Inhalt

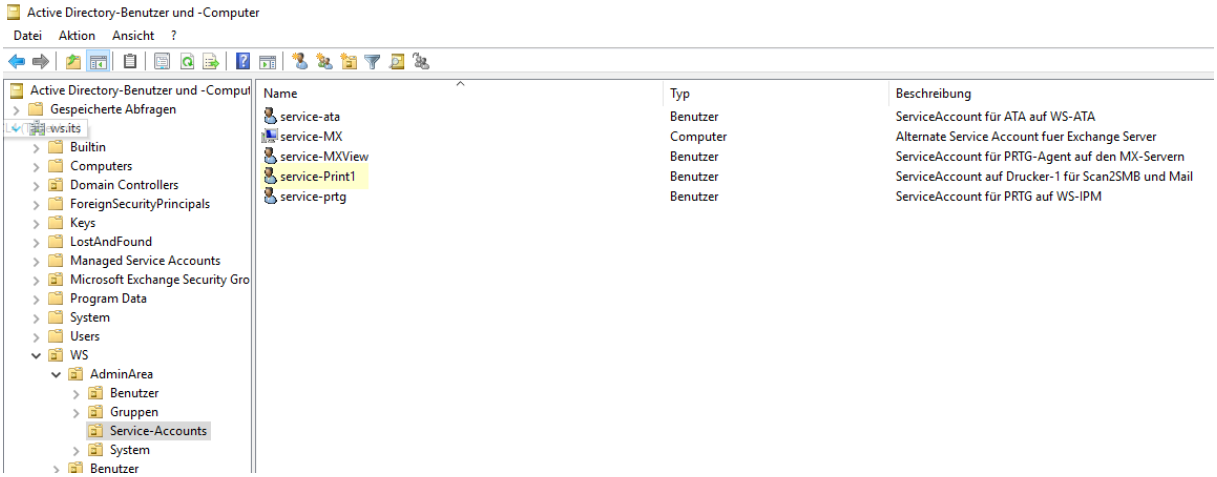
Zielsetzung

aktuelle Situation

Nachdem mein alter Laserdrucker seinen wohlverdienten Ruhestand angetreten hat, musste ein Ersatz her. Da ich mehr einscanne als Drucke, war mir bei diesem Kauf die Scan-Option sehr wichtig. Das neue Gerät beherrscht die üblichen Funktionen wie Scan2Mail und Scan2SMB. Bisher hatte ich nur eine Scan2USB-Funktion.

Das Problem

Mein aktueller Druckserver läuft auf einem meiner Fileserver nebenbei mit. Ich habe das neue Multifunktionsgerät eingerichtet und wollte es mit einer Freigabe verbinden. Dabei wurden ein Benutzername und ein Passwort abgefragt. Den Account habe ich fix als Service Account erstellt und dann gingen meine Tests los.



Leider konnte sich der Drucker nicht mit dem Freigabeserver verbinden. Die erste Meldung lautete immer „Benutzername oder Passwort falsch“. Das konnte es aber nicht sein. Die Anmeldeinformationen habe ich mit Copy & Paste eingetragen.

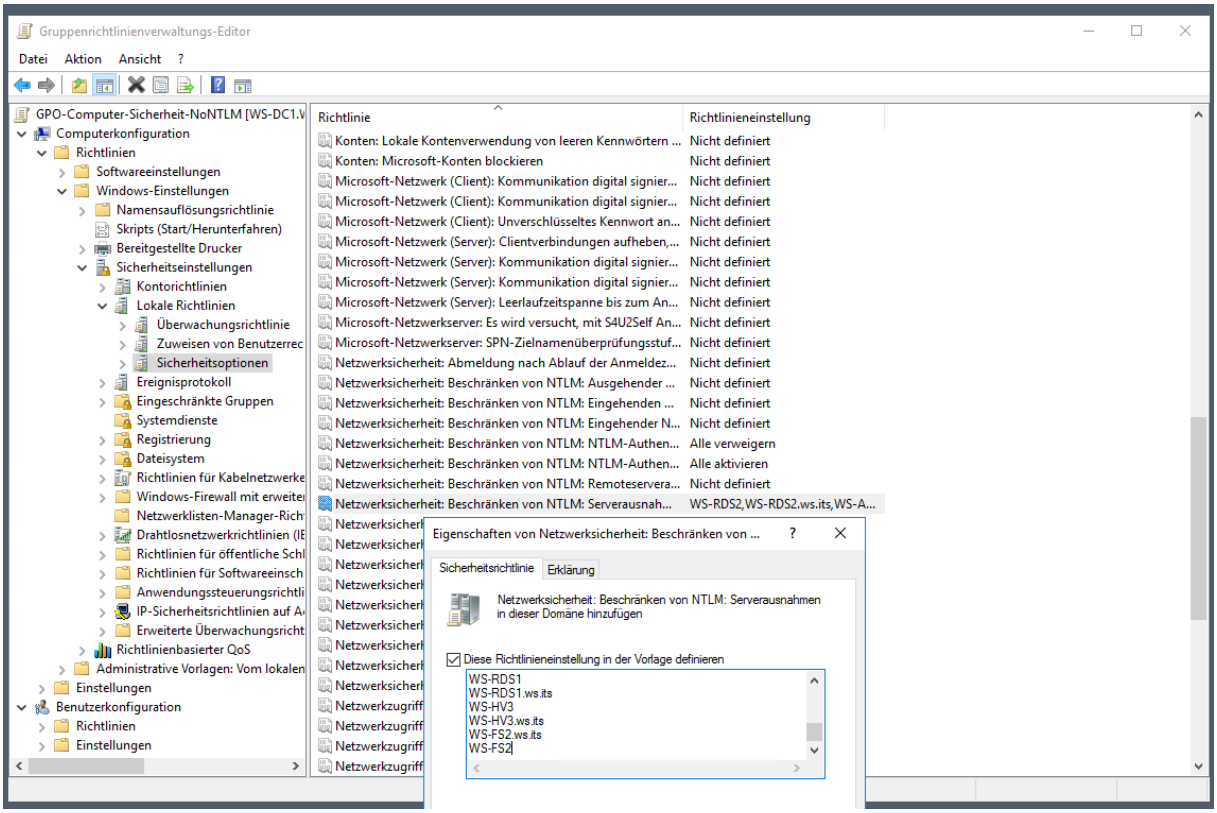
Ein Netzwerkproblem bzw. ein Firewall-Problem kann ich ausschließen: Der Drucker steht im gleichen Netzwerksegment wie mein Fileserver und ich kann das Gerät über das Webinterface administrieren.

Die Ursache liegt demnach bei der Authentifizierung. In meiner Infrastruktur habe ich NTLM generell deaktiviert. Nur ausgewählte Server dürfen das alte Verfahren verwenden. Für das Troubleshooting habe ich natürlich die Protokollierung aktiviert. Mit einem PowerShell-Script kann ich alle meine Domain Controller nach fehlgeschlagenen NTLM-Verbindungen durchsuchen. Und hier sehe ich die Kennung des neuen Service Accounts:

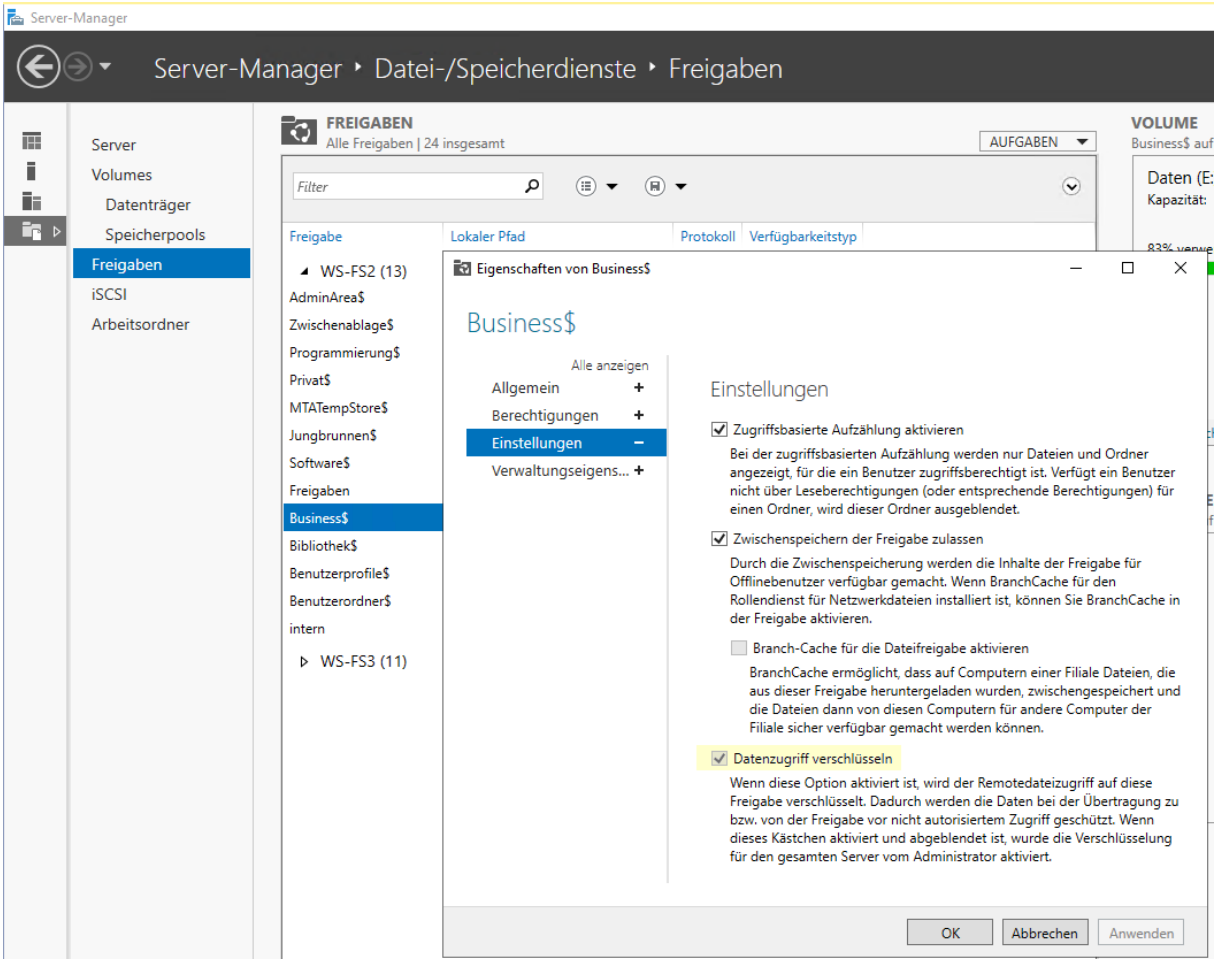
```
cls;
Invoke-Command -ComputerName (Get-ADDomain).ReplicaDirectoryServers -ScriptBlock {
    Get-WinEvent -Path C:\Windows\System32\Winevt\Logs\Microsoft-Windows-NTLM%4Operational.evtx -MaxEvents 20 -ErrorAction SilentlyContinue |
        Select-Object -Property @{ n='DC'       ; e={ $env:COMPUTERNAME                                                    } },
                @{ n='Datetime' ; e={ (Get-Date -Date $_.TimeCreated -Format u) -replace 'z'                               } },
                @{ n='Client'   ; e={ (($_.Message -split "`n" | select-string 'Arbeitsstationsname') -split ':')[1].trim()       } },
                @{ n='Server'   ; e={ (($_.Message -split "`n" | select-string 'Name des sicheren Kanals') -split ':')[1].trim() } },
                @{ n='Domain'   ; e={ (($_.Message -split "`n" | select-string 'Domänenname') -split ':')[1].trim()         } },
                @{ n='User'     ; e={ (($_.Message -split "`n" | select-string 'Benutzername') -split ':')[1].trim()          } }
} | Sort-Object -Property Datetime |
    Format-Table -Property DC,Datetime,Client,Server,Domain,User
```

```
DC     Datetime            Client    Server    Domain User
--     --------            ------    ------    ------ ----
WS-DC3 2020-05-22 04:00:14 WS-DC3    WS-FS3    ws.its service-ata
WS-DC2 2020-05-22 04:04:48 WS-DC2    WS-MON    ws.its service-ata
WS-DC2 2020-05-22 04:22:23 WS-DC2    WS-CA1    ws.its service-ata
WS-DC2 2020-05-22 04:30:01 WS-DC2    WS-HV1    ws.its service-ata
WS-DC1 2020-05-22 04:30:01 WS-DC2    WS-HV2    ws.its service-ata
WS-DC2 2020-05-22 04:30:02 WS-DC2    WS-FS2    ws.its service-ata
WS-DC2 2020-05-22 04:45:38 WS-DC2    WS-CM     ws.its service-ata
WS-DC2 2020-05-22 05:35:37 WS-DC2    WS-FS1    ws.its service-ata
WS-DC1 2020-05-22 05:36:29 WS-DC2    WS-NPS1   ws.its service-ata
WS-DC3 2020-05-22 06:41:36 WS-ATA    WS-CL3    ws.its service-ata
WS-DC1 2020-05-22 06:51:14 WS-MON    WS-HV1    ws      service-prtg
WS-DC1 2020-05-22 06:51:14 WS-MON    WS-HV1    ws      service-prtg
WS-DC1 2020-05-22 07:00:03 WS-ATA    WS-HV1    ws.its service-ata
WS-DC3 2020-05-22 07:00:50 WS-ATA    WS-CL3    ws.its service-ata
WS-DC3 2020-05-22 07:10:46 WS-DC2    WS-CL3    ws.its service-ata
WS-DC2 2020-05-22 07:22:16 WS-ATA    WS-CM     ws.its service-ata
WS-DC1 2020-05-22 07:22:21 WS-ATA    WS-FS1    ws.its service-ata
WS-DC2 2020-05-22 07:51:19 DRUCKER-1 WS-FS2    WS      service-print1
WS-DC2 2020-05-22 07:51:19 DRUCKER-1 WS-FS2    WS      service-print1
WS-DC2 2020-05-22 07:52:58 DRUCKER-1 WS-FS2    WS      service-print1
WS-DC2 2020-05-22 07:52:58 DRUCKER-1 WS-FS2    WS      service-print1
WS-DC1 2020-05-22 07:54:39 WS-ATA    WS-NPS1   ws.its service-ata
WS-DC2 2020-05-22 07:54:47 DRUCKER-1 WS-FS2    WS      service-print1
```

Das bedeutet, ich muss meinen Fileserver in die NTLM-Ausnahmen mit aufnehmen, damit ich Scan2SMB verwenden kann. Super! Aber für den Versuch nehme ich diese Konfiguration vor:



Jetzt versuche ich erneut einen Scan. Aber es funktioniert immer noch nicht. Der Scanner kann das SMB-Target nicht ansprechen. Die Ursache liegt auch hier wieder in der Security: Ich habe alle meine Freigaben mit SMB3-Encryption geschützt. Der Datentransfer über das Netzwerk ist also wie bei HTTPS geschützt. Das müssen aber Client und Server beherrschen. Und offenbar kann dieser moderne Drucker kein SMB3. 2x Super! Dabei wurde es explizit als sicheres Gerät beworben…

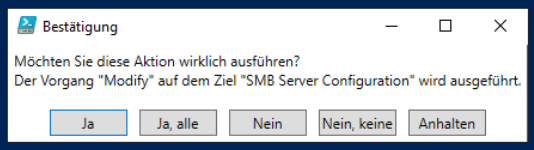


Für den Versuch deaktiviere ich die SMB-Encryption auf meinem Fileserver. Dafür muss ich die Encryption aber auch auf Serverebene deaktivieren (man erkennt vielleicht im Bild, dass der Dateizugriff verschlüsseln Haken ausgegraut ist). Das geht nur mit der PowerShell:

```
PS C:\Windows\system32> Get-SmbServerConfiguration


AnnounceComment                 :
AnnounceServer                  : False
AsynchronousCredits             : 512
AuditSmb1Access                 : False
AutoDisconnectTimeout           : 15
AutoShareServer                 : True
AutoShareWorkstation            : True
CachedOpenLimit                 : 10
DurableHandleV2TimeoutInSeconds : 180
EnableAuthenticateUserSharing   : False
EnableDownlevelTimewarp         : False
EnableForcedLogoff              : True
EnableLeasing                   : True
EnableMultiChannel              : True
EnableOplocks                   : True
EnableSecuritySignature         : True
EnableSMB1Protocol              : False
EnableSMB2Protocol              : True
EnableStrictNameChecking        : True
EncryptData                     : True
IrpStackSize                    : 17
KeepAliveTime                   : 2
```

```
PS C:\> Set-SmbServerConfiguration -EncryptData $false
```



Jetzt kommt die Datei auf dem Fileserver an. Nachdem meine Sicherheitsfeatures deaktiviert sind.

Lösungsansatz

Das Gerät ist sonst echt klasse. Die Geschwindigkeit, der Stromverbrauch, die Bedienung und auch das konfigurierbare Webserver-Zertifikat und der 802.1x-Zertifikatsupport gefallen mir sehr. Also werde ich es behalten. Und mal ehrlich: Welche Multifunktionsdrucker aus dem 21. Jahrhundert erfüllen meine Sicherheits-Anforderungen?

Aber meinen Fileserver möchte ich wieder absichern. Damit ist die Lösung einfach: **Ich erstelle einen neuen Server**, der als Printserver den Drucker im Netzwerk verfügbar macht und gleichzeitig eine Standard-Freigabe für das Scan2SMB anbietet. Diesen Server nehme ich in der NTLM-Einschränkung aus und die Freigabe wird nicht mit SMB3 verschlüsselt.

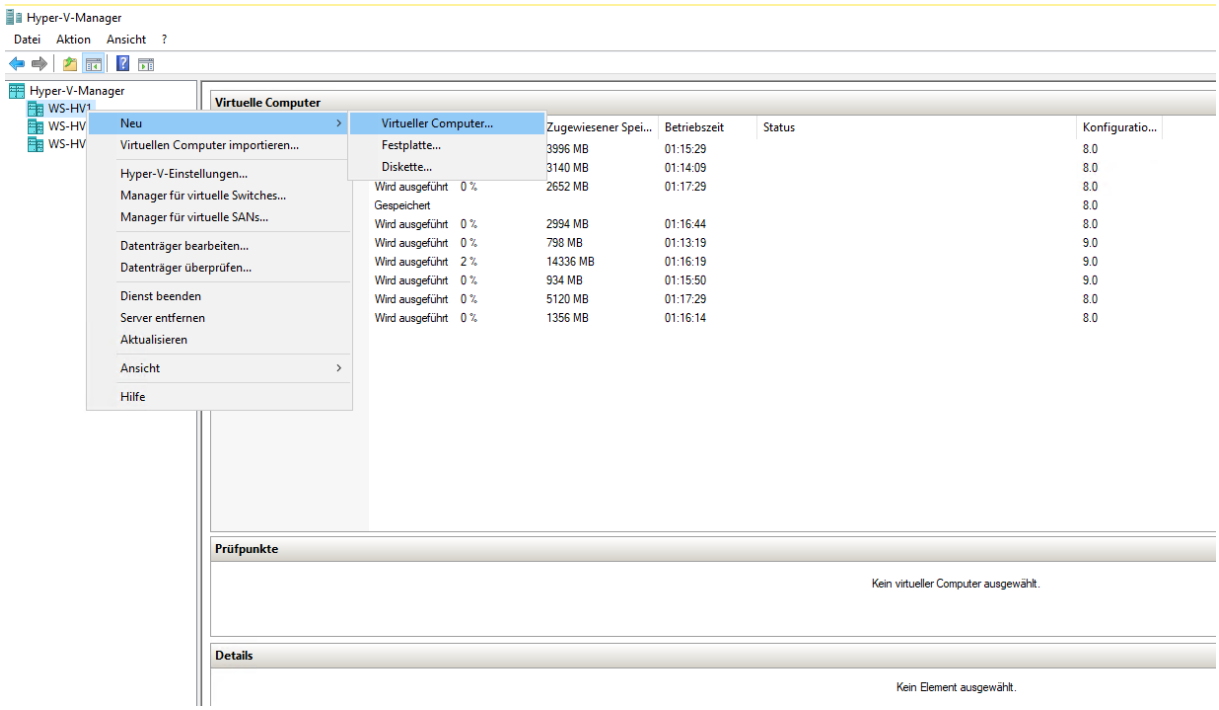
Ein netter Benefit kommt durch den Einsatz meines DFS-Namespaces dazu: Die Freigabe kann ich im DFS veröffentlichen und so den Zugriff an die Clients weitergeben.

Das klingt nach einem Plan.

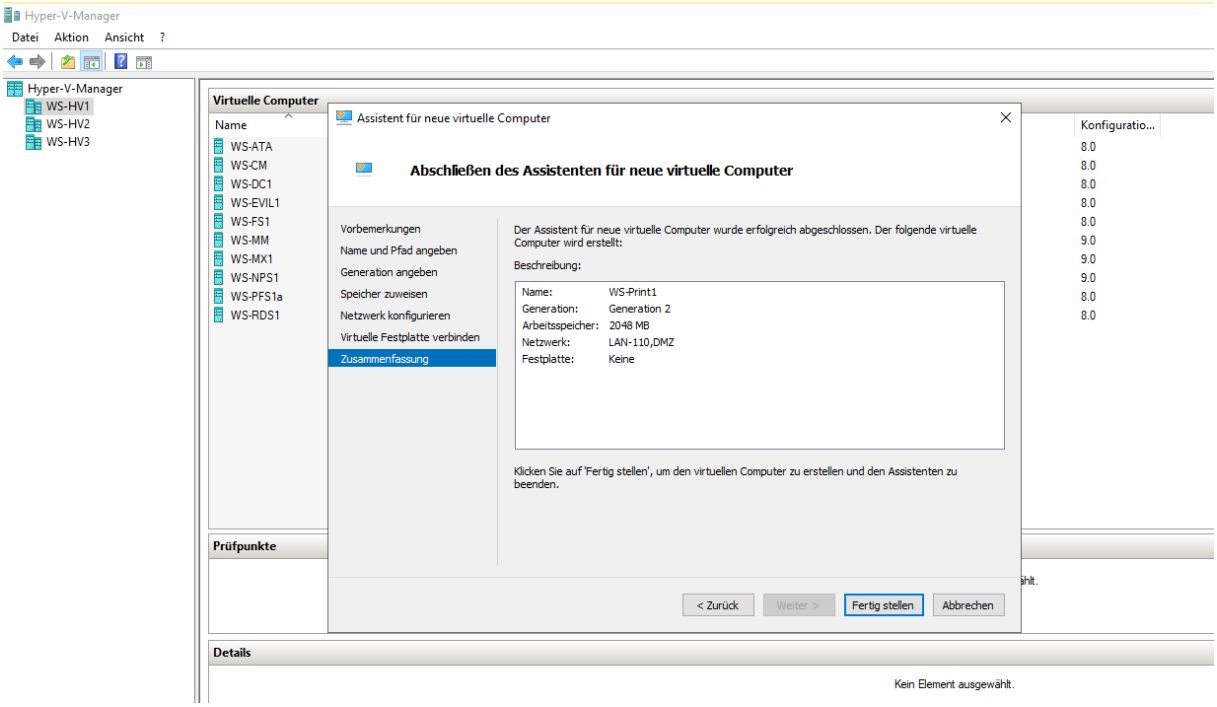
Aufbau des neuen Servers

Konfiguration der virtuellen Maschine

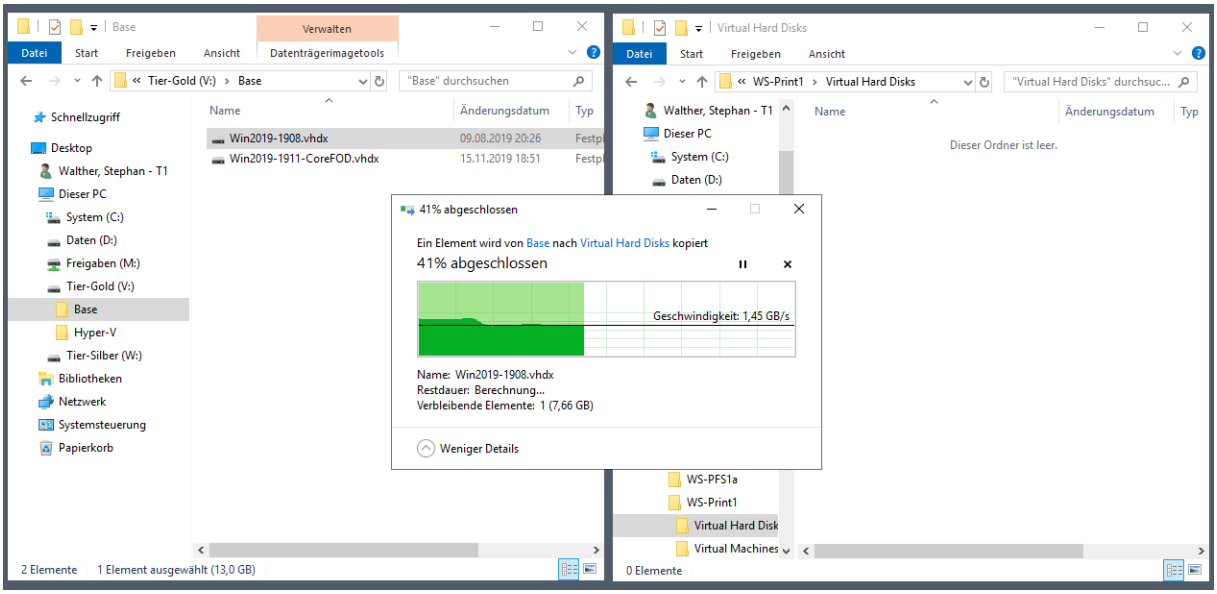
Der neue Server wird als virtuelle Maschine in einem meiner Hyper-V-Hosts Platz finden. Ich habe am Druckerstandort 2 Hosts. Einer davon hat aktuell mehr Platz auf der VM-Festplatte und gleichzeitig auch mehr freien Arbeitsspeicher: Mein WS-HV1. Hier erstelle ich eine neue VM:



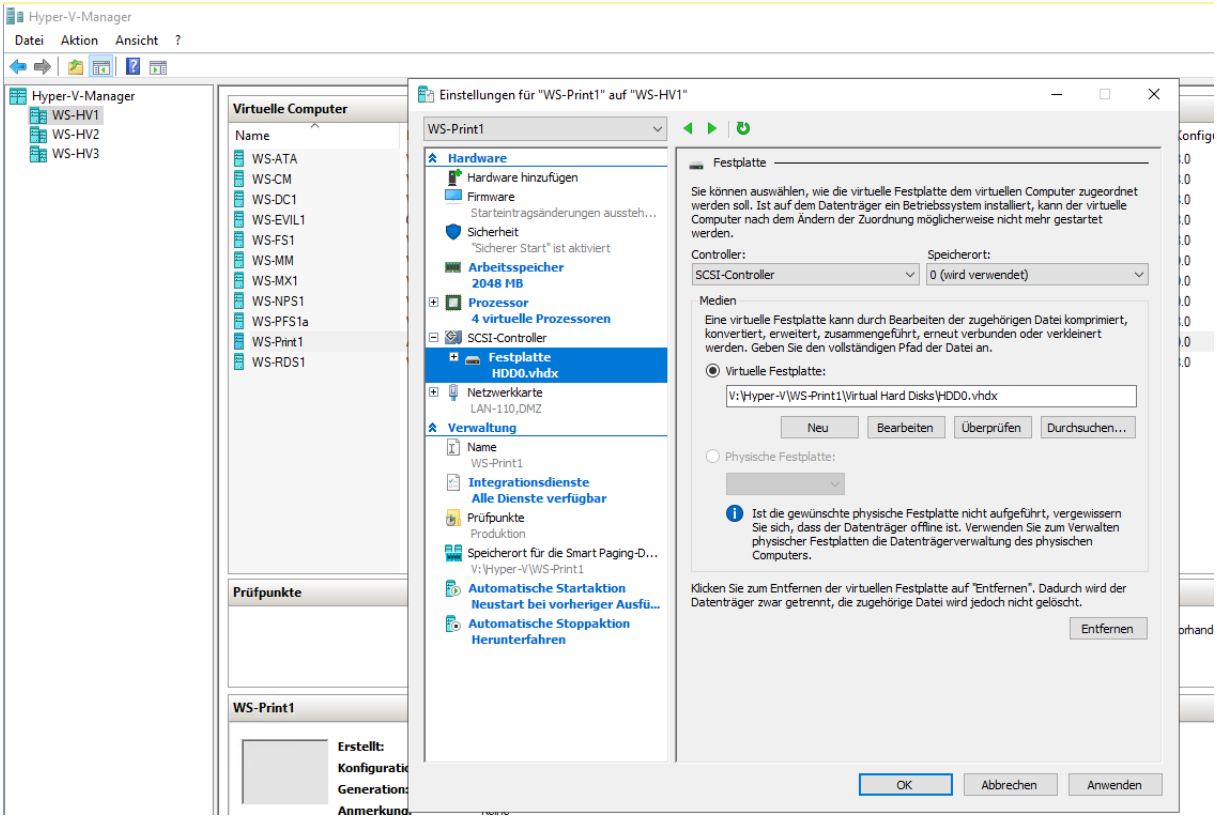
Die virtuelle Maschine bekommt noch keine Festplatte und wird erst einmal in das Client-Netz geschoben:



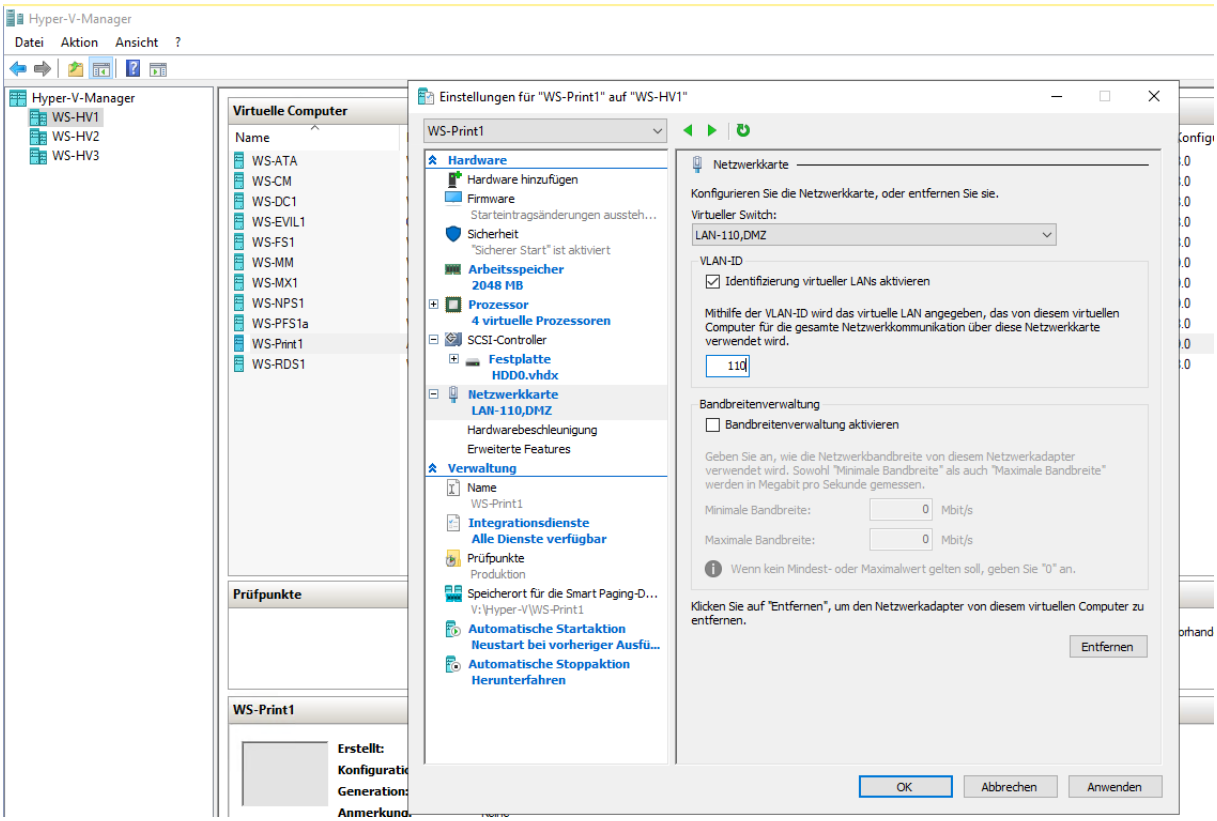
Das Betriebssystem habe ich in einer Basis-VHDX-Datei bereits generalisiert. Diese Datei kopiere ich in das neue VM-Verzeichnis auf dem Hyper-V-Host:



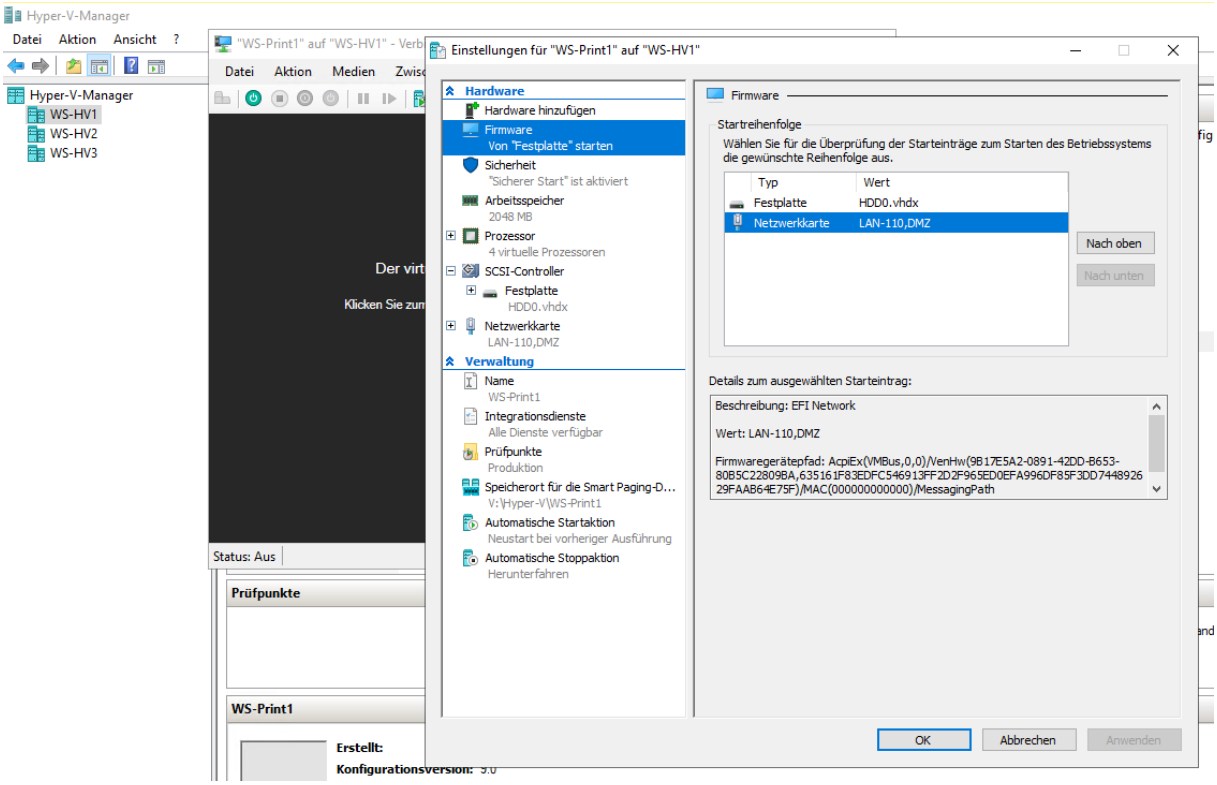
Jetzt bekommt die neue VM ihren Feinschliff: mehr CPU, mehr Arbeitsspeicher, die neue Festplatte und einige andere Grundkonfigurationen werden zugewiesen:



Das Client-Netz arbeitet mit einem VLAN. Dessen VLAN-ID trage ich in die Netzwerkkarte ein:



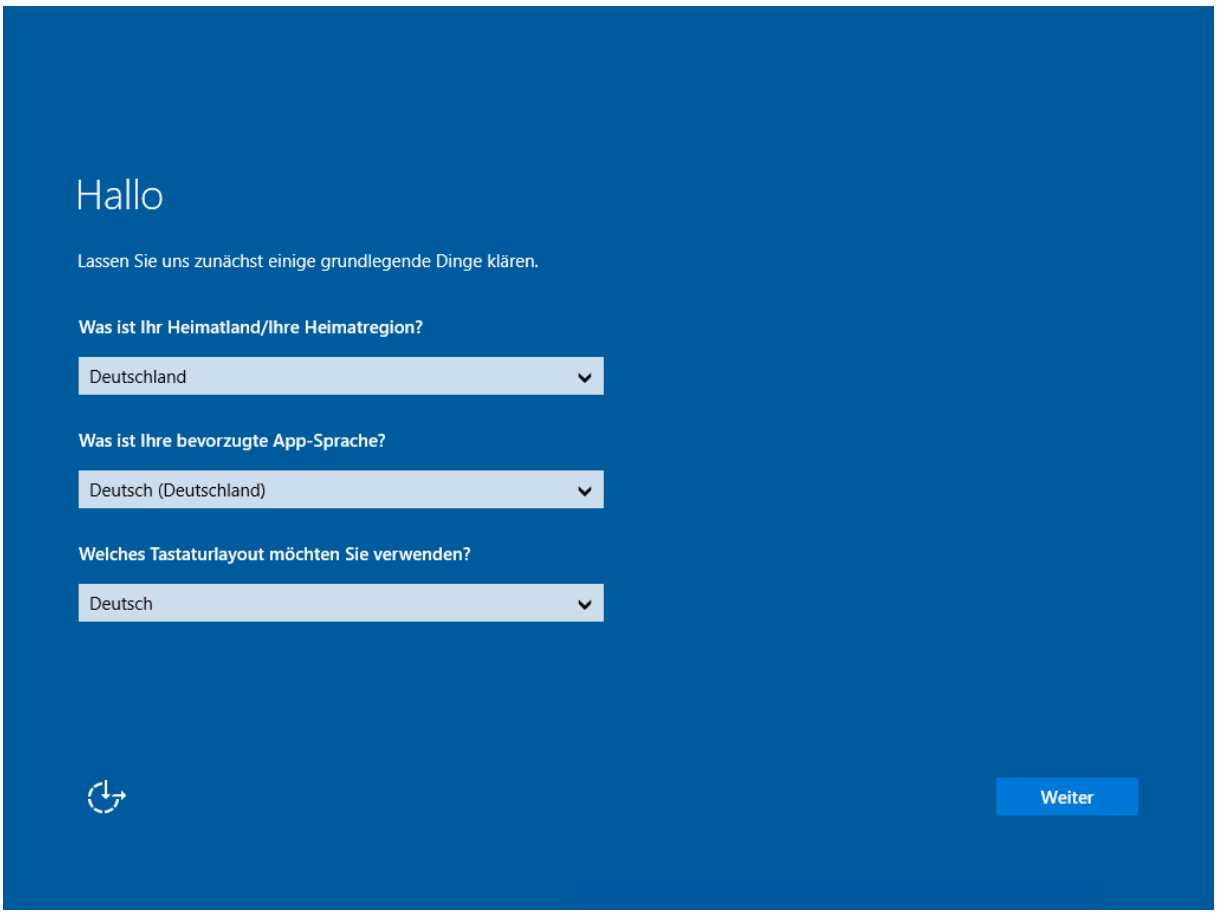
Jetzt verändere ich noch die Startreihenfolge:



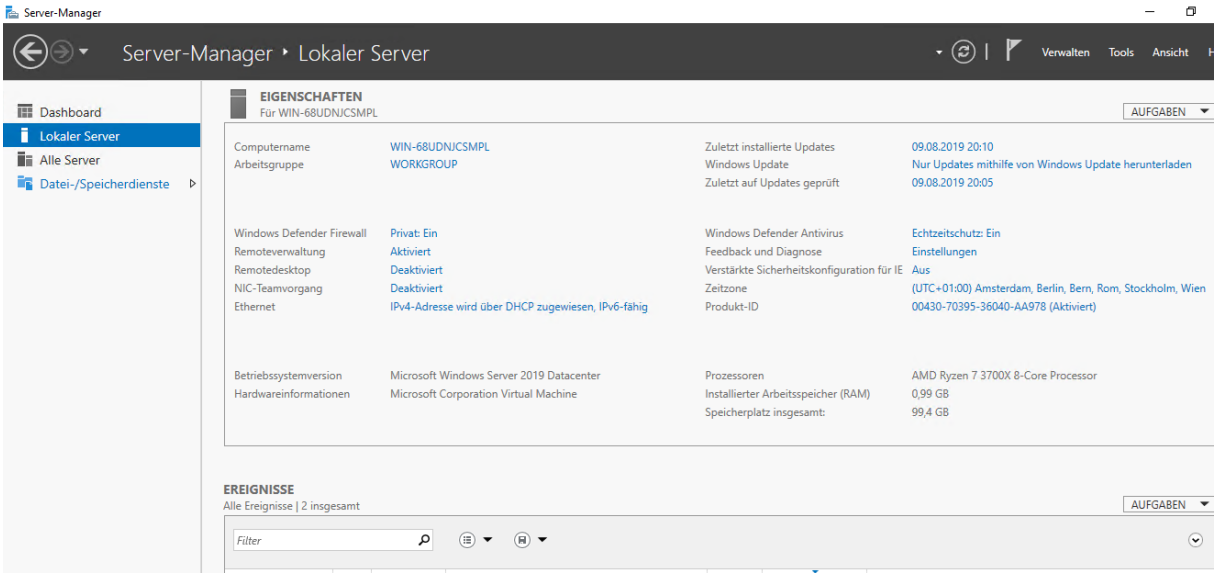
Dann ist die neue VM einsatzbereit.

Grundkonfiguration des Windows Servers

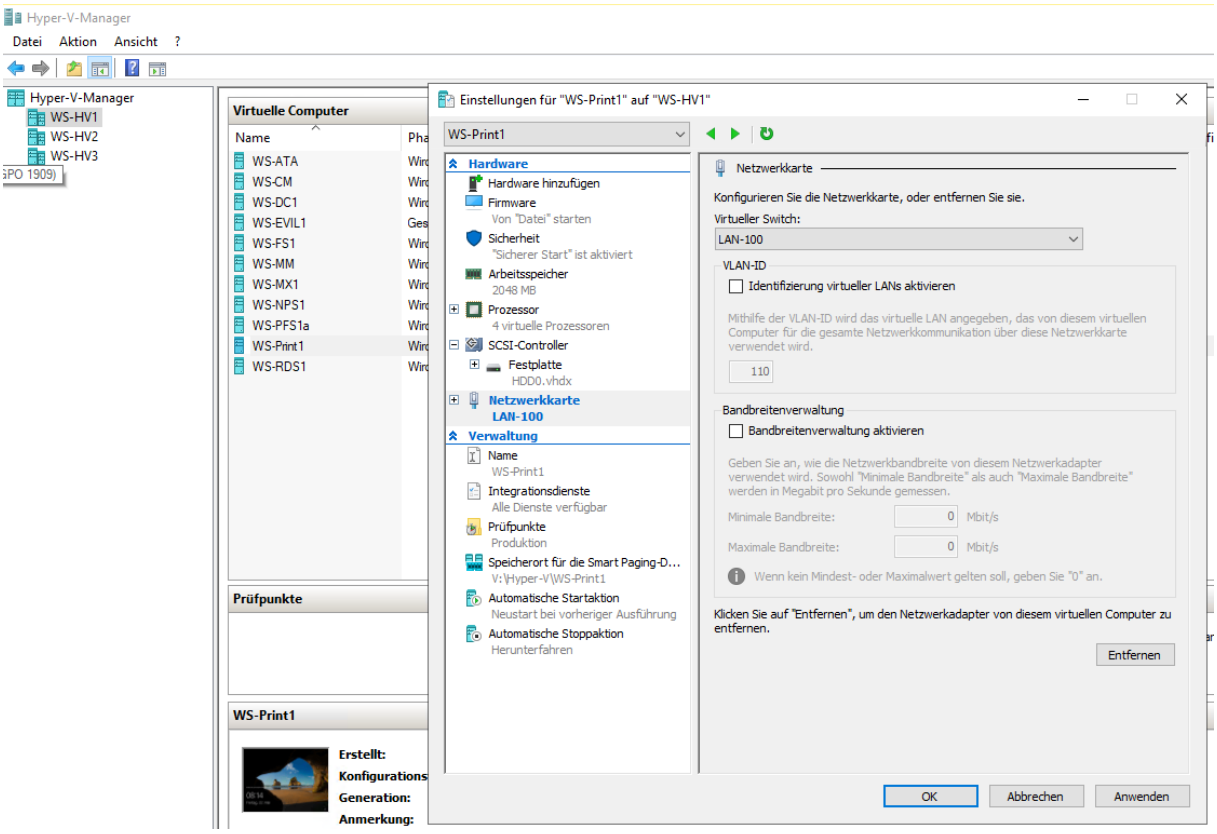
Nach dem Einschalten beginnt der Windows Server 2019 seine Erstkonfiguration:



Nach wenigen Minuten kann ich mich anmelden. Wie üblich werde ich vom Server Manager begrüßt. Durch den Anschluss im Client-Netzwerk und einer automatischen IP-Adressvergabe kann der Server problemlos ins Internet und kann sich dort automatisch aktivieren:



Mehr muss der Server aber nicht erledigen. Daher patche ich die VM in das Servernetz:

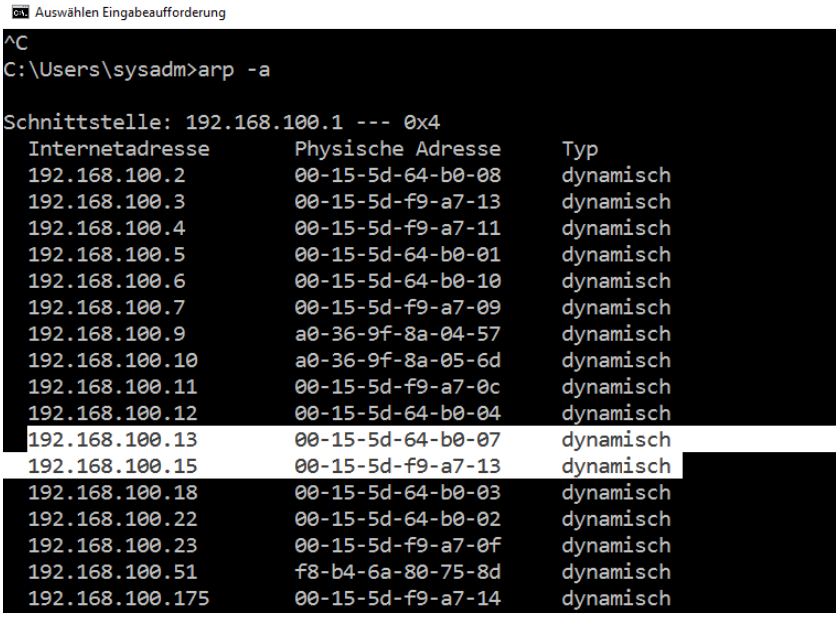


Jetzt benötige ich eine freie IPv4-Adresse.

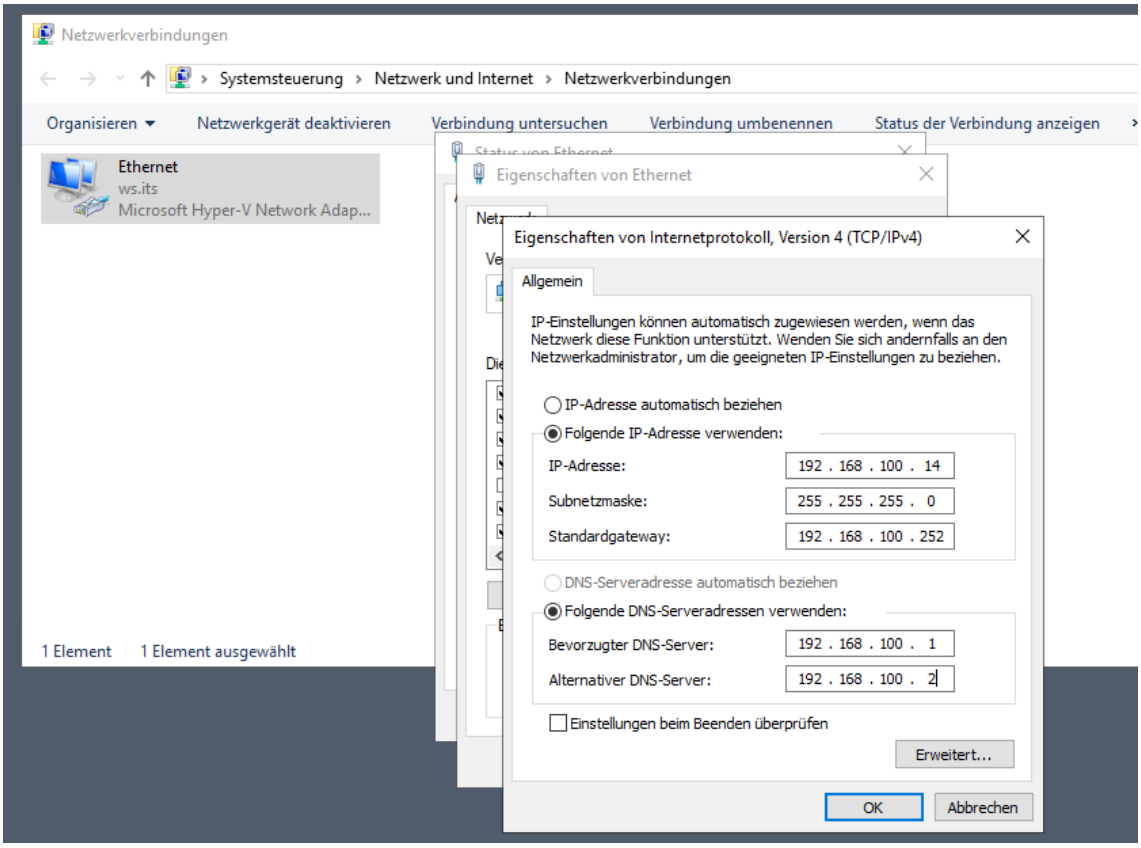
Praxistipp:
Bei meinen Kunden sehe ich da immer wieder die Verwendung vom Befehl ping. „Erhalte ich keine Antwort, dann ist die IP-Adresse frei.“ Aber nicht jedes System reagiert auf ICMP-Echo-Requests – also die Nachrichten, die im OSI-Layer 4 verschickt werden. Ich schaue lieber auf den Layer 2. Dieser kann nicht so einfach unterdrückt werden. Mit ARP kann ich mir auf einem anderen Server im gleichen Netzwerksegment die MAC-Adressen ansehen. Dazu werden die IPv4-Adressen

ausgegeben. ARP-Nachrichten werden mit Broadcasts verteilt. Jeder im Segment angeschlossene Client kann also diese Nachrichten hören und im ARP-Cache speichern. Und da heutzutage jedes System im Netz ins Internet will, sucht es nach dem Gateway. Und dafür braucht es dessen MAC-Adresse. Der Cache ist also üblicherweise immer gut gefüllt.

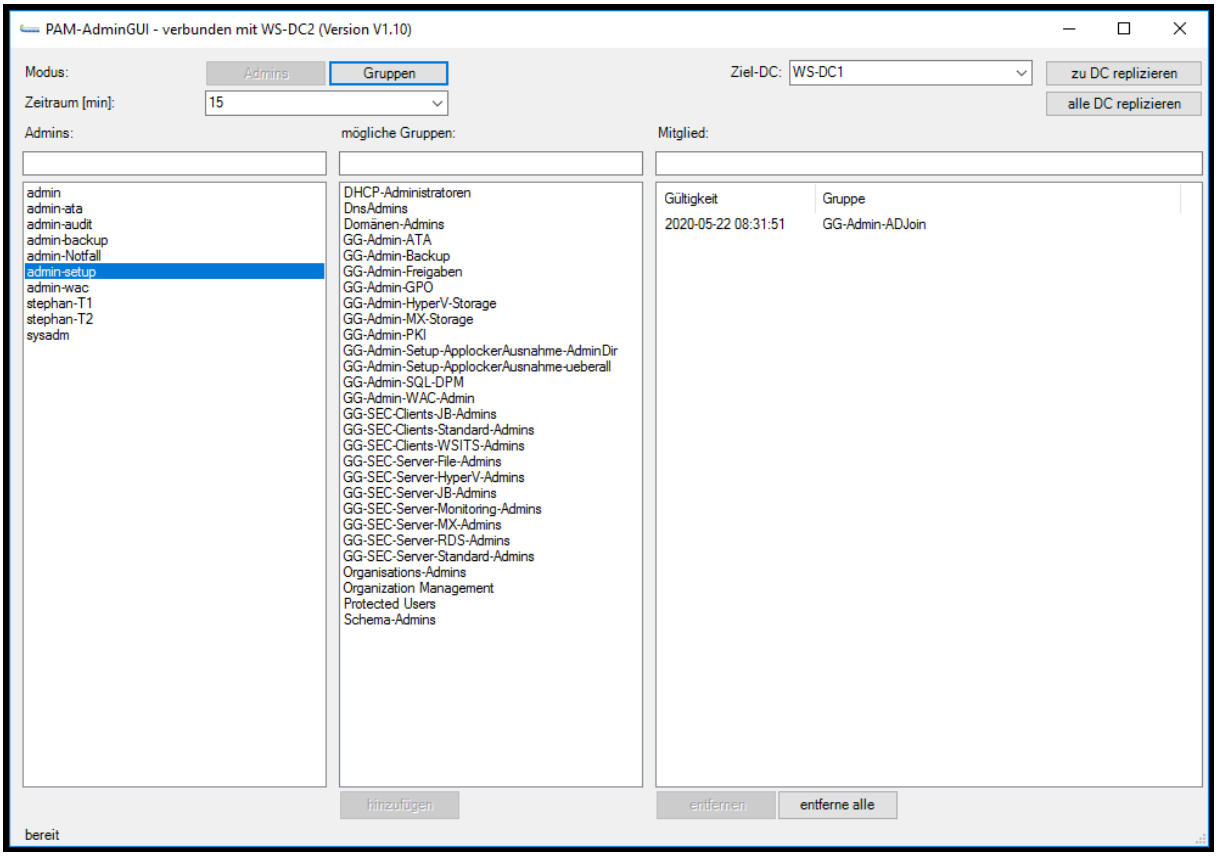
So finde ich schnell eine Lücke in den Server-IP-Adressen:



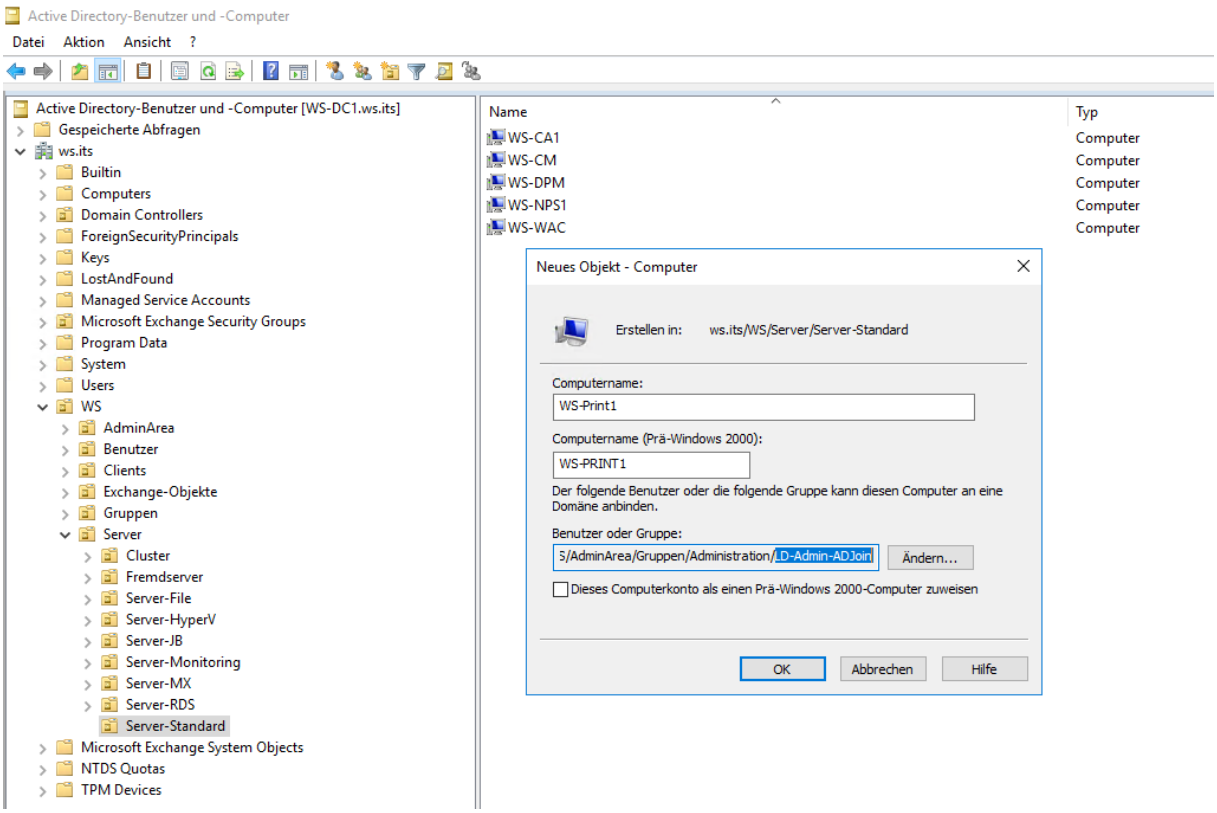
Die freie IP-Adresse nehme ich statisch in die Konfiguration auf:



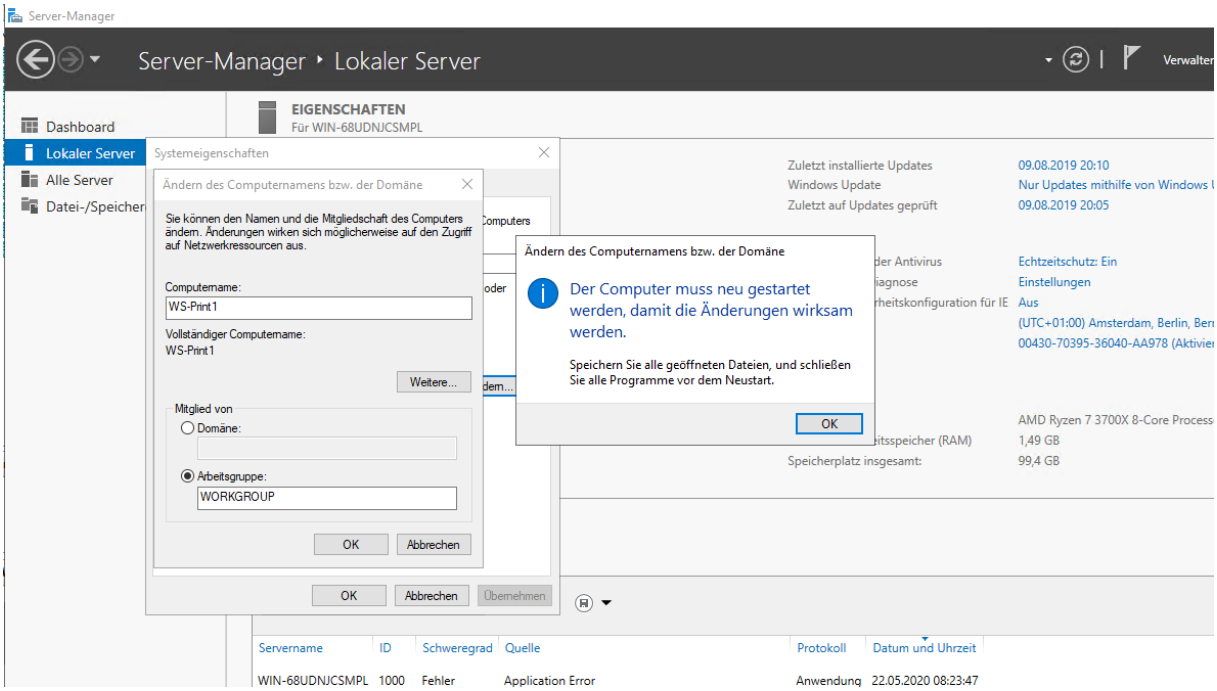
Jetzt kommt der Domain Join. Dafür richte ich wieder einen temporären Admin mit meinem PAM-Tool her:



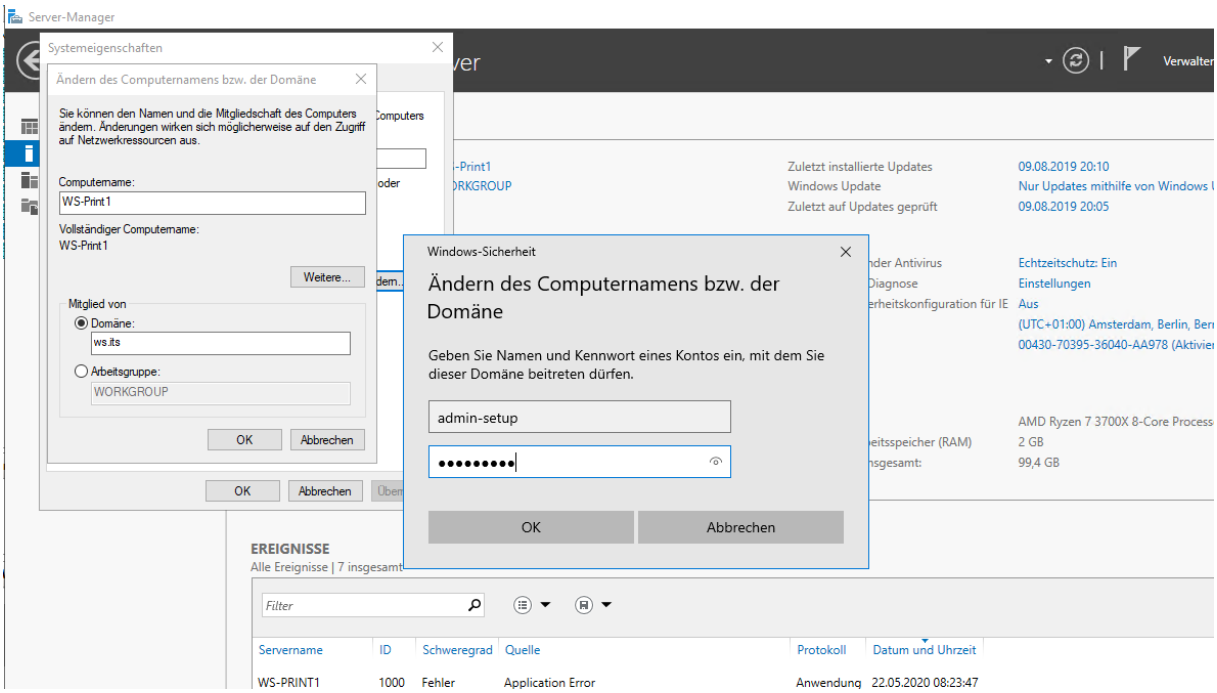
Im Active Directory erstelle ich das Computerkonto für den neuen Server in der richtigen Organisationseinheit. Das Recht des Domain Joins delegiere ich an meine lokale Domain Group „LD-Admin-ADJoin“:



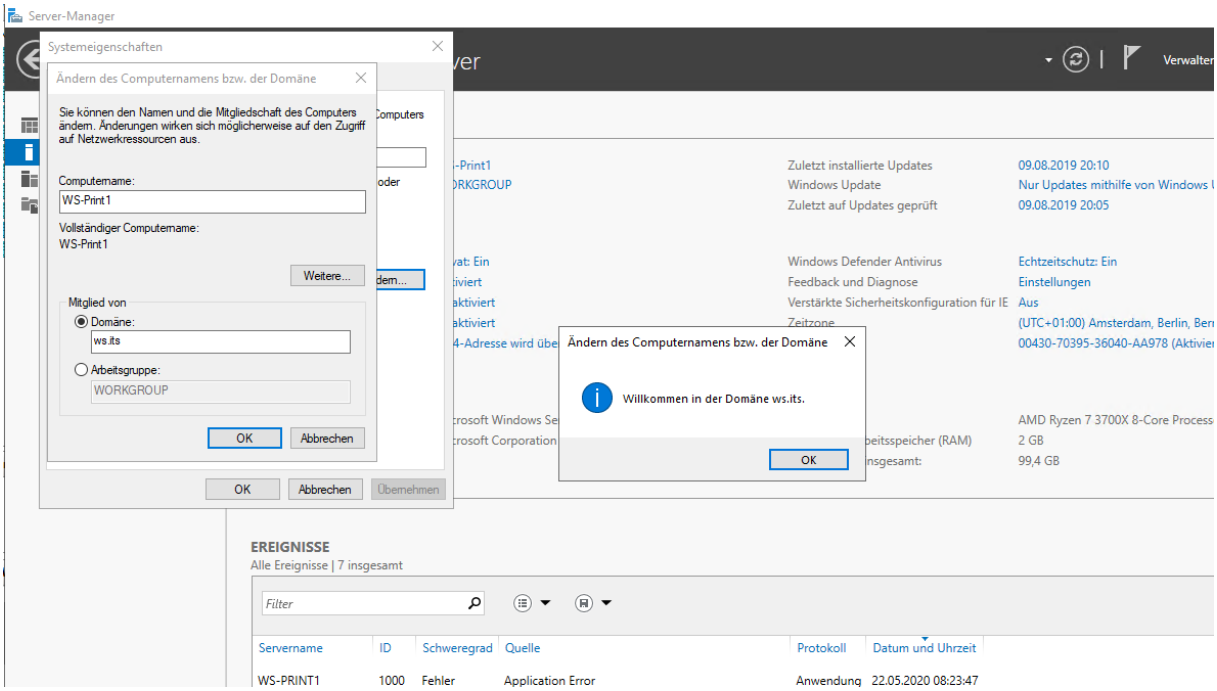
Den neuen Server muss ich aber vor dem Domain Join umbenennen. Diese Aktion benötigt einen Neustart:



Der Server ist schnell wieder online und kann nun mit der Vorbereitung in die Domäne aufgenommen werden:

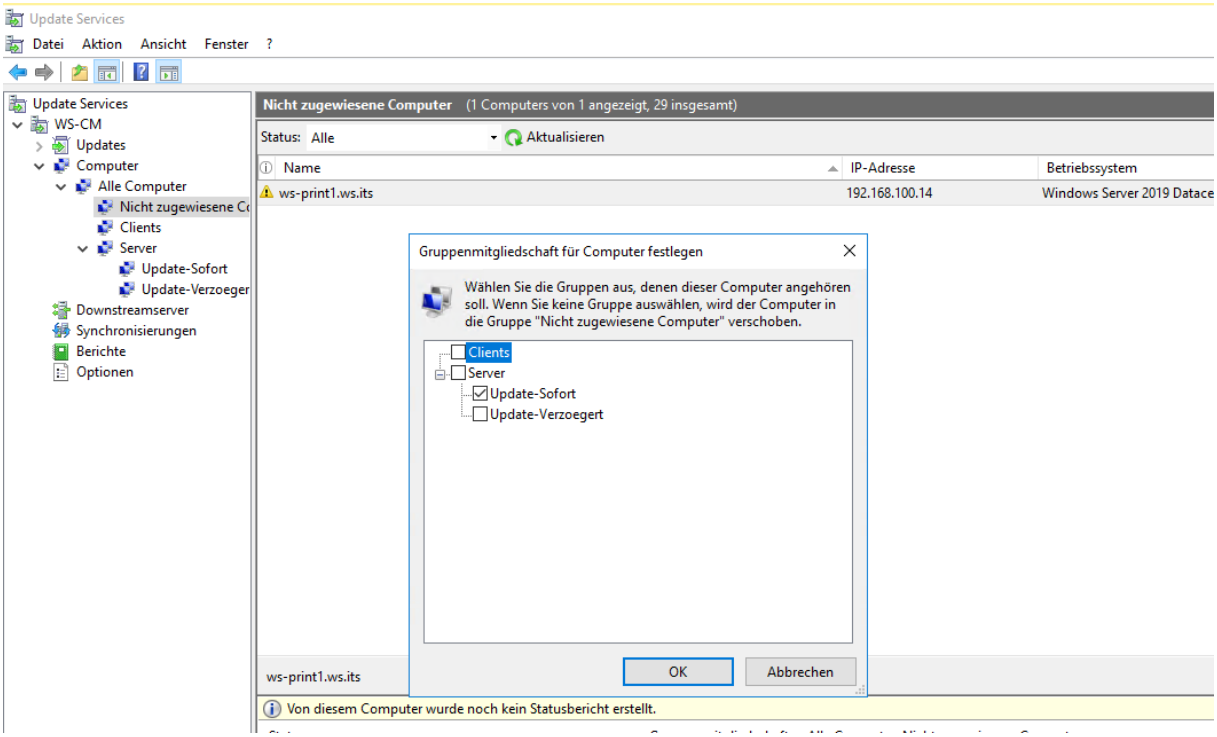


Das hat wie erwartet funktioniert. Abschließend ist ein Neustart erforderlich:

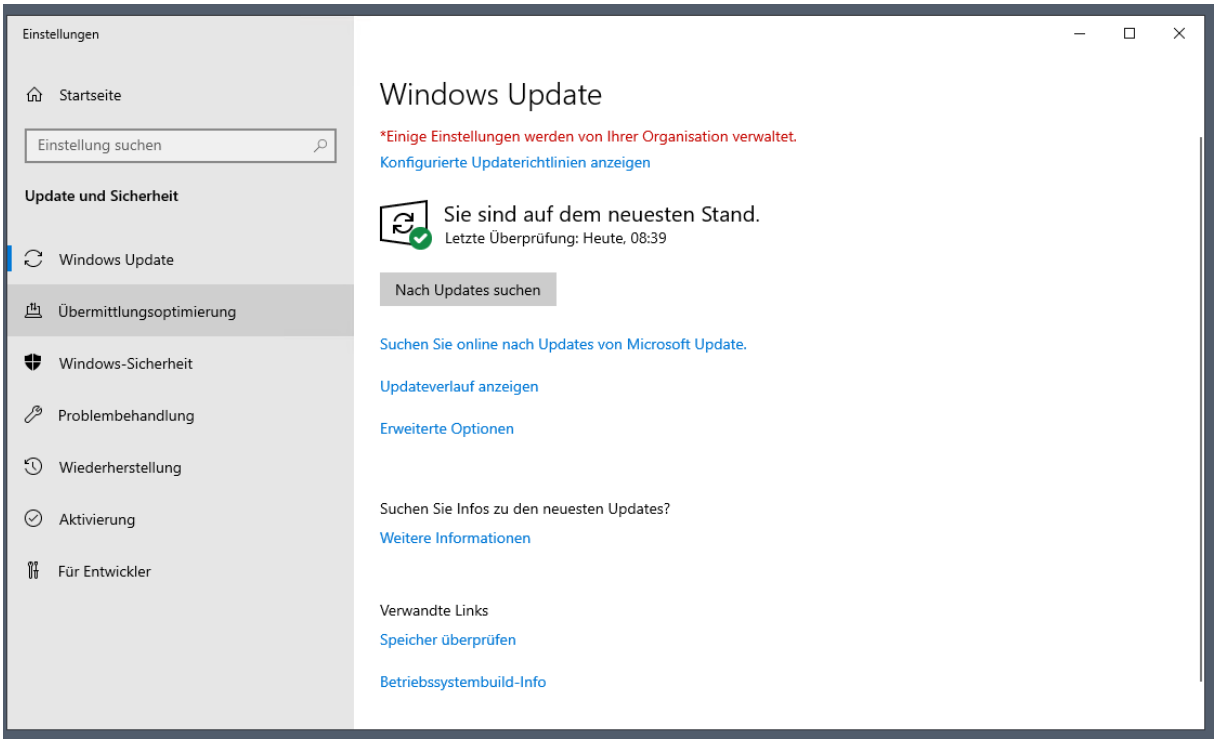


Windows Update

Durch die auf die Organisationseinheit wirkenden Gruppenrichtlinien meldet sich der Server auch beim WSUS. Hier weise ich ihm einen Container zu, mit dem ich den automatischen Installationszeitpunkt für Windows Updates zuweisen. Die Genehmigung meiner Updates habe ich durch ein Script automatisiert. Die beiden Container erhalten die Genehmigung mit einem zeitlichen Versatz. So schieße ich mir nicht alle Server auf einmal ab. Dieser Server ist von der Funktion her eher unspektakulär. Daher darf er in der ersten Update-Welle mitschwimmen:

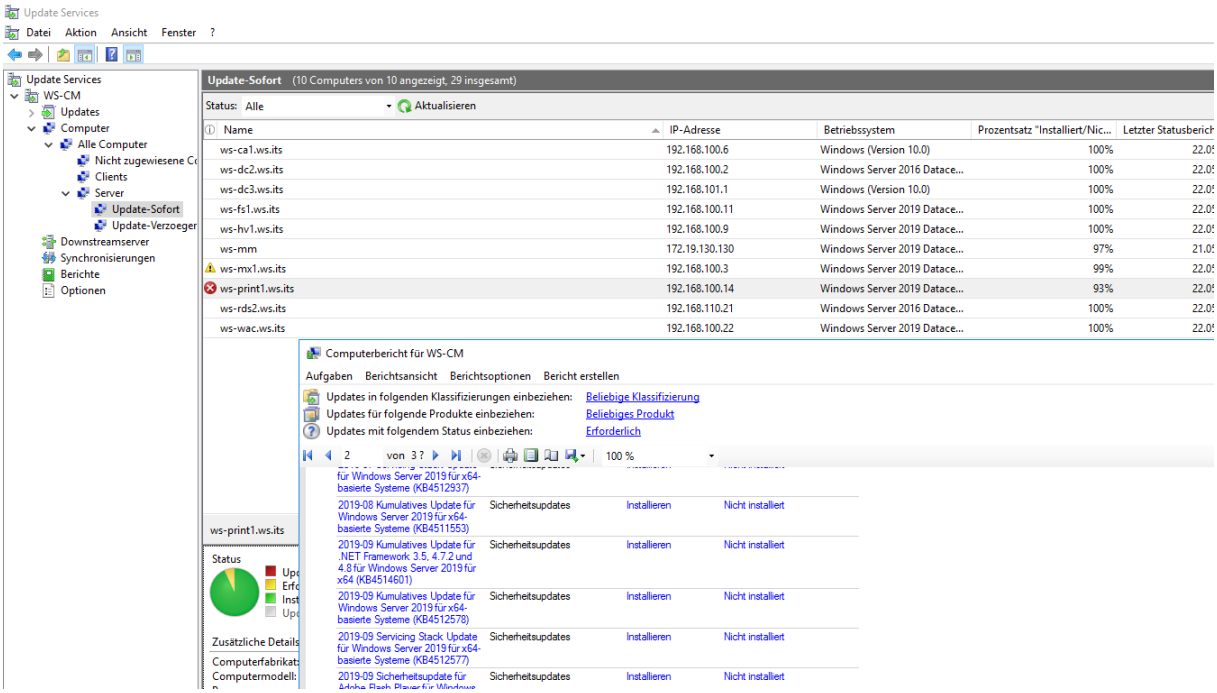


Jetzt suche ich nach Updates. Die Einstellungen zeigen das Wirken der Gruppenrichtlinien an:



Die vorbereitete VHDX für meine neue VM hatte ich mit einem Windows Server 2019 mit dem Patchlevel 1908 hergestellt. Da sollten also einige Updates fehlen. Aber die Suche zeigt keine Treffer…

In der WSUS-Konsole dagegen sehe ich etliche Updates, die noch fehlen:



Warum werden diese nicht installiert?

Hintergrund:
Der Windows Update Server lädt sich die Update-Kataloge von Microsoft herunter. Üblicherweise müssen die Updates genehmigt werden, bevor die eigentlichen Update-Dateien heruntergeladen werden. Ohne Genehmigung sieht ein Client die Updates nicht – und denkt, er ist Uptodate. Das ist bei mir aber nicht der Fall. Wie im Bild sichtbar sind meine Updates mit dem Flag „installieren“ versehen. Der Client muss sie also anwenden.

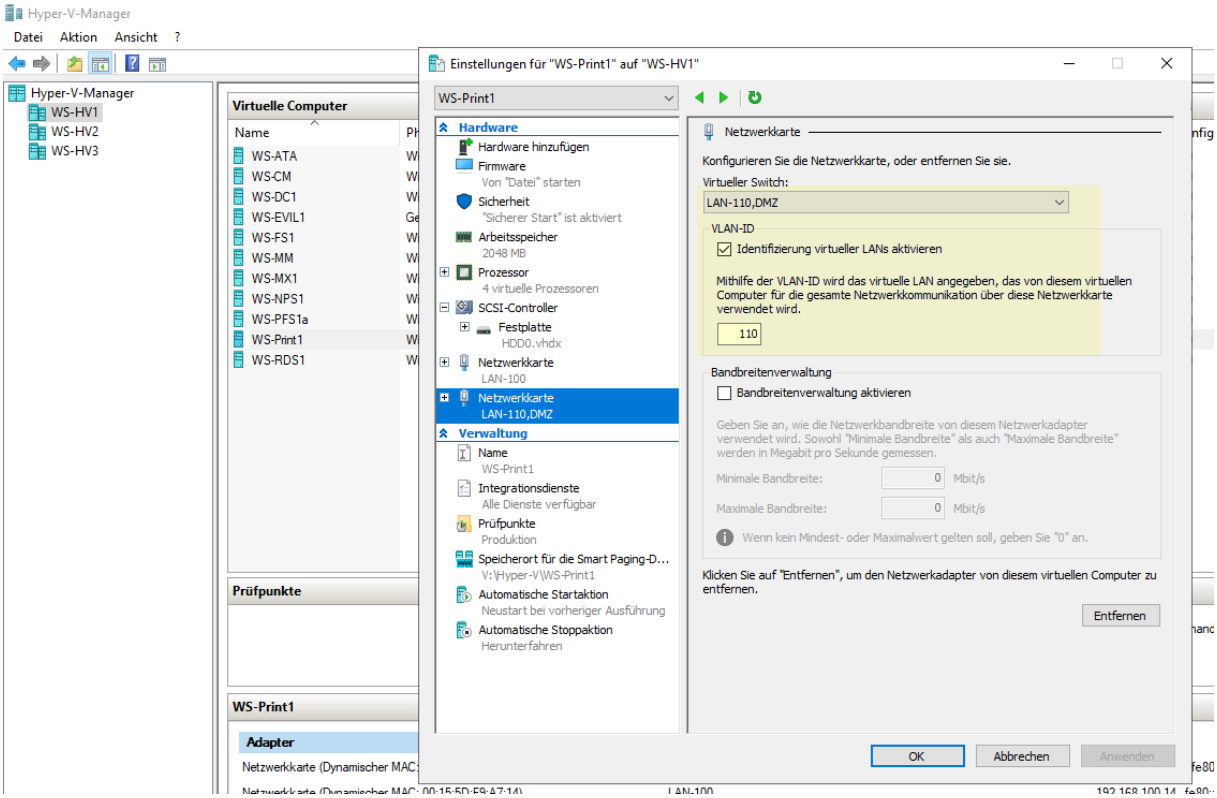
Jetzt ist es aber auch die Regel, dass neuere Updates die Vorgänger ersetzen können. So braucht ein Client nur die neuere Version zu installieren. Ebenso kann aber ein Update auch ein anderes als Installationsvoraussetzung erwarten. Dann muss dieses zuerst installiert werden. Und dann kommt noch eine Bereinigungsoption im WSUS dazu, die ich immer mal wieder anwende, wenn mir die Update Dateien zu viel Platz benötigen.

So entsteht schnell folgende Kausalitätskette:

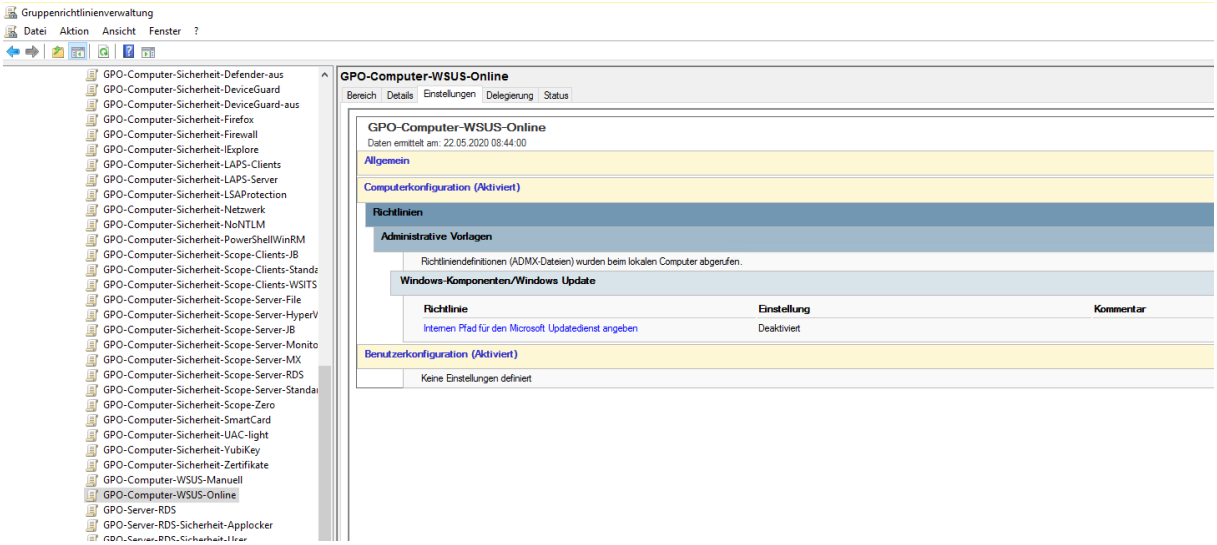
- Von einem veralteten Update, für das ein Nachfolger existiert, wurde durch eine Bereinigung die Update Dateien entfernt.
- Das veraltete Update war aber die Installationsvoraussetzung für ein neueres Update.
- Der Client findet das veraltete Update nicht mehr und muss es daher auch nicht installieren.
- Da die Voraussetzung für ein neueres Update damit nicht erfüllt ist, muss er dieses auch nicht installieren.
- Da der Client keine Updates installiere muss, zeigt er diesen Status an:

Nur im WSUS kann man diesen Fehler erkennen. Das hätte Microsoft wirklich besser lösen können…

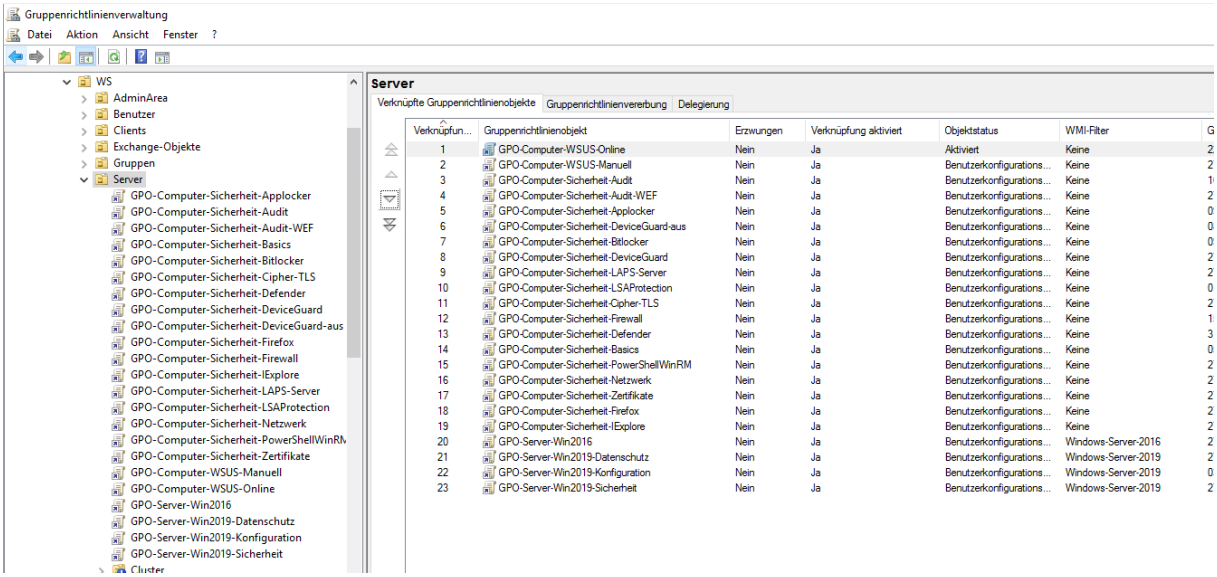
Die veralteten Updates können auch nicht wieder im WSUS heruntergeladen werden. Also kann ich den Server nur ins Internet lassen, damit er sich dort die Updates zieht. Also patche ich ihn wieder in das offene Client-Netzwerk:



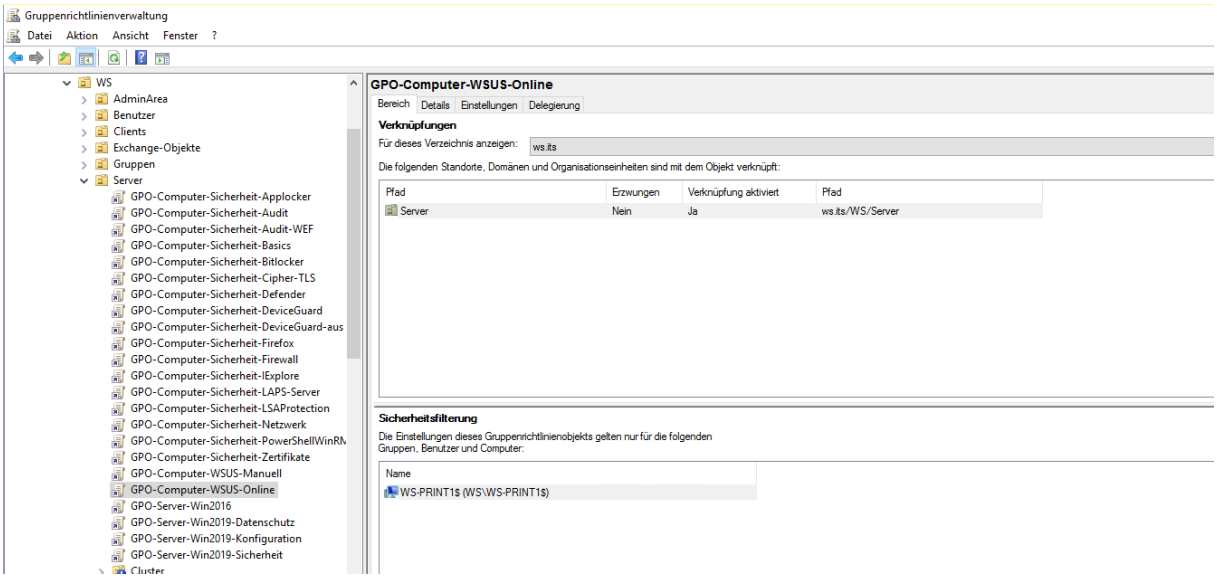
Zusätzlich nehme ich ihn in den Sicherheitsfilter einer Gruppenrichtlinie auf, die ihn vom WSUS abklemmt. Die Einstellung ist recht einfach: Die Verwendung des internen WSUS wird deaktiviert:



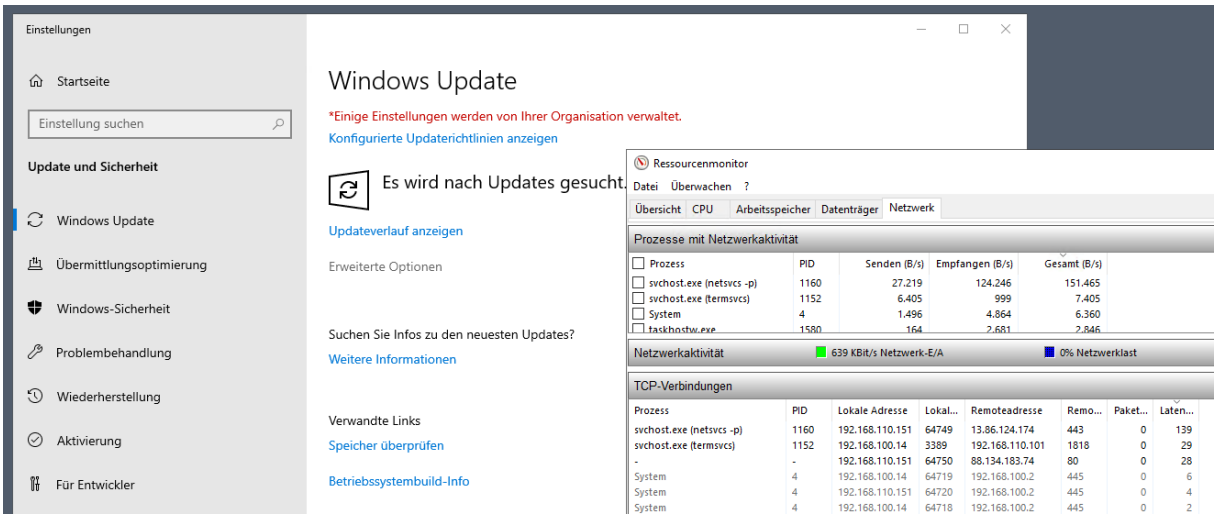
Die Einstellung setzt sich gegen die meiner eigentlichen WSUS-GPO durch eine Überlagerung durch. Hier spielt die Verarbeitungsreihenfolge eine wichtige Rolle:



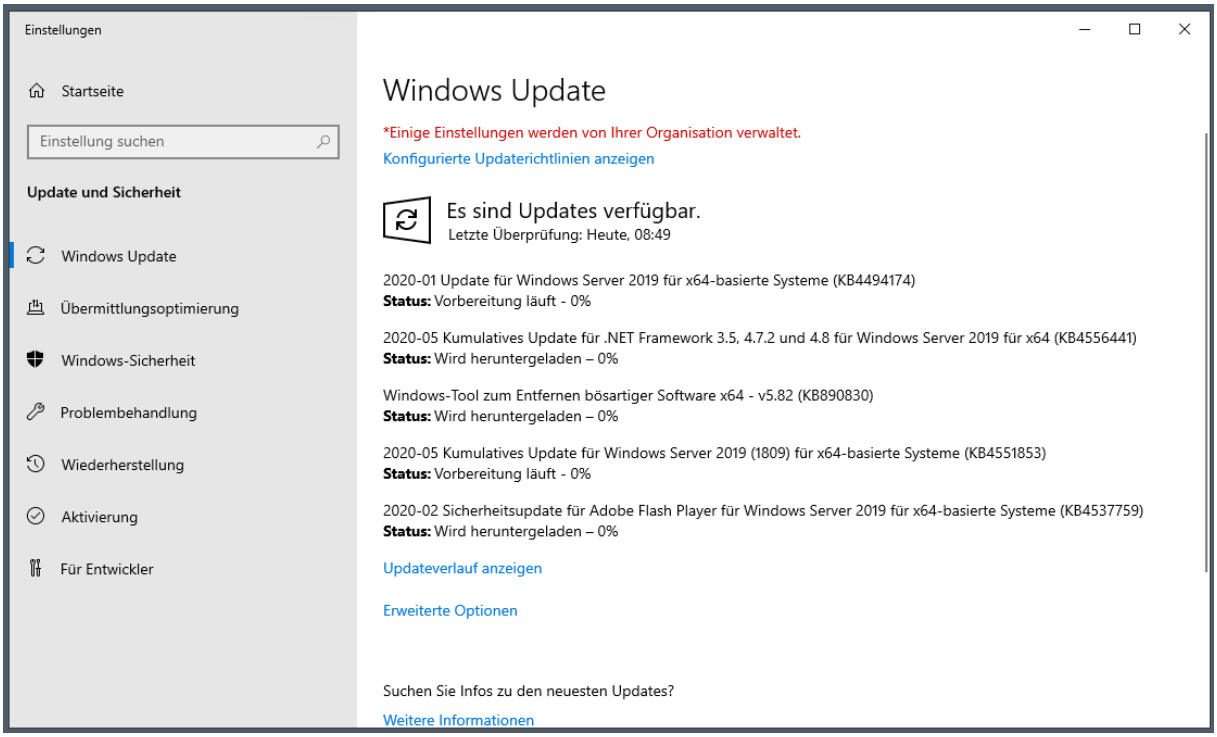
Die GPO wirkt aber durch den Sicherheitsfilter nur für meinen neuen Print-Server:



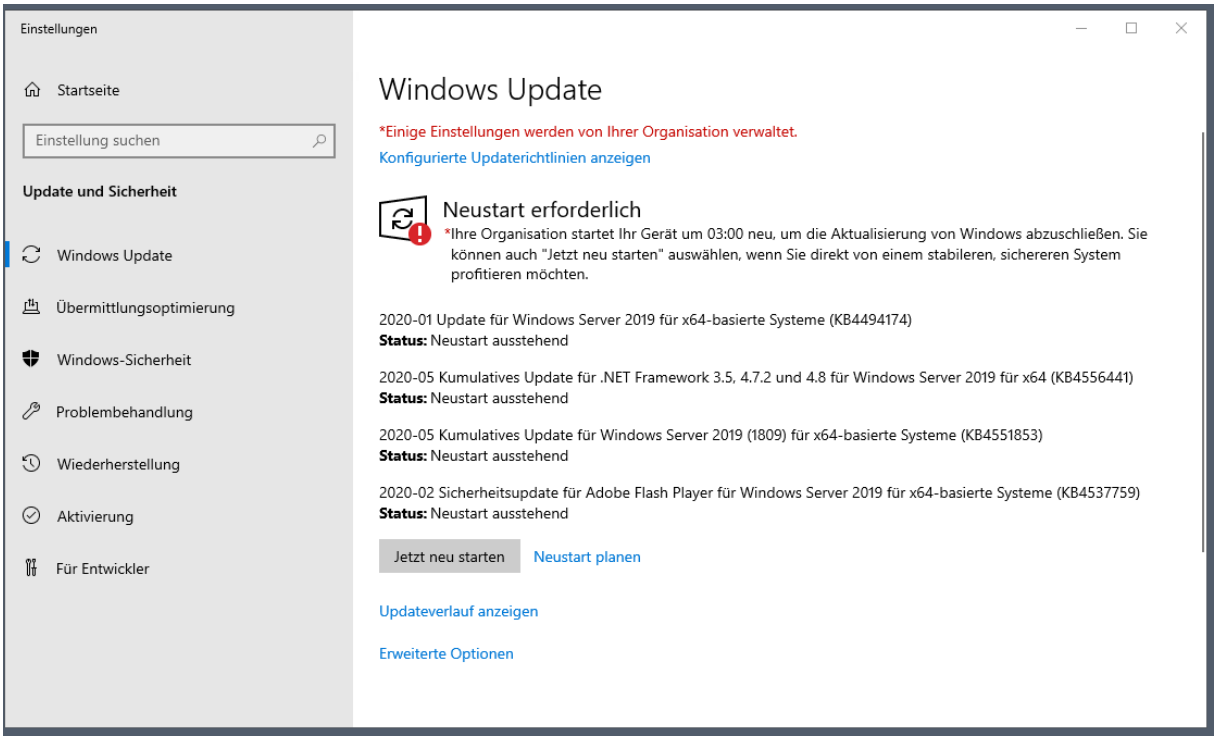
Ein gpupdate später suche ich auf dem Server wieder nach Updates. Dieses Mal geht er direkt ins Internet:



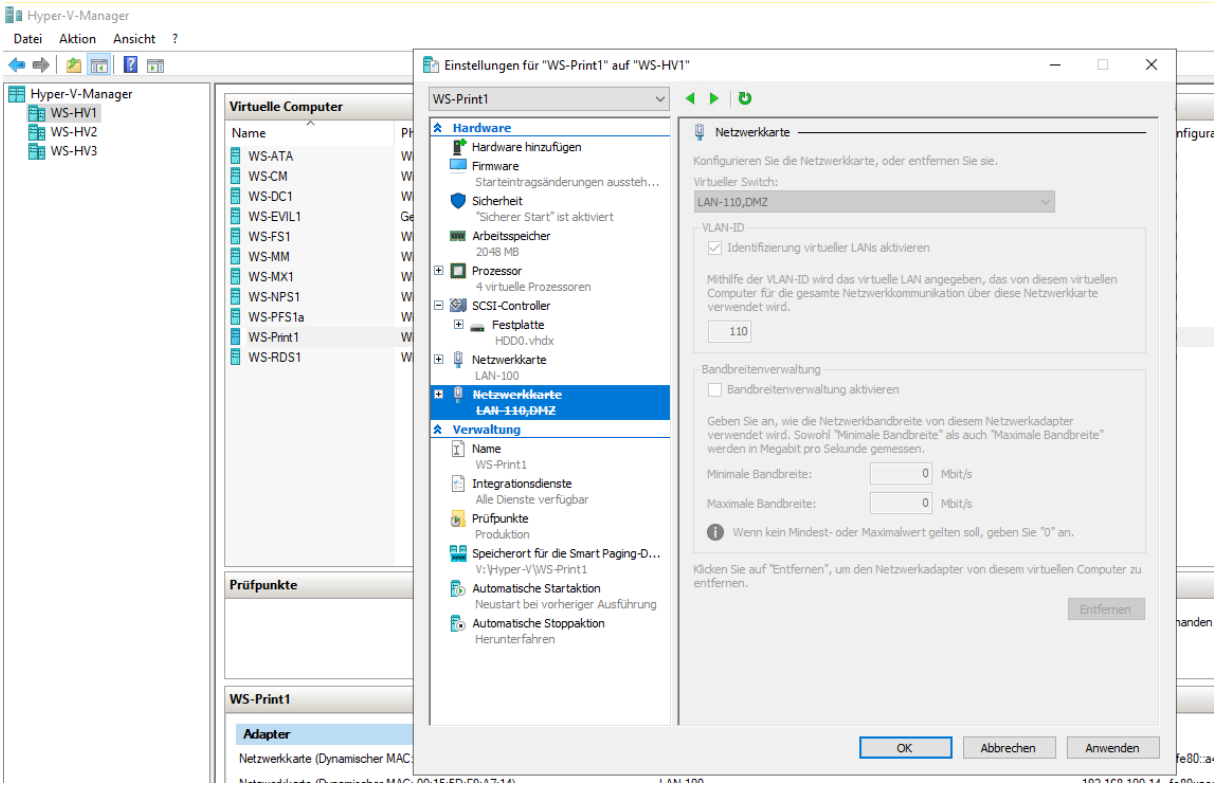
Dort sind alle Updates vorhanden und die Installation beginnt direkt nach dem Download:



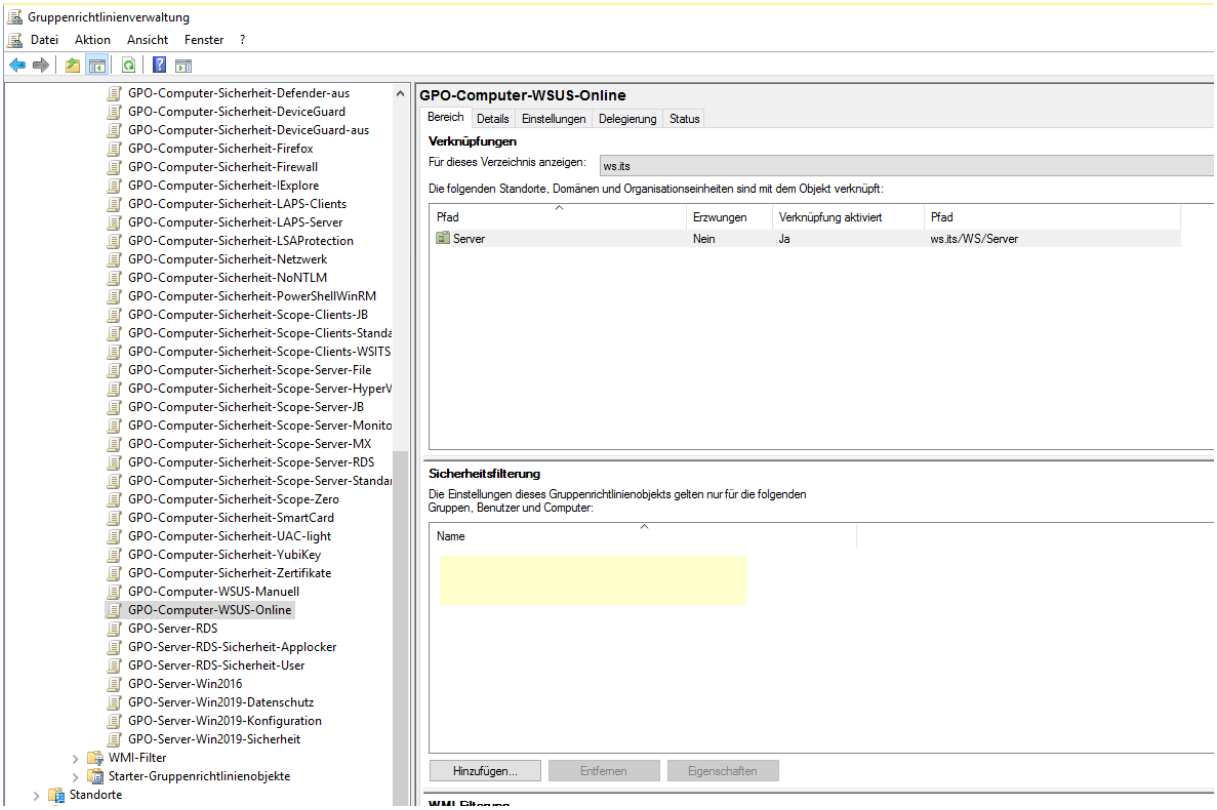
Wenige Minuten später ist wieder ein Neustart fällig:



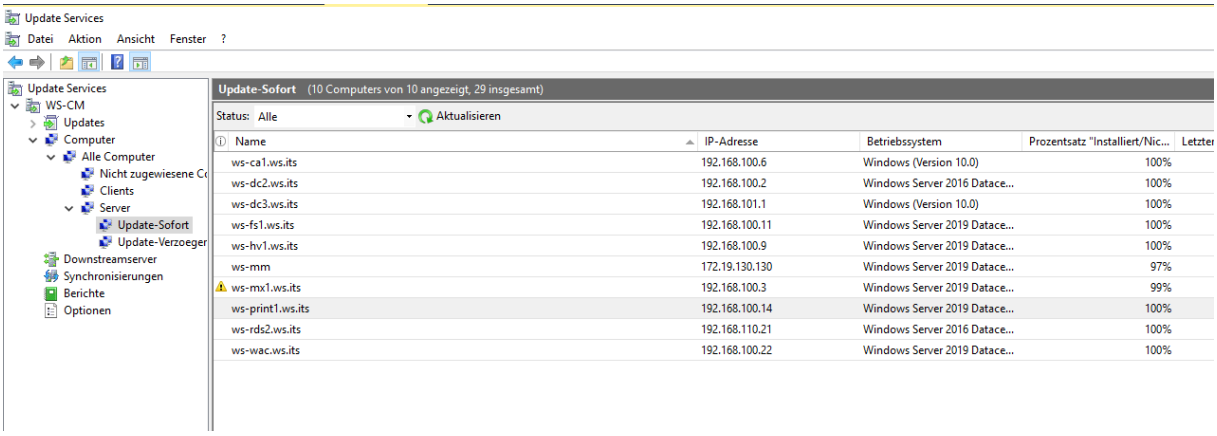
Nach dem Neustart suche ich erneut nach Updates. Da können durchaus Folgeupdates kommen. Hier ist aber alles gut. Daher klemme ich den Server wieder in das Server-Netz und stelle die statische IP-Adresse wieder ein:



Dann entferne ich ihn aus der Gruppenrichtlinie mit den Internet-Updates:



Ich suche erneut nach Updates. Dieses Mal geht der Server wieder zum WSUS. Dort prüft er sein Patchlevel mit dem der freigegebenen Updates. Er hat alles installiert. Und das meldet er an den WSUS. In der Konsole kann ich das prüfen:



Praxistipp:
Viele Administratoren verwenden vorbereitete Images oder VM-Templates, in denen das Betriebssystem jeden Tag weiter veraltet. Es lohnt sich, in regelmäßigen Abständen das Patchlevel der daraus erstellten Server zu verifizieren. Gerne kann dazu auch die Build-Nummer des Betriebssystems genutzt werden. Internetsuchmaschinen helfen hier schnell weiter. Wenn man dann eine Abweichung feststellt, dann sollte das Image bzw. Template gegen den öffentlichen Windows Update Service aktualisiert werden. Daraus kann dann schnell mit sysprep ein neues Template generiert werden. Das hält dann wieder einige Monate her.

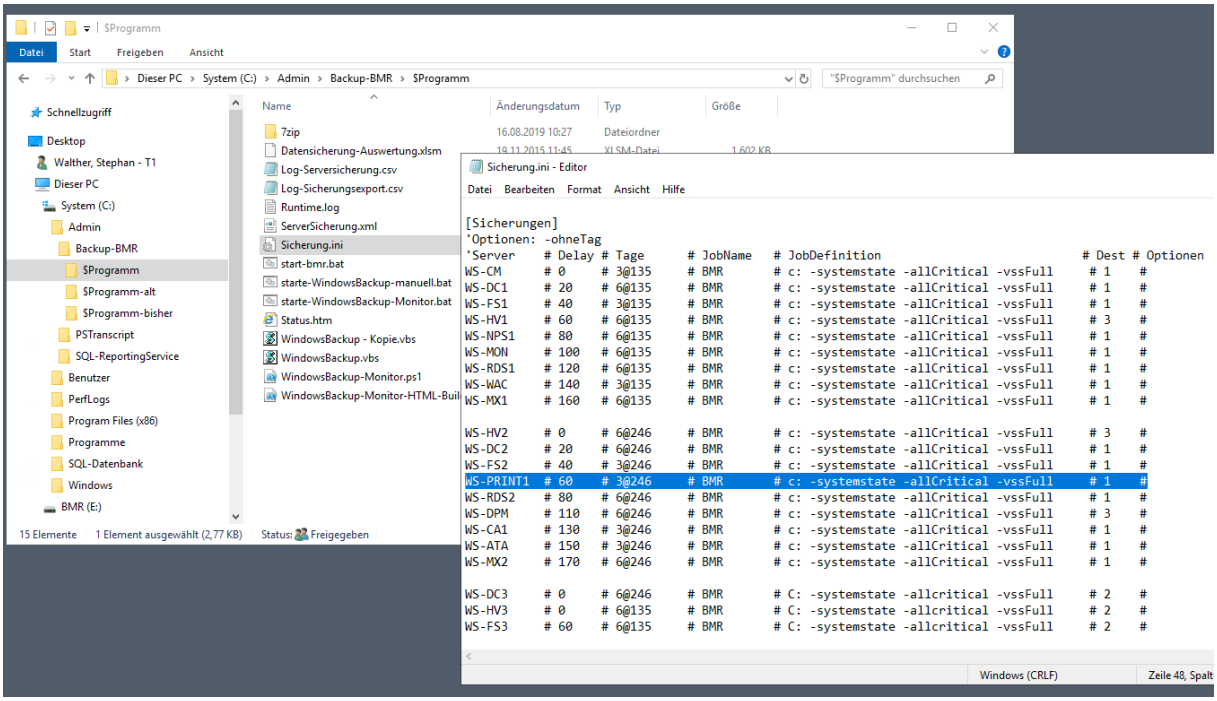
Wichtig ist auch, dass ein Verzicht auf die WSUS-Updatebereinigung keine Option ist. Es kam auch schon vor, dass Updates durch mehrfache Erneuerung in nicht auflösbare Abhängigkeitsschleifen geführt haben. Also immer schön im Detail prüfen und nicht einfach dem Windows Update in den Einstellungen glauben…

Datensicherung

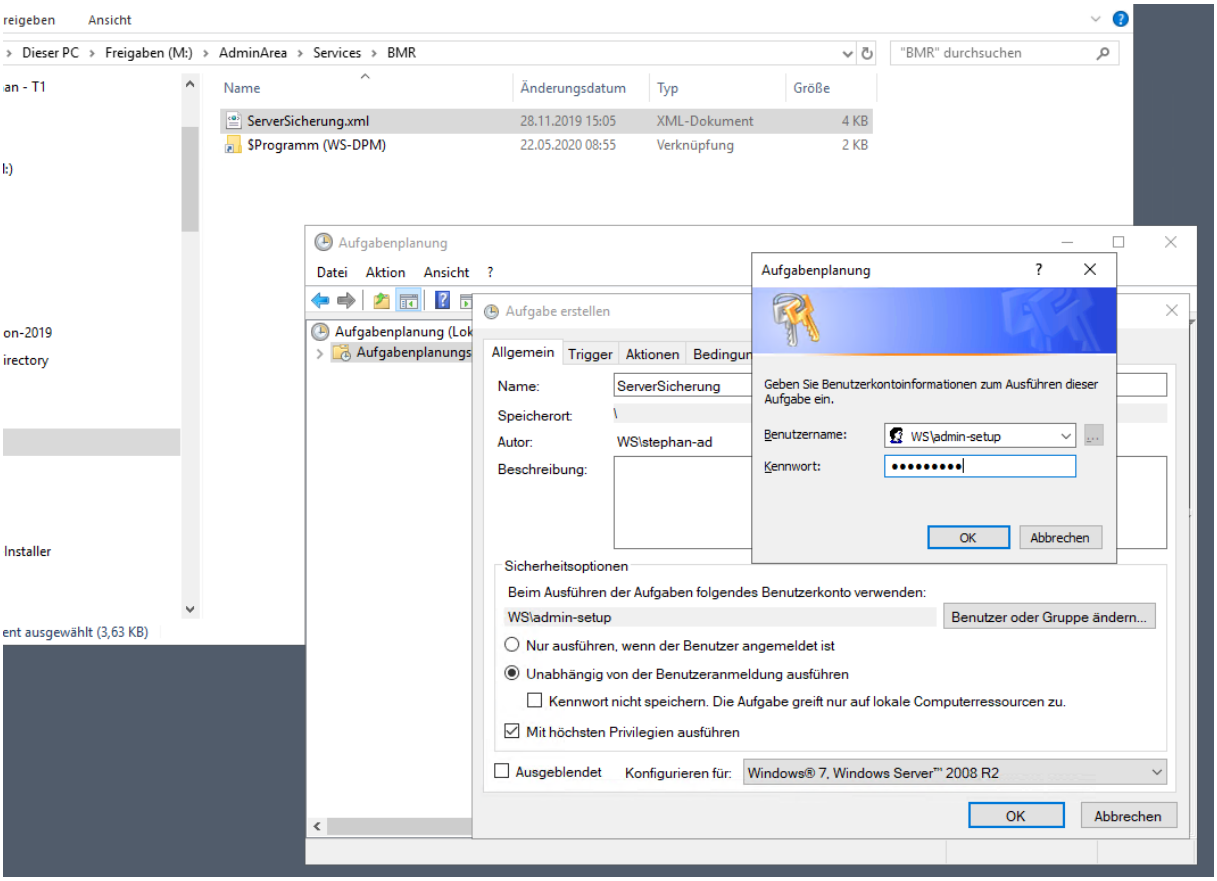
Bevor der neue Server in die Produktion geht, bekommt er eine Datensicherung. Hier verwende ich für das Betriebssystem das mitgelieferte Windows Server Backup. Dieses starte ich durch einen Script-Task. Das Script sucht dann in einer zentral abgelegten Konfigurationsdatei nach der Sicherungskonfiguration. Aus den Zeilen, die mit dem Namen des Servers beginnen wird dann ein wbadmin-Befehl gerendert und ausgeführt. Abschließend wird der Sicherungsstatus in der zentralen Freigabe protokolliert.

Zuerst trage ich in der Konfigurationsdatei einen Sicherungsjob für den neuen Server ein.

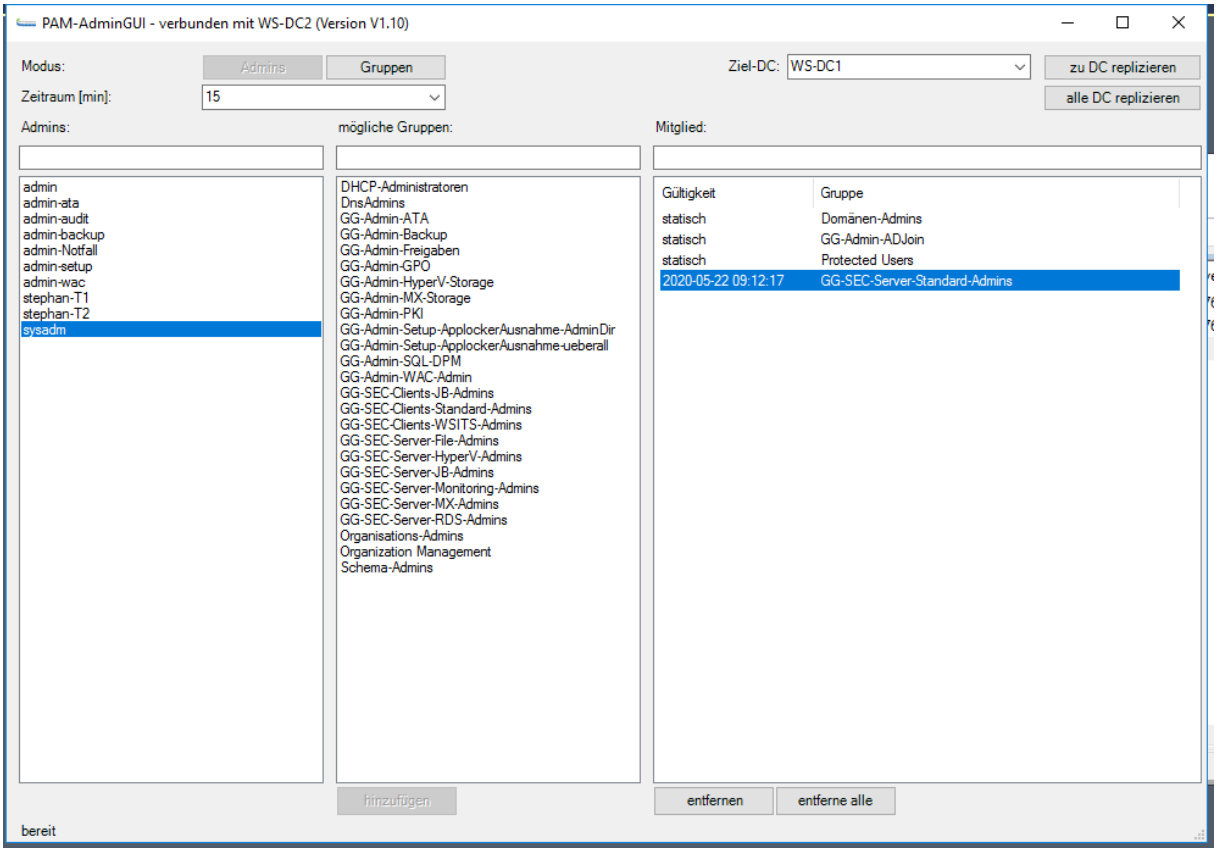
- Er soll 60 Minuten nach dem Task-Start beginnen (die Tasks starten auf allen Servern zur gleichen Zeit. So kann ich zentral durch einen Versatz die Reihenfolge anpassen).
- Es sollen 3 Sicherungen rotieren, die am Dienstag, Donnerstag und Samstag erstellt werden (3@246).
- Gesichert werden soll der Systemstate – also alles, was zum Betriebssystem gehört.
- Das Sicherungsziel ist die #1. Der dazugehörige Pfad wird weiter oben in der Konfiguration angegeben. Er zeigt auf eine geschützte SMB-Freigabe.



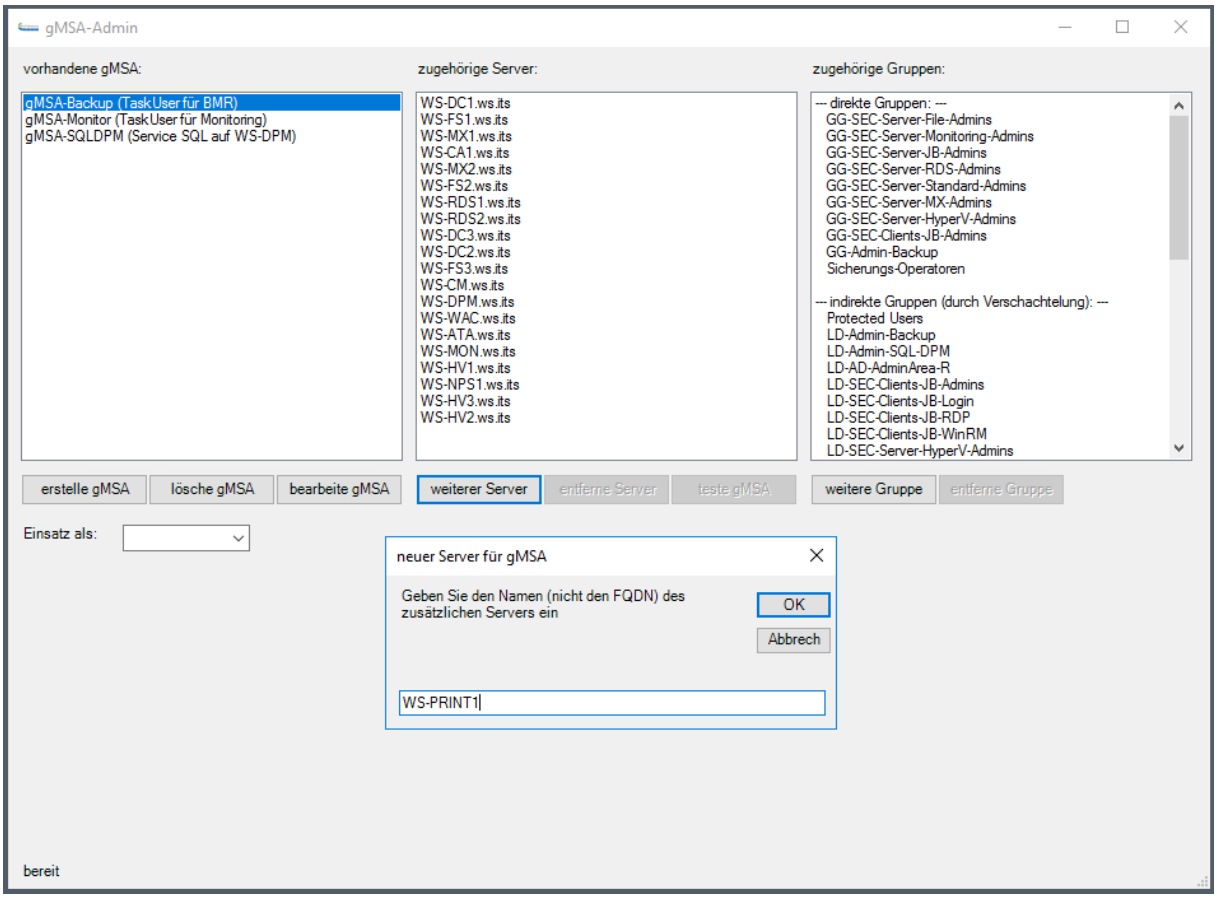
Jetzt importiere ich den Sicherungstask mit einer XML-Datei in die Aufgabenplanung des Servers. Dabei muss ich einen Dummy-Account angeben:



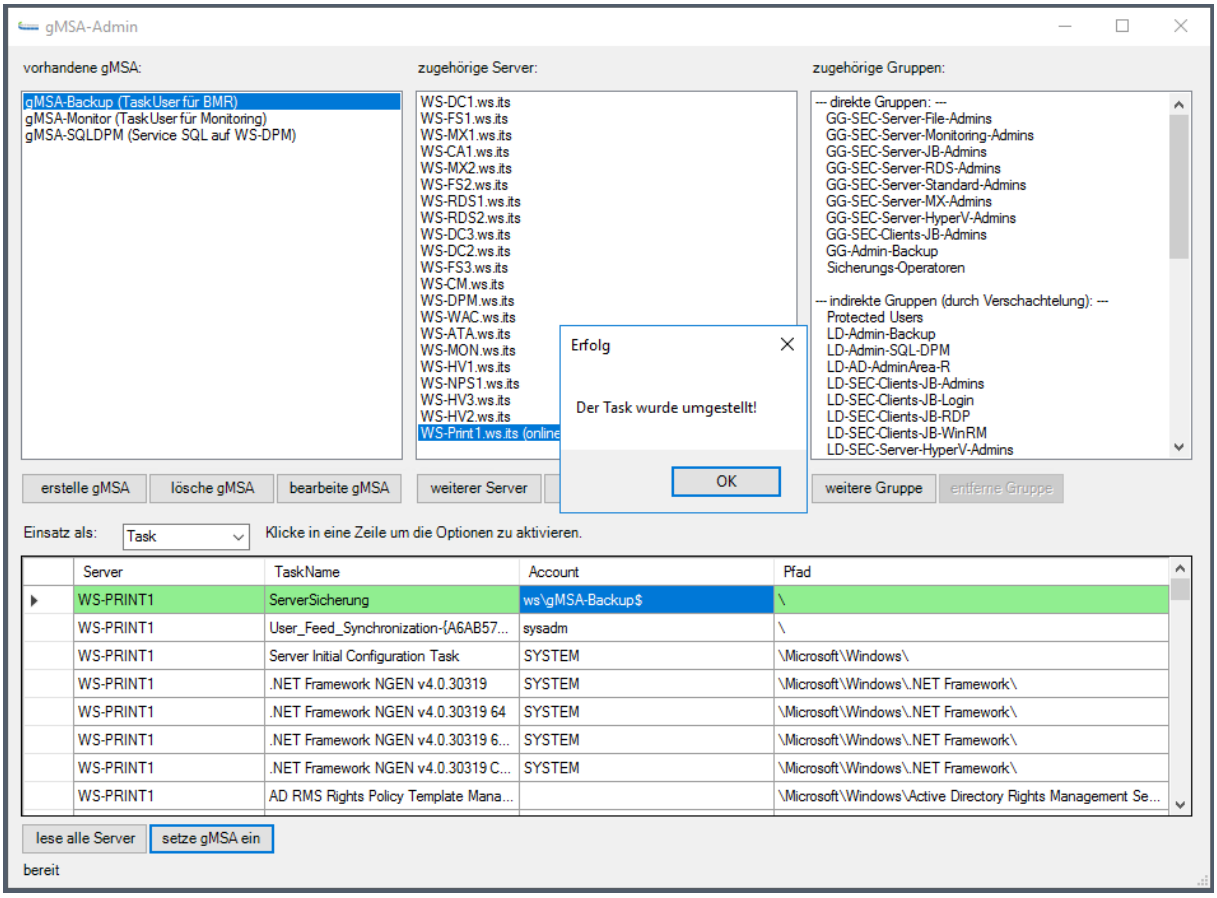
Der eigentliche Account für die Sicherung ist ein Group Managed Service Account. Diesen Accounttypen kann man aber auch unter Windows Server 2019 nicht in der Aufgabenplanung konfigurieren. Das geht nur mit der PowerShell. Auf meinem Domain Controller habe ich ein Script dafür abgelegt. Zuvor muss ich meinen Administrator-Account noch für den Printserver berechtigen (meine Domain Admins haben KEINE Rechte auf Memberservern und Clients):



Jetzt kann ich mein gMSA-PowerShell-Script mit einer GUI starten und dem neuen Server den Zugriff auf die Anmeldedaten des ServiceAccounts erlauben:



Danach verbindet sich das Script mit dem neuen Printserver und ich kann die dort registrierten Aufgaben und Dienste editieren. So stelle ich den Task-Account auf den gMSA-Account um:



Ich kontrolliere einige Tage später den Sicherungserfolg. Der Scripttask protokolliert das Ergebnis zentral in einer CSV-Datei. Ein anderes PowerShell-Script wertet die Daten der CSV täglich aus und informiert mich per Mail:

Serversicherung:

Server	JobName	StartZeit	EndZeit	Groesse	Status	Zeitplan	Slot
WS-WAC	BMR	--	--	0	OK	135	--
WS-RDS1	BMR	--	--	0	OK	135	--
WS-MX1	BMR	--	--	0	OK	135	--
WS-HV3	BMR	--	--	0	OK	135	--
WS-FS3	BMR	--	--	0	OK	135	--
WS-MON	BMR	--	--	0	OK	135	--
WS-CM	BMR	--	--	0	OK	135	--
WS-DC1	BMR	--	--	0	OK	135	--
WS-FS1	BMR	--	--	0	OK	135	--
WS-NPS1	BMR	--	--	0	OK	135	--
WS-HV1	BMR	--	--	0	OK	135	--
WS-HV2	BMR	01:00:01	01:04:52	27042	OK	246	5
WS-DC3	BMR	01:00:04	01:23:12	29442	OK	246	6
WS-DC2	BMR	01:20:03	01:31:06	35386	OK	246	3
WS-FS2	BMR	01:40:02	01:47:44	22558	OK	246	3
WS-PRINT1	BMR	02:00:01	02:06:46	17573	OK	246	3
WS-RDS2	BMR	02:20:02	02:58:57	44091	OK	246	3
WS-DPM	BMR	02:50:03	02:58:54	33638	OK	246	2
WS-CA1	BMR	03:10:03	03:15:06	13946	OK	246	2
WS-ATA	BMR	03:30:00	03:44:34	42322	OK	246	1
WS-MX2	BMR	03:50:00	04:08:44	58134	OK	246	4

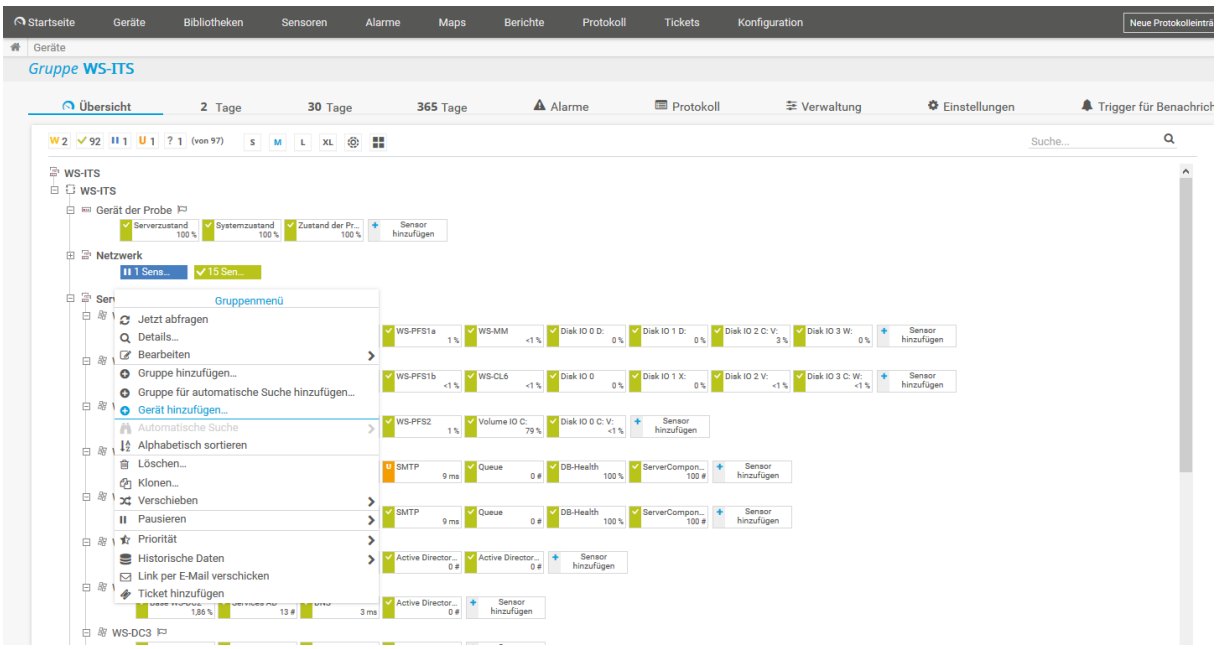
Knapp 7 Minuten für 17,5GB
Es wurde bereits der dritte Slot beschrieben. Das bedeutet mind. 3 erfolgreiche Sicherungen.

