Inhalt

Einleitung

Zielsetzung

Im meinem Außenstandort in Neufahrn habe ich einen Hyper-V-Host mit dem Namen WS-RDS3 aufgestellt. Dieser betreibt die virtuellen Maschinen mit den Services, die ich dort benötige. Der Server läuft aktuell mit Windows Server 2016.

Der Server soll auf Windows Server 2019 umgestellt werden. Dazu sind Anpassungen an den Services notwendig.

Die Umstellung findet in der Zeit des Betriebsurlaubes zwischen den Jahren (zwischen Weihnachten und Silvester) statt. Es kann also mit einer Downtime gearbeitet werden.

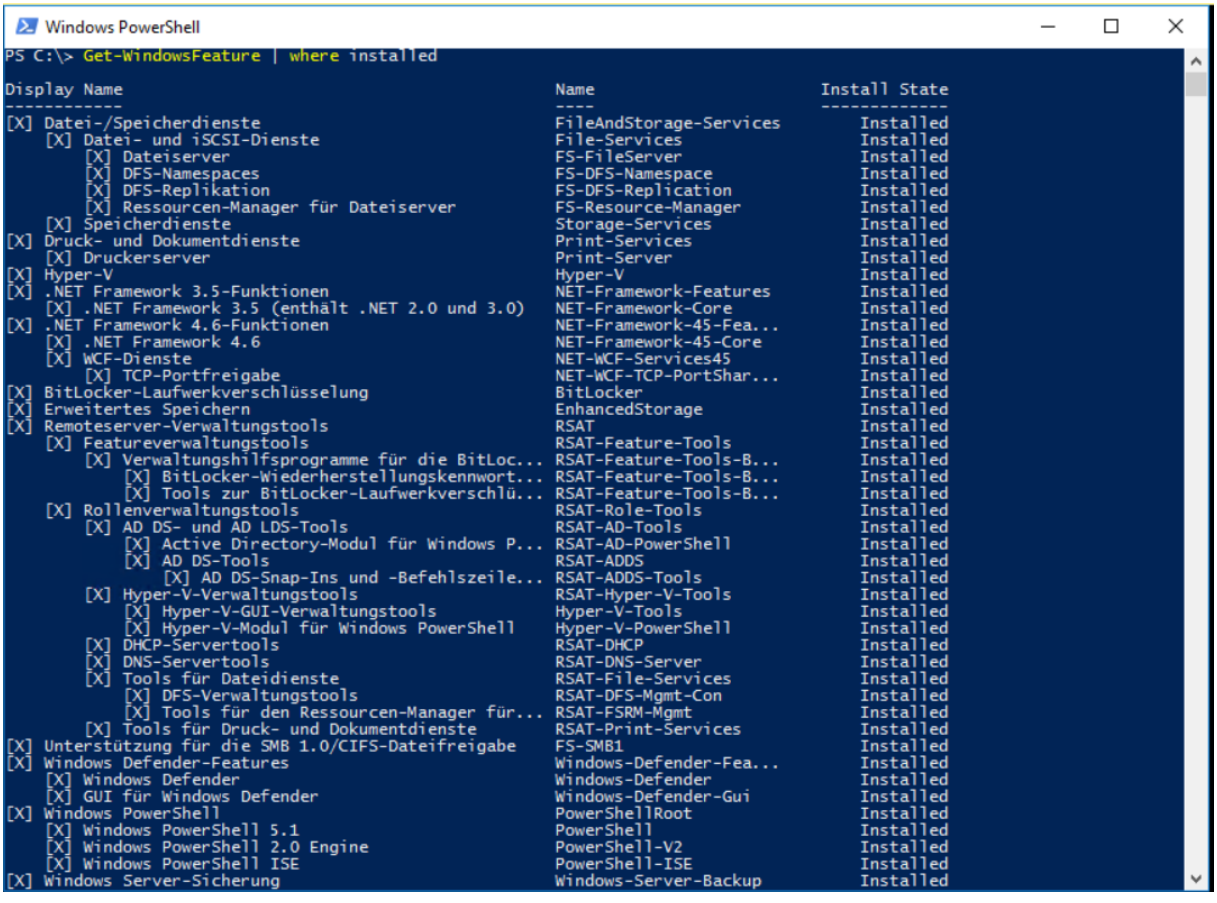
Die Hardware soll wiederverwendet werden.

Analyse des alten Servers

Der Server hat eine durchaus bewegte Vergangenheit hinter sich. Ich wollte ursprünglich nur einen Server in dem Standort aufstellen. Meine Kolleginnen sollten darauf alle Dienste und Anwendungen vorfinden, die zum Arbeiten erforderlich sind.

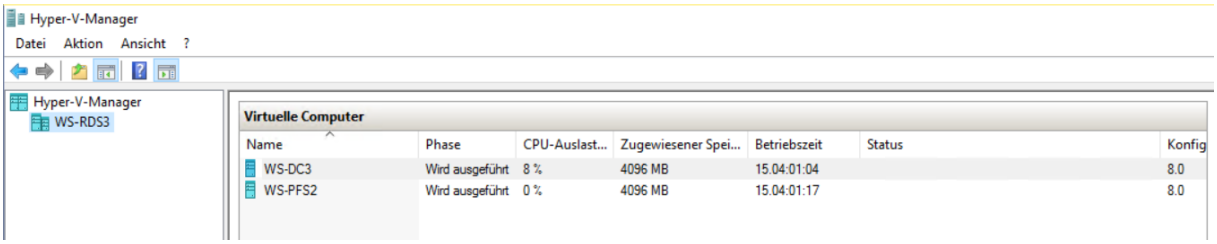
Da die Hardware sehr begrenzt ist (Quadcore, 16GB RAM, 120GB SSD, 1x Gbit) musste ich Dienste auf den Servern zusammenfassen. Auch eine RDP-Anmeldung sollte möglich sein (daher der Name WS-RDS3). Im Nachhinein war das keine so gute Idee. Aber mit dieser Migration kann ich jetzt einige Korrekturen vornehmen.

Zuerst verschaffe ich mir einen Überblick über die installierten Rollen und Features:

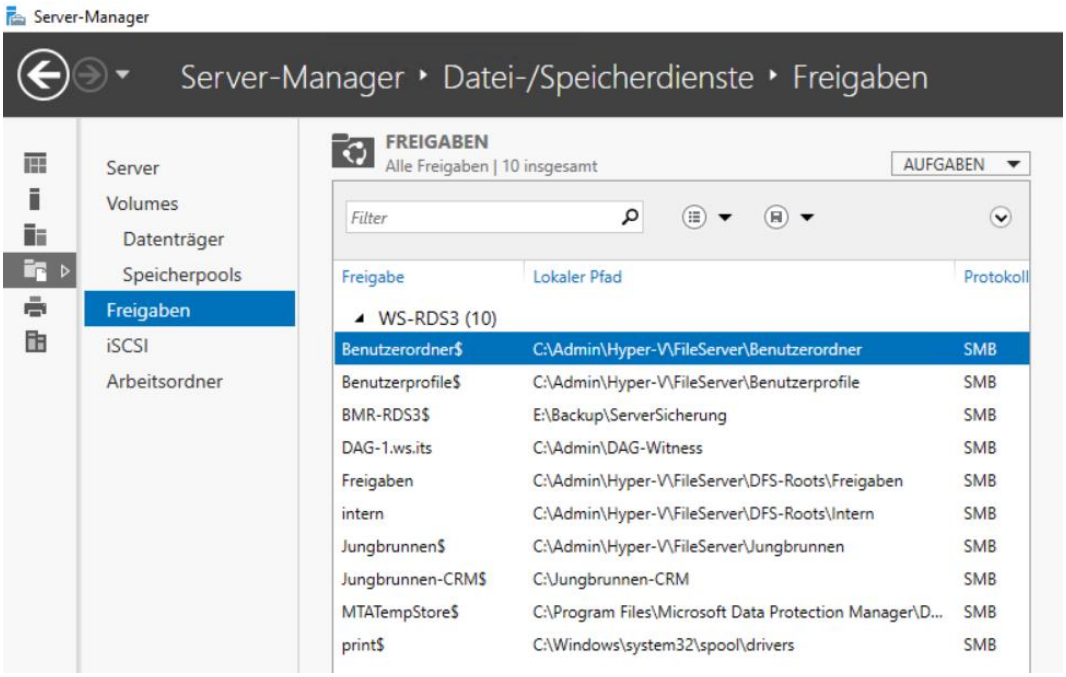


Der Server ist also ein Hyper-V-Host, ein Fileserver mit DFS-Namespace und DFS-Replica, und der lokale Druckserver.

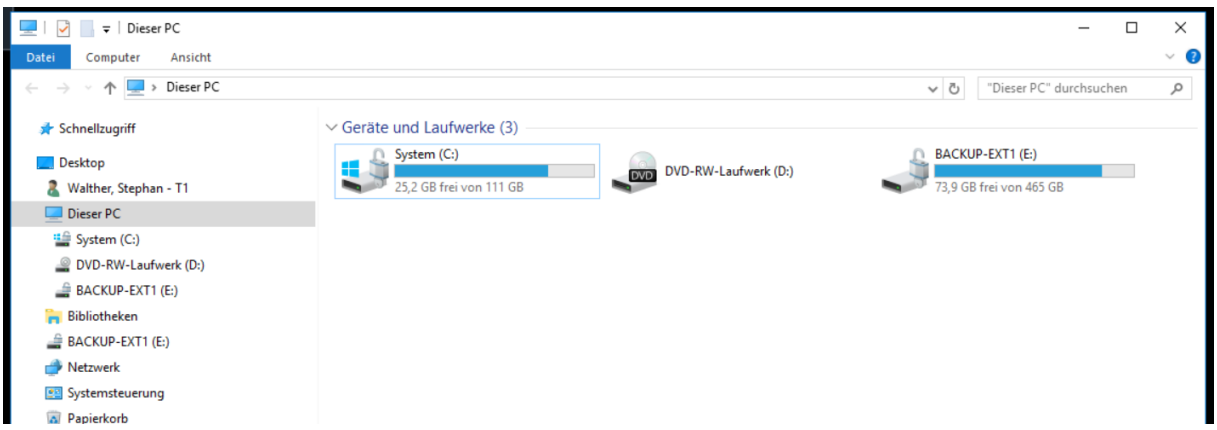
Über Hyper-V werden diese beiden VMs bereitgestellt: ein Domain Controller mit DHCP und DNS und eine virtuelle PFSense (Firewall):



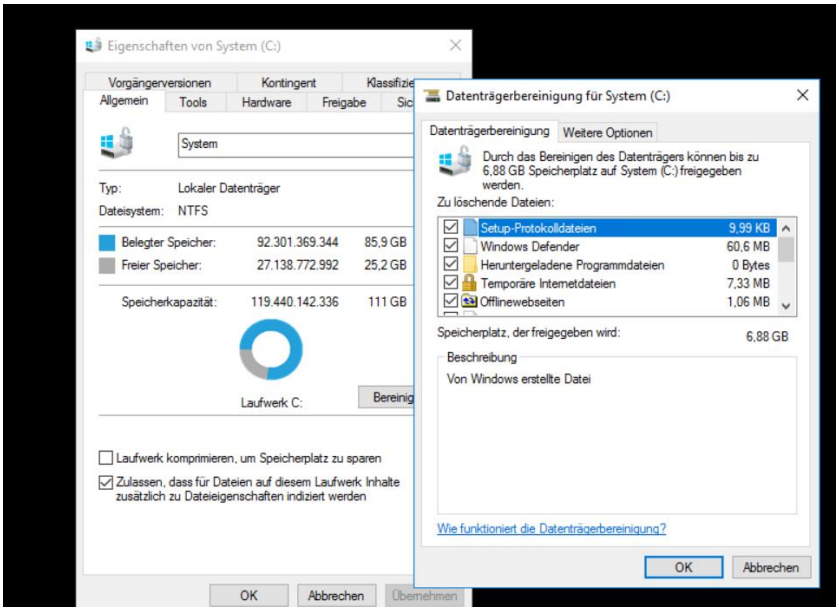
Alle Freigaben musste ich direkt auf die Systempartition ablegen – genauso auch die Dateien der virtuellen Maschinen. Für eine zusätzliche Partitionierung war einfach kein Platz mehr:



Nur eine weitere Festplatte ist noch über USB angeschlossen. Auf dieser werden Datensicherungen gespeichert. Auf dem Systemdatenträger ist kaum noch freier Platz vorhanden:

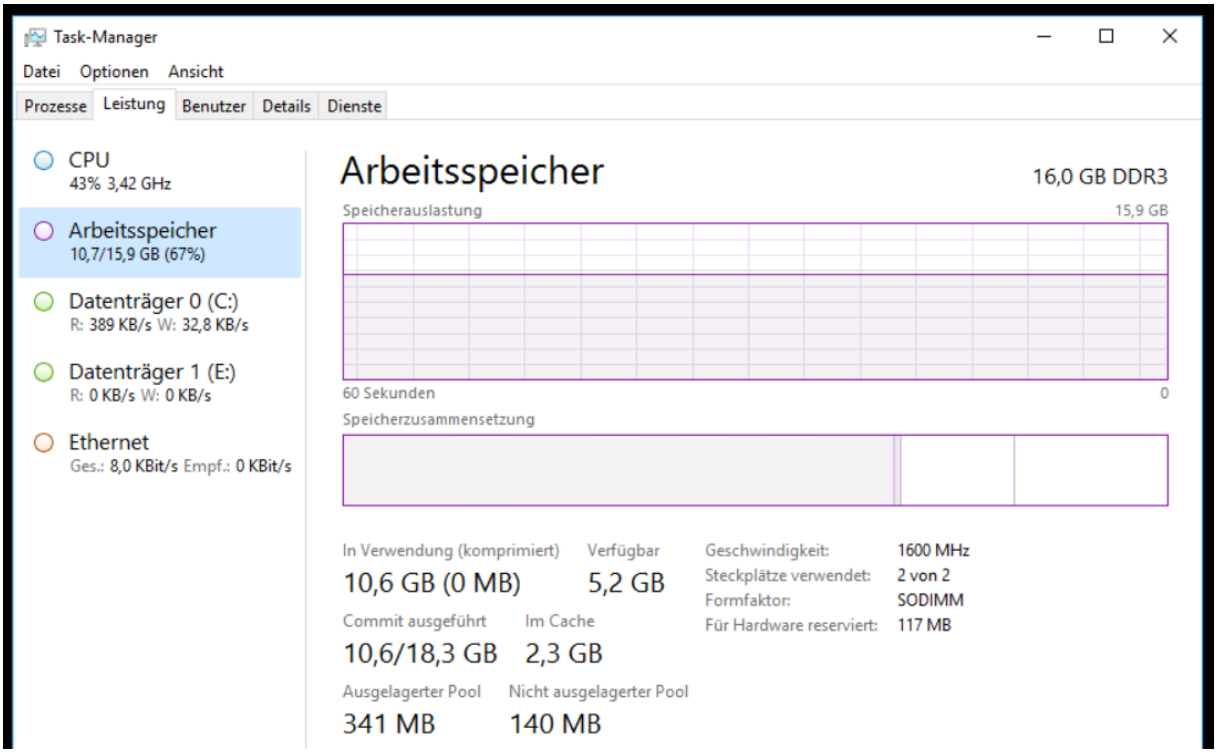


Vielleicht wird Speicherplatz durch nicht mehr benötigte Dateien belegt? Ich starte die Datenträgerbereinigung und durchsuche dabei auch Systemverzeichnisse. Naja, besser als Nichts:



Warum ich nicht einfach eine weitere Festplatte einbaue? Ganz einfach: der Server ist eigentlich ein Mini-PC. Der kann nur eine 2,5“-Festplatte aufnehmen. Und mit externen Datenträgern möchte ich nicht in Kombination mit virtuellen Maschinen arbeiten.

Wie sieht es denn mit den anderen Ressourcen aus? Der Arbeitsspeicher ist noch etwas belastbarer. Aber eine Aufrüstung ist nicht möglich. Alle Slots sind belegt:



Ich durchsuche die installierten Anwendungen. Lokal ist ein Office 2016 vorhanden. Dieses war für den RDP-Zugriff gedacht, wird aber nicht (mehr) benutzt. Die Anmeldung ist mit DUO-Zweifaktor-Authentifizierung abgesichert. Und für eine spezielle Dateisicherung ist ein DPM-Agent installiert

Als weitere Besonderheit ist das Feature Bitlocker konfiguriert. Damit wird die gesamte SSD verschlüsselt.

Und zusätzlich habe ich meinen Exchange-Servern ein DAG-Witness-Share auf dem Server bereitgestellt.

In der Rolle Printserver ist nur ein Drucker freigegeben. Diese Freigabe hatte aber immer wieder Probleme und daher wird der Drucker von den Clients direkt angesprochen. Die Rolle wird nicht mehr verwendet.

Planung der Migration

Wenn ich die erforderlichen Services um die nicht mehr benötigten bereinige, dann verbleiben die Rollen Hyper-V und der Fileservice. Diese beiden haben keinen Bezug zueinander und sollten daher auch nicht in einem Betriebssystem konsolidiert sein. Daher werde ich den Server **WS-RDS3** durch die Server **WS-FS3** und **WS-HV3** ersetzen. WS-FS3 wird dabei als neue VM unter WS-HV3 laufen und zusammen mit den Freigaben auch den DFS-Namespace und die DFS-Replikation bereitstellen.

Die zusätzliche VM kann mit dem verbleibenden Arbeitsspeicher gut auskommen. Für die CPU und die Netzwerkkarte sehe ich keine Engpässe.

Aber die derzeitige SSD wird mit 120GB nicht mehr ausreichen. Daher werde ich die SSD durch eine neue ersetzen. Eine SSD mit 500GB sollte hier bis zum Ende der Hardwarelaufzeit genügen. Dies spielt mir auch beim Migrieren der VMs positiv zu, denn so muss ich die VMs im Vorfeld nicht erst verschieben. Ich werde nach der Installation des WS-HV3 auf der neuen SSD einfach die alte SSD über USB anschließen und die VMs kopieren.

Die Migration der Server wird in 2 Schritten erfolgen:

- Zuerst separiere ich den Fileservice auf eine neue VM. Damit kann ich auch die Lieferzeit der neuen SSD überbrücken, auch wenn es sehr eng auf der alten SSD werden wird.
- Im zweiten Schritt wird der Server dann als WS-HV3 neu installiert.

Schritt 2 - Neuinstallation des Hyper-V-Services

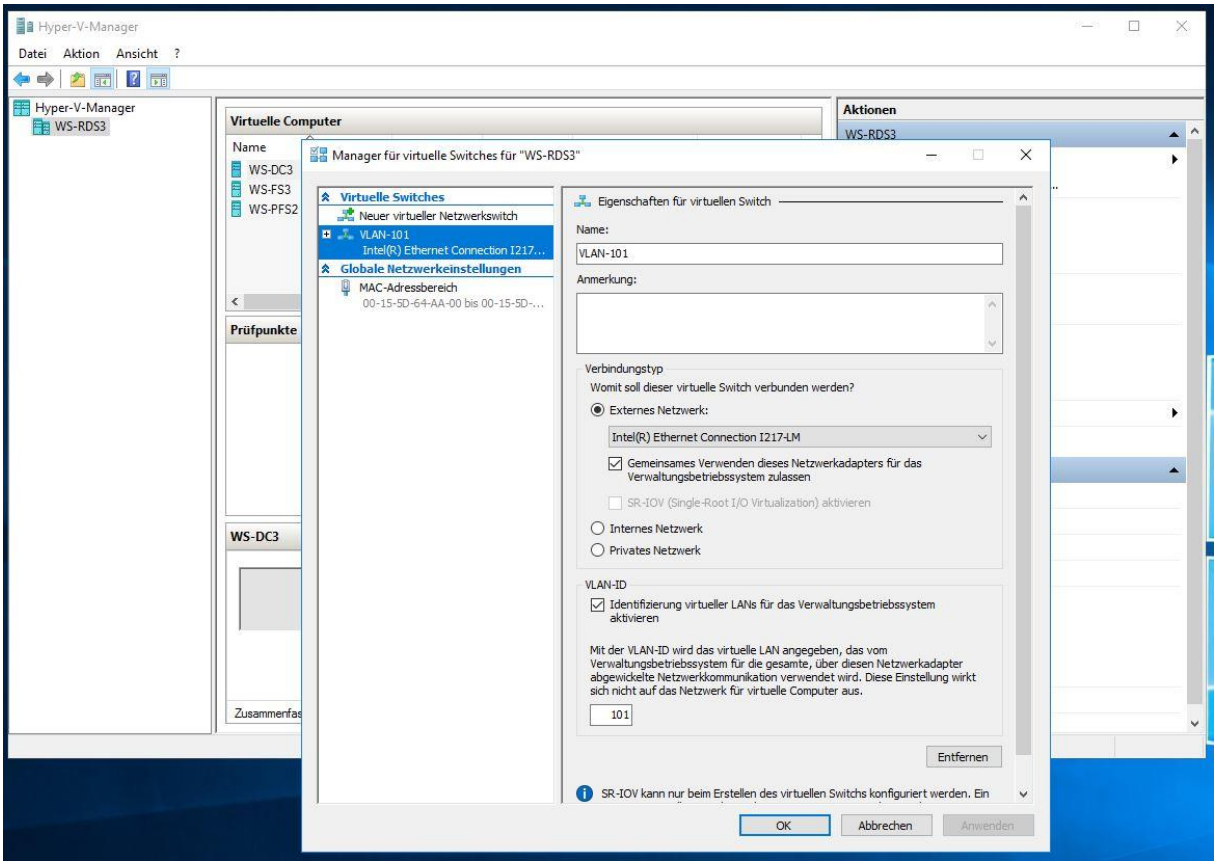
Vorbereitung

Die Herauslösung des Fileservices ist abgeschlossen. Damit stellt der Server WS-RDS3 nur noch den Service Hyper-V zur Verfügung. Aktuell läuft er mit Windows Server 2016. In diesem Schritt wird er als Windows Server 2019 neu installiert.

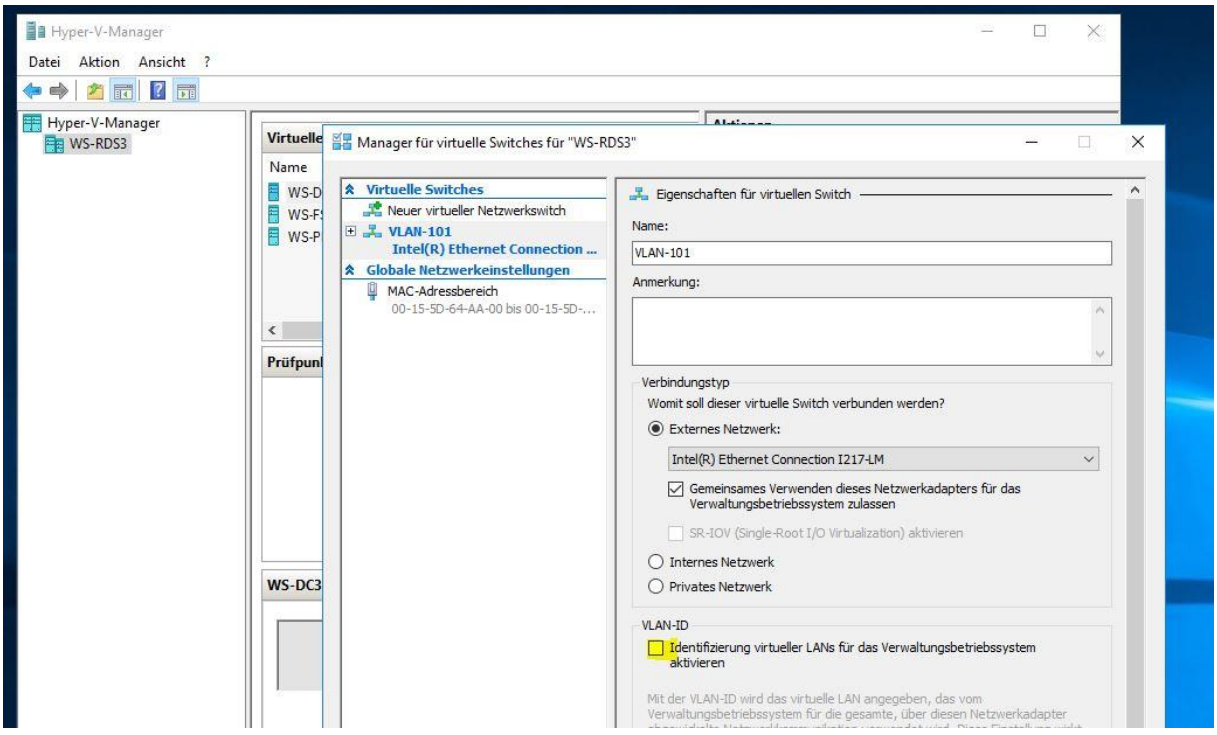
WS-RDS3 steht in meinem Außenstandort in Neufahrn Niederbayern. Dort ist er als Hyper-V-Host allein. Es gibt also keine weiteren Systeme, welche den Betrieb aufrechterhalten könnten. Das ist aber im Betriebsurlaub auch nicht erforderlich. Daher kann ich die Neuinstallation einfach vornehmen und die virtuellen Maschinen dabei ausgeschaltet lassen.

Der Server ist recht kompakt. Seine Hardware ist für den aktuellen Einsatz ausreichend. Nur die SSD ist mit 120GB Größe sportlich vollgelaufen. Daher werde ich diese für die durch eine größere erneuern. Das vereinfacht die Migration, da ich die VMs auf der alten Platte nicht verschieben muss. Ich schließe die alte Platte einfach später mit USB an und kopiere die VMs.

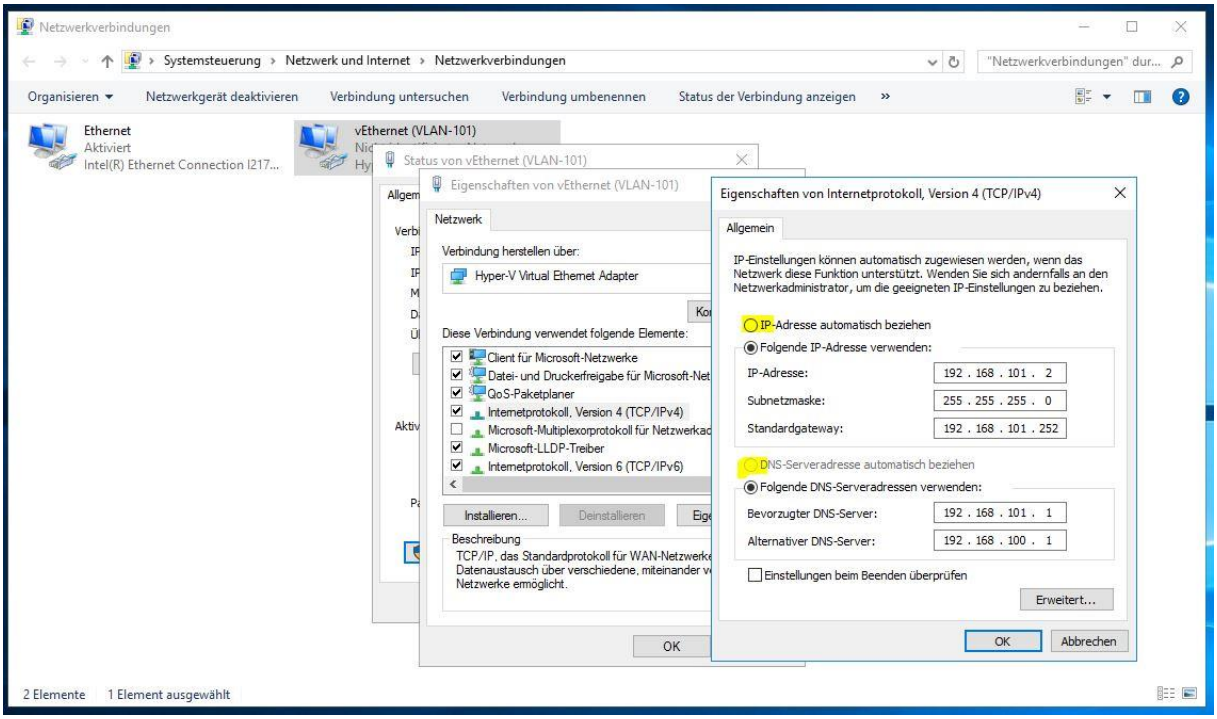
Den Umbau kann und möchte ich nicht im Außenstandort vornehmen. Daher modifiziere ich im ersten Schritt die Netzwerkkonfiguration des Servers. Aktuell ist nur eine Netzwerkkarte verbaut. Diese wird von den VMs und dem Hypervisor verwendet. Dennoch habe ich mein Netzwerk mit VLANs segmentiert. Die VLANs unterscheiden sich dabei natürlich von meinen im Hauptstandort. In Neufahrn verwenden die Server das VLAN 101:



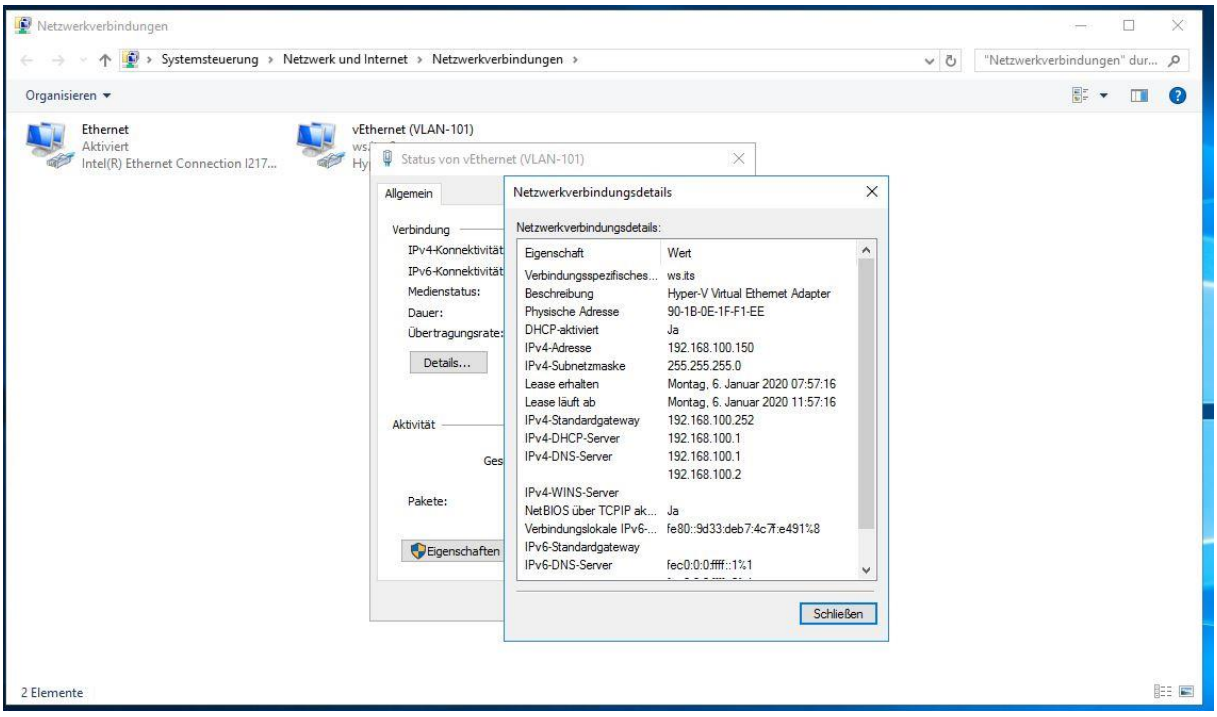
Diese Konfiguration nehme ich raus, damit ich daheim ungetagged anschließen kann:



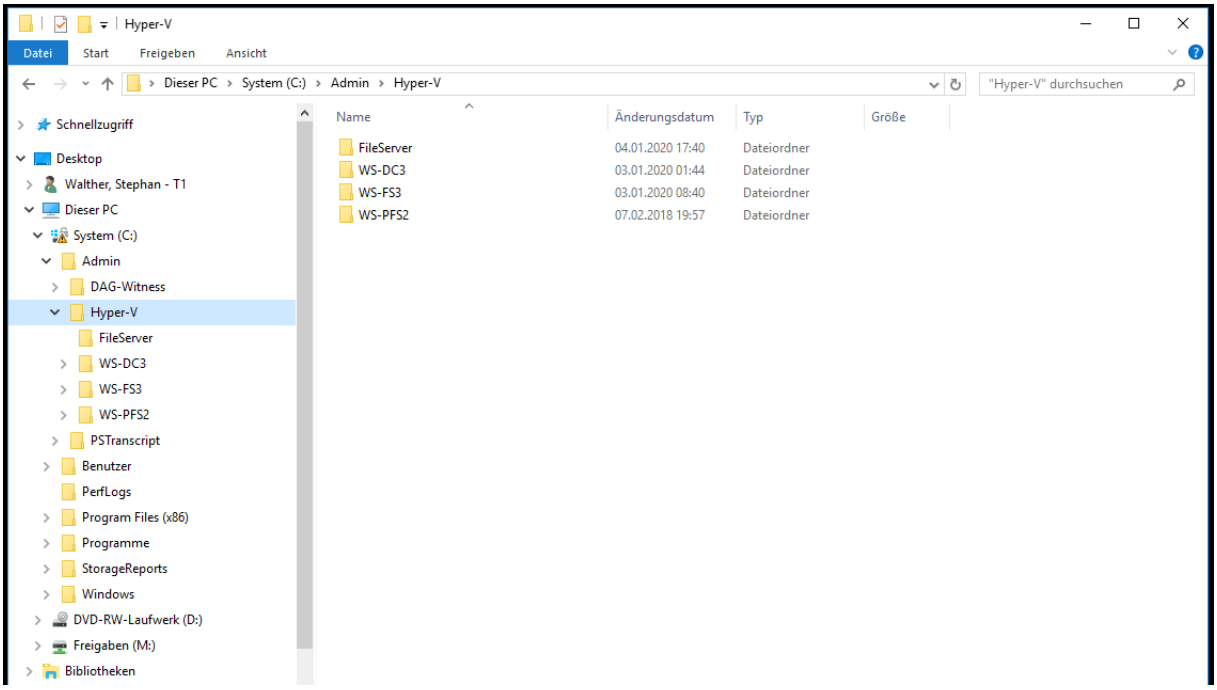
Dazu stelle ich die statische Konfiguration für IPv4 auf dynamisch um. So bekommt der Server daheim eine IP-Adresse vom DHCP und ist im Netz administrierbar:



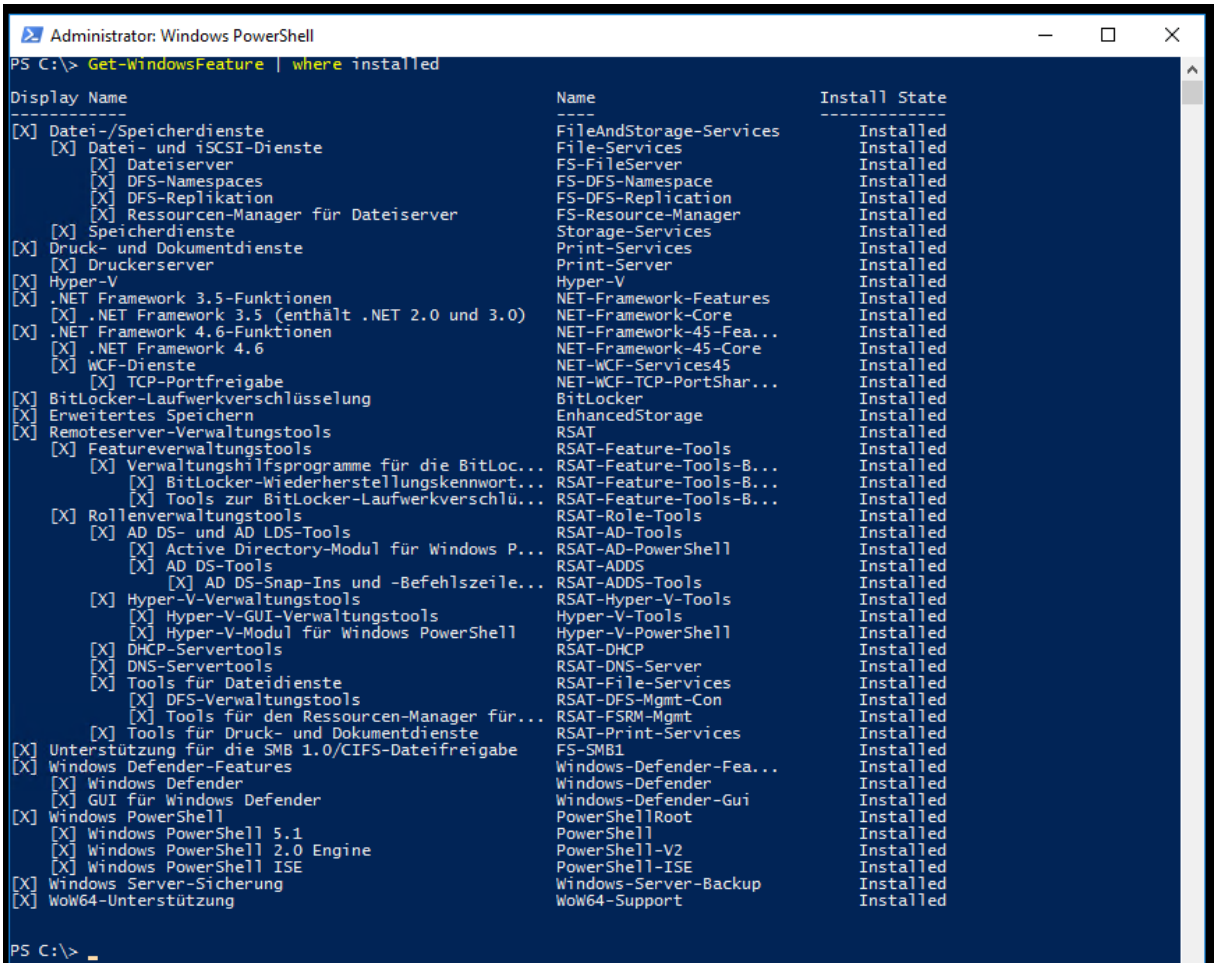
Jetzt fahre ich alle VMs herunter. Danach kann ich den Server WS-RDS3 selbst herunterfahren und abbauen. Wenig später ist der Server daheim ans Servernetz angeschlossen und hochgefahren. Wie erwartet hat er nun eine IPv4-Konfiguration aus meinem heimischen Servernetz:



