Inhalt

Zielsetzung 2
Ist-Situation 2
Soll-Situation 2
Migrationsszenario 2
Vorbereitung 3
Berechtigungen 3
Review 3
Aufbau der neuen VM (mit Windows Update Problem) 7
Aufbau der neuen VM (Neuinstallation) 16
CleanUp PKI 19
Anpassungen an der PKI 23
Backup PKI 31
Migration 33
Austausch des Servers 33
Rolleninstallation 35
Problem mit 802.1x 37
Migration der ADCS 38
Anpassung und Überarbeitung der Sperrlistenverteilung 48
Grundkonfiguration und Ausstellung eines neuen Zertifizierungsstellen-Zertifikates 52
TroubleShooting – Korrektur des LDAP-Sperrlistenverteilungspunktes 57
Fortsetzung der Grundkonfiguration 62
Bereitstellung des Online Responders 63
Bereitstellung von CEPCES 74
Testphase 81
Aktivierung der PKI 88
Nacharbeiten 89
Konfiguration der Datensicherung 89
Monitoring 90
Updates 91
Bereinigungen 92
Problem Smartcard Logon 93
Problem Snort 97
Zusammenfassung 99

Zielsetzung

Ist-Situation

Ich betreibe in meiner Windows Server Infrastruktur eine eigene PKI. Diese wird durch den Server WS-CA1 bereitgestellt und läuft aktuell auf einem Windows Server 2016 Server Core. Es handelt sich dabei um eine Active Directory integrierte Root-Zertifizierungsstelle. Damals hatte ich keine Anforderung für eine mehrstufige PKI mit Offline-Root-CA.

Verschiedene Zertifikatvorlagen werden von meinen Systemen bei der automatischen Ausstellung von Zertifikaten genutzt. Die Zertifikate werden dabei nicht traditionell, sondern über **CEP-CES** ausgestellt (**C**ertificate **E**nrollment **P**olicy Service & **C**ertificate **E**nrollment Web **S**ervice). Diese beiden zusätzlichen Services auf meiner Zertifizierungsstelle ermöglichen die Anfrage und die Ausstellung von Zertifikaten über **HTTPS** statt **RPC-DCOM**, was eine Platzierung der Zertifizierungsstelle hinter einer Firewall deutlich vereinfacht.

Das Zertifizierungsstellen-Zertifikat läuft in den nächsten 12 Monaten aus.

Soll-Situation

Ich habe verschiedene Anforderungen für die Migration definiert:

- Das Betriebssystem soll auf Windows Server 2019 umgestellt werden.
- CEP-CES soll weiter genutzt werden.
- Zusätzlich soll ein Online Responder eine Echtzeitsperrprüfung ermöglichen.
- Das Sperrsystem soll später auch über das Internet erreichbar sein.
- Sperrlisten sollen nicht mehr über LDAP oder http erreichbar sein. Ich möchte nur noch über OCSP (Online Responder) verwenden.
- Eine Bereinigung der bisher ausgestellten Zertifikate ist durchaus sinnvoll.
- Das Zertifizierungsstellen-Zertifikat muss erneuert werden. Dabei können auch die neuen Certificate Revocation List Distribution Points veröffentlicht werden, denn diese werden in alle neuen Zertifikate integriert.
- Der Server Core hat mir mehr Probleme verursacht als Nutzen gebracht. Daher soll der neue Server mit der grafischen Oberfläche bereitgestellt werden.

Migrationsszenario

Durch die Besonderheit des erforderlichen neuen Zertifizierungsstellen-Zertifikats muss die Migration über mehrere Schritte geführt werden:

- **Cleanup**: Zuerst werde ich die Bereinigung und die Erneuerung des Zertifizierungsstellen-Zertifikats auf der alten Windows CA durchführen.
- **Migration**: Dann werde ich mittels Wipe & Load den Service auf das neue Betriebssystem verschieben. So kann ich den Namen des Betriebssystems weiterführen und die IPv4-Adresse wiederverwenden und erspare mir verschiedene Anpassungen in Richtlinien und Firewall-Regeln.
- **Erweiterung**: Zuletzt erweitere ich die PKI mit einem Online Responder.

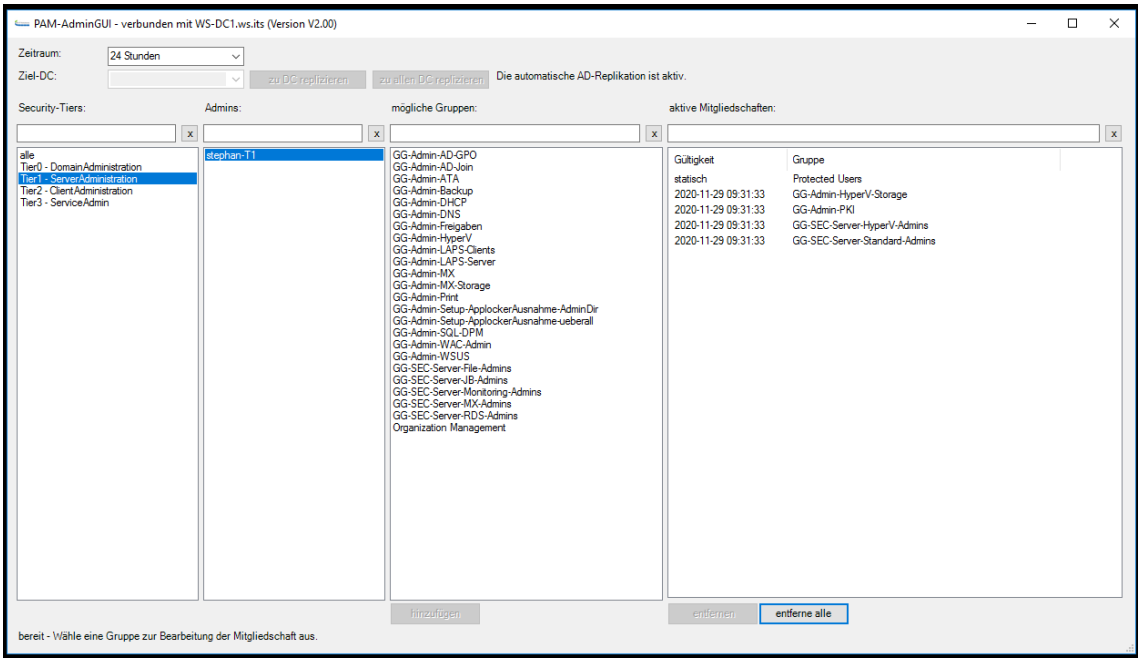
Generell kann man an dieser Stelle auch über eine parallele, neue Zertifizierungsstelle sinnieren. Diese könnte frisch aufgesetzt und nach Wunsch konfiguriert zuerst ausgiebig getestet werden. Anschließend könnte man alle bisher ausgestellten und noch gültigen Zertifikate auf allen Systemen mit der neuen CA erneut ausstellen. Wenn keine Zertifikate der alten CA mehr in Verwendung sind, dann könnte diese einfach abgeschaltet werden. Durch das bereits ablaufende Zertifizierungsstellen-Zertifikat ist die Restlaufzeit einiger ausgestellter Zertifikate bereits reduziert (Eine Zertifizierungsstelle kann keine Zertifikate über ihre eigene Gültigkeit hinaus signieren). Die vielen Windows Systeme könnten bequem durch Gruppenrichtlinien automatisiert migriert werden. Aber in meinem Fall sind auch fast alle Nicht-Windows-Systeme mit Zertifikaten versorgt: Das sind z.B. alle Netzwerkgeräte. Und da würde mir eine Umstellung der Zertifikate mehr Arbeit bereiten als eine Bereinigung und Migration der alten CA.

Daher bleibe ich beim Wipe & Load.