Statistik:

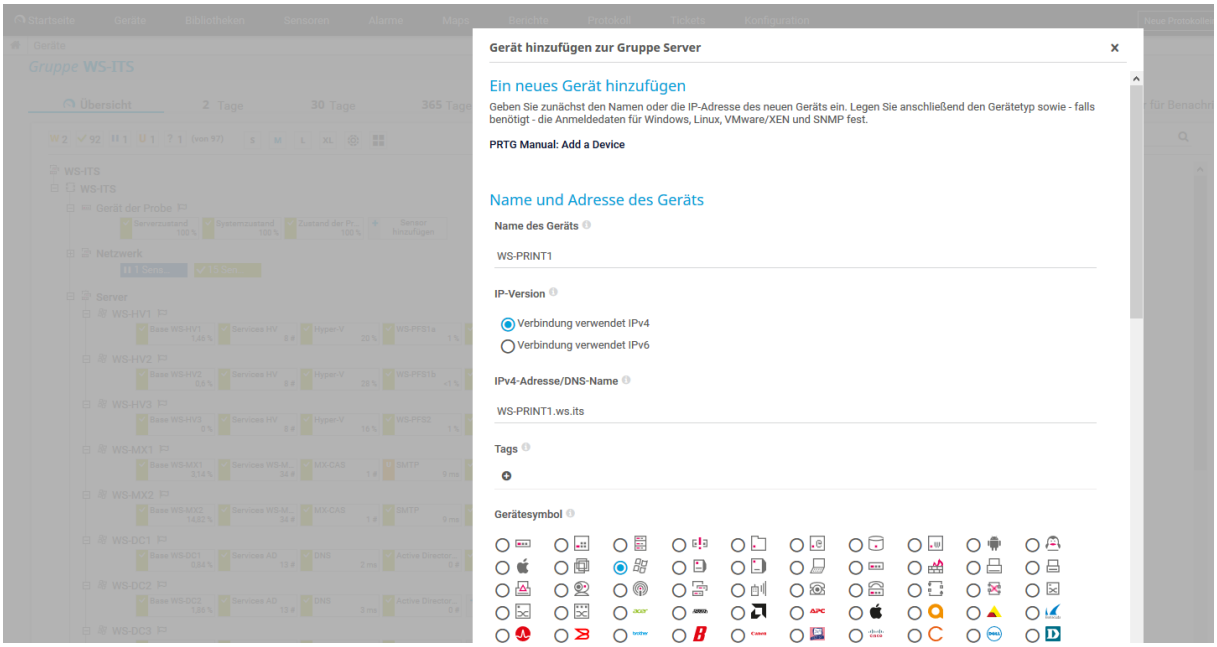
Sicherungsvolumen [MB]:	324132
Sicherungsdauer [min]:	189
Dauer effektiv [min]:	140
Geschwindigkeit [MB/min]:	2315

Monitoring

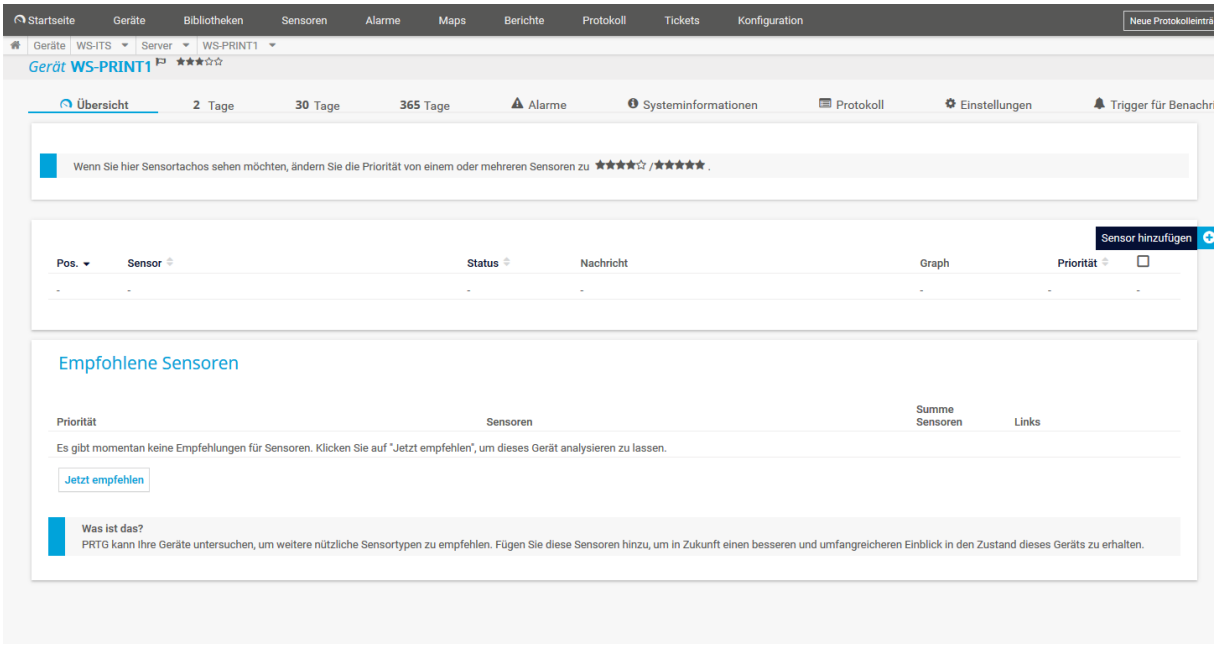
Natürlich darf der neue Server auch im Monitoring nicht fehlen. Hier setze ich auf PRTG. In der Konsole erstelle ich einen neuen Eintrag für den Printserver:



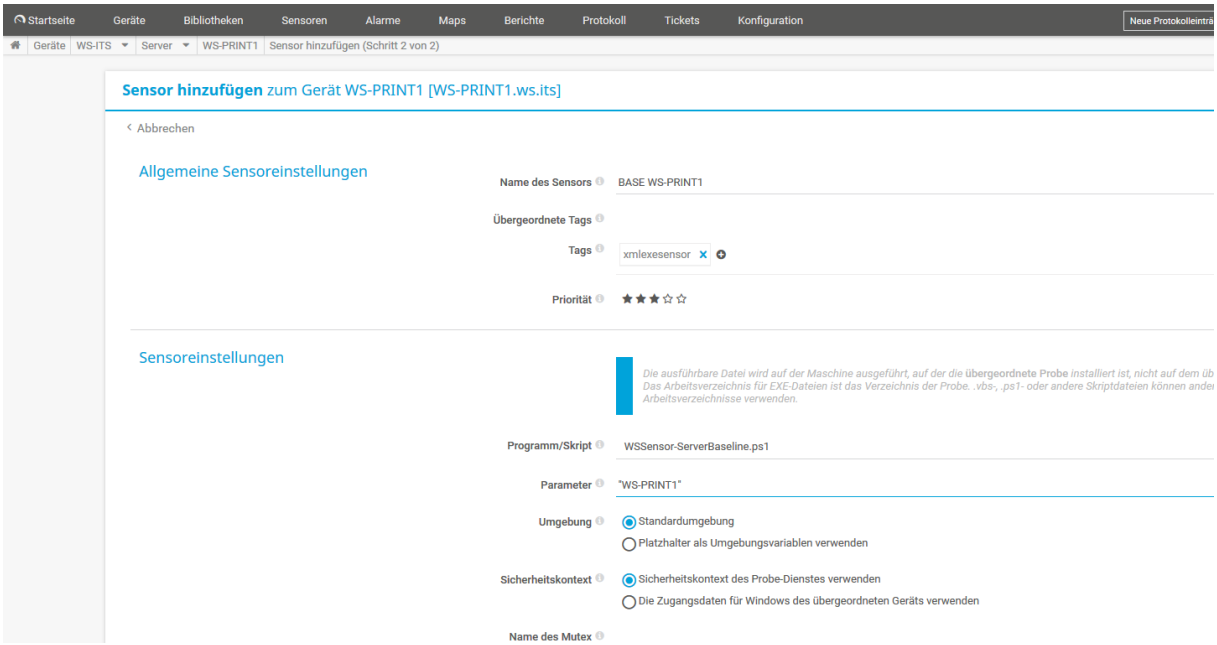
Dann muss ich den Namen und die Adresse hinterlegen. Da verwende ich den FQDN:



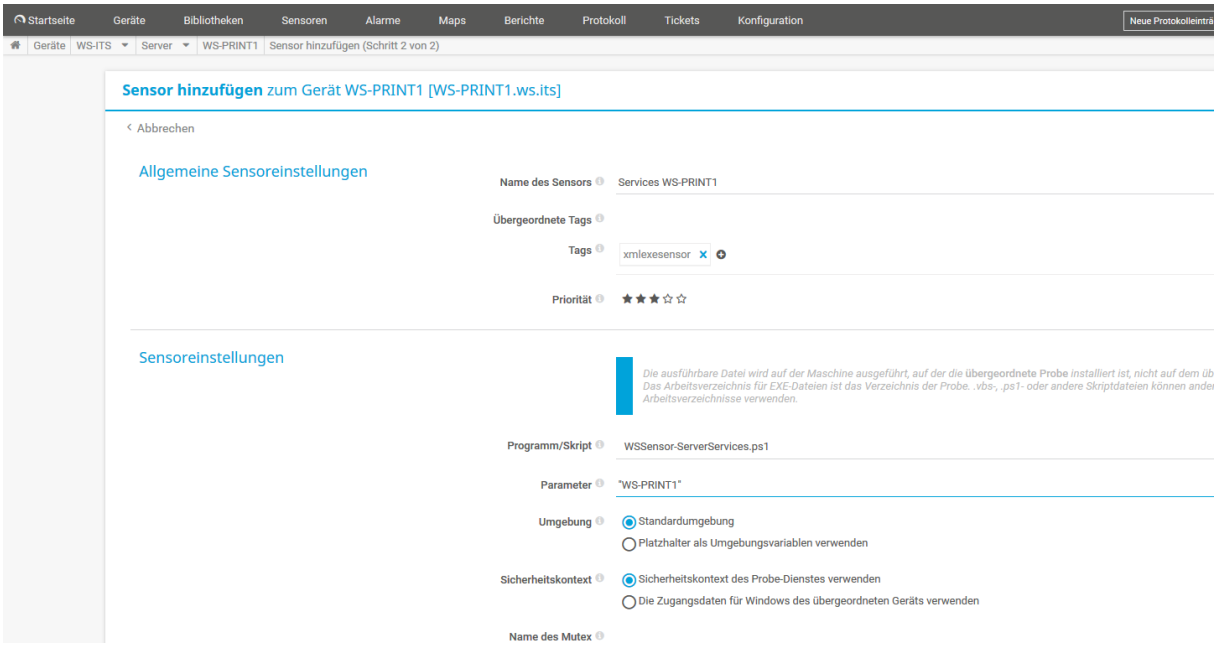
In dem neuen Container kann ich nun Sensoren eintragen:



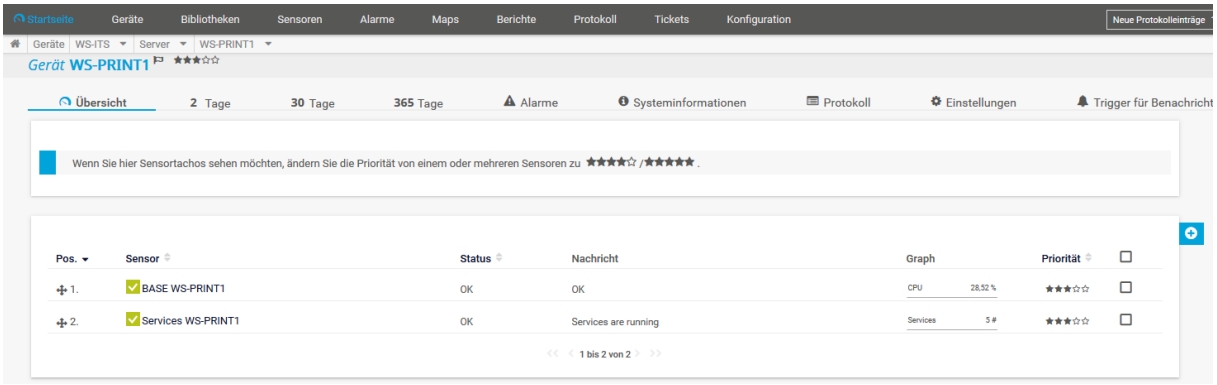
Ich habe 2 eigene PowerShell-Scripte, die der PRTG regelmäßig ausführen soll. Das erste Script ermittelt Basisinformationen vom Windows Server – also die aktuelle CPU-Belastung, den RAM-Gebrauch, den Füllstand der Volumes und den Netzwerk-Traffic. Aus dem Dropdown-Feld wähle ich mein Script „WSSensor-ServerBaseline“. Dann konfiguriere ich die Parameter. Hier muss noch einmal der Servername angegeben werden:



Der zweite Sensor überwacht die relevanten Serverdienste. Dazu zählen eine Reihe von Standarddiensten, die durch Angabe eines Parameters spezialisiert werden können. Das Script „WSSensor-ServerServices“ wähle ich im Dropdown-Feld aus und gebe als Spezialisierung „Print“ im Parameter mit an:



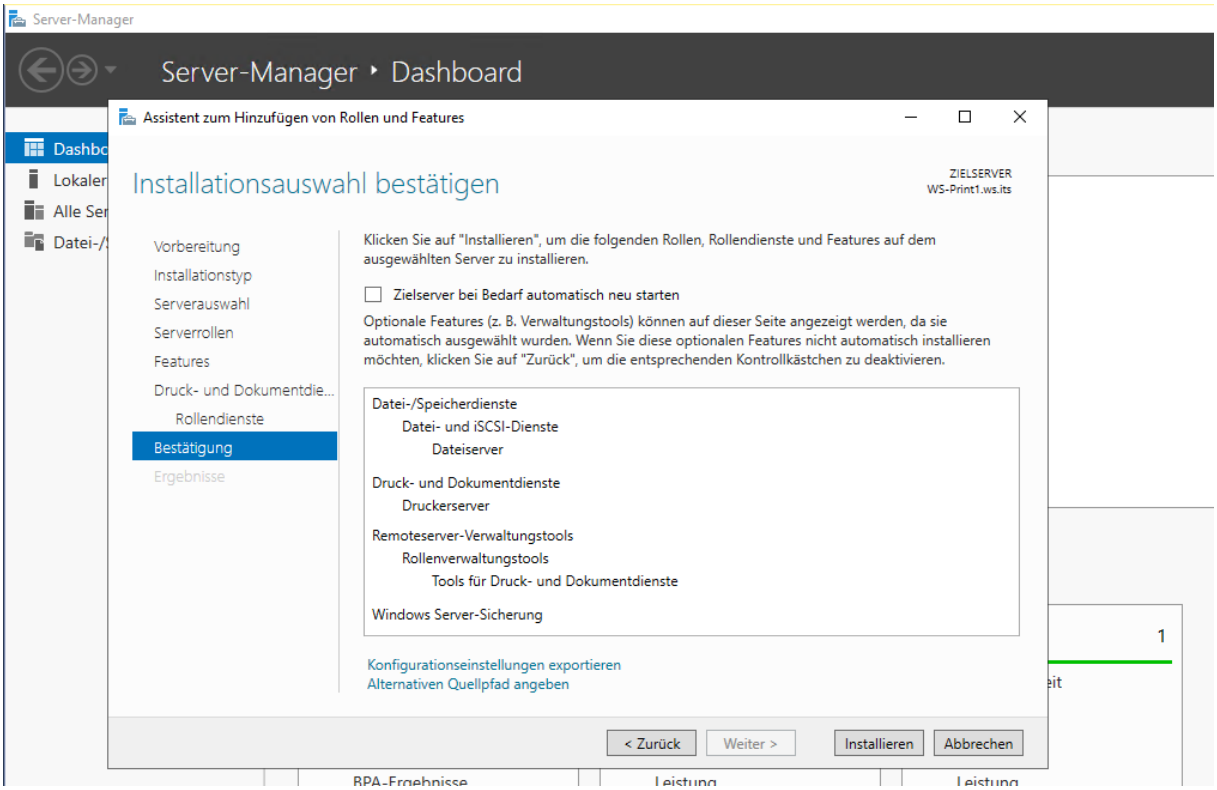
Nach wenigen Sekunden wurden die Scripte ausgeführt und der Server wird überwacht:



Konfiguration Printserver und ScanServer

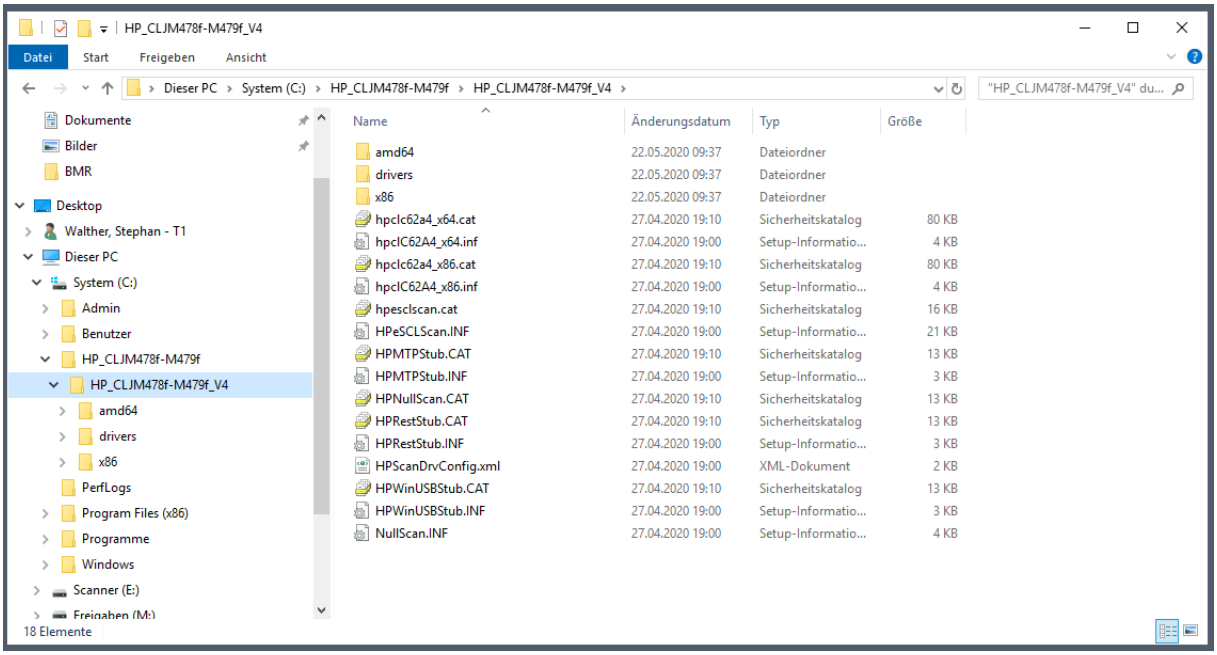
Rolleninstallation

Das Basis-Betriebssystem ist fertig. Jetzt kommt die eigentliche Konfiguration der Serverfunktion. Ich starte mit der Installation der erforderlichen Rollen und Features. Der Server soll als Printserver arbeiten und zusätzlich eine Dateifreigabe für das Scan2SMB anbieten. Dazu kommt noch das Feature der Windows Server Sicherung (sonst wird das nichts mit der Datensicherung):

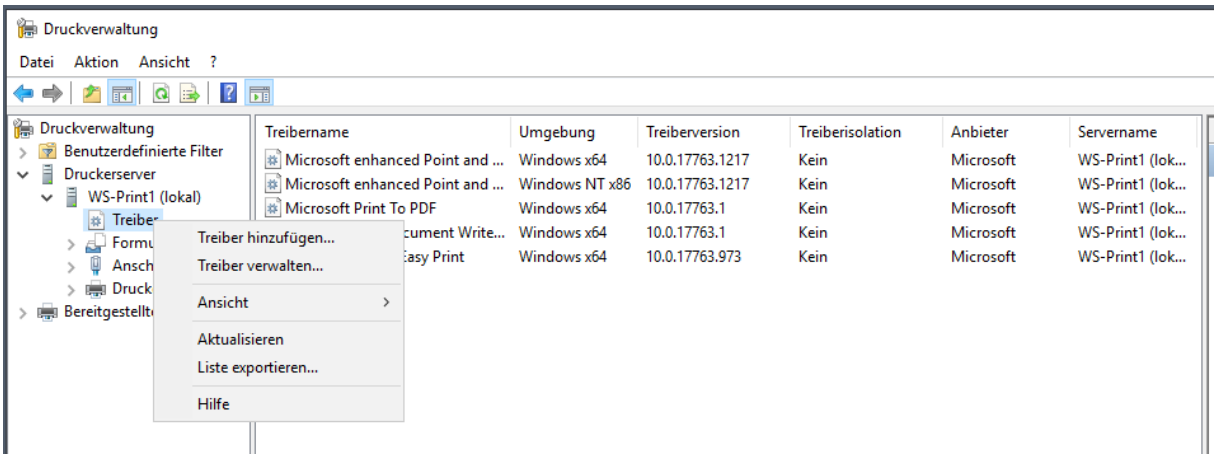


Konfiguration als Printserver

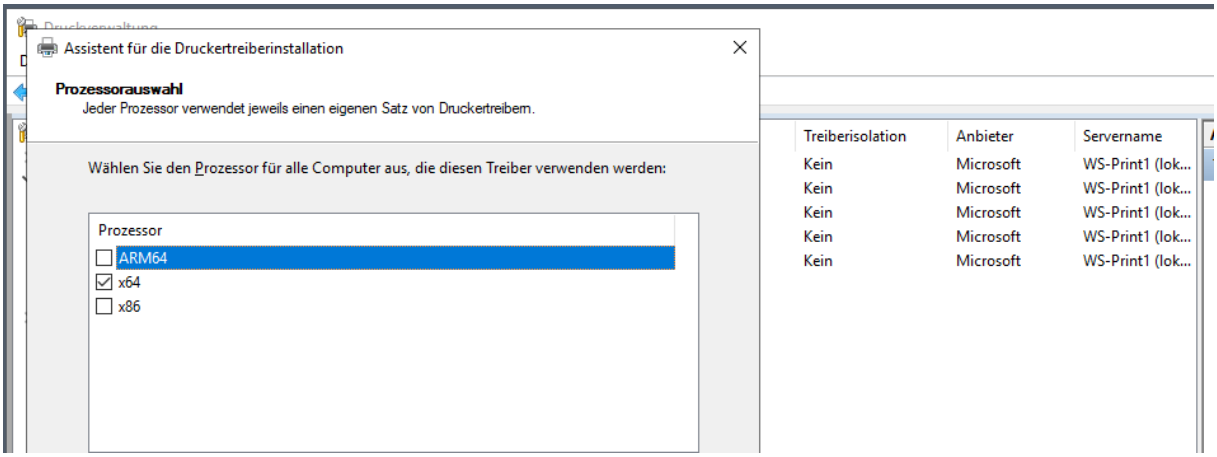
Ich lade mir den passenden Treiber für meinen neuen Drucker beim Hersteller herunter und packe die Dateien auf dem Printserver aus:



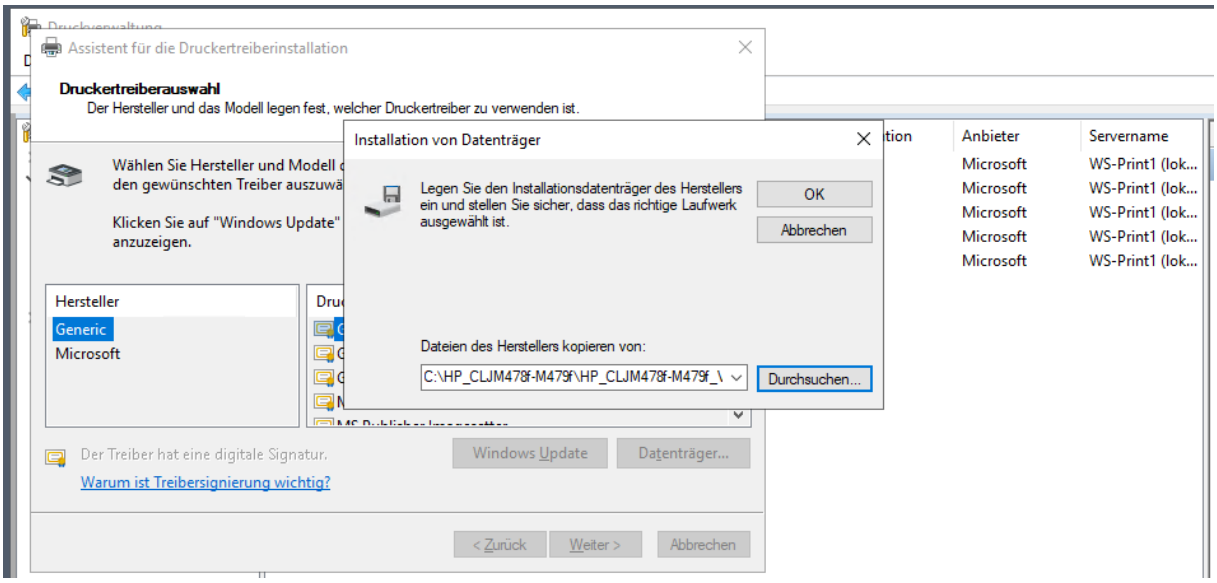
In der Druckverwaltung füge ich den Treiber nun hinzu:



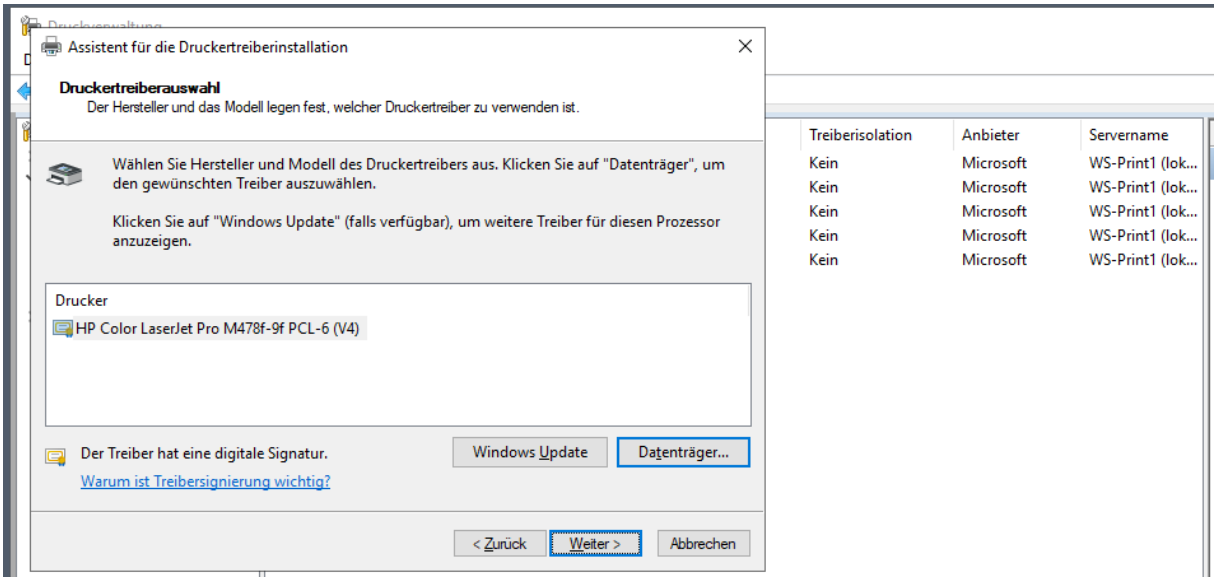
Ich verwende ausschließlich x64-Betriebssysteme. Zudem ist dieser Treiber nicht x64-kompatibel:



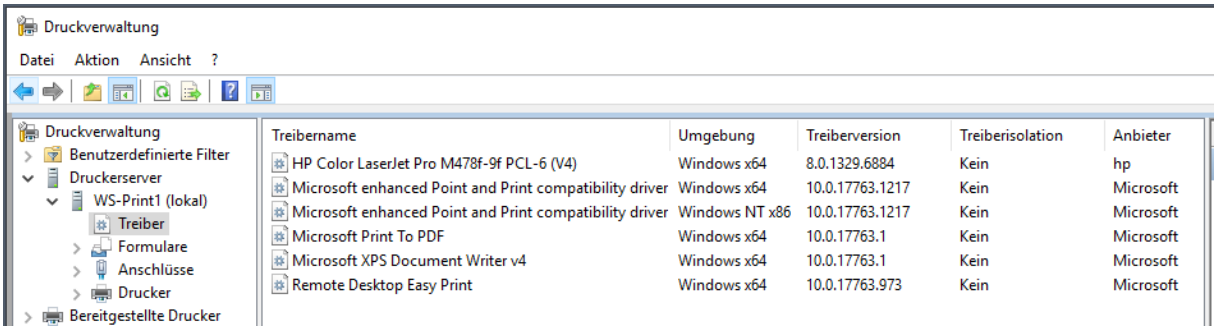
Jetzt verweise ich den Assistenten auf das Verzeichnis:



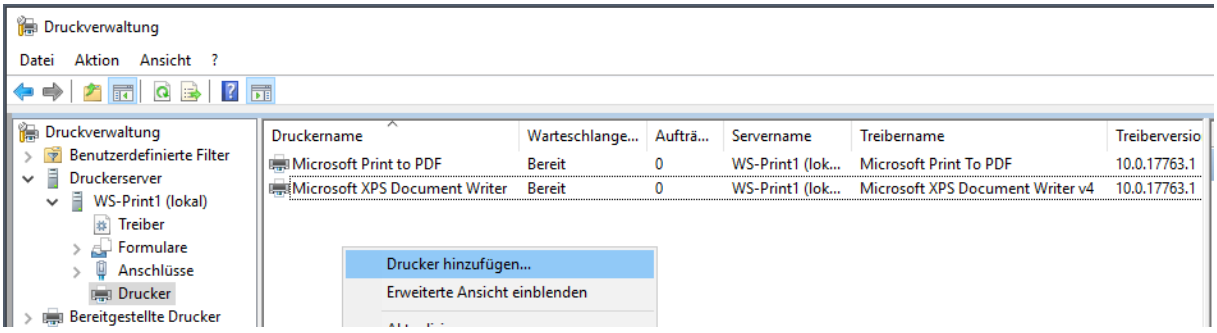
Darin ist nur ein Druckmodell enthalten. Und das passt mit meinem Gerät überein:



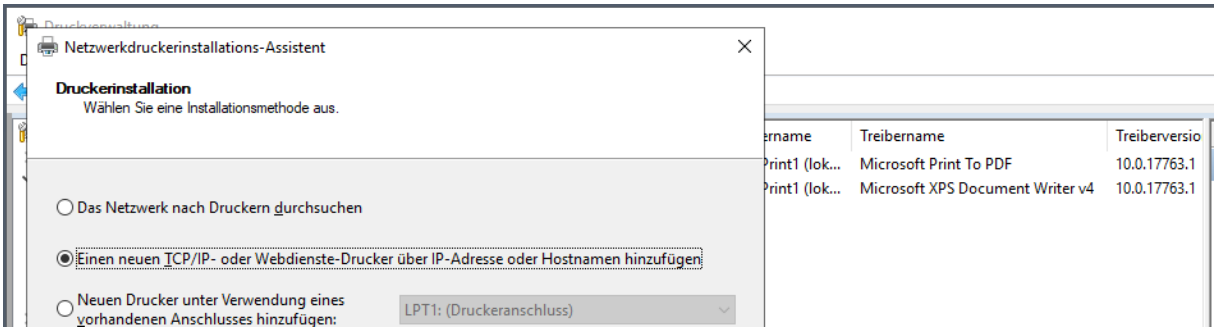
Danach ist der Treiber im Repository vorhanden:



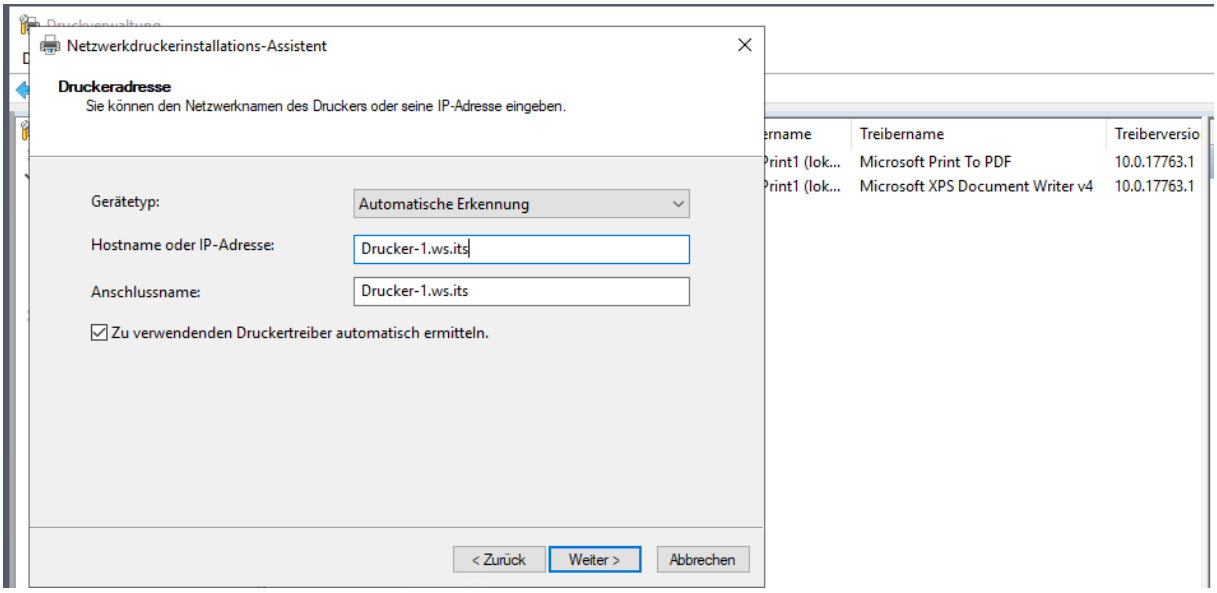
Jetzt verbinde ich das Druckgerät:



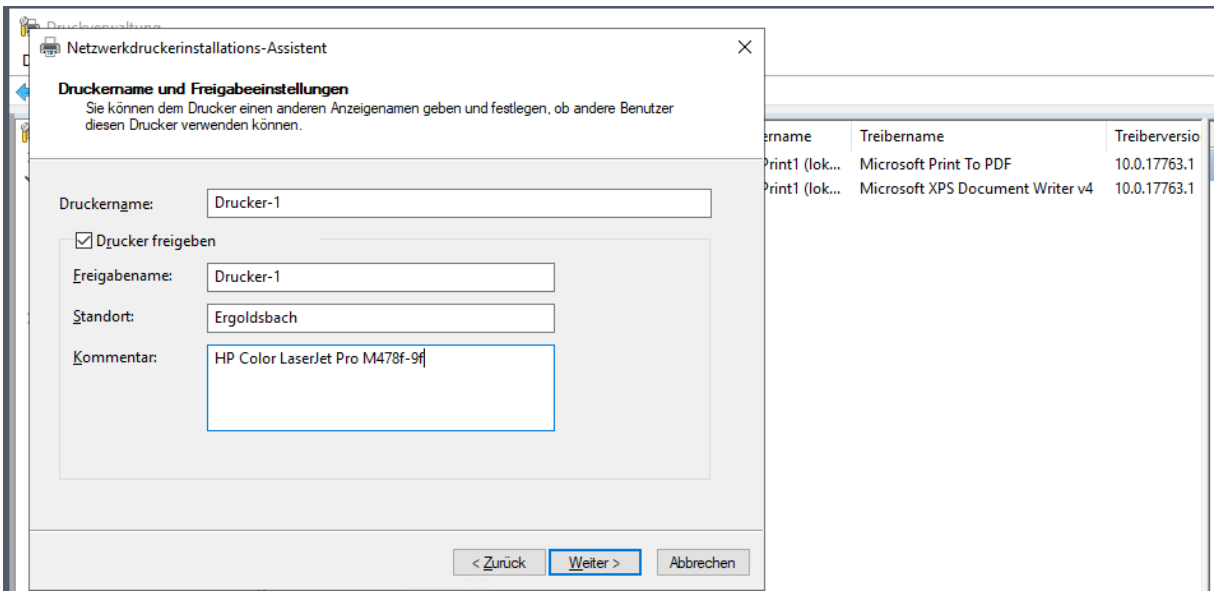
Der Drucker ist über eine Ethernet-Schnittstelle direkt erreichbar:



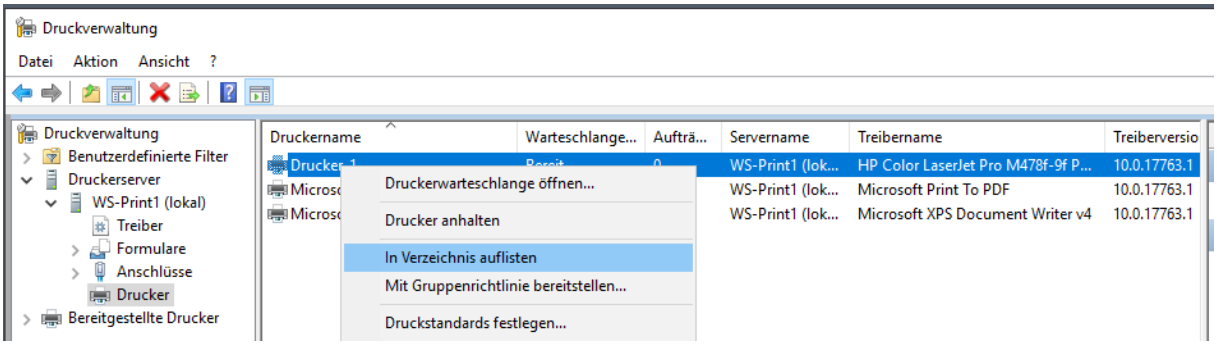
Natürlich habe ich für den Drucker einen HOST-A-Record im DNS erstellt. Ich kann mich hier also auf seinen FQDN beziehen:



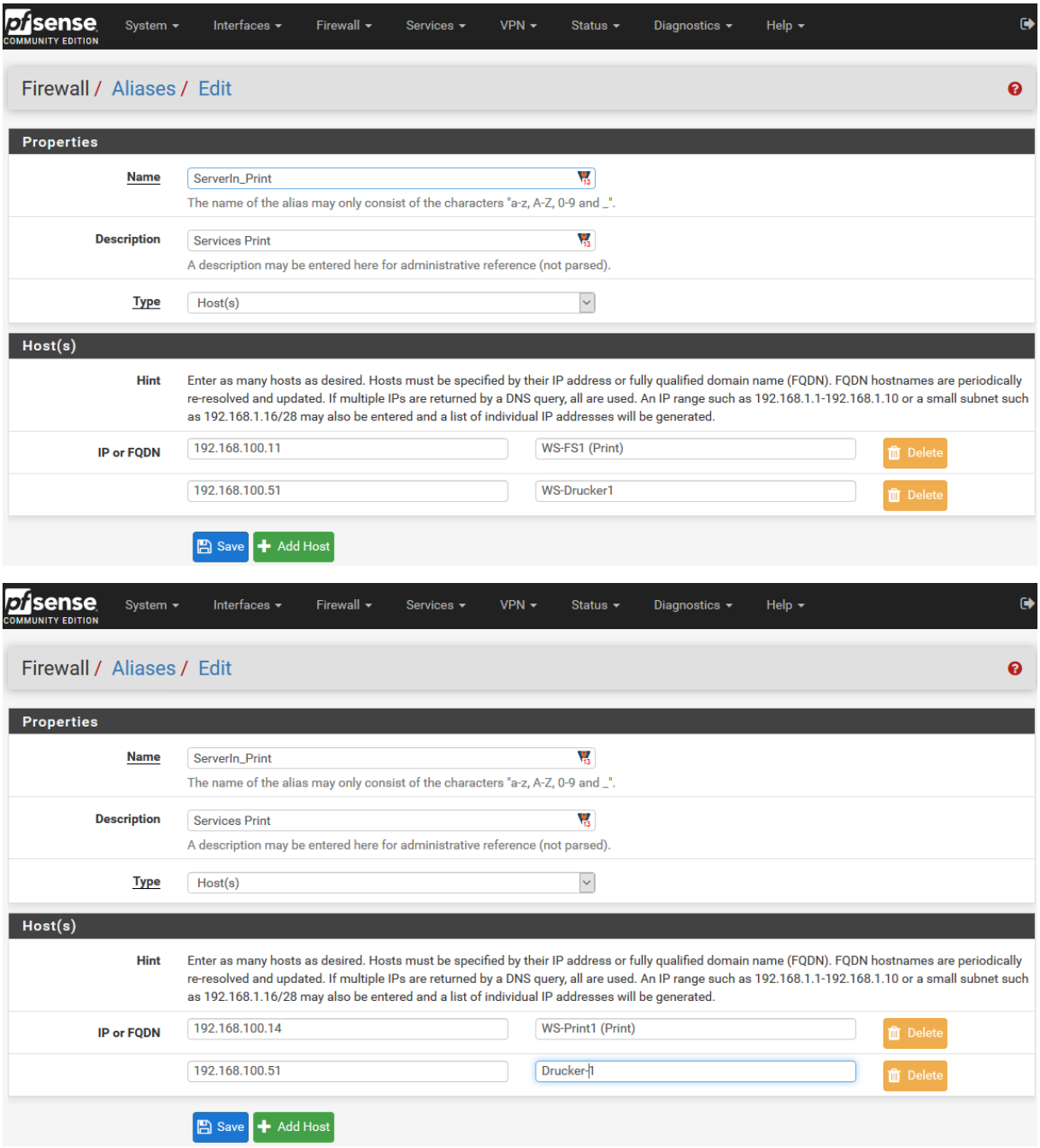
Dann gebe ich noch ein paar Zusatzinformationen ein und lasse auch gleich die Freigabe erstellen:



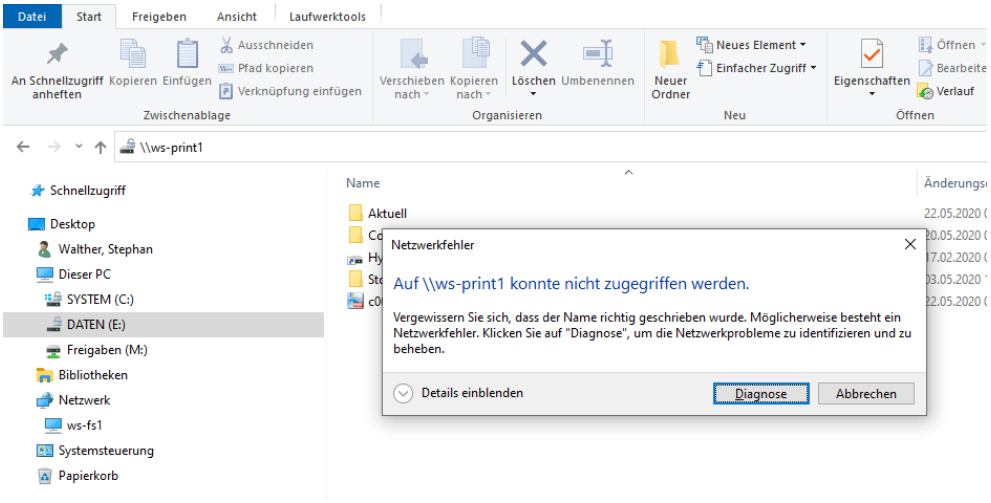
Der Drucker soll auch im Active Directory gesucht werden können. Das vereinfacht die Suche bei vielen Anwendungen:



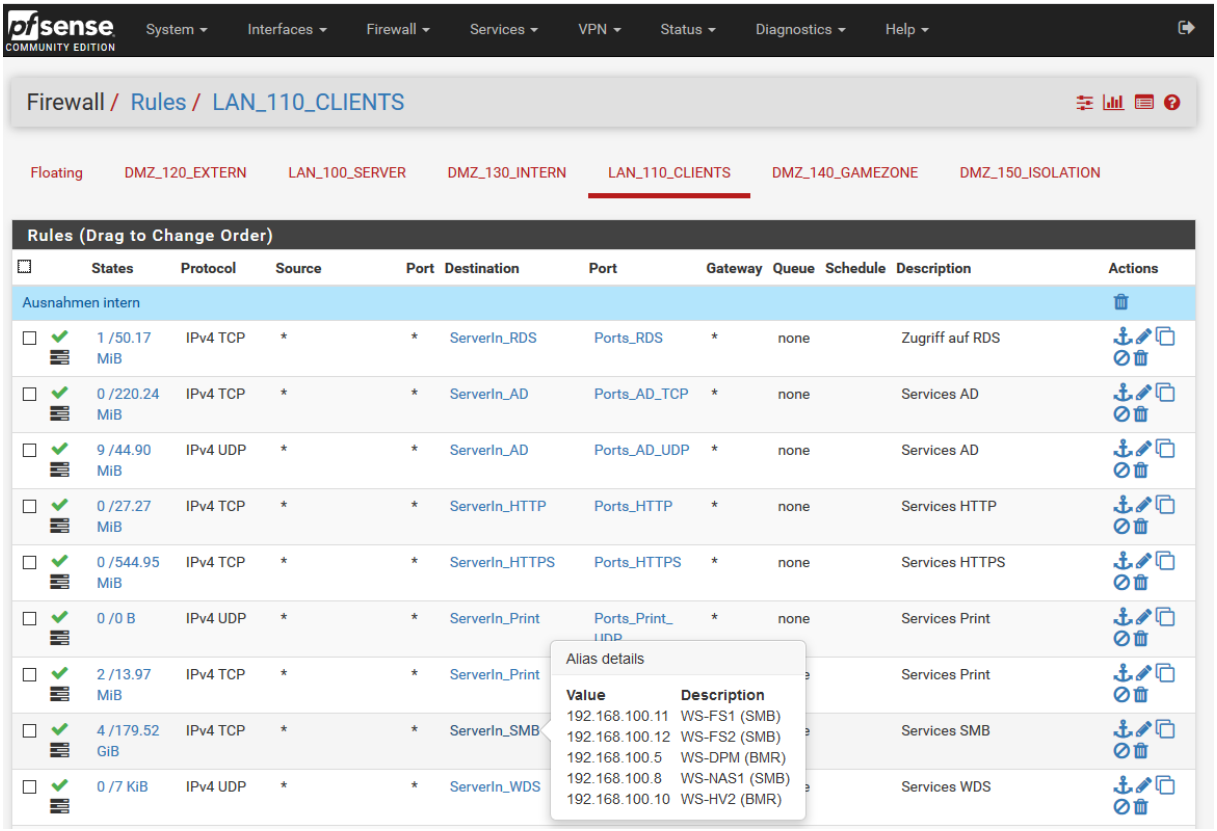
Ab jetzt können sich meine Clients mit der Druckerfreigabe verbinden und dann über den neuen Printserver Druckjobs an den Drucker senden. Theoretisch. Denn meine Firewall zwischen dem Client- und Servernetz hat da auch ein Wort mitzureden. Diese filtert alles, bis ich eine entsprechende Ausnahme definiere. Für den Druckservice hatte ich bisher eine Ausnahme zu meinem Fileserver, denn dieser stellte ja zusätzlich die Druckdienste bereit. Jetzt ändere ich die IP-Adresse im Alias der Ausnahme ab:



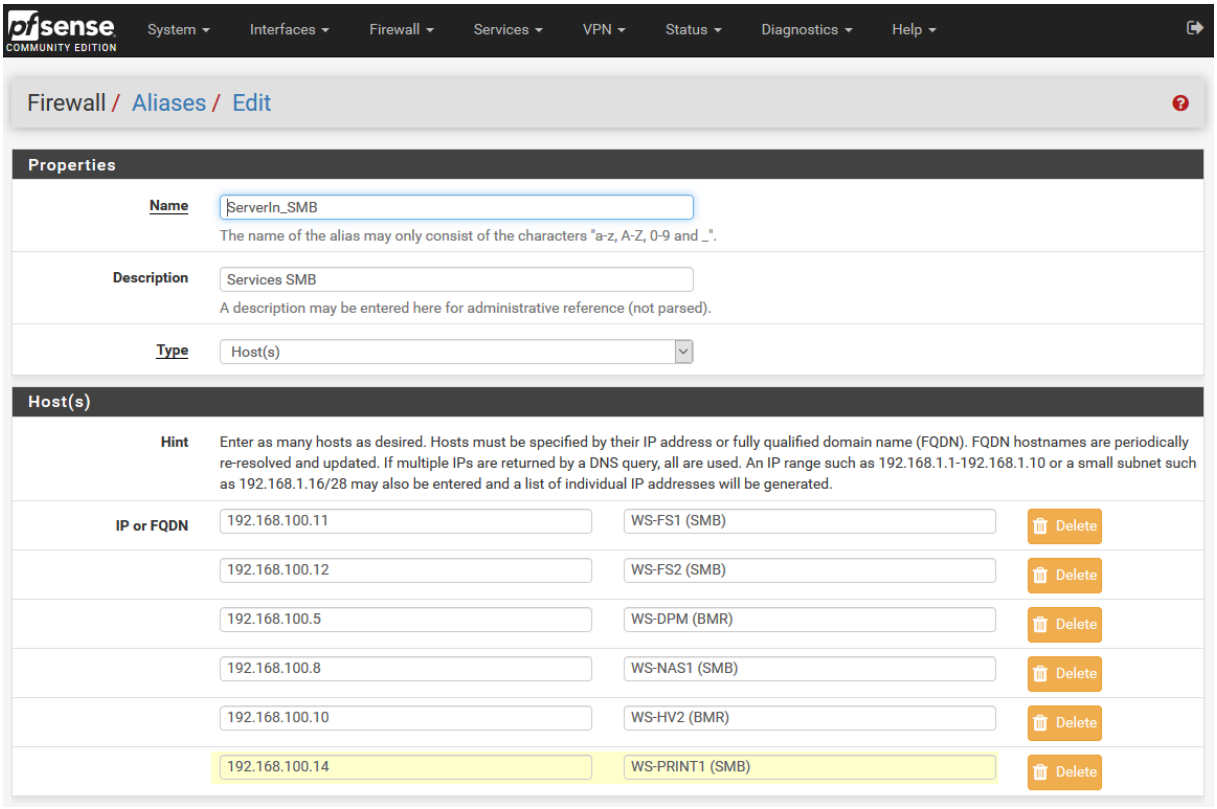
Ein Test vom Client aus schlägt aber immer noch fehl, wenn ich eine SMB-Verbindung starte. Drucken ist eben nicht das gleiche wie SMB:



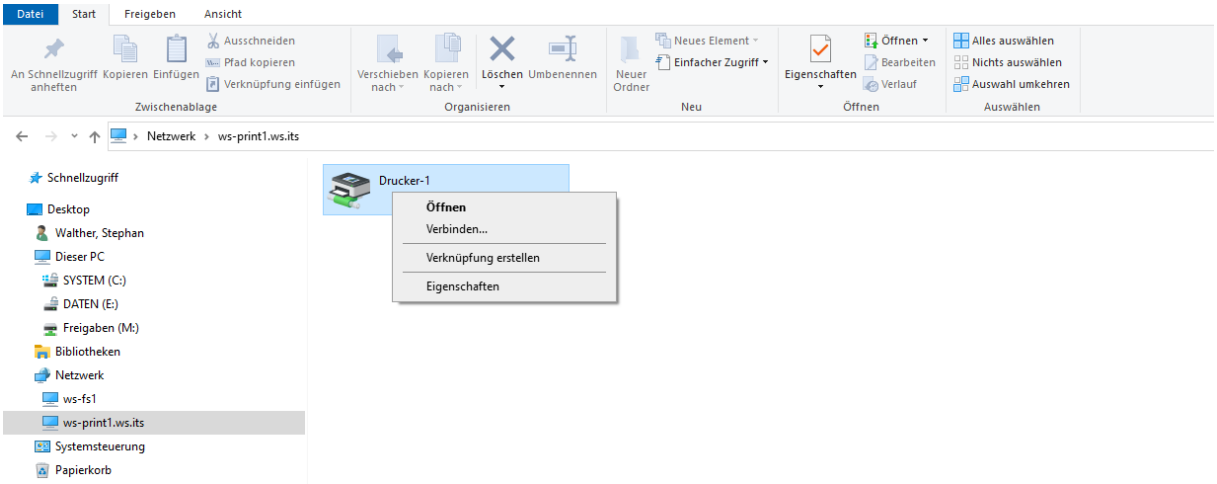
Ein Blick in die Firewall-Regeln zeigt, warum die Verbindung nicht aufgebaut werden kann:



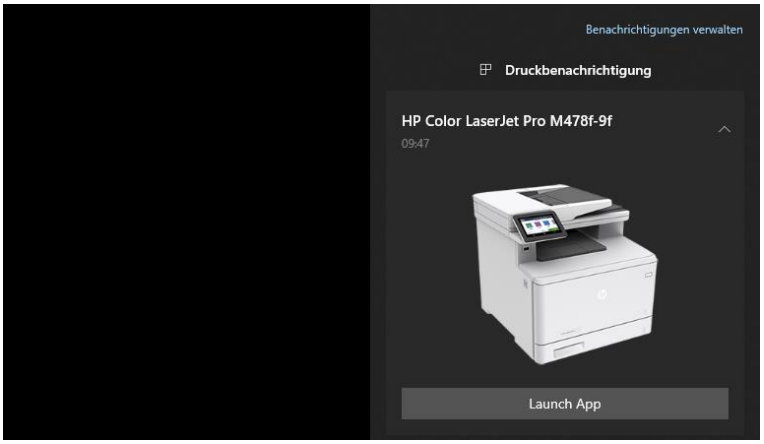
Aktuell sind für den Druckserver auch nur TCP-Ports freigegeben, die für das Drucken erforderlich sind. Ich nehme seine IP-Adresse mit in den Alias für die SMB-Server auf:



Jetzt kann sich mein Client mit dem Druckserver verbinden und dort den Drucker installieren:

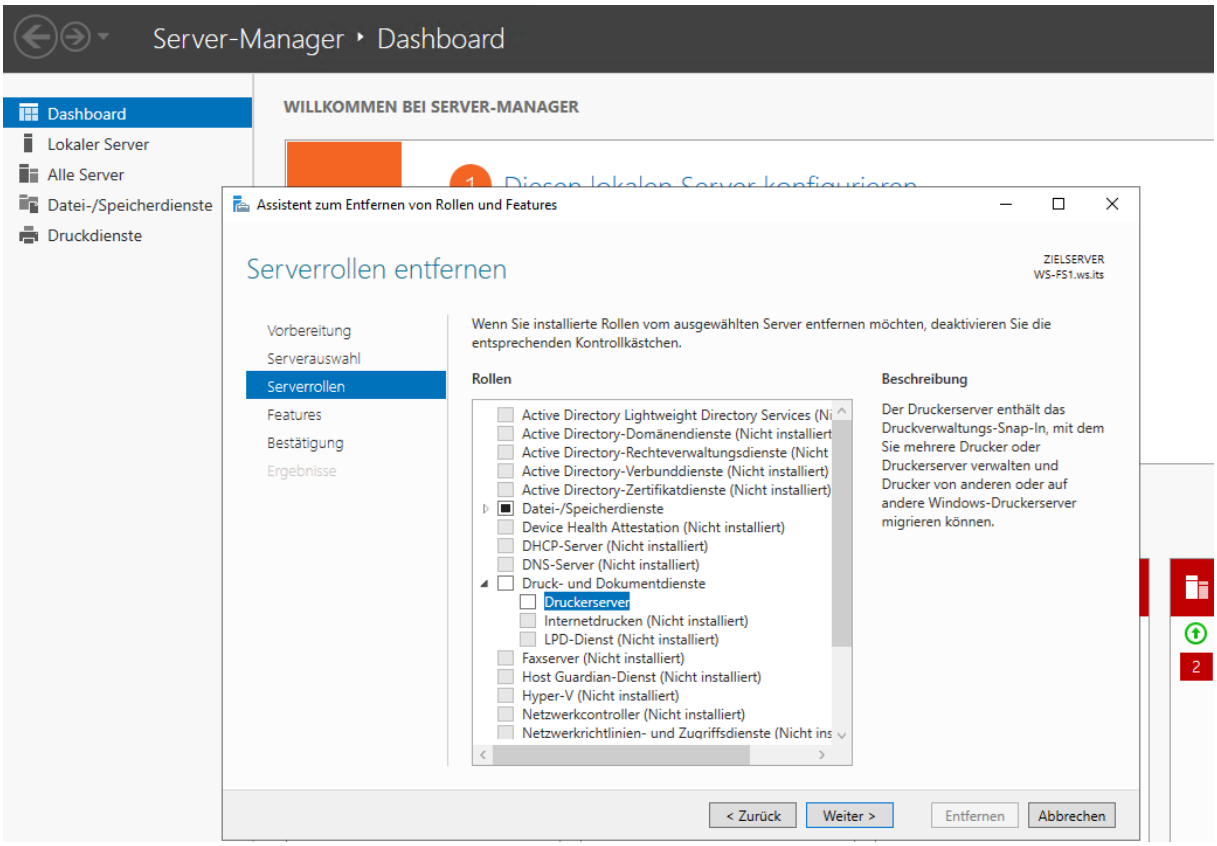


Kurz darauf taucht im Benachrichtigungscenter meines Windows 10 Clients der Drucker auf:

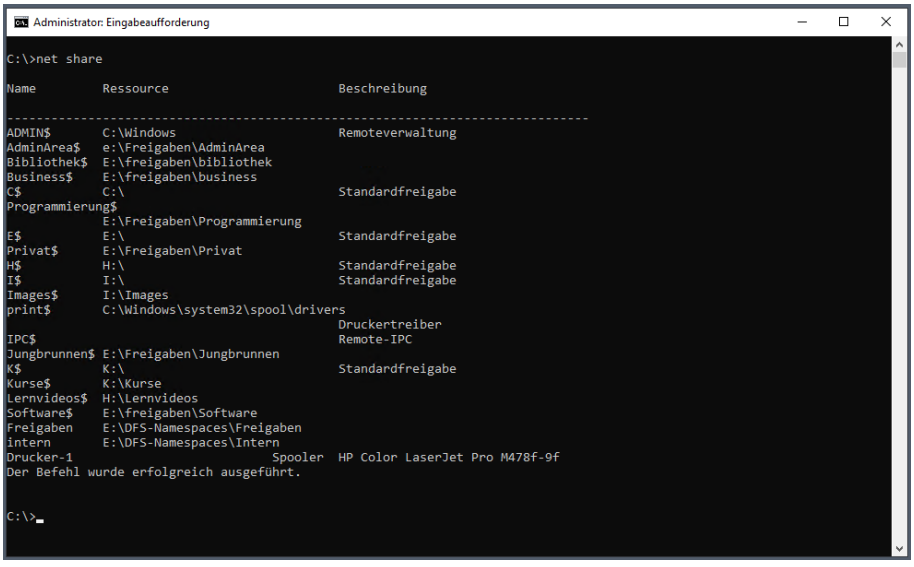


Entfernung der Printserver-Rolle auf WS-FS1

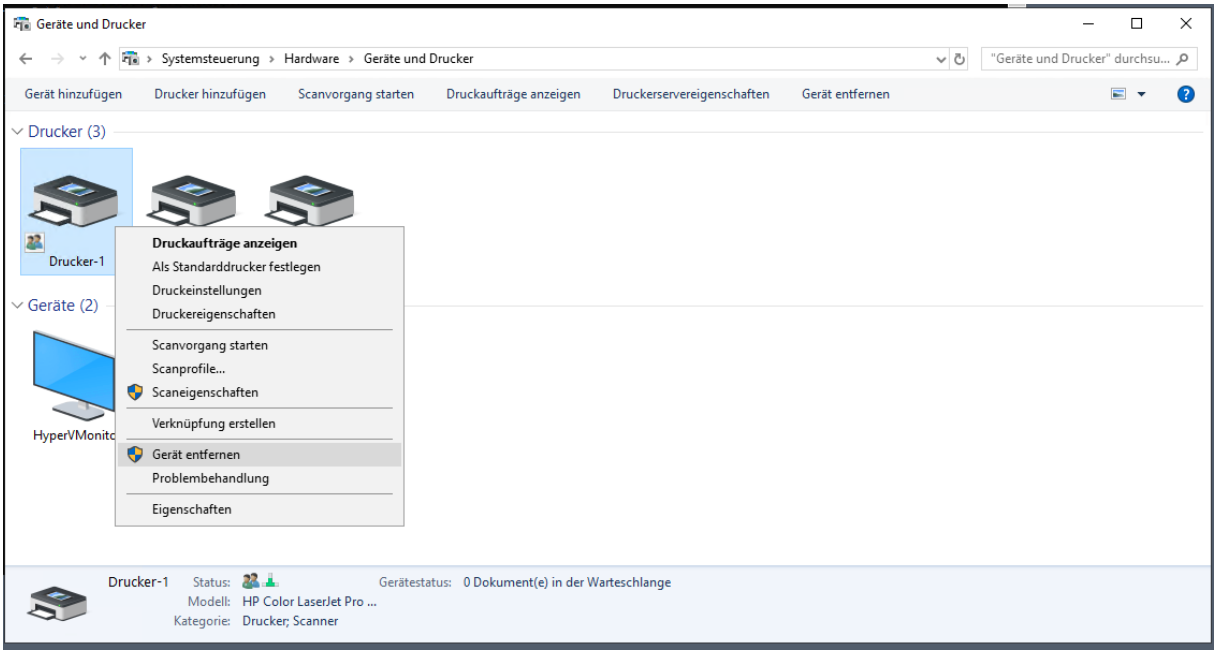
Die Druckerdienste werden jetzt auf dem Fileserver nicht länger benötigt. Ich deinstalliere die Rolle:



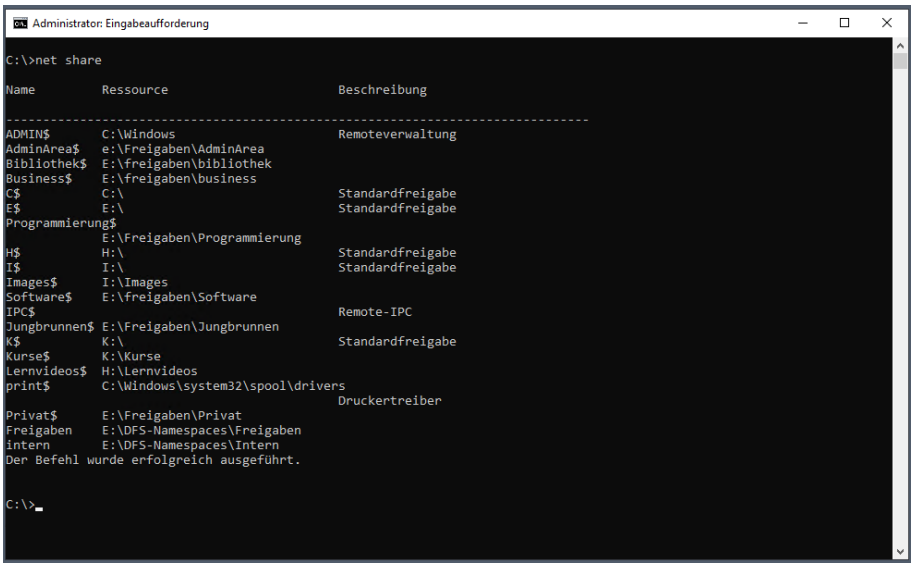
Leider habe ich die Freigabe des alten Druckers vergessen:



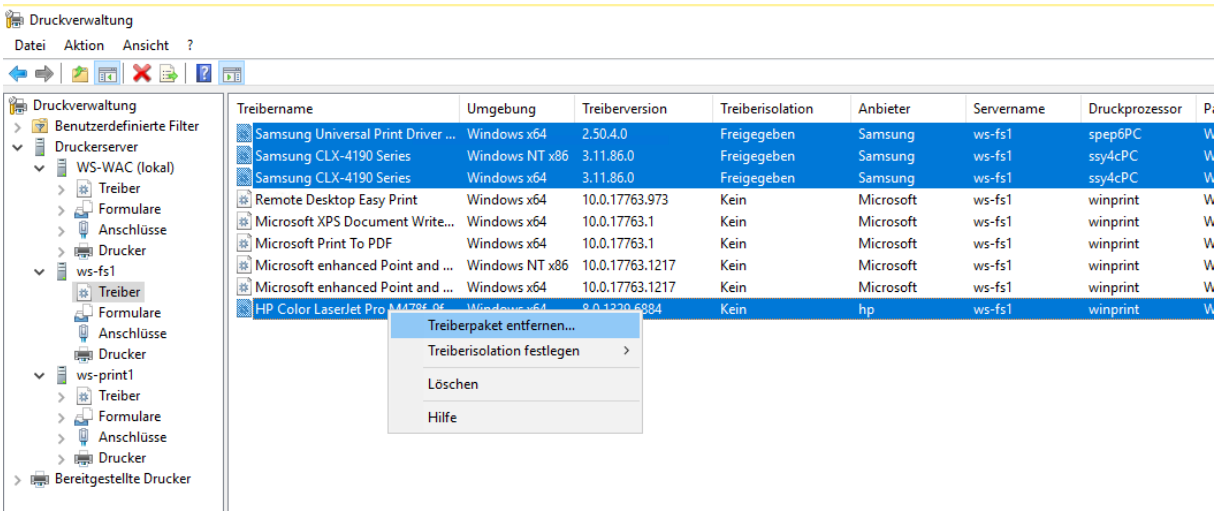
Aber die Systemsteuerung kann da auch weiterhelfen. Dort entferne ich den alten Drucker:

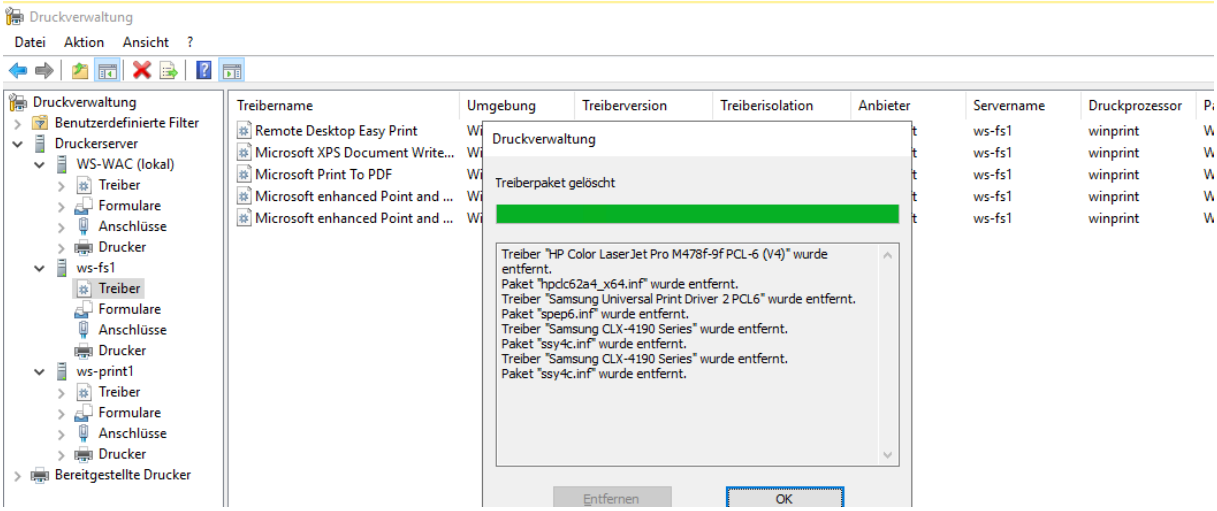


Damit ist auch die Freigabe Geschichte:



Alternativ kann man sich aber auch von einer anderen Maschine mit der Druckerverwaltung verbinden. So lkann ich auch den alten Treiber entfernen:

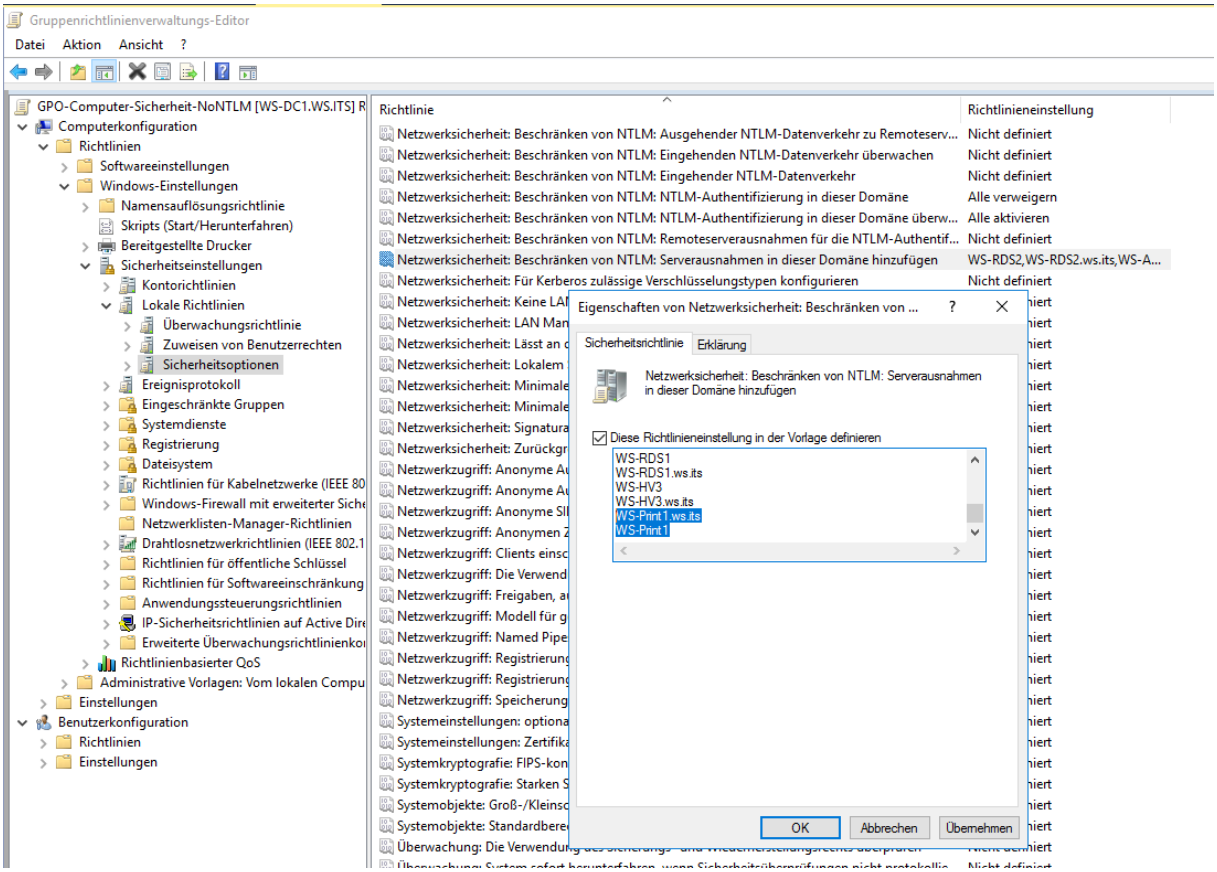




Damit kann ich jetzt wieder drucken und der alte Server ist aufgeräumt.

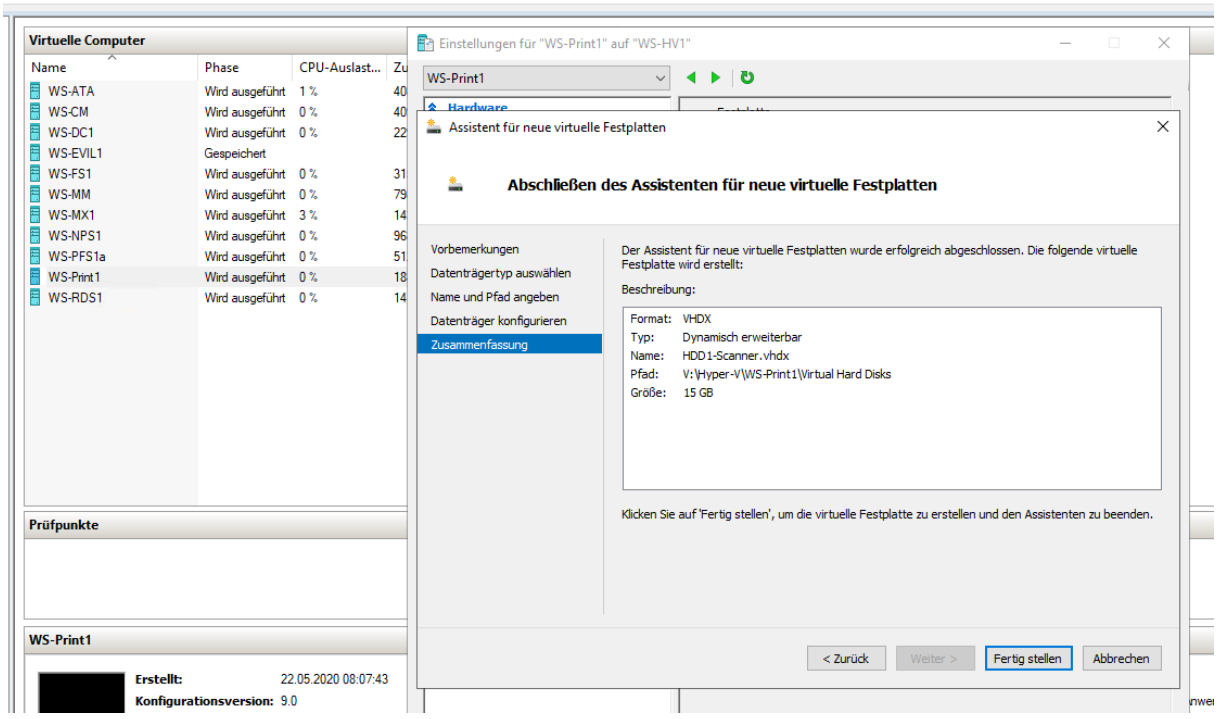
Security

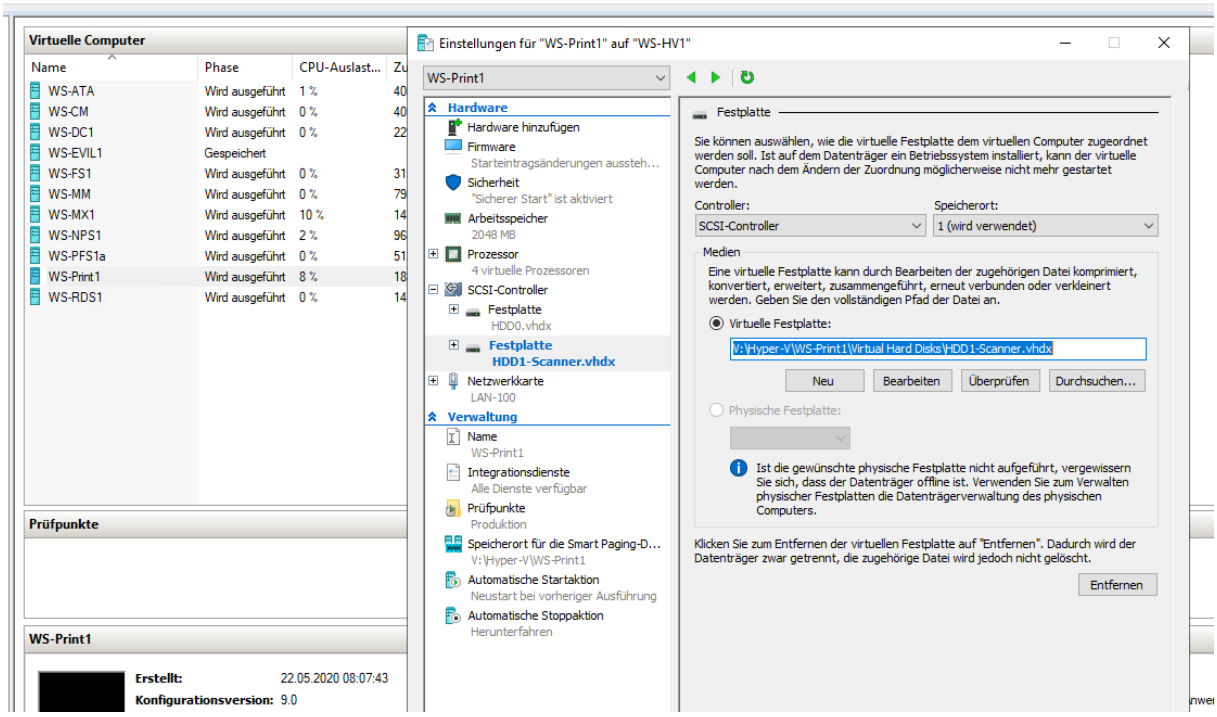
Damit der Scanner auf den neuen Printserver via SMB zugreifen kann, muss ich eine NTLM-Ausnahme vornehmen. NTLM hatte ich vor etlichen Monaten domänenweit deaktiviert. Nur wenige Server kommen damit nicht klar. Die Ausnahmen kommen in die gleiche Gruppenrichtlinie:



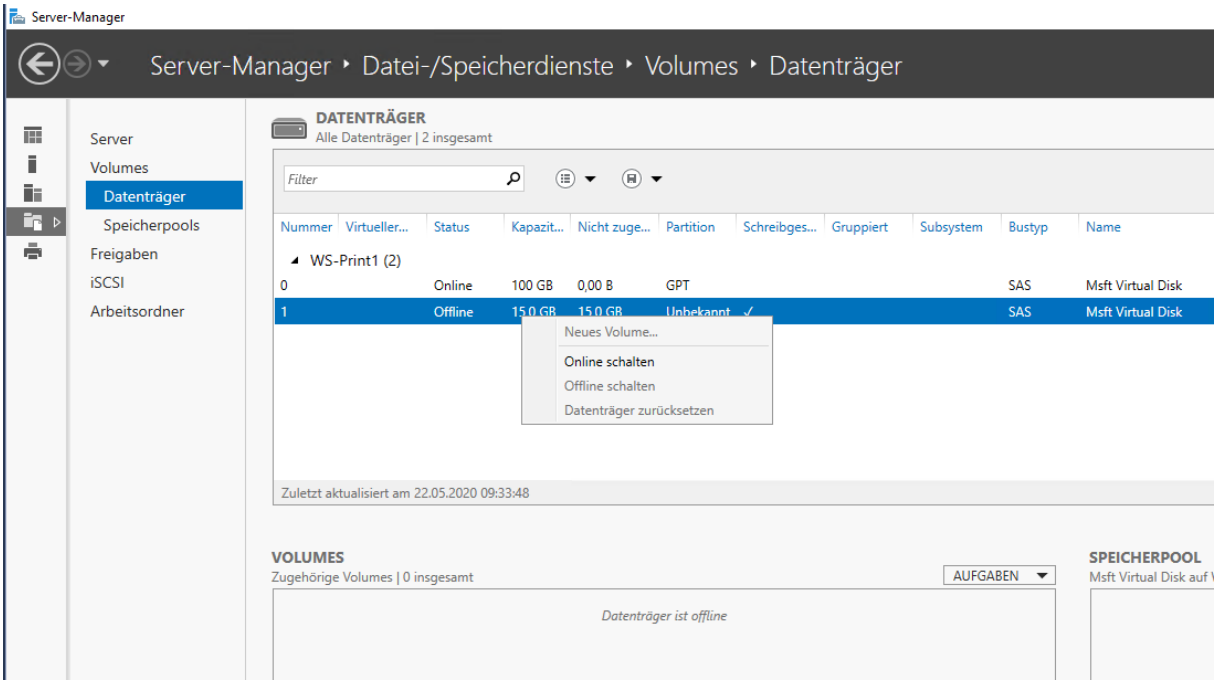
Aufbau der Scanfreigabe

Jetzt fehlt noch die Scan-Freigabe. Die Daten werden mit Sicherheit nicht sehr viel Platz einnehmen, aber ich möchte sie dennoch nicht auf der Systempartition unterbringen. Daher spendiere ich dem Server eine zusätzliche, virtuelle Festplatte im Hyper-V:

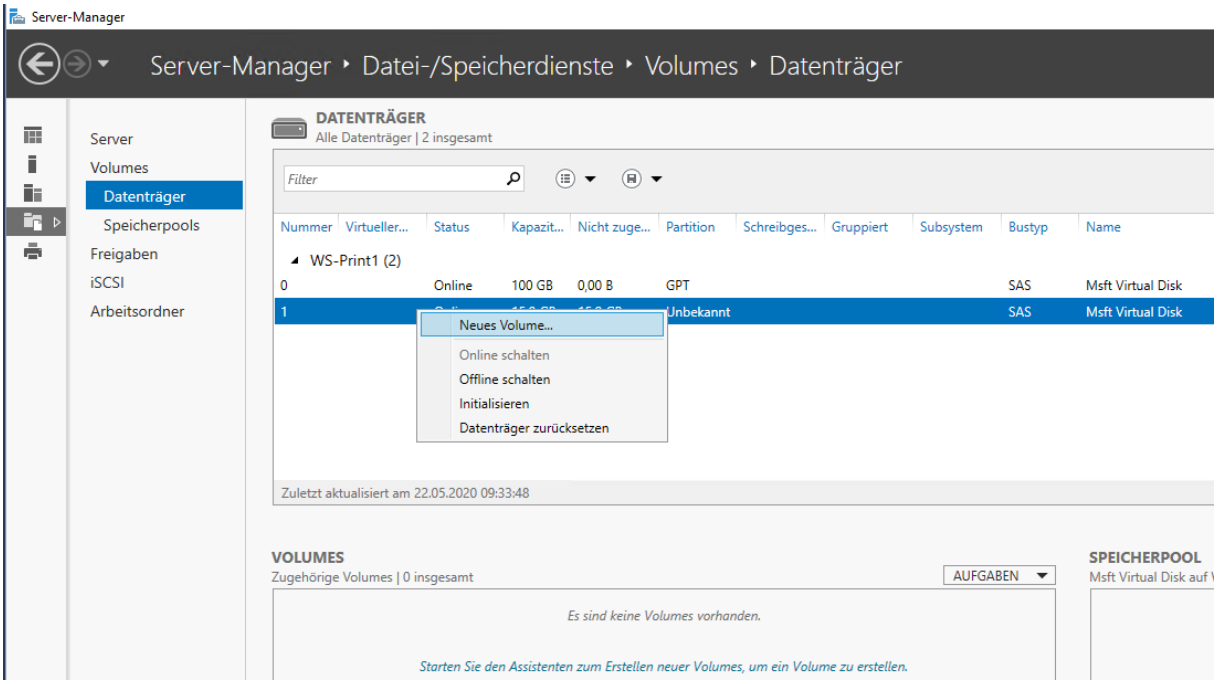


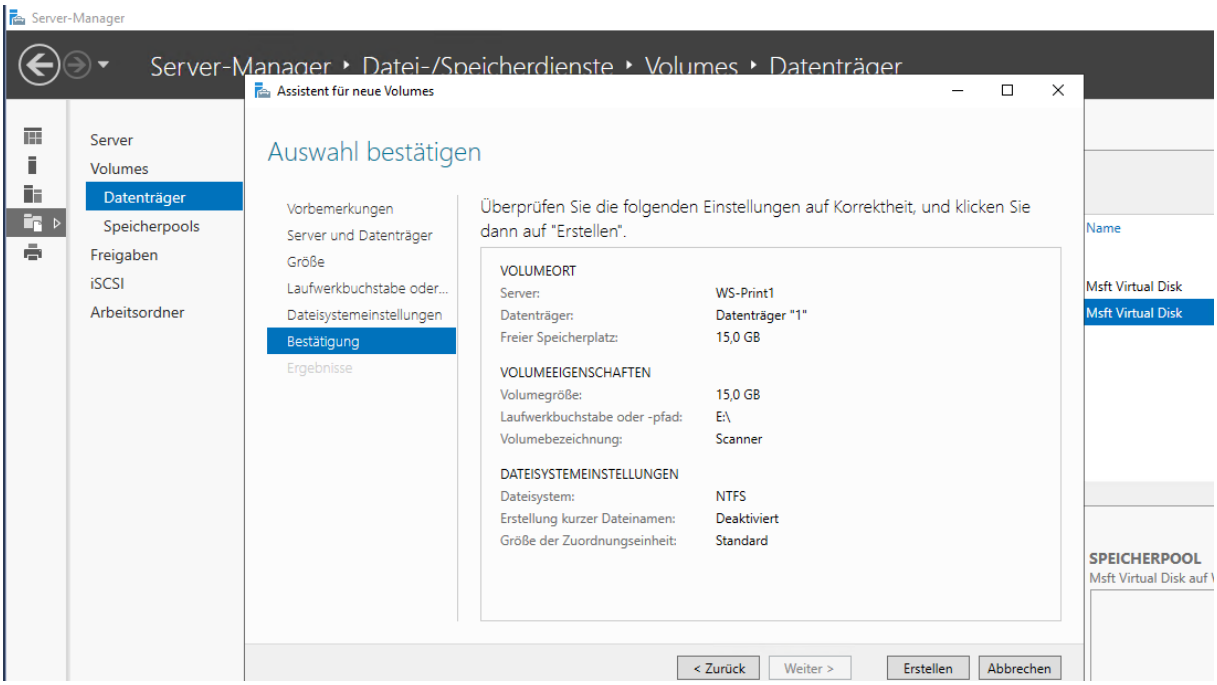


Im Servermanager des Printservers nehme ich danach die Platte online:

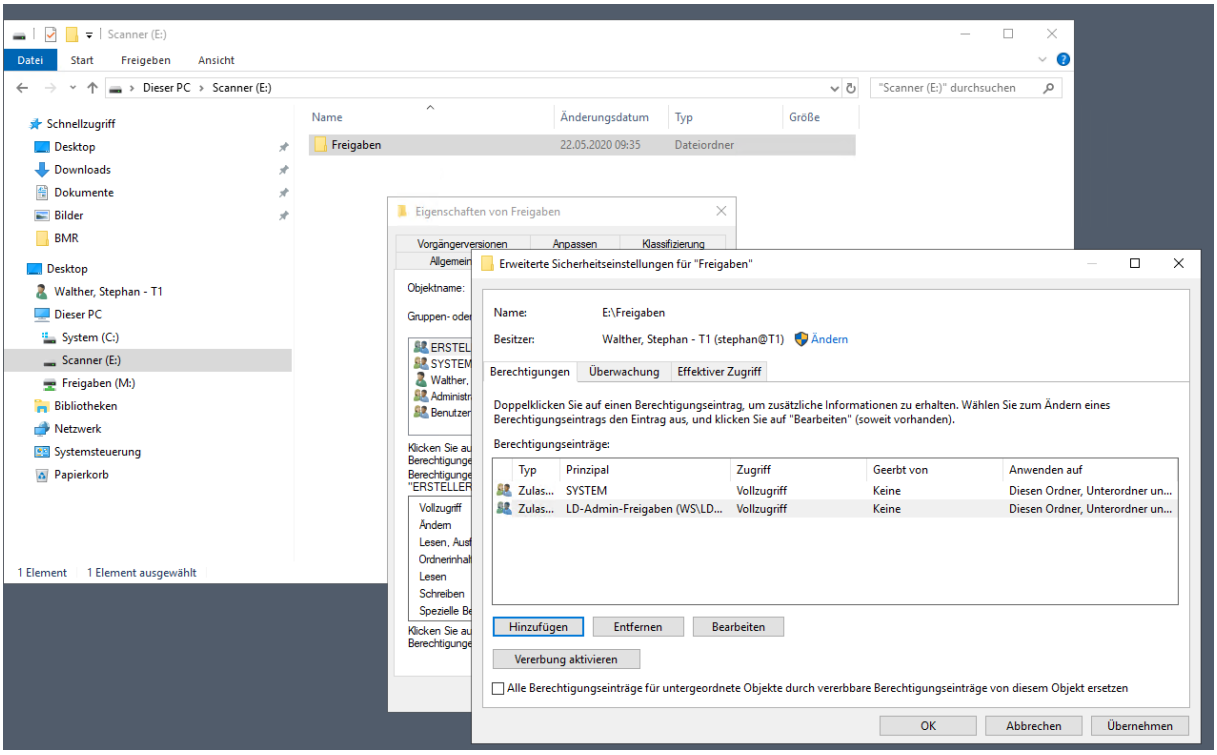


Dann erstelle ich ein neues Volume. Da gibt es nicht viel zu erklären:





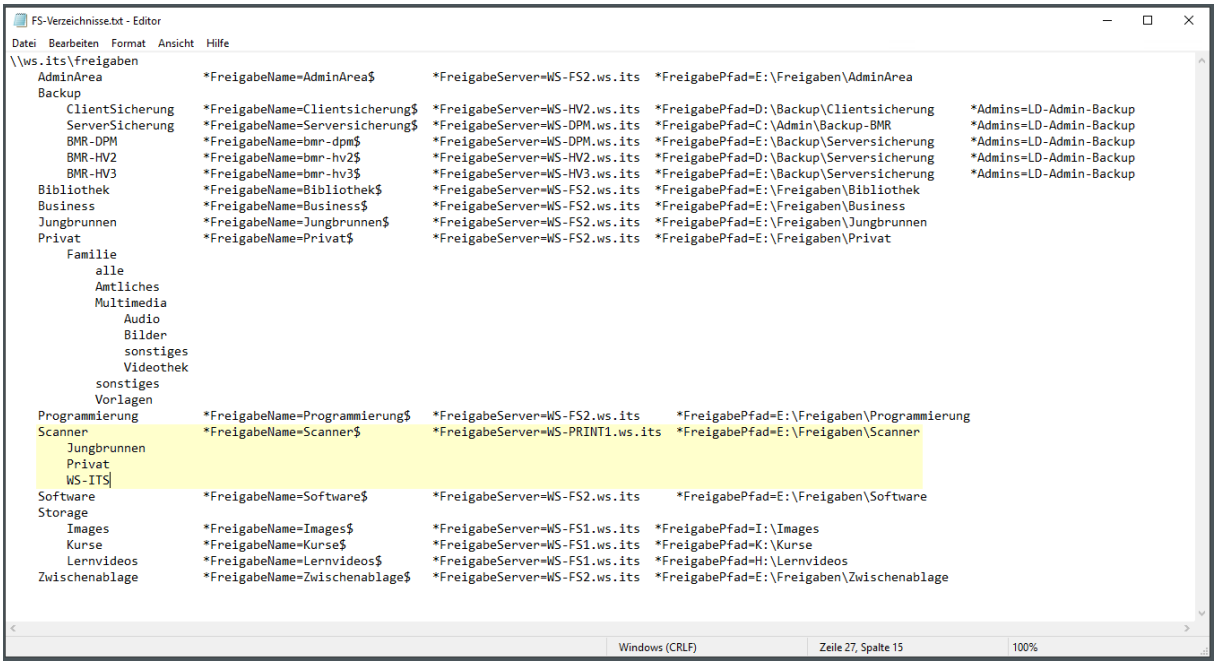
In dem neuen Volume erstelle ich eine Hauptordner und verändere dessen ACL. So kann nur das System oder meine Sicherheitsgruppe „LD-Admins-Freigaben“ – bzw. deren Mitglieder – darauf zugreifen:



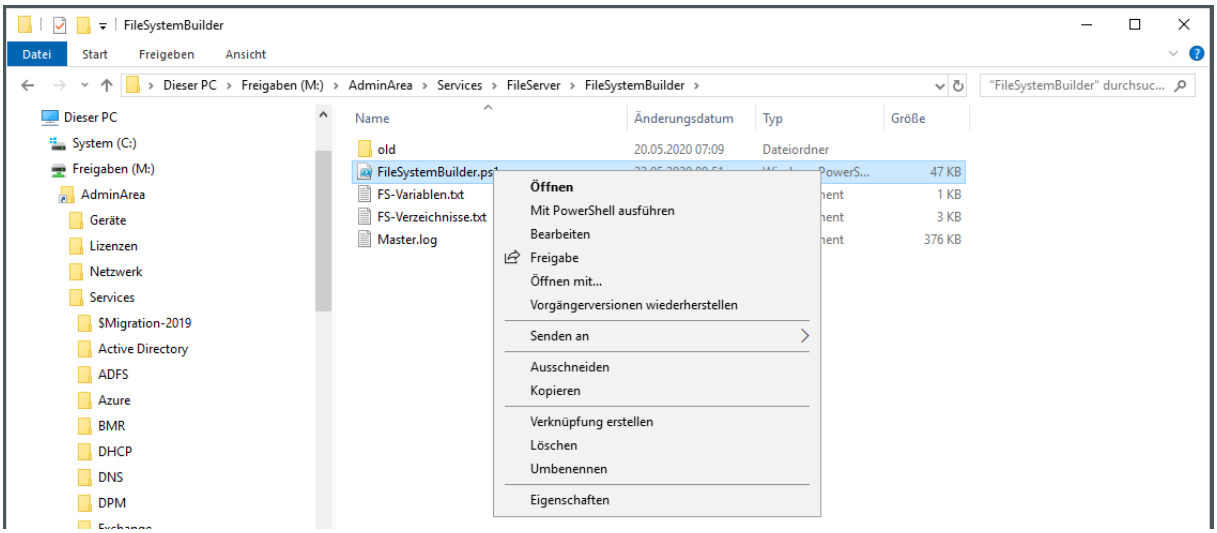
Als nächstes benötige ich folgende Dinge:

- Ich muss einen Ordner erstellen, den ich mit einer SMB-Freigabe versehe.
- Darunter möchte ich gerne 3 weitere Unterordner erstellen, die ich separat berechtigen kann
- Für jeden der 4 Ordner benötige ich 5 Sicherheitsgruppen im Active Directory inklusive einer definierten Verschachtelung mit deren Mitgliedschaften.
- Dann muss die neue Freigabe in meinem DFS-Namespace aufgenommen werden.