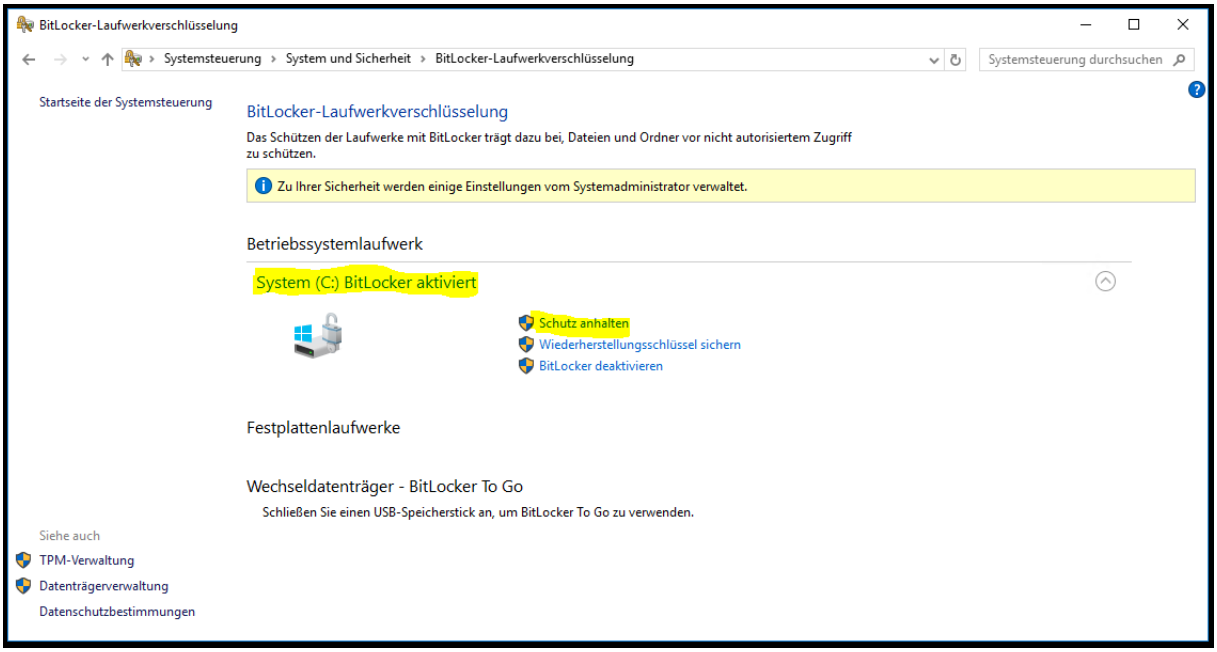
Es geht weiter mit der Sichtung der Bestandskonfiguration. Damals habe ich aus Kapazitätsgründen die virtuellen Maschinen direkt auf die C-Partition gelegt. Das hat nur Nachteile. Aber heute werde ich es richten. Ebenso liegt hier der Ordner mit den Freigaben – bis vor wenigen Tagen war WS-RDS3 ja noch selber der Fileserver (so konnte ich mir eine virtuelle Maschine und deren Platz sparen):

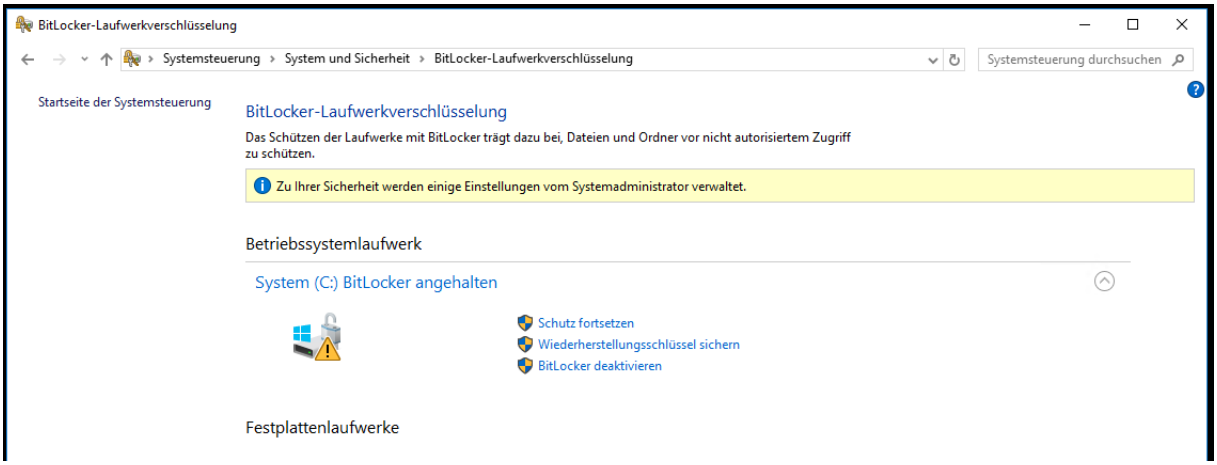


Demnach sind natürlich etliche Rollen und Funktionen auf dem kleinen Server installiert:

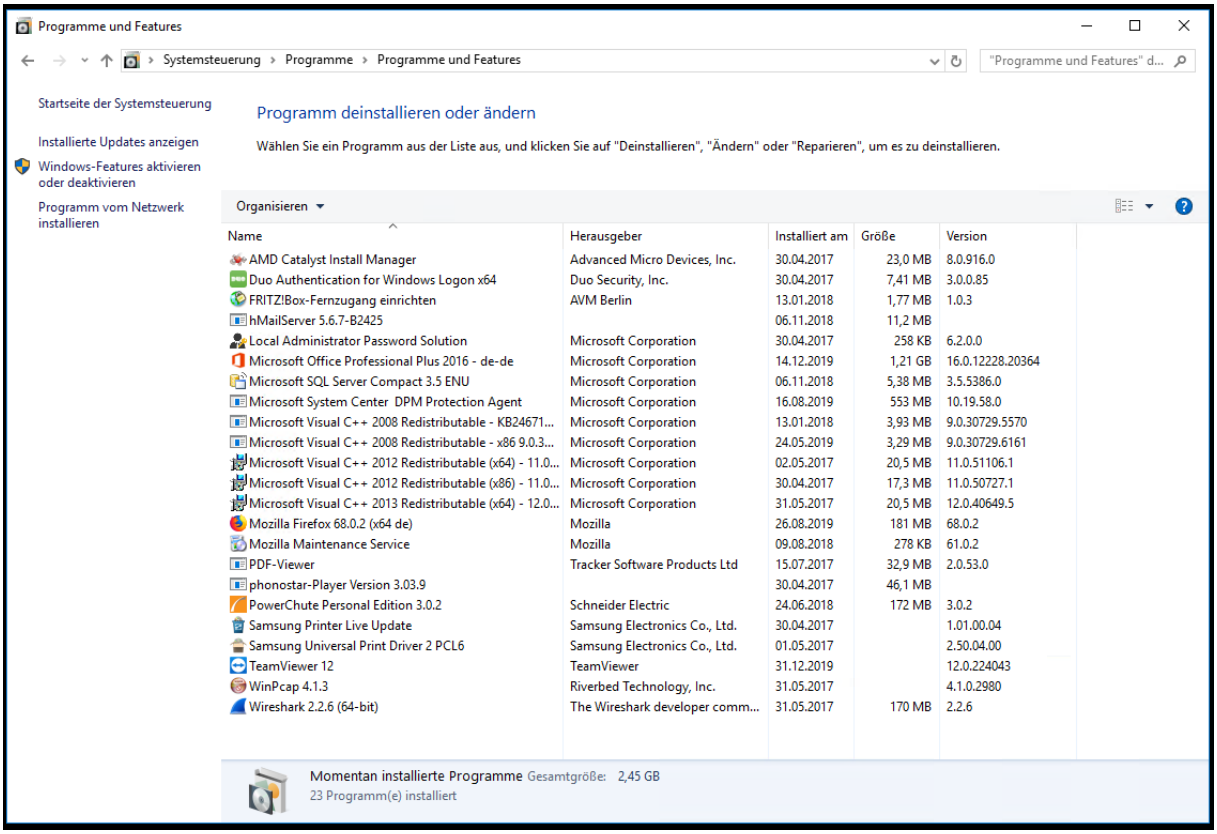


Die SSD mit dem Betriebssystem ist mit Bitlocker verschlüsselt. Damit ich später die VMs via USB auf den neuen Server kopieren kann, halte ich die Verschlüsselung an. Dabei bleibt der eigentliche Inhalt der SSD verschlüsselt. Aber der dazu verwendete Schlüssel liegt jetzt unverschlüsselt auf dem Volume. Daher geht diese Aktion auch schön schnell:

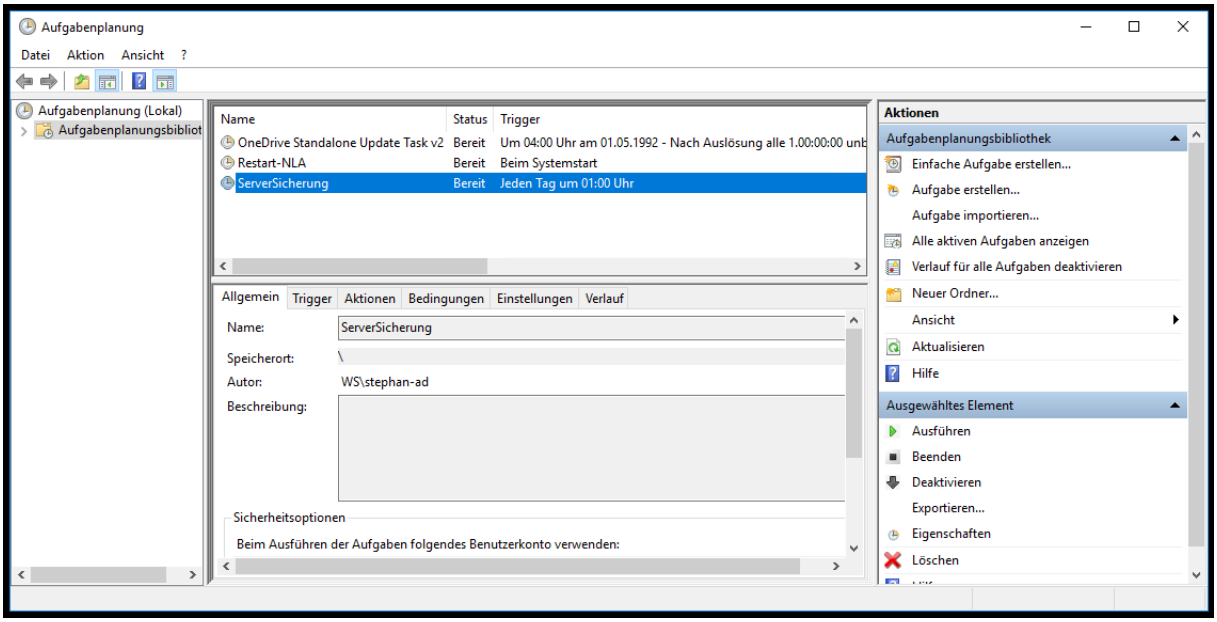




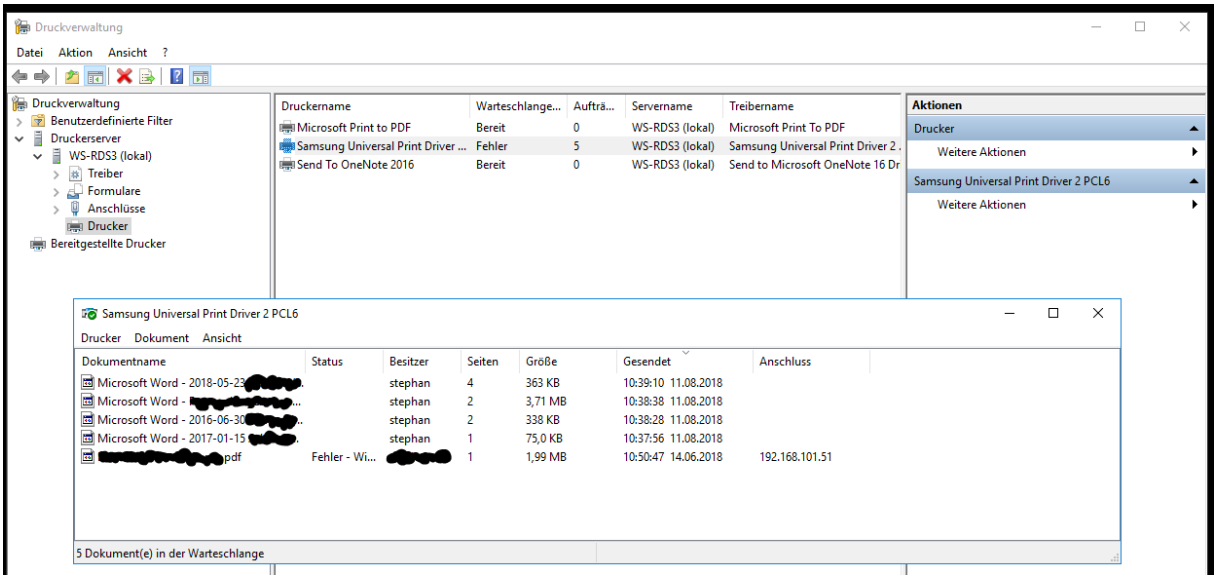
Ich prüfe noch, welche Anwendungen auf dem Server installiert sind. Davon benötige ich fast alles nicht länger:



In der Konsole Aufgabenplanung prüfe ich auf Konfigurationen. Vielleicht ist ja hier noch was Brauchbares dabei? Der Task „Restart-NLA“ ist interessant. Daher exportiere ich diesen.



Natürlich lief hier auch mein Printservice drauf. Der hat aber immer wieder Probleme verursacht. Daher habe ich den Client direkt mit dem Netzwerkdrucker verbunden. Man sieht immer noch die alten Druckjobs:

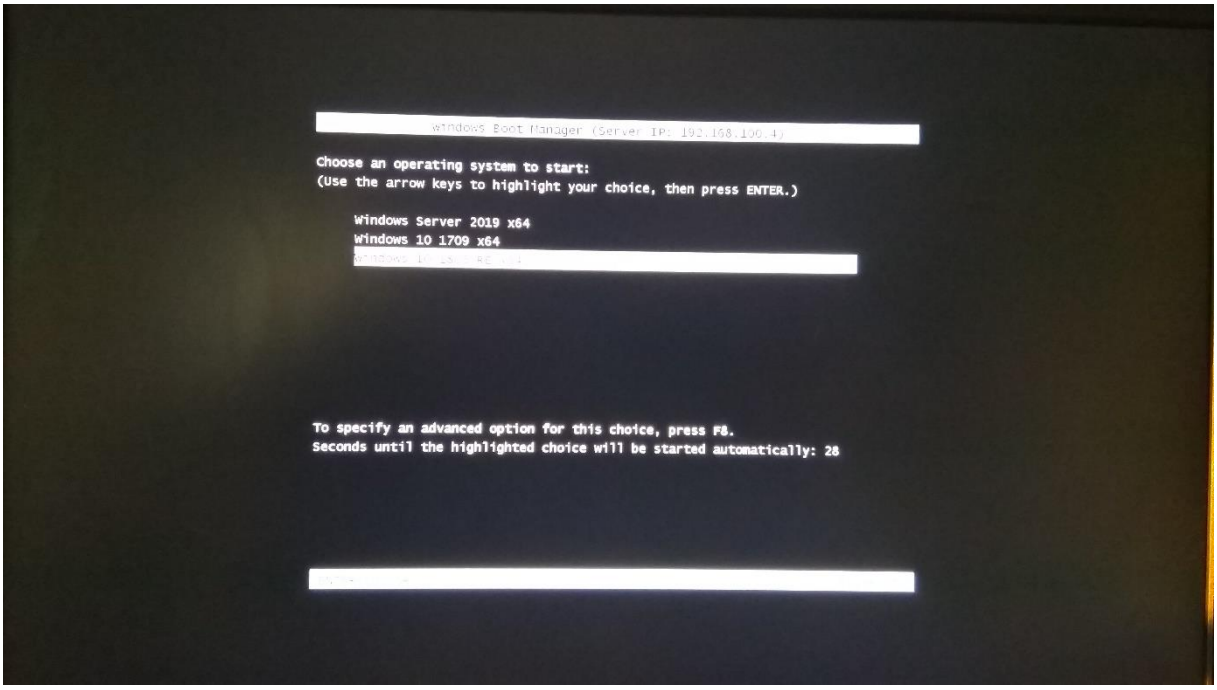


Ich denke, jetzt habe ich alles. Daher schalte ich den Server aus.

Einbau neue SSD und Neuinstallation als WS-HV3

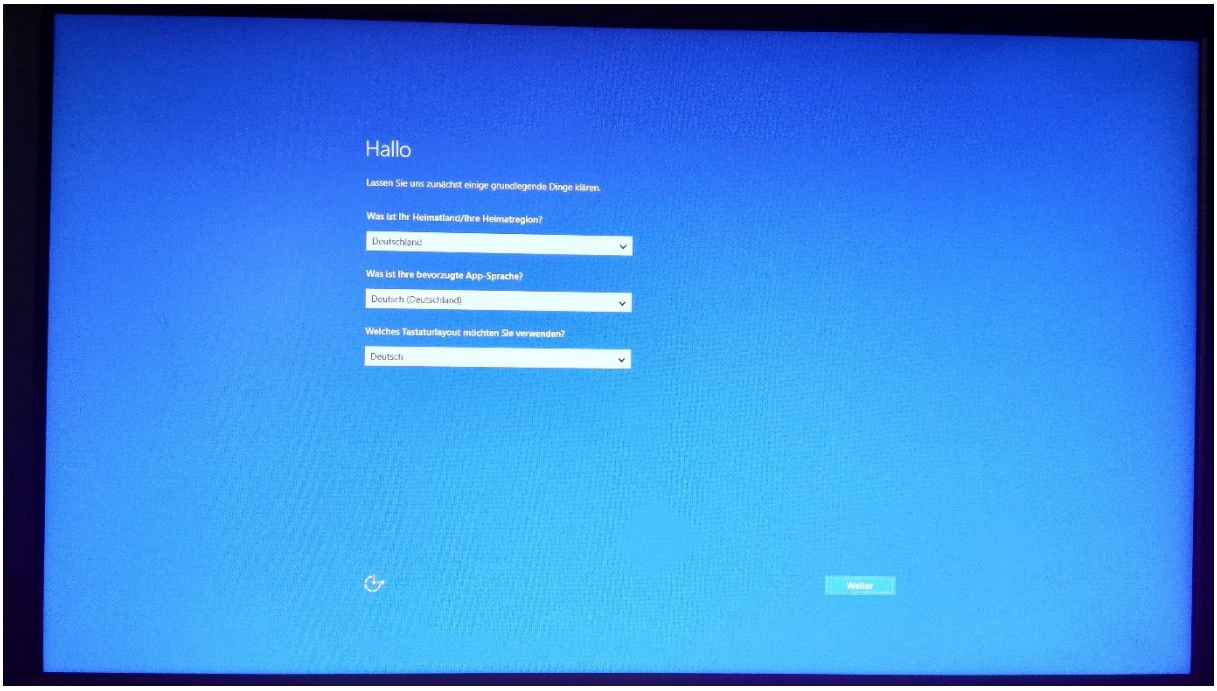
Ich trenne die Hardware von der Stromversorgung und entferne die alte SSD. Dann baue ich die neue ein. Diese hatt 500GB statt 120GB Kapazität. Danach schließe ich das System wieder an und starte den Server.

Aktuell ist er direkt mit meinem Servernetz verbunden. Daher startet er einen PXE-Boot mit IPv4. Hier wähle ich das Image für meinen Windows Server 2019 aus:

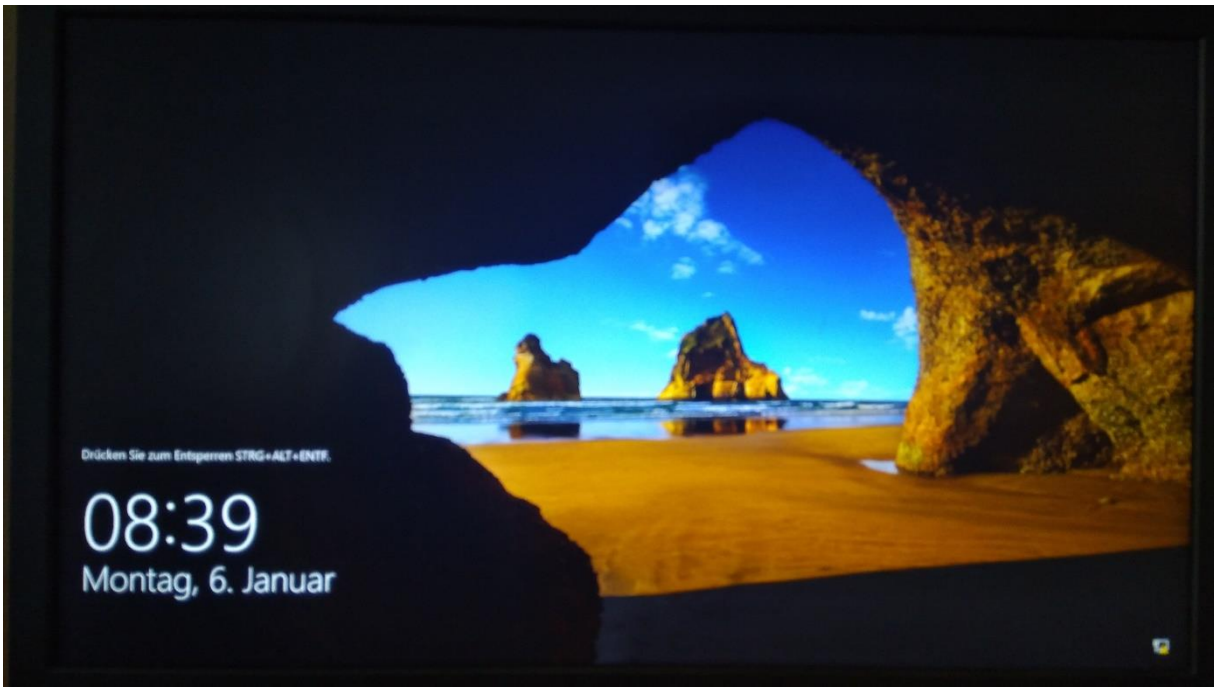


Die Bilder danach spare ich mir einmal. Ein Setup habt ihr bei mir schon oft gesehen.

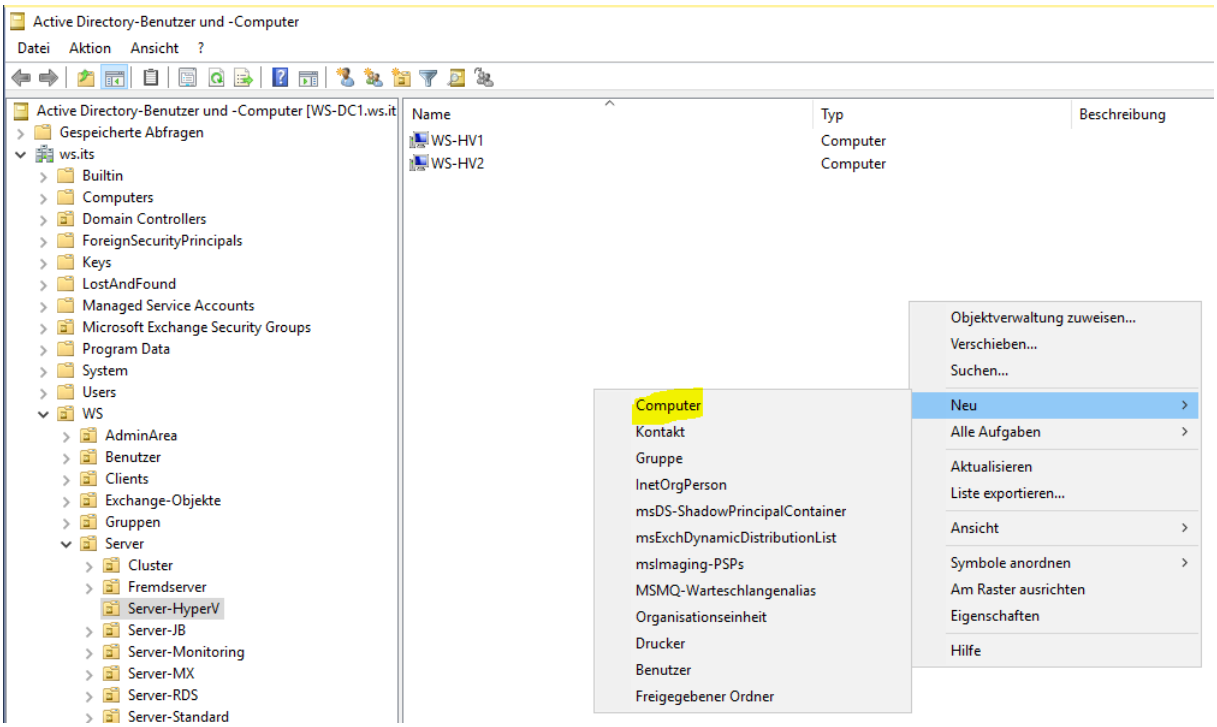
Für die Installation verwende ich nur 100GB. Den restlichen Speicher auf der SSD lasse ich frei. Kurze Zeit später beende ich die Installation:



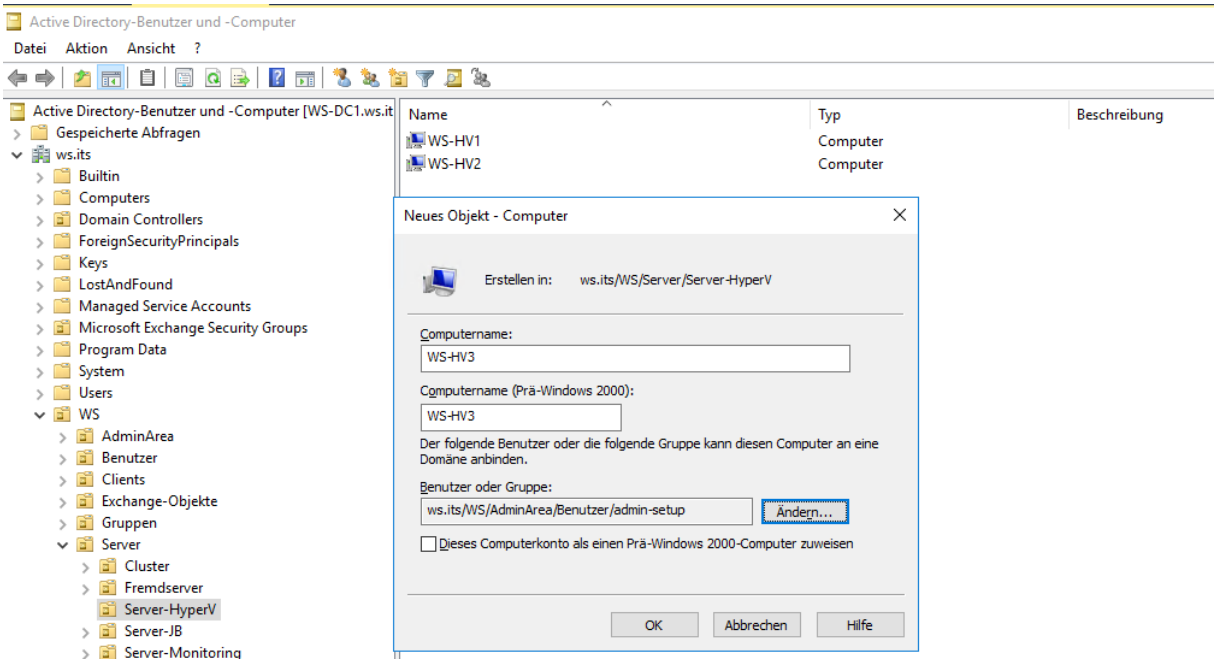
Nach der Eingabe des Passwortes ist das System aktiv:



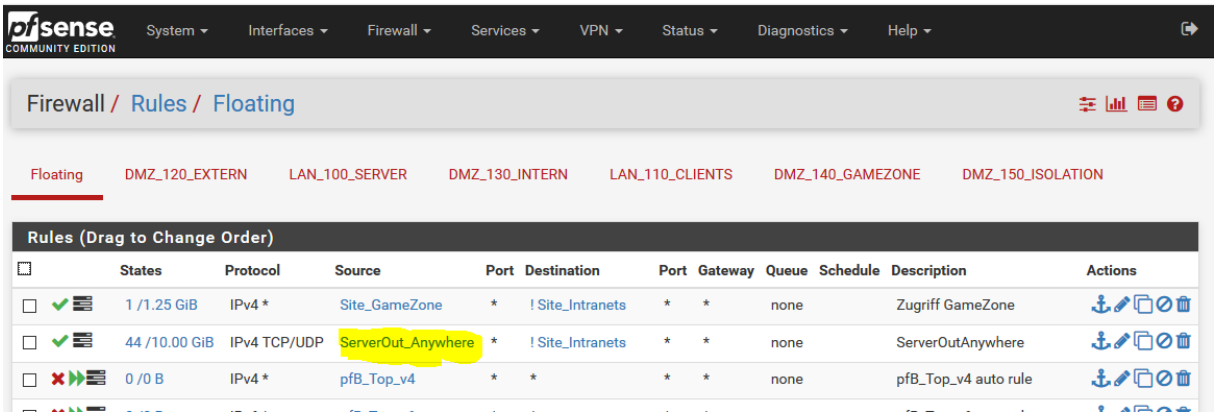
Für den Domain Join bereite ich ein Computerkonto im Active Directory vor. Den Namen WS-HV3 hatte ich ja bereits durch das Umbenennen der beiden anderen Hyper-V-Hosts freibekommen:



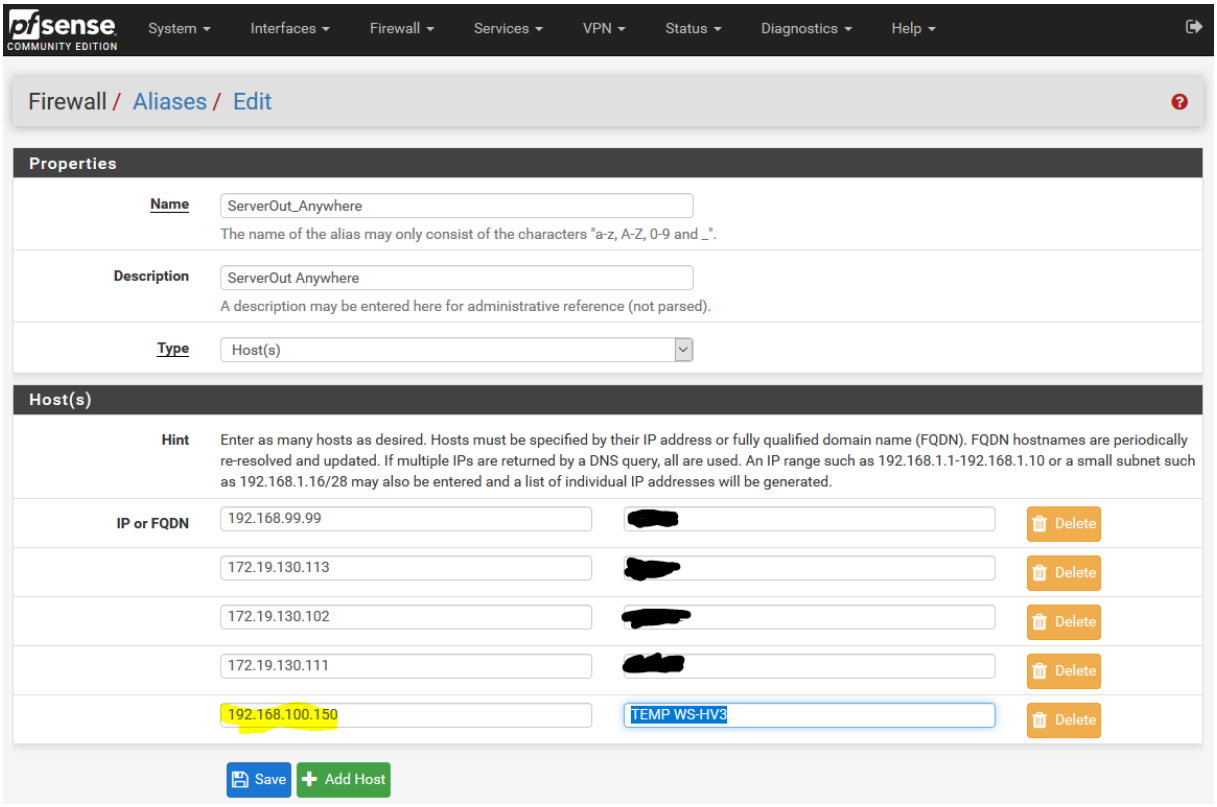
Den Domain Join führe ich mit einem Benutzeraccount aus:



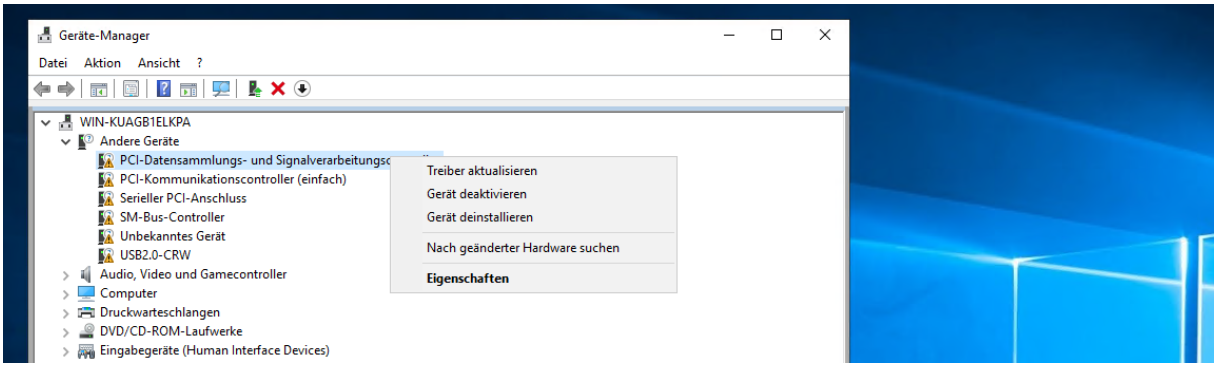
Für die Aktivierung des Betriebssystems muss der Server das Internet erreichen können. Das erlaubt meine Firewall standardmäßig nicht. Daher nehme ich den Server kurzfristig in eine vorbereitete Gruppe auf. So kann das System auch Treiberaktualisierungen von Microsoft herunterladen. Ich suche die erforderliche Firewall-Ausnahme in meiner PFSense. Ein Klick auf den Namen des Alias bringt mich zur Ausnahmeliste:



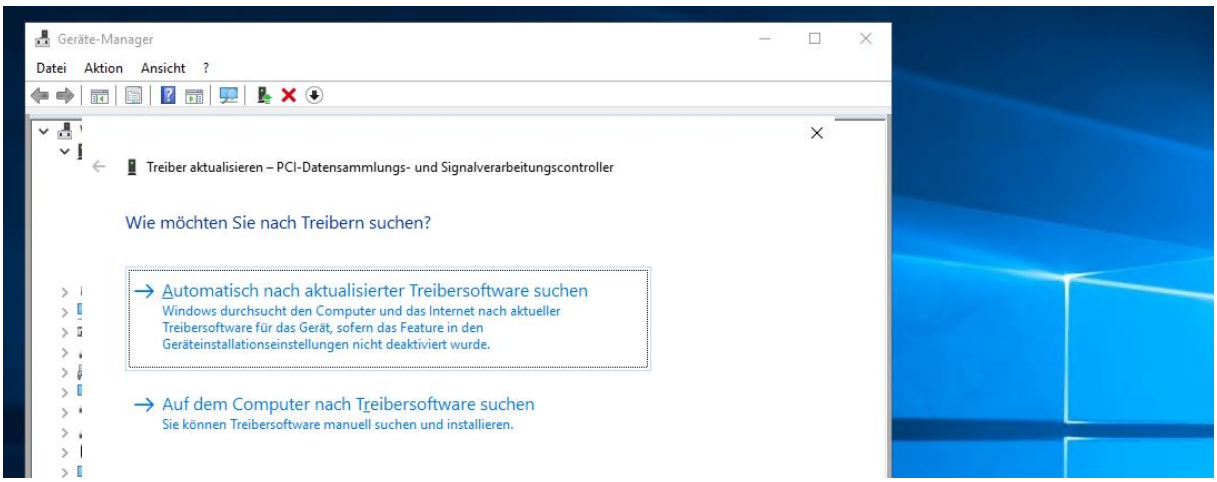
Und hier trage ich die aktuelle IP-Adresse des neuen Servers ein:



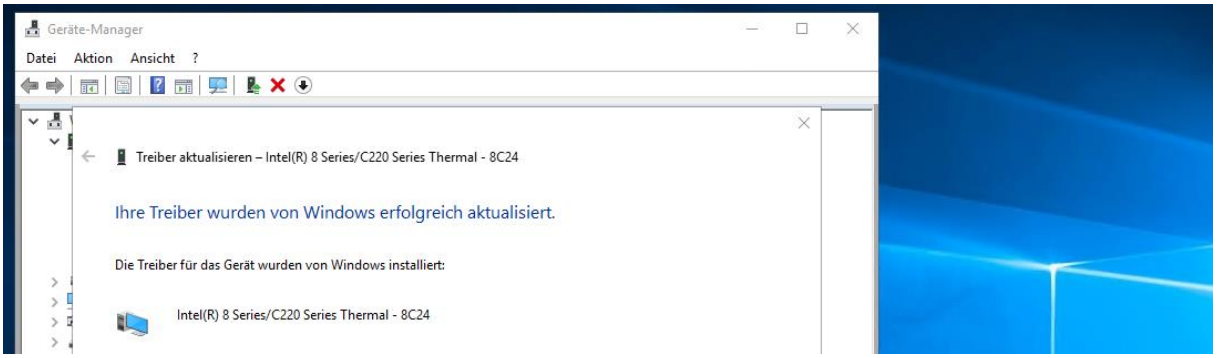
Die Netzwerkkarte hat das System automatisch erkannt. Aber einige andere Teile fehlen. Für diese starte ich die Treiberaktualisierung:



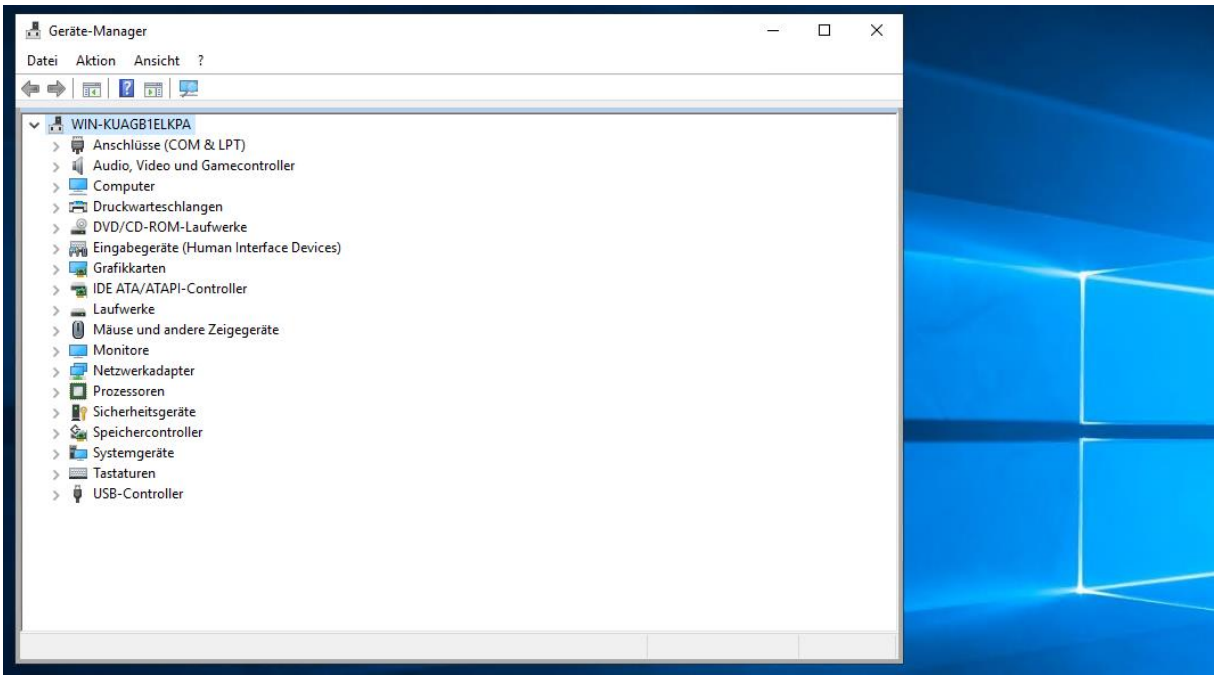
Das System kann diese bei Microsoft selbst suchen:



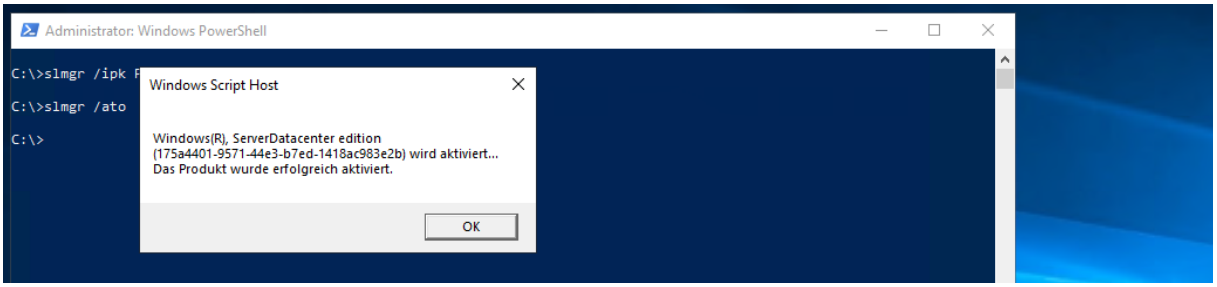
Wenige Sekunden später ist dieses Element installiert:



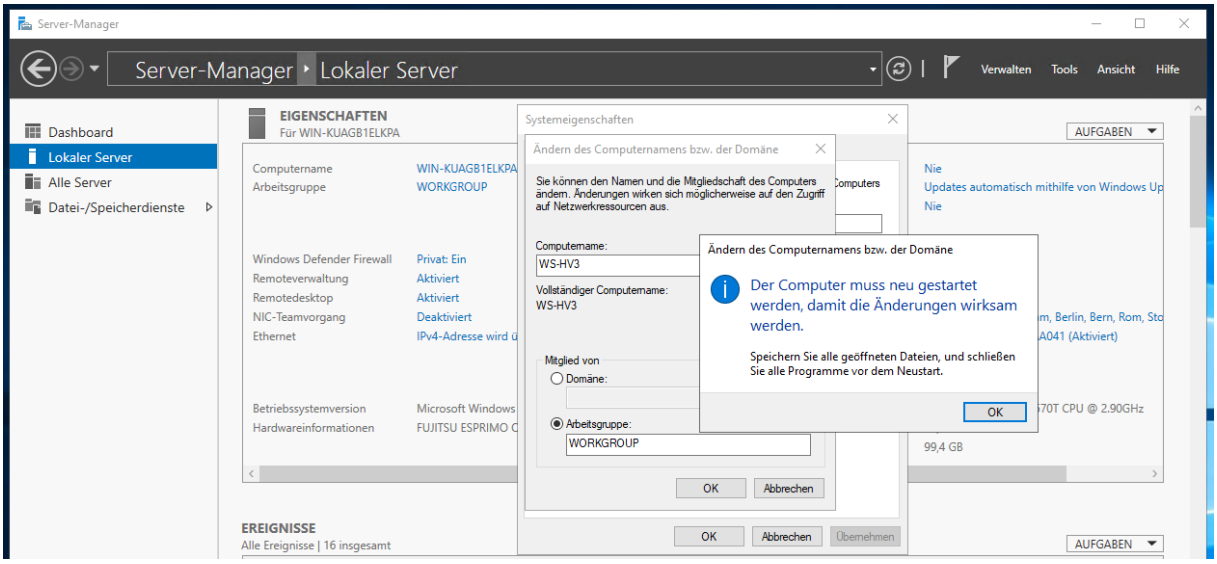
Und so verfahre ich mit allen anderen Komponenten. Jetzt ist alles einsatzbereit:



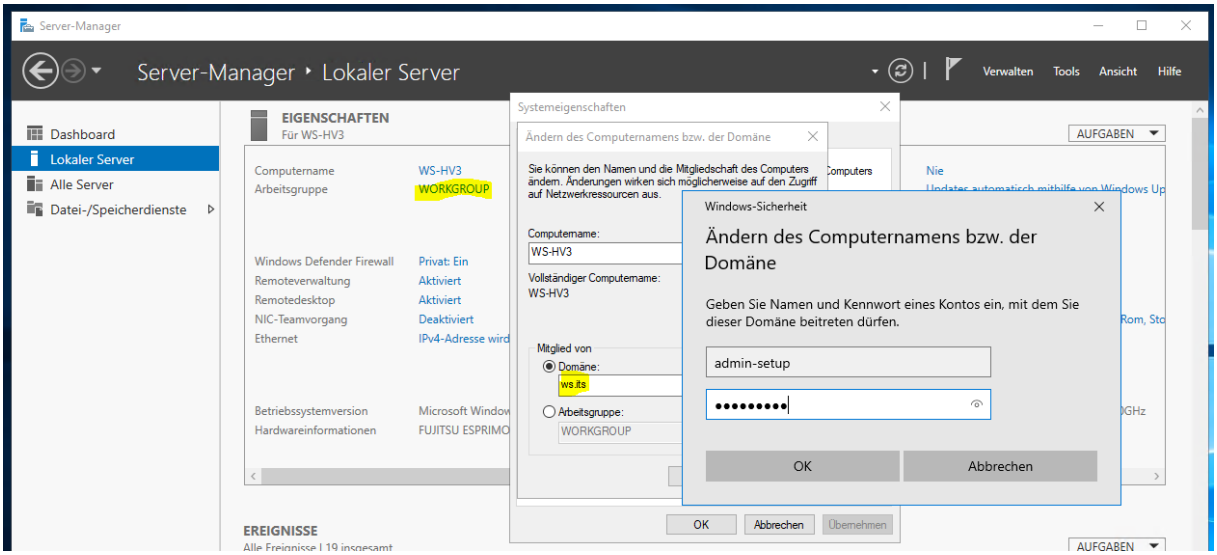
Mit der Internetverbindung kann auch die Aktivierung des Windows Servers vorgenommen werden. Ich installiere den Produktschlüssel mit der cmd. Anschließend aktiviere ich die Installation:



Danach bekommt der Server seinen neuen Namen WS-HV3. Die Aktion schließe ich mit einem Neustart ab:



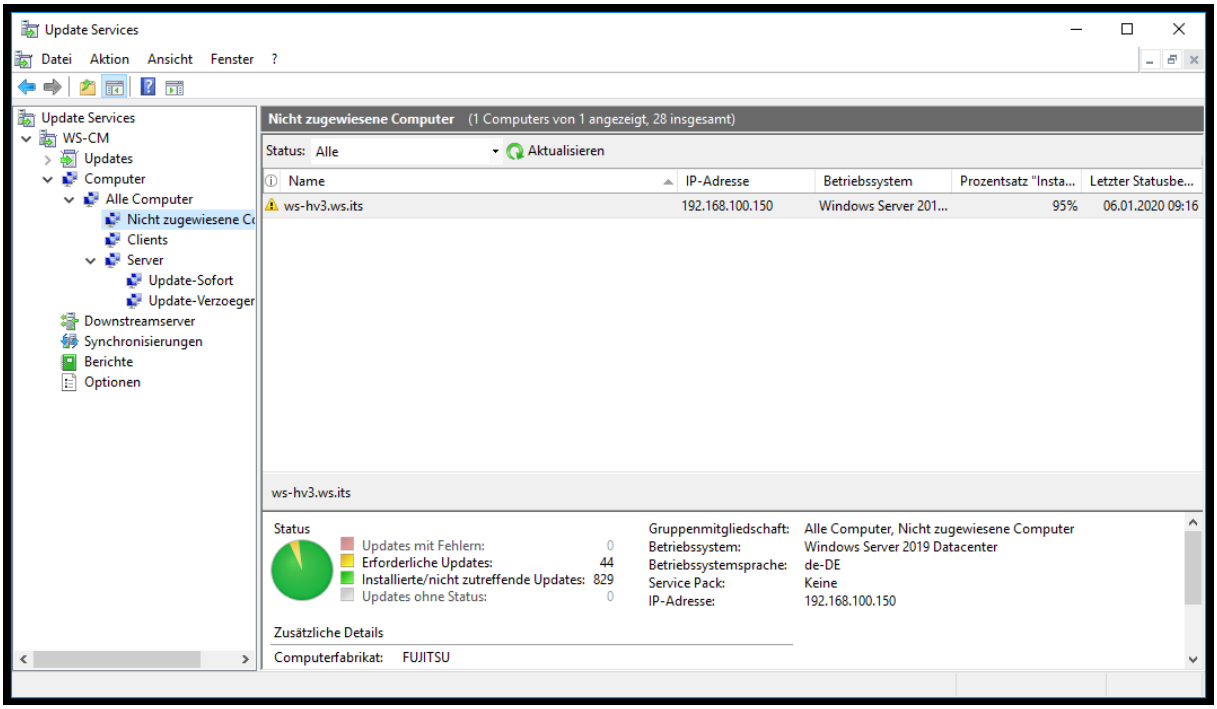
Nach der Neuanmeldung nehme ich den Server in meine Domain auf. Dabei verwende ich den Account, den ich beim Erstellen des Computerkontos angegeben habe:



Einen Neustart später ist das System Mitglied meiner Domain.

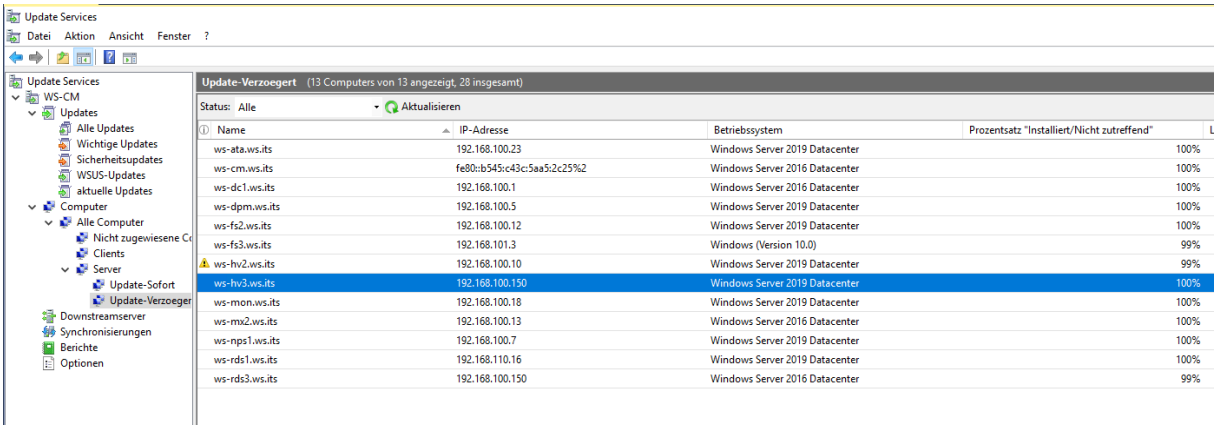
Jetzt fehlen noch die aktuellen Updates. Diese bekommt der Server dank Gruppenrichtlinien-Konfiguration von meinem WSUS-Server:

Der neue Server hat sich in meinem WSUS eingetragen. Der Container „nicht zugewiesene Computer“ bekommt bei mir immer alle Updates automatisch genehmigt. Damit werden neue Systeme immer auf den neusten Stand aktualisiert:



Für den Dauerbetrieb verwende ich 2 andere Server-Container. Diese bekommen zeitverzögert neue Updates genehmigt. So kann ich steuern, wann meine Server ihren erforderlichen Neustart ausführen. „Updates-sofort“ ist eine Woche vor „Updates-verzoegert“ dran. So kann ich schlechte Updates im laufenden Betrieb erkennen und verhindern, dass diese auf die Nachzügler angewendet werden. In der Gruppe „Updates-sofort“ stehen demnach nur Server, deren Services hochverfügbar sind.

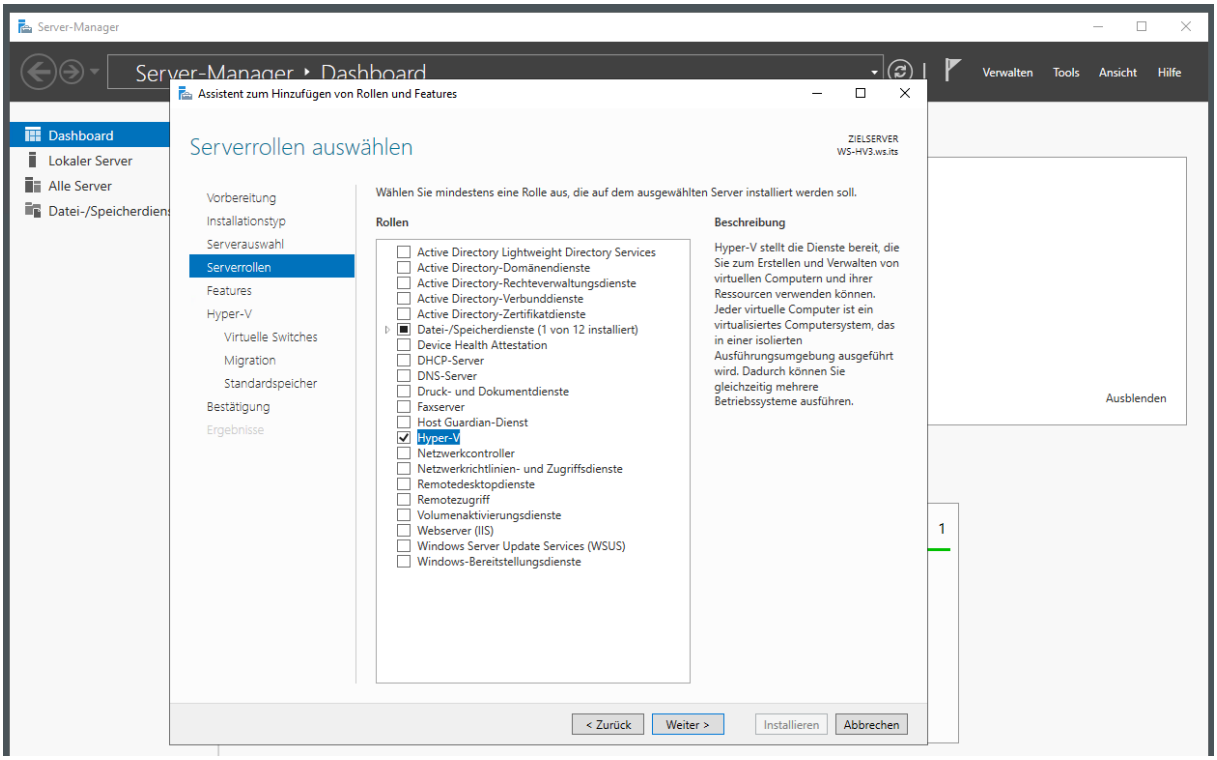
Mein Hyper-V-Service in Neufahrn wird aber nur vom Server WS-HV3 bereitgestellt. Daher bekommt er die Updates verzögert. Ich weise den Server der Gruppe zu:



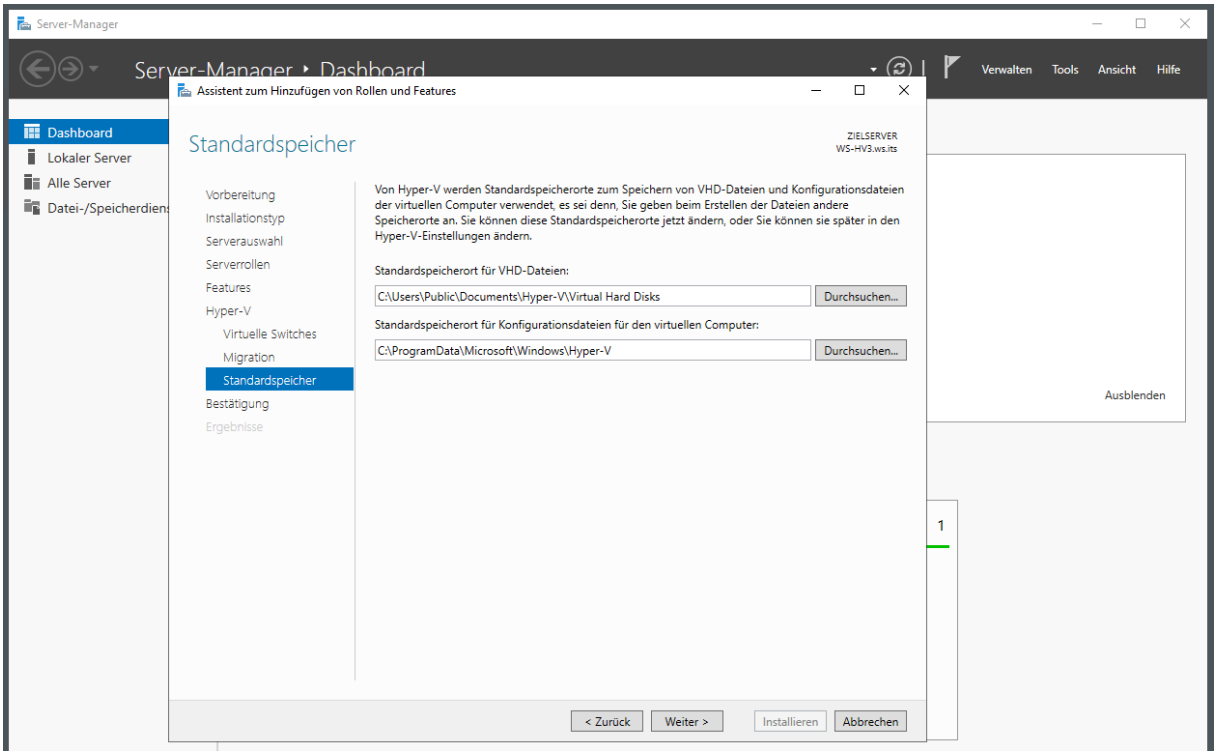
Der Server hat nun alle Updates installiert und den erforderlichen Neustart ausgeführt.

Installation der Rollen und Features

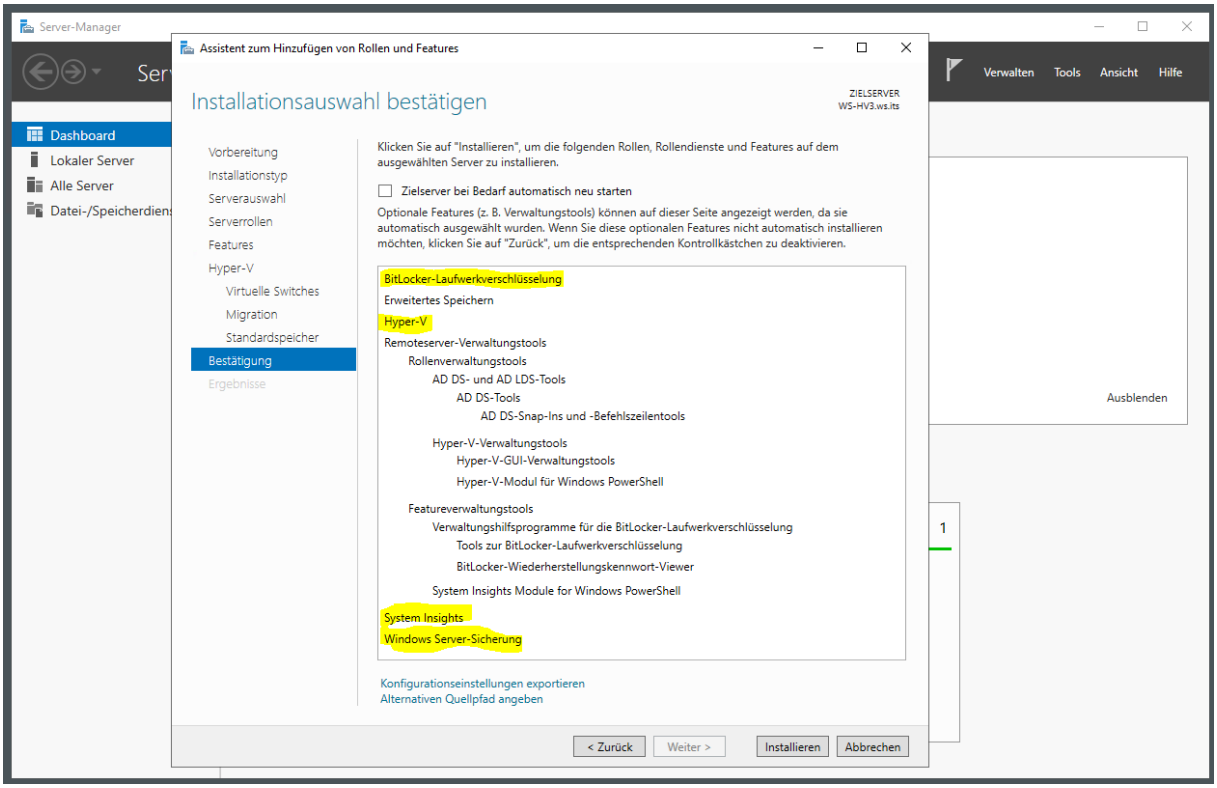
So kann es zur Rolleninstallation gehen. Die Auswahl wird dieses mal auf das notwendige beschränkt: Das System soll nur noch als Hyper-V-Host arbeiten:



Die Konfigurationsoptionen lasse ich bestehen. Das mach ich später richtig:



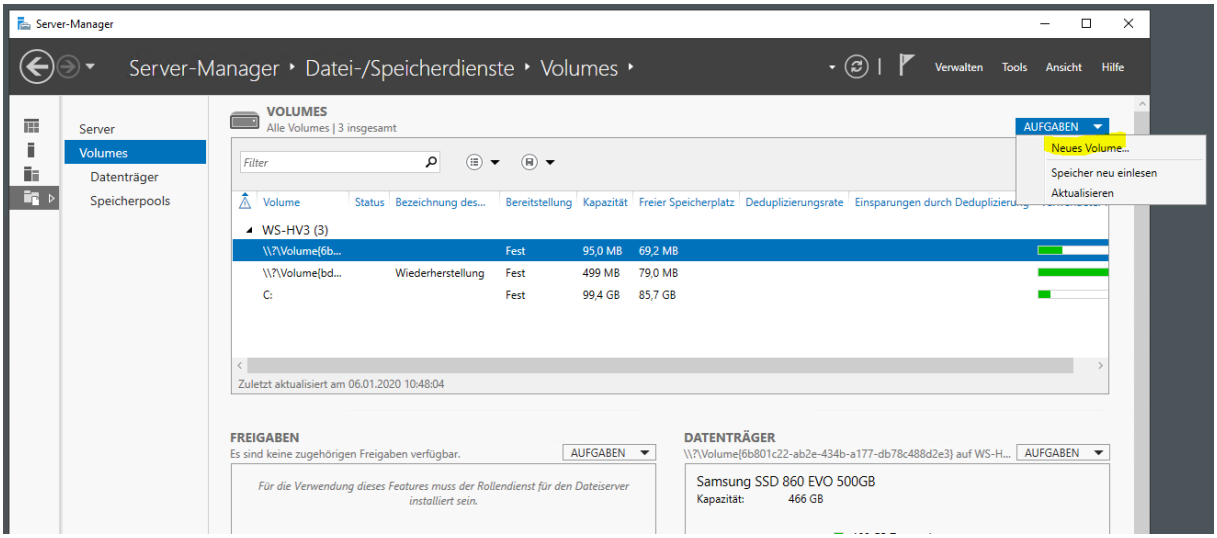
Aus den Features wähle ich noch den Bitlocker-Support dazu. Diesen benötige ich für die Festplattenverschlüsselung. Mit System Insights kann mein Windows Admin Center Prognosen zur Hardwareauslastung erstellen. Und die Windows Server Sicherung verwende ich für das Backup des SystemImages:



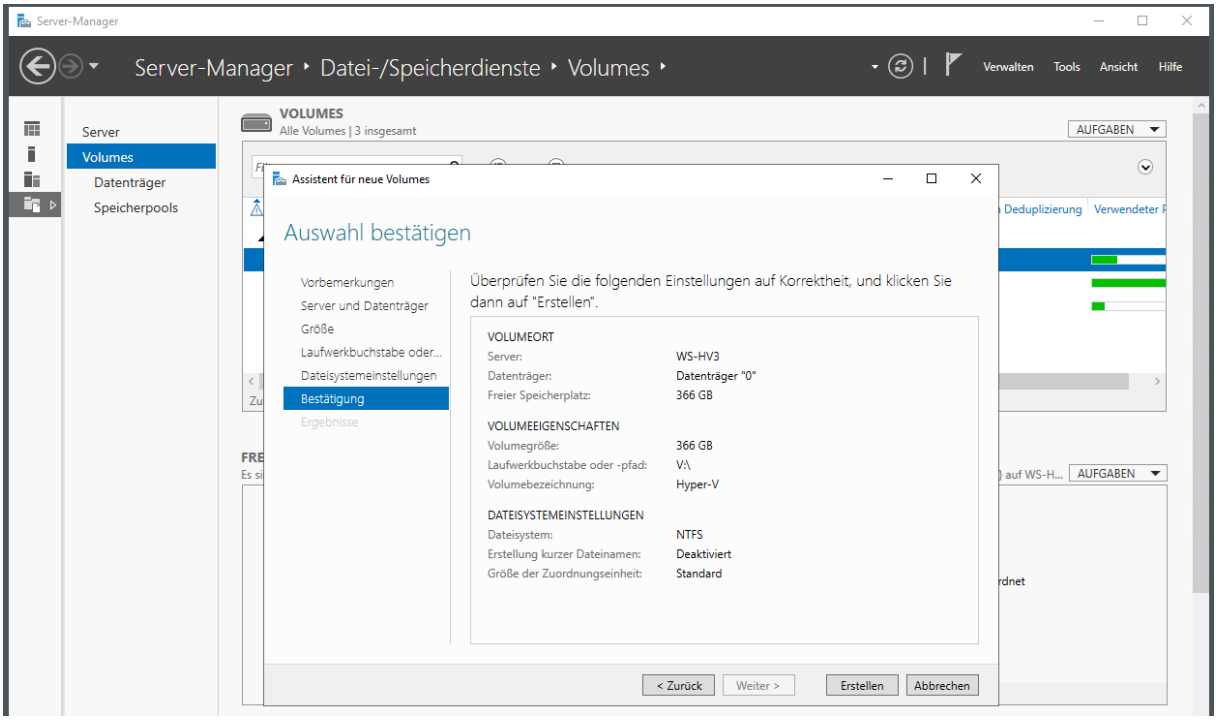
Die Rolleninstallation schließe ich mit einem Neustart ab.

Konfiguration von Hyper-V und Migration der VMs

Weiter geht es mit der Einrichtung des Hyper-V-Hosts. Die virtuellen Maschinen sollen nicht wie zuvor auf der Systempartition liegen. Dafür habe ich beim Setup reichlich Platz auf der SSD gelassen. Diesen freien Speicher nutze ich jetzt für ein neues Volume. Das erstelle ich mit dem Server-Manager:

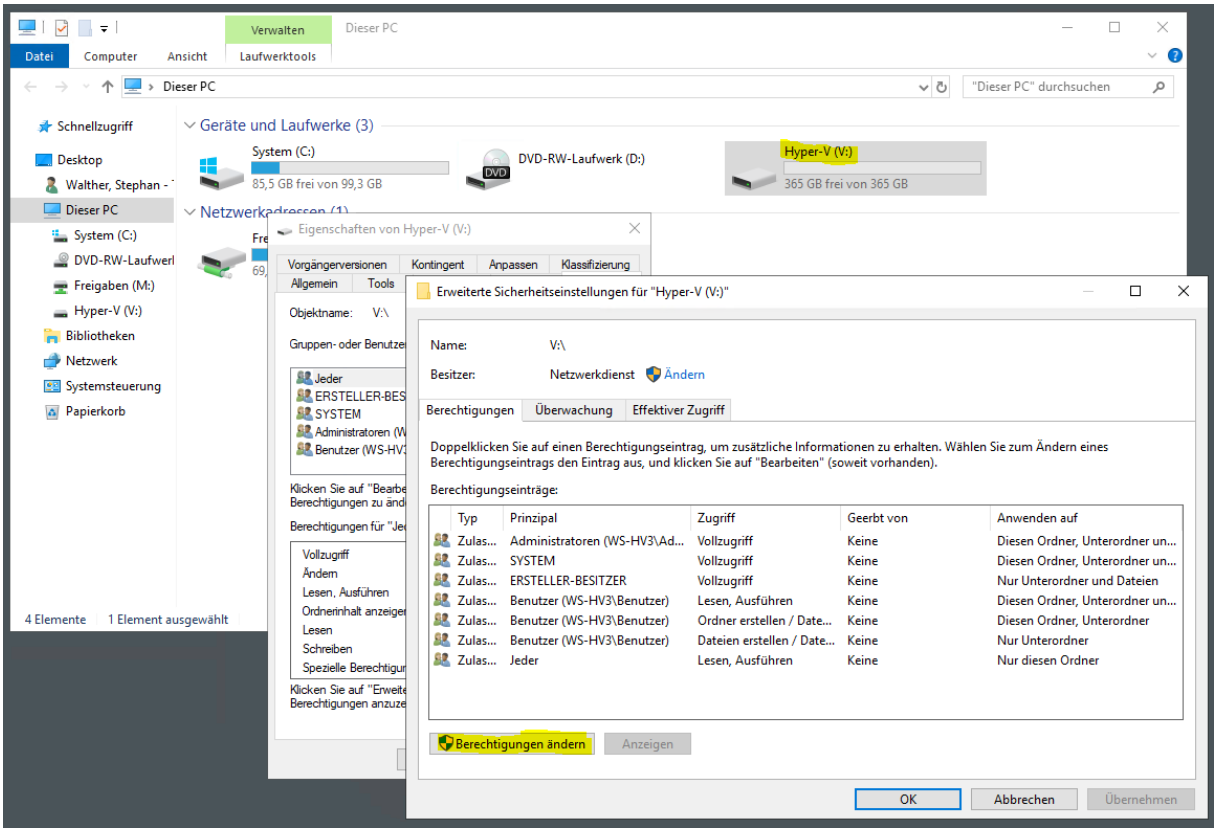


Ich verwende den verbliebenen Speicher, formatiere mit NTFS und vergebe den passenden Laufwerksbuchstaben V (wie VM). Diesen verwenden meine anderen Hyper-V-Hosts auch:

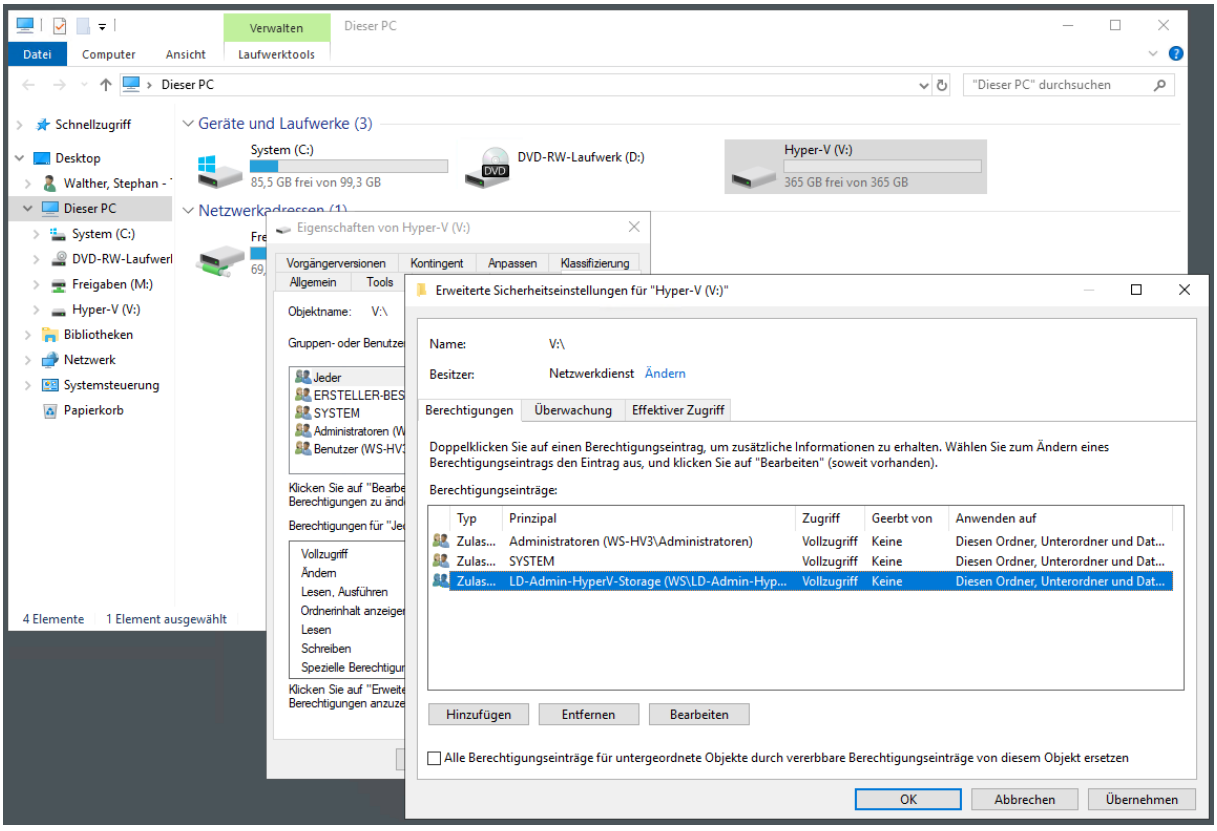


Das neue Volume wird alle virtuellen Maschinen aufnehmen. Für den Zugriff auf das Volume möchte ich das Modell der Rollen-Administration verwenden. Meine administrativen Accounts haben keine statischen Gruppenmitgliedschaften. Mit meiner Privileged Access Management Lösung kann ich die erforderlichen Gruppenmitgliedschaften auf eine begrenzte Zeit verwenden. Damit sichere ich meine administrativen Konten ab. Und weiter gehe ich mit der Einschränkung des Datenzugriffs: Meine AdminAccounts haben auf Datenverzeichnisse keinen Zugriff, bis ich sie zusätzlich in eine Rollengruppe aufnehme.

