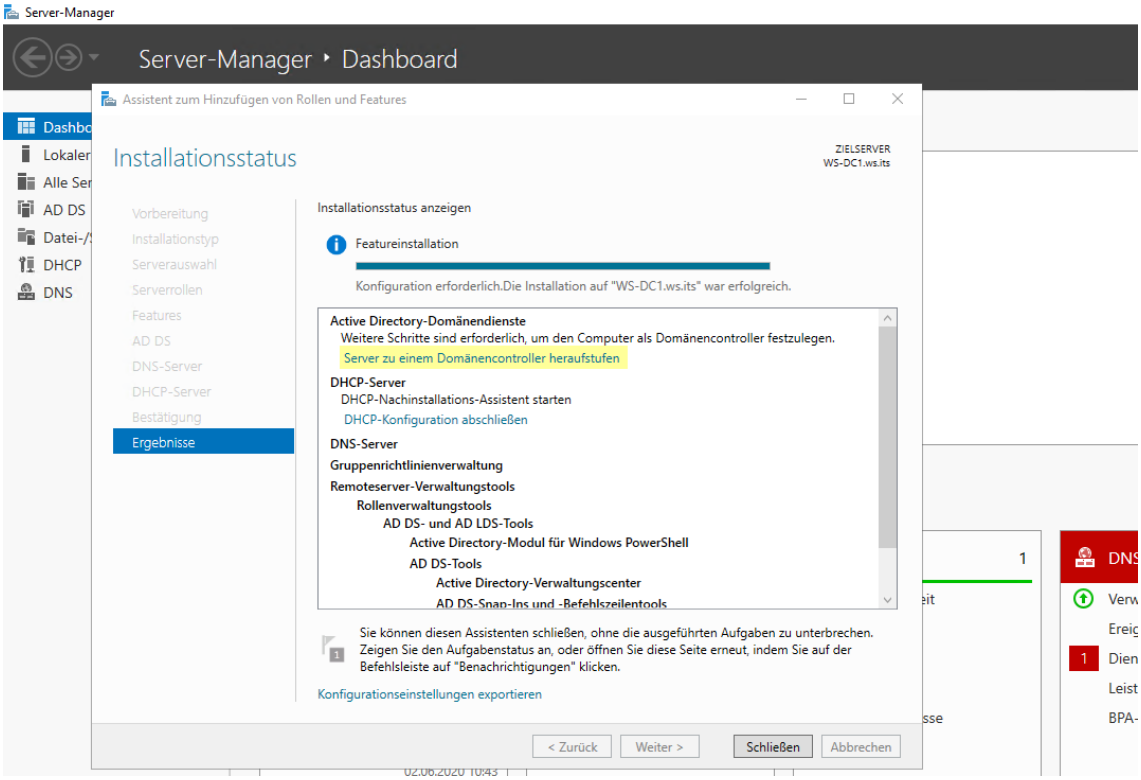


Ab jetzt würde ich wieder in die gleiche DNS-Problematik reinrutschen wie nach der Entfernung des alten DCs. Daher konfiguriere ich jetzt einen DNS-Forwarder: den WS-DC2. Alle Anfragen werden also an den funktionalen DC weitergeleitet. Die Clients können den WS-DC1 dadurch erfolgreich befragen:

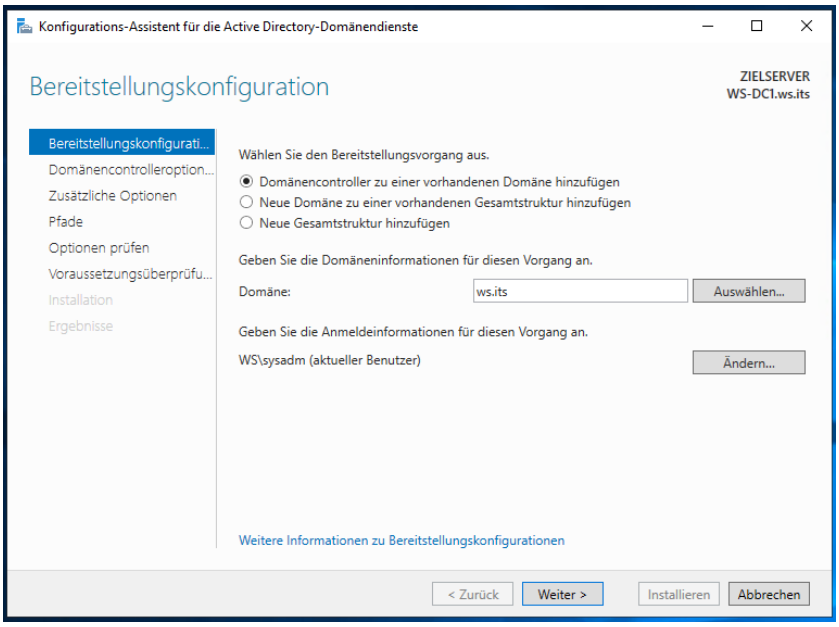


Installation der Rolle Active Directory

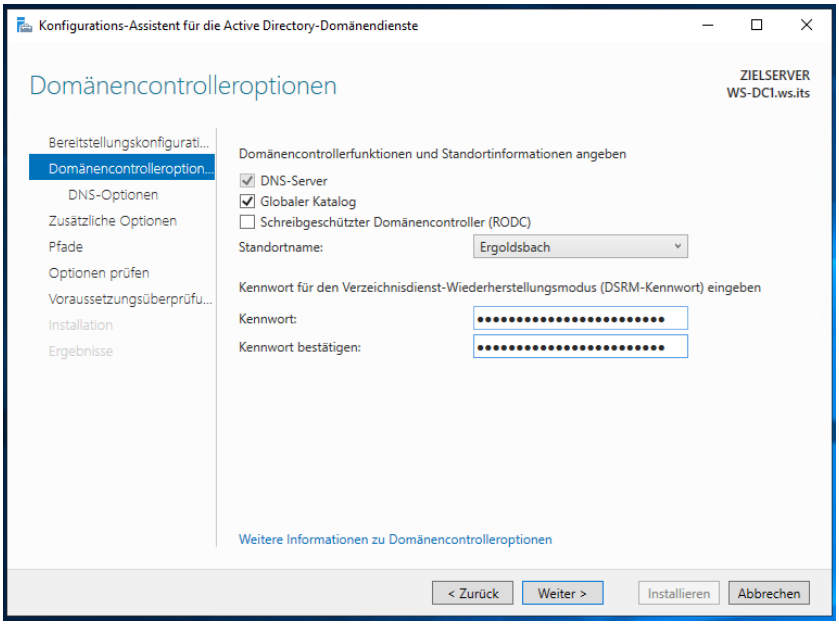
Ich starte die Heraufstufung zu einem Domain Controller im Server Manager:



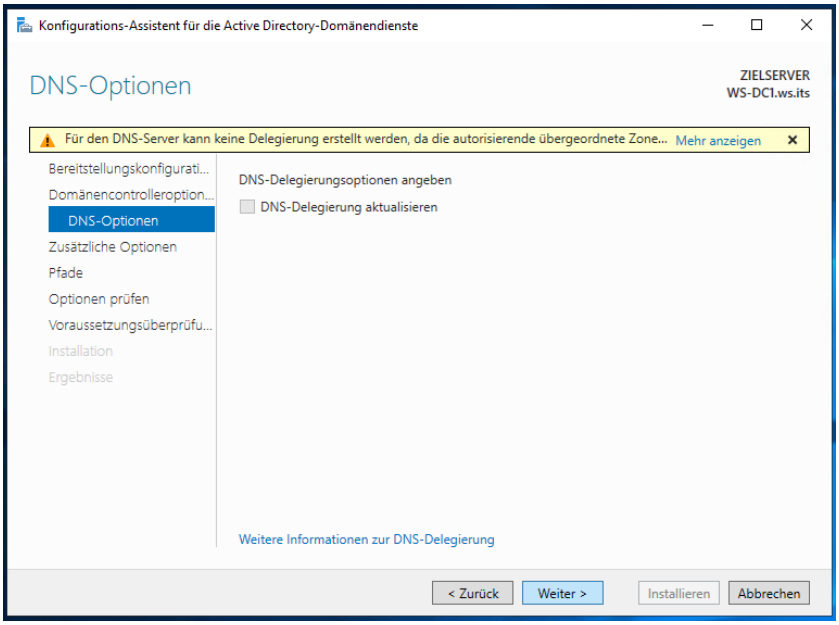
Der Prozess ist recht einfach. Der neue WS-DC1 soll ein weiterer Domain Controller meiner bestehenden Domain werden:



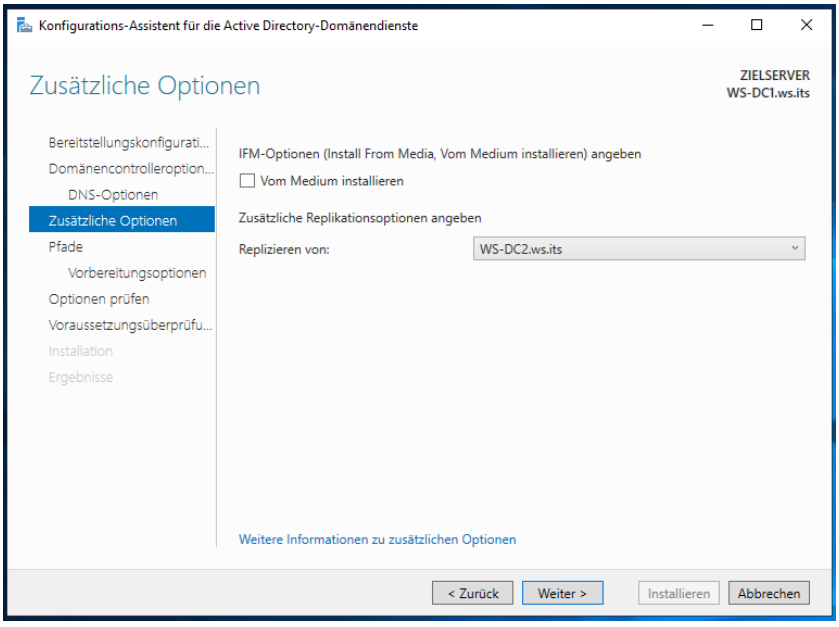
Vom RODC halte ich nicht (mehr) so viel. Das Wiederherstellungspasswort hinterlege ich in meinem Passwortsafe:



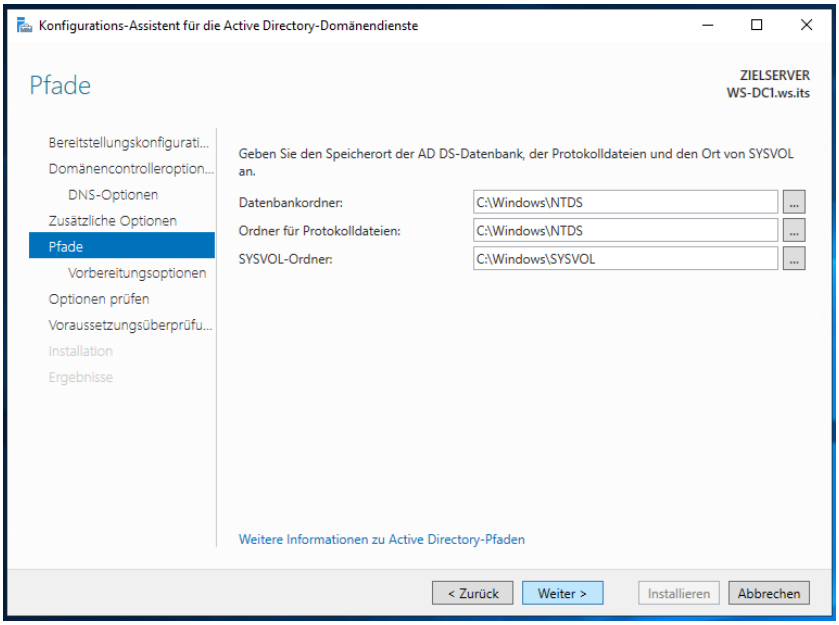
Die Warnmeldung vom DNS kann ignoriert werden. Die DNS-Zone its. gibt es nicht. Also kann ich darin keinen DNS-Server für die Delegierung der Subzone ws.its. bitten:



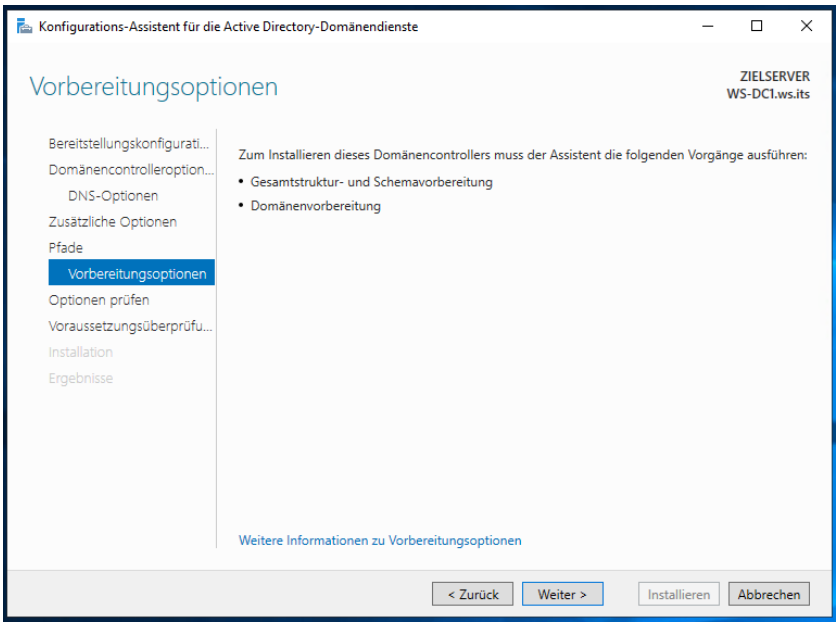
Die Replikation der AD-Objekte starte ich mit meinem WS-DC2 im gleichen Standort:



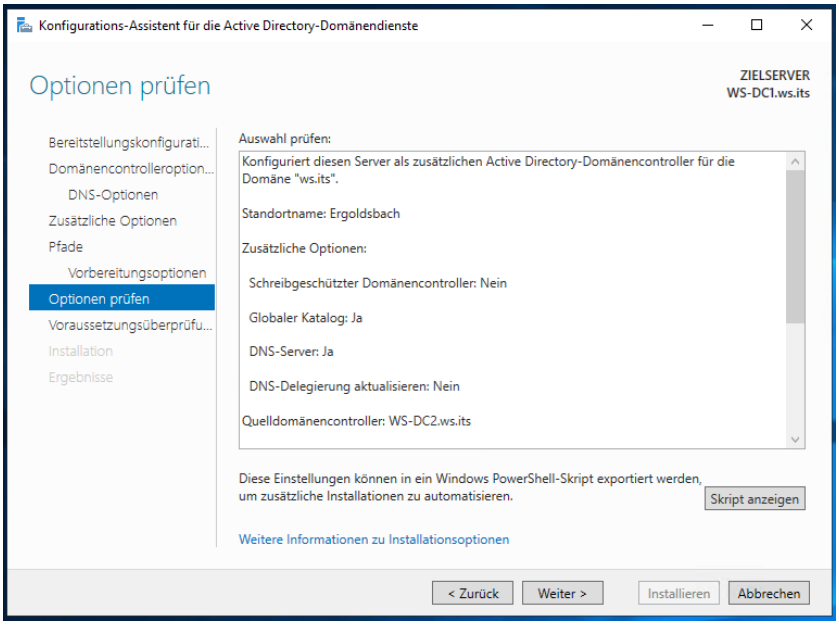
Die Pfade belasse ich im Default:



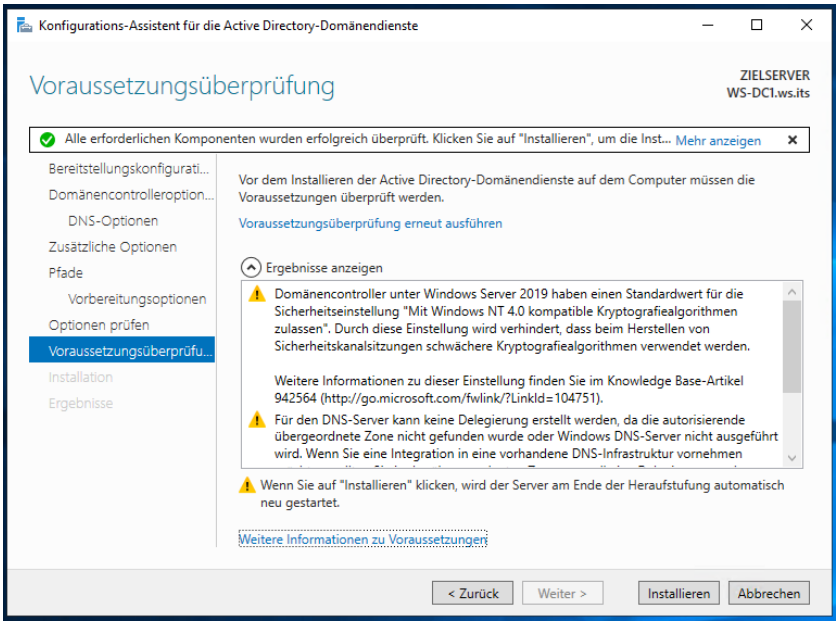
Diesen Schritt kann und sollte man auslagern, wenn man eine große und gewachsene Struktur mit vielen Domain Controllern hat. Windows Server 2019 bringt aber kaum Neuerungen mit. Daher gehe ich bei mir das Risiko ein und starte die Aktualisierung auf einem von 2 Domain Controllern:



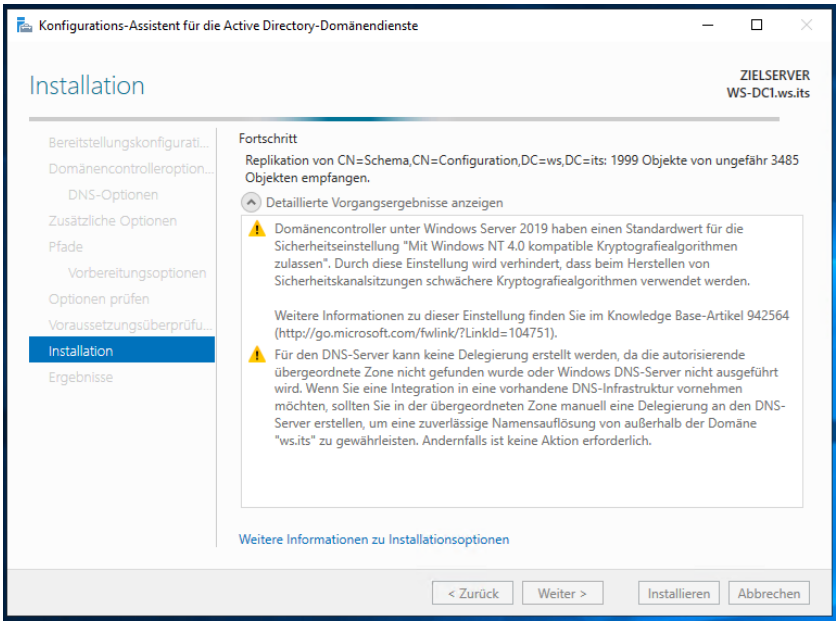
Die Zusammenfassung sieht richtig aus:



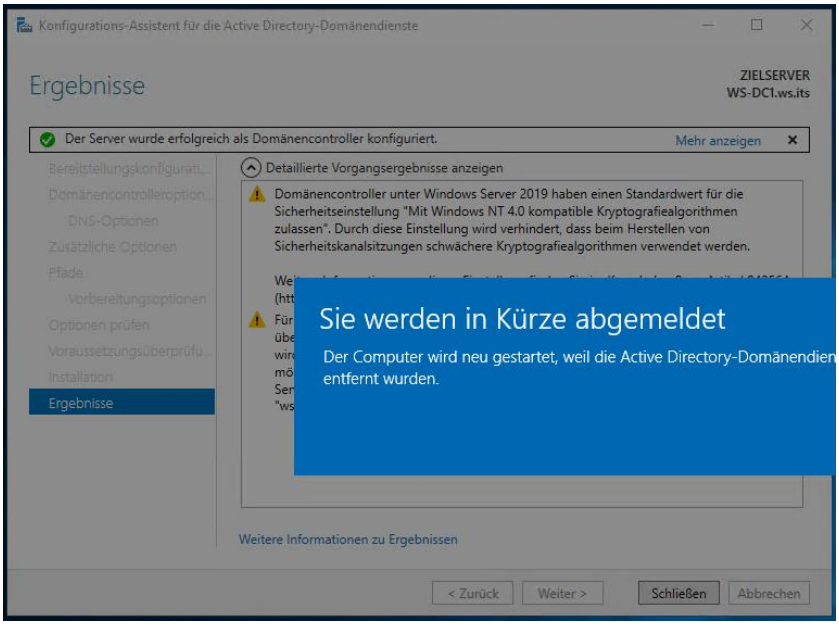
Die Prüfungen sind eher oberflächlich:



Und dann geht es auch schon los:

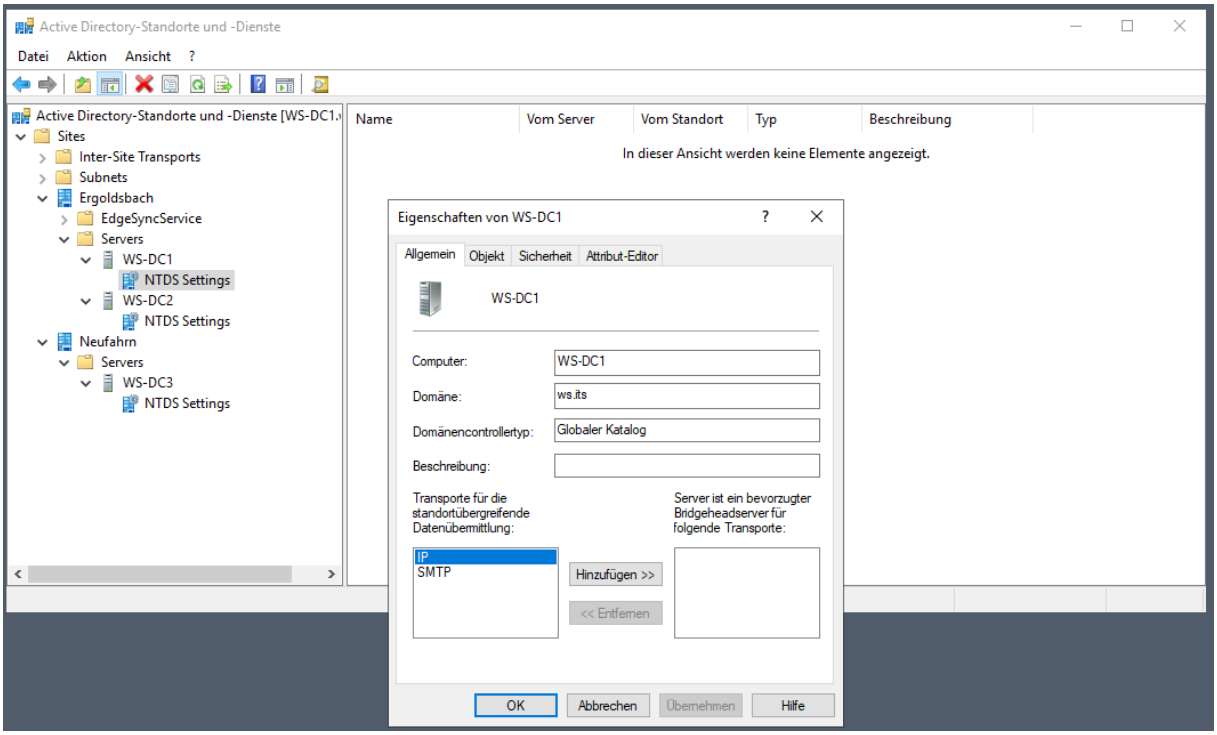


Nach einigen Sekunden startet der Server neu:

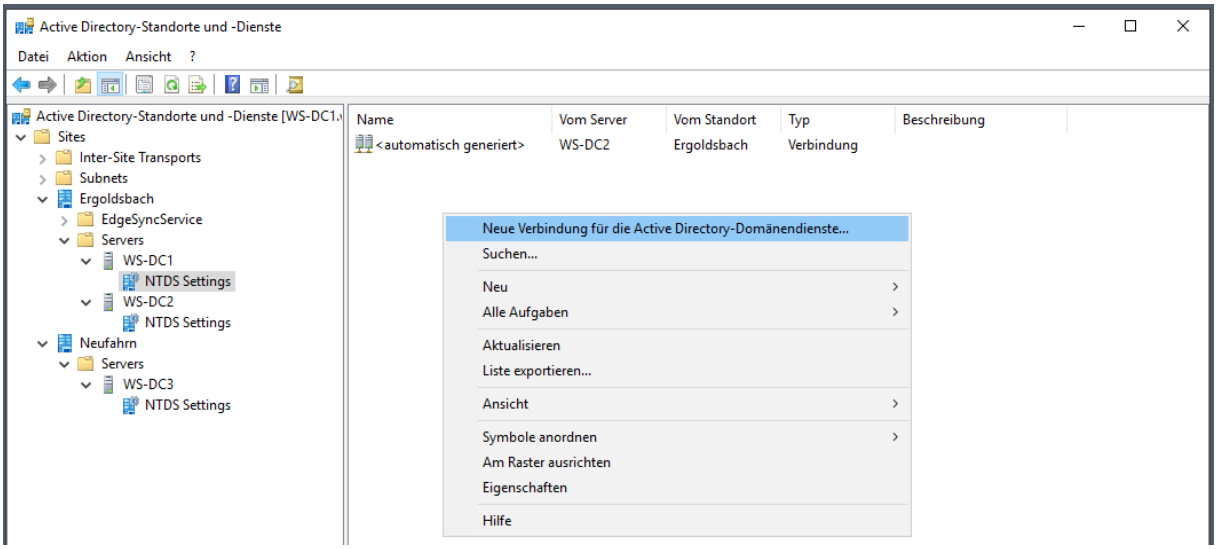


Und mein Domain Controller WS-DC1 ist installiert.

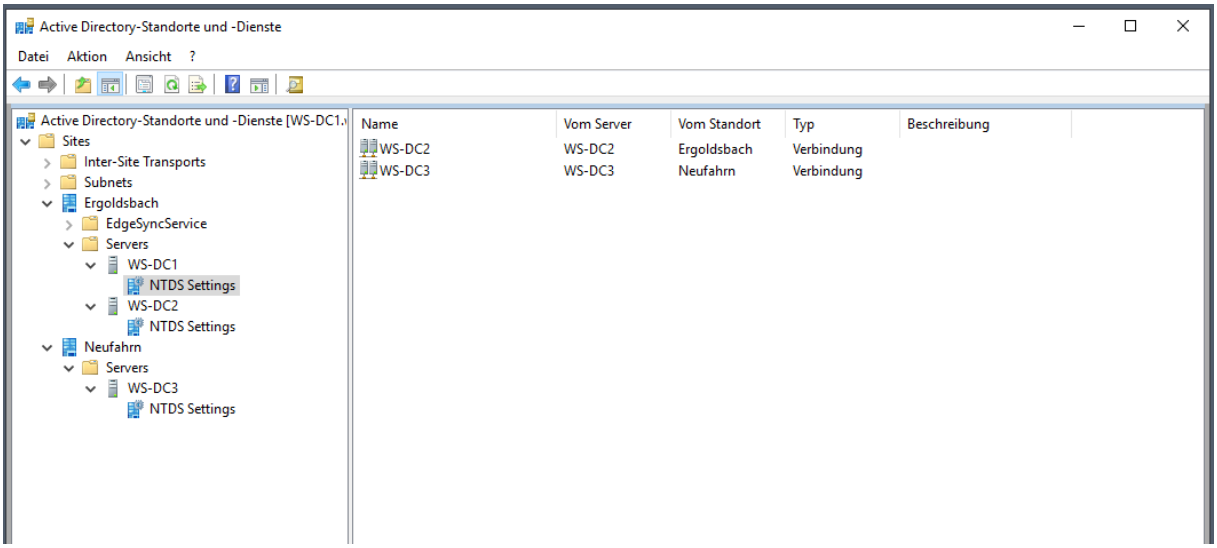
Ich kontrolliere zuerst wieder die AD-Replikation. Der neue Server wird wieder als IP-Bridgehead konfiguriert:



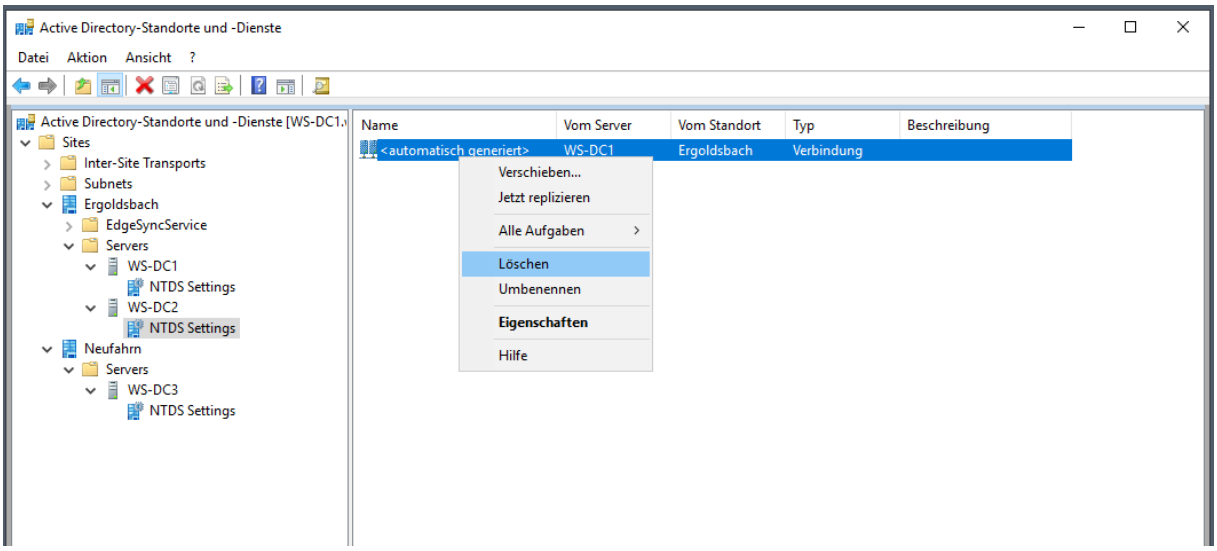
Es wurde bereits eine automatische Replikationsverbindung eingerichtet. Ich erstelle aber lieber meine eigenen:

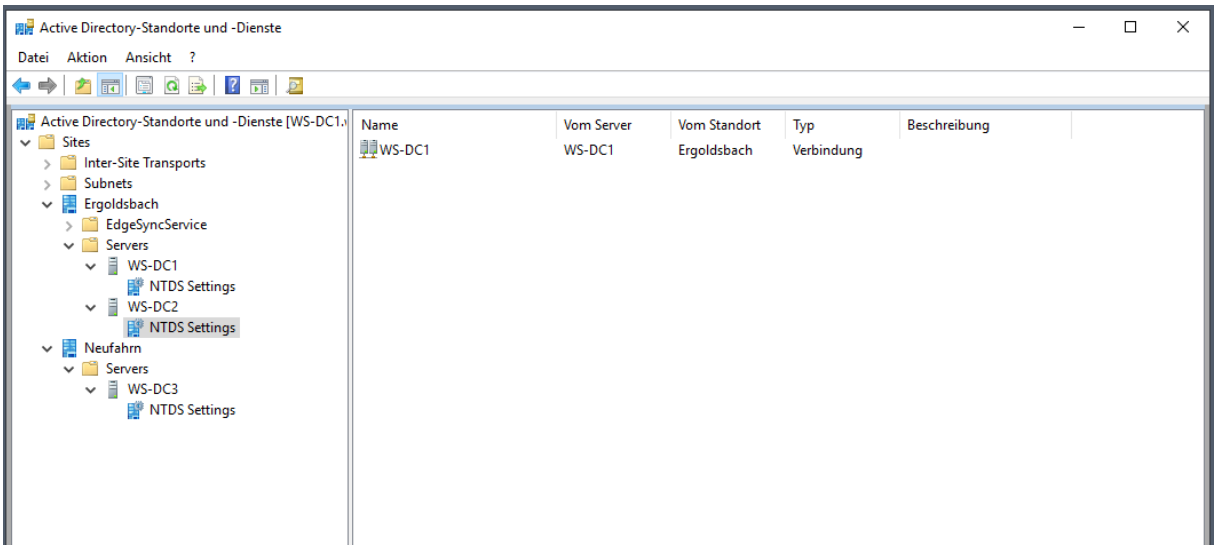


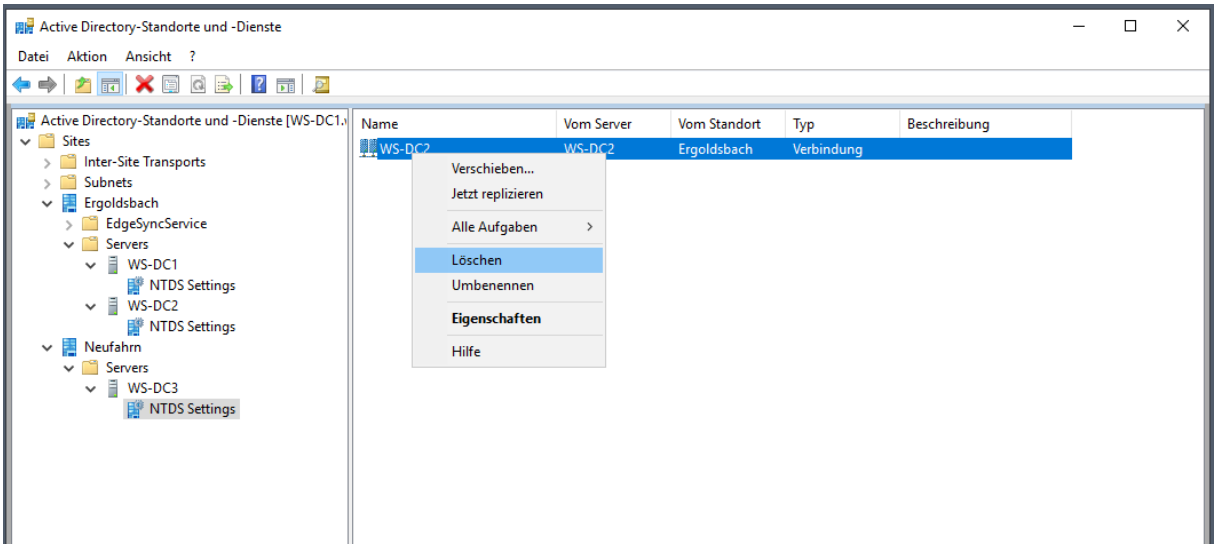
Der Server soll wieder zwischen den beiden anderen Servern vermitteln. Die automatische Verbindung entferne ich:

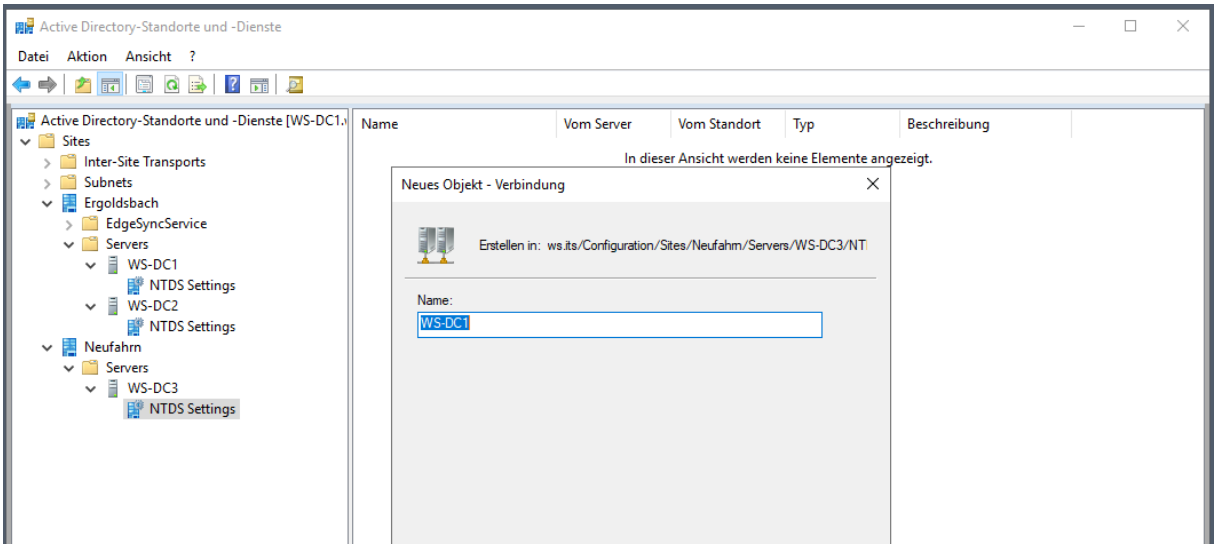


Den Vorgang wiederhole ich für jeden Domain Controller:

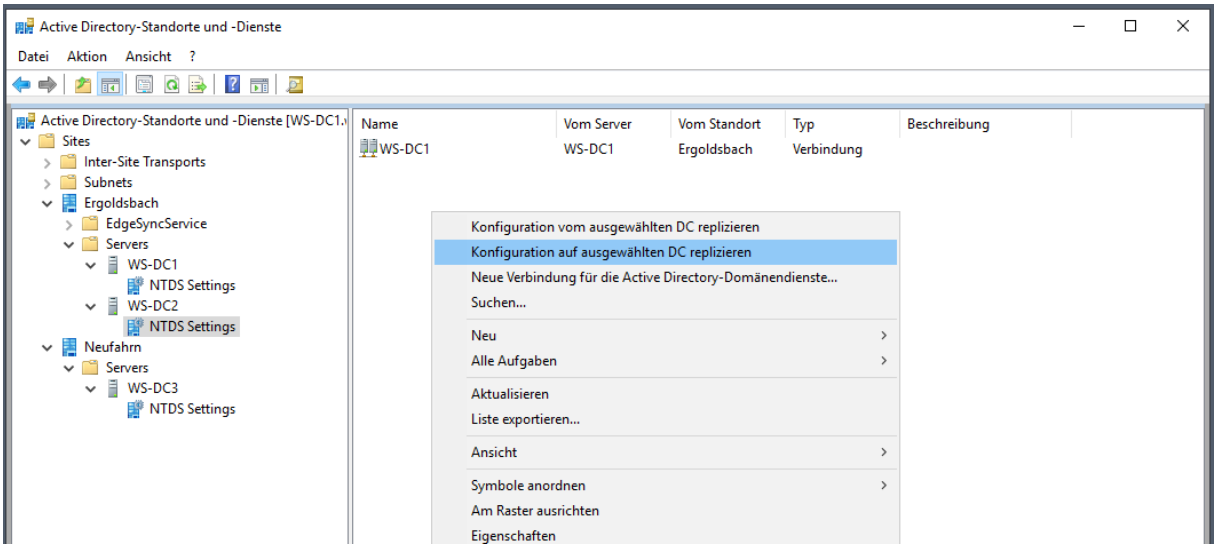


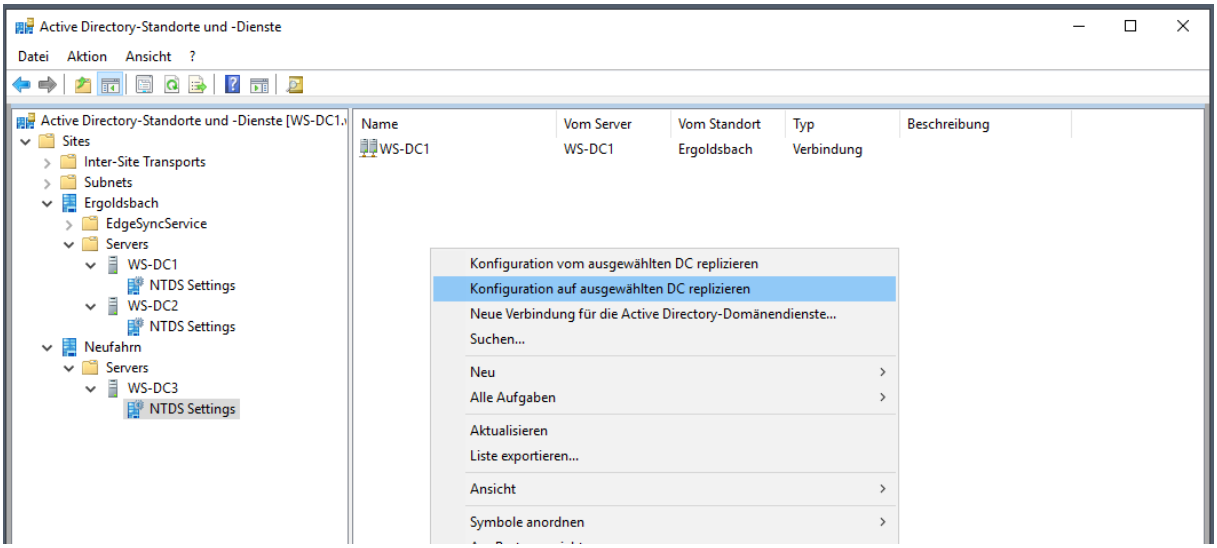




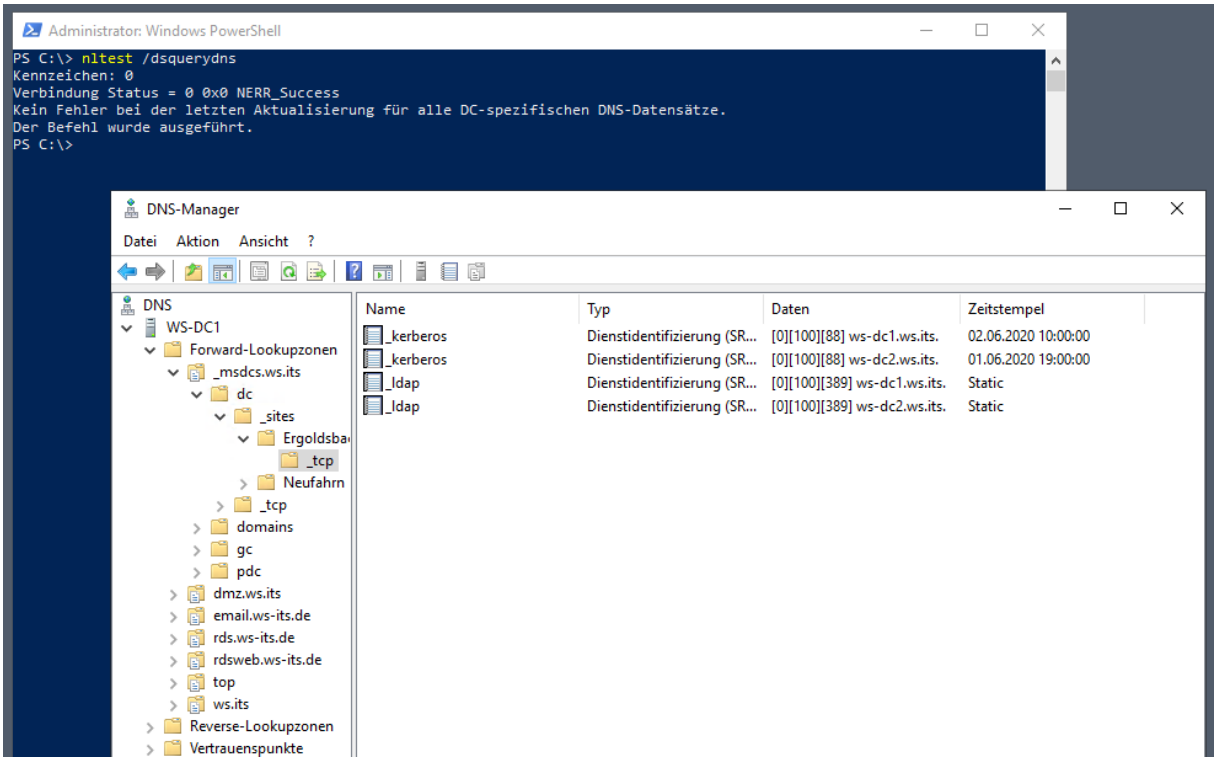


Jetzt kennt der Server WS-DC1 meinen Replikationsplan. Diesen muss ich noch auf die beiden anderen DCs übertragen:



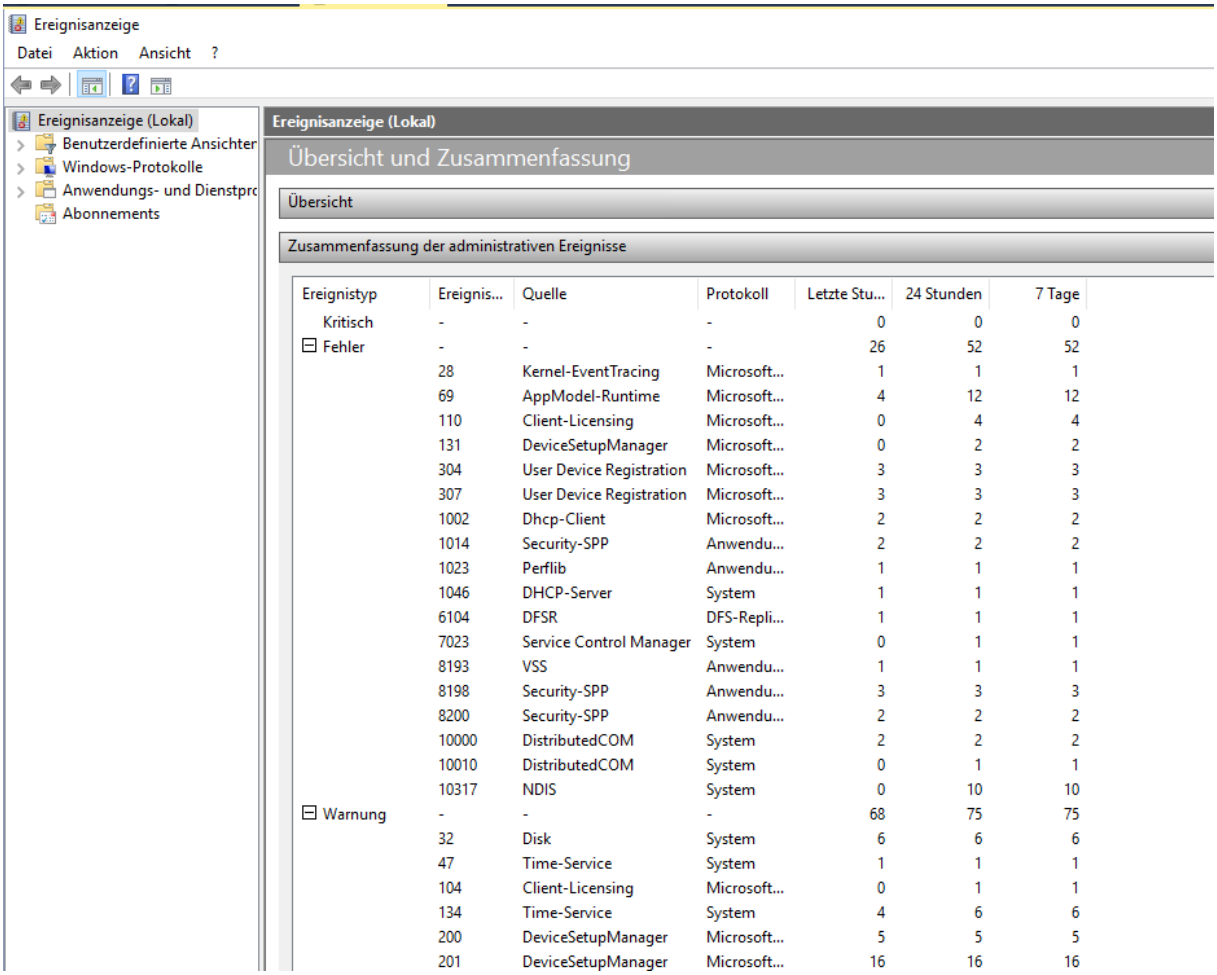


Jetzt prüfe ich, ob sich der neue WS-DC1 korrekt im DNS registriert hat. Das geht mit nltest sehr einfach. Aber auch eine Stichprobe im DNS kann nicht schaden:



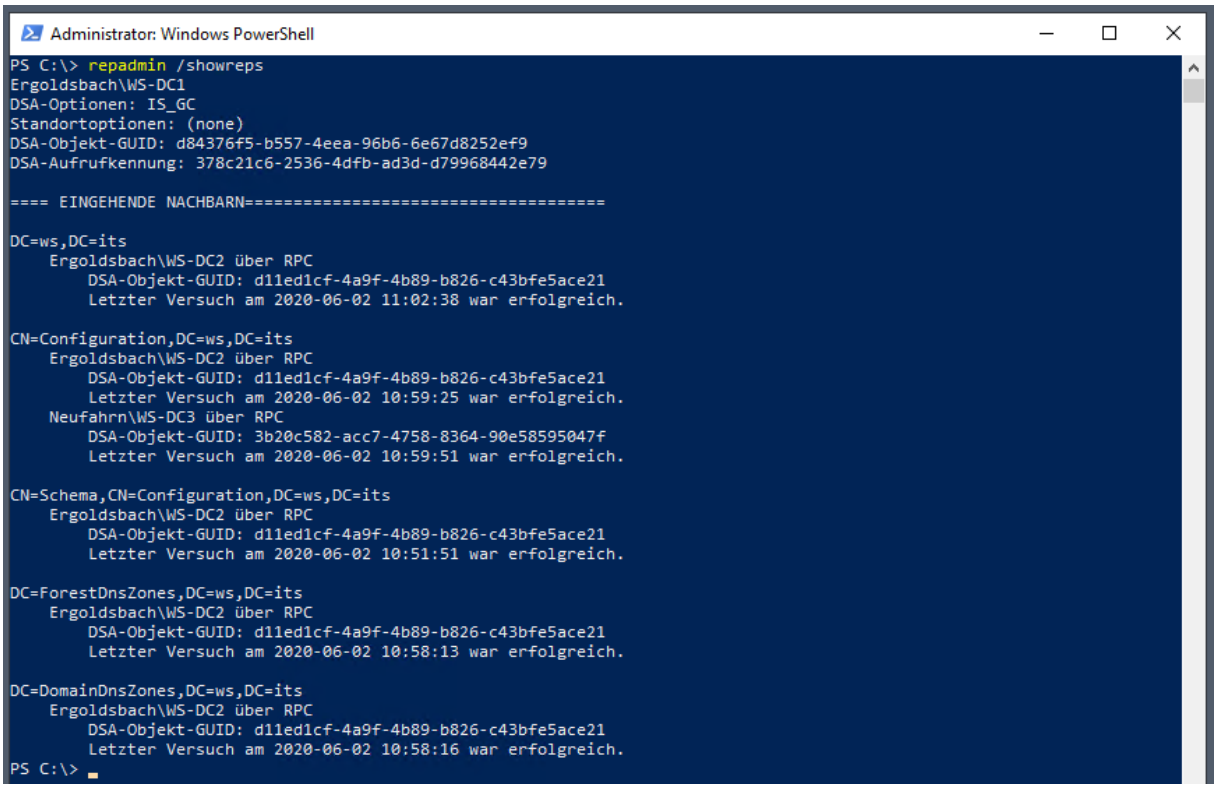
Fein: Der neue DC kann von den Clients gefunden werden.

Spätestens jetzt sollten auch die Eventlogs untersucht werden. Gab es Probleme?



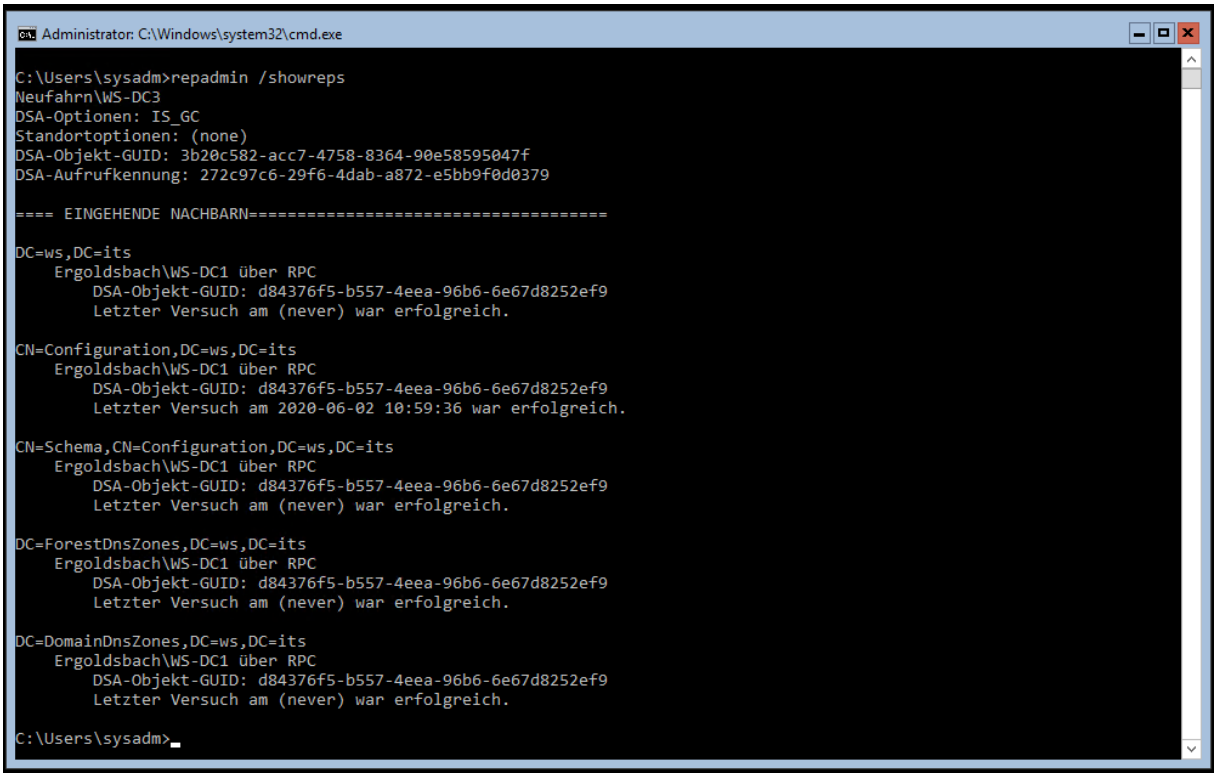
Ereignistyp	Ereignis...	Quelle	Protokoll	Letzte Stu...	24 Stunden	7 Tage
Kritisch	-	-	-	0	0	0
⊟ Fehler	-	-	-	26	52	52
	28	Kernel-EventTracing	Microsoft...	1	1	1
	69	AppModel-Runtime	Microsoft...	4	12	12
	110	Client-Licensing	Microsoft...	0	4	4
	131	DeviceSetupManager	Microsoft...	0	2	2
	304	User Device Registration	Microsoft...	3	3	3
	307	User Device Registration	Microsoft...	3	3	3
	1002	Dhcp-Client	Microsoft...	2	2	2
	1014	Security-SPP	Anwendu...	2	2	2
	1023	Perflib	Anwendu...	1	1	1
	1046	DHCP-Server	System	1	1	1
	6104	DFSR	DFS-Repli...	1	1	1
	7023	Service Control Manager	System	0	1	1
	8193	VSS	Anwendu...	1	1	1
	8198	Security-SPP	Anwendu...	3	3	3
	8200	Security-SPP	Anwendu...	2	2	2
	10000	DistributedCOM	System	2	2	2
	10010	DistributedCOM	System	0	1	1
	10317	NDIS	System	0	10	10
⊟ Warnung	-	-	-	68	75	75
	32	Disk	System	6	6	6
	47	Time-Service	System	1	1	1
	104	Client-Licensing	Microsoft...	0	1	1
	134	Time-Service	System	4	6	6
	200	DeviceSetupManager	Microsoft...	5	5	5
	201	DeviceSetupManager	Microsoft...	16	16	16

Das sieht eigentlich ganz gut aus. Die Replikationsverbindungen sollten inzwischen genug Zeit zur Angleichung haben. Also kontrolliere ich die Ergebnisse:

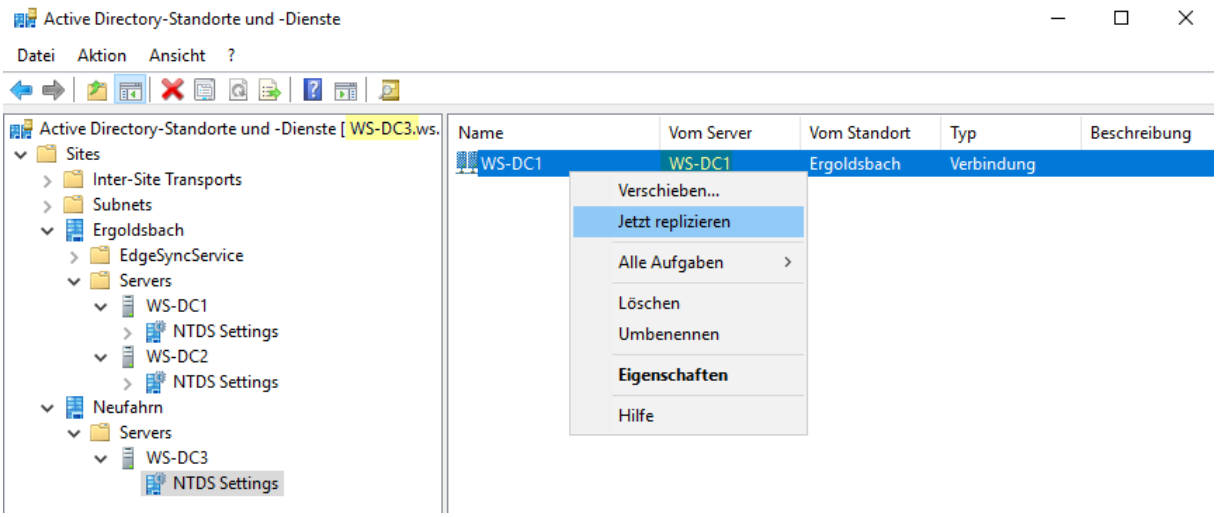


Mmh, da fehlen einige Verbindungen: Eigentlich sollte jede der 2 Partitionen über 2 eingehende Verbindungen verfügen – so wie die Configuration-Partition… WS-DC3 macht irgendwie noch nicht richtig mit.