Vorbereitung

Berechtigungen

Für den Zugriff auf meine PKI bzw. den Review sind mehrere Gruppenmitgliedschaften erforderlich. Diese delegiere ich mit meinem PAM-Script für 24 Stunden an meine Admin-Kennung:

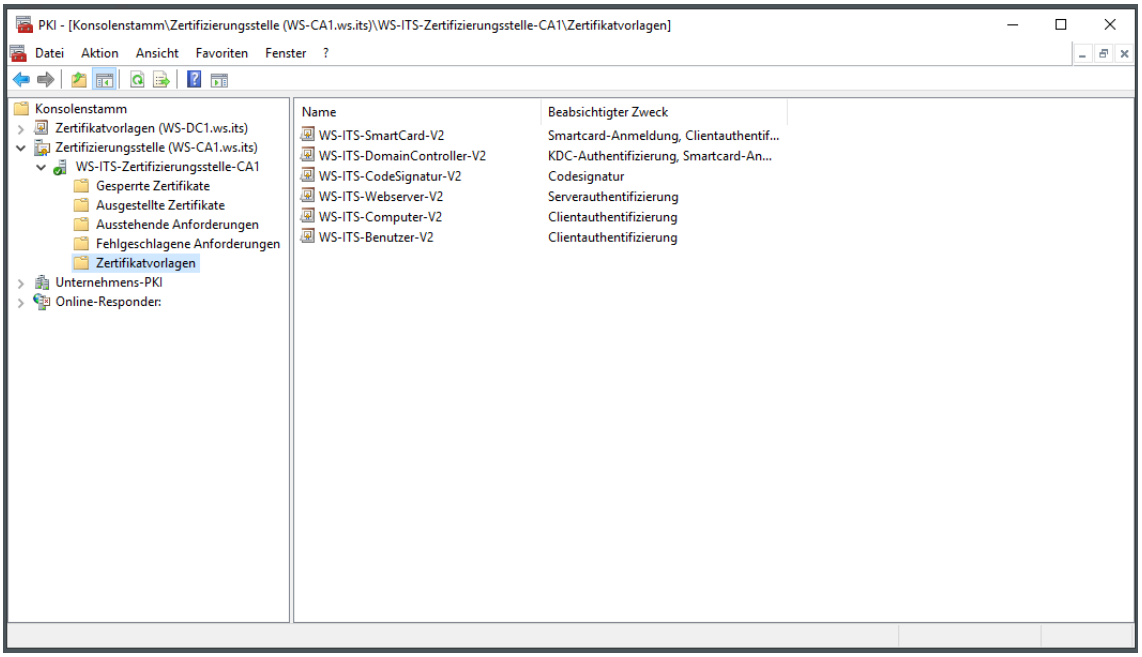


Die Gruppe „GG-Admin-PKI“ ist in eine Gruppe „LD-Admin-PKI“ verschachtelt, welcher ich die administrativen Rechte in meiner PKI delegiert habe.

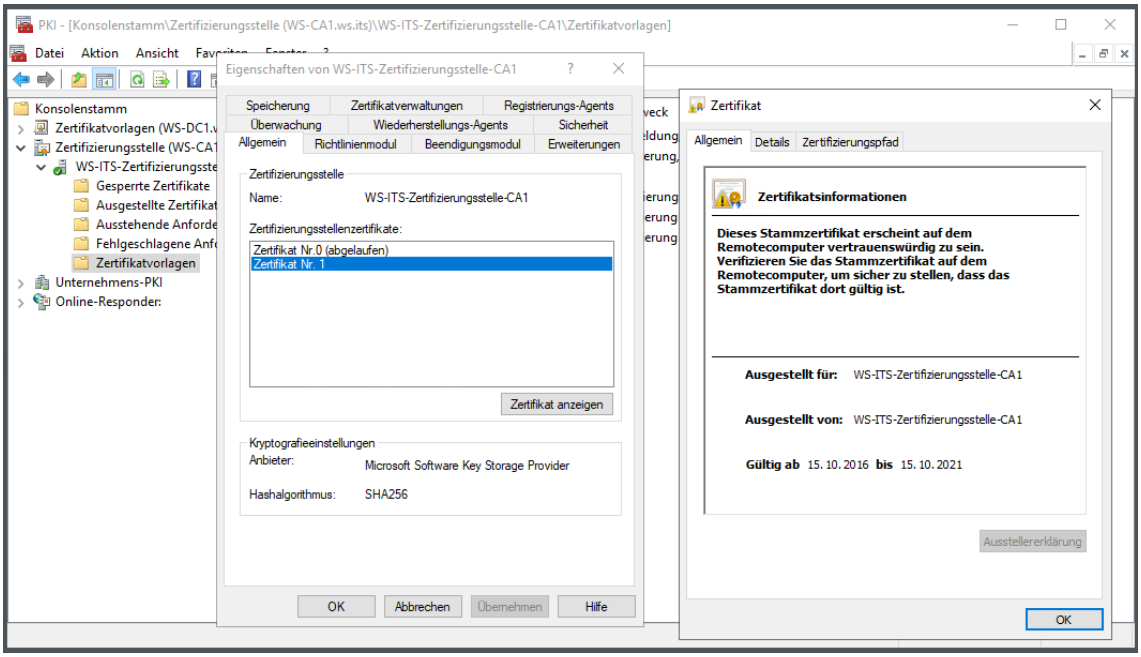
Review

So ausgestattet verbinde ich mich mit meiner Admin-Kennung auf meinen Admin-Server. Dort hatte ich mir eine Management-Konsole für den Remotezugriff auf die PKI-Services erstellt – der Server WS-CA hat ja als Server Core keine grafische Oberfläche.

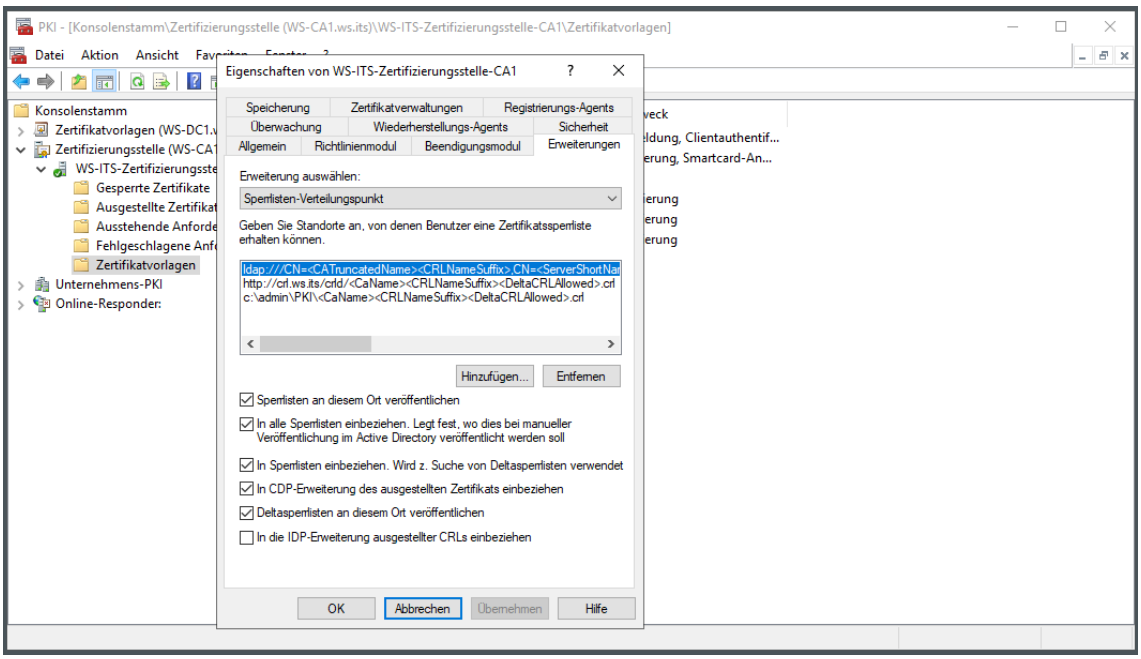
Beginnen wir mit den Zertifikatvorlagen. Ich denke, die Namensgebung sagt genug aus:



Das Zertifizierungsstellen-Zertifikat wurde bereits einmal erneuert. Das aktuelle ist jetzt weniger als 12 Monate gültig:

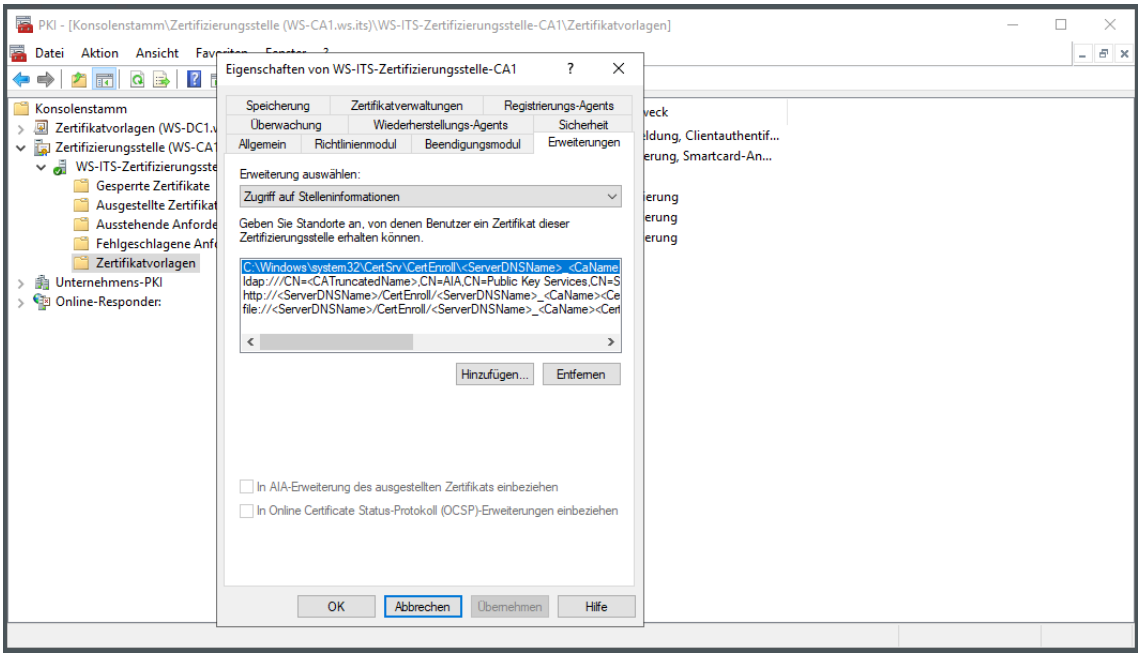


Zertifikate müssen auch gesperrt werden können, wenn sie z.B. kompromittiert wurden. Dafür müssen aber im Vorfeld Sperrlistenverteilungspunkte definiert werden. Ich habe damals den Webserver, der auf der Windows CA für CEPCES erforderlich ist, auch für die Veröffentlichung der CRL (**C**ertificate **R**evocation **L**ist) verwendet. Den Zugriff auf die Web-Ressource hatte ich an einen CNAME crl.ws.its gebunden. So hätte ich jederzeit eine Verschiebung auf einen anderen Webserver vornehmen können. Zusätzlich habe ich im Active Directory die Sperrliste veröffentlicht:

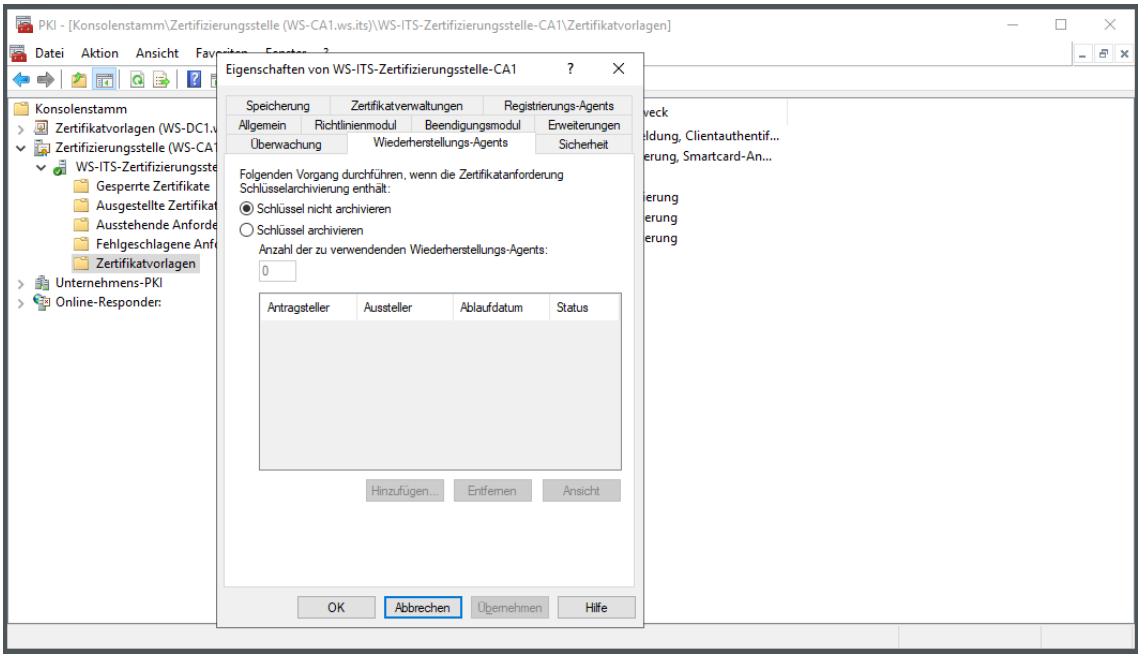


Der Pfad c:\admin… in den Verteilungspunkten ist das Backend-Verzeichnis des lokalen Webservers. Hier legt die CA die Sperrliste ab. Über den mittleren Eintrag wird dann der Zugriff über http definiert.

In den Stelleninformationen schaut es weniger angepasst aus. Diese arbeiten mit den Default-Settings:

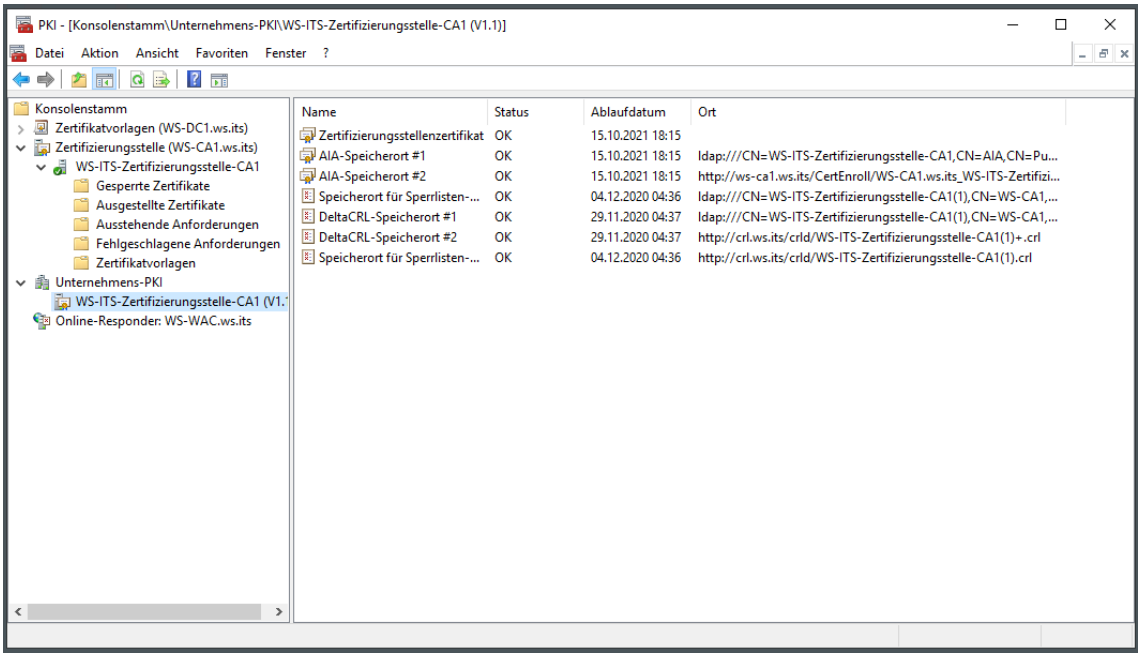


Ein wichtiger Punkt bei der Planung der Migration ist das Vorhandensein von archivierten, privaten Schlüsseln. Diese können Teil eines Recovery-Szenarios sein und machen immer dann Sinn, wenn Benutzer die ausgestellten Zertifikate für Datenverschlüsselungen verwenden (können). Verliert der Benutzer nach einer Verschlüsselung sein Zertifikat mit dem Private Key, dann kann er seine Daten selber nicht mehr entschlüsseln! In meiner CA habe ich darauf geachtet, dass die ausgestellten Zertifikat-Vorlagen keine Verschlüsselung ermöglichen. Daher hatte ich für die Archivierung auch keine Notwendigkeit:

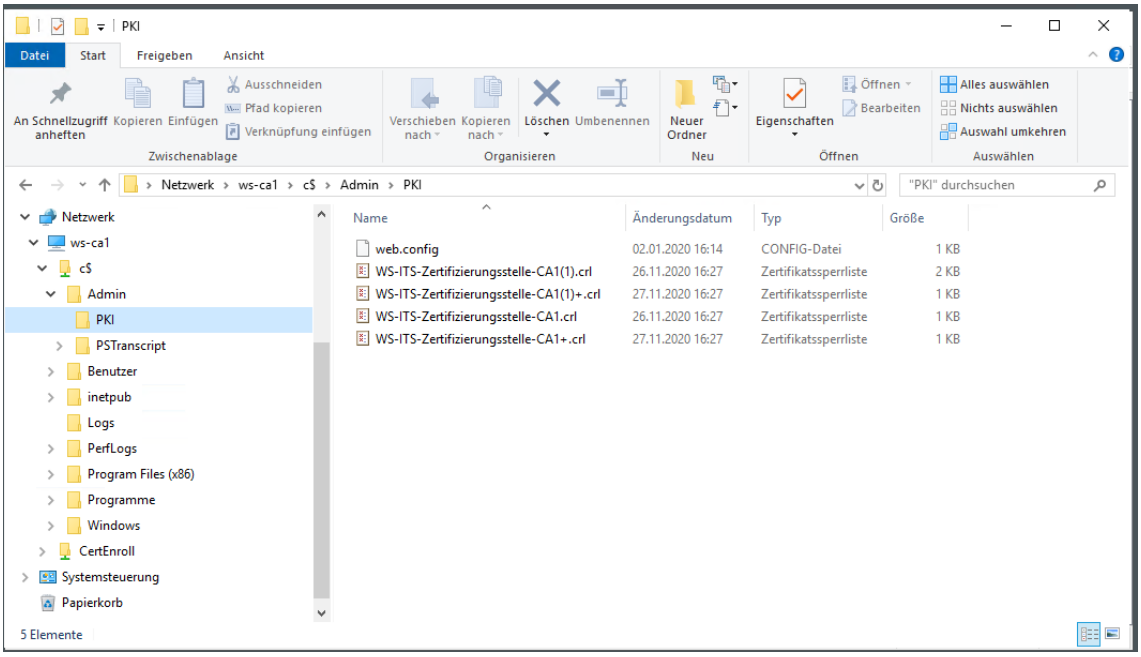


Hinweis: Wenn die alte CA über archivierte Schlüssel verfügt, dann ist eine Side-by-Side-Migration – also der Aufbau einer neuen CA und das Ablösen der alten CA – wesentlich schwieriger. Denn selbst wenn alle Benutzer neue Zertifikate erhalten haben und diese für Verschlüsselungen verwenden können: Es kann immer noch Daten geben, die mit den alten Zertifikaten verschlüsselt wurden. Und für eine Entschlüsselung benötigen die Benutzer demnach auch die alten Zertifikate!

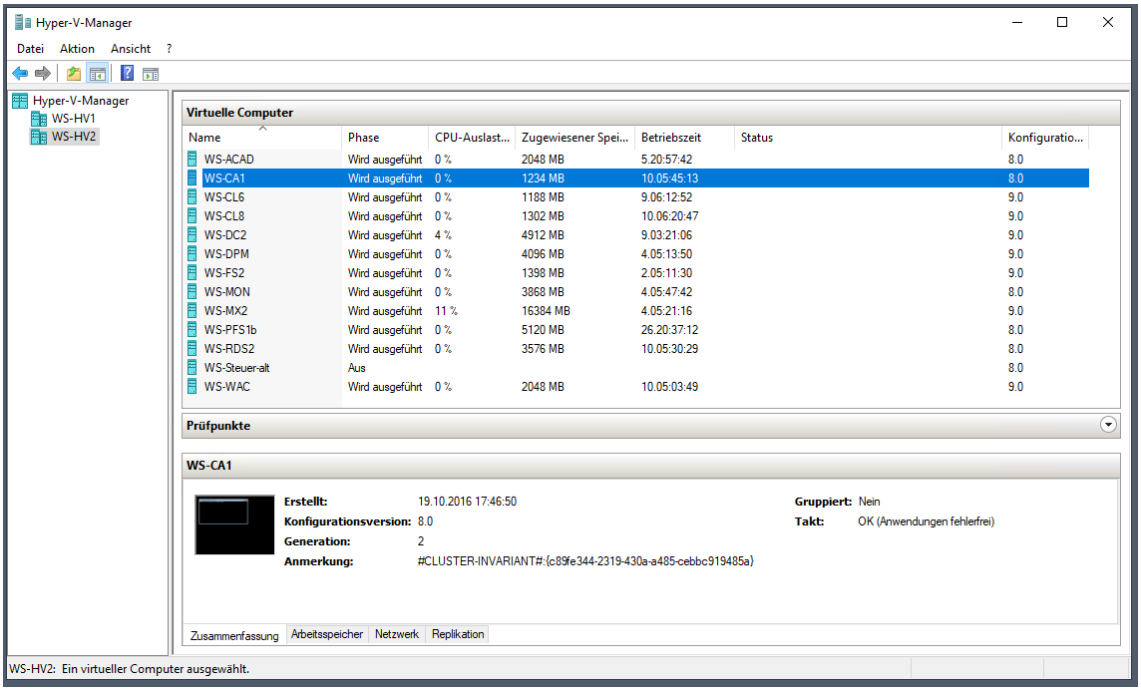
Mit PKIVIEW prüfe ich den Zustand der Verteilungspunkte. Hier ist alles ok:



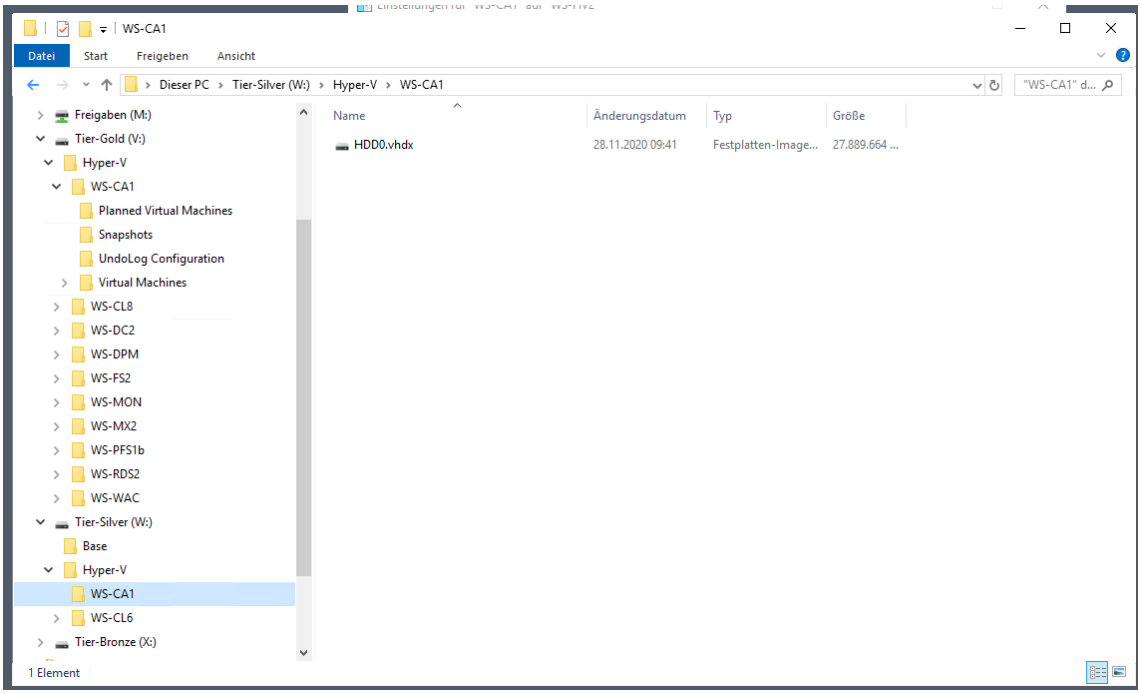
Das Verzeichnis c:\admin… enthält die Sperrlisten-Dateien:



Der Server wird als VM auf einem meiner Hyper-V-Server ausgeführt. Die Systemanforderungen sind überschaubar:



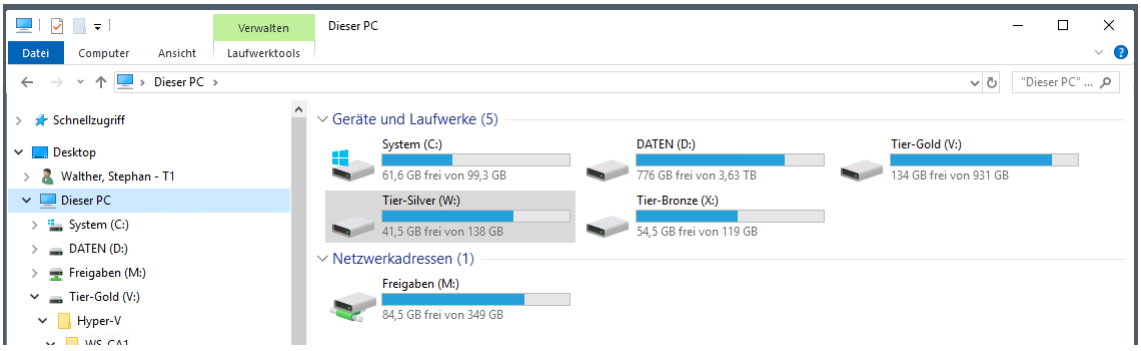
Die System-Festplatte ist recht klein geblieben:



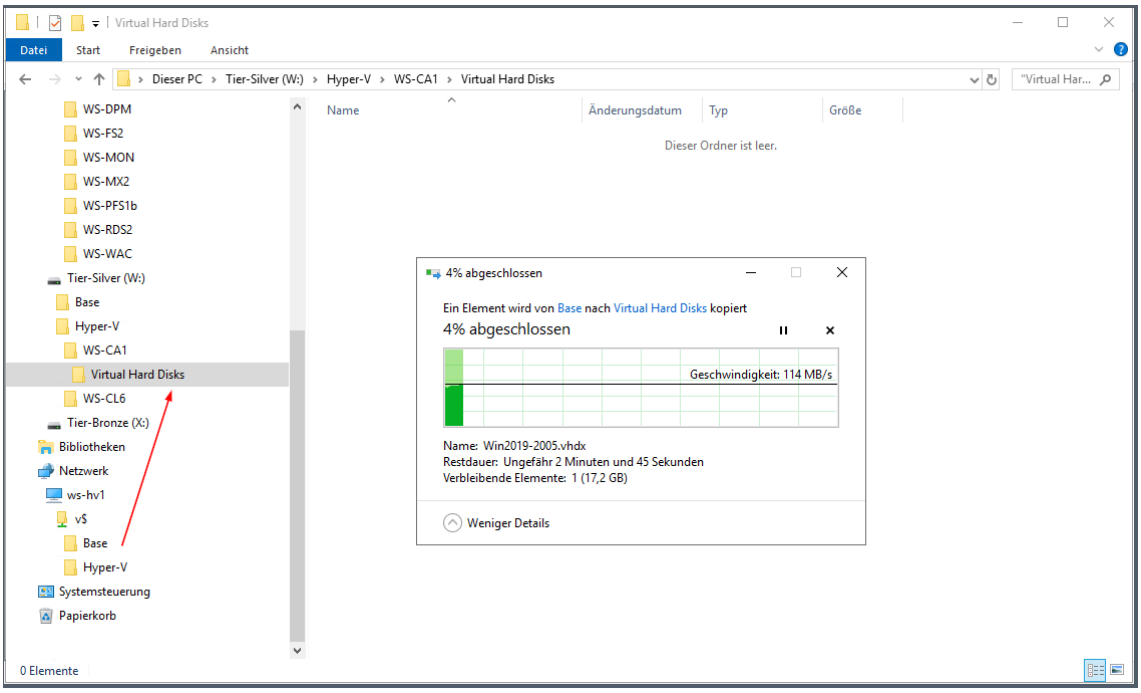
Es gibt keine geplanten Aufgaben.

Aufbau der neuen VM (mit Windows Update Problem)