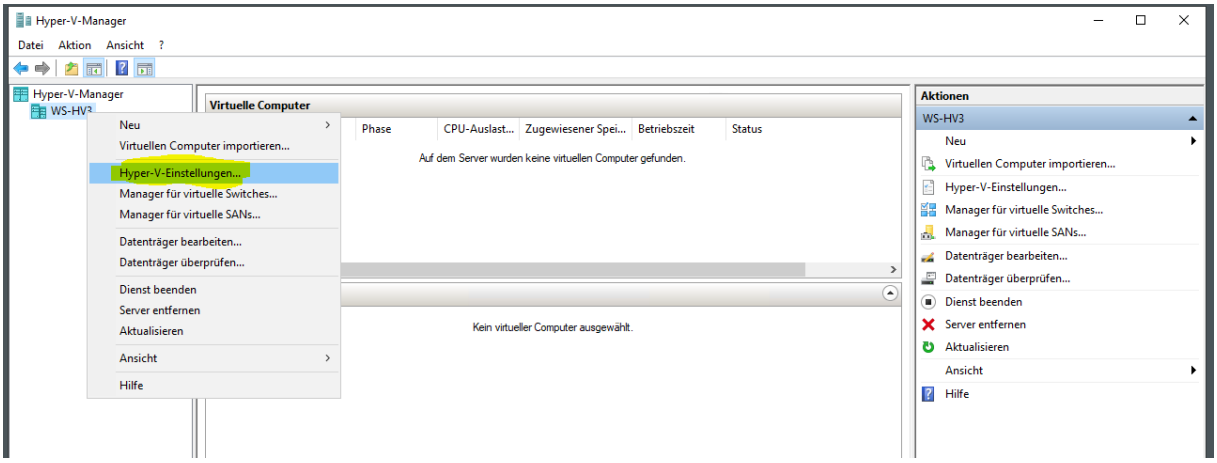
Das muss natürlich vorbereitet sein. In diesem Fall handelt es sich um das Volume mit den virtuellen Maschinen. Da sollen nur Mitglieder der Rechtegruppe LD-Admins-HyperV-Storage Zugriff erhalten. Die Gruppe gibt es schon. Also editiere ich nur noch die NTFS-Berechtigungen. Zuerst entferne ich die Vererbung:



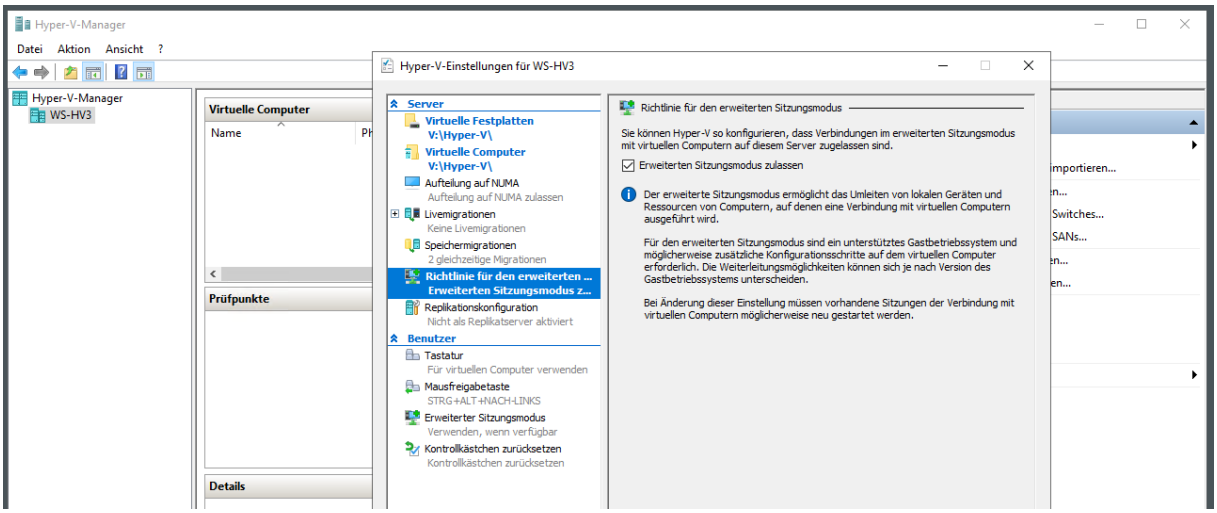
Dann schränke ich den Zugriff ein:



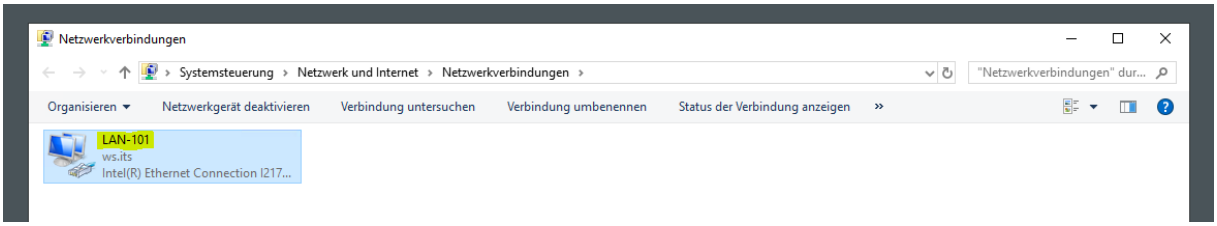
Das wars auch schon. Jetzt geht es an die Feinkunfiguration des Hyper-V-Hosts:



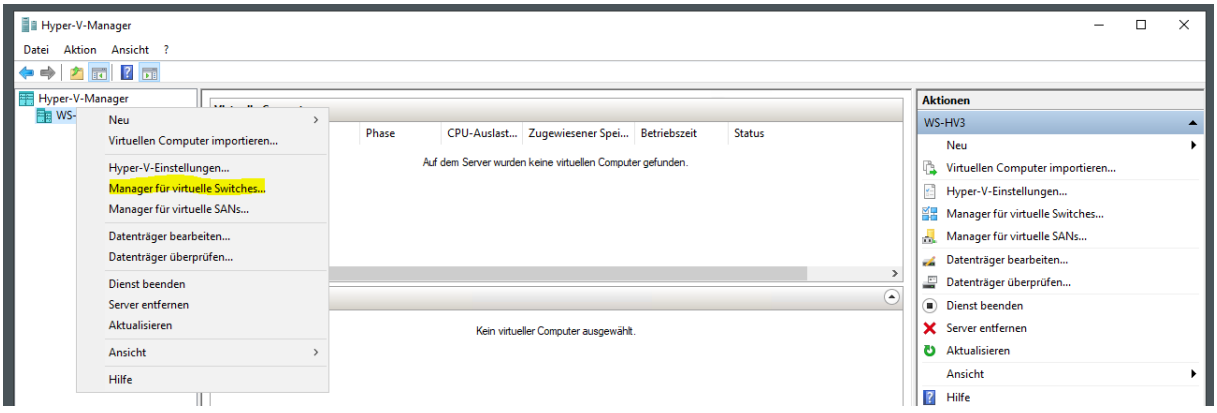
Den Standardspeicherort lege ich auf das neue Volume V: und zusätzlich aktiviere ich den erweiterten Sitzungsmodus. Mit diesem steht mir eine RDP-ähnliche Verbindung zu meinen VMs zur Verfügung:



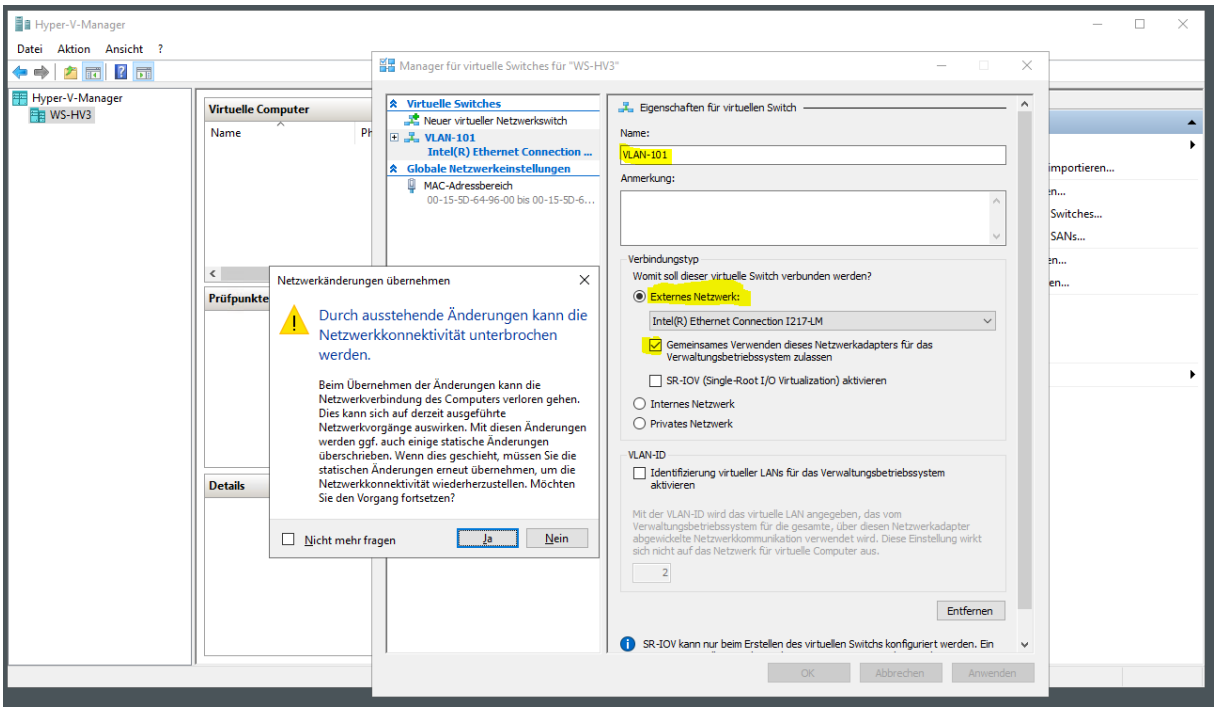
Den einen Netzwerkadapter benenne ich in der Systemsteuerung um. Das macht eigentlich keinen Sinn, aber so ist es mit den anderen Hyper-V-Hosts gleich konfiguriert. Ich liebe Standards…



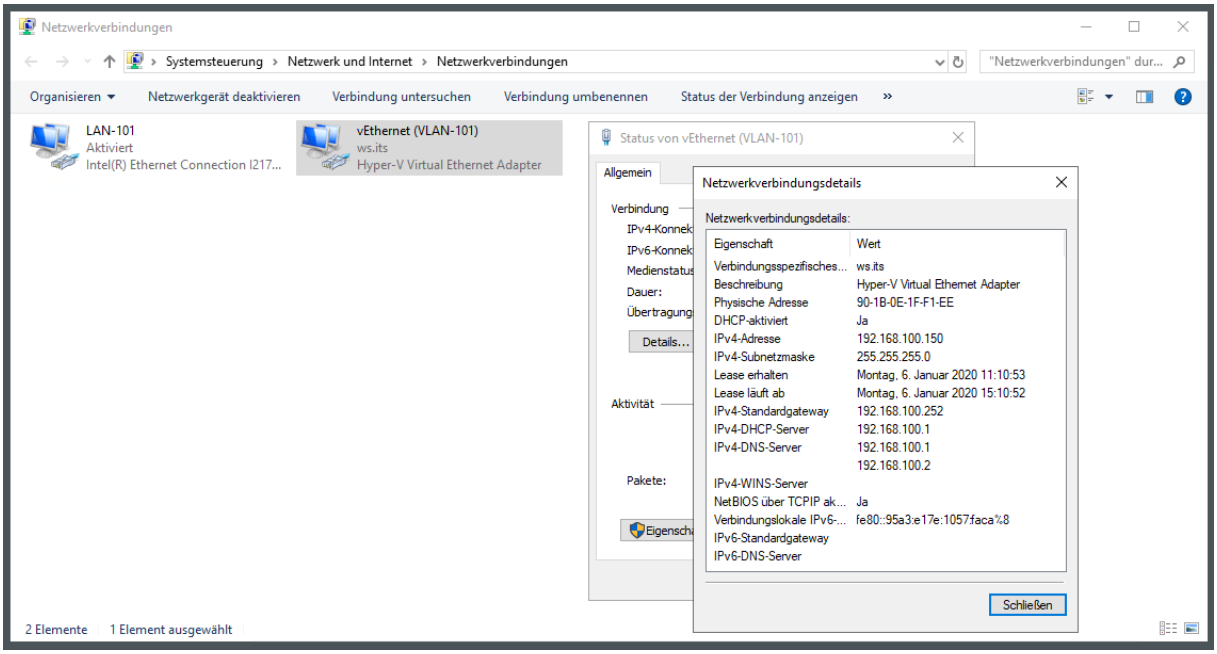
Nun aktiviere ich auf diesem Adapter einen neuen virtuellen Switch:



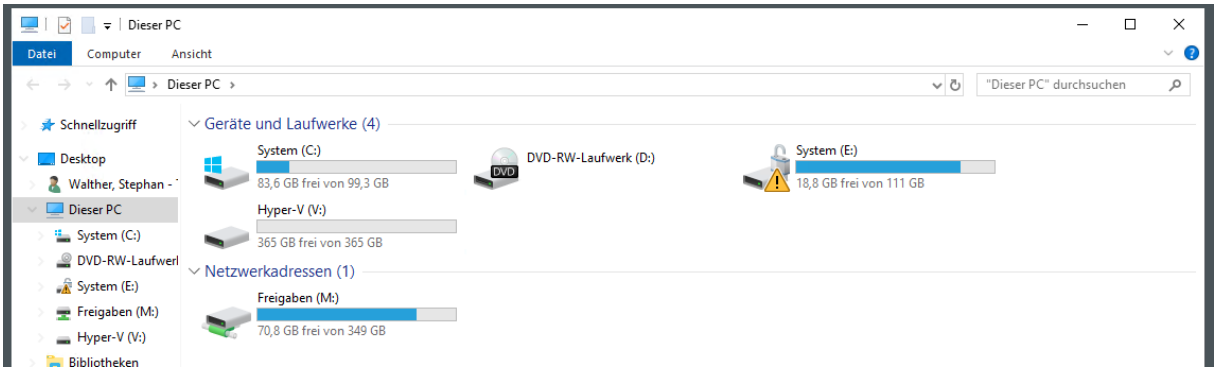
Dieser muss als externer Switch fungieren. Ebenso muss ich den gemeinsamen Zugriff aktivieren. Denn anderenfalls kann mein Hyper-V-Host selber nicht mit dem Netzwerk kommunizieren. Das VLAN lasse ich noch raus. Wichtig ist, dass der vSwitch den gleichen Namen hat wie auf dem alten Server. Sonst bekomme ich beim VM-Import Probleme:



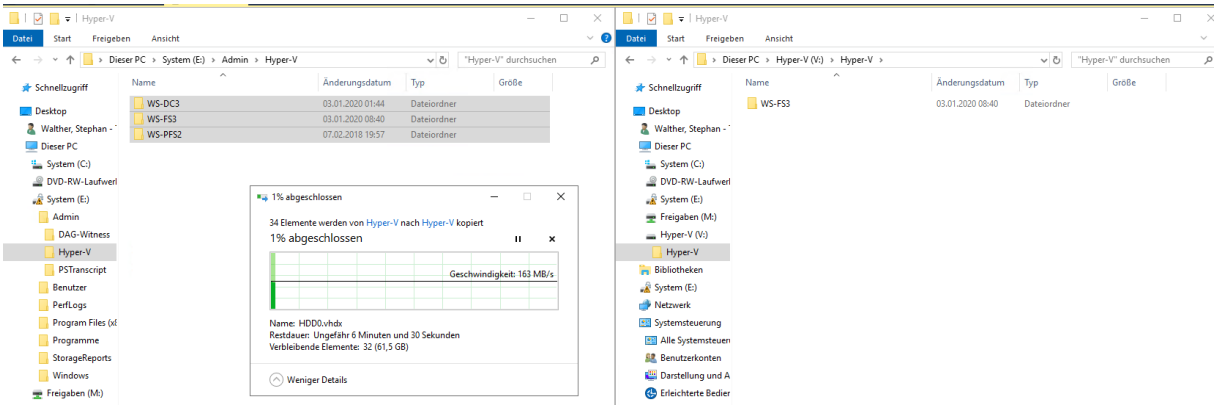
Der Assistent erstellt einen virtuellen Adapter, mit dem der Hyper-V-Host über den virtuellen Switch an den realen Adapter angeschlossen ist. Die IPv4-Konfiguration wird in der Regel übernommen:



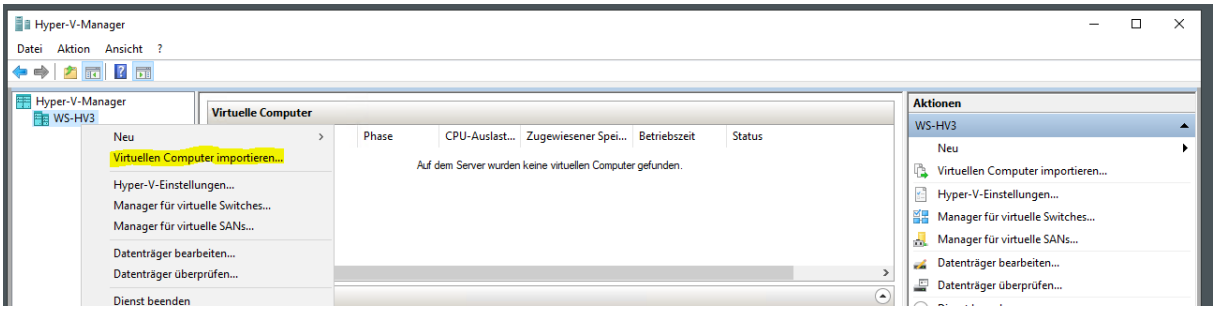
Jetzt kann ich die virtuellen Maschinen importieren. Dazu baue ich die alte Festplatte in eine Dockingstation ein und schließe diese mit USB3 an den Server an. Die alte Systempartition mit den VM-Dateien ist weiterhin mit Bitlocker verschlüsselt. Ich benötige aber keinen Recoverykey, da ich die Verschlüsselung zuvor angehalten hatte:

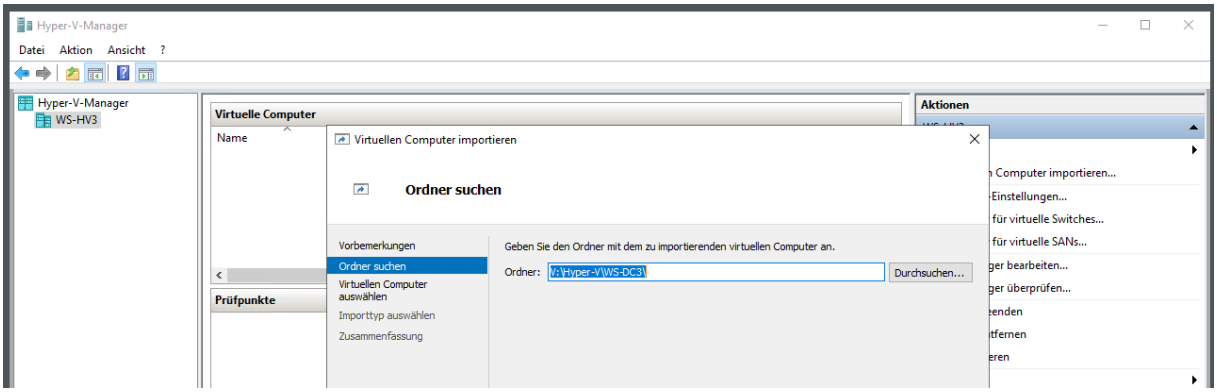


Die VMs sind jetzt nur Dateien und Ordner. Diese kopiere ich in einen neuen Ordner auf die neue SSD:

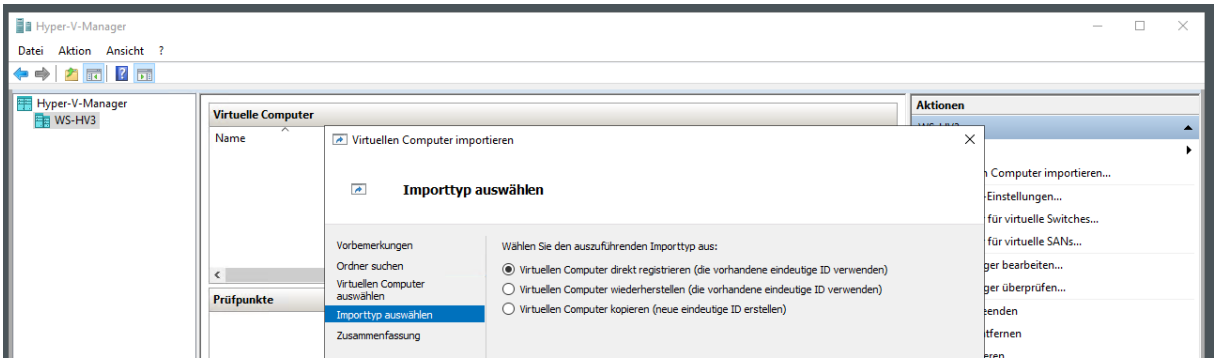


Danach importiere ich die 3 VMs mit dem Hyper-V-Manager:

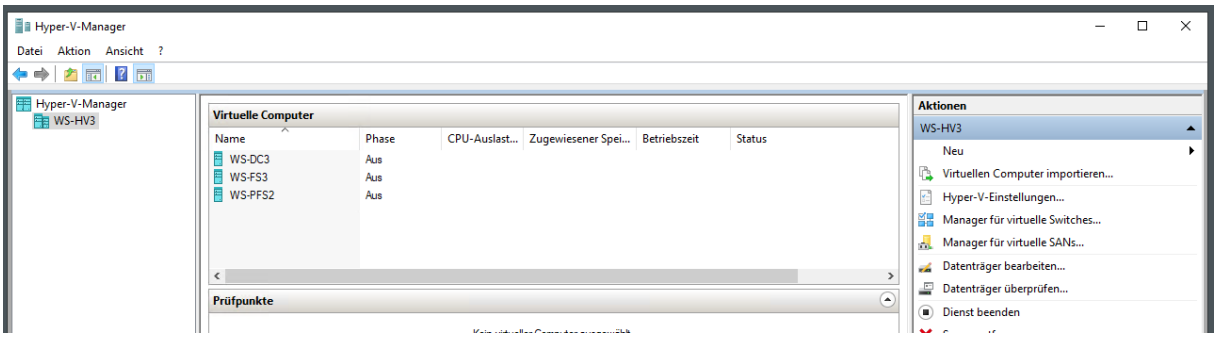




Die ID der VM behalte ich bei:

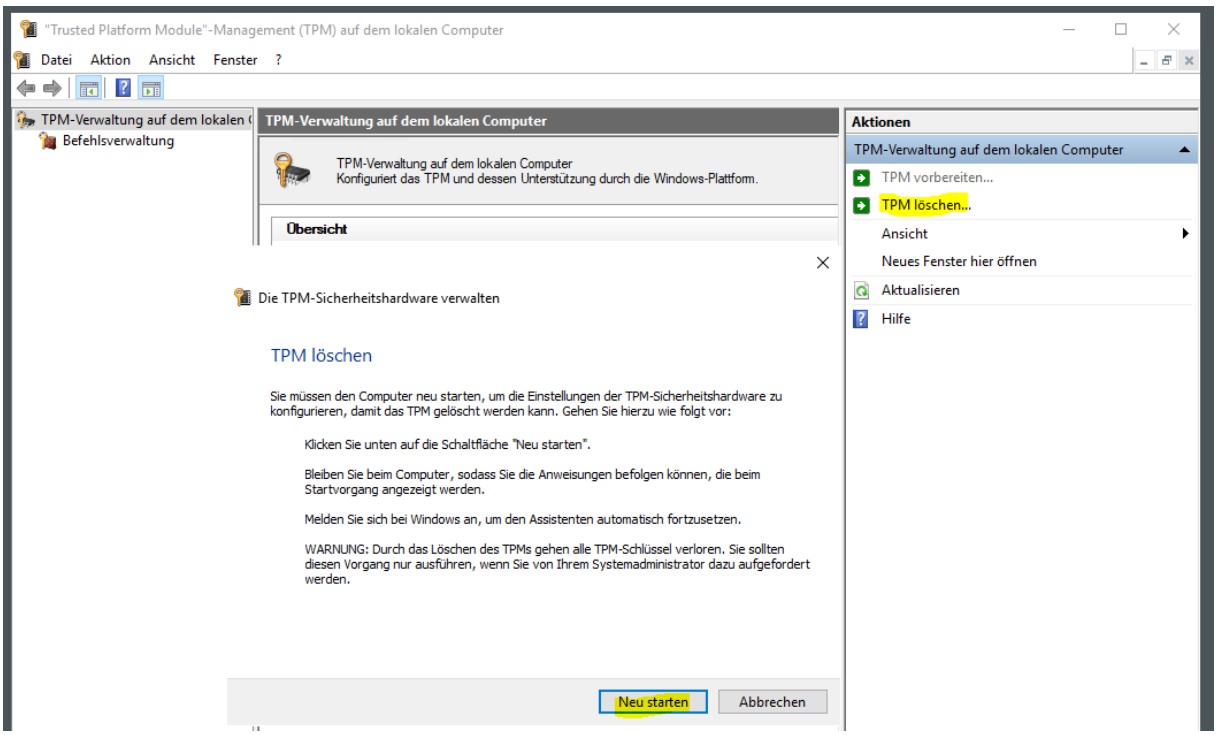


Nach wenigen Klicks sind die drei VMs registriert. Da lohnt sich ein PowerShell-Script nicht wirklich. Die VMs lasse ich noch ausgeschaltet, denn sie würden sich im falschen Netzwerksegment befinden. Vorher muss ich den Server doch wieder an seinen Platz im Außenstandort bringen:

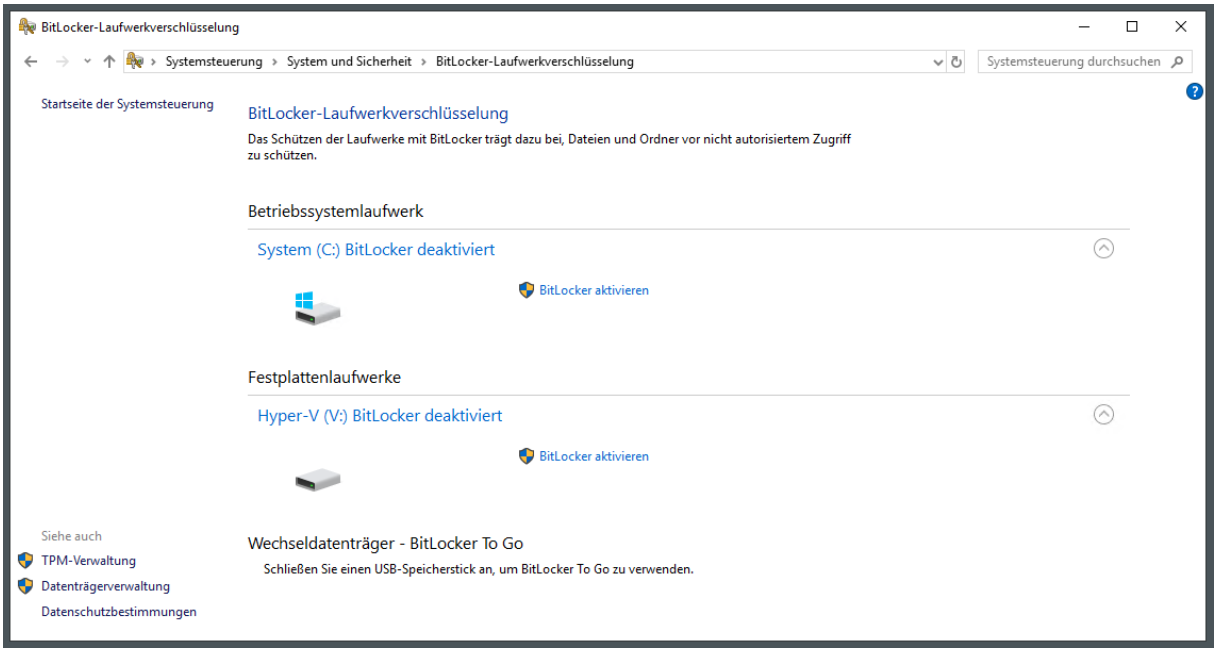


Absicherung mit Bitlocker

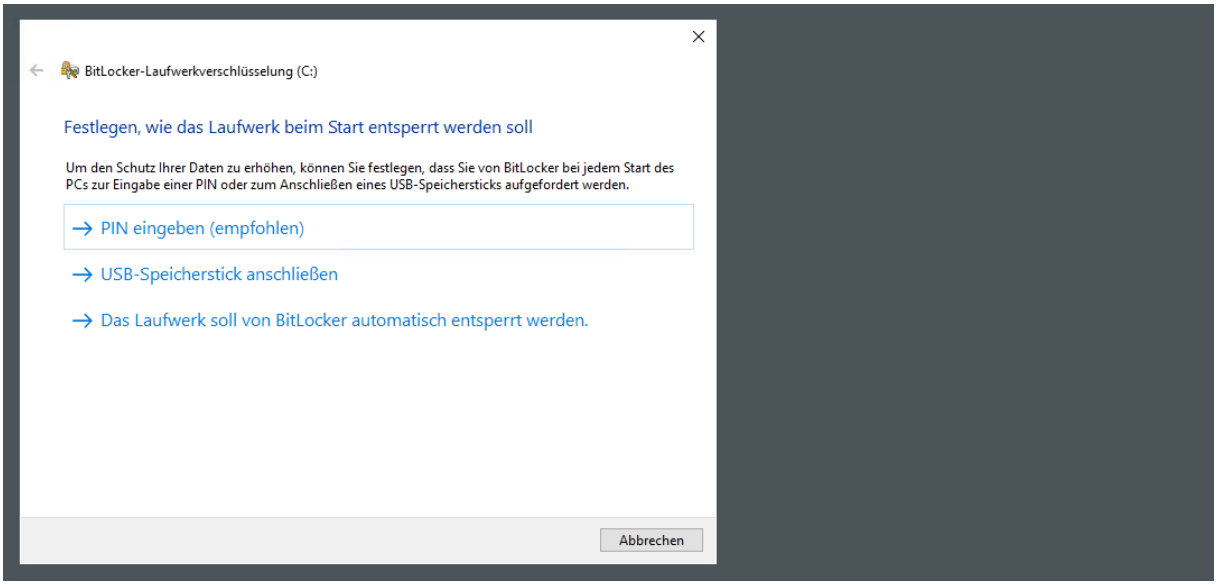
Aber die Absicherung kann ich im Hauptstandort vornehmen. Der TPM ist noch vom Windows Server 2016 in Verwendung. Daher lösche ich seinen Inhalt. Die Aktion wird bei einem Neustart durchgeführt:



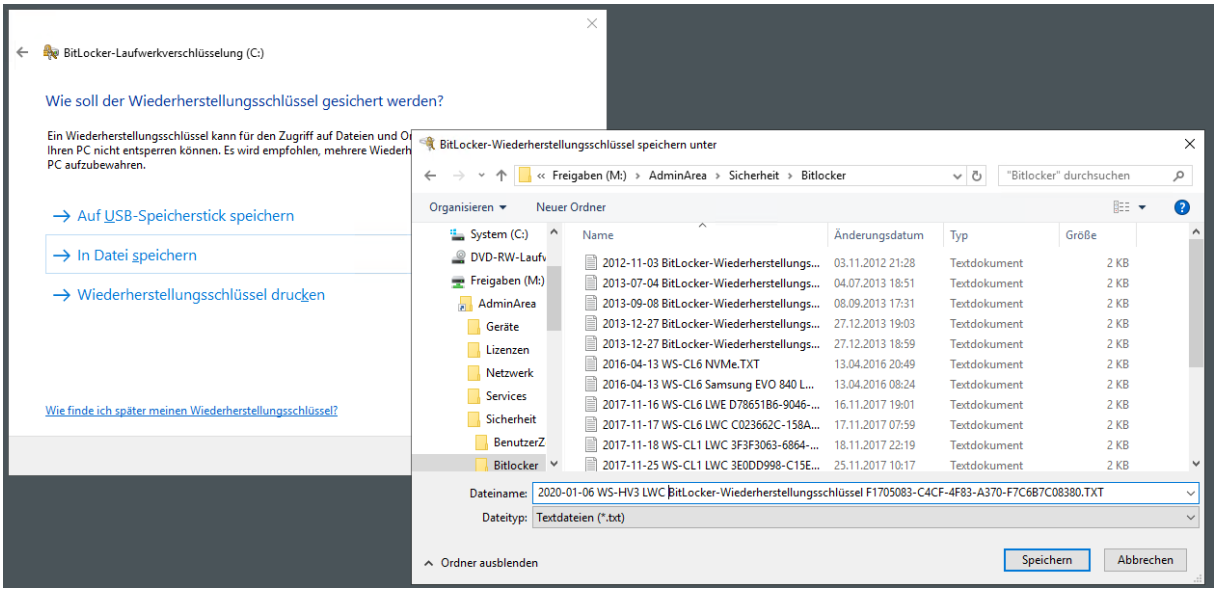
Nun kann ich die Verschlüsselung der Volumes auf der neuen SSD vornehmen:



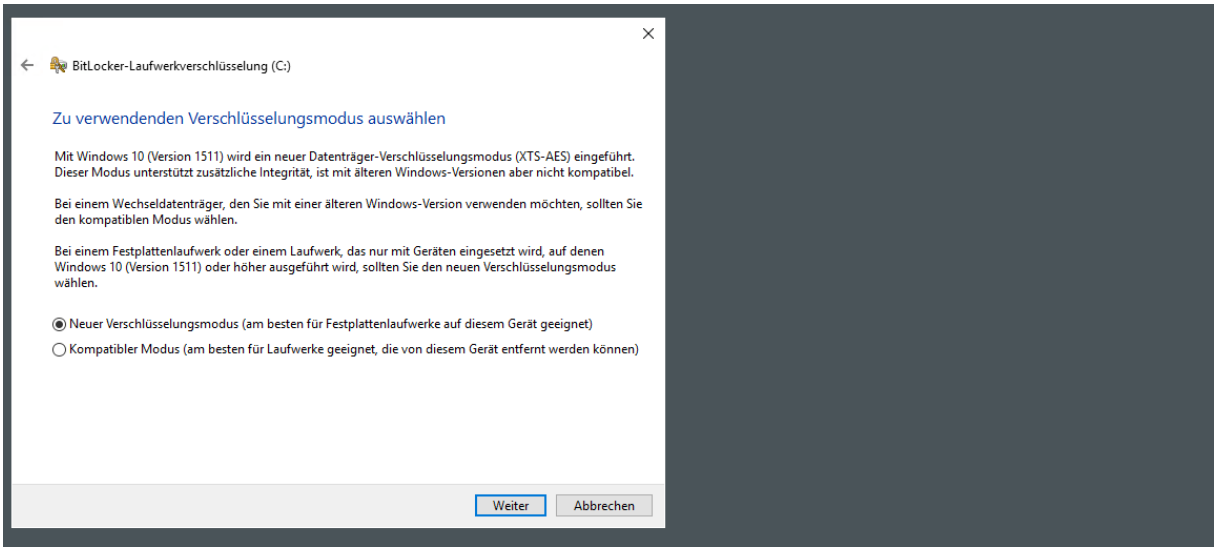
Damit ich den Server aus der Ferne neustarten kann bzw. damit er nach Updates neustartet, wähle ich die Option mit der automatischen Entsperrung. Das klingt vielleicht im ersten Moment unsicher, ist es aber nicht. Der TPM prüft beim Start mit der UEFI, ob sich relevante Bauteile der Serverhardwar verändert haben. Ebenso werden Firmwareveränderungen erkannt. Sollte es eine Manipulation geben, dann wird der TPM den Entschlüsselungsschlüssel nicht freigeben und die SSD bleibt verschlüsselt. Ist alles integer, dann wird die Platte freigegeben. Ab diesem Moment übernimmt das Betriebssystem die Absicherung. Nur bei Fehlern im TPM, der UEFI-Firmware oder der Verschlüsselung selber könnte zu einem unberechtigten Zugriff führen. Natürlich kann auch das laufende Betriebssystem über Lücken verfügen, welche den Zugriff gestatten. Aber diese würde ich nicht mit einem Start-PIN verhindern. Also wähle ich die Automatik:



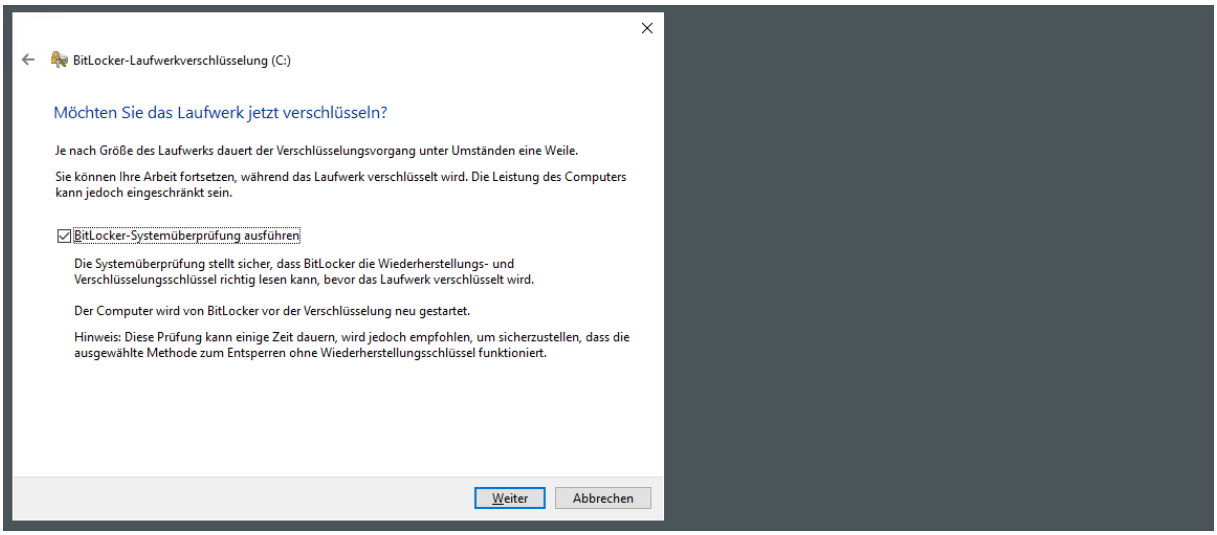
Für die Wiederherstellungsschlüssel habe ich einen geschützten Ordner in meinem AdminShare. Zusätzlich wird der Schlüssel auch im Active Directory abgelegt. Hier sichere ich mich doppelt ab:



Die modernere Variante passt auf mein Szenario:

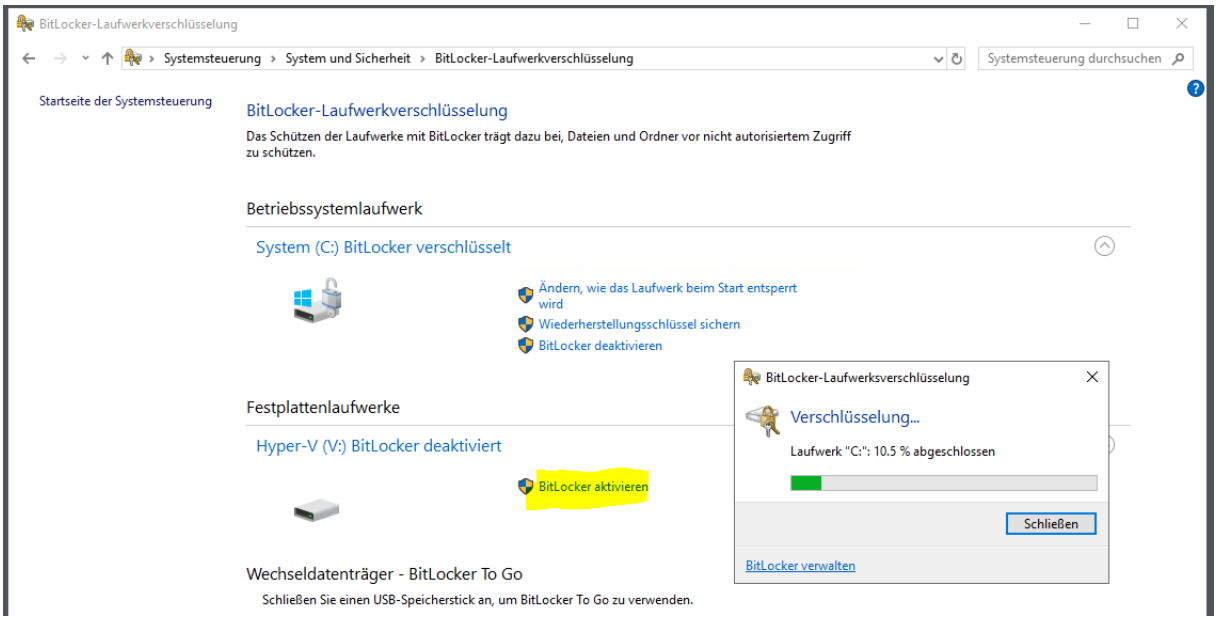


Die Systemprüfung lasse ich mit durchlaufen:

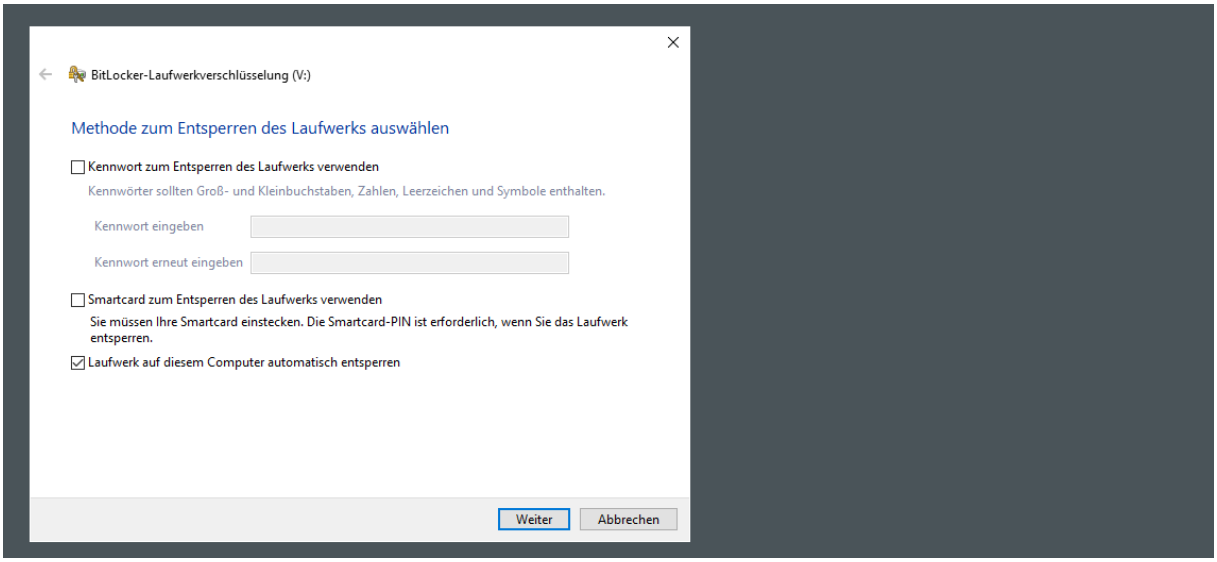


Nach Abschluss des Assistenten startet die Verschlüsselung.