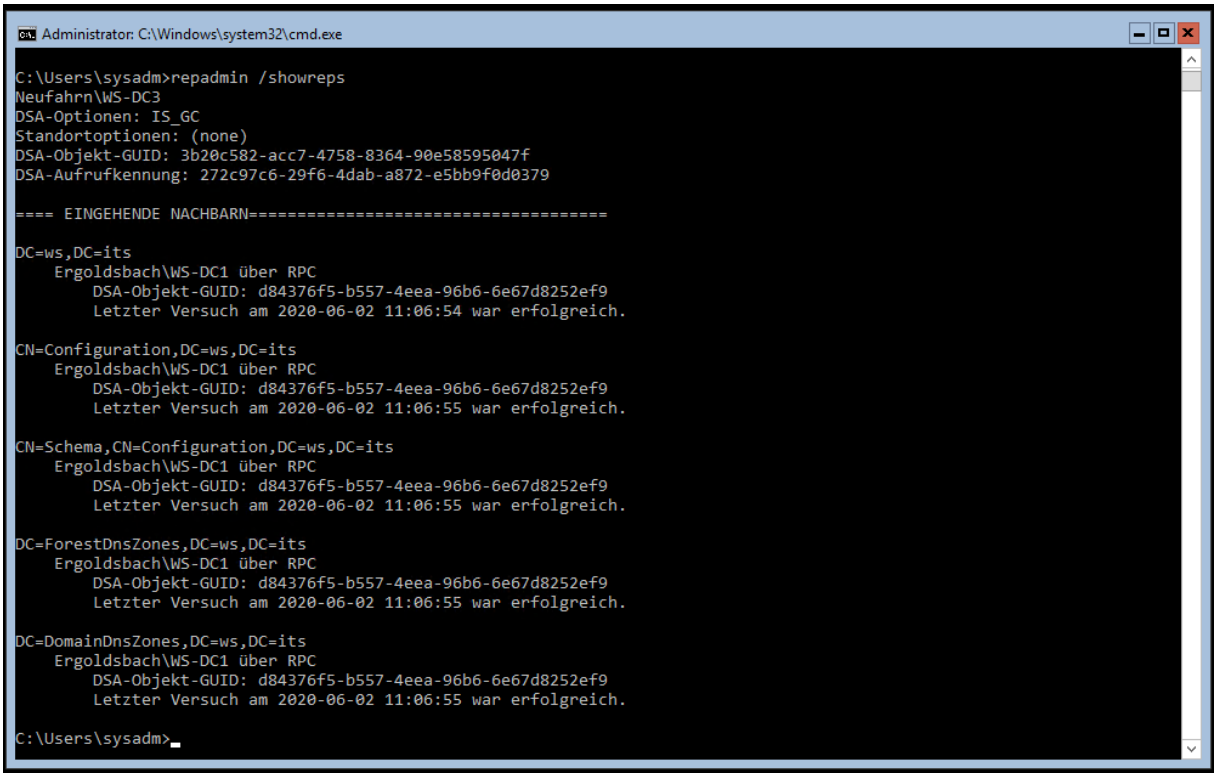
Wichtig ist auch, das repadmin nur eingehende Verbindungen anzeigt. Die Replikation ist aber bidirektional. Also kontrolliere ich auch die beiden anderen DCs. Das hier ist das Ergebnis vom WS-DC3 in Neufahrn:



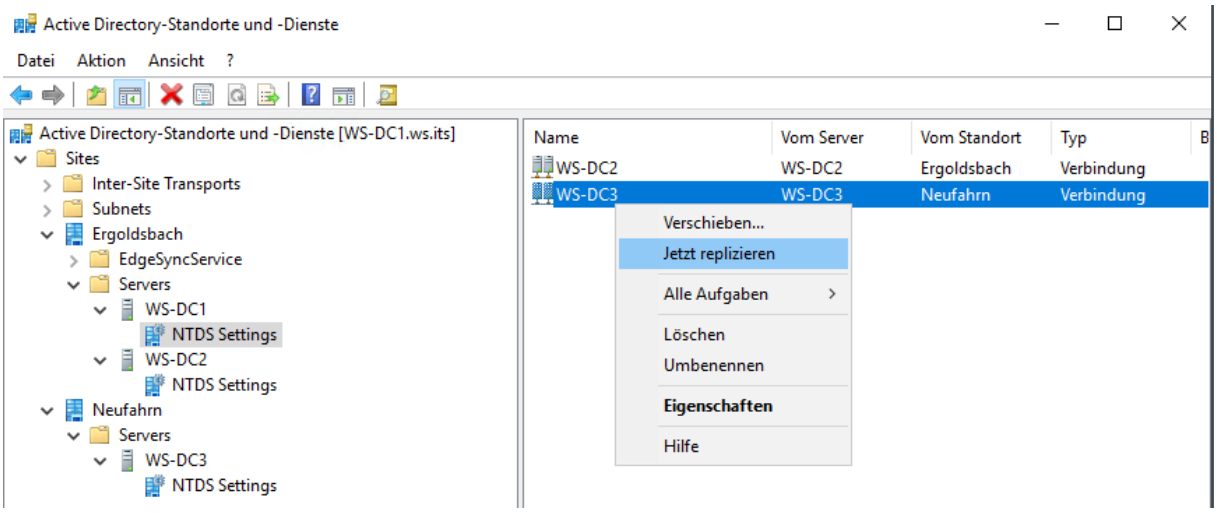
Die Verbindungen waren noch nie erfolgreich. Das kann kurz nach der Einigung auf eine Replikationstopologie durchaus so sein. Aber ich helfe mal nach:



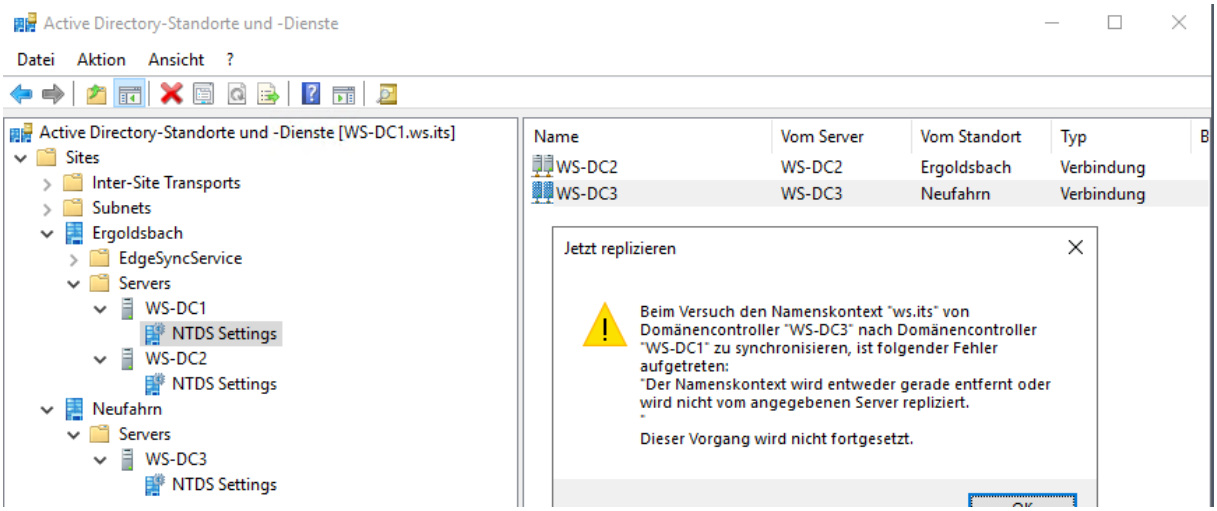
Nach wenigen Sekunden ist der WS-DC3 versorgt:



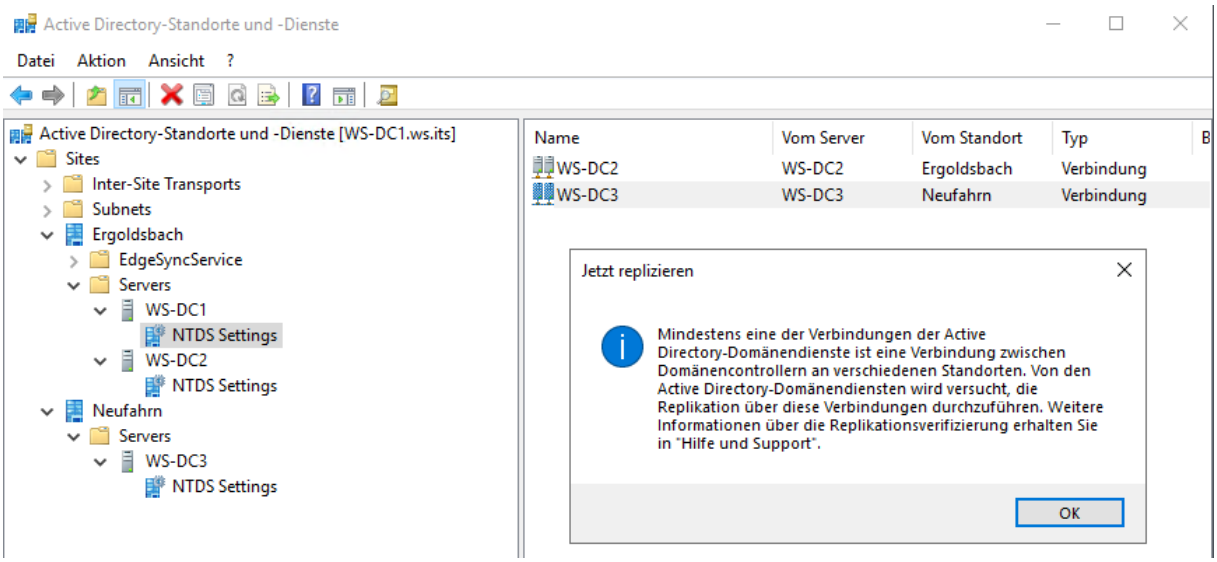
Nun ist noch der Rückweg dran:



Der Server WS-DC1 hatte aber noch einige Lücken im repadmin. Die machen sich hier bemerkbar. Er weigert sich, weil die Replikationsverbindung noch nicht vollständig eingetragen wurde:



Das machen die Domain Controller alleine aus. Da hilft also ein weiterer Kaffee. Na also, es geht doch:



```
Administrator: Windows PowerShell
PS C:\> repadmin /showreps
Ergoldsbach\WS-DC1
DSA-Optionen: IS_GC
Standortoptionen: (none)
DSA-Objekt-GUID: d84376f5-b557-4eea-96b6-6e67d8252ef9
DSA-Aufrufkennung: 378c21c6-2536-4dfb-ad3d-d79968442e79

==== EINGEHENDE NACHBARN====================================

DC=ws,DC=its
    Neufahrn\WS-DC3 über RPC
        DSA-Objekt-GUID: 3b20c582-acc7-4758-8364-90e58595047f
        Letzter Versuch am 2020-06-02 11:12:58 war erfolgreich.
    Ergoldsbach\WS-DC2 über RPC
        DSA-Objekt-GUID: d11ed1cf-4a9f-4b89-b826-c43bfe5ace21
        Letzter Versuch am 2020-06-02 11:13:09 war erfolgreich.

CN=Configuration,DC=ws,DC=its
    Ergoldsbach\WS-DC2 über RPC
        DSA-Objekt-GUID: d11ed1cf-4a9f-4b89-b826-c43bfe5ace21
        Letzter Versuch am 2020-06-02 11:11:50 war erfolgreich.
    Neufahrn\WS-DC3 über RPC
        DSA-Objekt-GUID: 3b20c582-acc7-4758-8364-90e58595047f
        Letzter Versuch am 2020-06-02 11:12:58 war erfolgreich.

CN=Schema,CN=Configuration,DC=ws,DC=its
    Ergoldsbach\WS-DC2 über RPC
        DSA-Objekt-GUID: d11ed1cf-4a9f-4b89-b826-c43bfe5ace21
        Letzter Versuch am 2020-06-02 11:09:30 war erfolgreich.
    Neufahrn\WS-DC3 über RPC
        DSA-Objekt-GUID: 3b20c582-acc7-4758-8364-90e58595047f
        Letzter Versuch am 2020-06-02 11:12:58 war erfolgreich.

DC=ForestDnsZones,DC=ws,DC=its
    Ergoldsbach\WS-DC2 über RPC
        DSA-Objekt-GUID: d11ed1cf-4a9f-4b89-b826-c43bfe5ace21
        Letzter Versuch am 2020-06-02 11:09:30 war erfolgreich.
    Neufahrn\WS-DC3 über RPC
        DSA-Objekt-GUID: 3b20c582-acc7-4758-8364-90e58595047f
        Letzter Versuch am 2020-06-02 11:12:58 war erfolgreich.

DC=DomainDnsZones,DC=ws,DC=its
    Ergoldsbach\WS-DC2 über RPC
        DSA-Objekt-GUID: d11ed1cf-4a9f-4b89-b826-c43bfe5ace21
        Letzter Versuch am 2020-06-02 11:12:42 war erfolgreich.
    Neufahrn\WS-DC3 über RPC
        DSA-Objekt-GUID: 3b20c582-acc7-4758-8364-90e58595047f
        Letzter Versuch am 2020-06-02 11:12:59 war erfolgreich.
PS C:\>
```

Die AD-Replikation ist vollständig aktiv und arbeitet nach dem vorgesehenen Plan.

Weiter geht es mit den FSMO-Rollen. Die verschiebe ich wieder auf den WS-DC1 zurück:

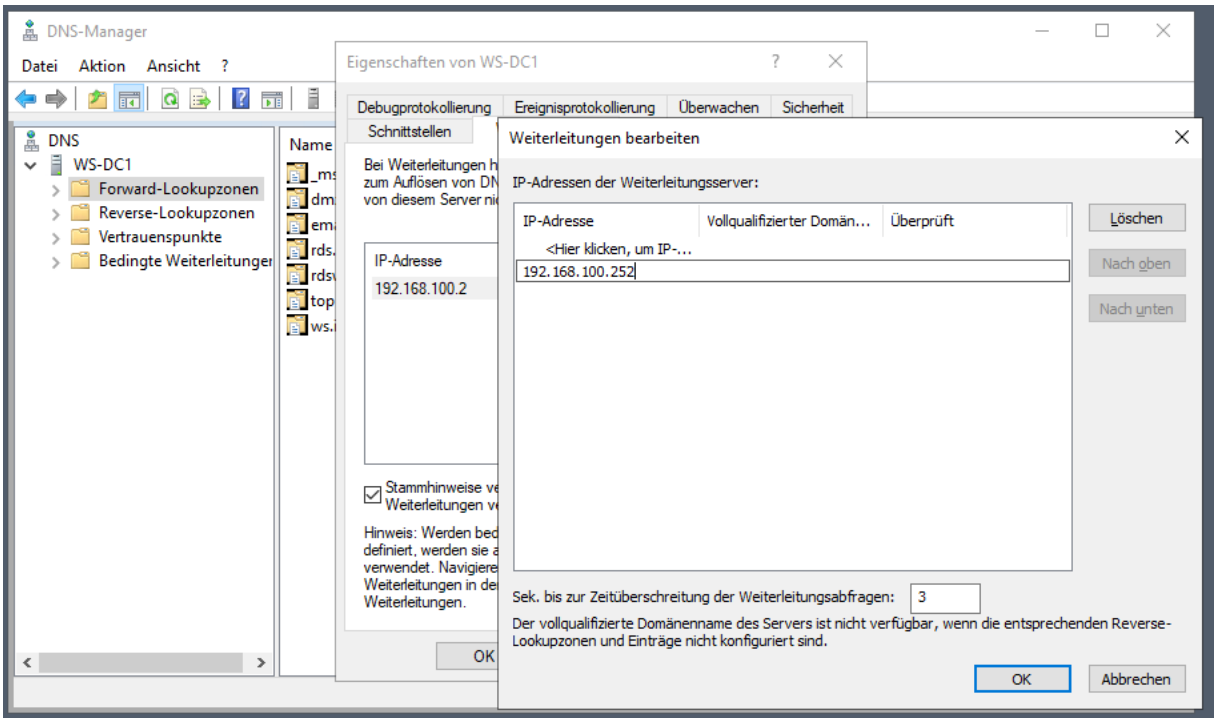
```
1  Import-Module ActiveDirectory
2
3  Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole DomainNamingMaster
4  Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole InfrastructureMaster
5  Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole PDCEmulator
6  Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole RIDMaster
7  Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole SchemaMaster
8
9  netdom /query fsmo
```

```
PS C:\> netdom /query fsmo
Schemamaster               WS-DC2.ws.its
Dom„nennamen-Master        WS-DC2.ws.its
PDC                         WS-DC2.ws.its
RID-Pool-Manager            WS-DC2.ws.its
Infrastrukturmaster       WS-DC2.ws.its
Der Befehl wurde ausgefhrt.


PS C:\> Import-Module ActiveDirectory

PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole DomainNamingMaster

PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole InfrastructureMaster

PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole PDCEmulator

PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole RIDMaster

PS C:\> Move-ADDirectoryServerOperationMasterRole -Identity WS-DC1 -OperationMasterRole SchemaMaster

PS C:\> netdom /query fsmo
Schemamaster               WS-DC1.ws.its
Dom„nennamen-Master        WS-DC1.ws.its
PDC                         WS-DC1.ws.its
RID-Pool-Manager            WS-DC1.ws.its
Infrastrukturmaster       WS-DC1.ws.its
Der Befehl wurde ausgefhrt.
```

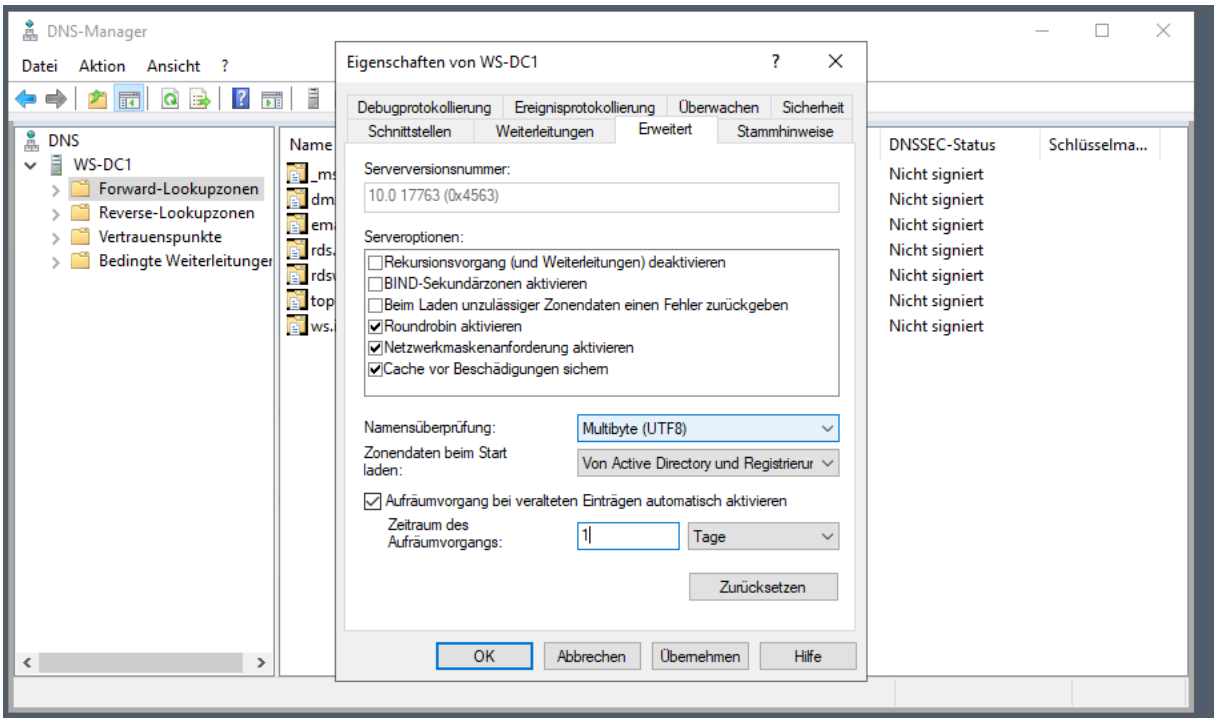
Der Domain Controller ist einsatzbereit.

Installation der Rolle DNS

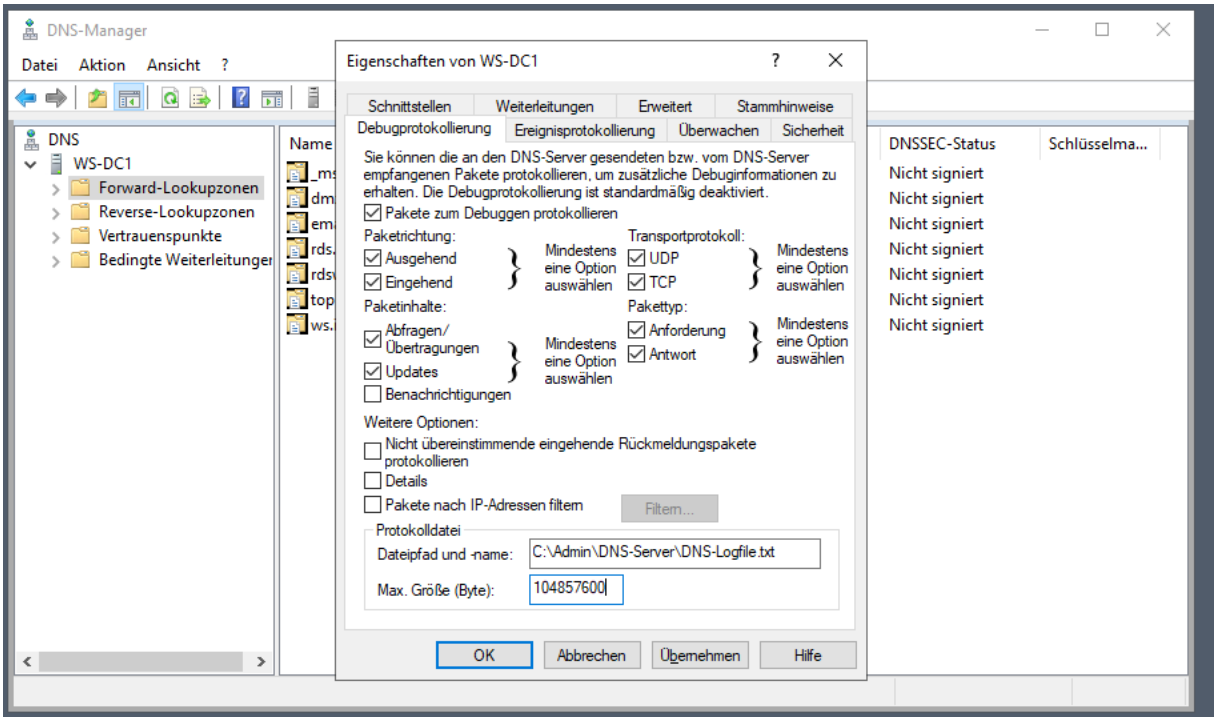
Die Rolle DNS wurde im Setup des Active Directory mit konfiguriert. Hier fehlt nur etwas Feintuning. Zuerst stelle ich den Forwarder auf meine Fritzbox um. Hier stand bis eben noch die IP-Adresse des Domain Controllers WS-DC2:



Dann passe ich die Zeiten für das Aufräumen an:



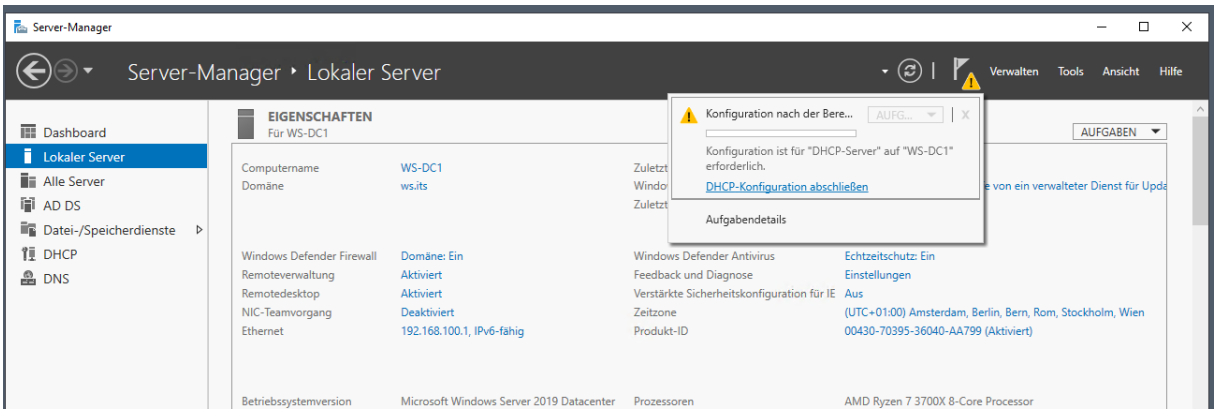
Das Logging aktiviere ich ebenfalls. Die Logdatei lenke ich aber in ein anderes Verzeichnis um:

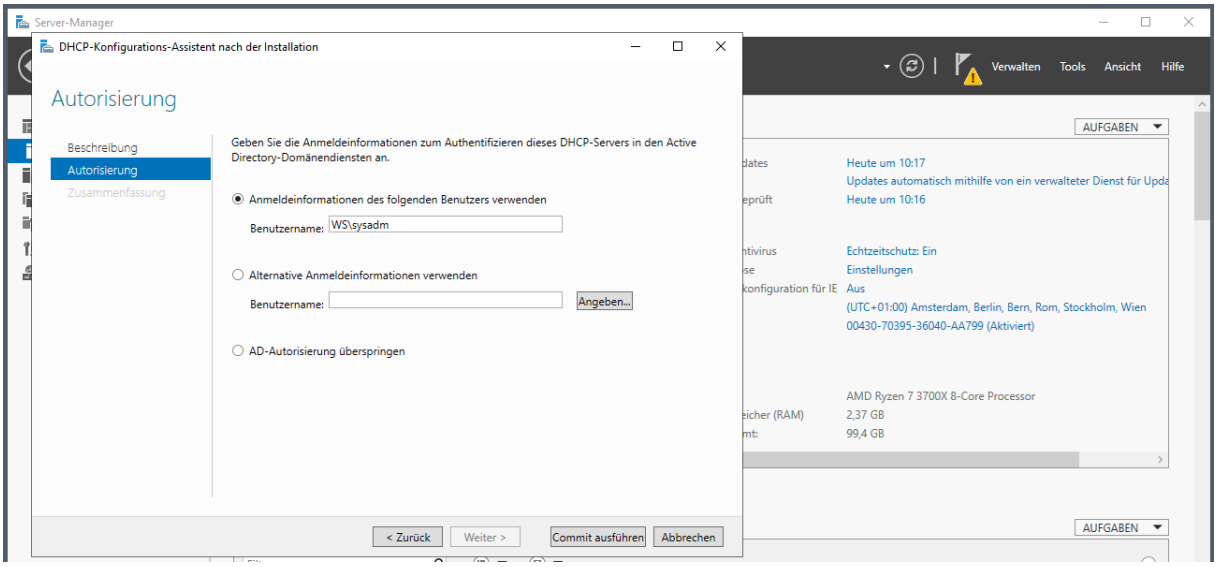


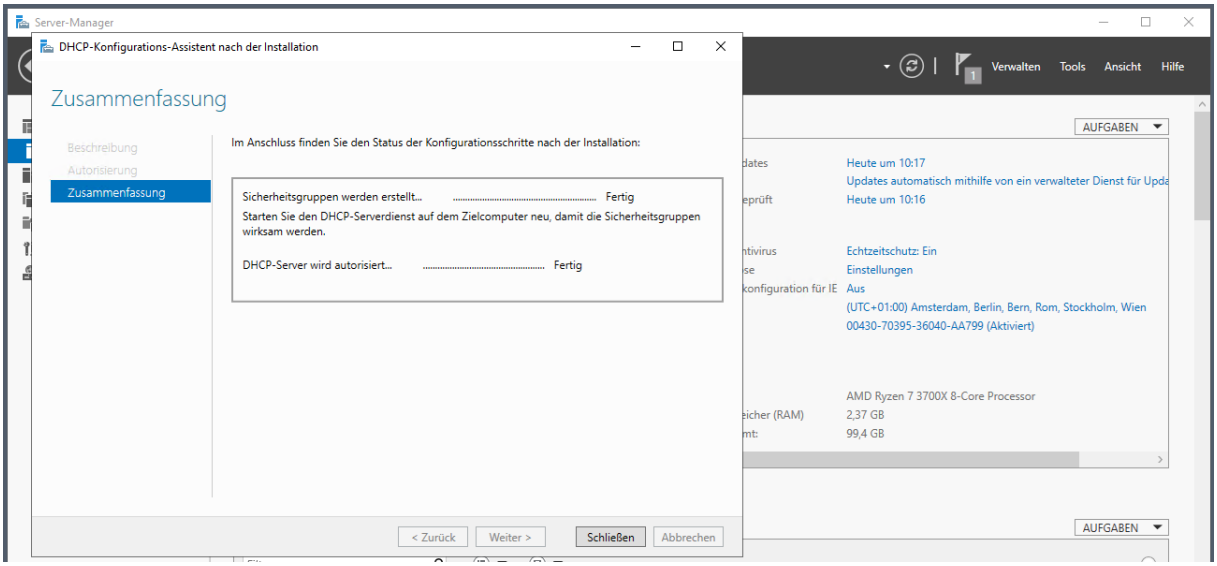
Das war auch schon alles. Der ganze Rest kam über die Active Directory Integration.

Installation der Rolle DHCP

Die Rolle DHCP habe ich vorhin mitinstalliert. Deren Konfiguration starte ich im Server Manager:

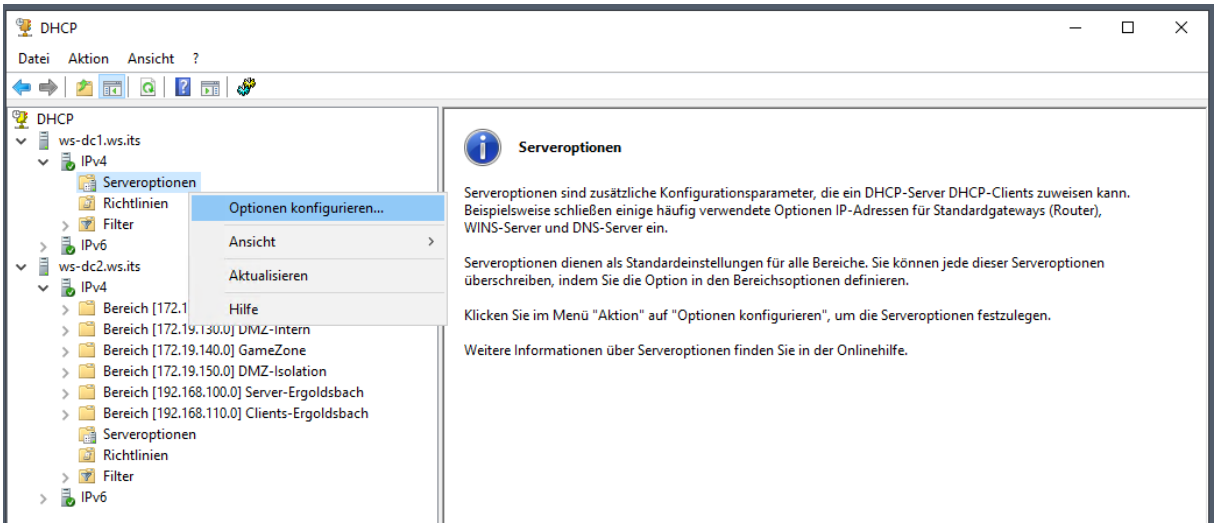




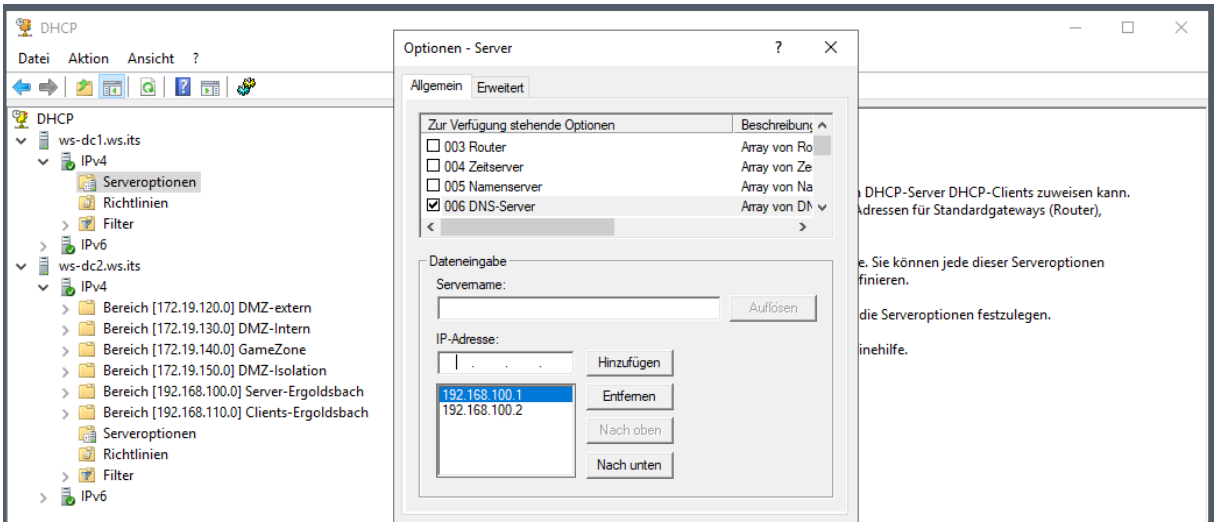


Die Autorisierung wurde erfolgreich vorgenommen.