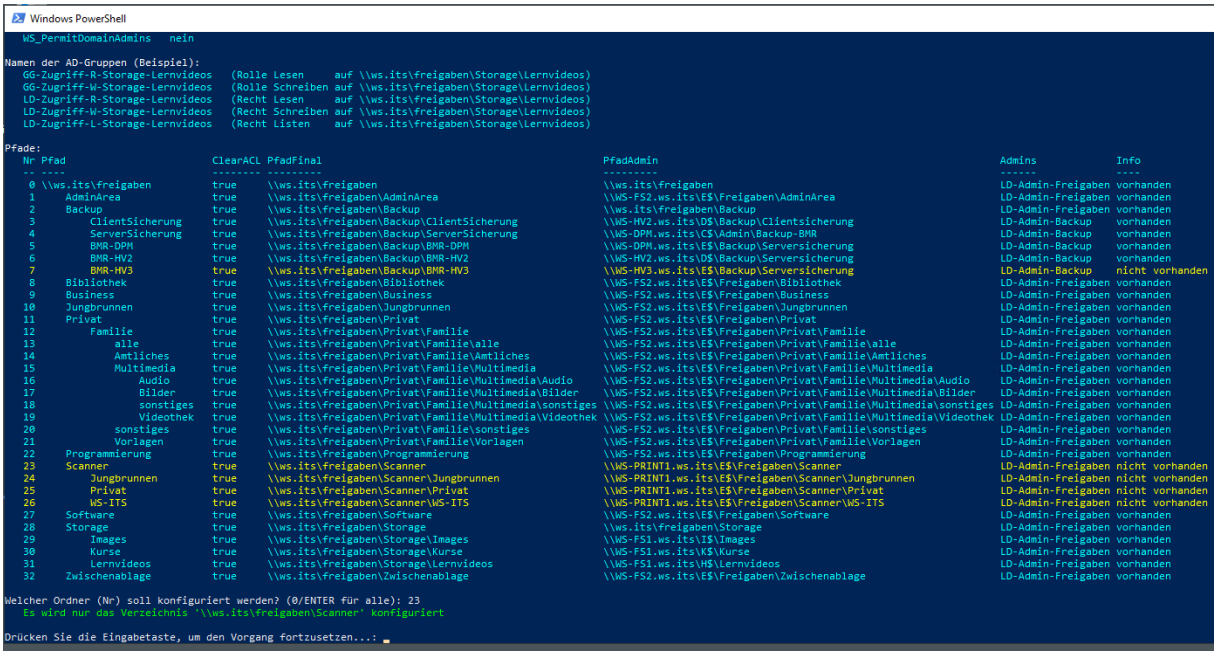
Das ist viel Arbeit. Kleine Fehler können ein aufwendiges Troubleshooting nach sich ziehen. Daher baue ich seit vielen Monaten mein Dateisystem für die Freigaben mit einem selbstprogrammierten PowerShell-Script auf (Na, wer ist jetzt überrascht?). Die Funktionen darin sind sehr komplet. Dafür ist die Konfiguration neuer Freigaben und Ordner extrem einfach. Ich deklariere alles in einer einzigen Textdatei. Hier kommt der gelb hinterlegte Block neu dazu. Dieser enthält alle relevanten Informationen:



Die Datei ist gespeichert. Nun starte ich mein PowerShell-Script:

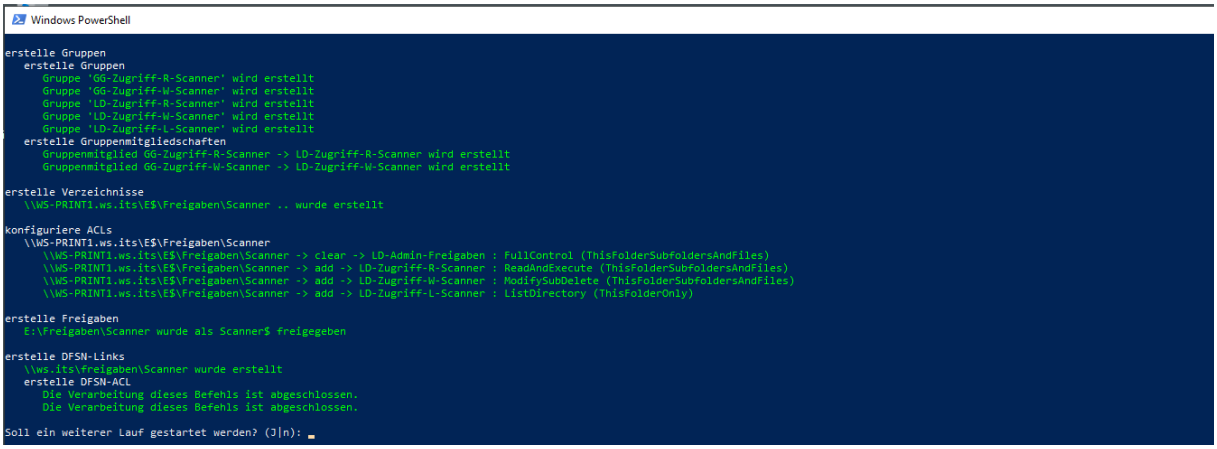


Das Script erkennt automatisch die neuen Verzeichnisse. Diese kann ich jetzt nummeriert anwählen:

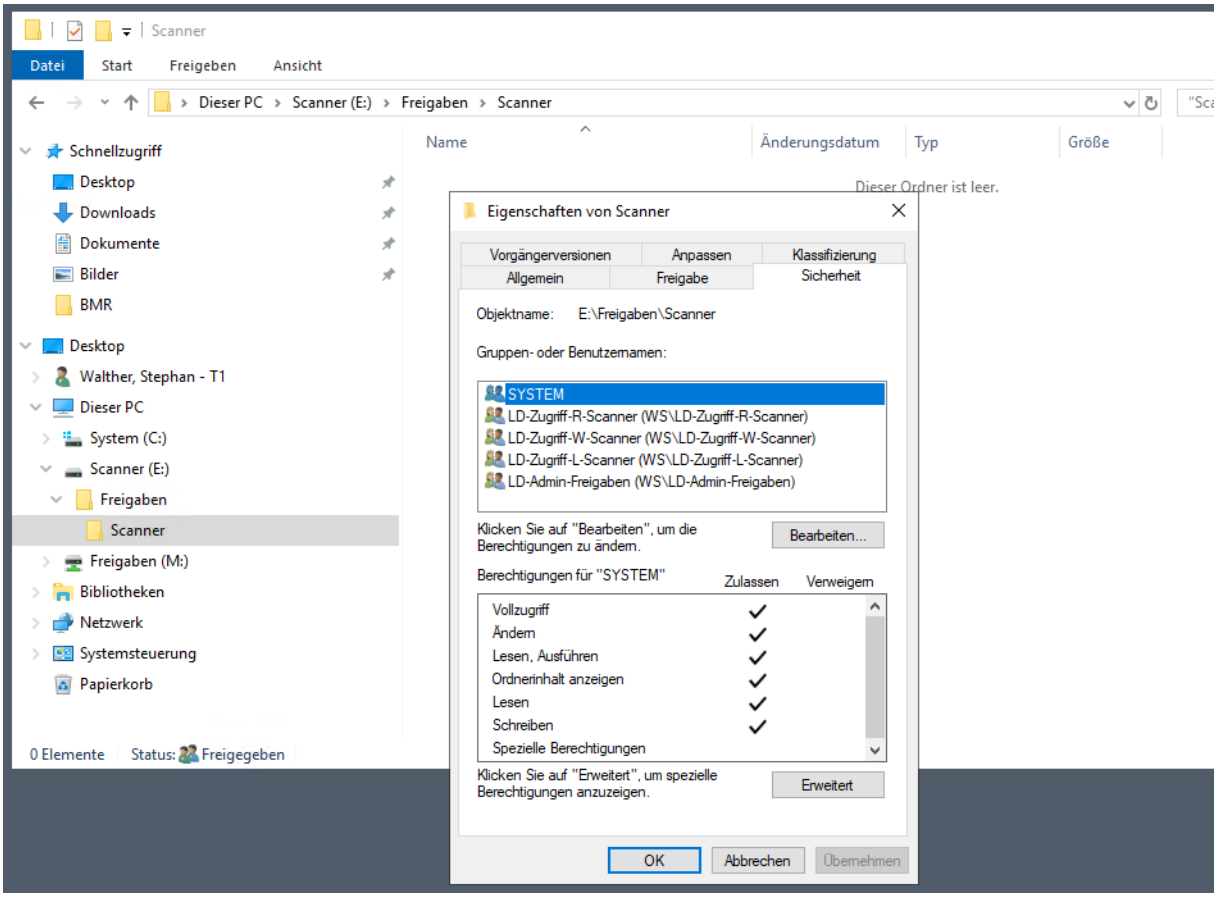


Dann werden die Aktionen abgefragt. Das Hauptverzeichnis darf die Prozedur komplett durchlaufen:

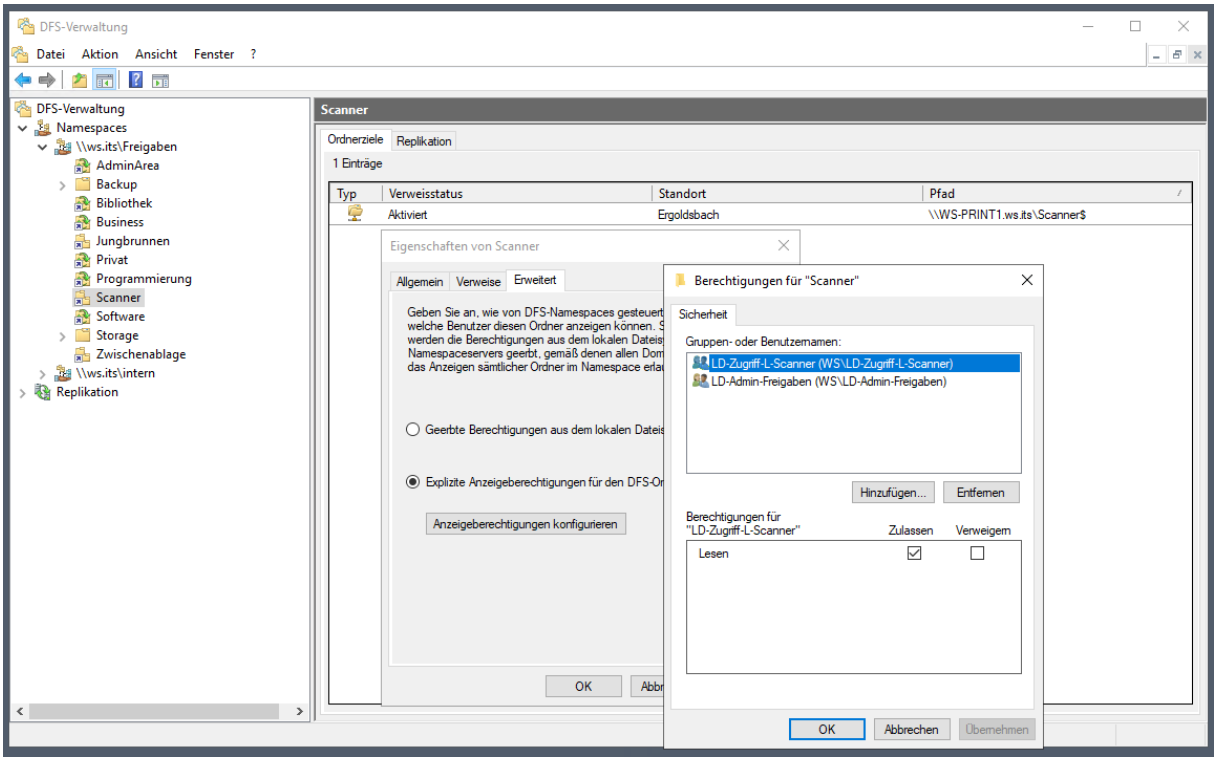
Und dann werden alle Aufgaben in der richtigen Reihenfolge erledigt:



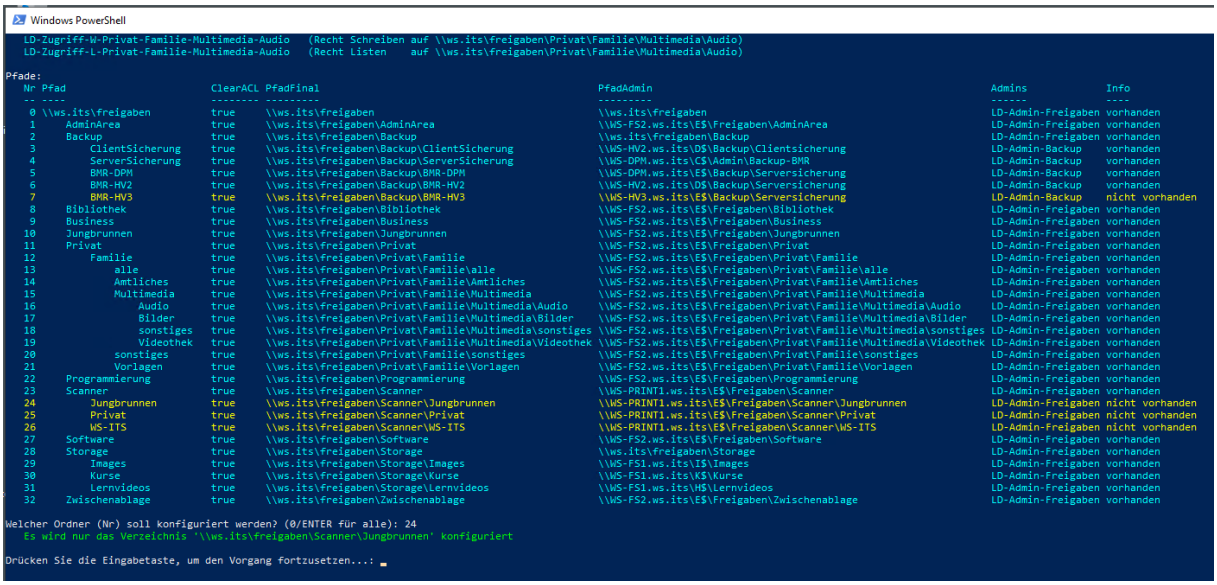
Eine kurze Kontrolle auf dem Printserver zeigt die korrekt hinterlegte ACL auf dem Hauptordner:



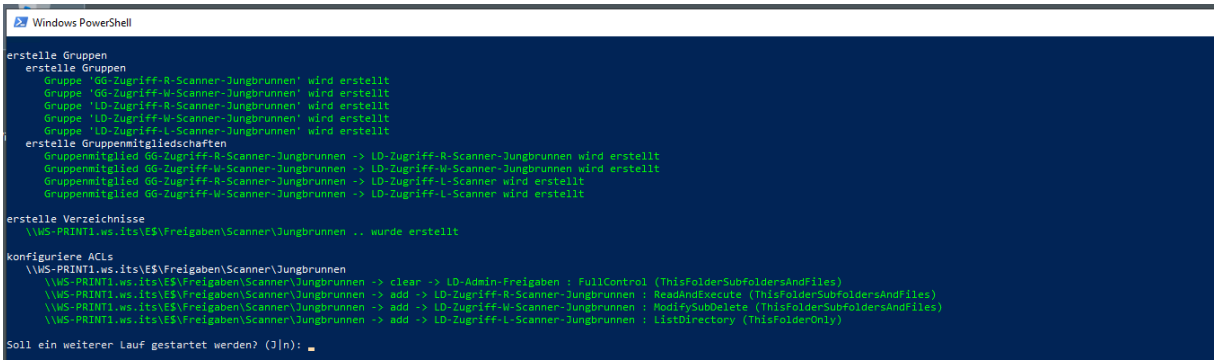
Ebenso kann mein Script die neue Freigabe im DFS-Namespace veröffentlichen. Natürlich mit korrekter ACL für eine Access Based Enumeration:



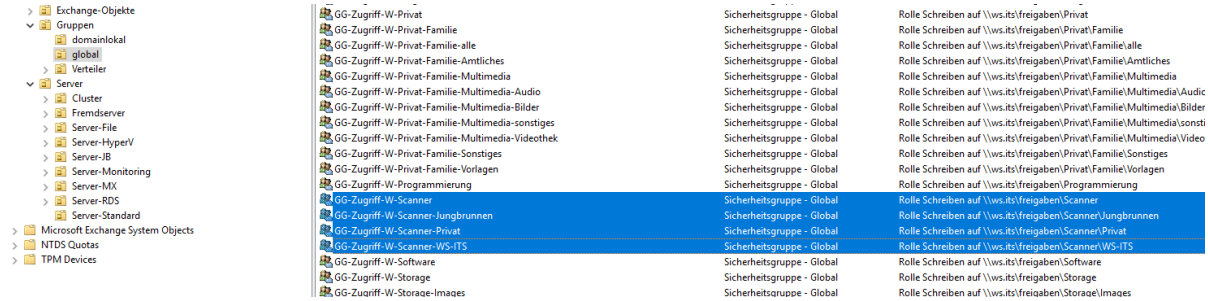
Unter der neuen Freigabe erstelle ich 3 weitere Unterordner. Das Script startet dazu eine Wiederholung. So kann ich bequem den nächsten Ordner auswählen:



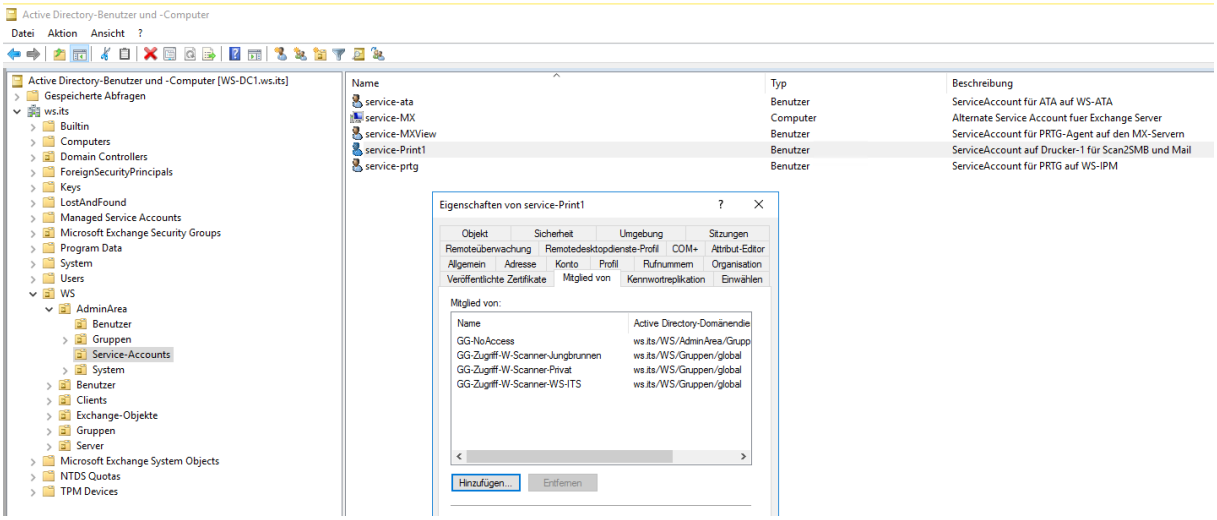
Diese Verzeichnisse benötigen nur die AD-Gruppen, die ACL und das Erstellen des Ordners:



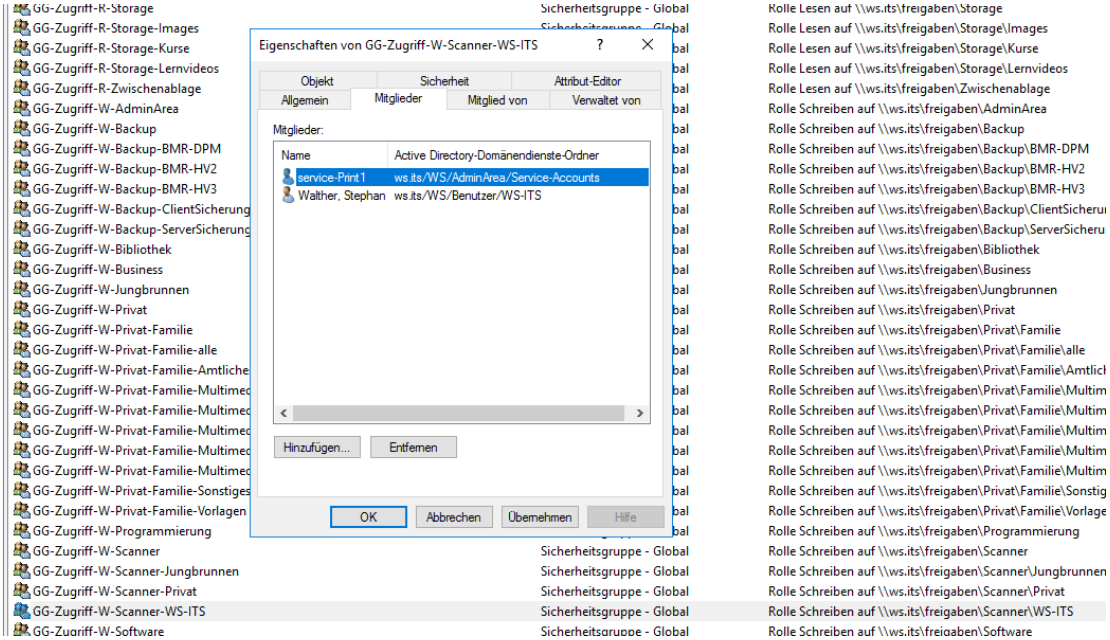
Nachdem ich alle 4 Verzeichnisse erstellt habe, lohnt sich auch ein Blick in das Active Directory. Hier sind 4 der 20 neuen Gruppen sichtbar:



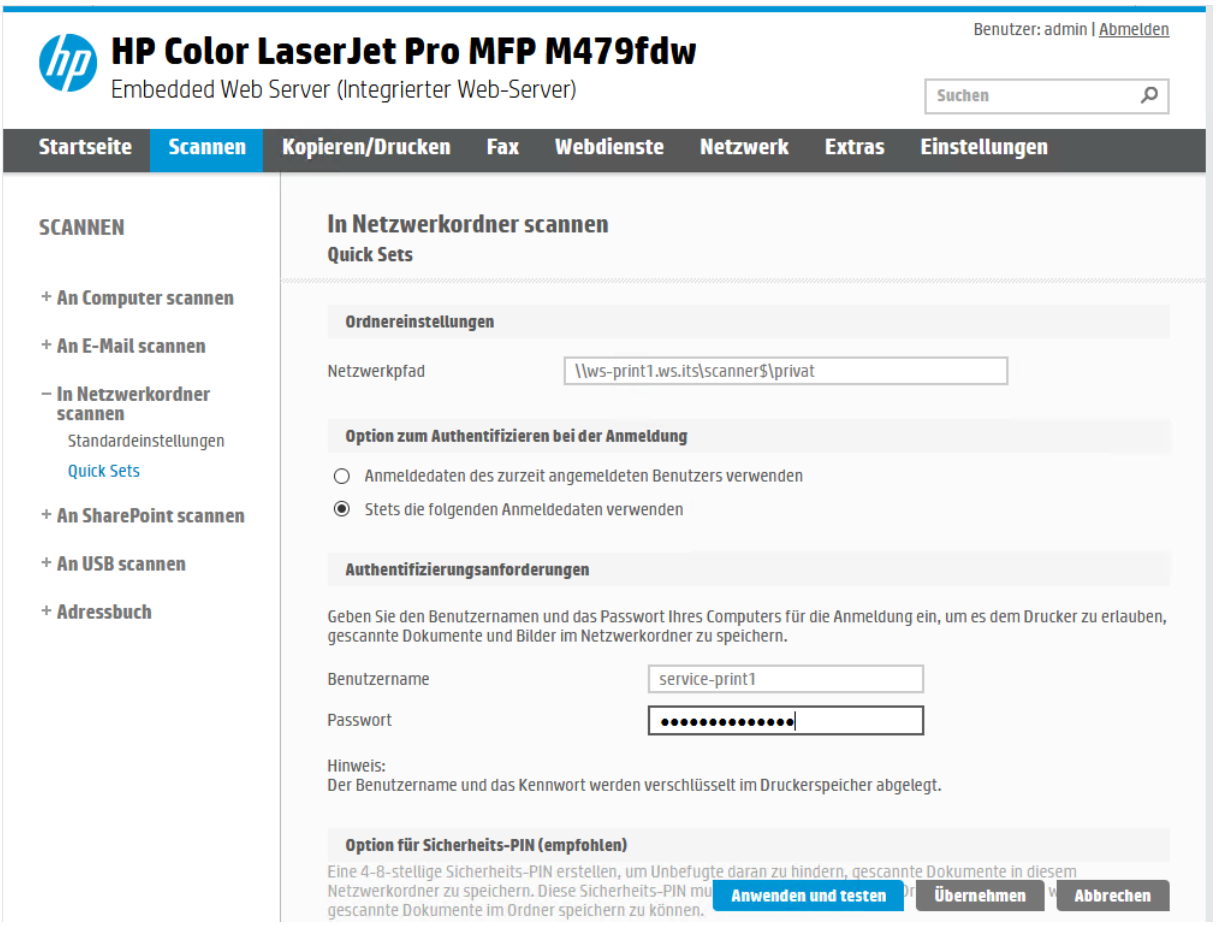
Der neue Scanner muss mit einem Account abgesichert auf das Verzeichnis zugreifen können. Diesen erstelle ich in meiner OU mit den ServiceAccounts. Dann nehme ich ihn in die 3 erforderlichen Schreibgruppen auf:



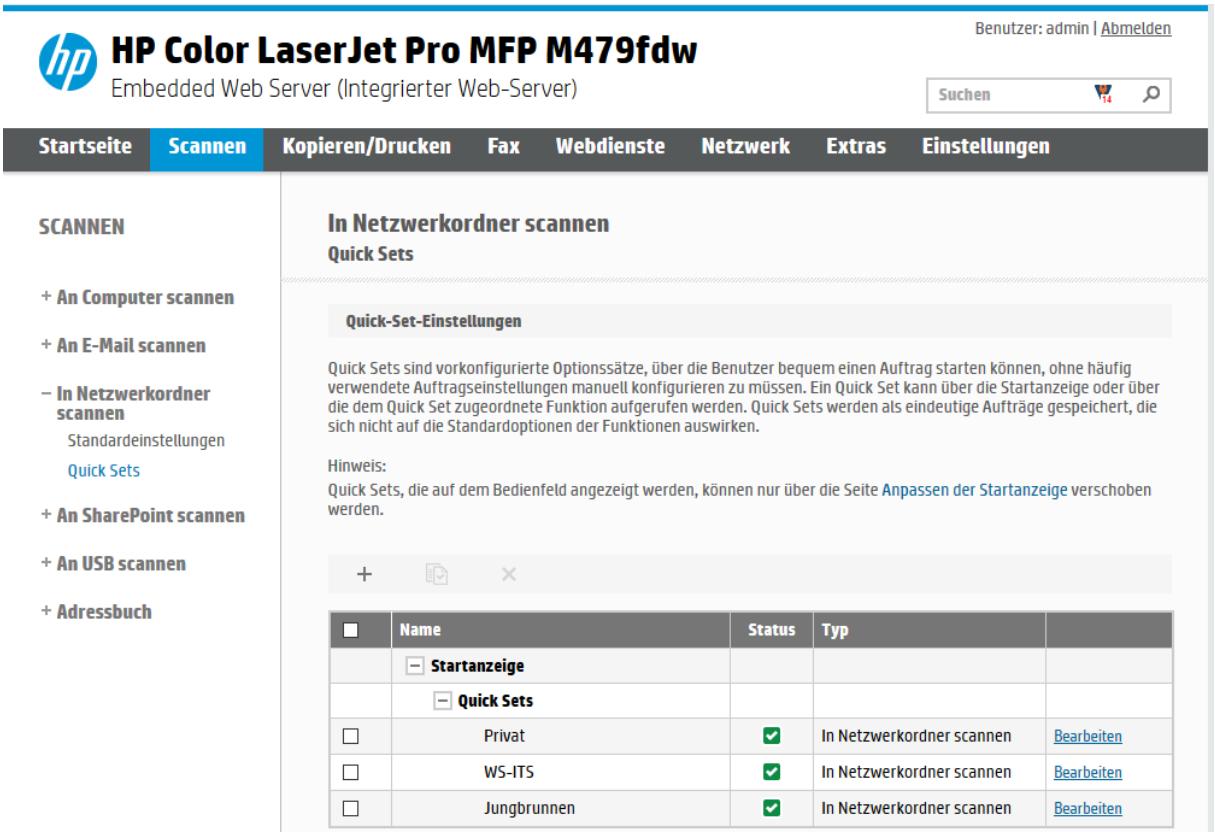
Mein eigener Account darf aber auch nicht in den Gruppen fehlen:



Jetzt kann ich den Account im Drucker hinterlegen:



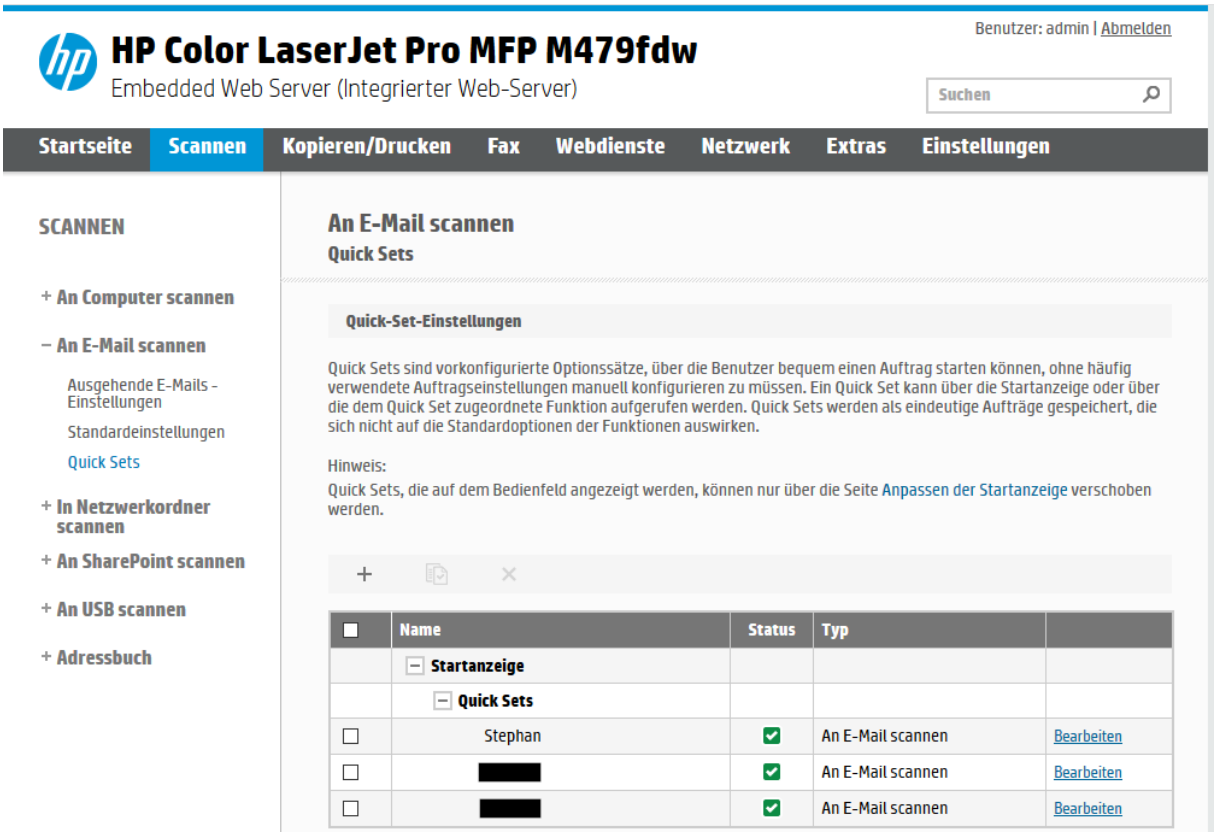
Insgesamt erstelle ich drei Scanner-Ziele. So kann ich die Dateien gleich mit den richtigen Berechtigungen ablegen:



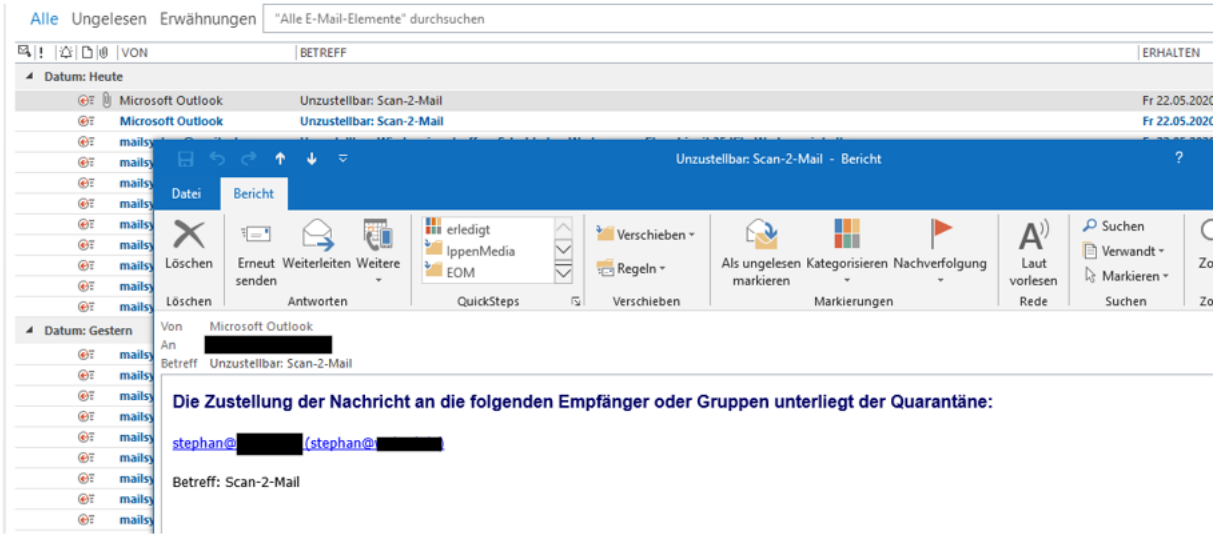
Ein Testlauf war sofort erfolgreich.

Scan to Mail

Auch diese Funktion soll nicht fehlen. Ich richte mehrere Quicksets ein:



Meinen Mailserver habe ich ebenfalls als Ziel angegeben. Ein Testscann kam aber nicht im Posteingang an. Dafür fand ich diese Nachricht in meinem Spam-Verzeichnis:



Mein Exchange-Server verwendet verschiedene Filtermechanismen. Hier hat er den Absender als gefälscht anerkannt. Daher nehme ich die Absenderadresse in meine Ausnahmen auf:

```
116 #region Konfiguration BypassedSenders
117     $Liste = (Get-ContentFilterConfig).BypassedSenders | sort-object
118     $Liste += [redacted]
119
120     $Liste = $Liste | Sort-Object -Unique
121     $Liste
122
123     Set-ContentFilterConfig -BypassedSenders $Liste
124
125 #endregion
```

Jetzt kommen auch die Scans per Mail an.