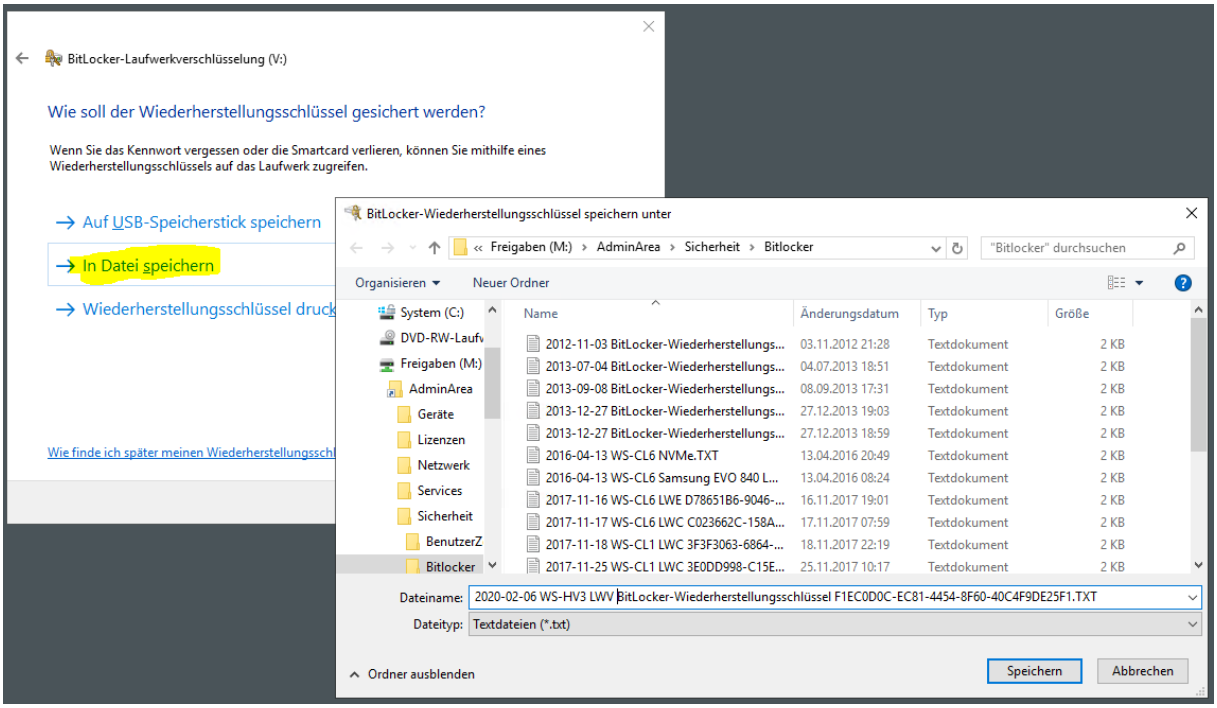
Weiter geht es mit dem Volume mit den virtuellen Maschinen. Auch hier starte ich die Verschlüsselung:



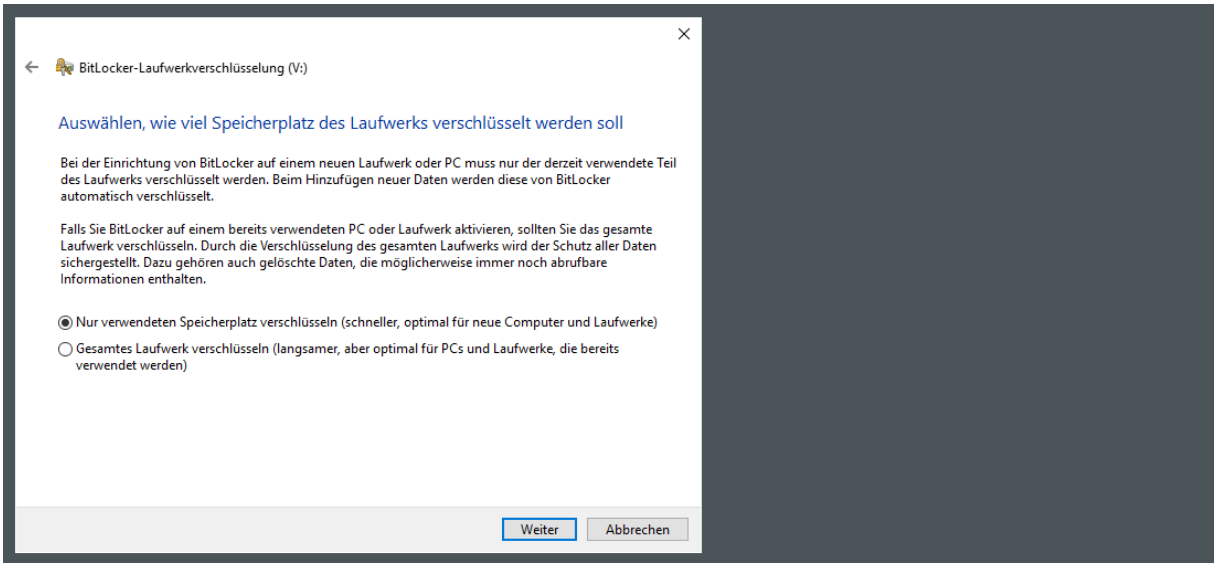
Dieses Volume ist nicht das Startvolume. Es soll aber mit diesem automatisch entsperrt werden:



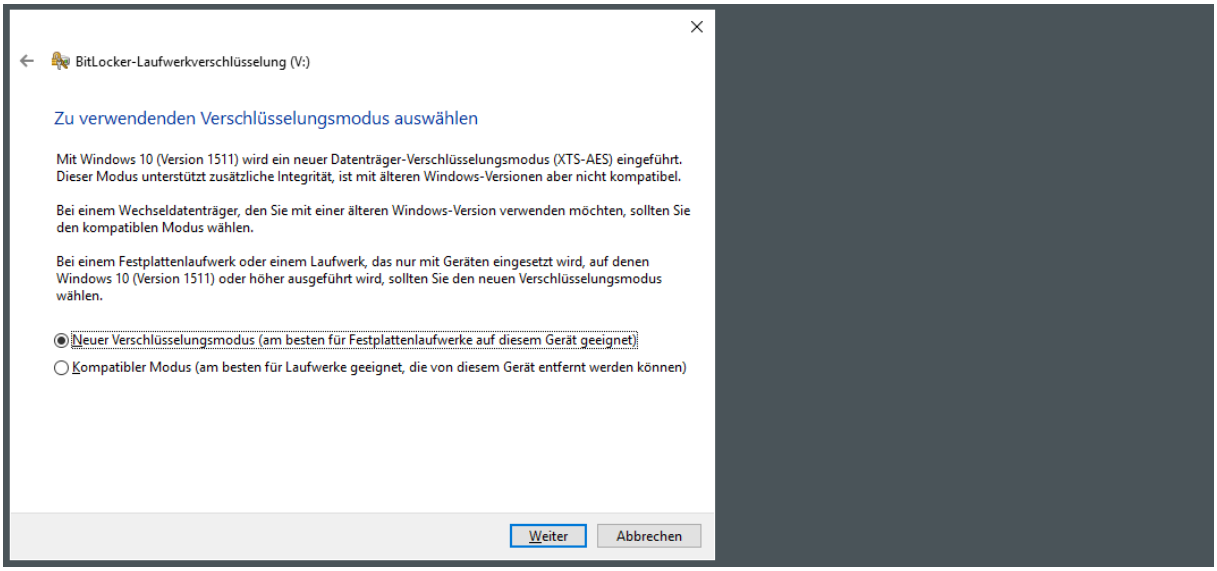
Auch dieser Wiederherstellungsschlüssel landet im AdminShare:



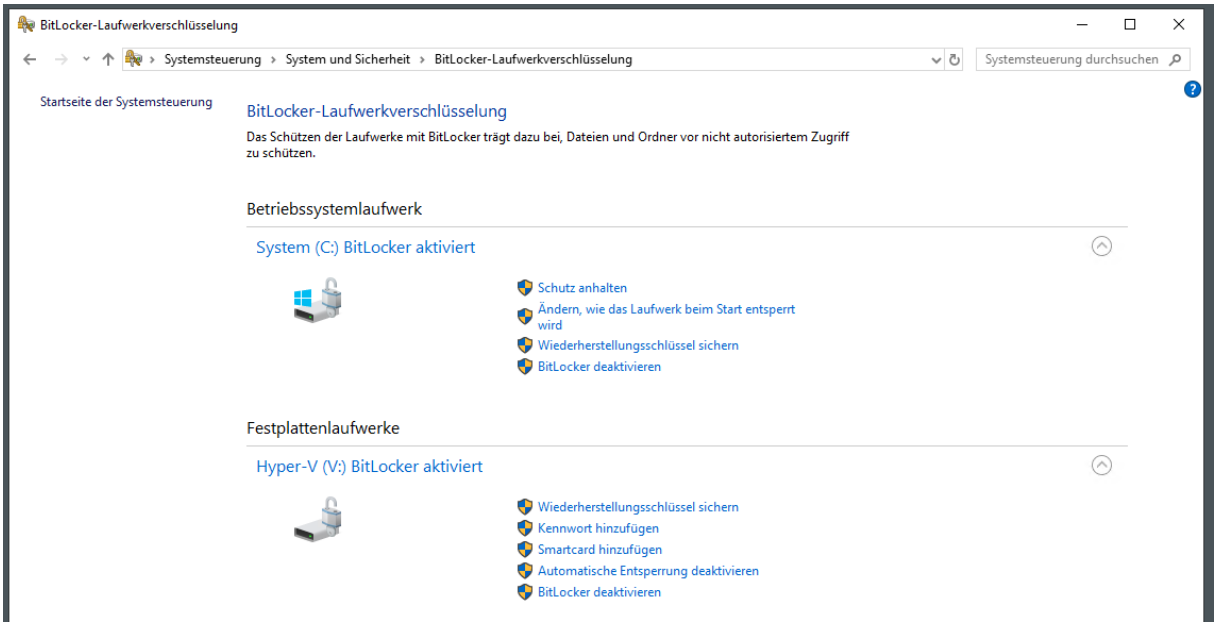
Das Volume ist neu und kaum belegt. Daher wähle ich die schnelle Variante. Bei normalen Daten könnten ggf. einzelne Fragmente ausgelesen werden. Bei mir liegen aber VHDX Contailerfiles auf dem Volume. Deren Fragmente sind durch die logische, interne Struktur so gut wie ausgeschlossen lesbar:



Der Rest des Assistenten ist Standard:



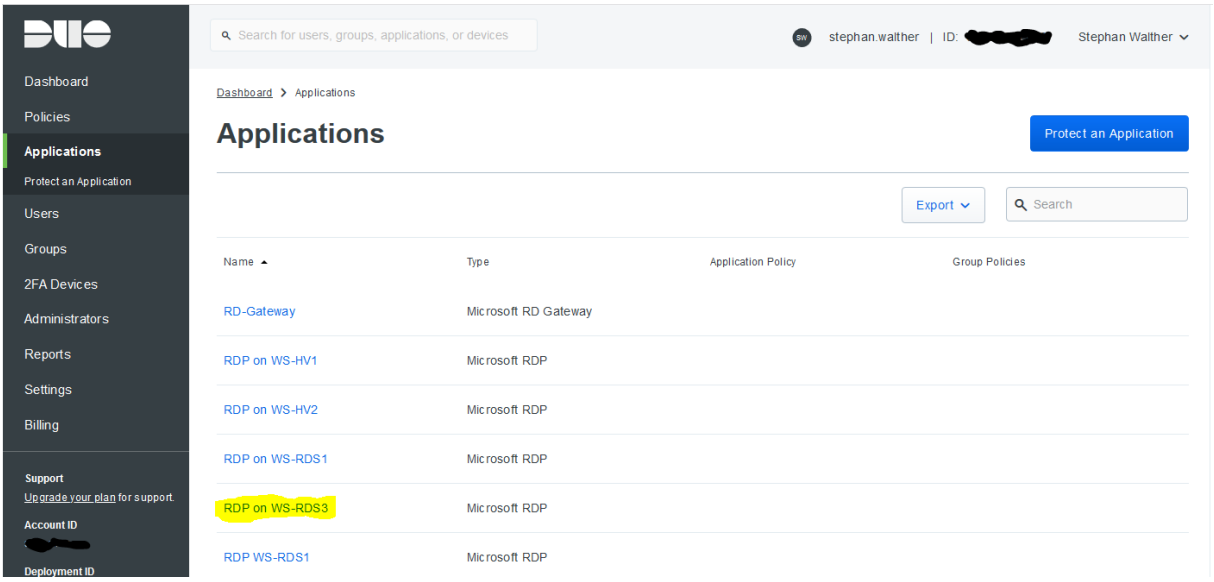
Jetzt ist der Inhalt des Datenträgers offline geschützt:



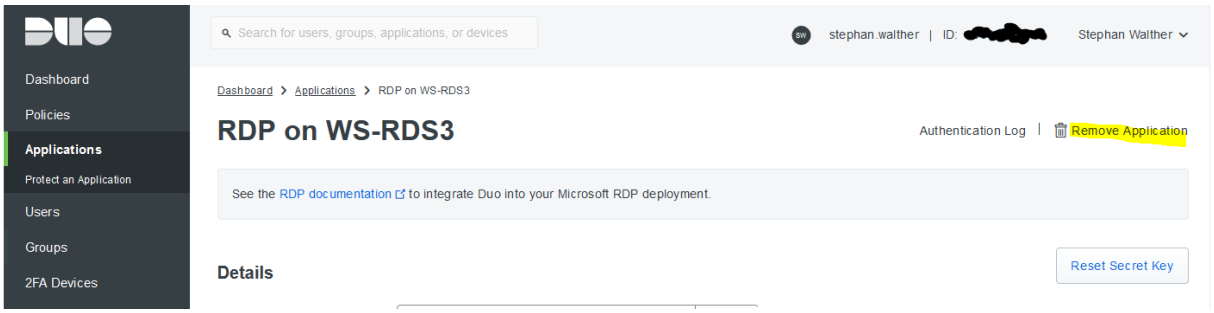
Absicherung mit DUO-2FA

Aber auch den Online-Zugriff auf das laufende Betriebssystem möchte ich absichern. Dazu implementiere ich für die Anmeldung lokal und via RDP eine Zweifaktor-Authentifizierung. Als Anbieter wählte ich vor einiger Zeit DUO – für kleine Strukturen ist die Lösung kostenfrei.

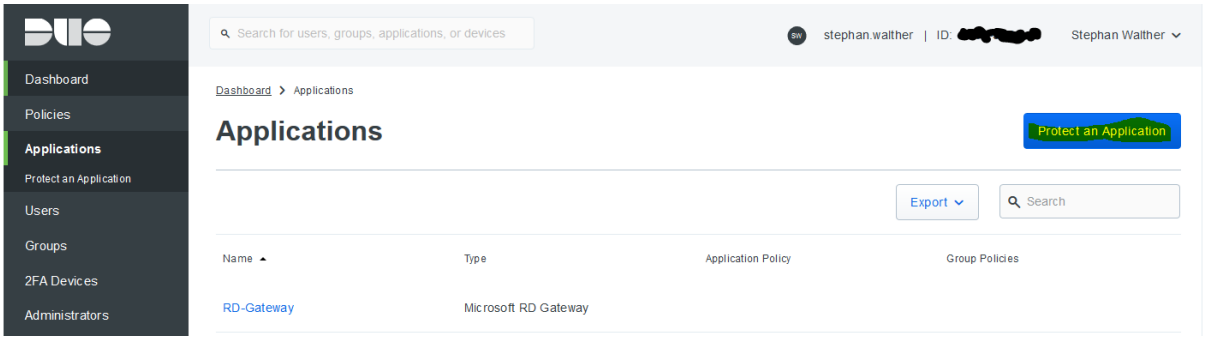
Im DUO-Onlineportal existiert noch der alte Anmeldeschutz des Servers WS-RDS3. Jede Absicherung im DUO wird als Application bezeichnet:



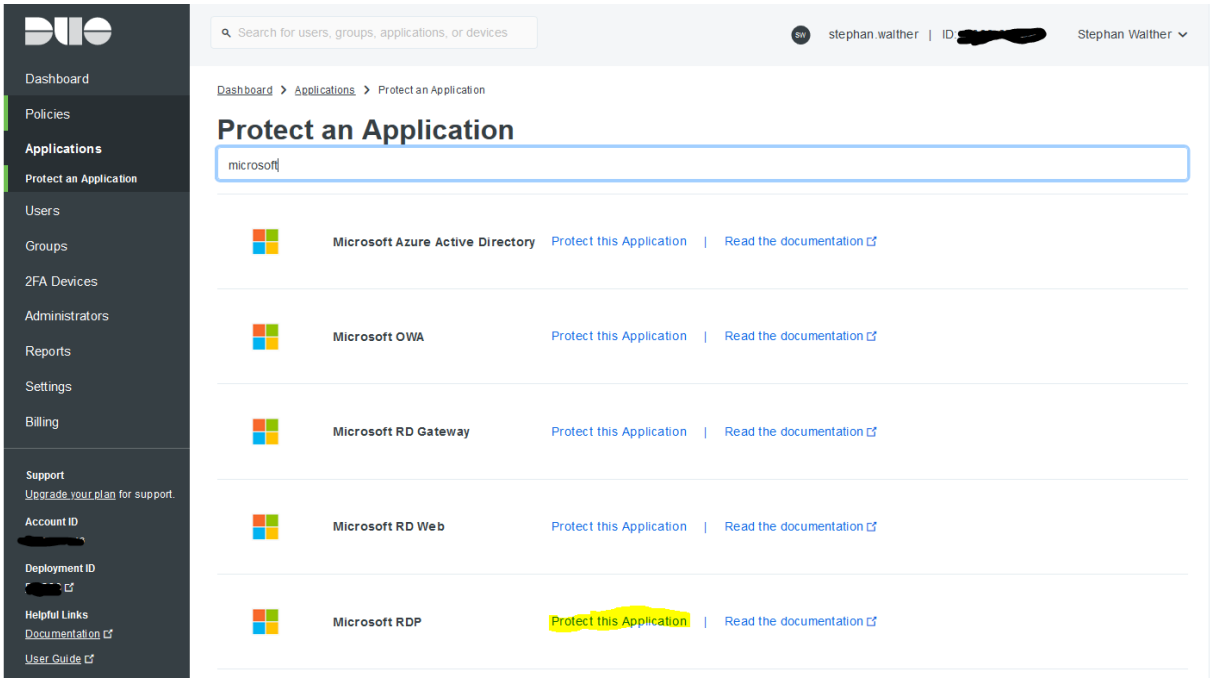
Nach der Auswahl dieser Application kann ich sie löschen:



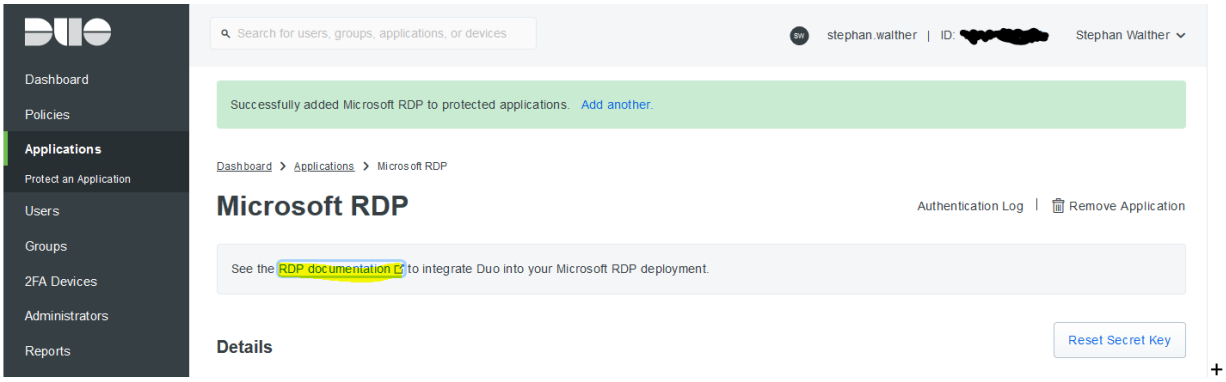
Anschließend erstelle ich eine neue Application:



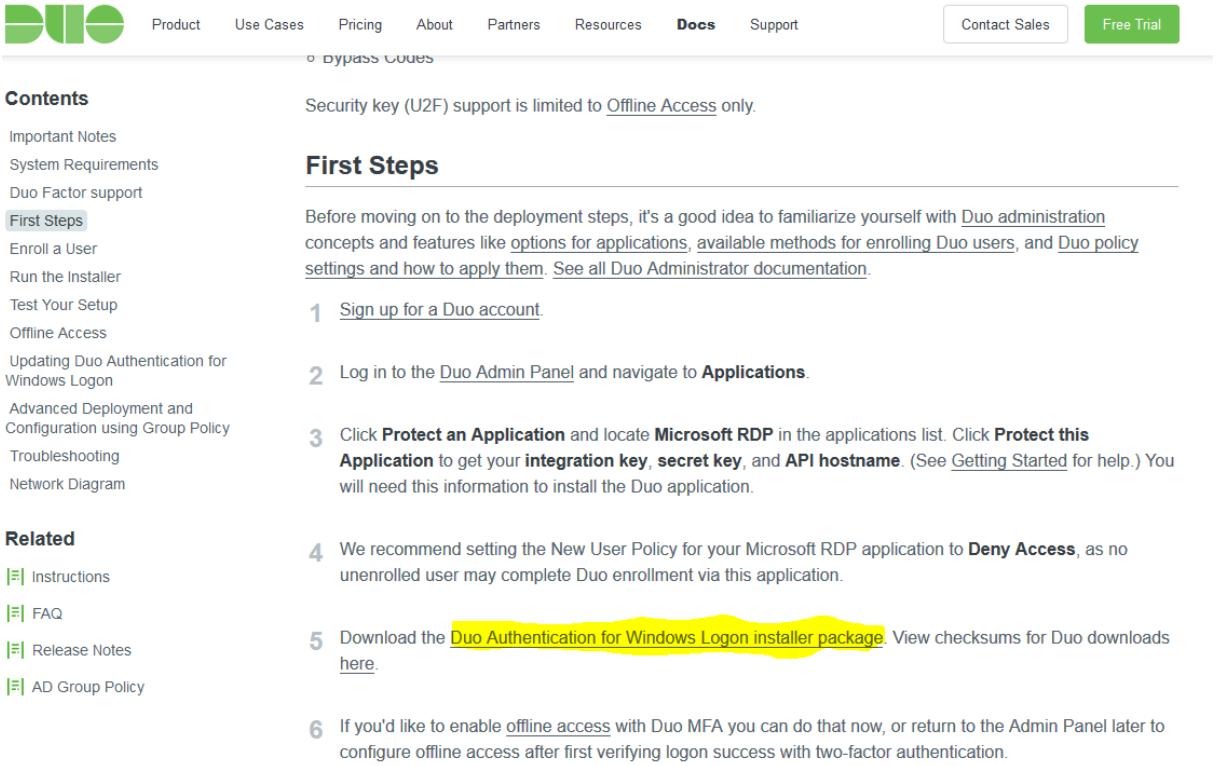
Im nächsten Schritt wähle ich einen passenden Typ aus. Die RDP-Application kann Rmeotezugriffe und lokale Anmeldungen absichern:



Details zur Einrichtung gibt es hinter diesem Link. Dort steht auch der Download der msi-Installationsdatei für den Zielserver zur Verfügung:



Ich lade die aktuelle msi-Datei herunter:

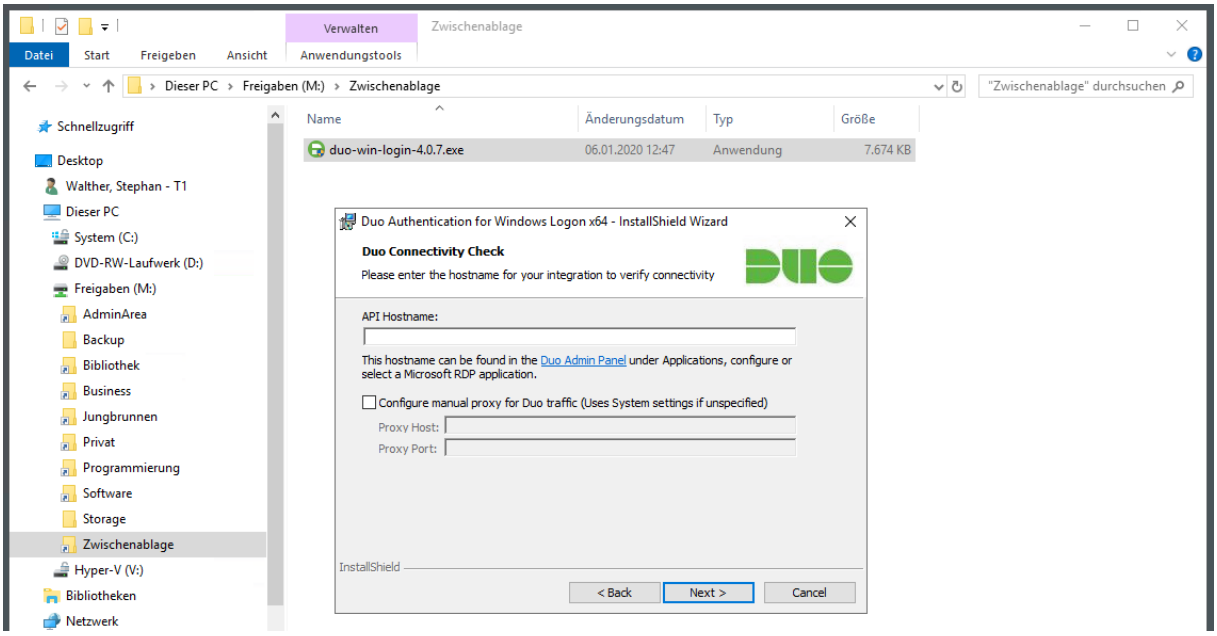


Das Setup starte ich auf dem neuen WS-HV3. Hier werden einige Fragen gestellt, für die ein wenig Hintergrundwissen sinnvoll ist. Der Anmeldeprozess wird durch die Installation verändert:

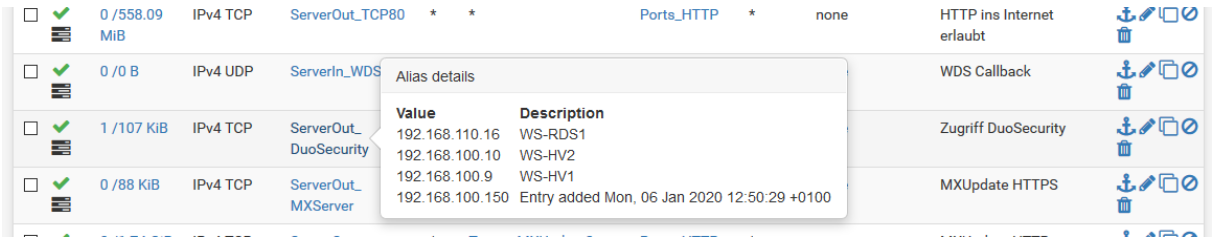
- Bei einer Anmeldung am System wird zuerst ganz regulär die Kombination aus Benutzername und Passwort geprüft. Ist diese falsch, dann gibt es die bekannte Fehlermeldung.
- Ist die Anmeldung aber korrekt, dann wird das Plugin von DUO gestartet. Es sendet an einen API-Hostname (FQDN eines Servers im Internet) eine Anmeldeanfrage für den aktuellen Benutzernamen.
- Die Cloudkomponente prüft, ob der Account bekannt ist und ob für ihn ein zweiter Faktor registriert ist.
- Dann wird eine Push-Notification an die Smartphone-App des Benutzers gesendet.
- Bestätigt der Benutzer die Anmeldung am Smartphone, dann sendet die App an die Cloudkomponente das OK zurück.
- Diese wiederum kommuniziert mit der noch offenen Verbindung zum DUO-Plugin auf dem Server und reicht das OK durch.
- Erst dann wird die Anmeldung fortgesetzt.

Jede Unterbrechung, Fehlkonfiguration oder Verzögerung (es gibt Timeouts) wird dazu führen, dass die Anmeldung am Server fehlschlägt.

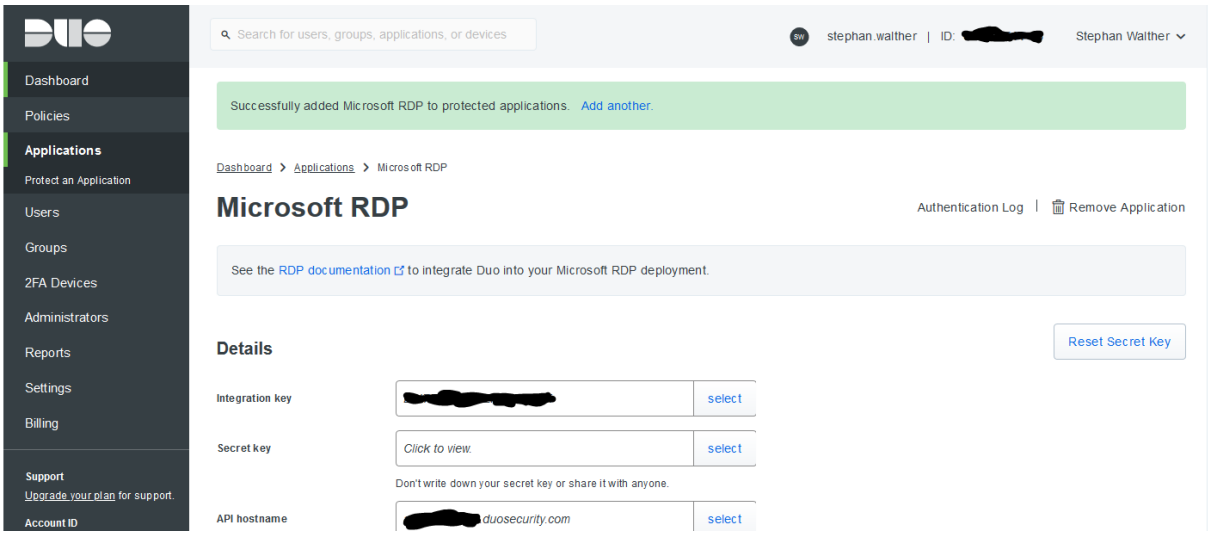
Jetzt ist der Dialog des Setup auf dem Server etwas sprechender, oder?