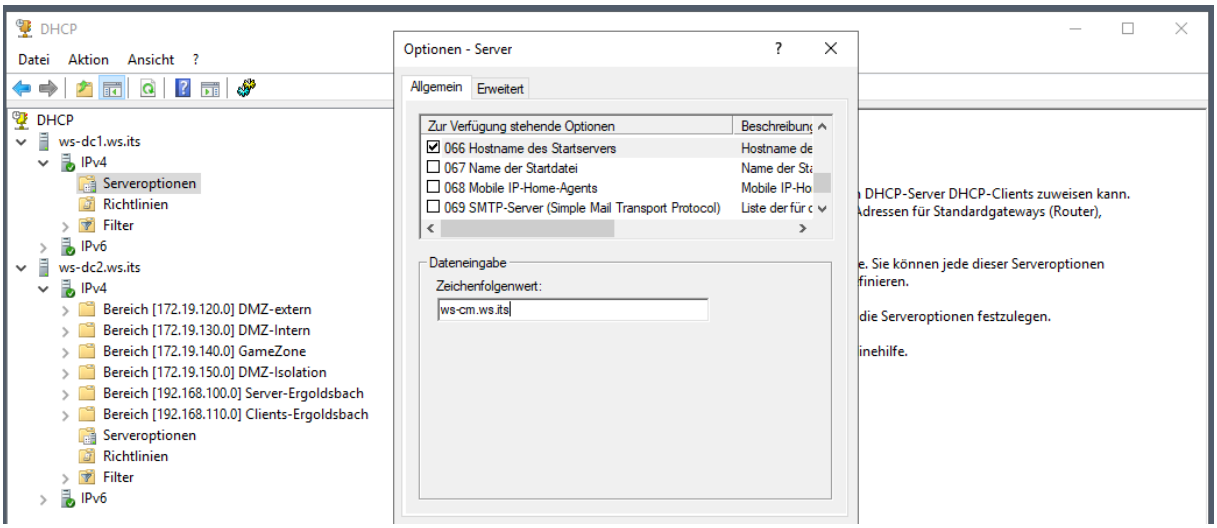
Im nächsten Schritt konfiguriere ich die Serveroptionen:

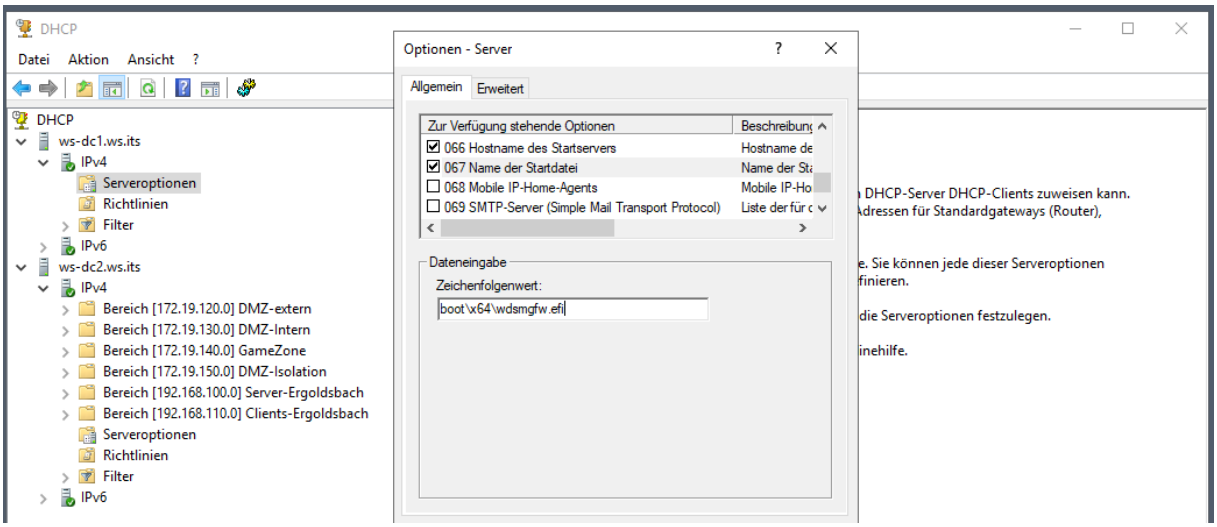


Die DNS-Server und ihre Reihenfolge werden in alle Scopes übernommen. WS-DC1 nennt seine IPv4 als primären DNS. Der WS-DC2 dagegen veröffentlich seine eigene IP als primär. So erreiche ich eine Lastverteilung, da ja auch die Clientanfragen auf beide DHCP-Server verteilt werden:

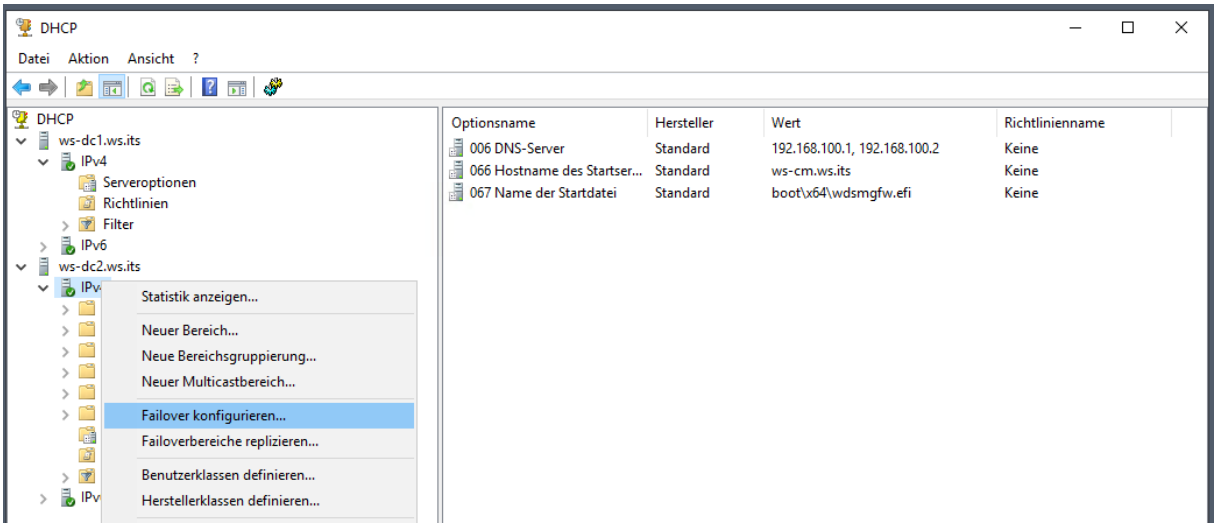


Mein Deployment-Server wird ebenfalls zentral im DHCP veröffentlicht:

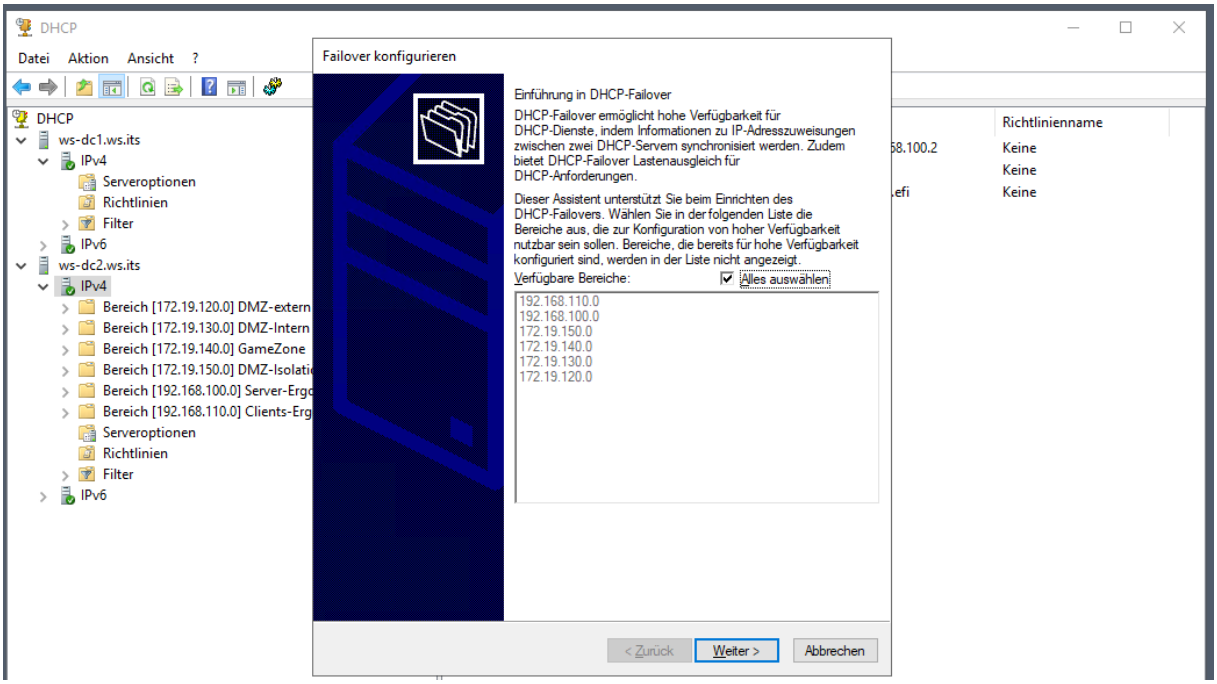




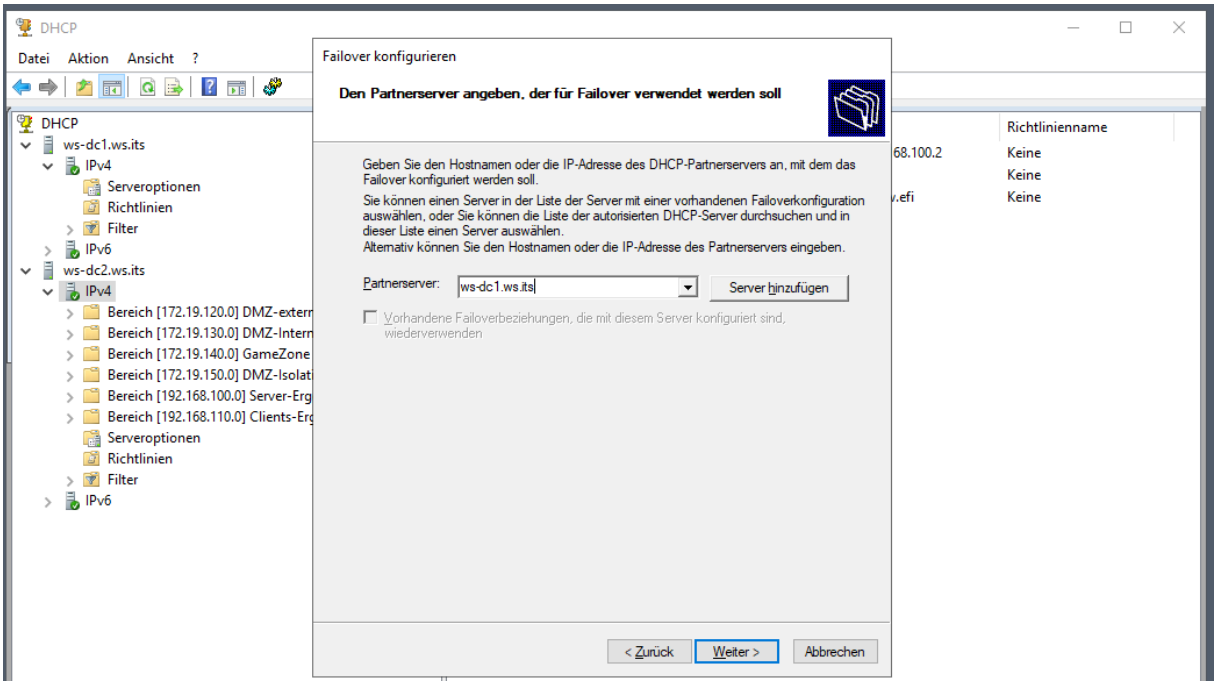
Mehr Optionen habe ich vorher auch nicht verwendet. Ich starte nun die Konfiguration des DHCP-Failovers auf dem noch aktiven DHCP-Server WS-DC2:



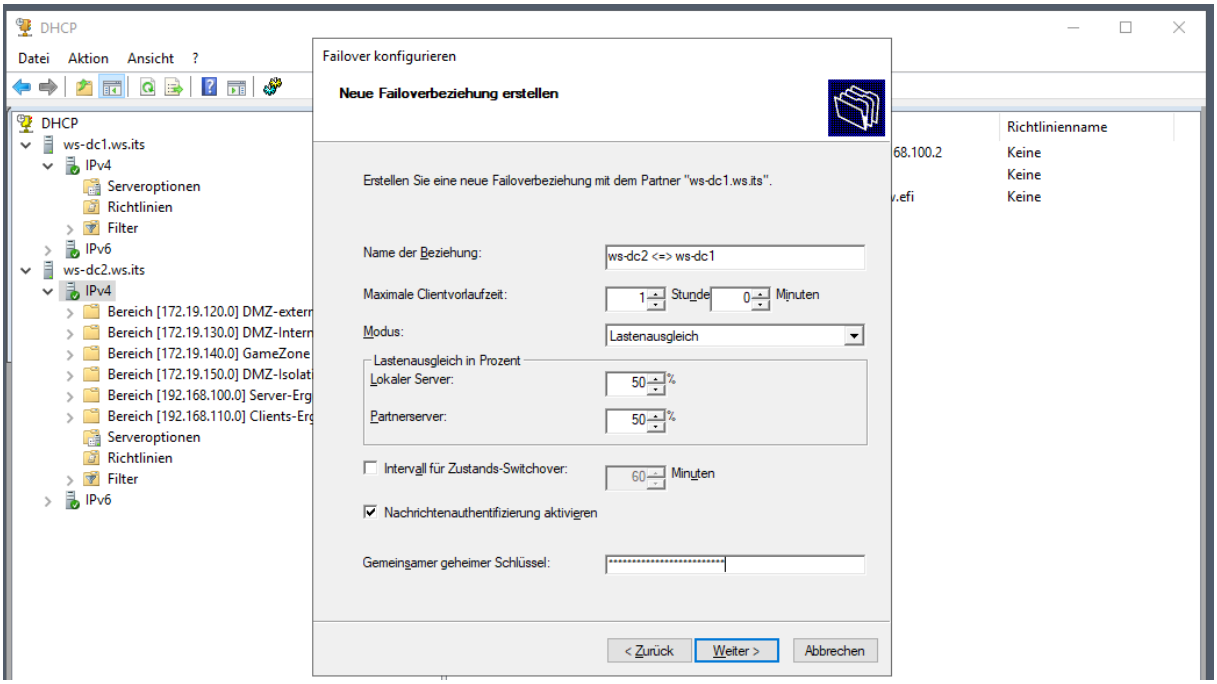
Ich möchte alle Scopes höher verfügbar gestalten:



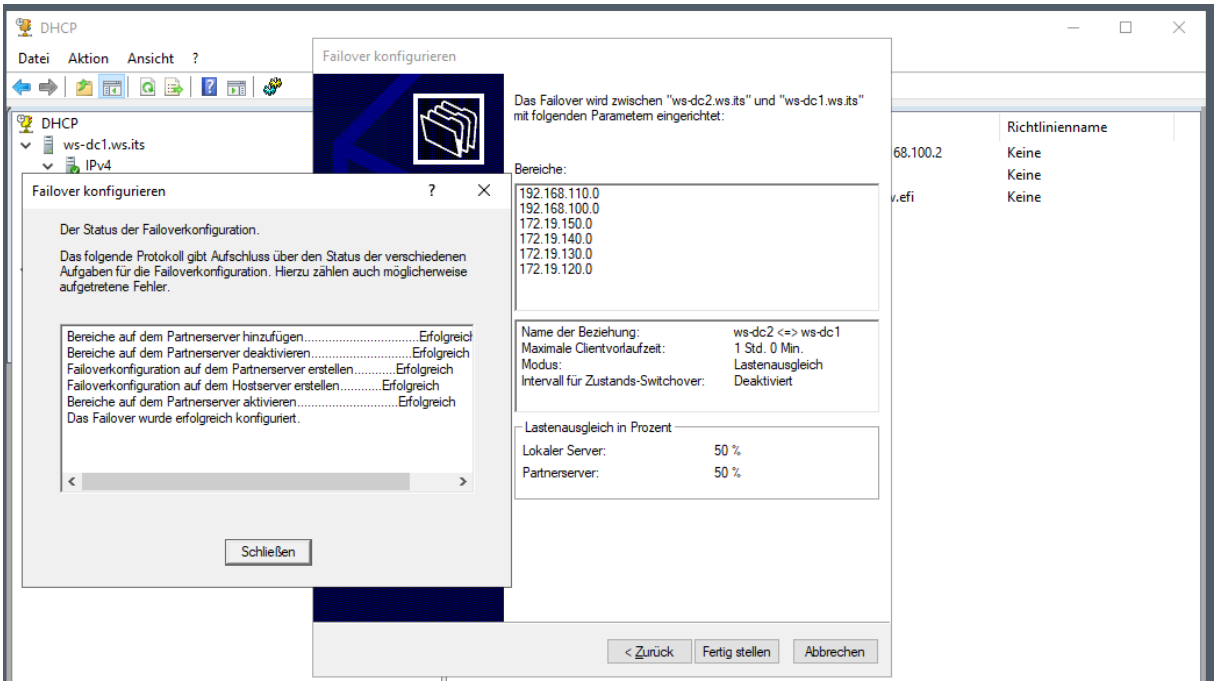
Der neue Partnerserver ist der neue Windows Server 2019:



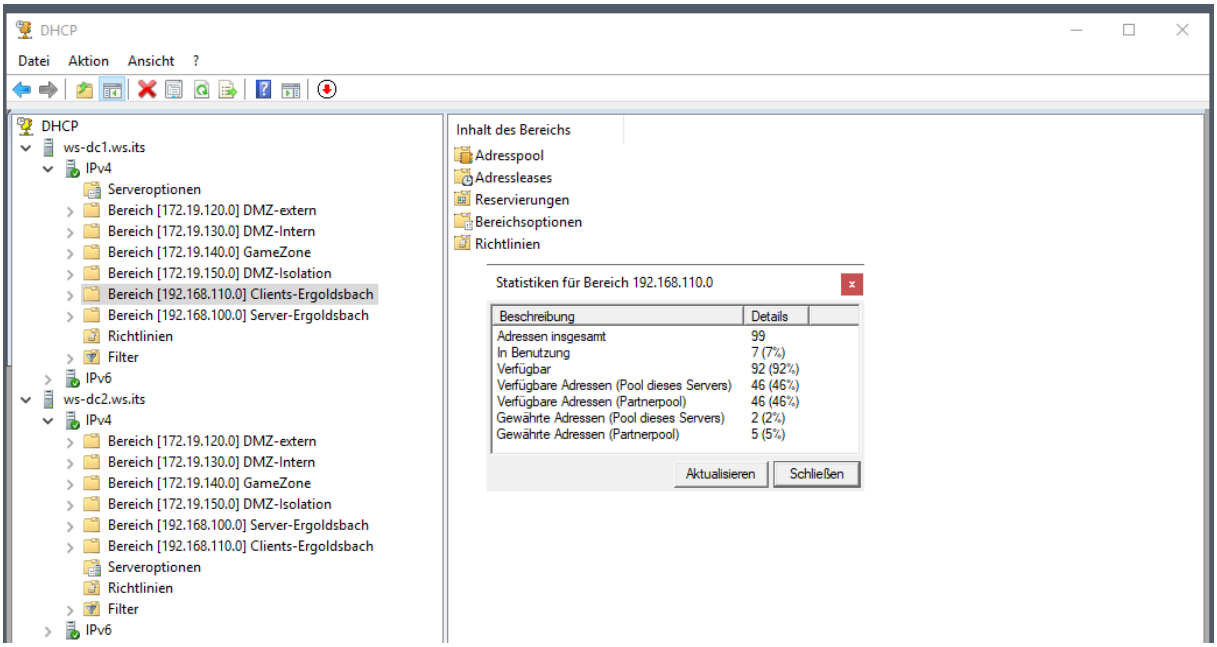
Die Konfiguration des Failovers belasse ich weitestgehend im Standard. Das Passwort ist eine zufällige Zeichenfolge auf der Tastatur:



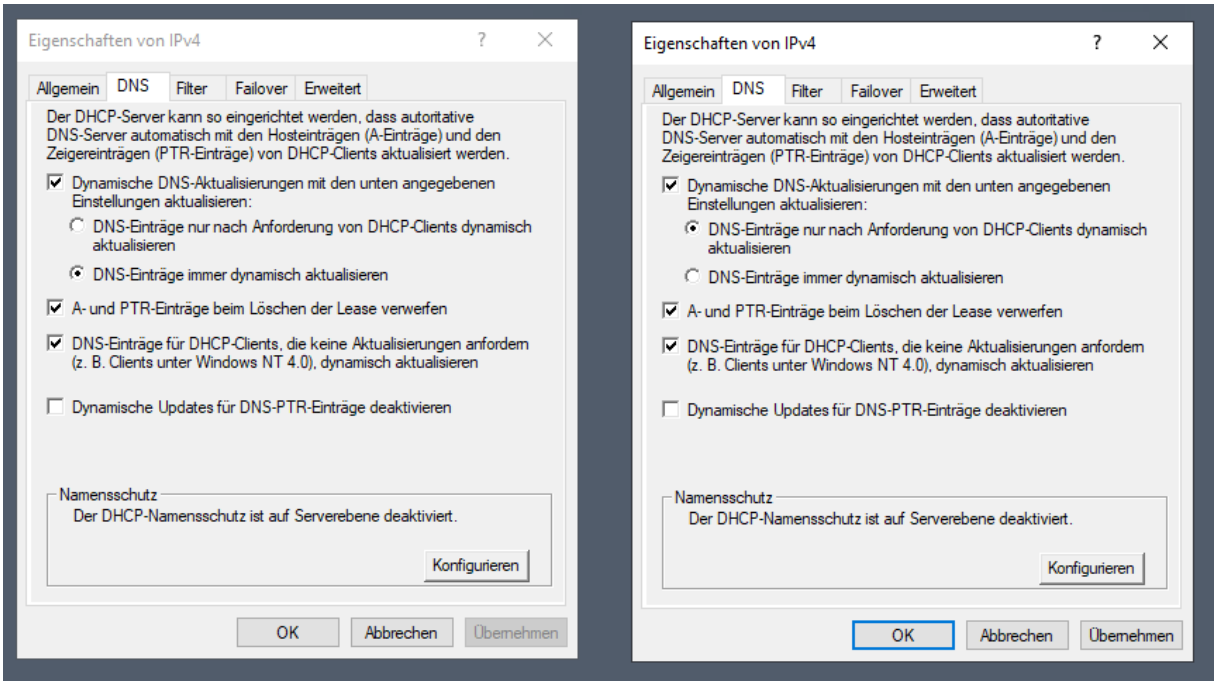
Die Angleichung dauert wenige Sekunden:



Nach einer Aktualisierung sind die Scopes sichtbar. Und auch die Aufteilung der IP-Adressen hat automatisch funktioniert. Clients werden bei einem DHCP-Renew keine Probleme haben:



Jetzt gleiche ich noch die DHCP-Konfiguration beider Server an:

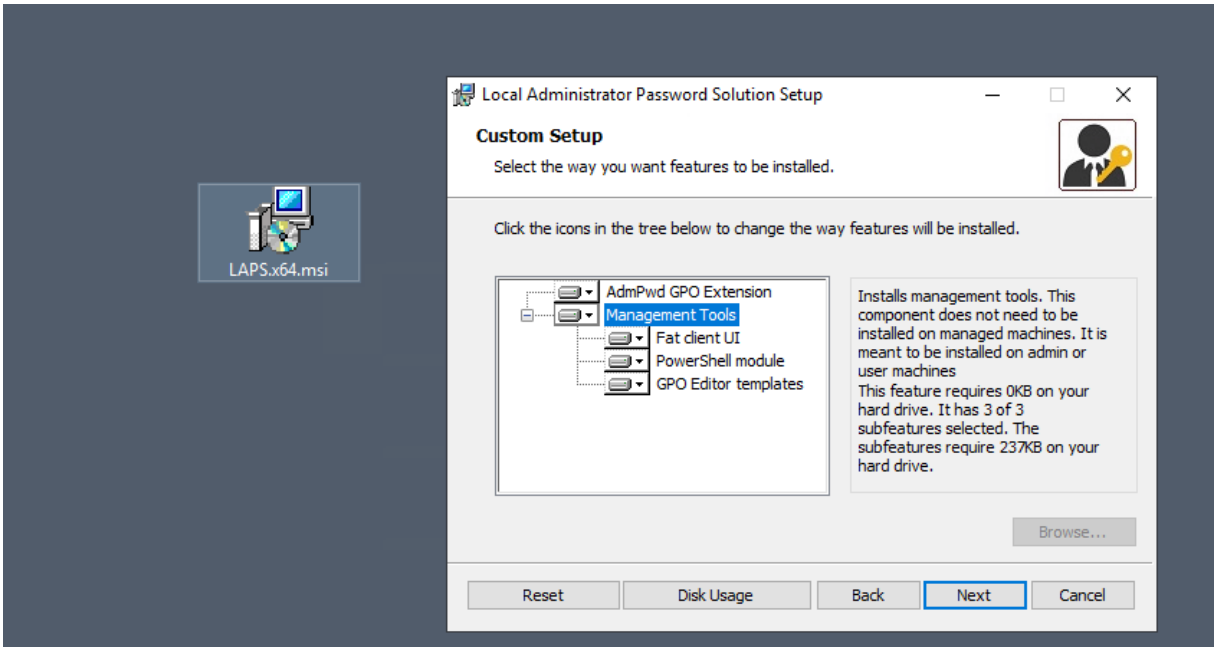


Das war sehr einfach.

Nacharbeiten

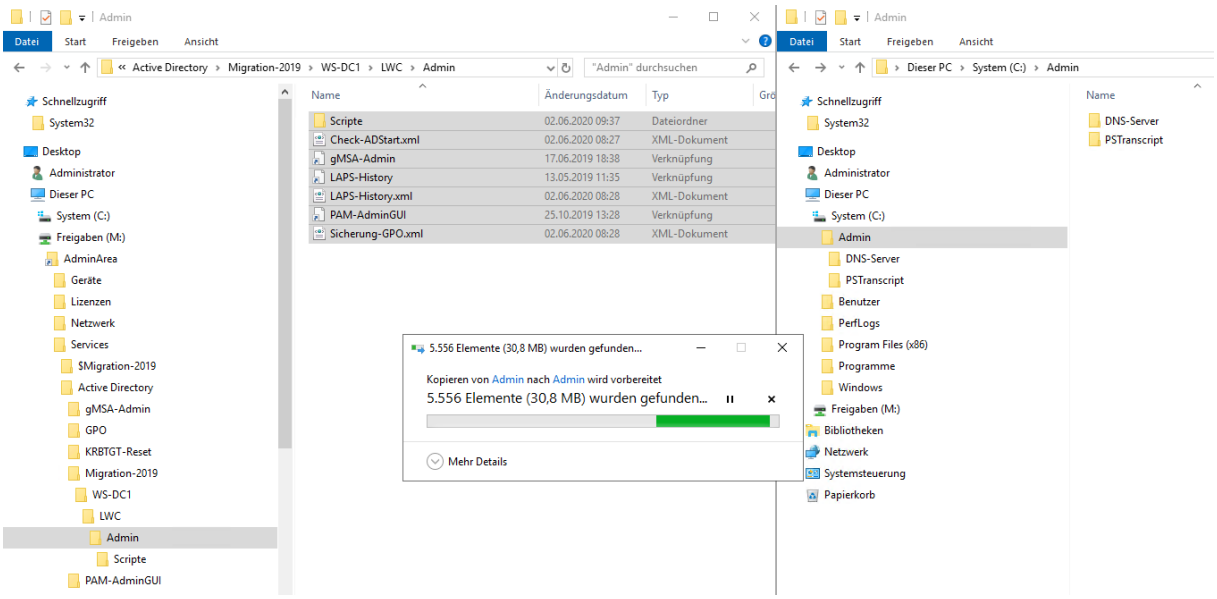
Installation LAPS

Es folgt die Installation des LAPS. Damit kann ich die Passworte der lokalen Admins mit einer GUI aus dem AD heraus auslesen:

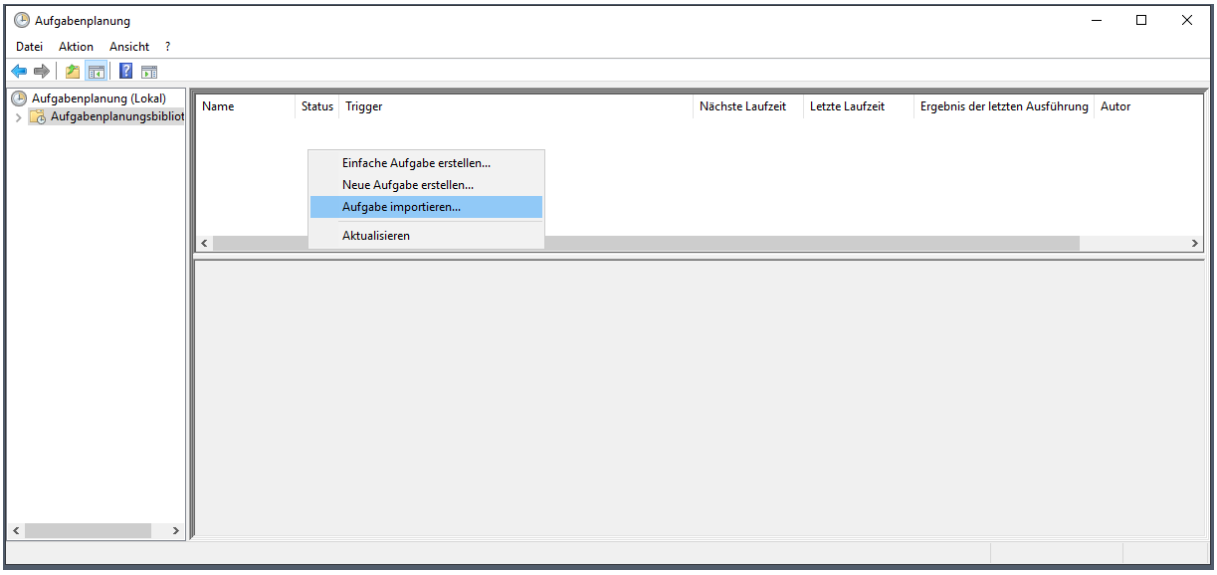


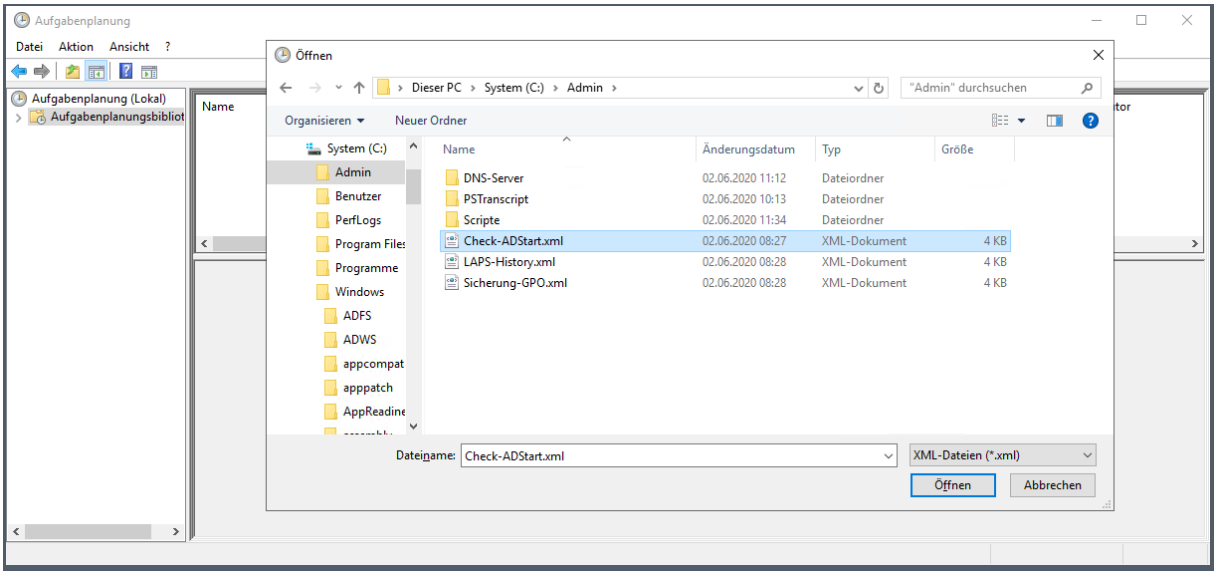
Adminverzeichnis

Auch das Adminverzeichnis c:\admin wird wieder befüllt:

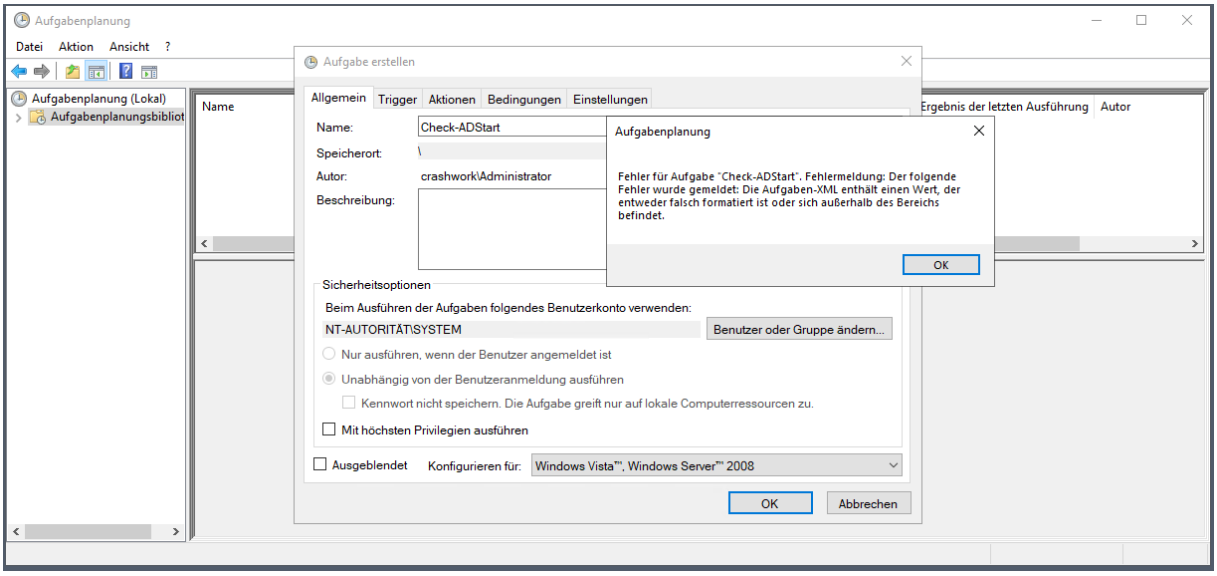


Darin sind auch die xml-Dateien der Aufgabenplaung enthalten. Diese importiere ich im Anschluss:

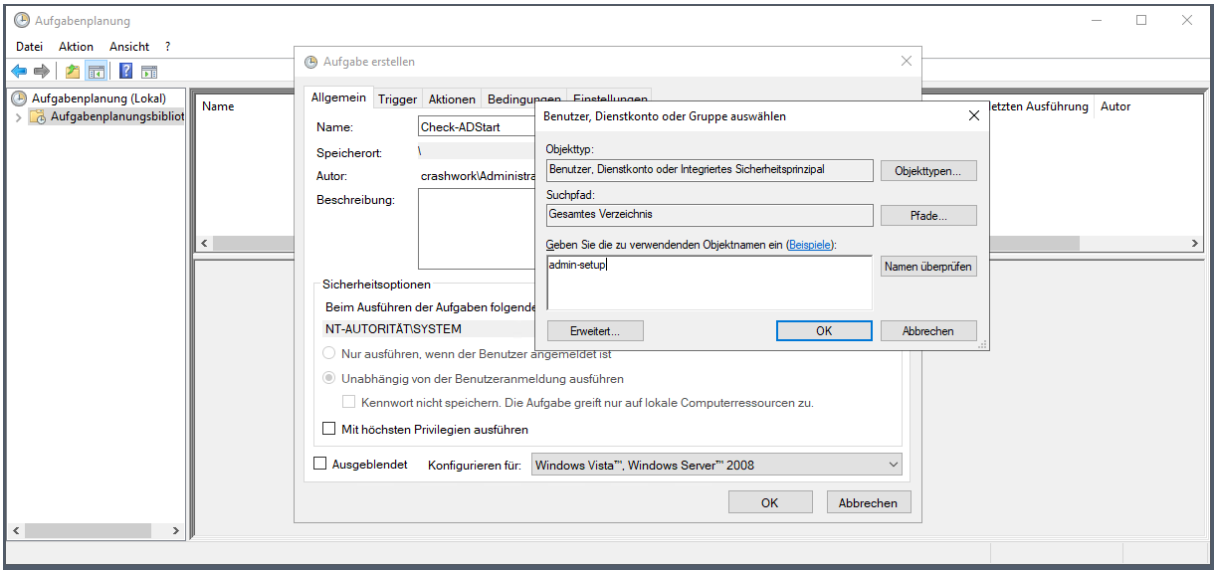




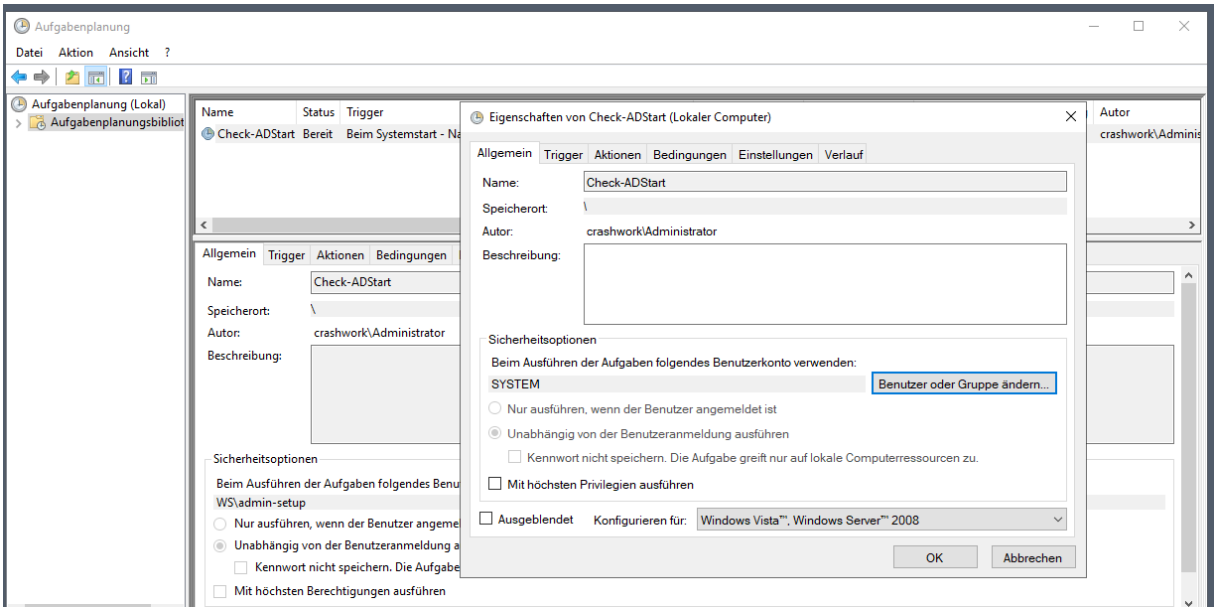
Der Import schlägt fehl:



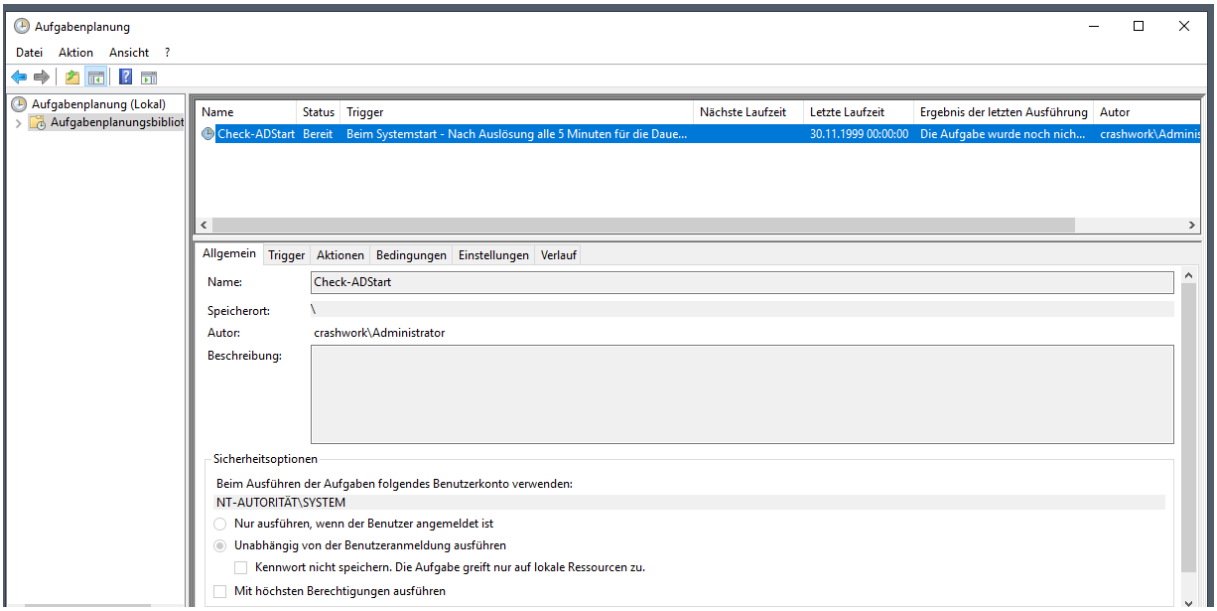
Die Ursache liegt wahrscheinlich beim hinterlegten Konto. Ich trage versuchshalber meinen Dummy-Admin ein:



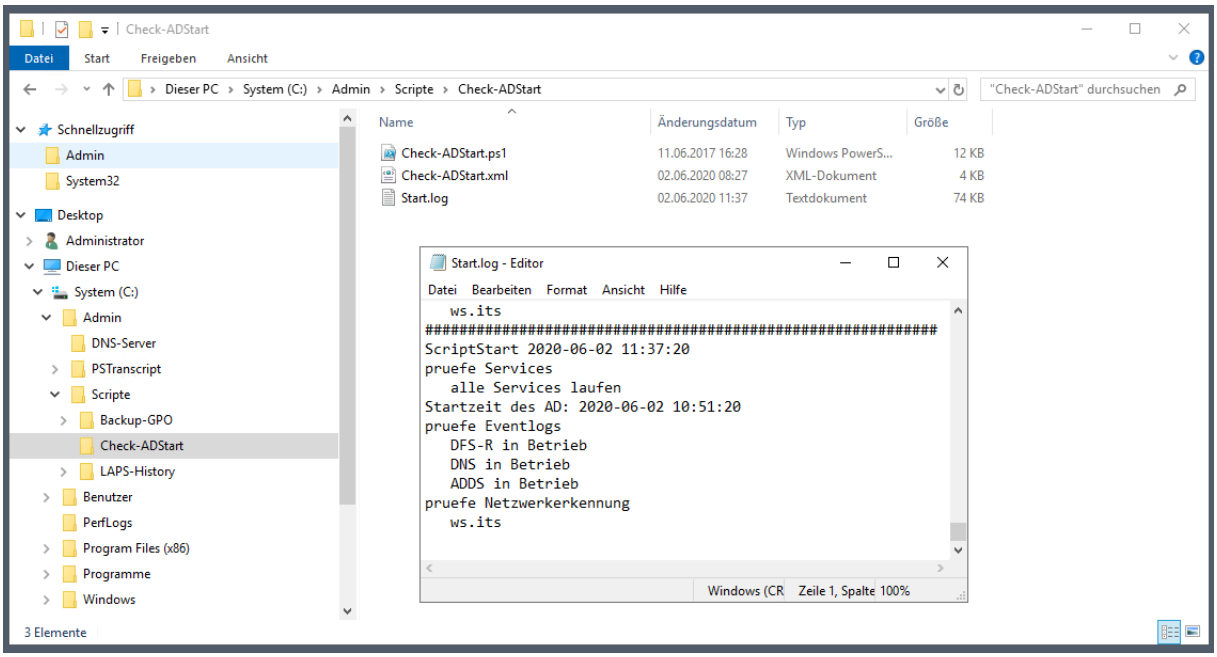
Das hat funktioniert. Jetzt stelle ich den System-Account wieder ein:



Auch das hat funktioniert…

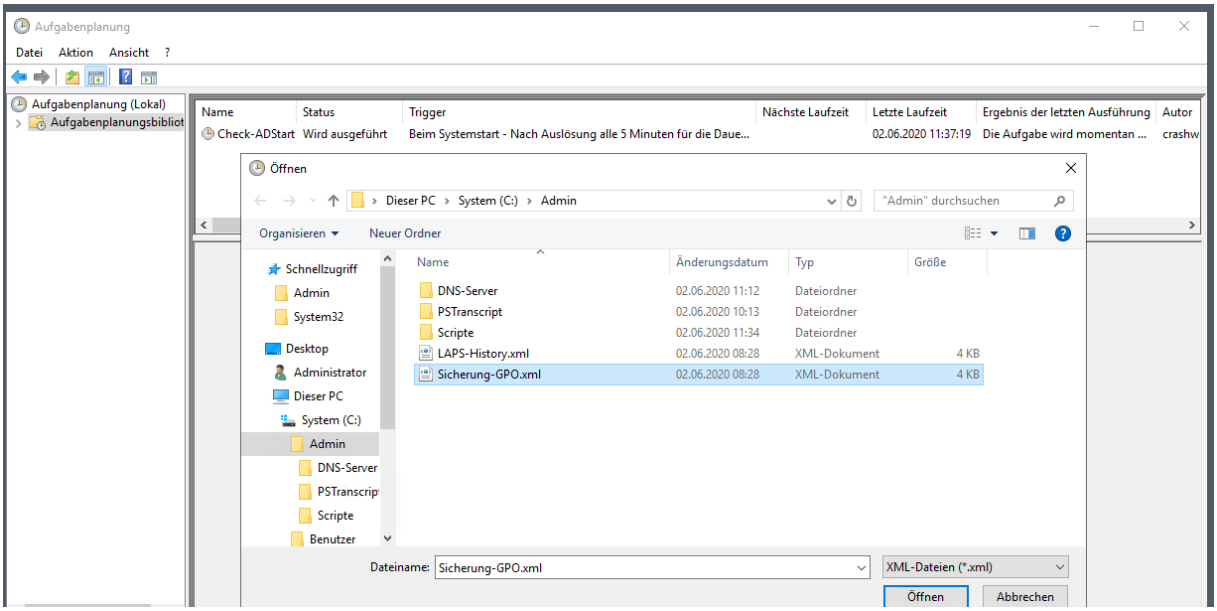


Diese erste Aufgabe wird nach jedem Start ausgeführt. Das aufgerufene PowerShell-Script prüft, ob alle relevanten Dienste des Domain Controllers gestartet wurden und ob die erwarteten Eventlogs protokolliert worden. Wenn da etwas nicht stimmt, dann wird eine Korrektur gestartet. Der gesamte Vorgang wird dann in einer Textdatei protokolliert. Der letzte Start war erfolgreich:

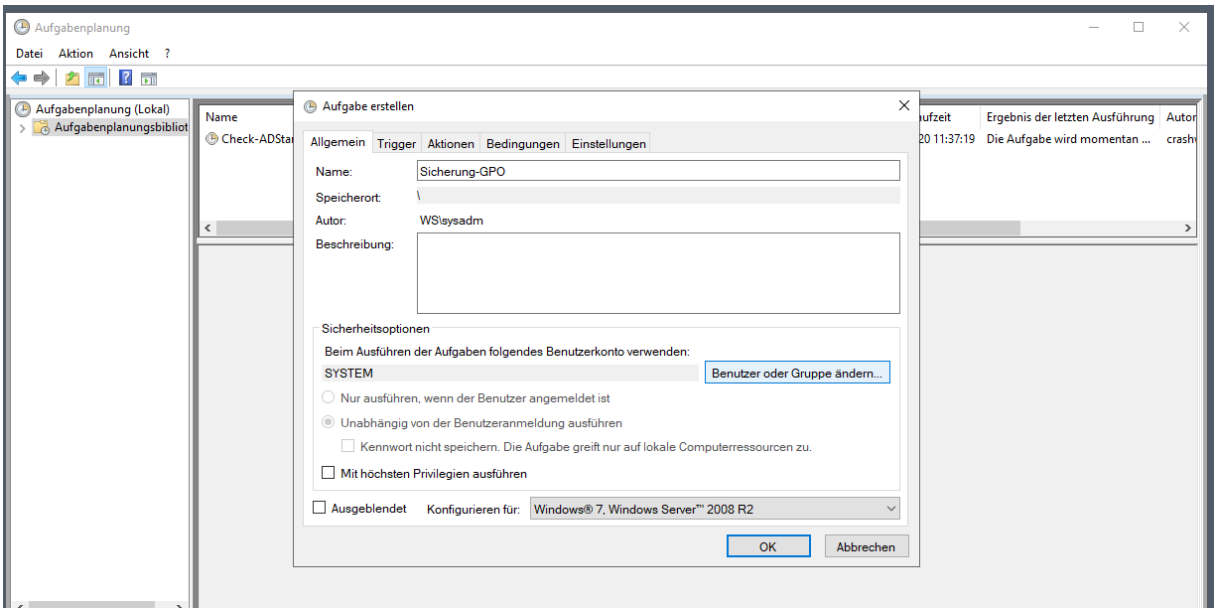


Datensicherung der GPO

Ich habe ein weiteres Script, das über einen Task gestartet wird. Dieses sichert mir in einem Rotationsverfahren alle Gruppenrichtlinien. Ich importiere den Task:



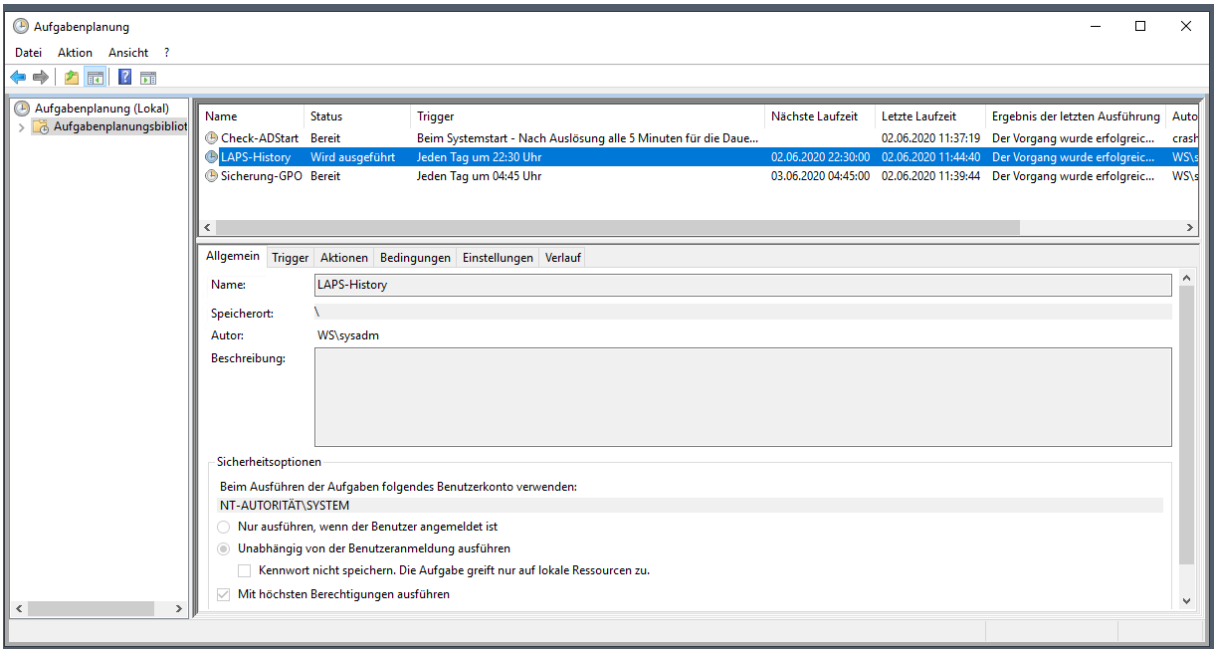
Auch hier stand NT-Autorität\System als Taskaccount in der XML-Datei drin. Ich verändere den Account auf System. So lässt sich die Aufgabe ohne den XML-Fehler speichern:



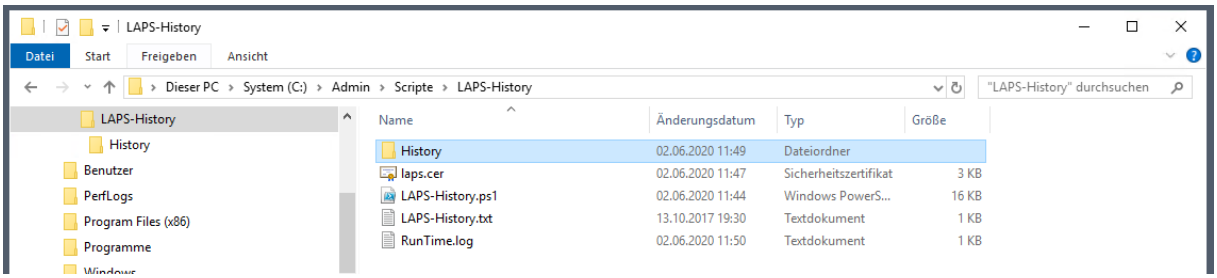
Ich starte den Task. Wenig später erhalte ich eine Mail mit der Information zur erfolgreichen Sicherung meiner GPO:

Datensicherung LAPS (Script LAPS-History)

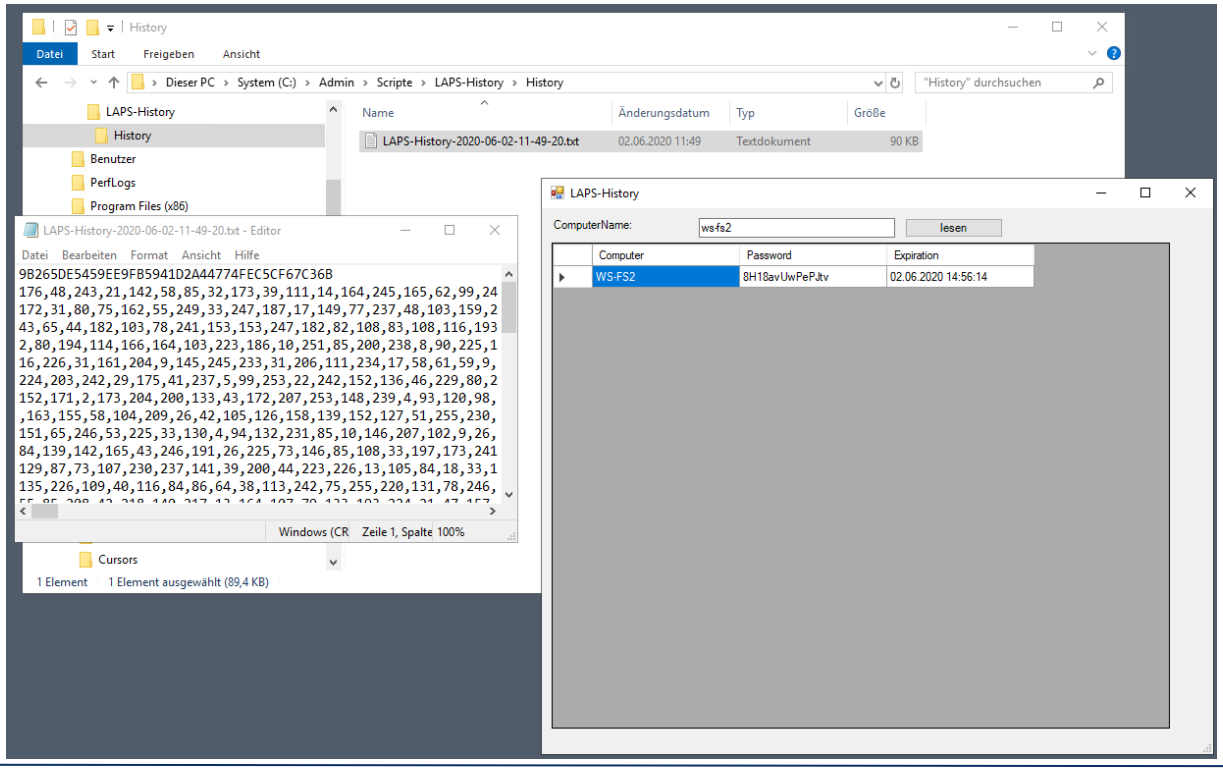
Ein weiteres Script sichert mir jeden Tag die Passwörter der lokalen Adminkonten aller Memberserver und Clients. Diese werden alle 30 Tage automatisch neu vergeben. Der Task ist schnell importiert. Das dazugeförige Script liegt bereits unter c:\admin\scripte:



Es handelt sich um ein PowerShell-Script:

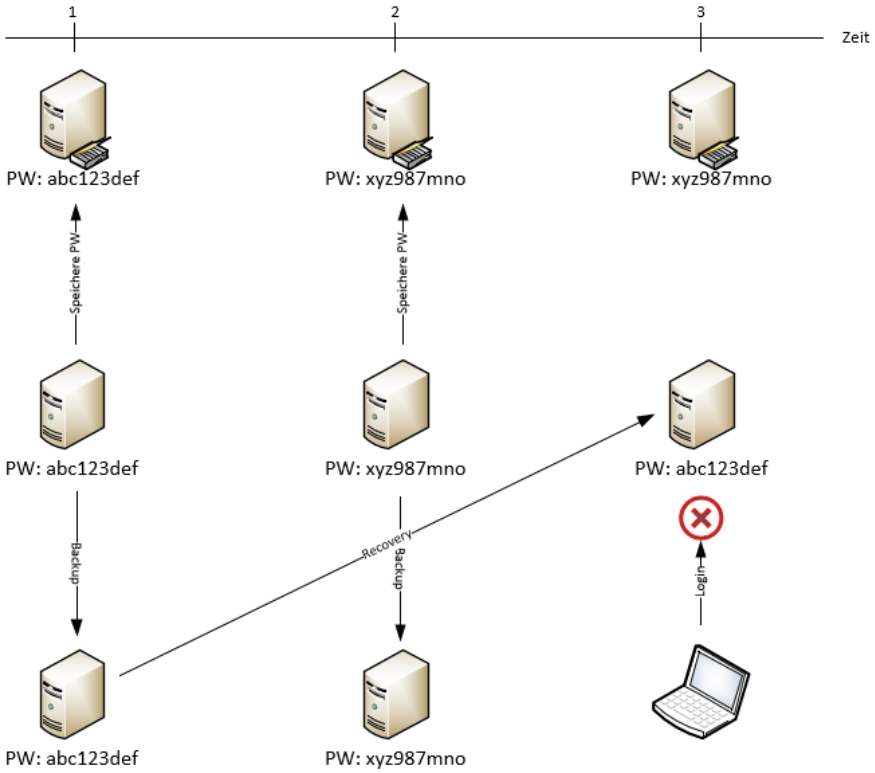


Das Script kann auch direkt gestartet werden. Dann zeigt es eine grafische Oberfläche, in der ich nach einem Computer suchen kann. Dann zeigt es mir alle gespeicherten Passwörter an. Die sensiblen Informationen werden natürlich gut geschützt: Das Script verschlüsselt die Daten mit einem Public-Key, der als CER-Datei im Scriptverzeichnis liegt:



Nur der Owner des dazugehörigen Private-Keys kann mit dem Script die Daten entschlüsseln.

Hintergrund:
Warum ich mir die alten LAPS-Passwörter merke? Ganz einfach: Für Recovery-Szenarien. Ich hab mal das typische Szenario mit einer Zeitachse dargestellt:



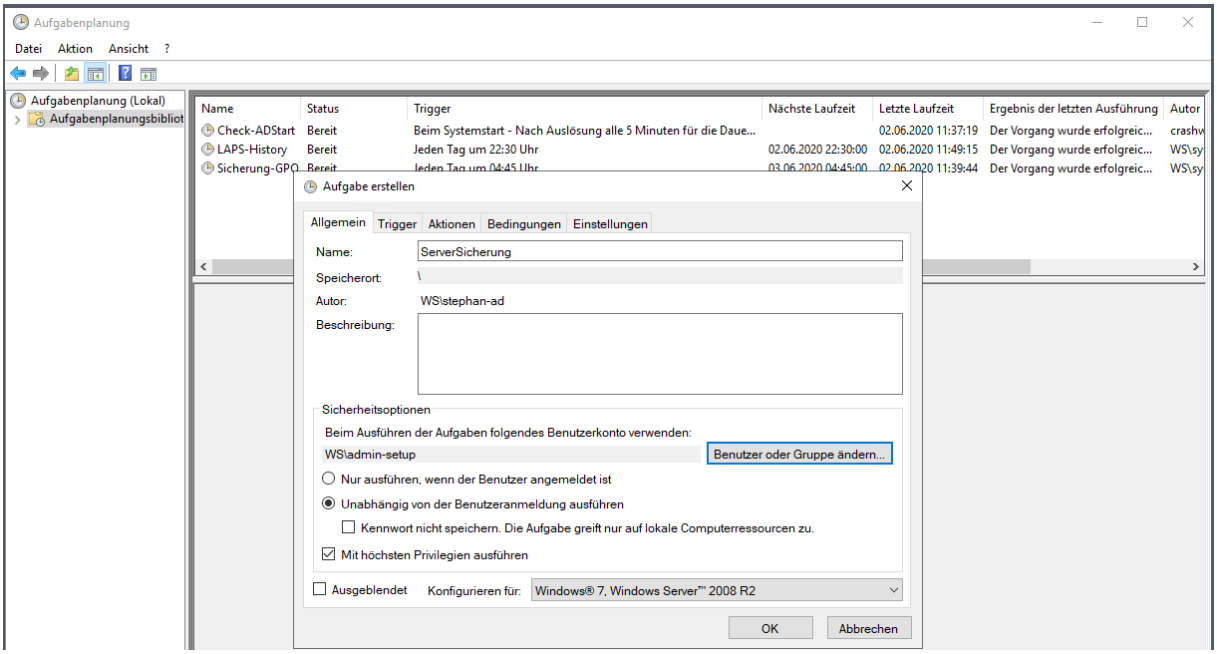
Der Computer erstellt zum Zeitpunkt 1 eine Datensicherung. Das Passwort ist mit dem Active Directory synchron. Zum Zeitpunkt 2 verändert der Computer sein Passwort und speichert es im Active Directory ab. Zum Zeitpunkt 3 muss der Server wiederhergestellt werden. Dabei wird die Sicherung vom Zeitpunkt 1 verwendet. Jetzt ist das Passwort des lokalen Admins nicht mehr mit dem Active Directory synchron. Eine Anmeldung mit dem lokalen Admin ist nicht mehr möglich.