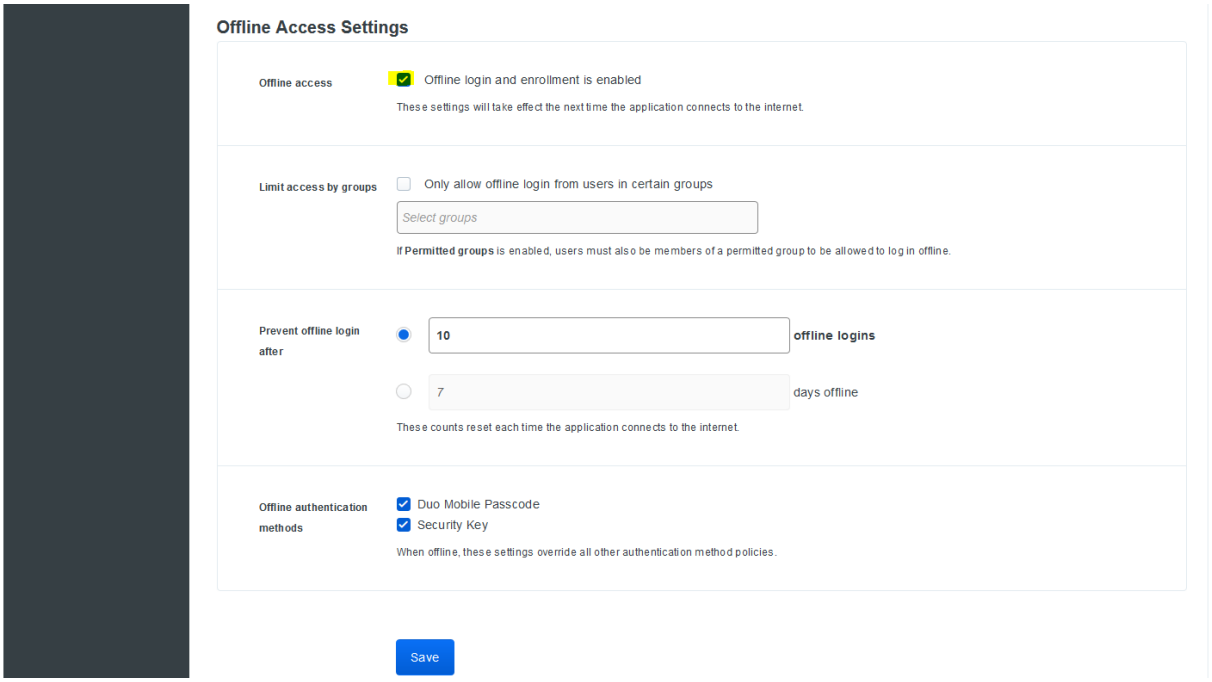
Für die Anmeldung ist eine ausgehende Internetverbindung erforderlich. In meinem Netzwerk ist in der zentralen Firewall erst einmal nichts erlaubt. Das gilt auch für die DUO-Authentication. Ich habe für diese eine eigene Regel mit einer Gruppe erstellt. Hier trage ich temporär die IPv4 des Servers ein. Die Adresse werde ich nach dem Umzug des Servers in seinen Standort wieder entfernen:



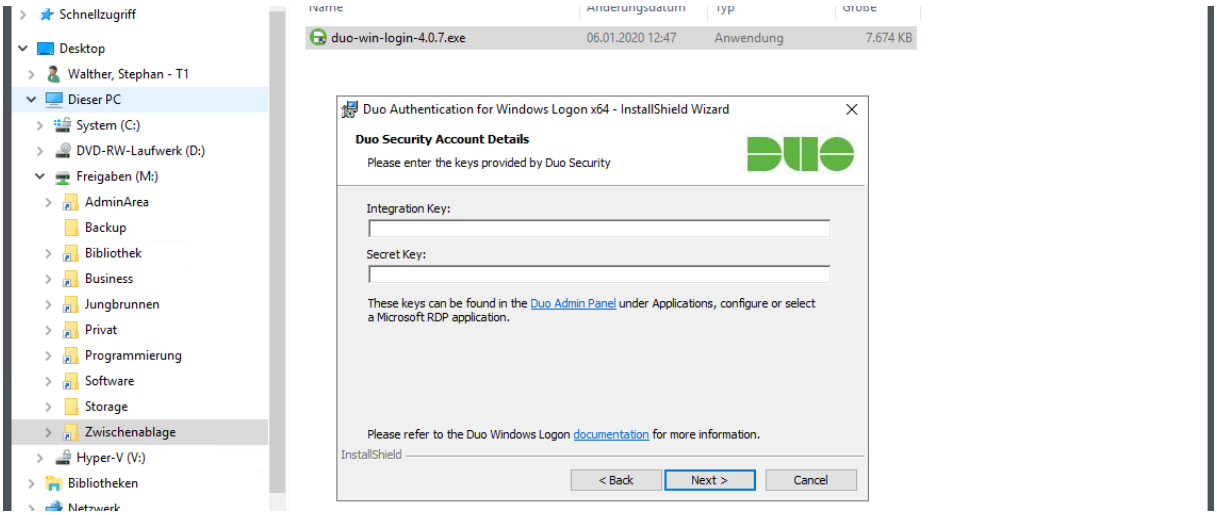
Der API-Hostname bleibt für die Application immer gleich. Eine Firewall-Ausnahme ist also sehr einfach konfigurierbar. Ich lese den Hostname aus dem DUO-AdminPortal aus. Hier gibt es auch den Integration Key, der meine Installation eindeutig dieser Application zuordnet. Und auch das Plugin muss sich am API-Hostname anmelden. Dazu wird der Secret-Key verwendet:



Gleichzeitig mit dem Auslesen der 3 Werte aktiviere ich noch den Offline-Zugriff. Mit diesem kann ich das Plugin bei der Anmeldung mit einem Offline-Token bestätigen. Das ist bei Netzwerkproblemen sehr hilfreich:

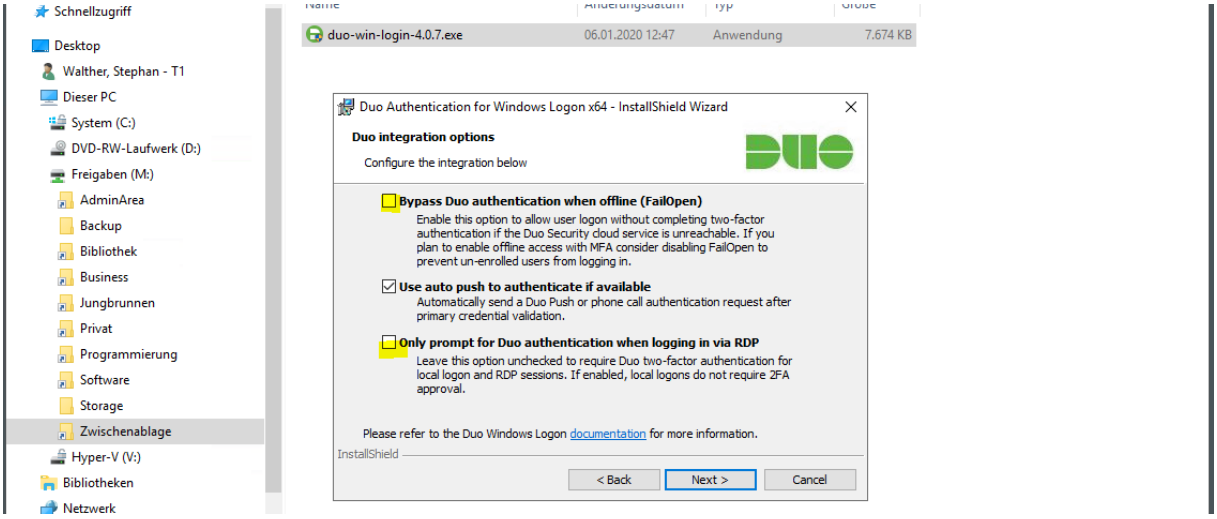


Die im Portal ausgelesenen Werte gebe ich in das Setup am Server ein:

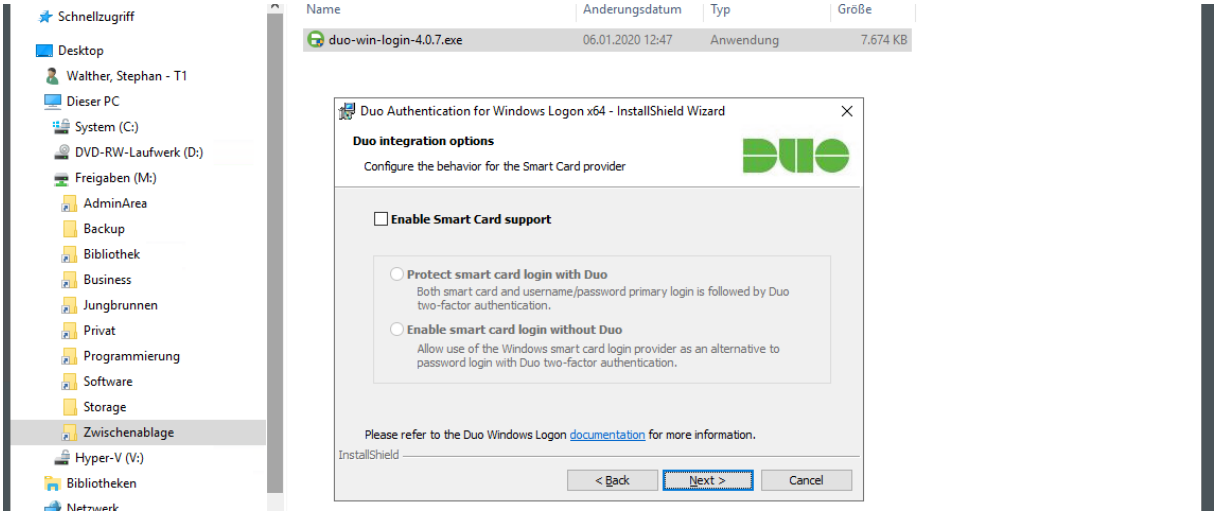


Ich erlaube keinen Bypass. Dabei würde nach Ablauf eines Timeouts das Plugin auch ohne Onlinebestätigung die Anmeldung des Benutzers fortsetzen. Mit einem lokalen Zugriff auf den Server könnte ich also den Netzwerkstecker bei der Anmeldung ziehen und der zweite Faktor wäre umgangen – natürlich wäre Benutzername und Passwort immer noch notwendig. Aber auch mit einem RDP-Connect wäre es denkbar, dass ich den API-Hostname mit einem Denial-of-Service kurzfristig lahmlege. Dafür müsste ich wissen, welchen FQDN dieser geheime Server hat. Aber er wird ja mittels DNS aufgelöst – im Klartext. Kontrolliere ich als Angreifer einige Netzwerkkomponenten, dann wäre das Denkbar. Nein, das ist keine Option. Abgesehen davon verwende ich den Offline-Zugriff über die Offline-Tokens.

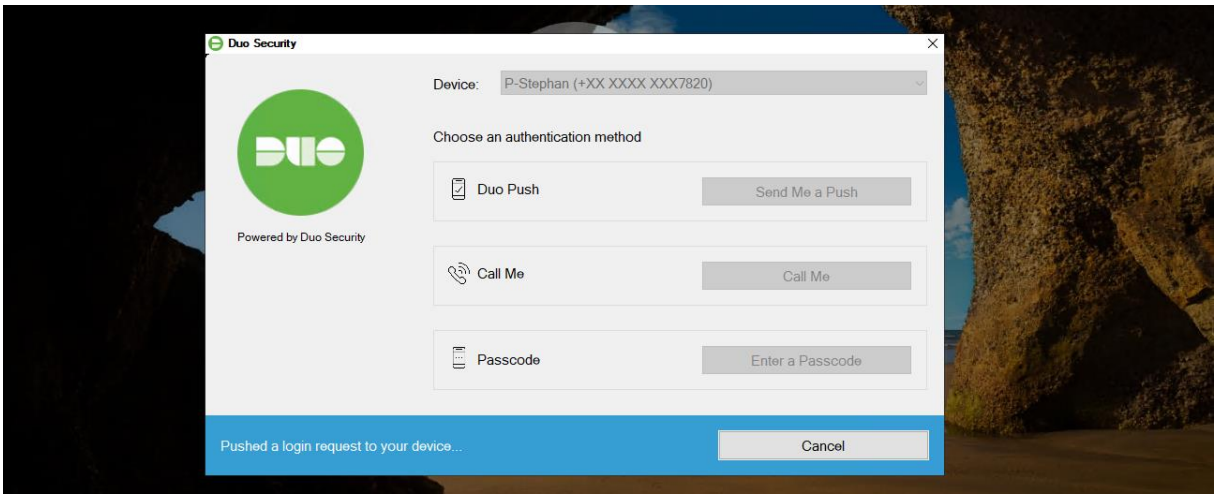
Die zweite Option erlaubt es dem Plugin, auch lokale Anmeldungen abzusichern:



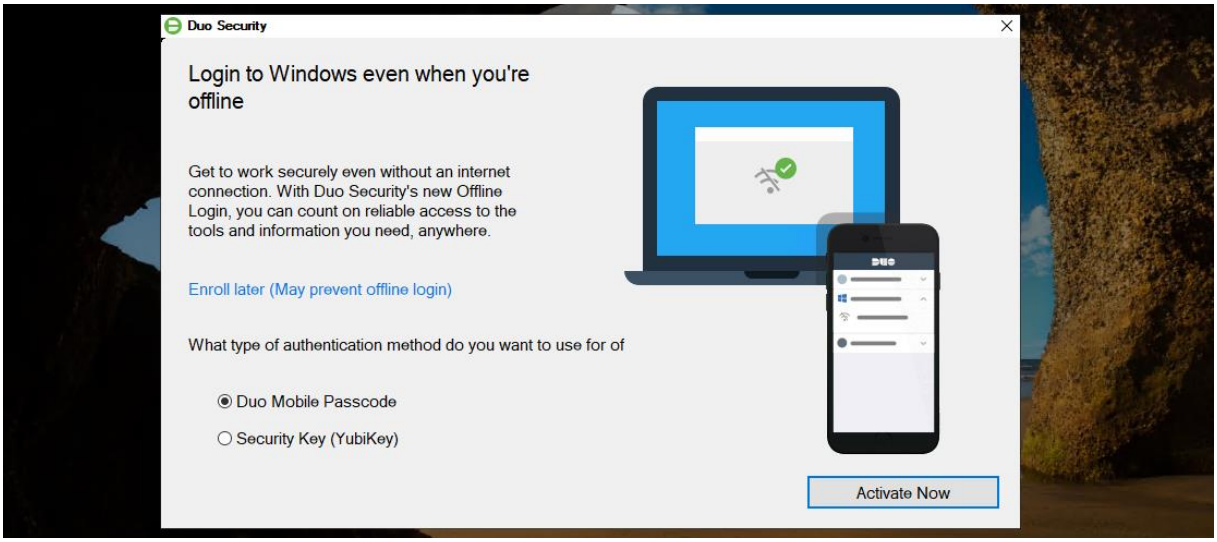
Den Smartcard-Support habe ich beim ersten Setup vergessen. Daher funktioniert später ein anderes, abgesichertes Anmeldeverfahren nicht. Ich lasse die Bilder aber mal im Protokoll drin. Auch aus Fehlern kann man lernen:



Das war auch schon alles. Mein administrativer Account ist bereits bei DUO mit meinem Smartphone assoziiert. Daher kann ich direkt in den Test einsteigen. Ich melde mich als Admin erneut am Server an. Benutzername und Passwort sind korrekt, daher wird das Plugin gestartet:

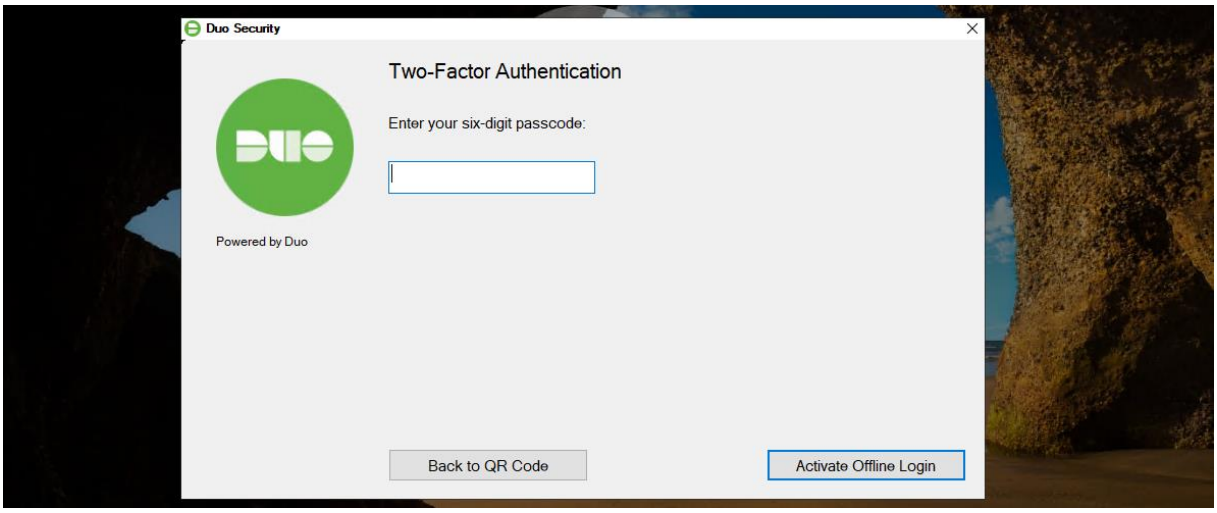


Ich bestätige am Smartphone die Anmeldeanfrage. Damit bin ich für die Anmeldung autorisiert. Mit dem Connect zum Cloudserver hat das Plugin aber den gewünschten Offline-Zugriff erkannt und startet die Einrichtung:



Ich nutze dafür die gleiche DUO-Mobileapp wie für die Online-Anfragen. Die Einrichtung ist denkbar einfach. Mit der App scanne ich den gezeigten QR-Code:

Anschließend zeigt mir die App den ersten Zahlencode. Diesen gebe ich zur Bestätigung ein:



Das war es auch schon. Diese Hürde ist hoch. Angriffe auf das System sollten nahezu unmöglich sein. Und das mit relativ viel Komfort!

Absicherung mit Notfall-Account und vSmartcard

Die nächste Absicherung ist ein spezielles Notfall-Szenario in meiner Infrastruktur. Folgende Punkte erfordern diese Lösung:

- Meine Infrastruktur hat ausschließlich virtuelle Domain Controller. Diese laufen auf unterschiedlichen Hyper-V-Hosts.
- Meine administrativen Accounts haben keine Rechte auf den Hyper-V-Hosts. Die Rechte vergebe ich nur temporär mit einer Privileged Access Management Lösung. Dieses PAM benötigt Zugriff auf die Domain Controller (denn da werden die Gruppenmitgliedschaften verändert).
- Meine administrativen Accounts sind Mitglied der Gruppe „Protected Users“. Daher werden erfolgreiche Anmeldungen nicht zwischengespeichert. Für jede Anmeldung ist der Kontakt zu einem Domain Controller erforderlich.
- Jeder Server hat bei mir genau einen lokalen, administrativen Account. Deren Passworte werden durch LAPS (Local Administrator Password Solution) regelmäßig verändert. Die aktuellen Passworte kann ich in den Domain Controllern auslesen.

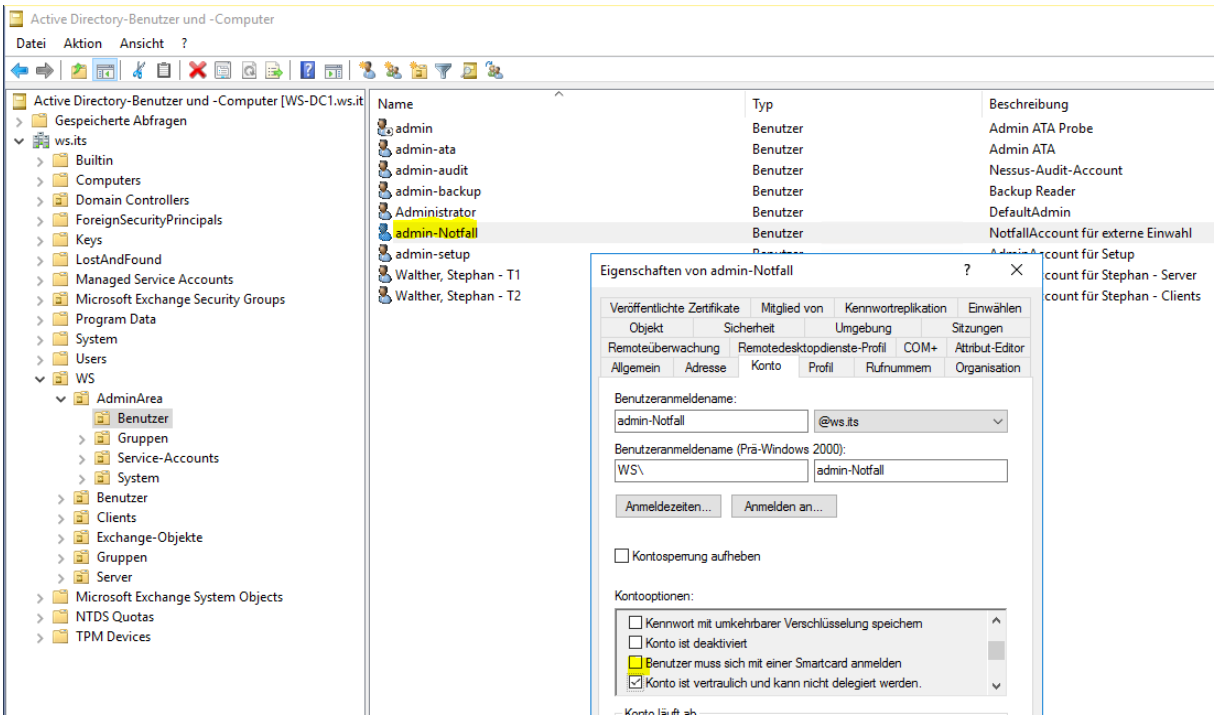
Und mit diesen Konfigurationen wird das Problem erkennbar: Ich benötige für jede Anmeldung zwingend eine Verbindung zu einem Domain Controller! Das ist ja eigentlich auch wünschenswert. Es existieren aber eine Reihe von Problemszenarien:

- Nach einem Stromausfall, der meine USV-Kapazität übersteigt, könnten die Hyper-V-Hosts Probleme beim Hochfahren der virtuellen Domain Controller haben. Und ich kann mich nicht anmelden, um zu helfen.
- Fehler im Active Directory könnten zum gleichen Problem führen.

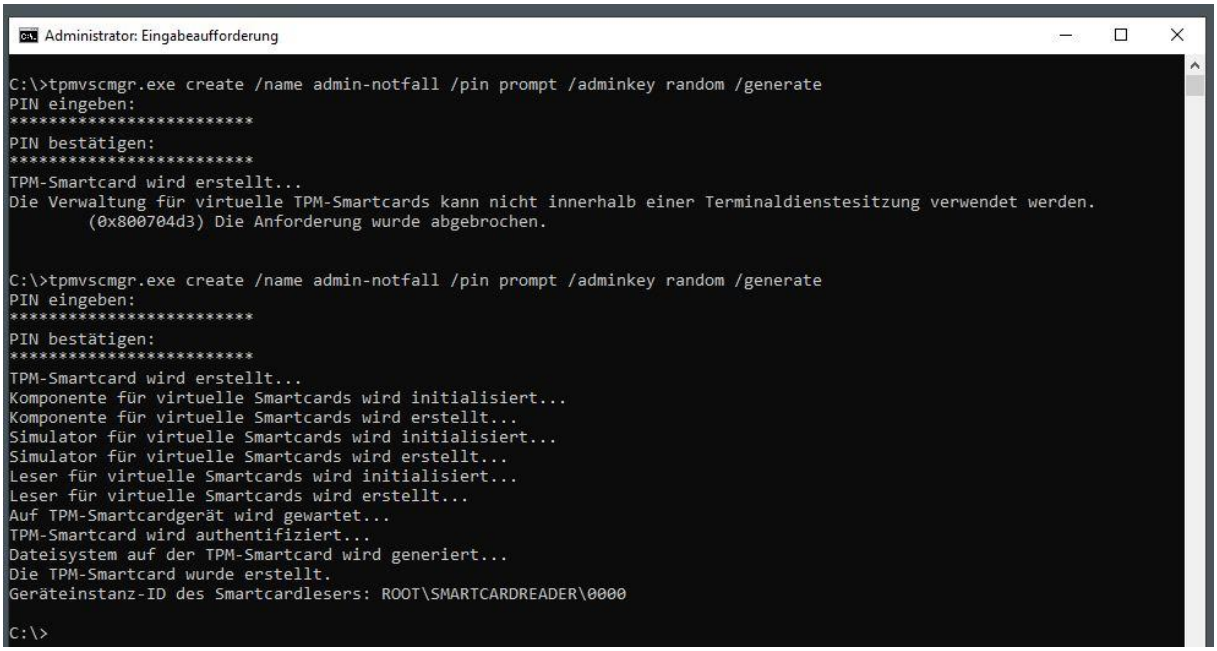
Beide Fälle habe ich bereits durchlebt. Bisher konnte ich immer über Umwege auf einen Domain Controller im anderen Standort zugreifen und so das Passwort des lokalen Admins auslesen. Aber da war auch immer viel Glück im Spiel. Daher habe ich einen speziellen AdminAccount erstellt: den admin-notfall. Dieser Account ist stark eingeschränkt:

- Er kann sich ausschließlich auf den Hyper-V-Hosts anmelden
- Auf den Hyper-V-Hosts hat er keine administrativen Systemrechte, kann aber den Service Hyper-V administrieren.
- Er benötigt für die Anmeldung eine Smartcard.
- Er ist aber kein Mitglied der Gruppe „Protected User“. Daher kann jeder Hyper-V-Host seine Anmeldung zwischenspeichern.

Mit diesem Account kann ich mich auf einem Hyper-V-Host anmelden und den virtuellen Maschinen (also auch den Domain Controllern) Starthilfe geben. Ich verwende virtuelle Smartcards. Diese muss ich auf jedem Computer neu erstellen. Dazu muss ich mich mit dem Account einmal ohne Smartcard anmelden. Ich entferne also die Option:

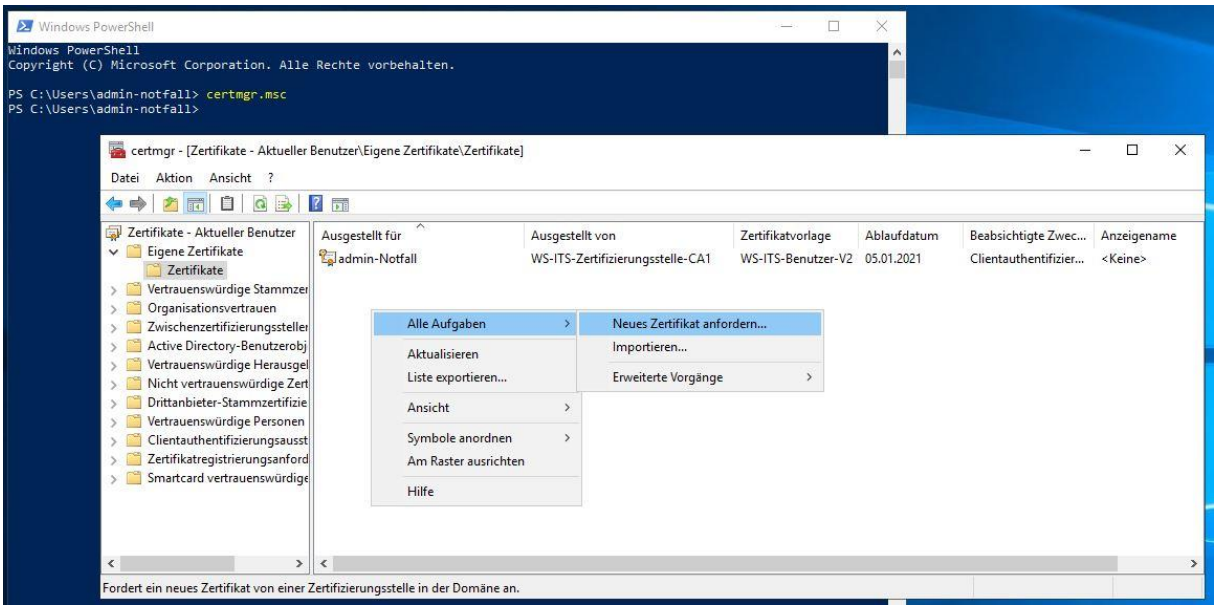


Mit meinem administrativen Account bereite ich im TPM-Chip eine neue, virtuelle Smartcard vor. Das geht nicht in einer RDP-Sitzung. Ich wechsle also in eine lokale Anmeldung:

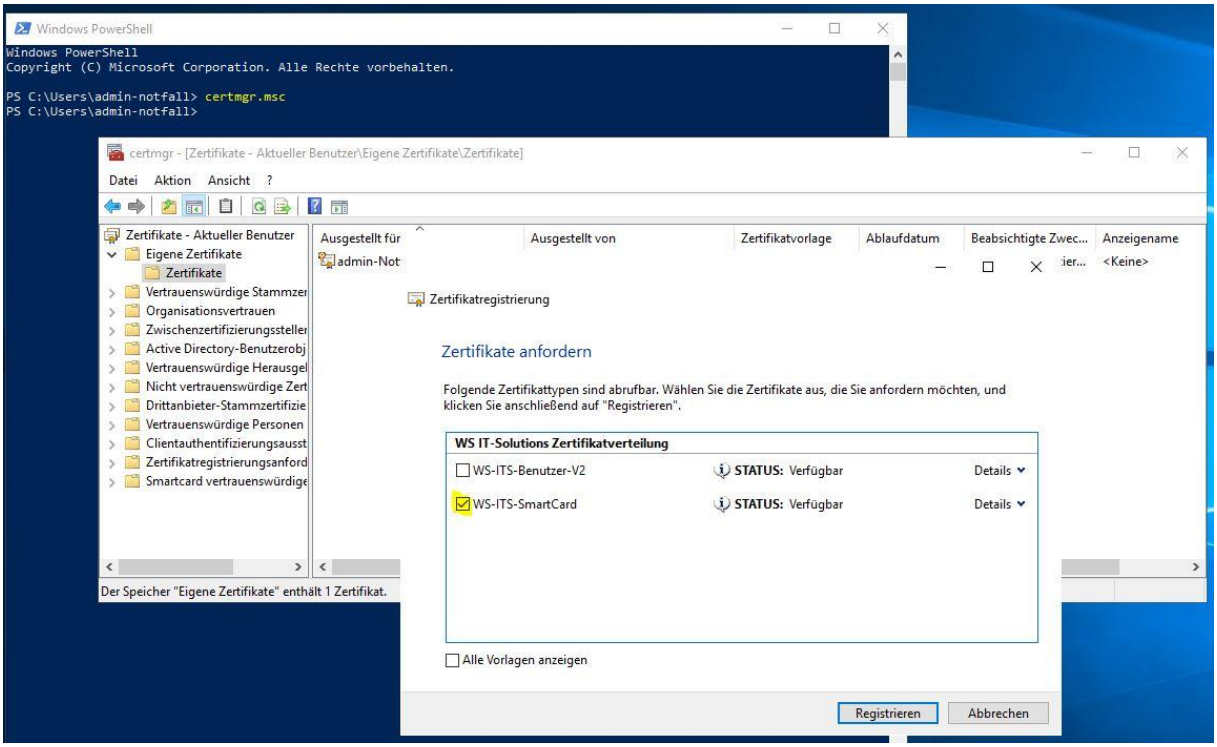


Dann melde ich mich als admin-notfall am Server lokal an. Die Berechtigung dazu habe ich bereits im Active Directory über AD-Gruppen und Gruppenrichtlinien auf die Hyper-V-Server konfiguriert. Das ist ja auch schon der dritte Server.

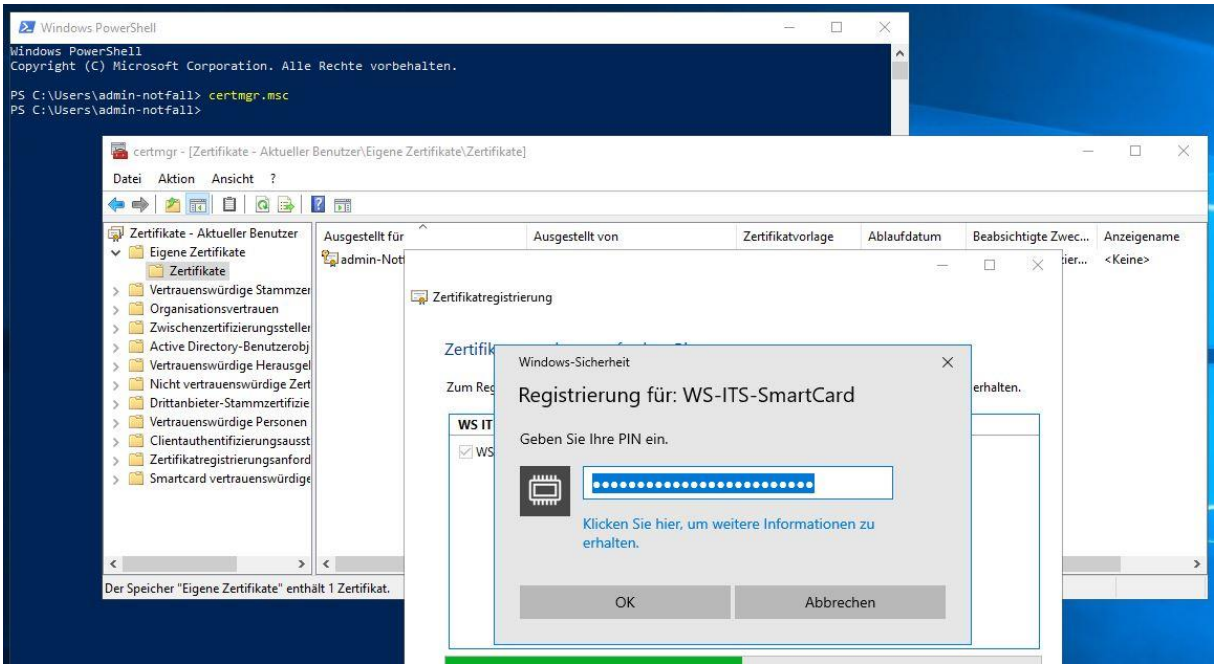
Jetzt fordere ich ein neues Smartcard-Zertifikat an:



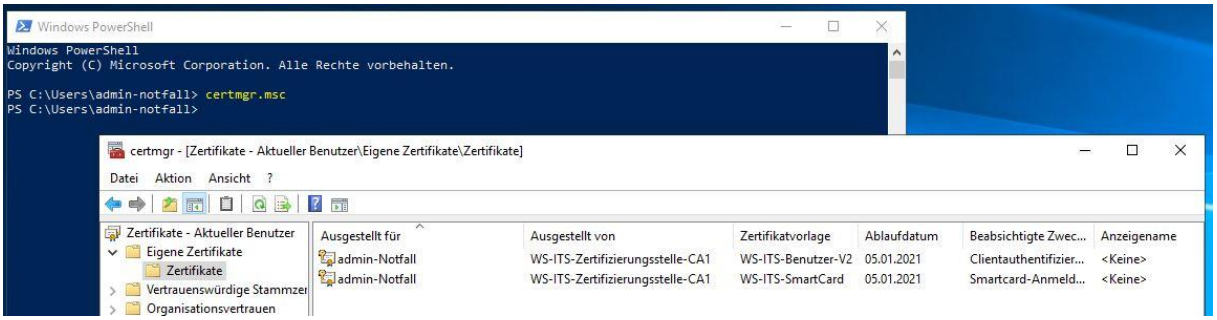
Der Benutzer hat auch dafür das erforderliche Recht in meiner PKI:



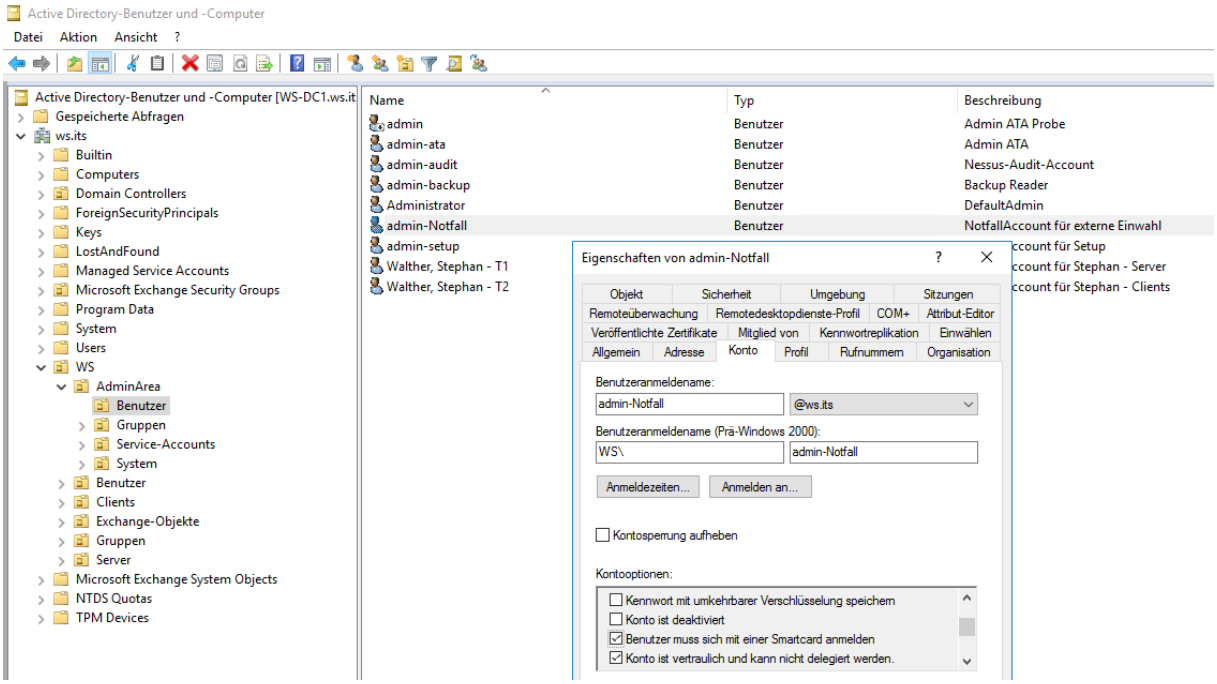
Die Vorlage habe ich so erstellt, dass Zertifikate auf einem entsprechenden Key Storage Provider gespeichert werden müssen. Hier meldet sich gleich die neue, virtuelle Smartcard. Das Zertifikat wird mit der PIN geschützt, die ich vorhin beim Erstellen eingegeben habe:



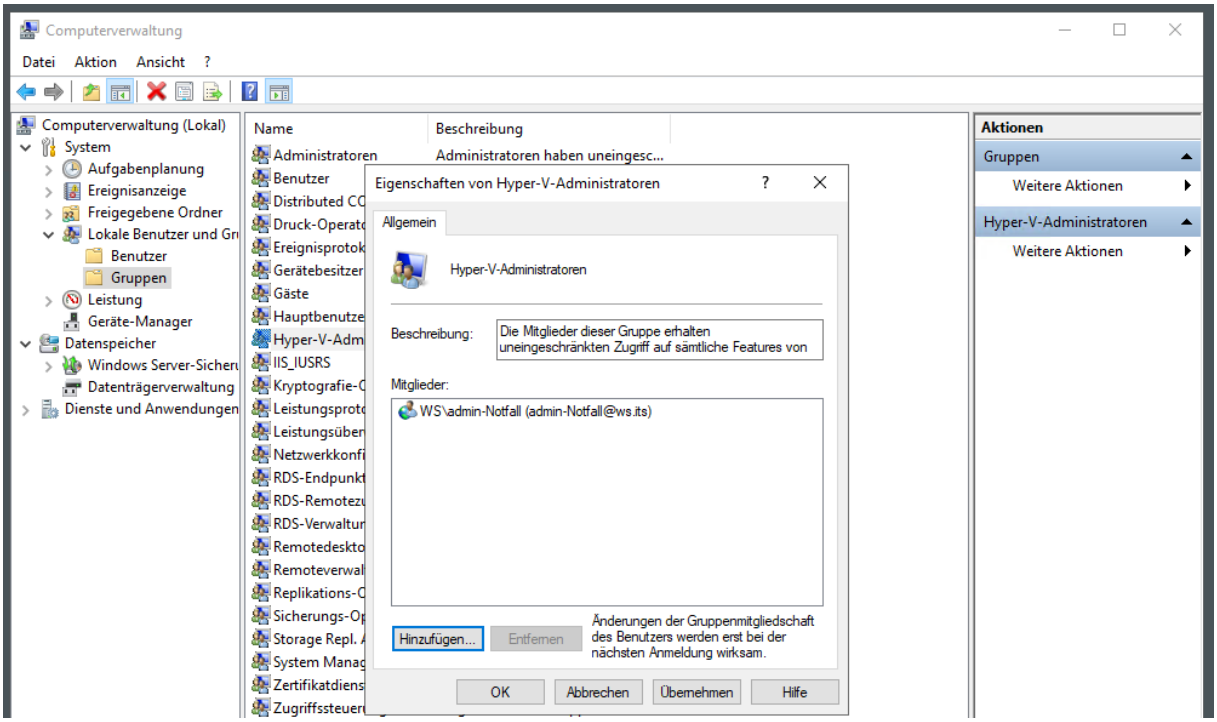
Und das war es auch schon. Die Smartcard mit dem Zertifikat ist einsatzbereit:



Jetzt aktiviere ich die Anforderung „Smartcard erforderlich“ wieder im Active Directory. Das Passwort des Accounts genügt nicht mehr:



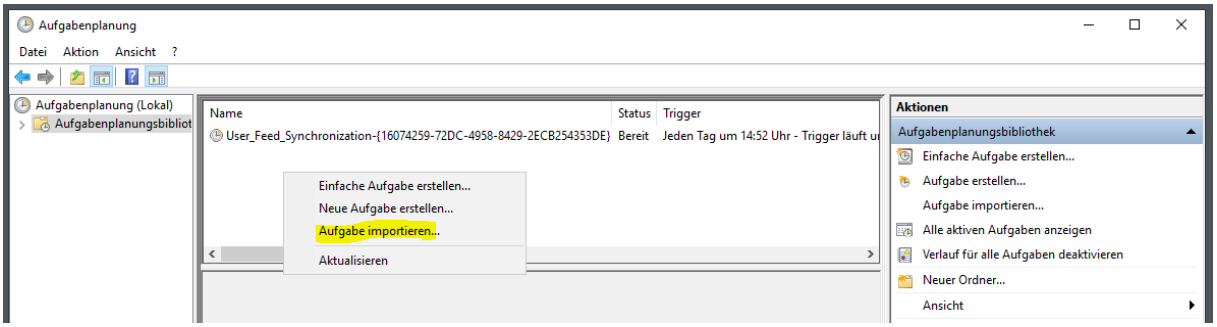
Die lokale Berechtigung zum Steuern des Hyper-V-Services habe ich nicht zentralisiert. Daher editiere ich die lokale Gruppe mit dem Administratoraccount:

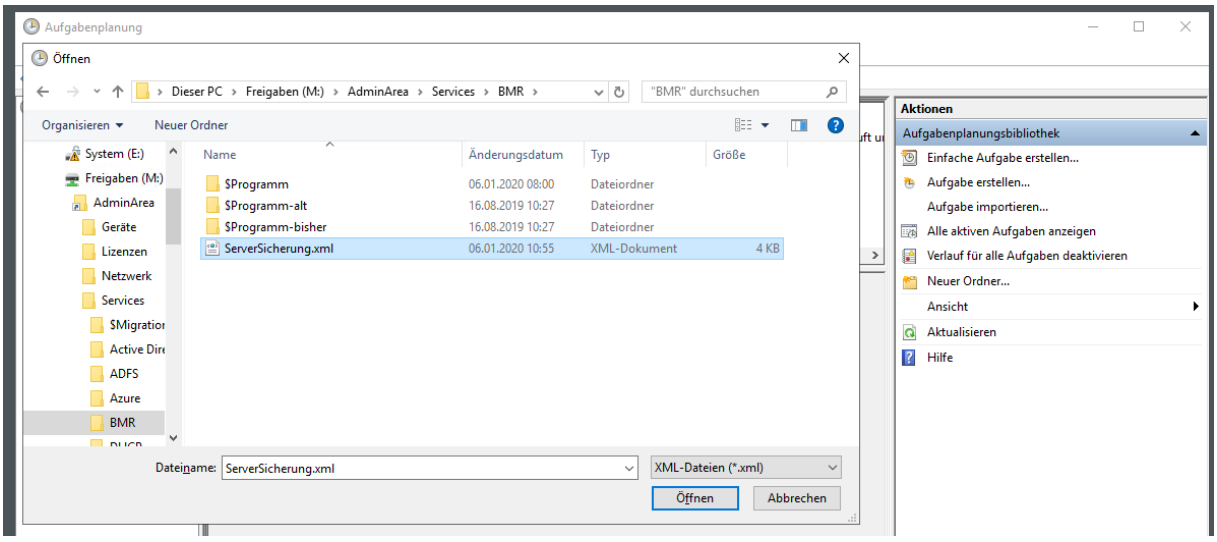


Den Testlauf führe ich später durch.

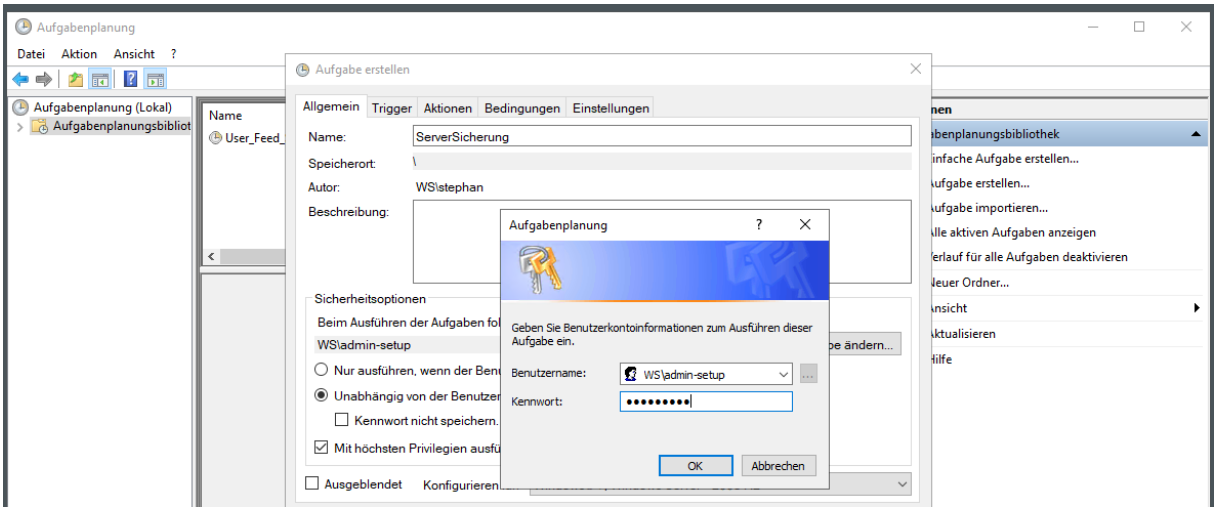
Konfiguration der Datensicherung – Windows Server Sicherung

Wie bei jedem neuen Server ist auch hier eine Datensicherung erforderlich. Die Systemstate-Sicherung übernimmt wieder meine Scriptlösung um die Windows Server Sicherung. Den Sicherungstask importiere ich wie üblich aus einer fertigen xml-Datei:

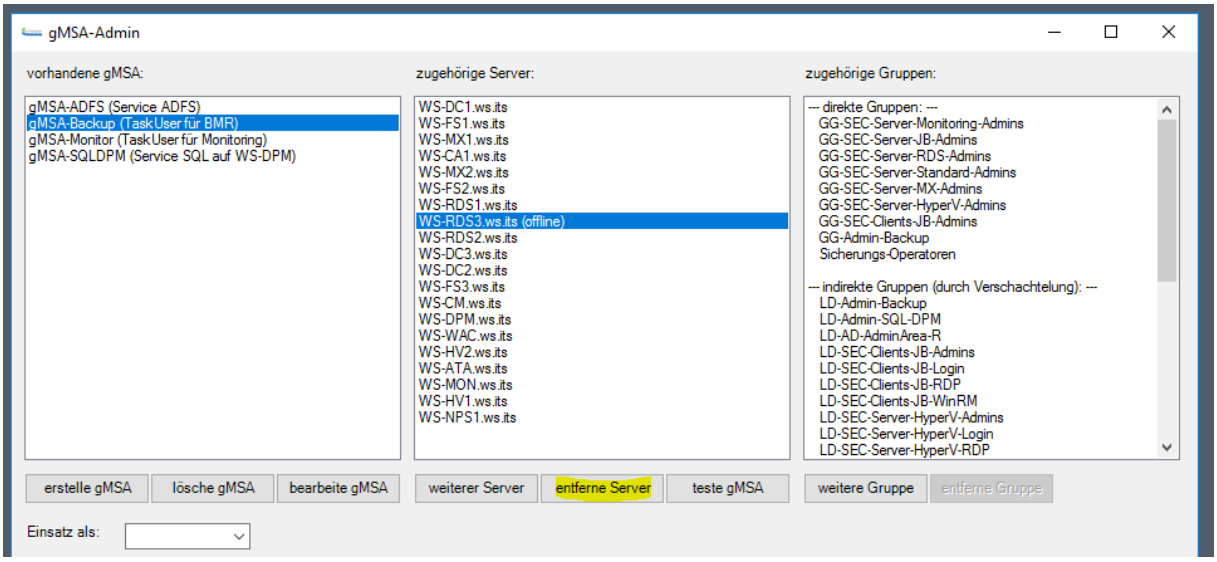




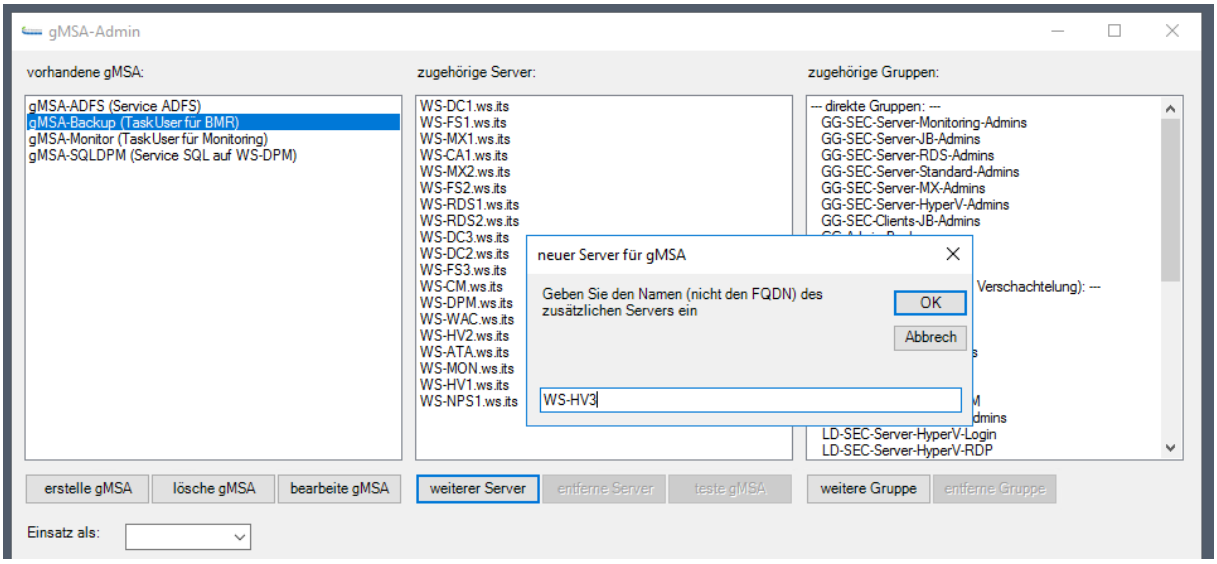
Und wie üblich trage ich einen Dummy-Account als Prinzipal ein:



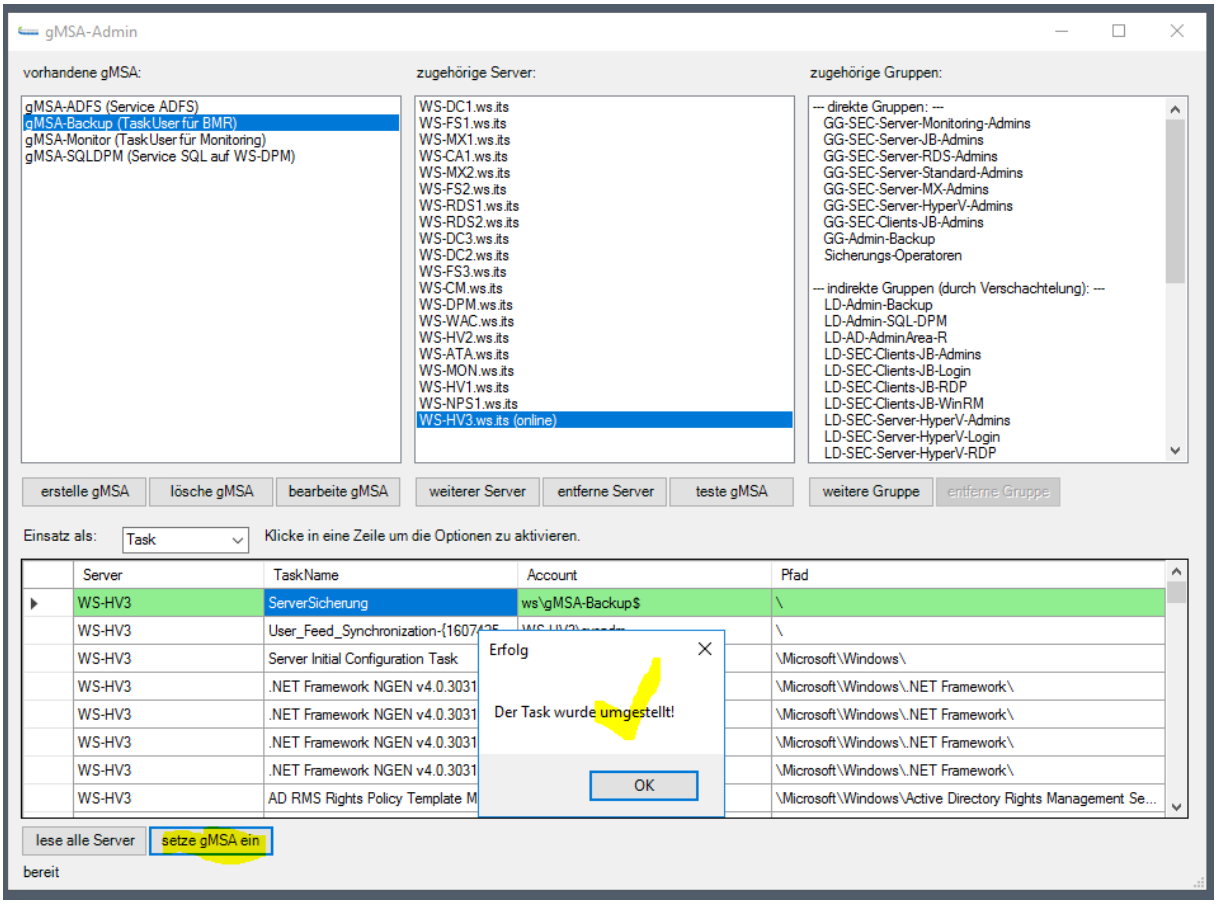
Danach wechsle ich auf meinen Domain Controller und richte den Group Managed Service Account für die Sicherung ein. Mein GUI-Script hilft mir dabei. Hier entferne ich zuerst den alten Server WS-RDS3:



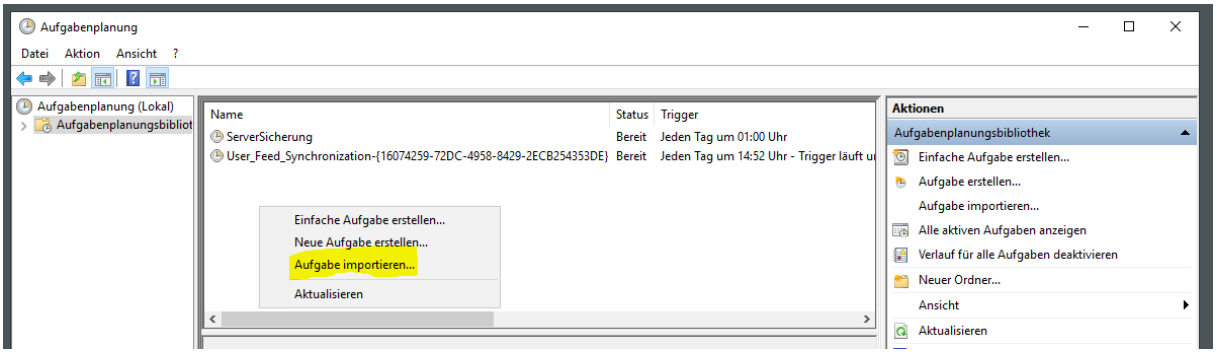
Dann kommt der neue Server in die Liste:

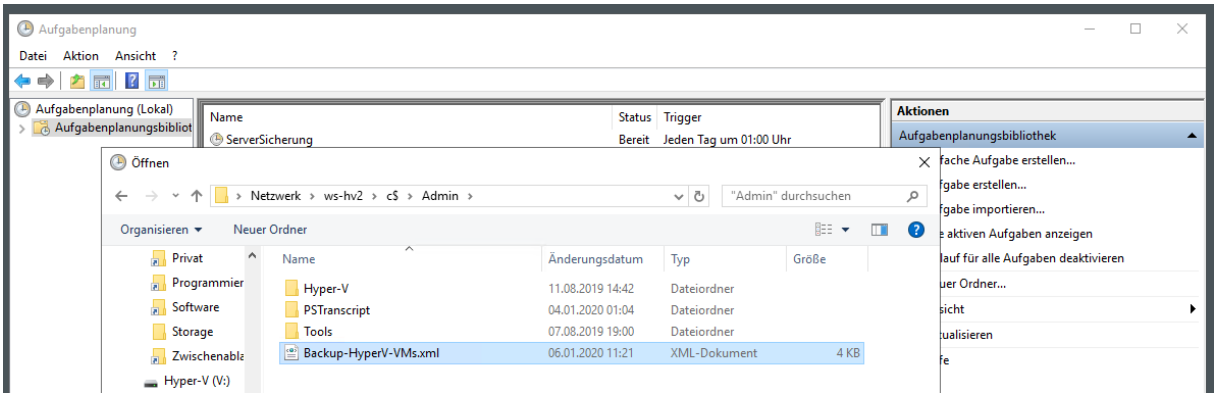


Danach trage ich den neuen gMSA als Sicherungstask-User ein. Die Aktion wird über PowerShell-Remoting durchgeführt:

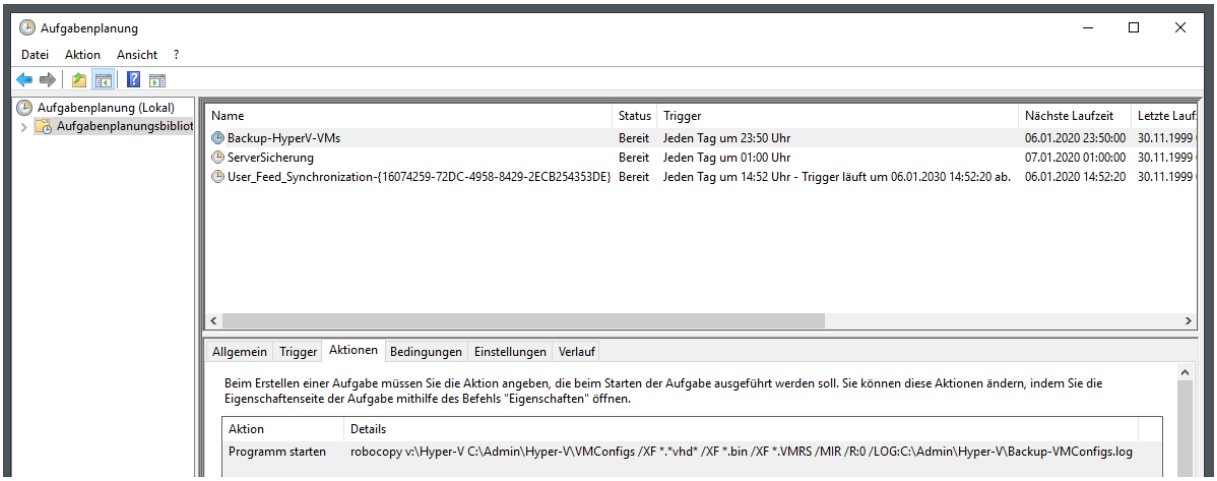


Weil ich auf WS-HV3 gerade bei den Aufgaben bin importiere ich gleich noch eine andere Aufgabe, die alle Hyper-V-Hosts bei mir aussführen. Diese Aufgabe kopiert die Konfigurationsdateien der virtuellen Maschinen in ein Verzeichnis, dass von der SystemState-Sicherung erfasst wird:

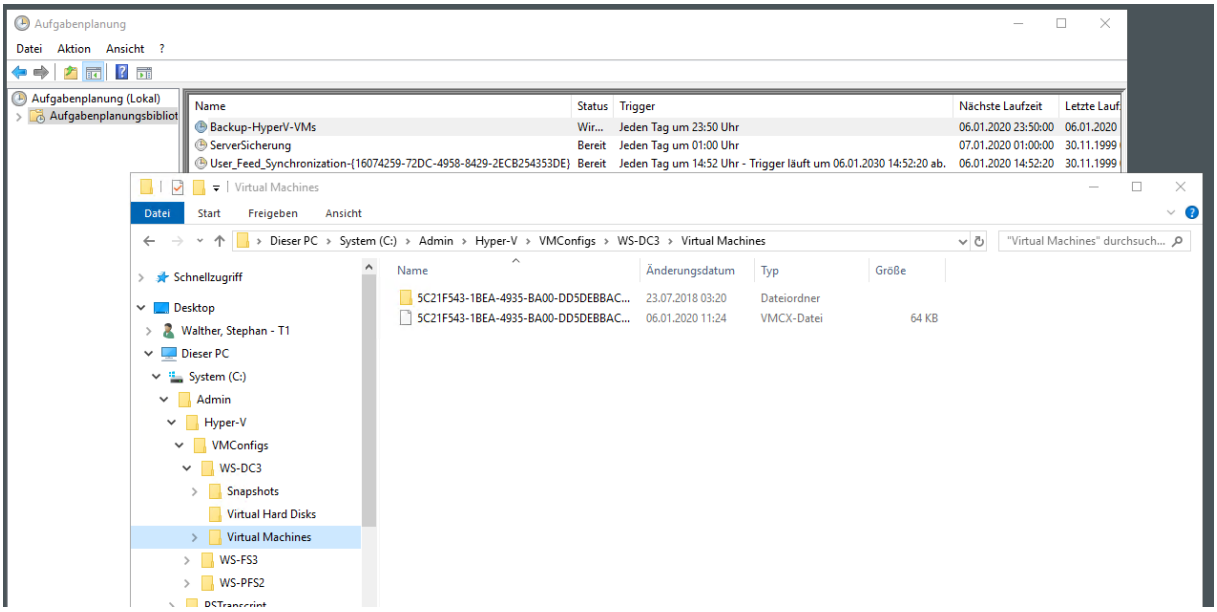




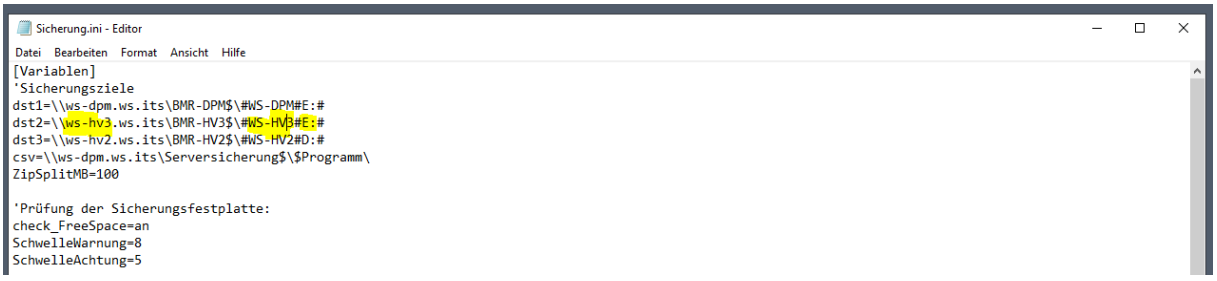
Ich starte diese Aufgabe:



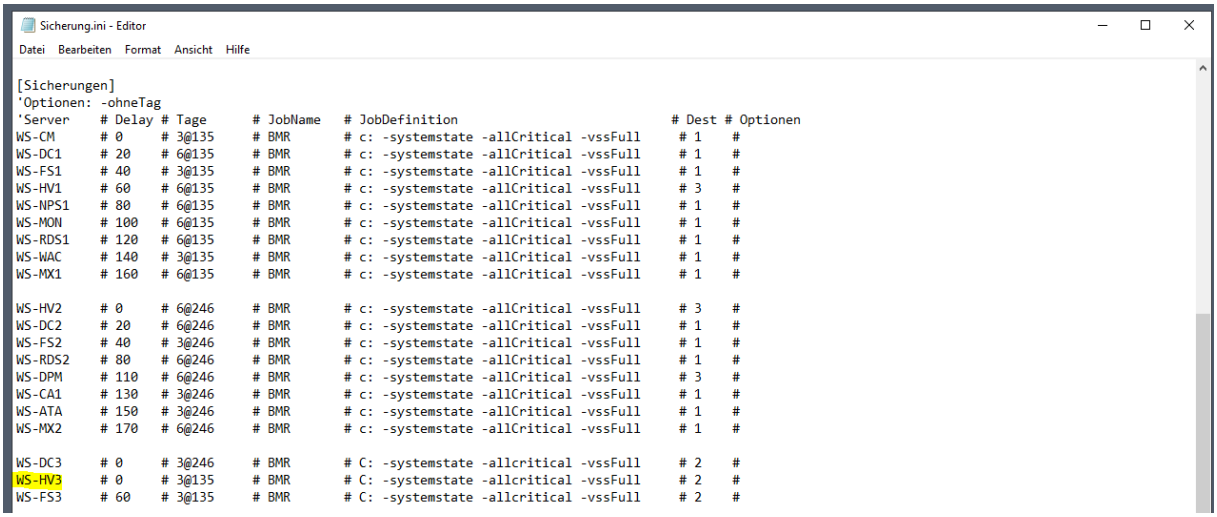
Danach sind die Konfigurationsdateien der VMs in einem Verzeichnis unter dem Systemlaufwerk dupliziert. Mit diesen Dateien kann ich die VMs ohne deren virtuelle Festplatten wiederherstellen:



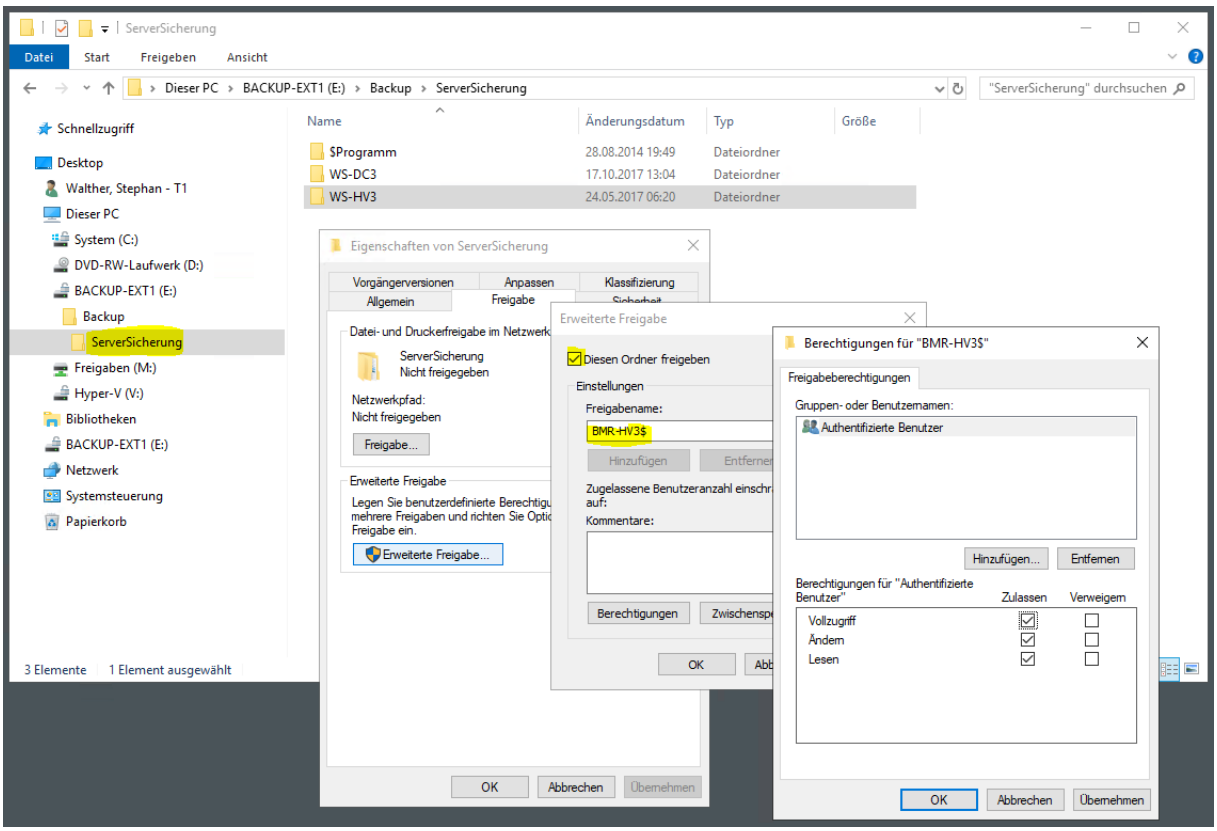
Für das Backup des SystemStates verwende ich wie bereits erwähnt eine Scriptlösung. Der lokale Task ist bereits eingerichtet. Aber die Steuerung muss mit der ini-Datei noch angepasst werden. Der neue Server WS-HV3 wird selber ein Sicherungsziel mit einer externen USB-Festplatte über eine Freigabe anbieten. Alle Server im Außenstandort werden dort hin sichern. In der ini-Datei verändere ich den Parameter und gebe den neuen Namen an:



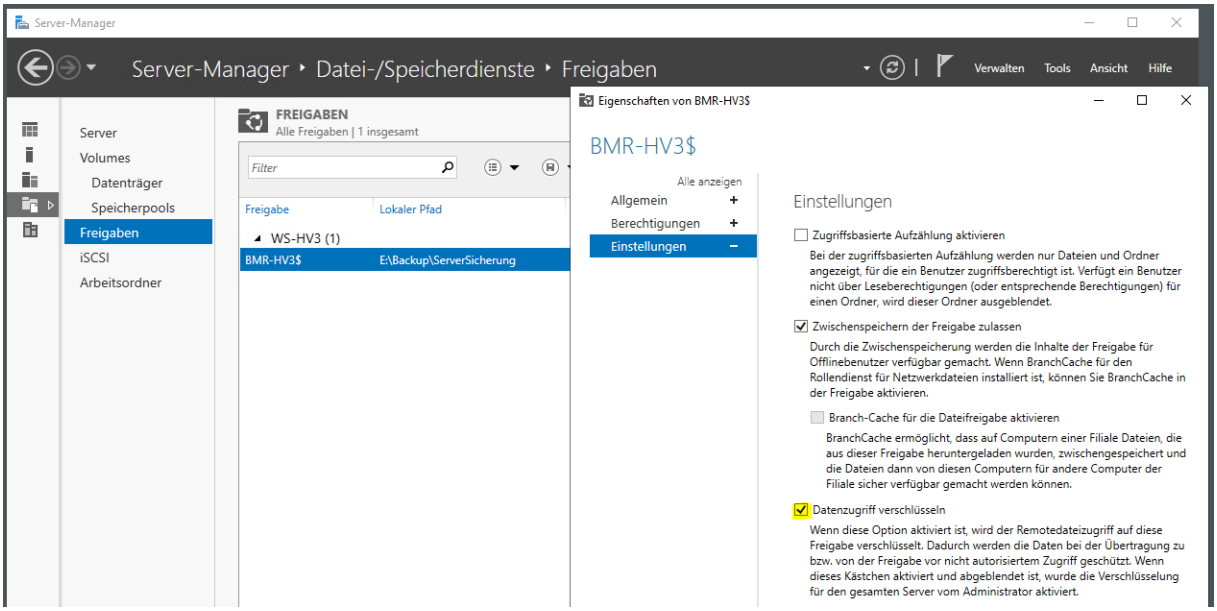
Im zweiten Block der ini-Datei trage ich auch die Zeile des Servers selber ein. Es ist eigentlich ganz einfach: Das Script wird über den Task gestartet, sucht in der ini-Datei nach der Zeile, die mit dem Namen des Servers beginnt. In der Zeile steht am Ende eine Ziffer. Diese stellt das Sicherungsziel dar. Die Ziffer 2 verweist dann auf das SMB-Share \\ws-hv3.ws.its\BMR-HV3$. Und dort wird dann gemäß Definition reingesichert:



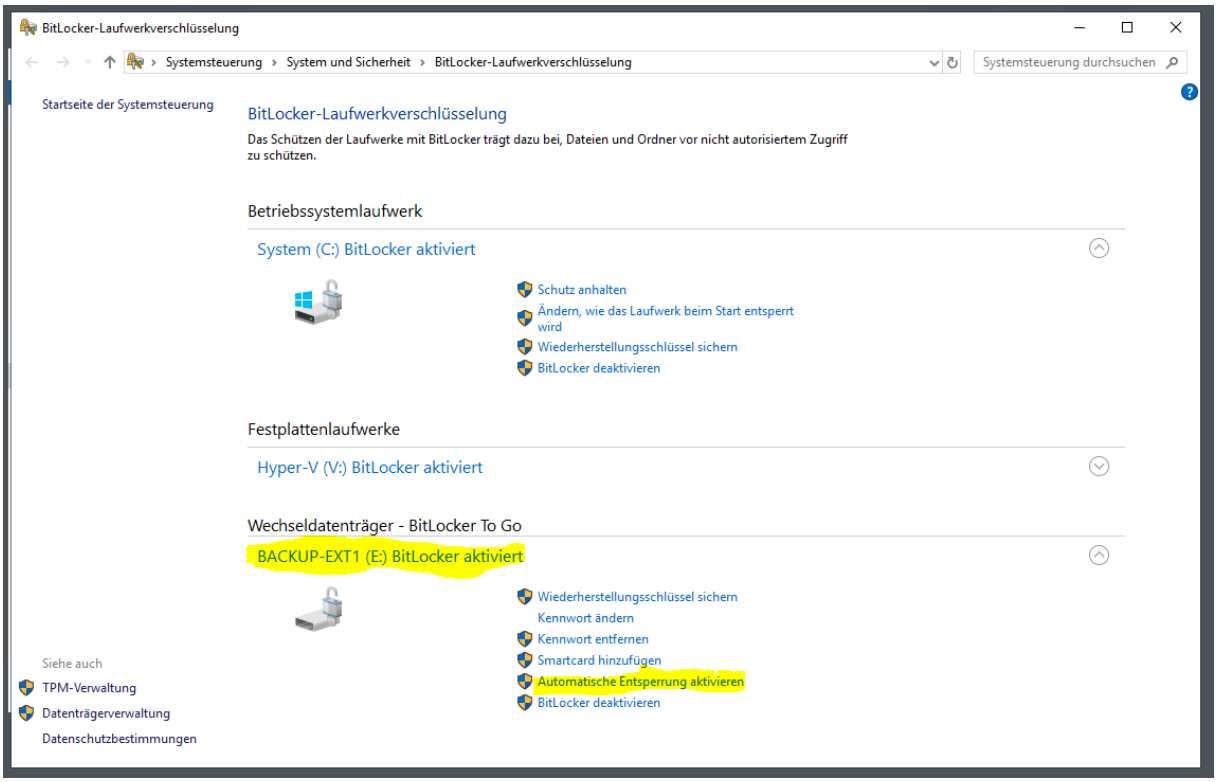
Das Sicherungsziel ist eine USB-Festplatte. Diese schließe ich jetzt an den neuen Server an. Die Freigabe existiert natürlich noch nicht. Aber das lässt sich einfach nachholen:



Zusätzlich schütze ich aber noch den Datenstrom mit SMB3-Encryption. Der Effekt ist mit dem Wechsel von http zu https vergleichbar: Die Daten können auf dem Transportweg nicht mehr im Klartext mitgelesen werden. Und das ist auch gut so. Denn auch mein virtueller Domain Controller wird in das Sicherungsziel sichern und dabei seine AD-Datenbank mit übertragen…

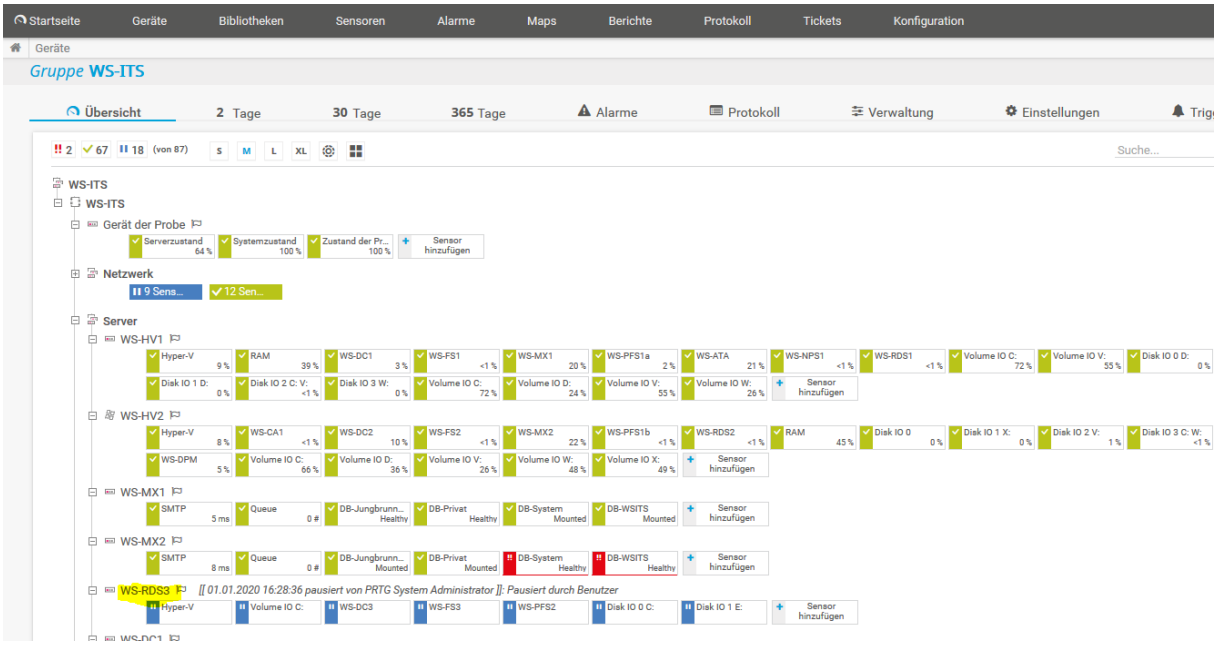


Die externe Festplatte selber stellt natürlich auch ein Sicherheitsrisiko dar. Daher verschlüssele ich die Partition mit Bitlocker To Go:

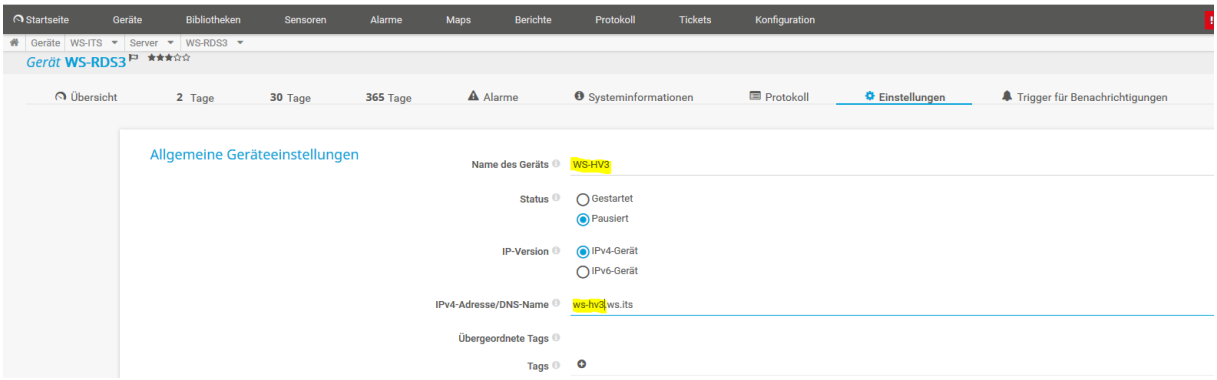


Konfiguration des Monitoring

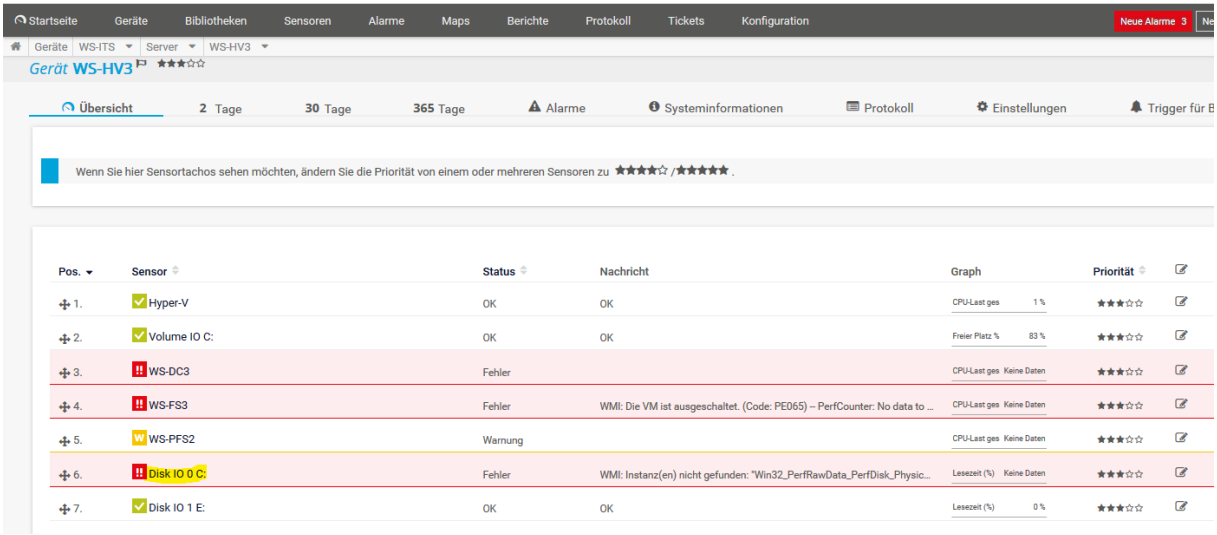
Weiter geht es mit der Integration in das Monitoring. Im PRTG hatte ich die Sensoren des alten Servers WS-RDS3 pausiert:



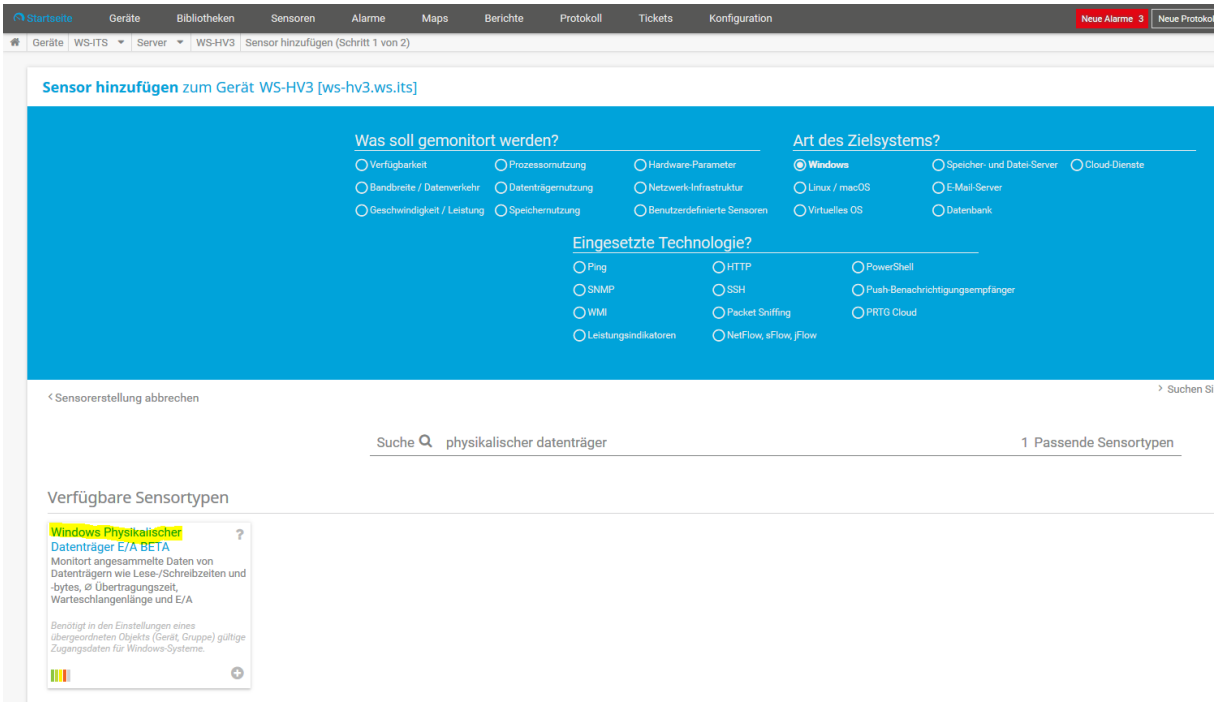
Den Eintrag kann ich editieren. Ich verändere den Anzeigenamen und den FQDN des Zielservers:



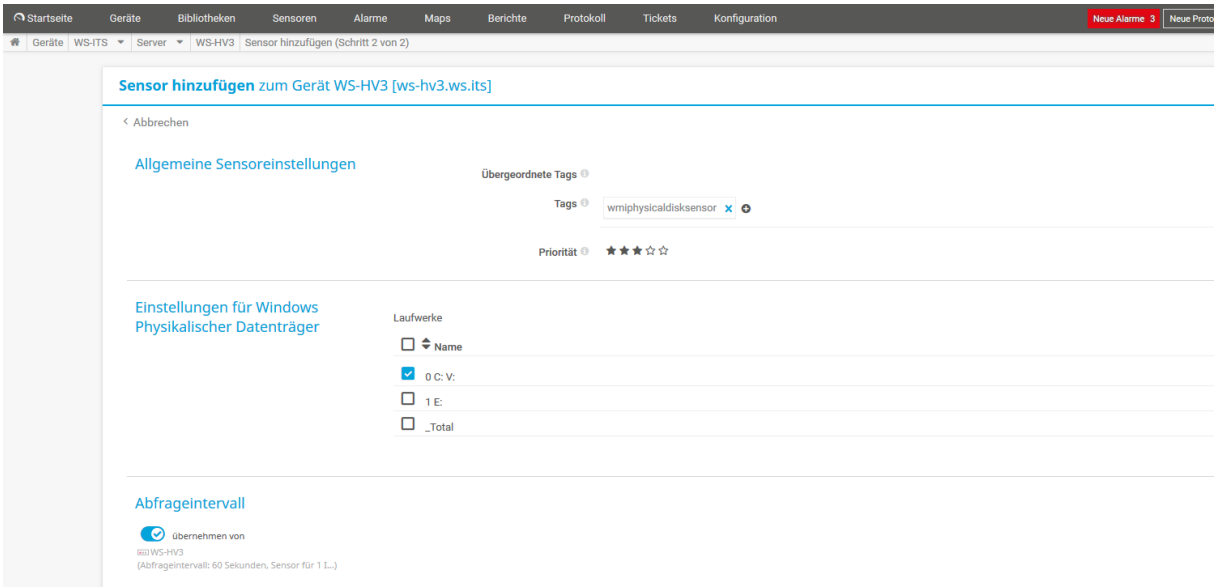
Die Assoziierung wird aber nicht komplett übernommen. Die Einträge für meine VMs sind nicht funktional. Und der neue Datenträger wird auch nicht erkannt. Egal, ich lösche sie und erstelle sie neu:



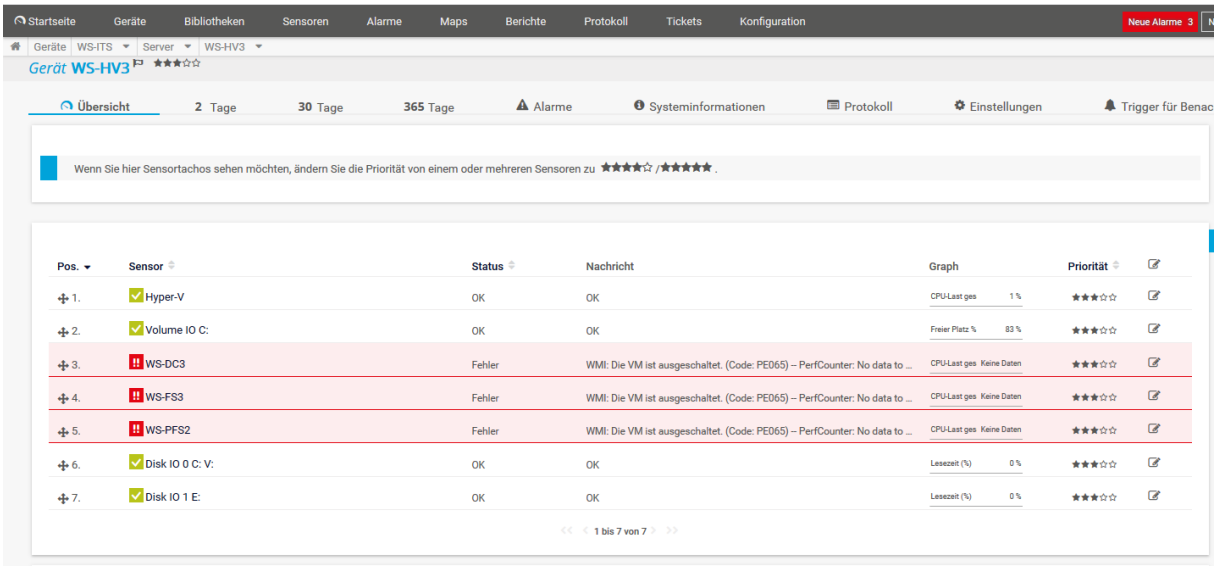
Dazu suche ich nach dem Sensor für physikalische Datenträger:



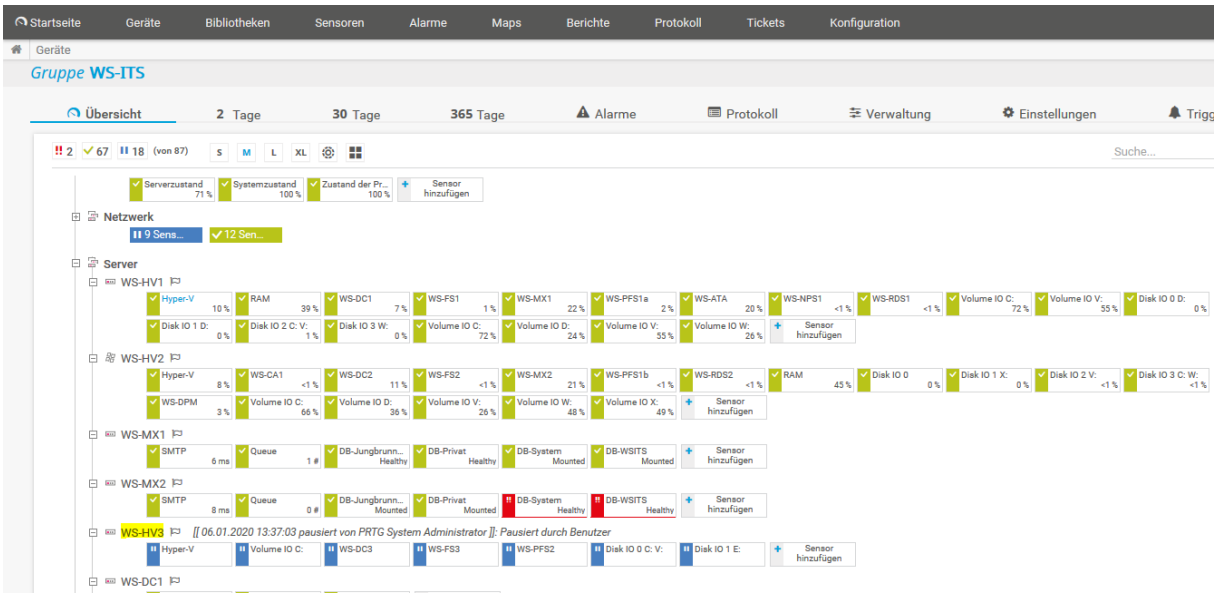
Der Dialog ist selbsterklärend:



Mit den virtuellen Maschinen verfahre ich gleich. Zuerst werden die Einträge gelöscht und dann wieder neu dazu genommen. Der Zustand bleibt zwar im Status „fehlerhaft“, aber die Meldung ist korrekt: die VMs sind ausgeschaltet:

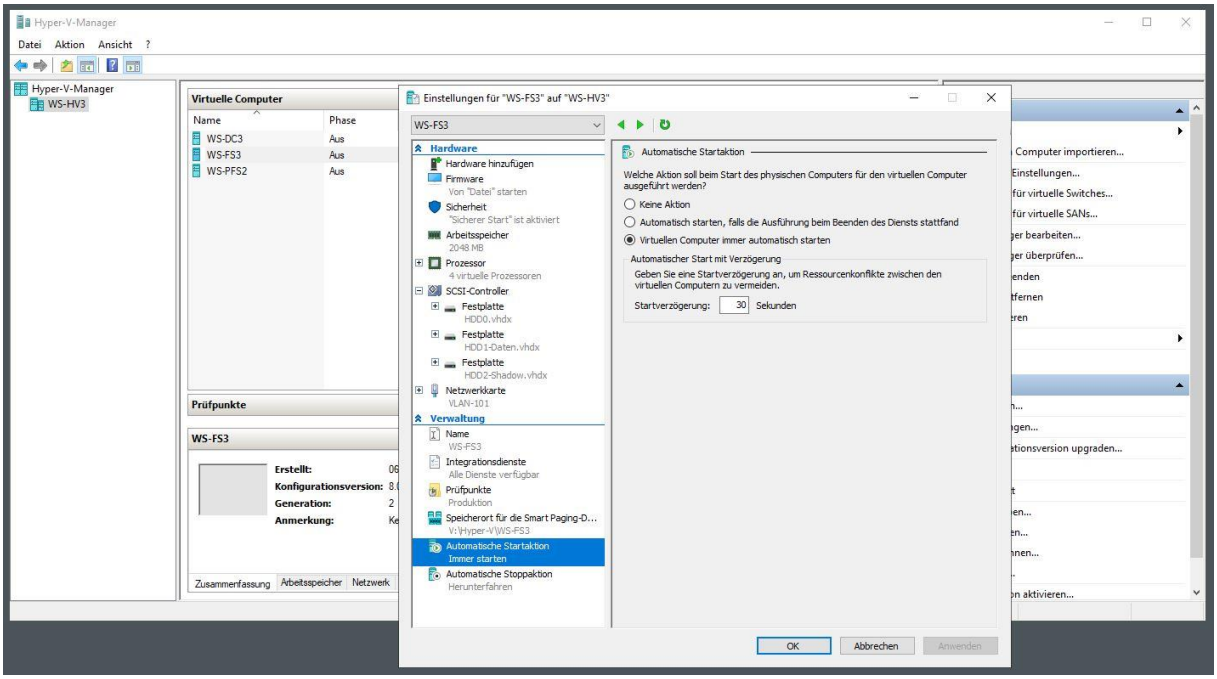


Für den bevorstehenden Umbau belasse ich die Sensoren im Pause-Modus:

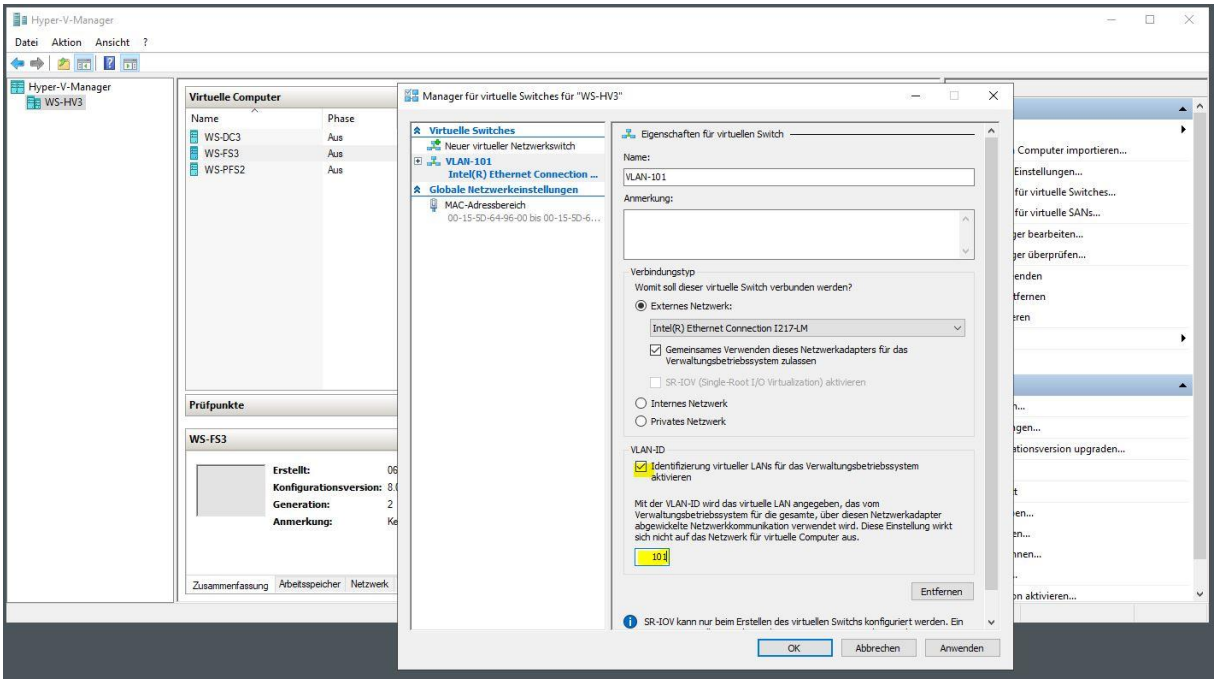


Einbau und Inbetriebnahme

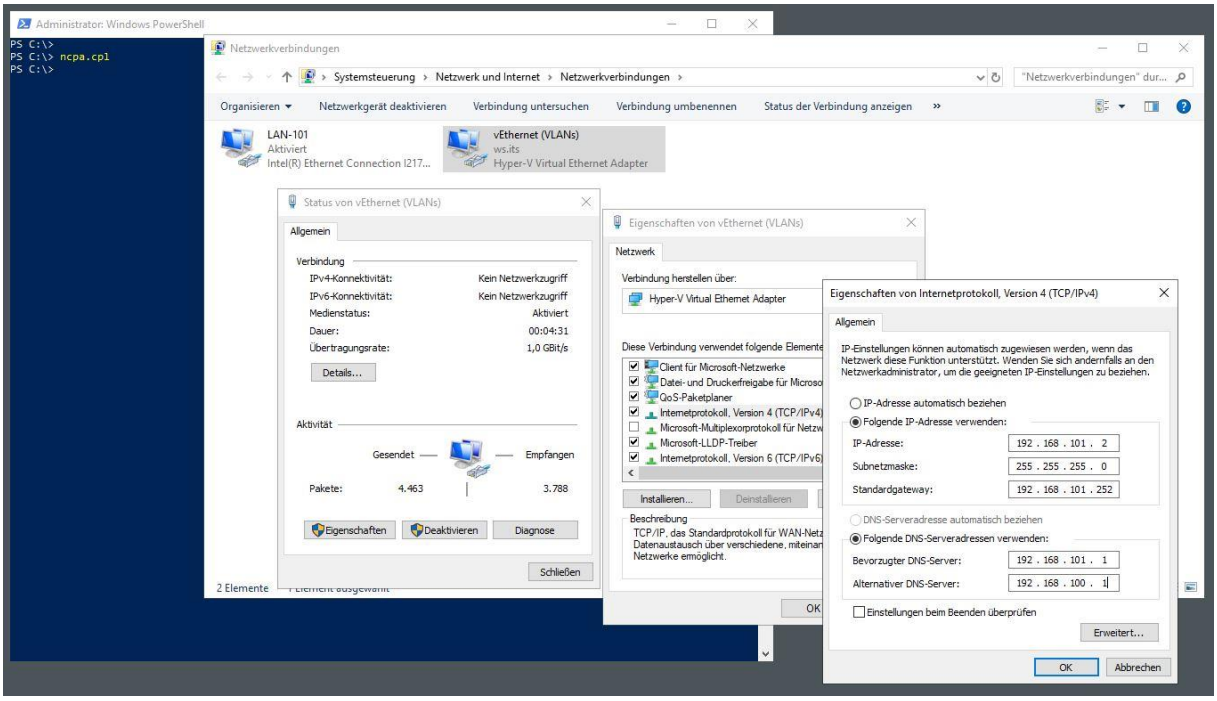
Der Server ist einsatzbereit. Es wird Zeit, dass er seine alte Position einnimmt. Zuerst editiere ich alle VMs für einen automatischen Start:



Dann bearbeite ich lokal die Einstellung des virtuellen Hyper-V-Switches. Der Server benötigt für das Zielnetzwerk wieder eine VLAN-ID:



Auch die statische IPv4-Konfiguration trage ich wieder ein:



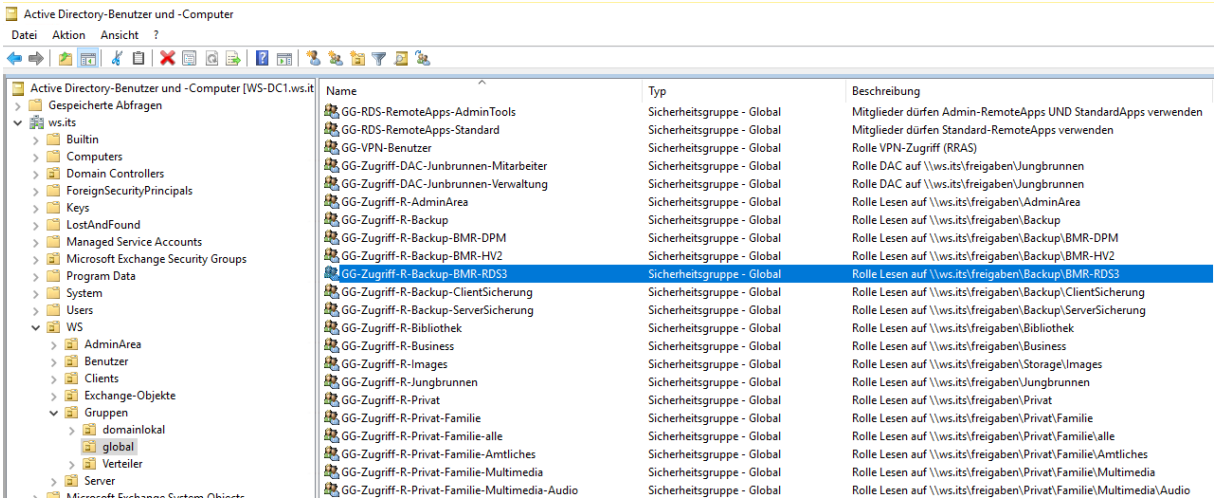
Es ist soweit: Ich schalte den Server im Hauptstandort aus und bringe in zum Außenstandort. Dort angeschlossen schalte ich das System ein und warte einen Moment. Die virtuellen Maschinen starten wie erwartet im Hintergrund. Mit dem Start der virtuellen PFSense wird auch das Netzwerk wieder aktiv. Der Domain Controller WS-DC3 repliziert die verpassten Änderungen des AD und der Fileserver WS-FS3 startet seine DFS-Replikation.

Die Umstellung des Servers scheint abgeschlossen zu sein.

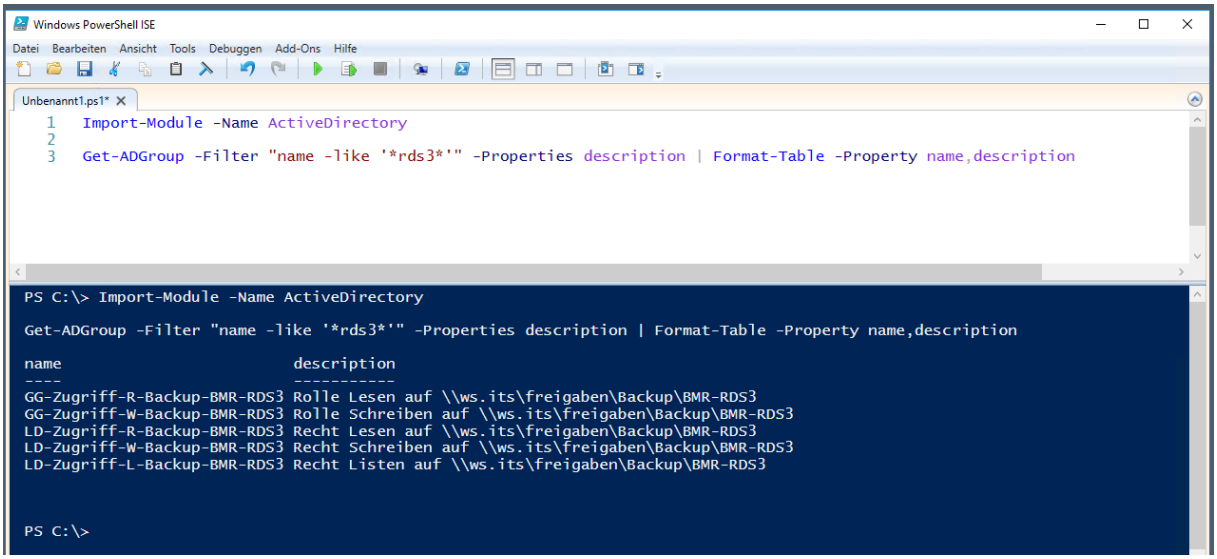
Nacharbeiten

Gruppenanpassungen im Active Directory

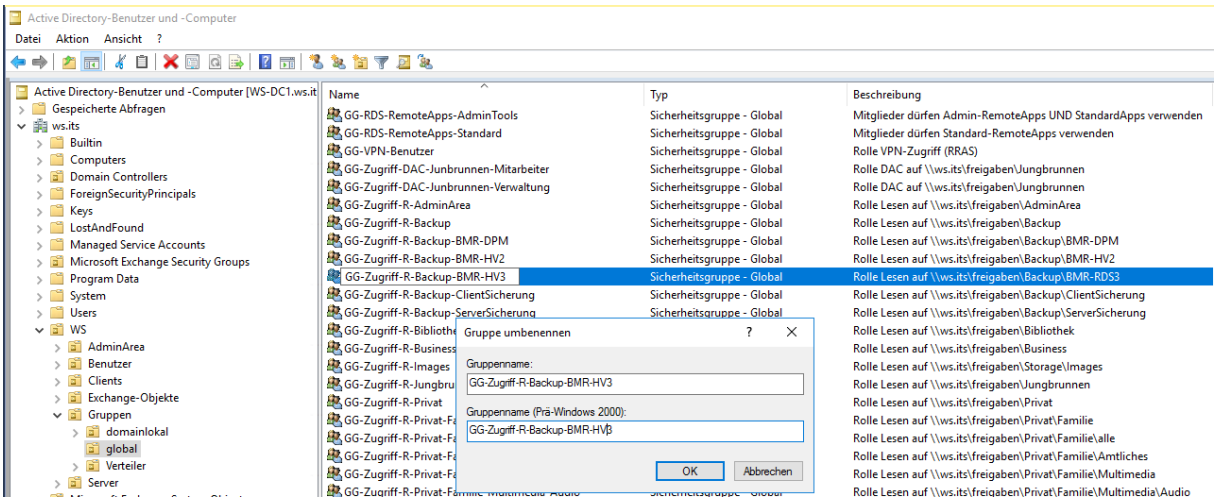
Aber wie so oft gibt es noch Reste, die eine Bereinigung benötigen. Im Active Directory gibt es beispielsweise einige AD-Gruppen, die den alten Servernamen im eigenen Bezeichner und der Beschreibung verwenden. Das stört mich. Daher suche ich diese Einträge und passe sie an:

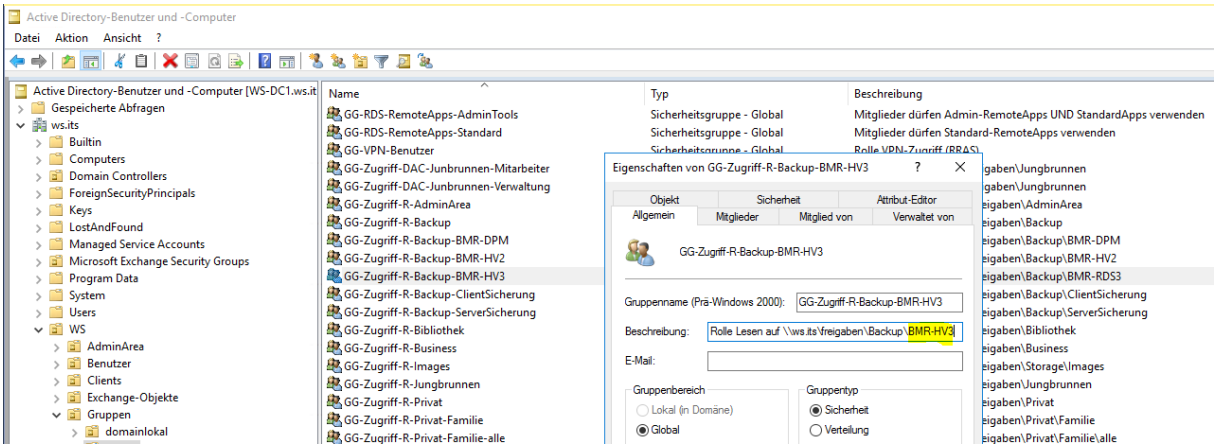


Für die Suche verwende ich eine PowerShell-Abfrage:



Die Umbenennung ist für 5 Gruppen inklusive Anpassung der Beschreibung etwas aufwändig. Daher ändere ich die Werte manuell:



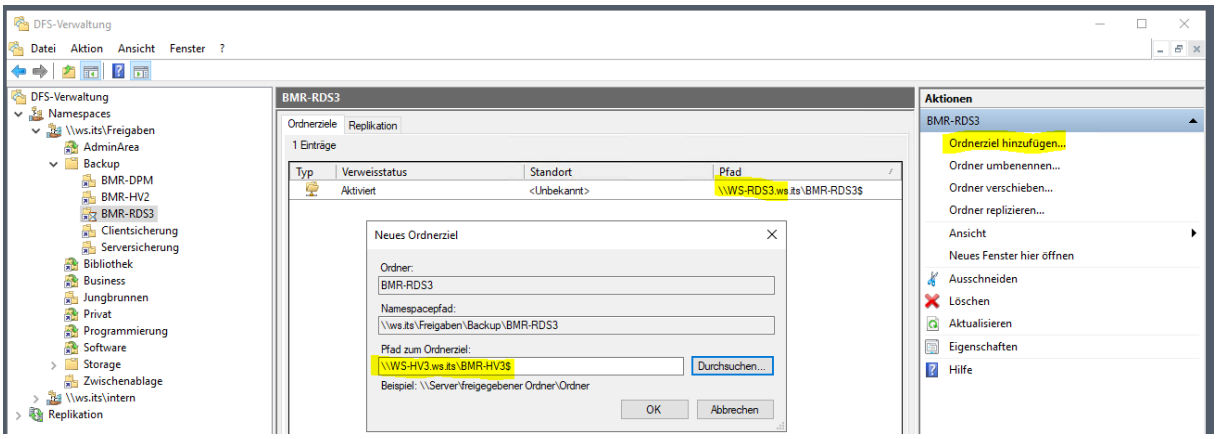


Eine Kontrolle mit der PowerShell zeigt die Bereinigung:

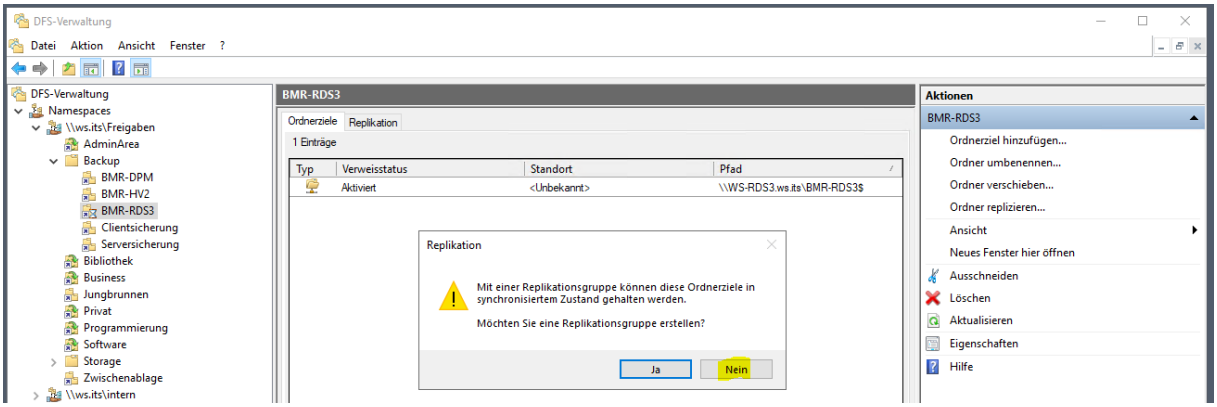
```
PS C:\> Get-ADGroup -Filter "name -like '*hv3*'" -Properties description | Format-Table -Property name,description

name                        description
----                        -----------
GG-Zugriff-R-Backup-BMR-HV3 Rolle Lesen auf \\ws.its\freigaben\Backup\BMR-HV3
GG-Zugriff-W-Backup-BMR-HV3 Rolle Schreiben auf \\ws.its\freigaben\Backup\BMR-HV3
LD-Zugriff-R-Backup-BMR-HV3 Recht Lesen auf \\ws.its\freigaben\Backup\BMR-HV3
LD-Zugriff-W-Backup-BMR-HV3 Recht Schreiben auf \\ws.its\freigaben\Backup\BMR-HV3
LD-Zugriff-L-Backup-BMR-HV3 Recht Listen auf \\ws.its\freigaben\Backup\BMR-HV3
```

Anpassungen im DFS-Namespace

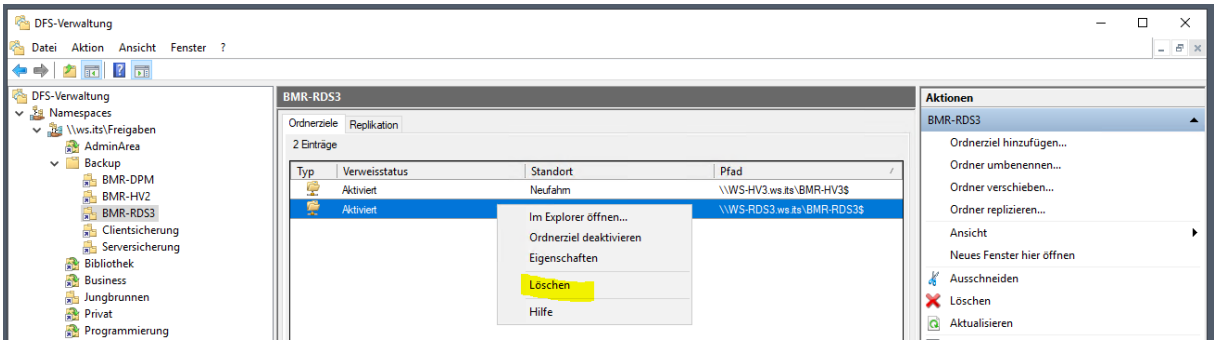
Auch in meinem DFS-Namespace existiert ein Ordner mit dem Namen des alten Servers. Über diesen Link konnte ich elegant auf die versteckte Freigabe des Sicherungsvolumes zugreifen. Zuerst ändere ich das Ziel. Dazu erstelle ich einen Link mit dem Pfad und dem Namen des neuen Servers:



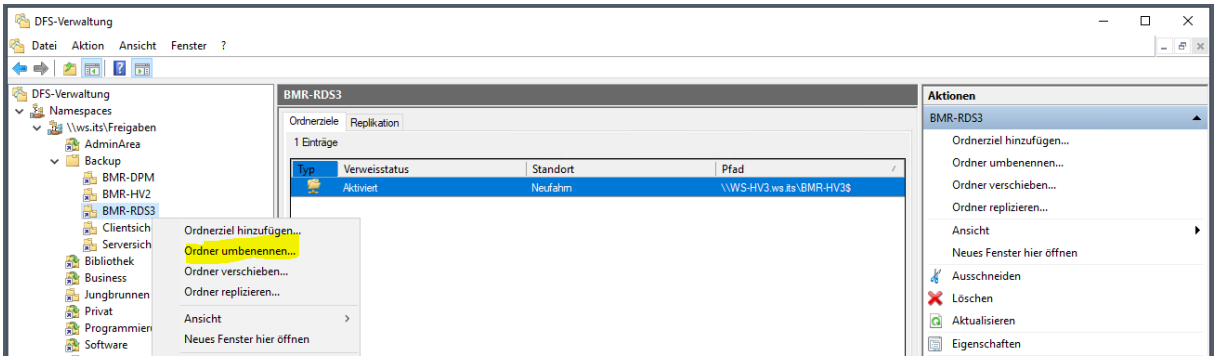
Ich richte keine Replikation ein. Den alten Server gibt es ja nicht mehr:

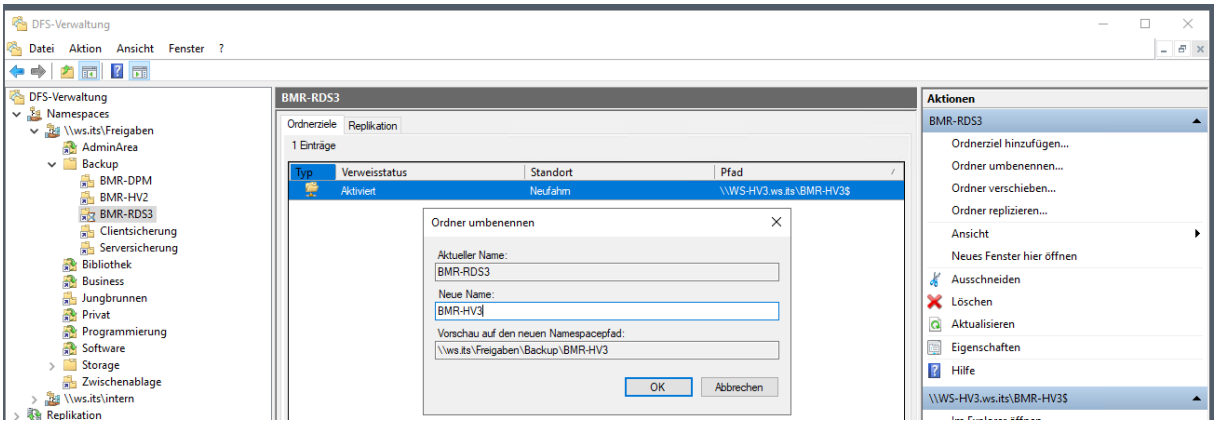


Dafür lösche ich den Link auf den alten Server. Jetzt werde ich auf das neue Share umgeleitet:



Den Namen des Ordners kann ich einfach umbenennen:





Die Änderung wird gemäß meiner Voreinstellung innerhalb von knapp 2 Minuten im Client aktiv.