Jetzt gibt es nur noch 3 Optionen:

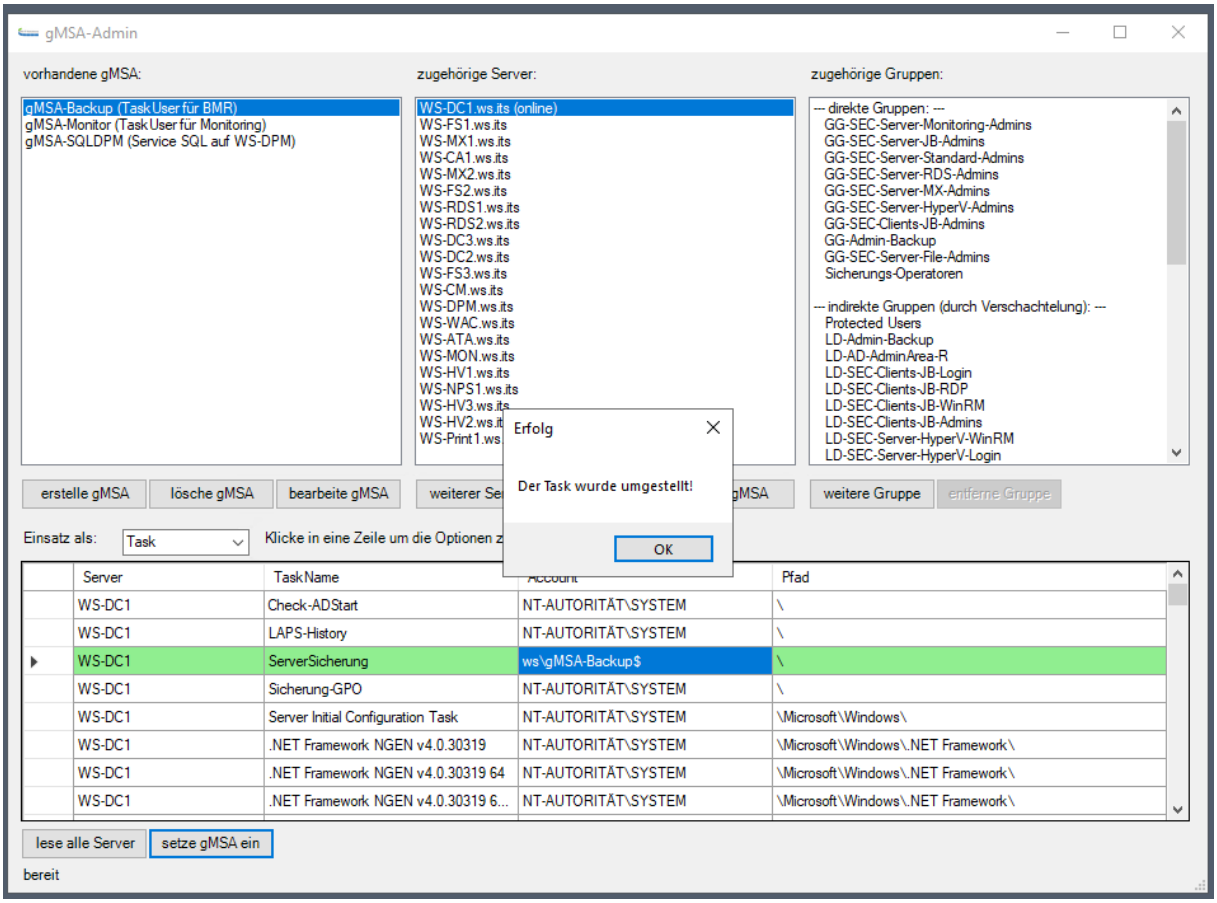
1) Das Computerkonto wird ebenfalls auf den Zeitpunkt 1 wiederhergestellt. Dafür ist eine gefilterte, autorisierende Wiederherstellung notwendig. Inklusive der Downtime eines Domain Controllers…
2) Man hackt sich in den Windows Server rein. Mit den Standardschutzmechanismen dauert das keine 5 Minuten.
3) Man hat eine LAPS-History und sucht darin das Passwort zum Zeitpunkt 1 heraus.

Datensicherung Windows Server

Nun fehlt noch die Datensicherung des Servers. Diese nehme ich wieder mit der Windows Server Sicherung und meinem zentralen Script-Task vor. Ich importiere die Aufgabe. Dabei gebe ich wieder meinen Dummy-Admin an:



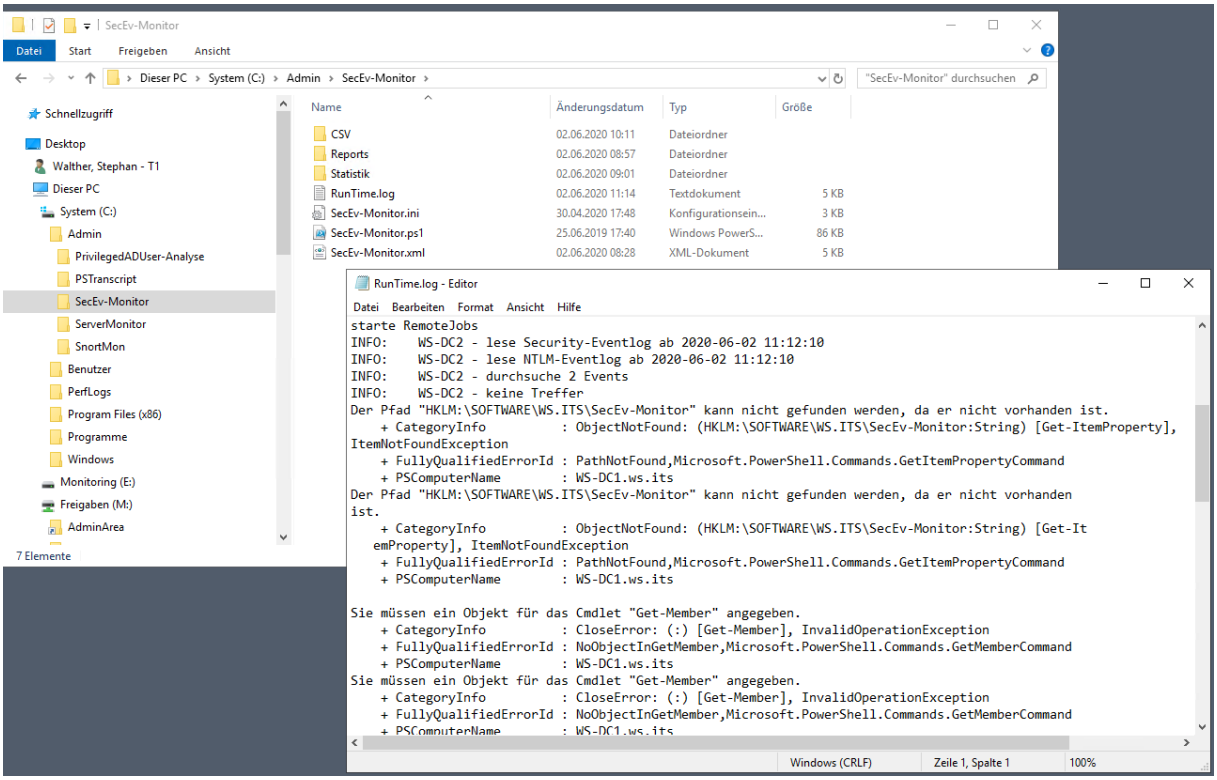
Den eigentlichen Sicherungs-Account trage ich wieder mit meinem PowerShell-Script gMSA-Admin ein:



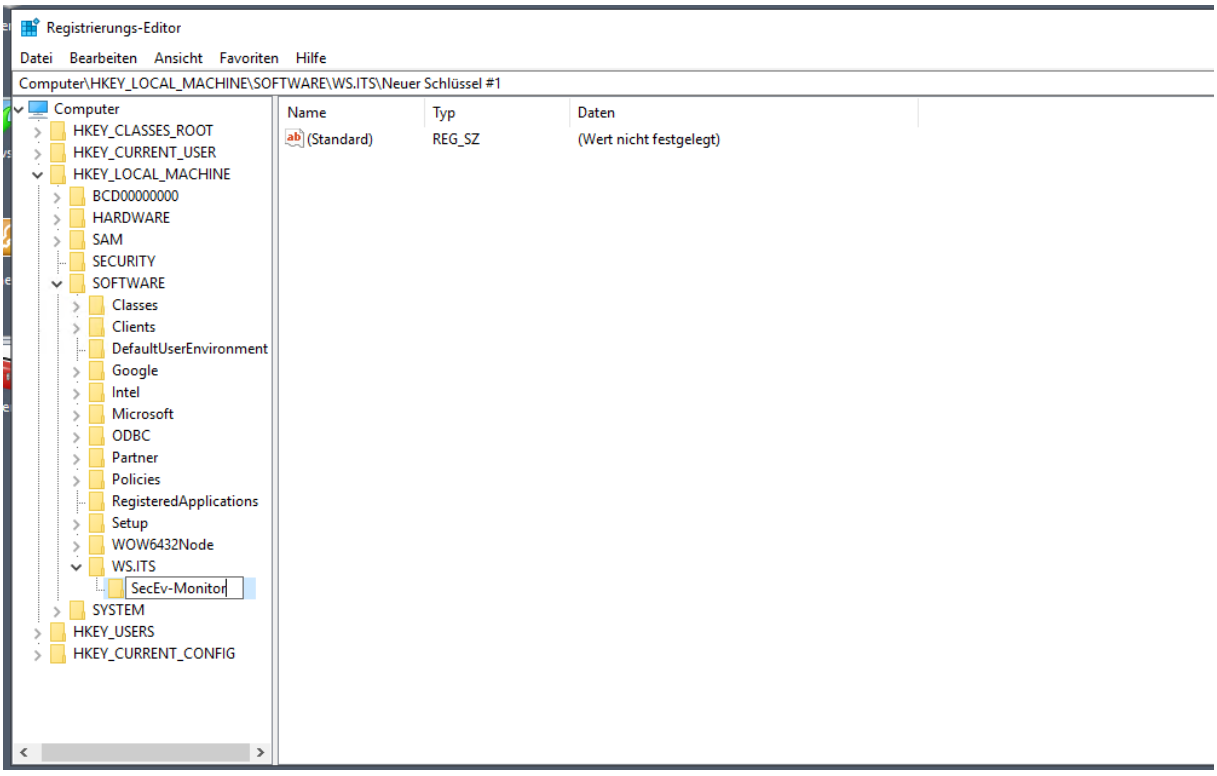
Die Konfiguration der Sicherung muss ich nicht anpassen. Der Server ist mit den vorherigen Einstellungen kompatibel.

TroubleShooting Monitoring

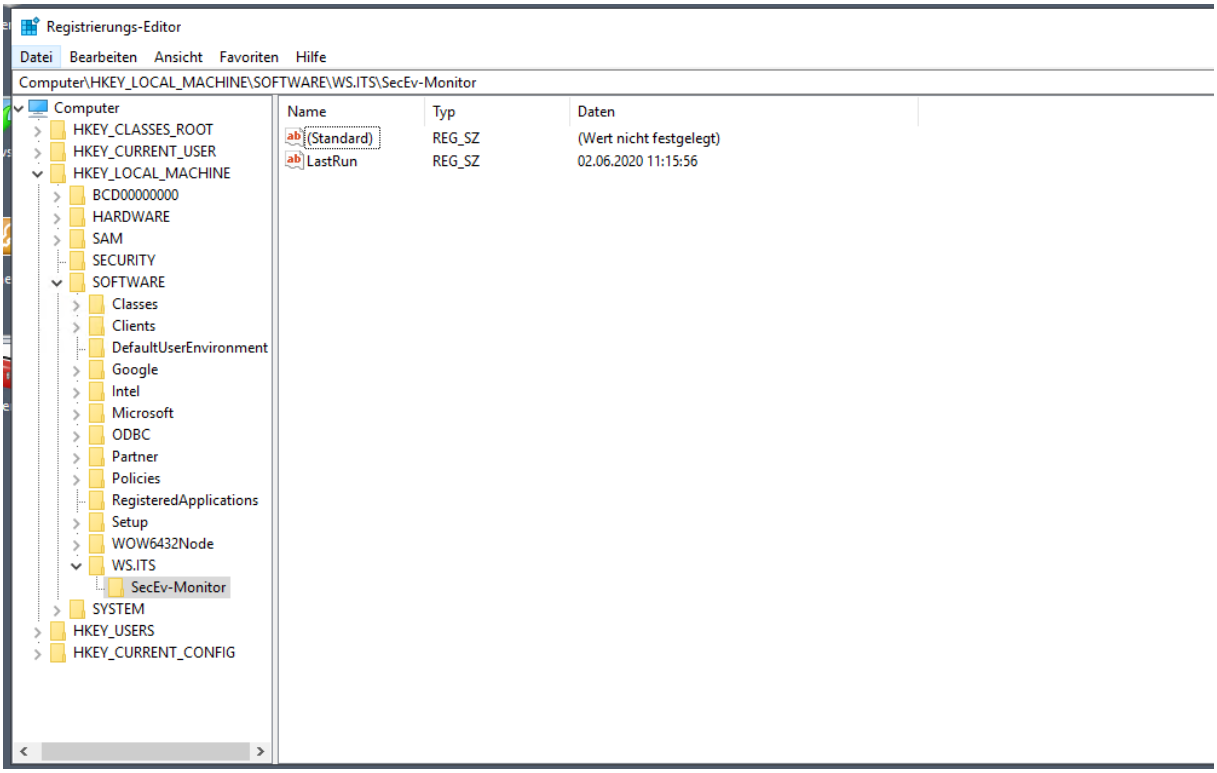
Ich kontrolliere nun das Mnoitoring. Vorhin habe ich mein Script SecEV-Monitor auf den Server WS-MON verschoben. Den alten WS-DC1 konnte ich damit einfach weiter kontrollieren. Der neue Server wehrt sich aber. Das kann ich im Runtime.log sehen:



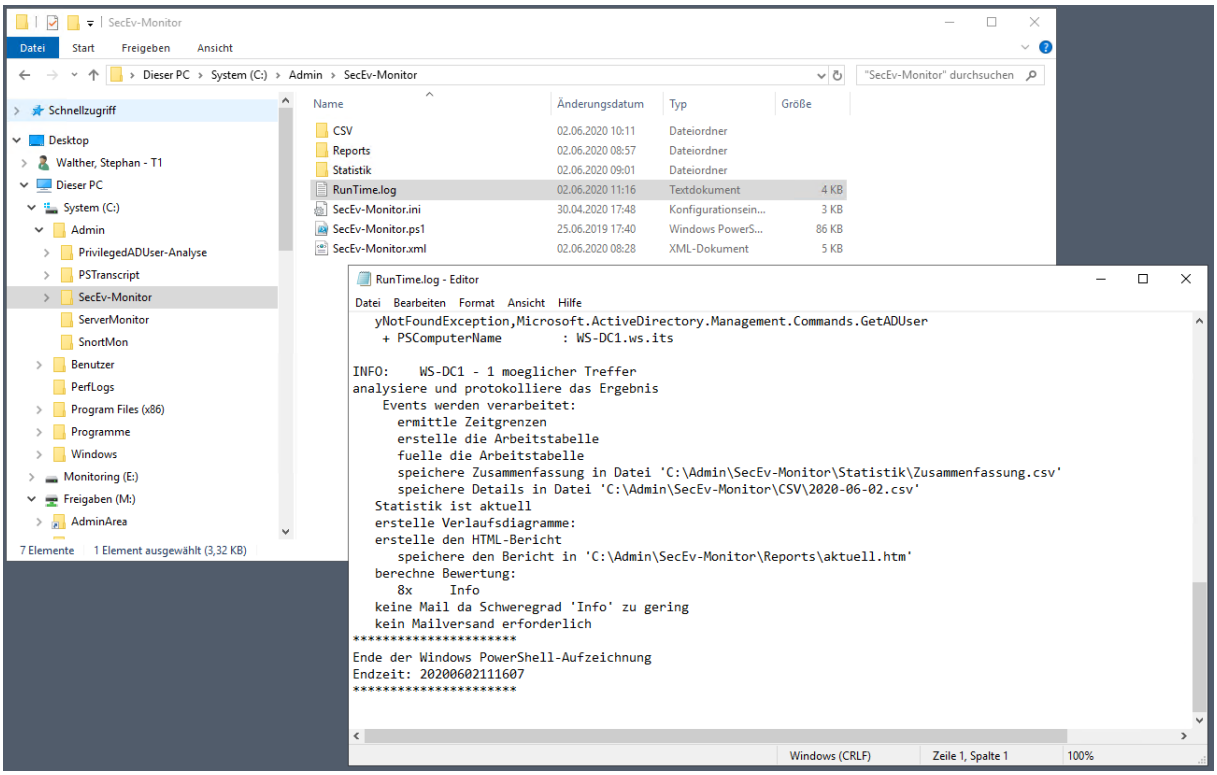
OK, da fehlen Registry-Keys auf dem neuen WS-DC1. Die sollte das Script eigentlich selber erstellen. Aber es ist immer noch eine unfertige Version. Daher helfe ich mal manuell nach:



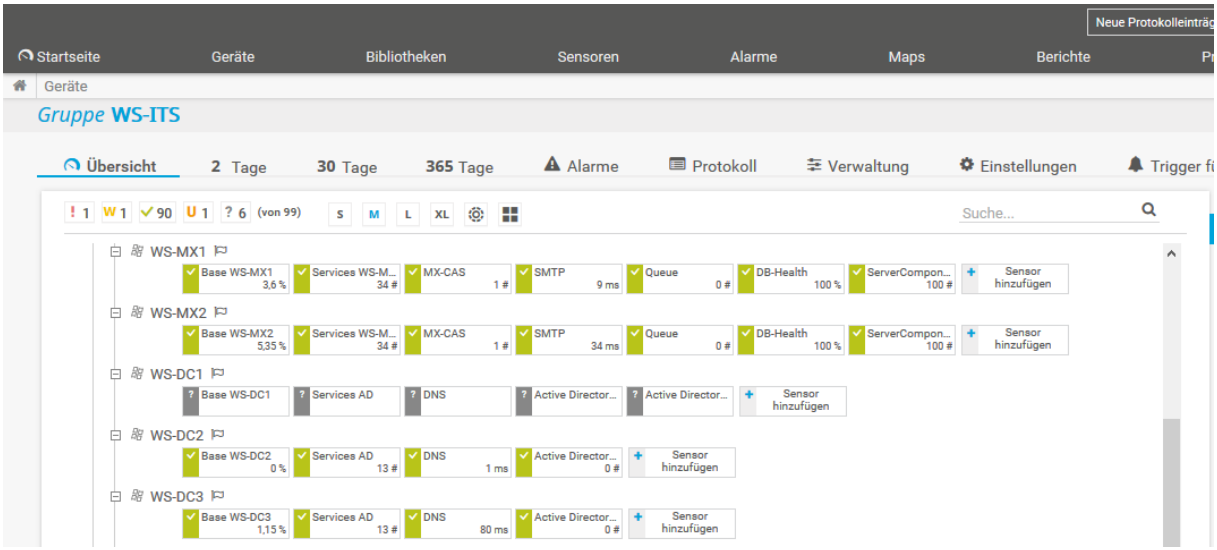
Der nächste Lauf des Scriptes erstellt die Einträge:



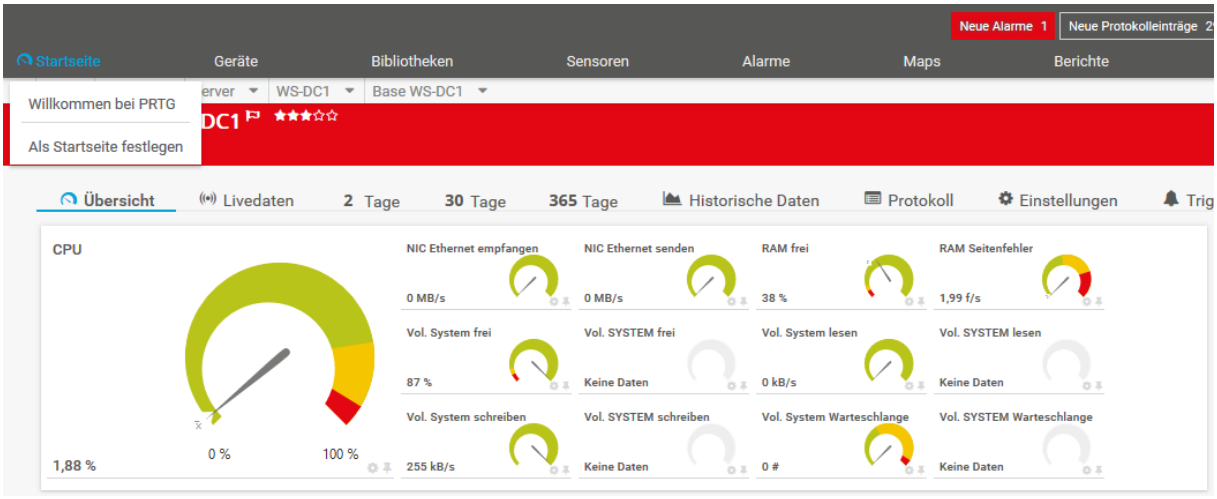
Und damit ist auch dieses Problem behoben:



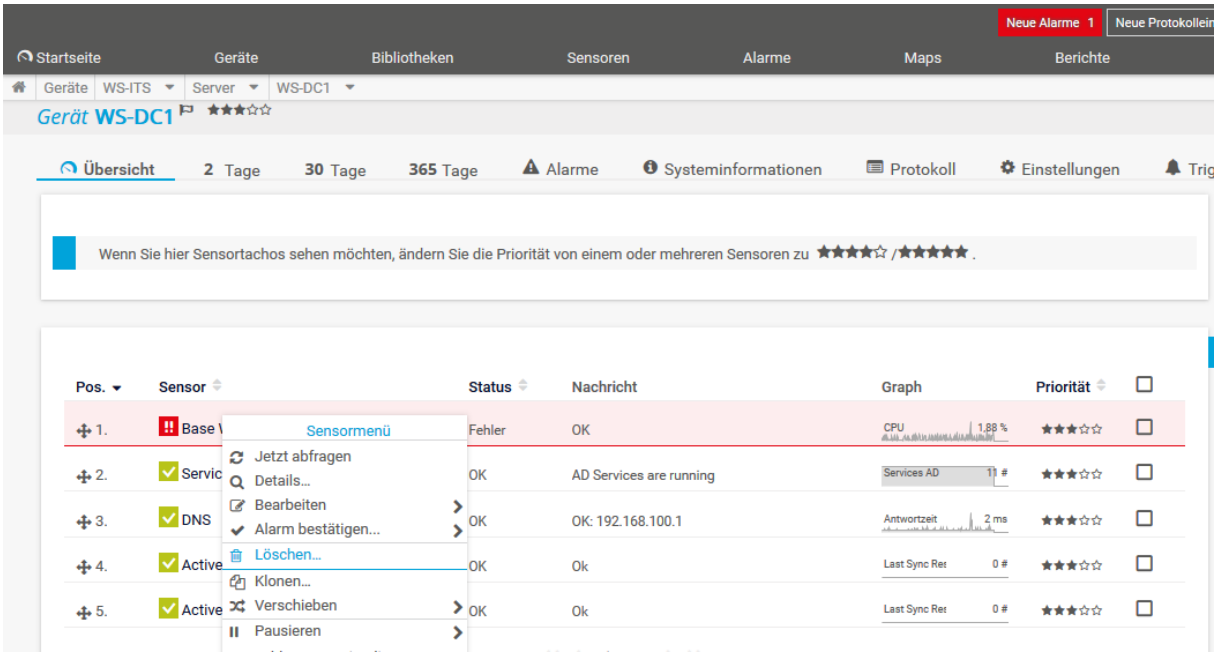
Und was sagt mein PRTG zum neuen WS-DC1? Nicht viel, denn der Server ist noch pausiert. Ich setze die Sensoren fort. Es dauert dann einige Sekunden:



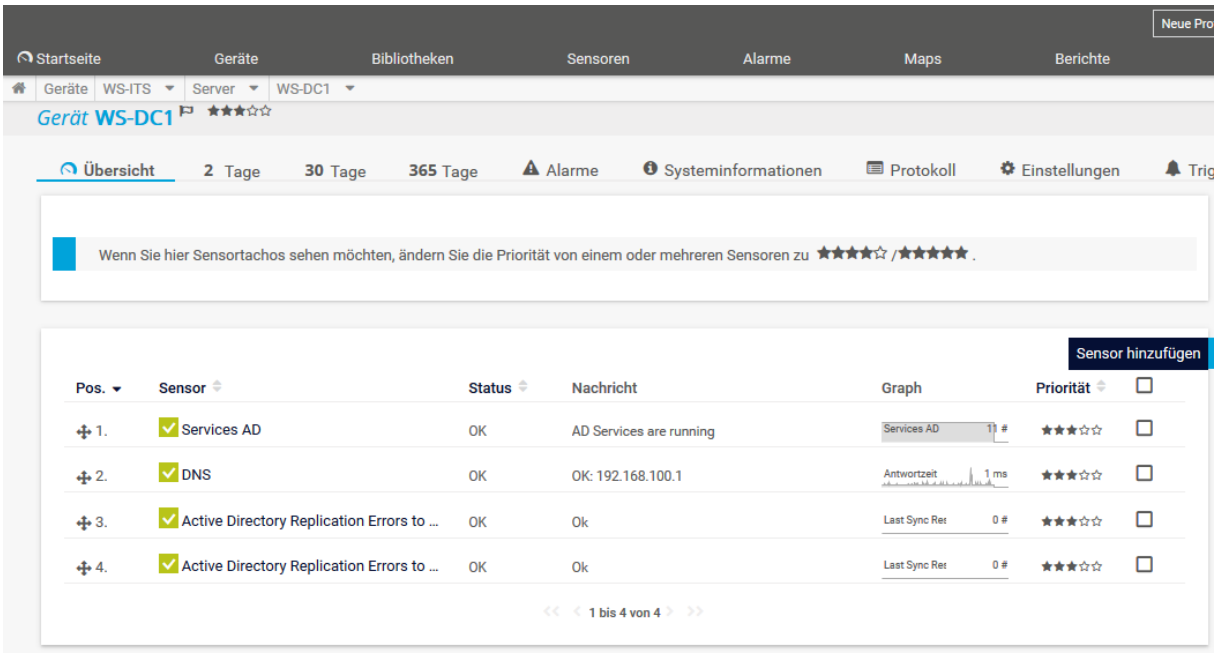
Dann tauchen Fehlermeldungen auf. Die Sensoren sind nicht 100% kompatibel:



Der Sensor ist ein von mir erstelltes PowerShell-Script. Da stimmt was mit den Historischen Daten nicht. Daher lösche ich den Sensor:

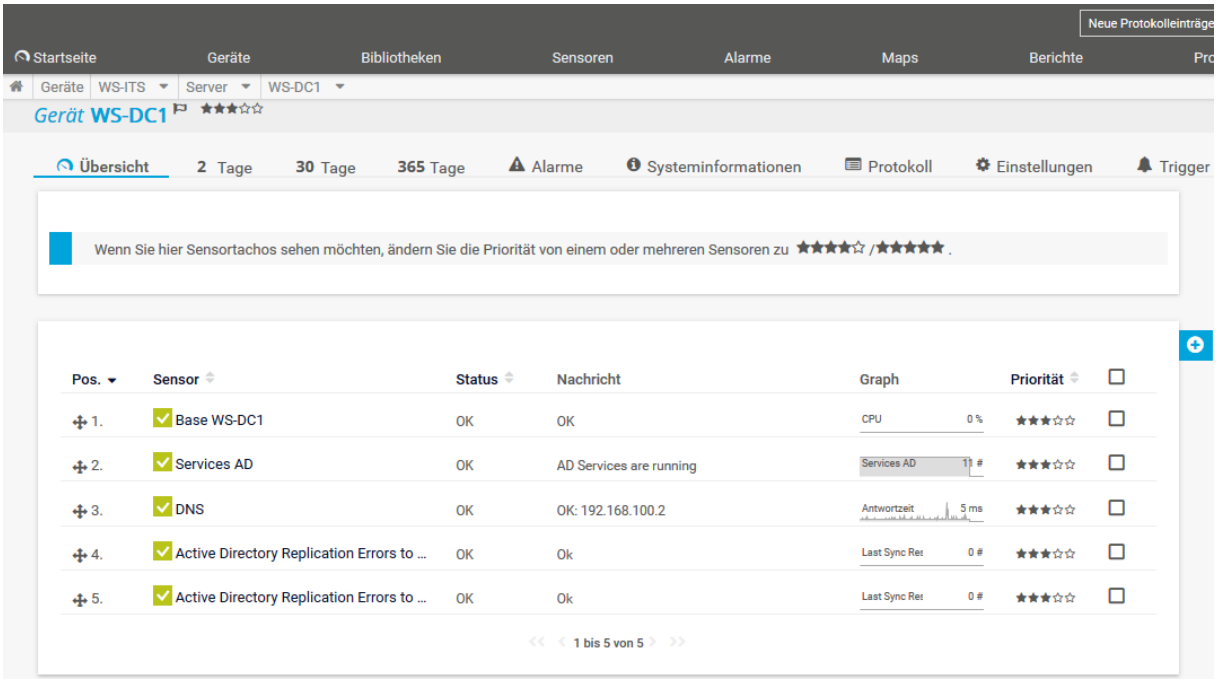


Anschließend kann ich ihn neu erstellen:



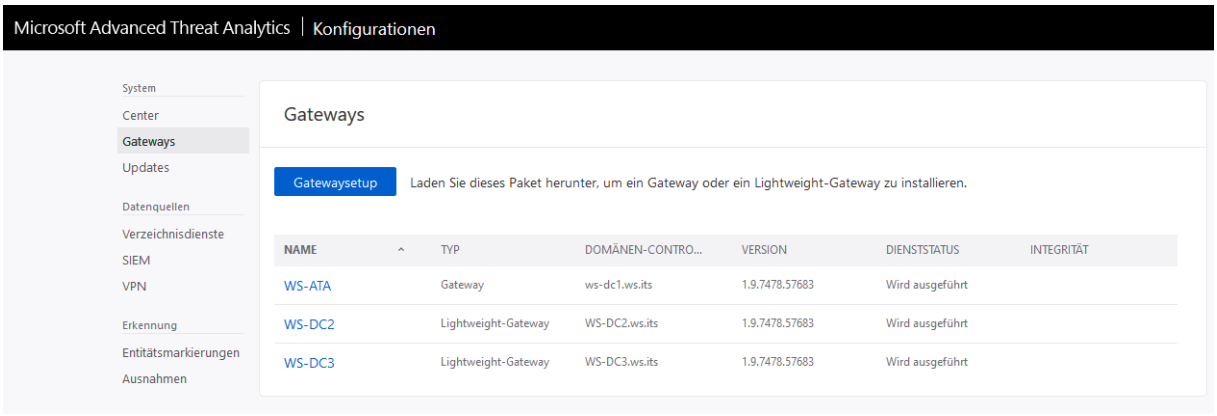
Ich editiere den Namen und wähle mein Sensor-Script für die Basisdaten eines Windows Servers aus:

Noch eine kurze Verschiebung nach oben und ein paar Sekunden warten zeigen wieder alles in einem frischen grün an:

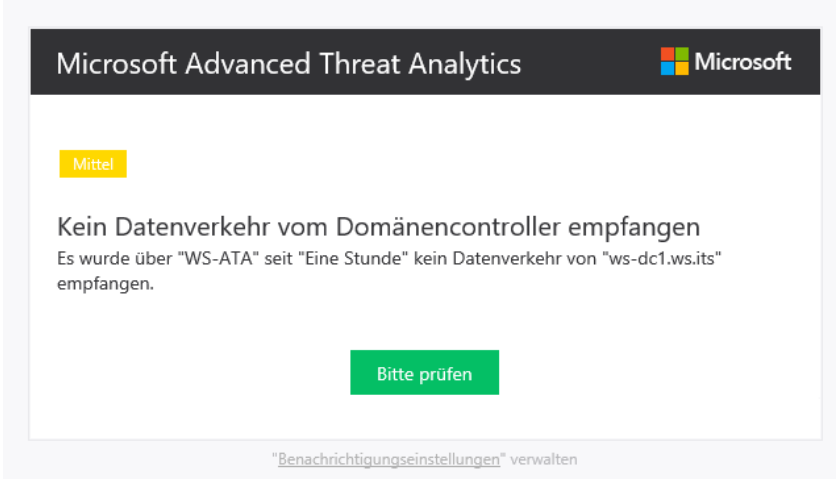


Integration ins ATA

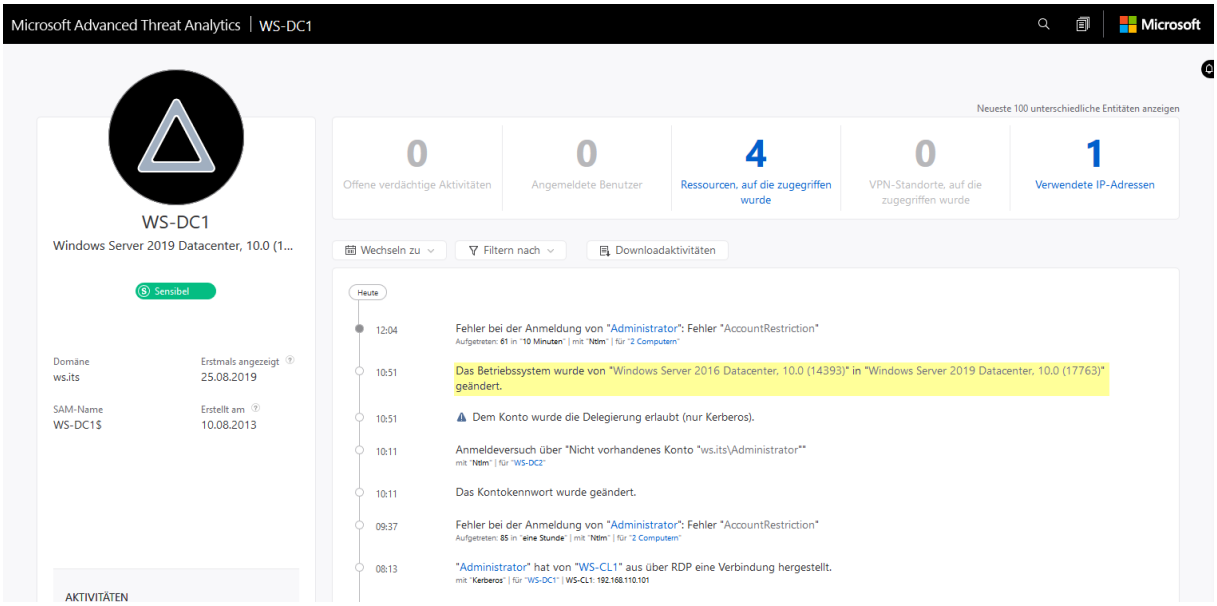
Mein ATA sollte eigentlich auch keine Probleme haben, da ich ja die neue Netzwerkkarte des neuen WS-DC1 bereits für die Datenspiegelung vorbereitet hatte. Und so zeigt es mir auch das Dashboard des ATA an: Alle Domain Controller sind erreichbar:



Zwischenzeitlich hatte ich aber eine Mail vom ATA erhalten. Der Wipe & Load- Vorgang blieb nicht unbemerkt:



Sehr interessant ist auch diese Meldung. Die musste ich aber im Computerobjekt im ATA suchen. Etwas Neugier schadet wohl nicht:



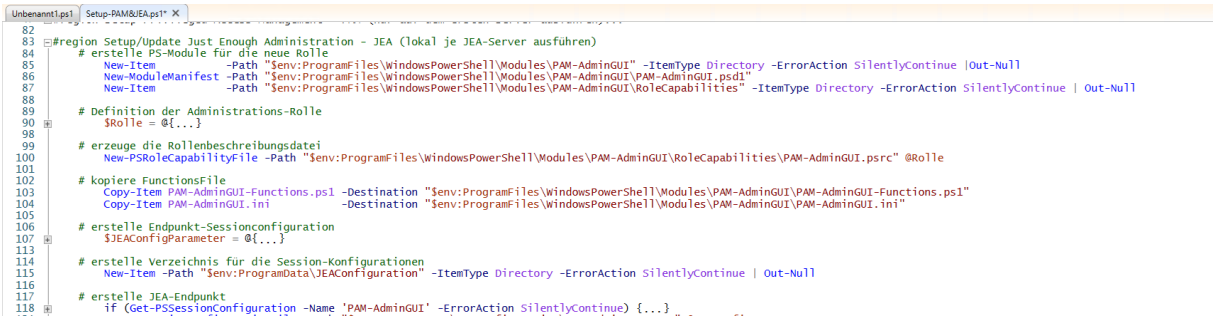
PowerShell JEA-PAM-AdminGUI

Final konfiguriere ich den neuen Server als JEA-Endpunkt für mein PAM-Script. Mit dieser selbstprogrammierten Lösung kann ich gesichert meine administrativen Accounts temporär in spezielle Berechtigungsgruppen aufnehmen. Dazu habe ich im Blog 2 weitere Artikel.

Die Installation ist denkbar einfach, denn ich habe dafür ein Script. Ich führe alle Zeilen der Region „Variablen“ aus:

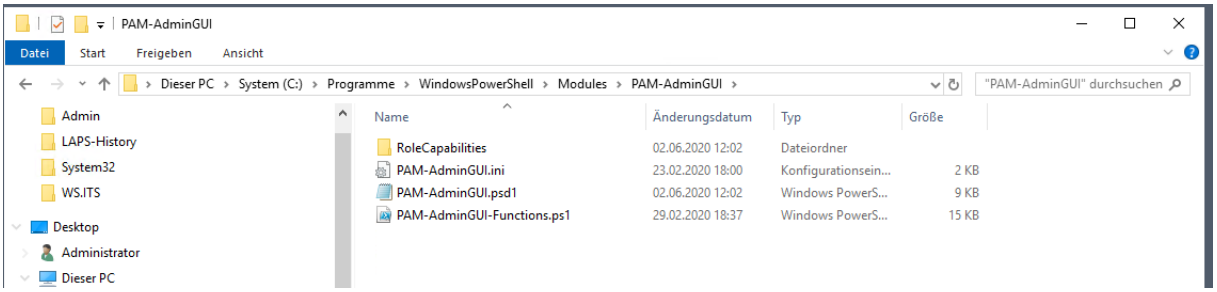
```
Unbenannt1.ps1 | Setup-PAM&JEA.ps1*
#### Scriptinfo ##########################################################################################
# Scriptreihe       PAM-AdminGUI\Setup-PAM&JEA
# Datum:            2020-02-29
# Version:          V1.10
# Programmierer:    Stephan Walther
##########################################################################################################

cd '\\ws.its\freigaben\AdminArea\Services\Active Directory\PAM-AdminGUI'

#region Variablen (immer ausführen)
    notepad.exe PAM-AdminGUI.ini
    pause

    Get-Variable -Scope global | Where-Object {'WS_OUAdminGruppen','WS_OUAdmins' -contains $_.name} | Remove-Variable -Scope global

    Get-Content -Path "PAM-AdminGUI.ini" |
        ForEach-Object {
            if ($_.length -gt 0) {
                if ('[' -notcontains ($_)[0]) {
                    if ($_ -like '*==*') {...}
                    if ($_ -like '*+=*') {...}
                }
            }
        }
#endregion
```

Und dann den Regionsblock „Setup/Update“:

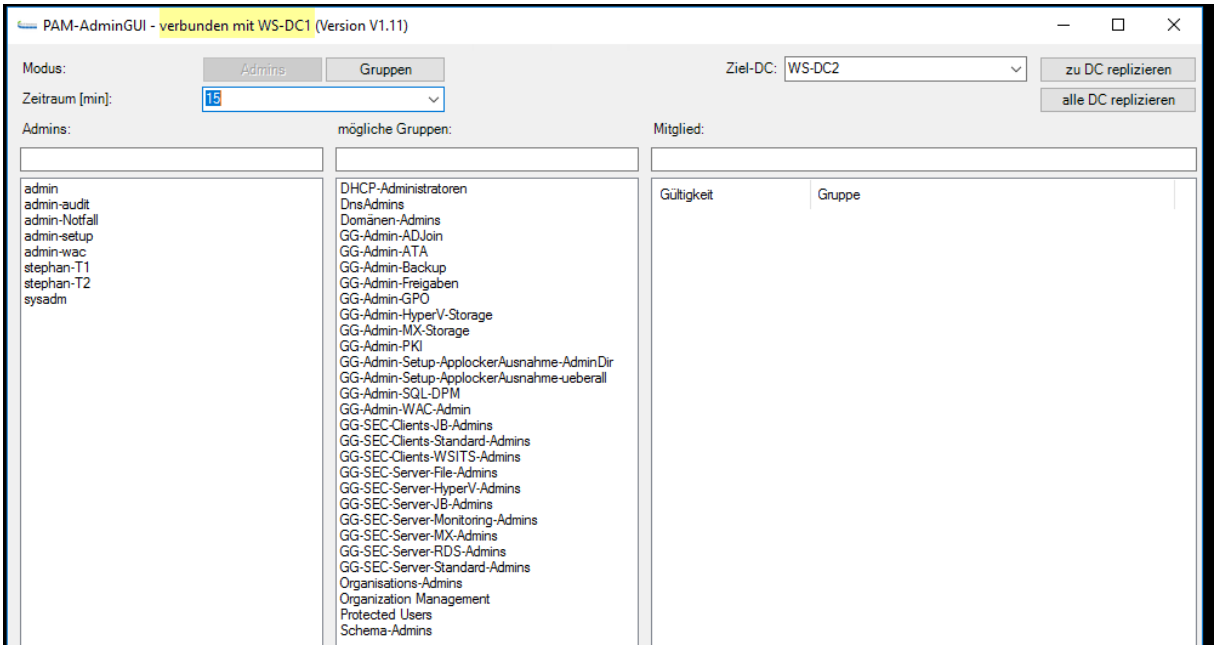


```
Unbenannt1.ps1 | Setup-PAM&JEA.ps1*

#region Setup/Update Just Enough Administration - JEA (lokal je JEA-Server ausführen)
    # erstelle PS-Module für die neue Rolle
        New-Item          -Path "$env:ProgramFiles\WindowsPowerShell\Modules\PAM-AdminGUI" -ItemType Directory -ErrorAction SilentlyContinue |Out-Null
        New-ModuleManifest -Path "$env:ProgramFiles\WindowsPowerShell\Modules\PAM-AdminGUI\PAM-AdminGUI.psd1"
        New-Item          -Path "$env:ProgramFiles\WindowsPowerShell\Modules\PAM-AdminGUI\RoleCapabilities" -ItemType Directory -ErrorAction SilentlyContinue | Out-Null

    # Definition der Administrations-Rolle
        $Rolle = @{...}

    # erzeuge die Rollenbeschreibungsdatei
        New-PSRoleCapabilityFile -Path "$env:ProgramFiles\WindowsPowerShell\Modules\PAM-AdminGUI\RoleCapabilities\PAM-AdminGUI.psrc" @Rolle

    # kopiere FunctionsFile
        Copy-Item PAM-AdminGUI-Functions.ps1 -Destination "$env:ProgramFiles\WindowsPowerShell\Modules\PAM-AdminGUI\PAM-AdminGUI-Functions.ps1"
        Copy-Item PAM-AdminGUI.ini           -Destination "$env:ProgramFiles\WindowsPowerShell\Modules\PAM-AdminGUI\PAM-AdminGUI.ini"

    # erstelle Endpunkt-Sessionconfiguration
        $JEAConfigParameter = @{...}

    # erstelle Verzeichnis für die Session-Konfigurationen
        New-Item -Path "$env:ProgramData\JEAConfiguration" -ItemType Directory -ErrorAction SilentlyContinue | Out-Null

    # erstelle JEA-Endpunkt
        if (Get-PSSessionConfiguration -Name 'PAM-AdminGUI' -ErrorAction SilentlyContinue) {...}
```

So werden die erforderlichen Dateien erstellt und im Windows Remote Management regirstriert:

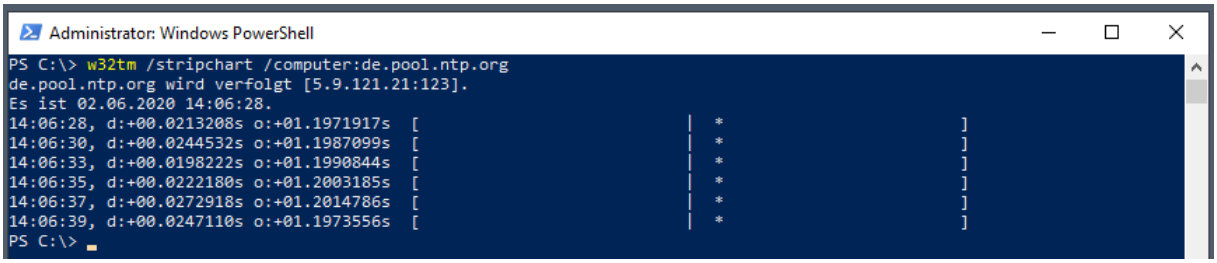


Ich starte das clientseitige Frontend. Die PowerShell rendert mir eine hübcshe GUI und verbindet sich mit dem neuen WS-DC1. Die Einrichtung war erfolgreich:

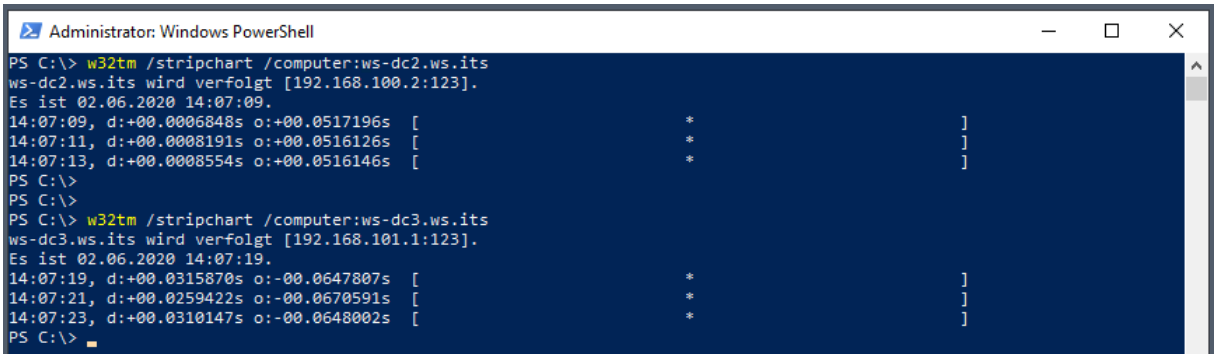


TroubleShooting des Zeitservers

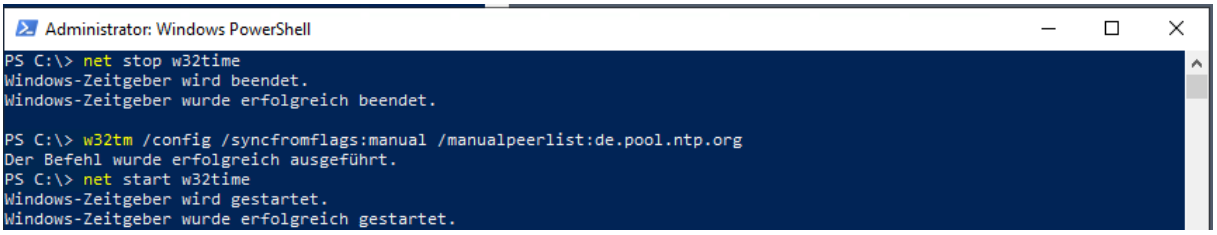
Diese Funktion wird gerne vergessen: korrekte Zeiten sind für das Authentifizierungsprotokoll enorm wichtig. Der neue Server ist mein PDC-Emulator. Daher muss er selber über eine geeignete Zeitquelle verfügen. Ohne diese gibt es schnell Abweichungen. Mit w32tm /stripchart kann ich die Differenz zu einem öffentlichen NTP-Server auswerten. Der Stern liegt nicht in der mitte zwischen den eckigen Klammern. Die Zeiten sind nicht synchron:



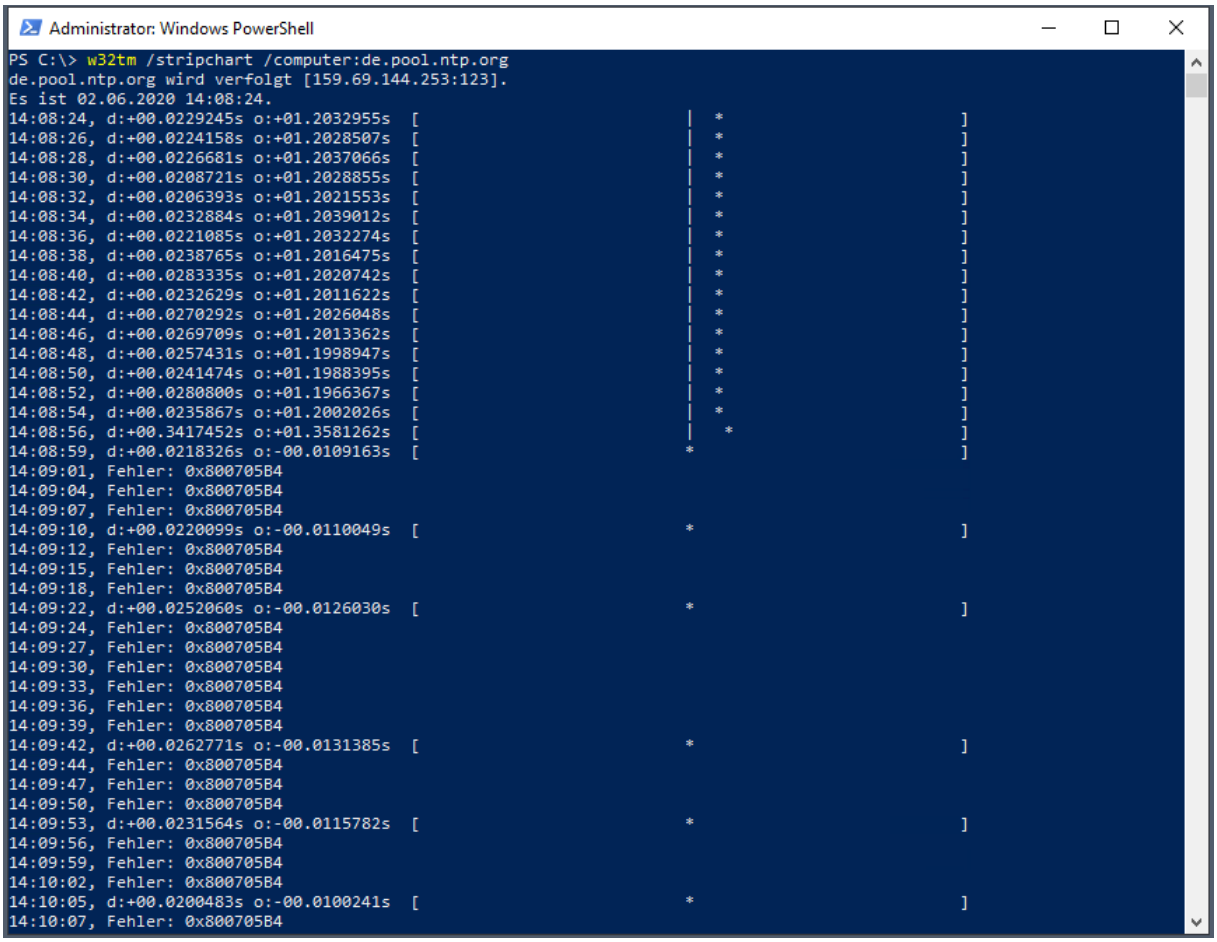
Die anderen Domain Controller holen sich ihre Uhrzeit beim PDC. Das funktioniert ohne Konfiguration. Meine beiden anderen DCs sind synchron mit der Zeit vom WS-DC1. Alle weichen gemeinsam ab:



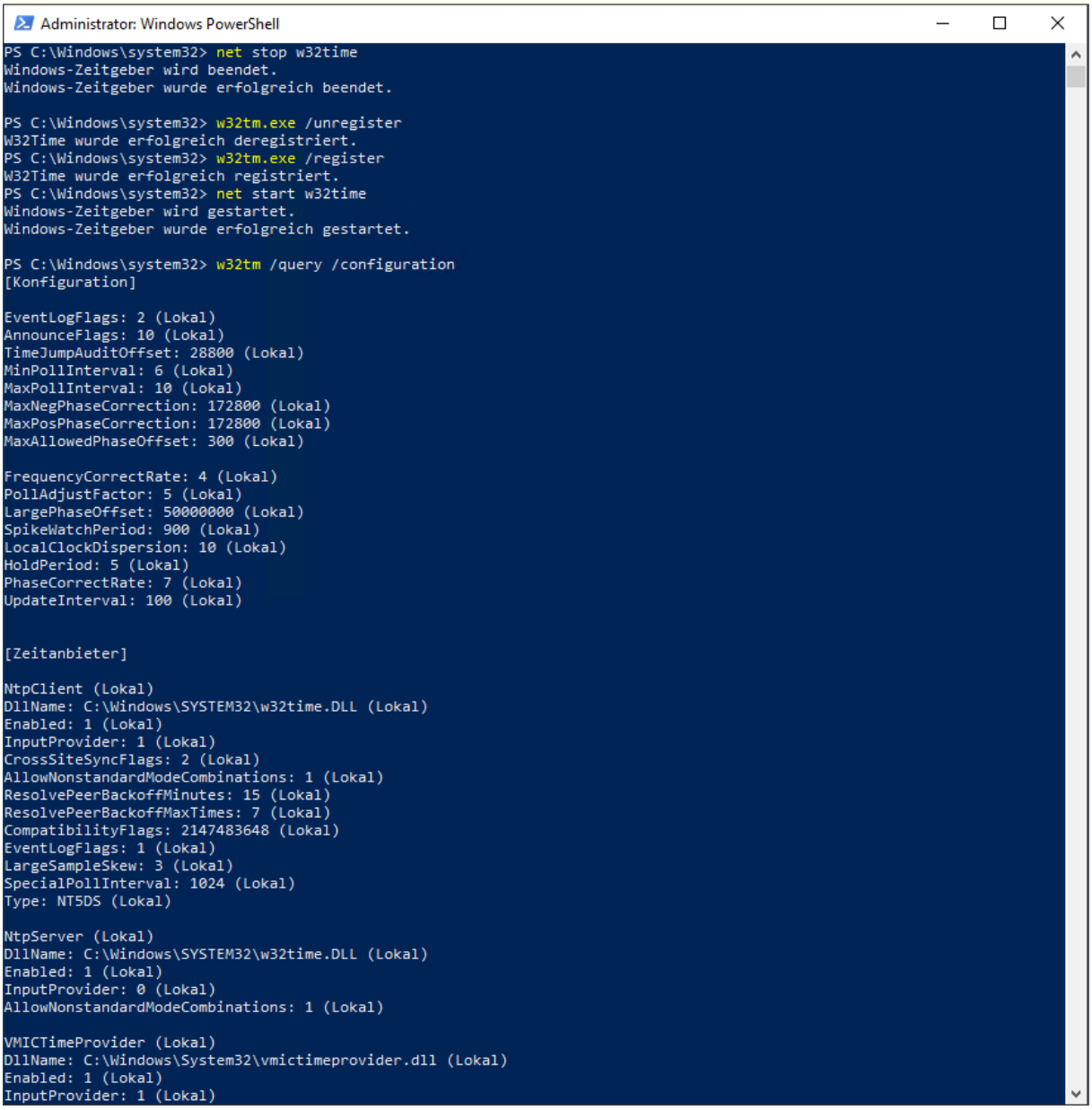
Ich konfiguriere den Zeitservice und trage einen öffentlichen NTP-Server statisch ein:



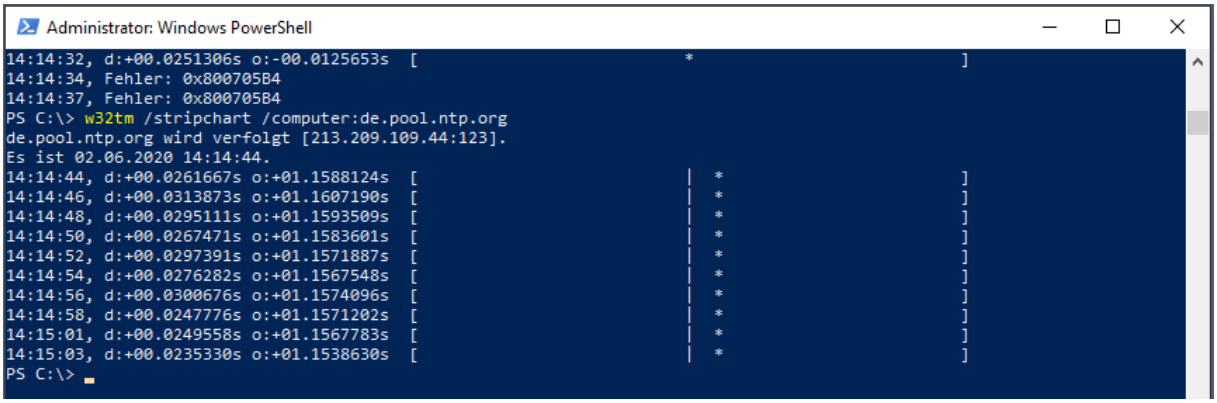
Dann prüfe ich wieder die Abweichungen. Einige Sekunden passiert nichts. Und dann ist die Zeit vom neuen Server synchron. Leider werden hier viele Fehler ausgegeben:



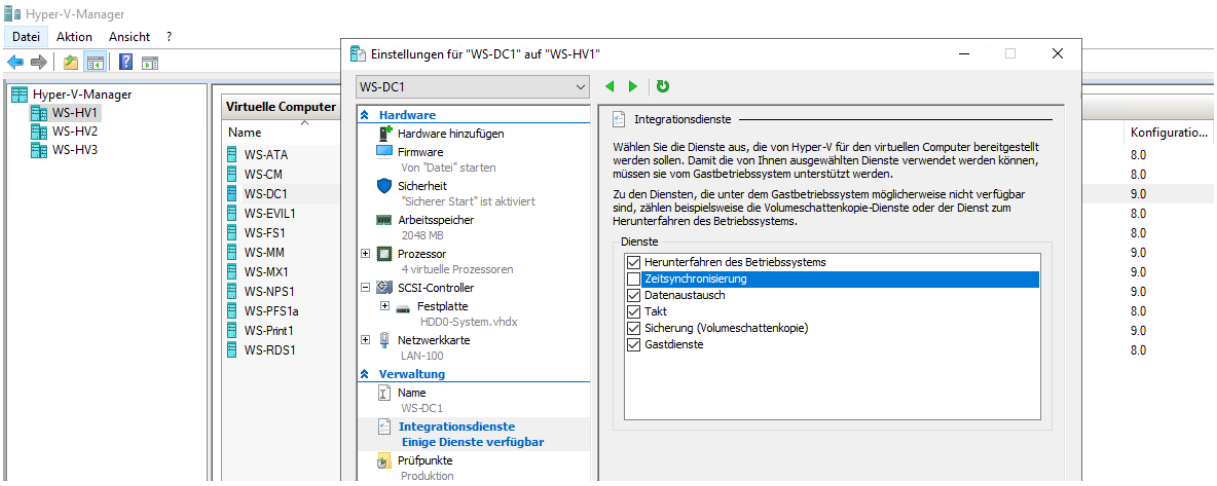
Vielleicht hat das etwas bei der Konfiguration Probleme verursacht? Ich setze die Einstellungen zurück:



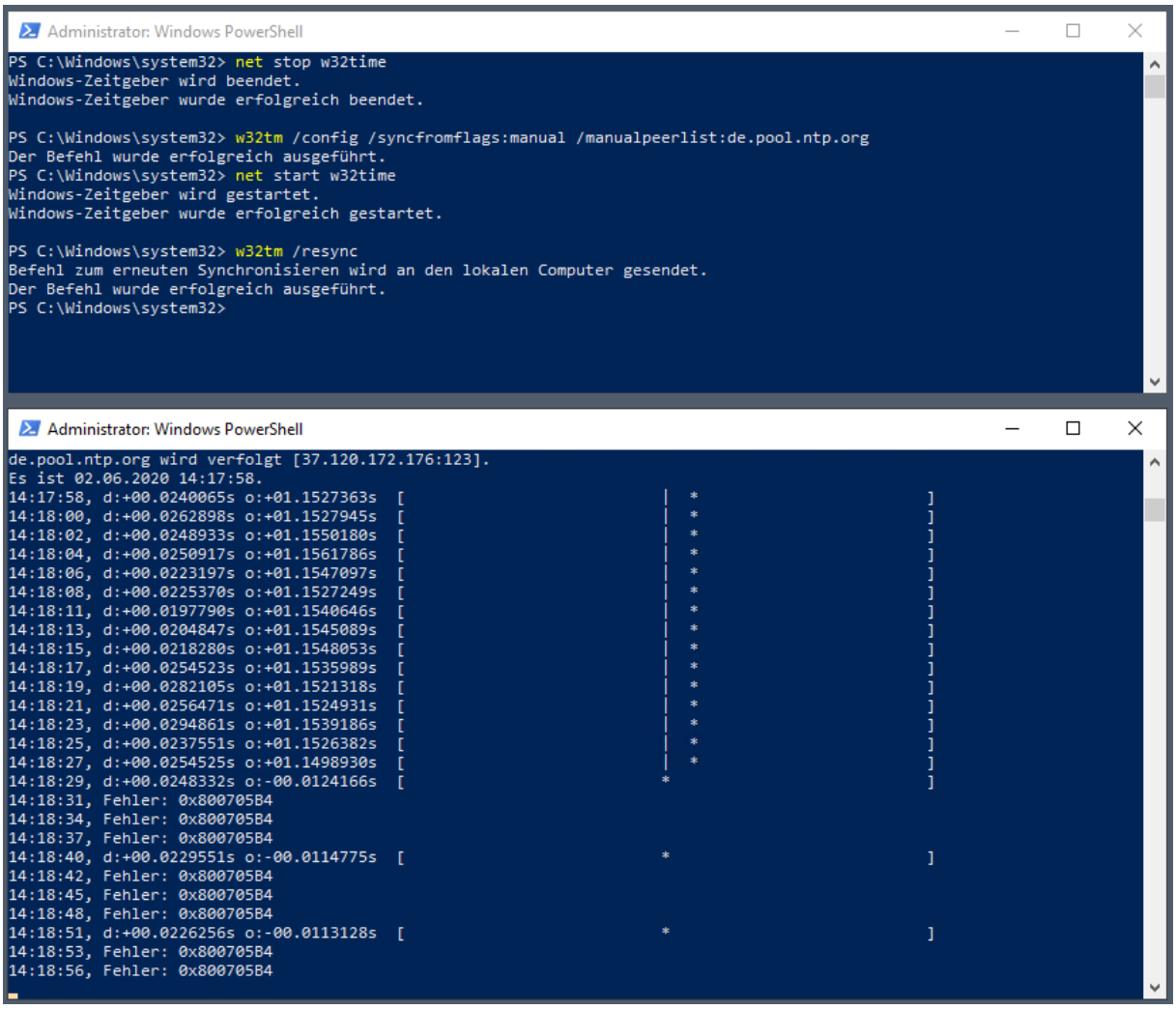
Sofort ist die Zeit wieder verschoben:



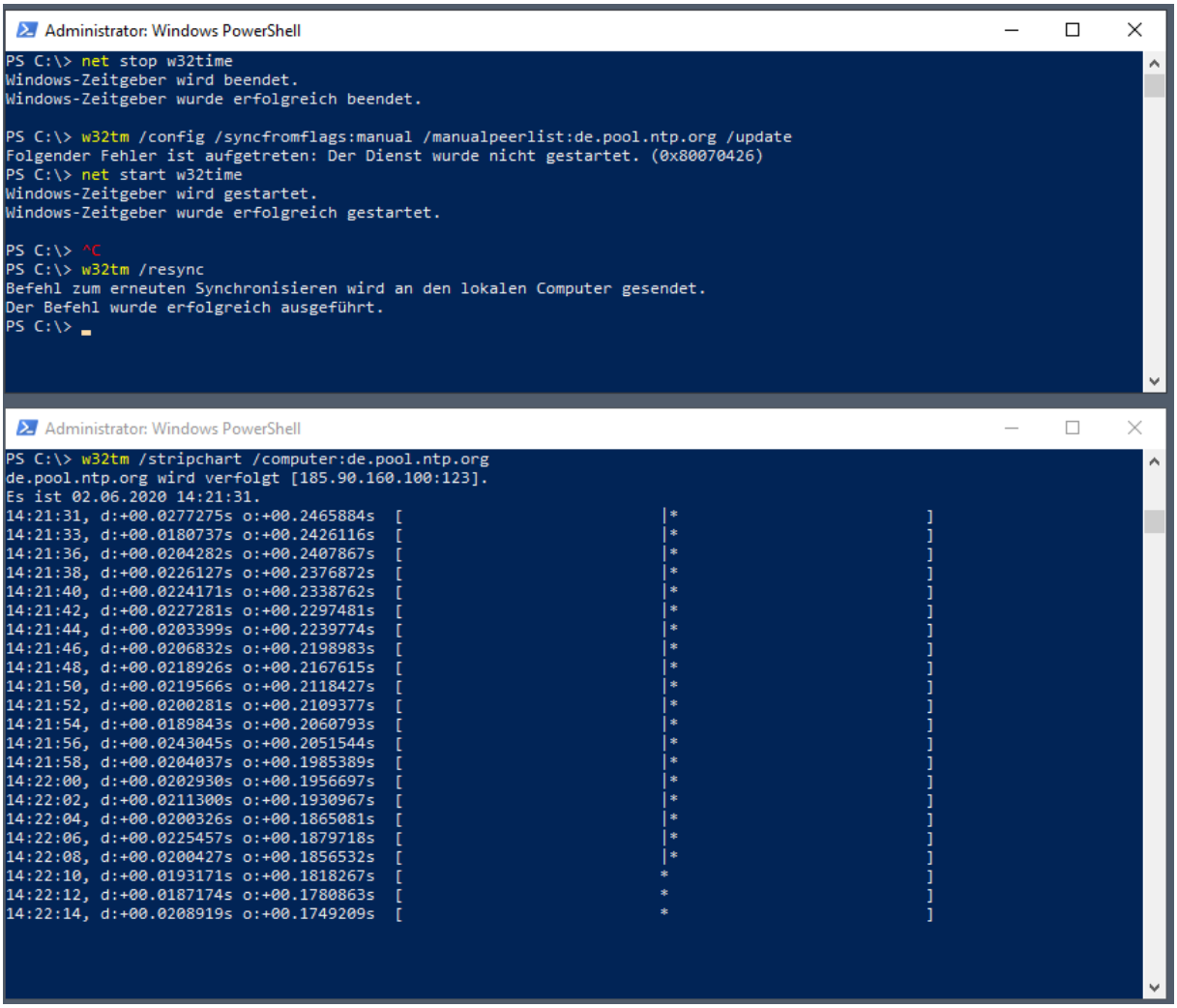
Das kann nur eines bedeuten: Der Server WS-DC1 hat noch einen anderen Zeitgeber! Wer in den ersten Bildern dieses Beitrages genau hingeschaut hat, wird das Problem schnell erkennen. Der Übeltäter ist mein Hyper-V-Host. Dieser gibt seine Systemzeit an den virtuellen WS-DC1 weiter. Blöderweise holt er sich als Domain Member von einem DC seine eigene Zeit ab… Die Einstellung kann ich in den Einstellungen der virtuellen Maschine sehen. Ich entferne den Haken:



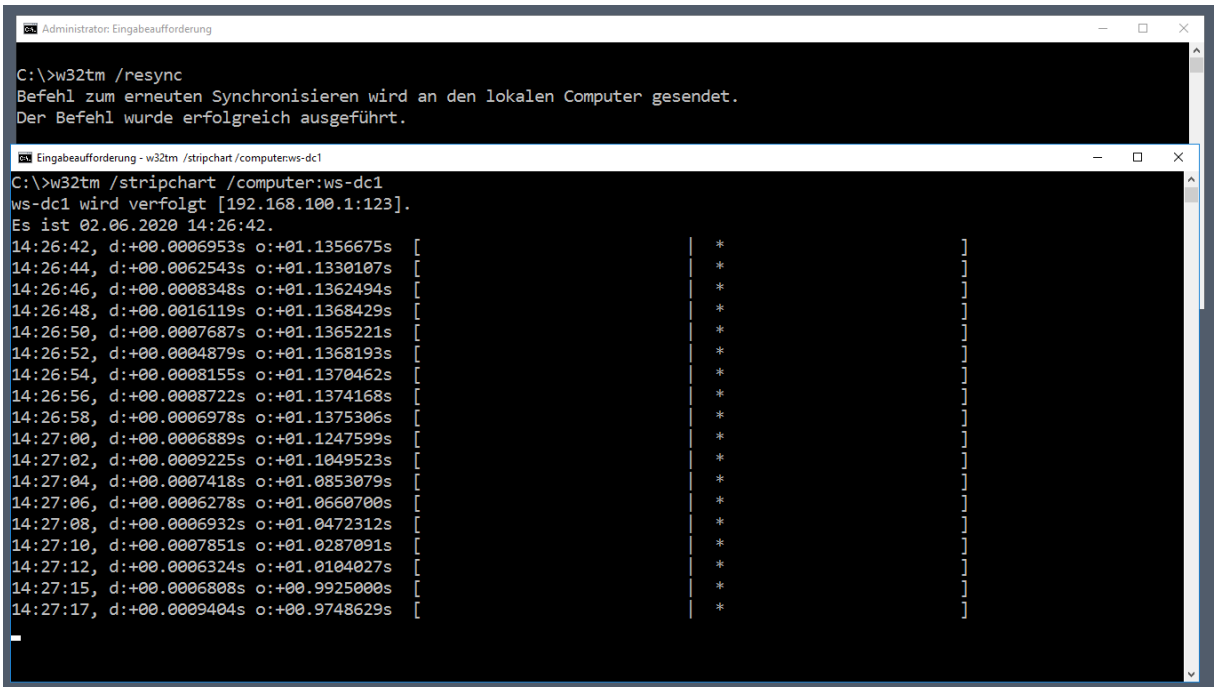
Jetzt wiederhole ich die Zeitkonfiguration. Anschließend prüfe ich wieder die Differenz zur öffentlichen Systemzeit:



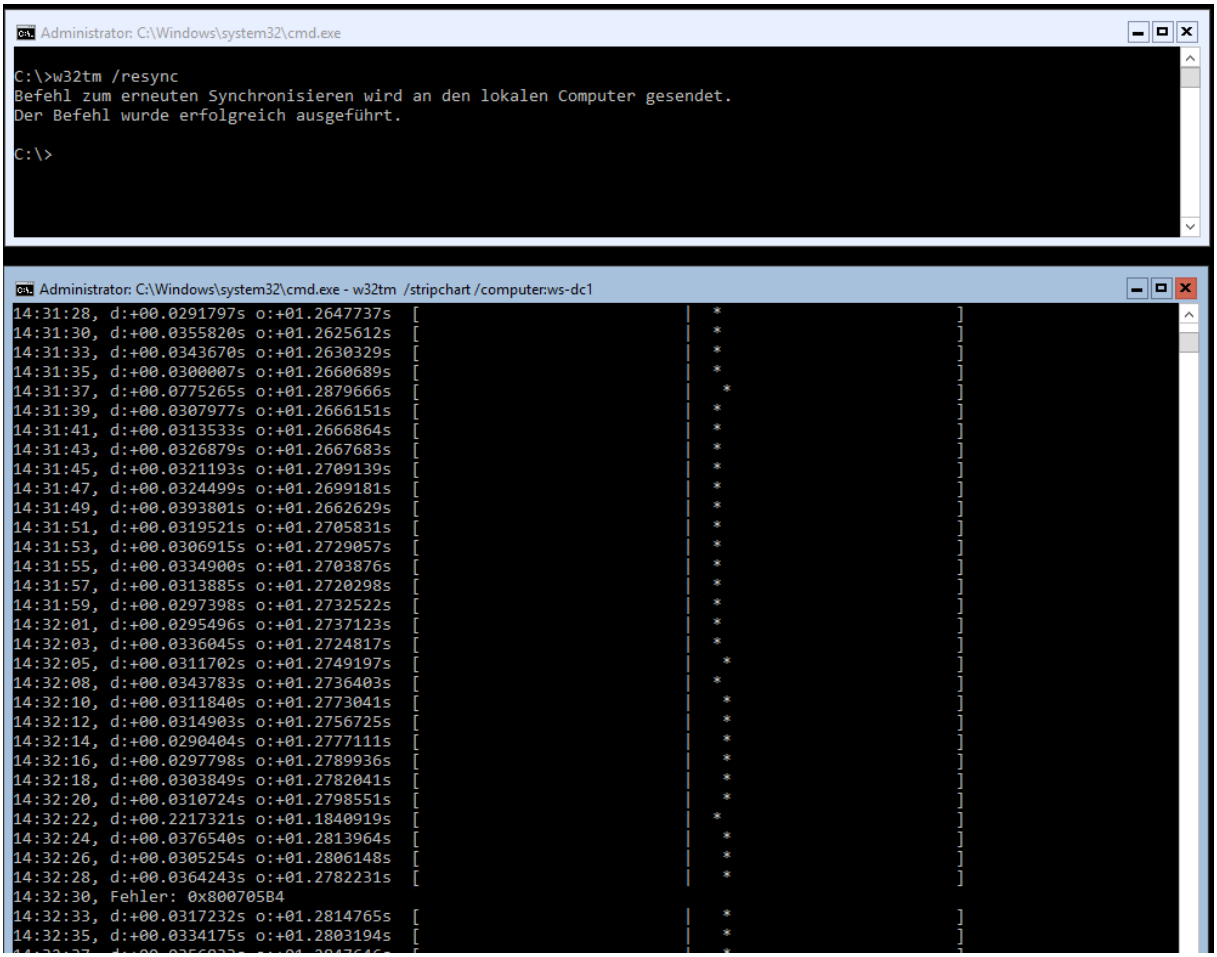
Mmh, das ist leider der gleiche Fehler wie vorher. Daher nehme ich bei der Konfiguration noch eine Option /Update dazu:



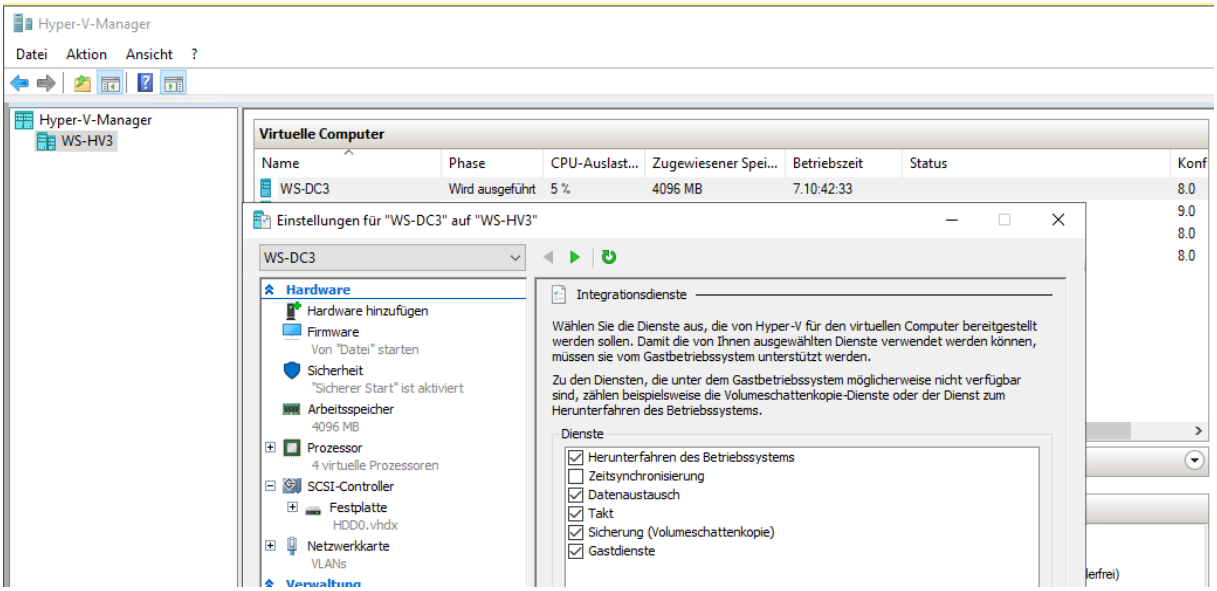
Das wars. Jetzt ist der Server synchron. Und jetzt holen sich die anderen Domain Controller die korrekte Zeit. Das kann einen Moment dauern, da die Zeit schrittweise angepasst wird:



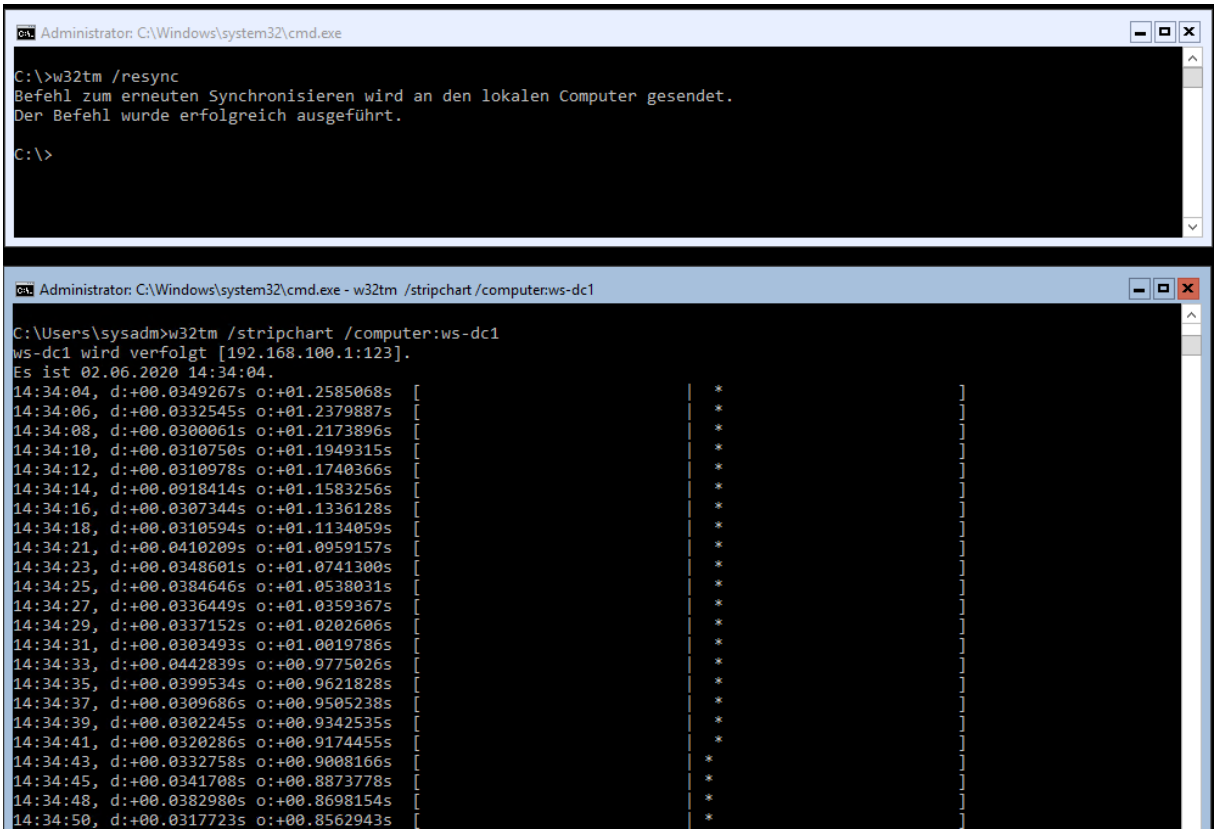
Und auch mein WS-DC3 im anderen Standort gleicht sich an. Hier erkennt man sehr gut die zwei kollidierenden Zeitquellen PDC und Hyper-Visor:



Ich rekonfiguriere also die Zeiteinstellung der virtuellen Maschine:

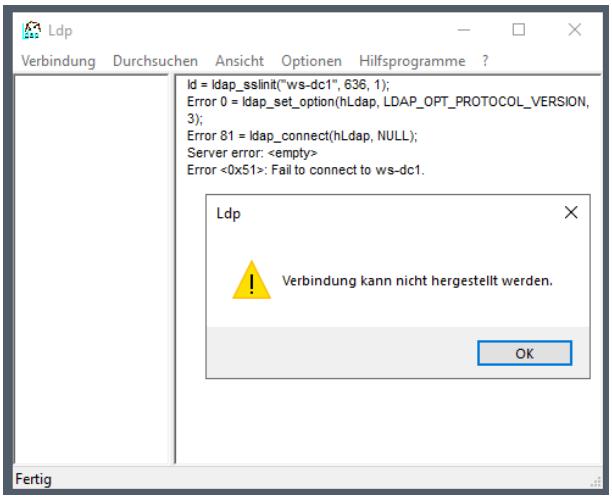


Und nahc wenigen Minuten ist die Uhrzeit synchron:

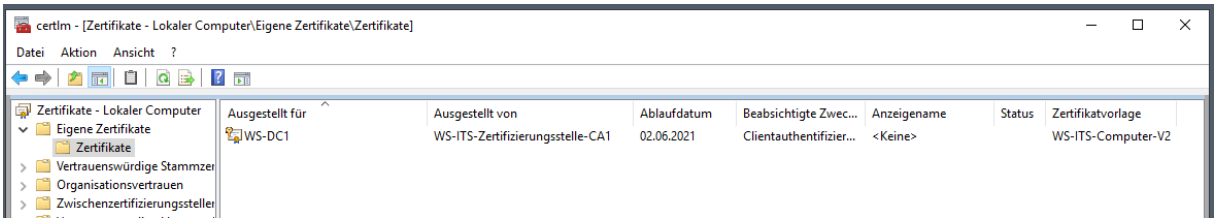


TroubleShooting LDAPS

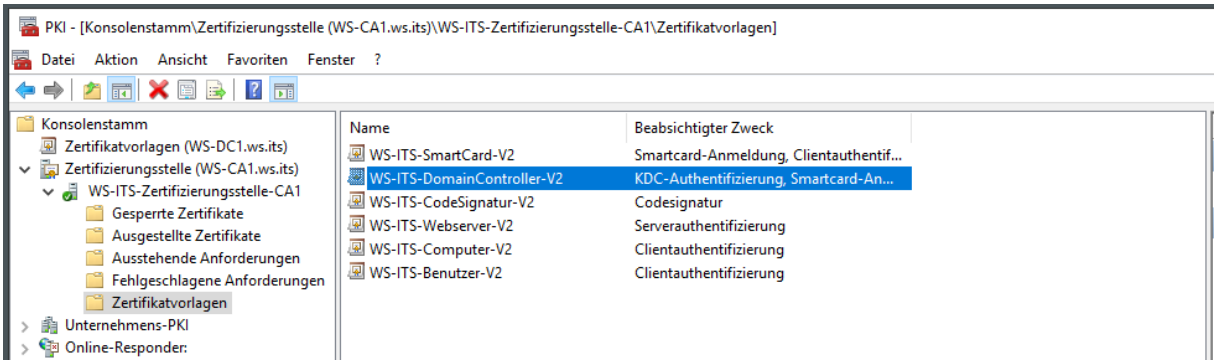
Alle Domain Controller sollten das sichere LDAPS beherrschen. Das kann recht einfach mit LDP.exe getestet werden. Mein neuer Domain Controller hat wohl noch Probleme:



Das eigentlich Gemeine an dem Problem: Es fällt zunächst nicht auf, wenn noch ein anderer Domain Controller LDAPS anbietet. Die Clients weichen einfach aus. Aber wenn dieser Domain Controller dann mal nicht erreichbar ist bzw. auch migriert wird, dann knallt es richtig. Die Ursache ist einfach: Der DC benötigt ein Zertifikat für LDAPS. Aber er hat kein passendes im Speicher:

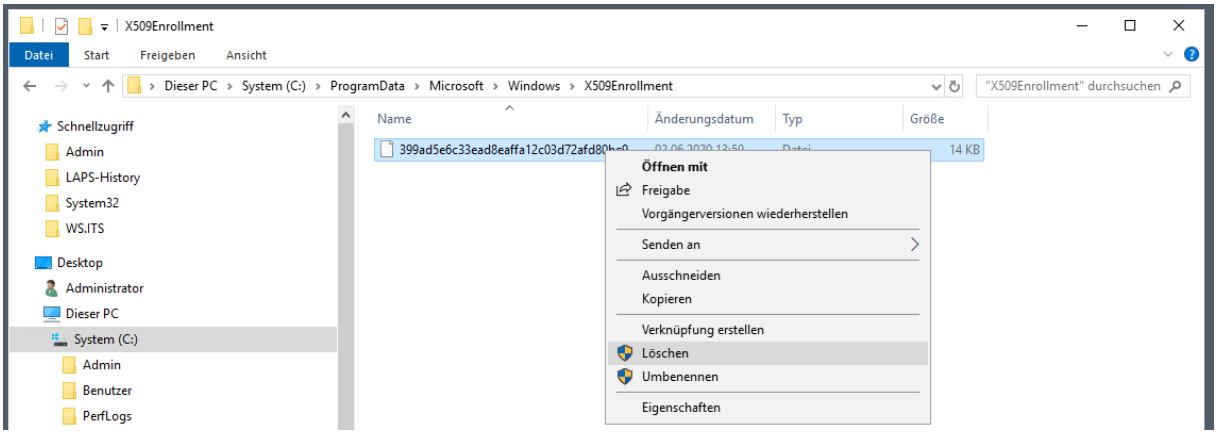


Merkwürdig. Eigentlich sollte er sich von meiner internen PKI ein Zertifikat basierend auf dieser Vorlage automatisch beziehen:

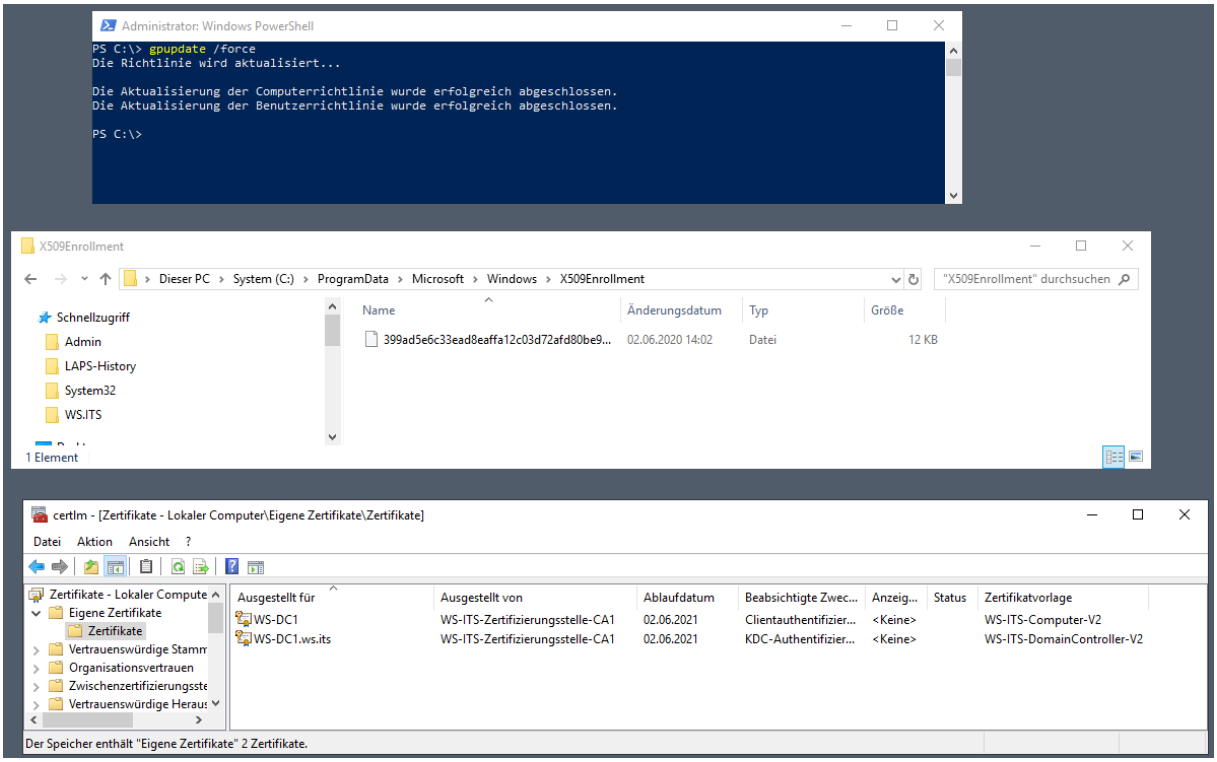


Ich habe vor einer Weile an den Vorlagen der Zertifikate herumgespielt. Und die Verteilung läuft über einen CEPCES. Das ist ein Webservice, der als Vermittler zwischen Client und Zertifizierungsstelle steht und gesichert über https angesprochen werden kann. Der CEP merkt sich dabei gerne die verfügbaren Informationen der Zertifizierungsstelle in einem Cache. Der sollte eigentlich alle 30 Minuten aktualisiert werden. Ich helfe mal durch einen iisreset nach (der CEPCES läuft natürlich in einem Windows Internet Information Service):

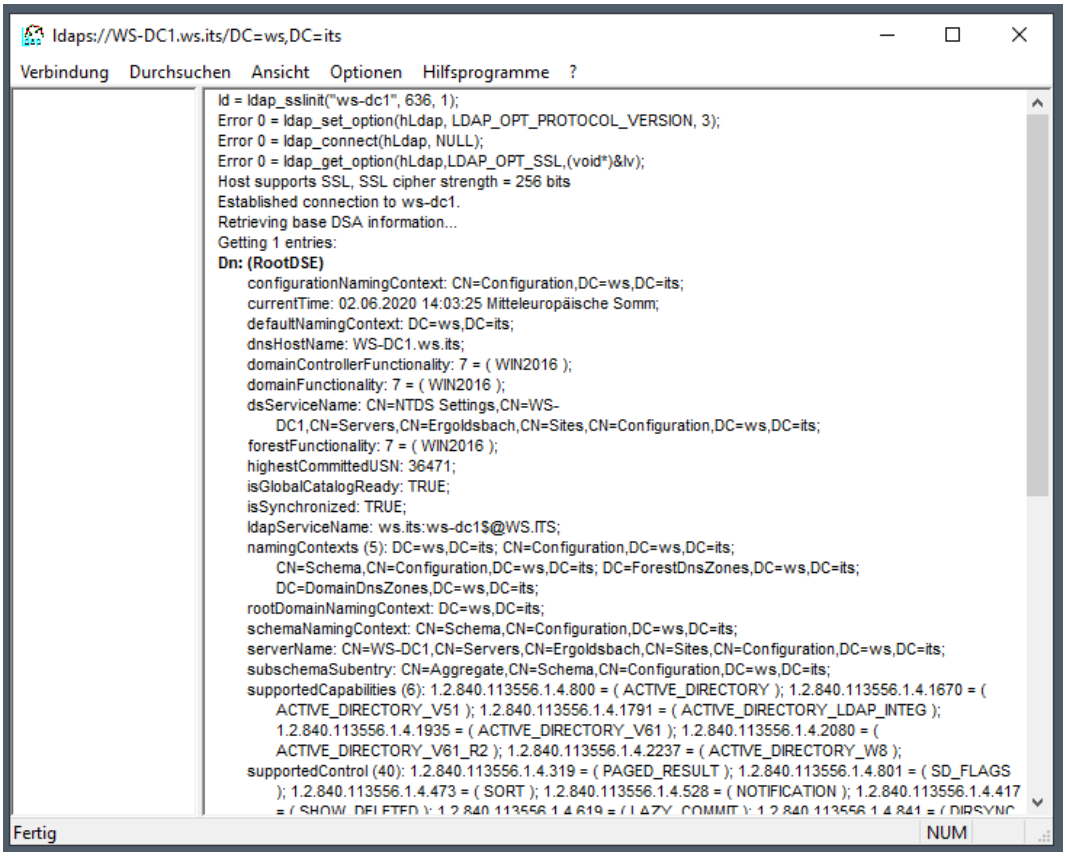
Auch der Client – hier ist es mein Domain Controller – merkt sich die letzte CEP-Antwort in einer Cache-Datei. Diese lösche ich. Dazu sind administrative Rechte erforderlich:



Dann versuche ich es erneut. Das AutoEnrollment meiner Zertifikate wird alle 8 Stunden angetriggert. Aber auch ein gpupdate kann da nachhelfen. Der Domain Controller kontaktiert den CEPCES, lädt sich die Infomationen in seine Cache-Datei herunter, erkennt, dass er ein neues Zertifikat bekommen soll und startet automatisch das Enrollment:



Das sollte eigentlich von alleine passieren. Aber wie so oft gilt bei Automatisierungen: Vertrauen ist gut, Kontrolle ist besser. Jetzt kann der neue Domain Controller auch LDAPS-Anfragen bedienen:



Zusammenfassung

Mein Active Directory arbeitet jetzt mit dem ersten Windows Server 2019. Funktional hat sich nichts mehr seit dem Windows Server 2016 verändert. Aber ich bin wieder einen Schritt weiter in Richtung Zielgerade. Die verbliebenen Altserver lassen sich nun an einer Hand abzählen.

Wie man sehen konnte, ist der Umstellungsprozess nicht sonderlich schwer. Dennoch gibt es eine Vielzahl von Stolpersteinen und Problemszenarien, auf die man sich vorbereiten sollte.