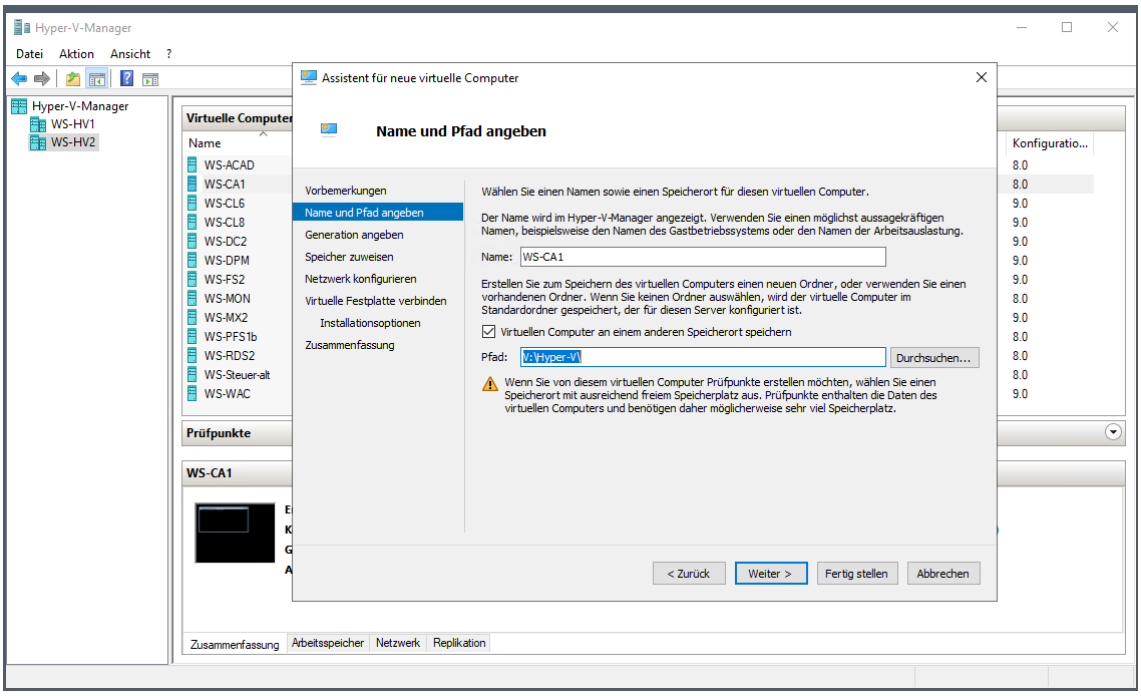
Auf meinem Hyper-V-Server ist noch ausreichend Platz für die neue VM:

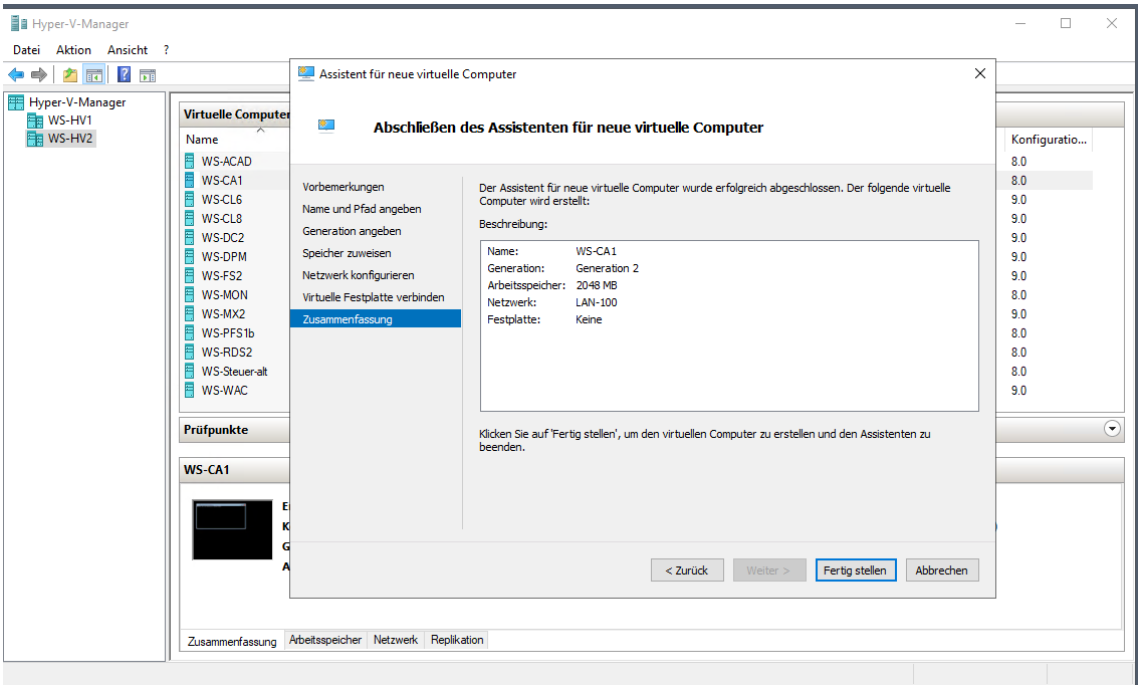


Ich kopiere mir ein Basefile mit Windows Server 2019 in das alte Verzeichnis:

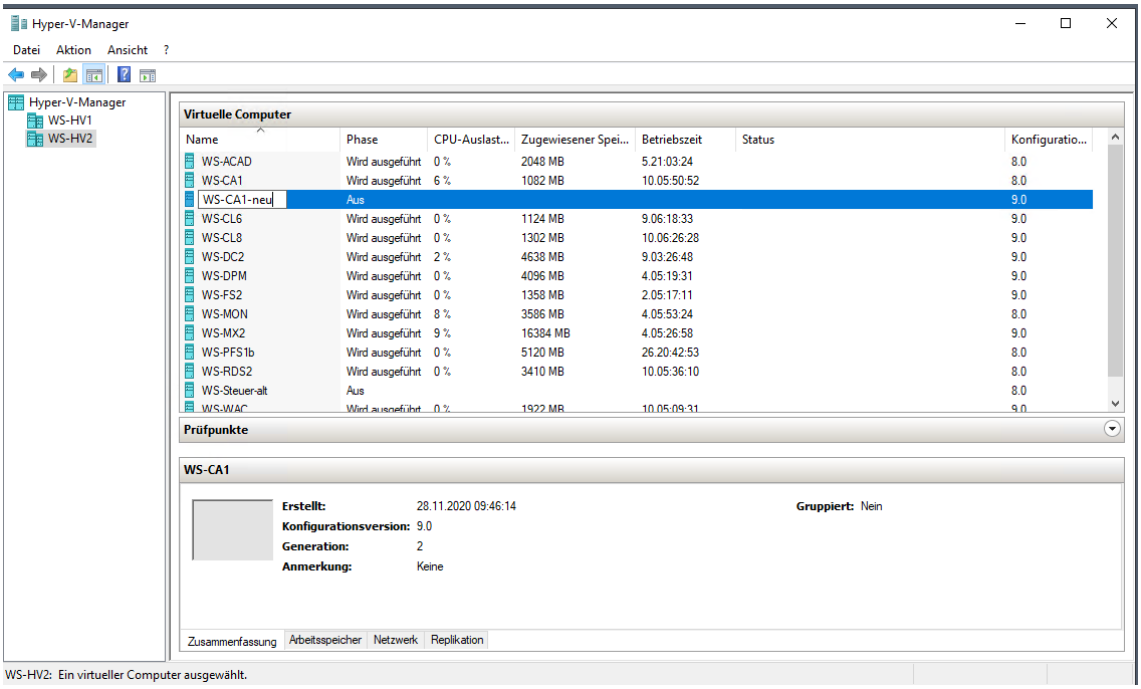


Dann erstelle ich eine neue VM mit dem gleichen Bezeichner im gleichen Verzeichnis:

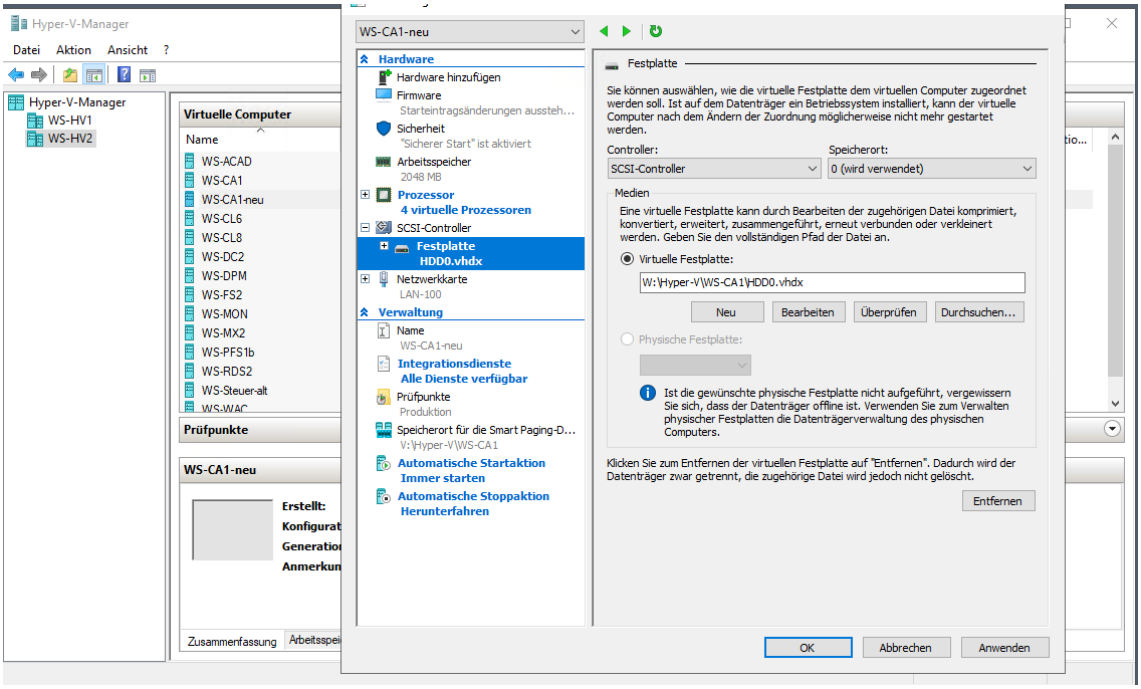


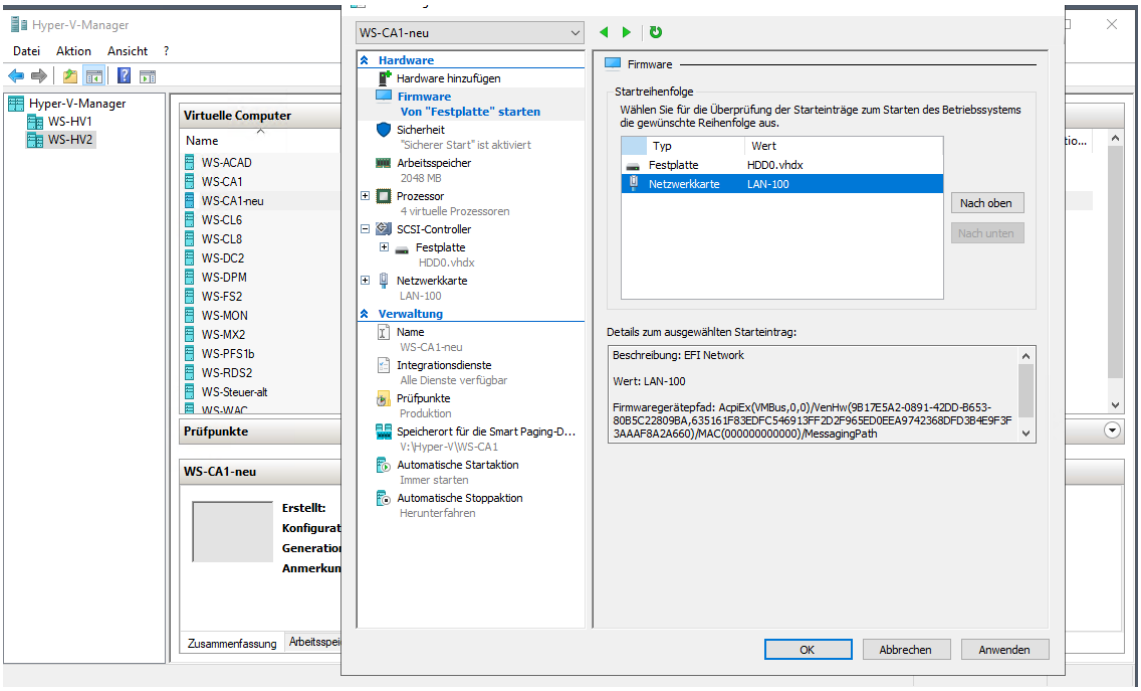


Damit ich nicht durcheinanderkomme, benenne ich den neuen Server um:

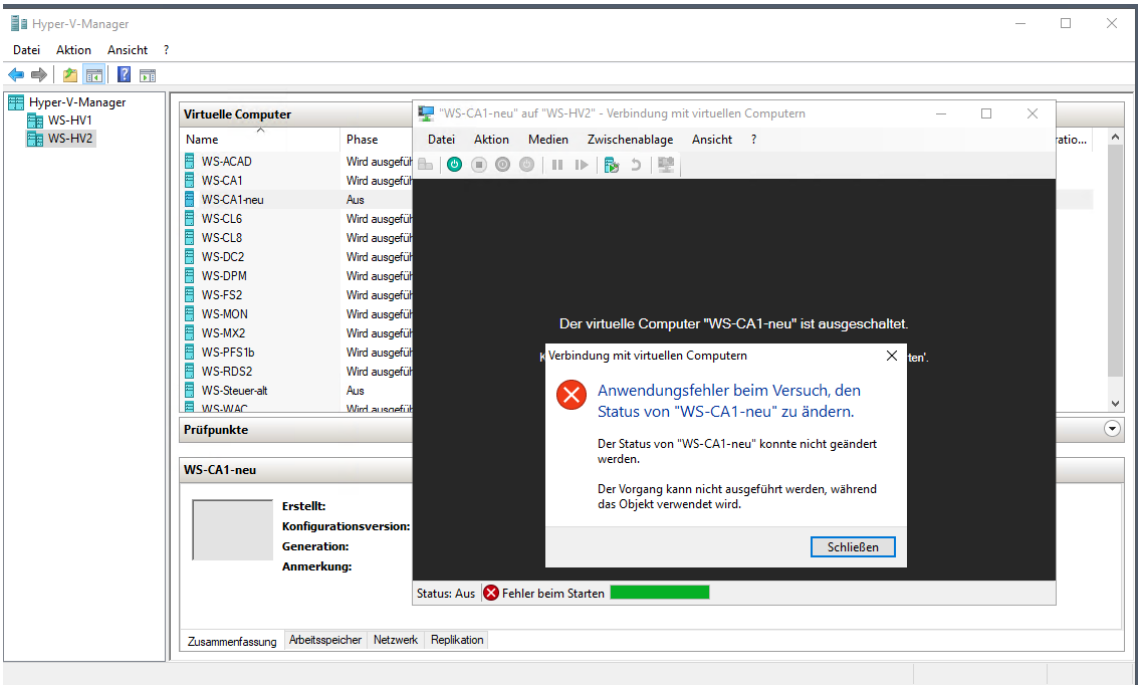


Jetzt passe ich noch einige Eigenschaften an und integriere die eben kopierte VHDX:

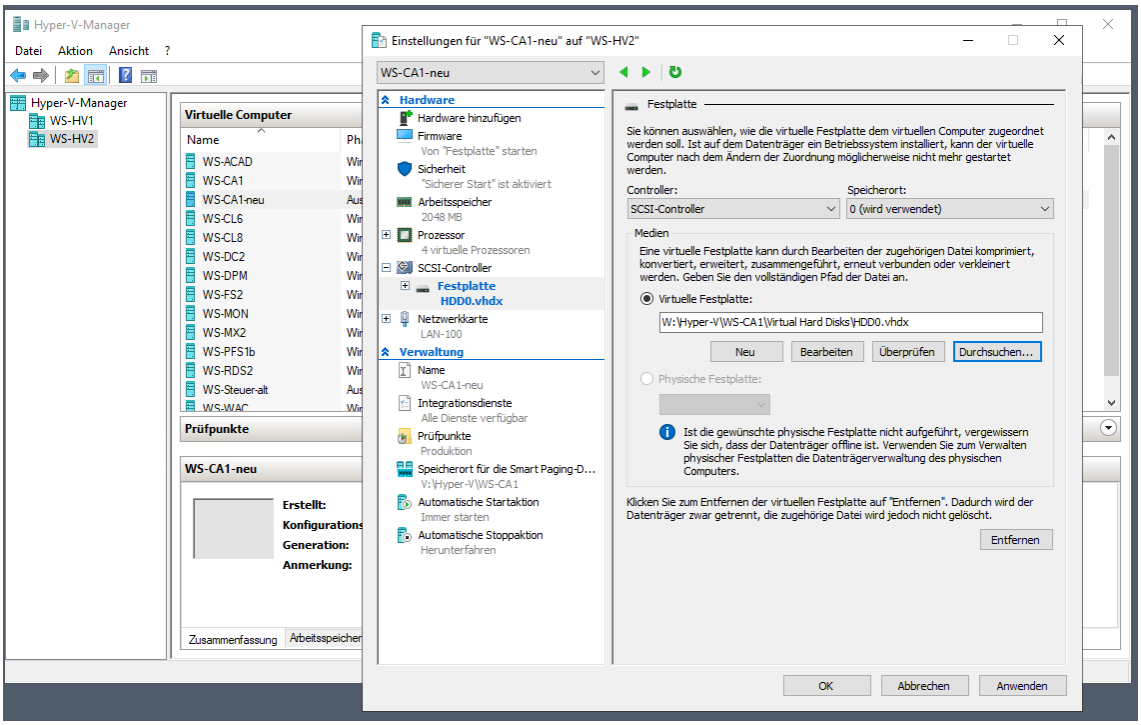




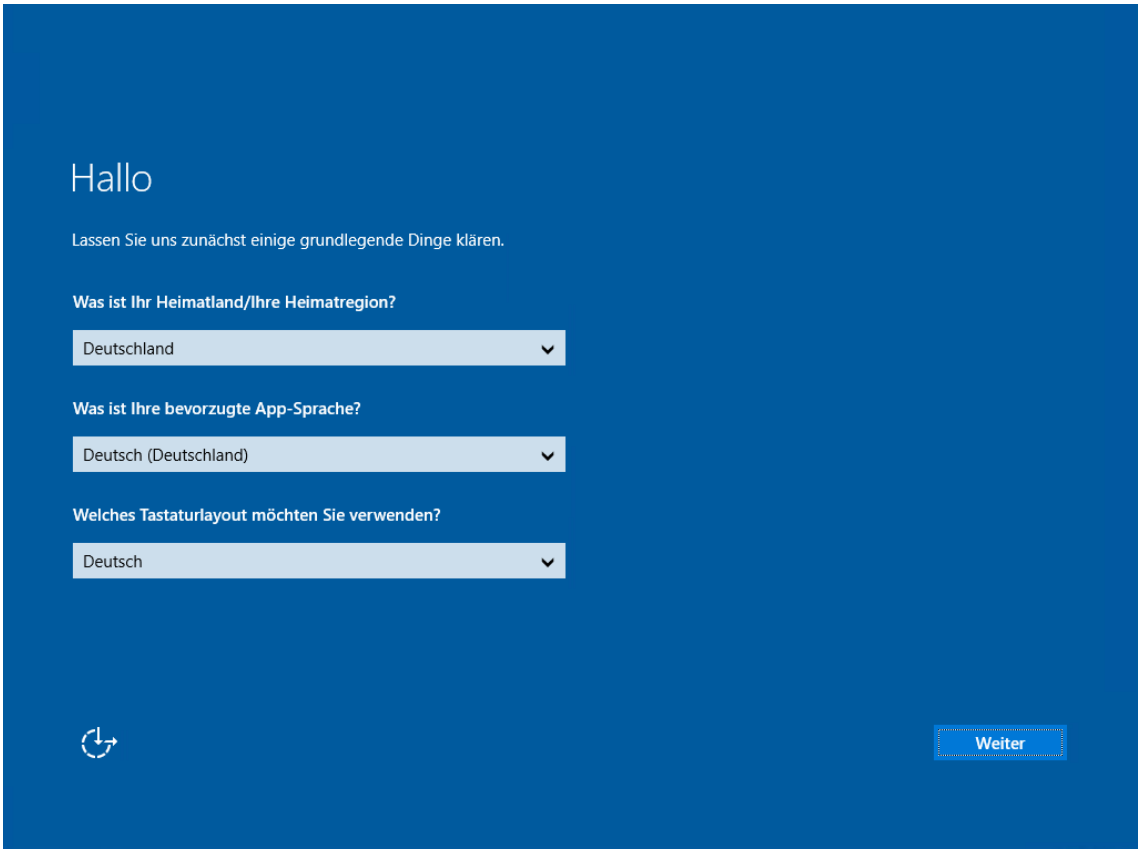
Nach dem Anpassen der Startreihenfolge kann es losgehen. Aber das System startet nicht:



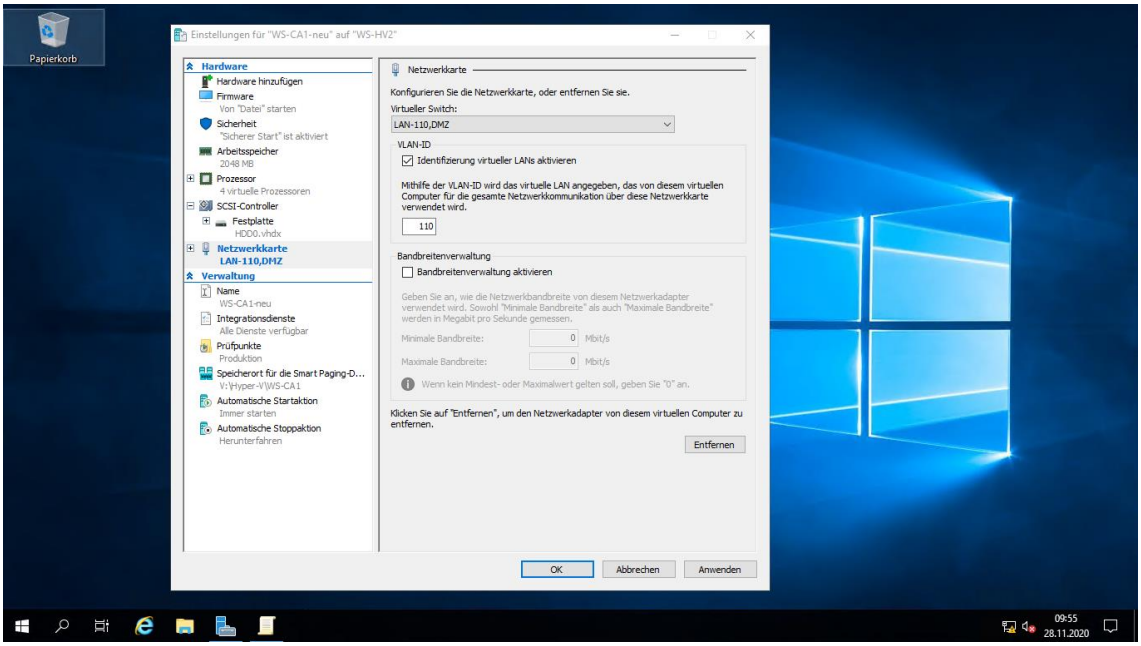
Ich hatte versehentlich die alte VHDX mit dem Windows Server 2016 zugewiesen. Aber die wird ja vom alten Server verwendet. Also passe ich den Pfad an:



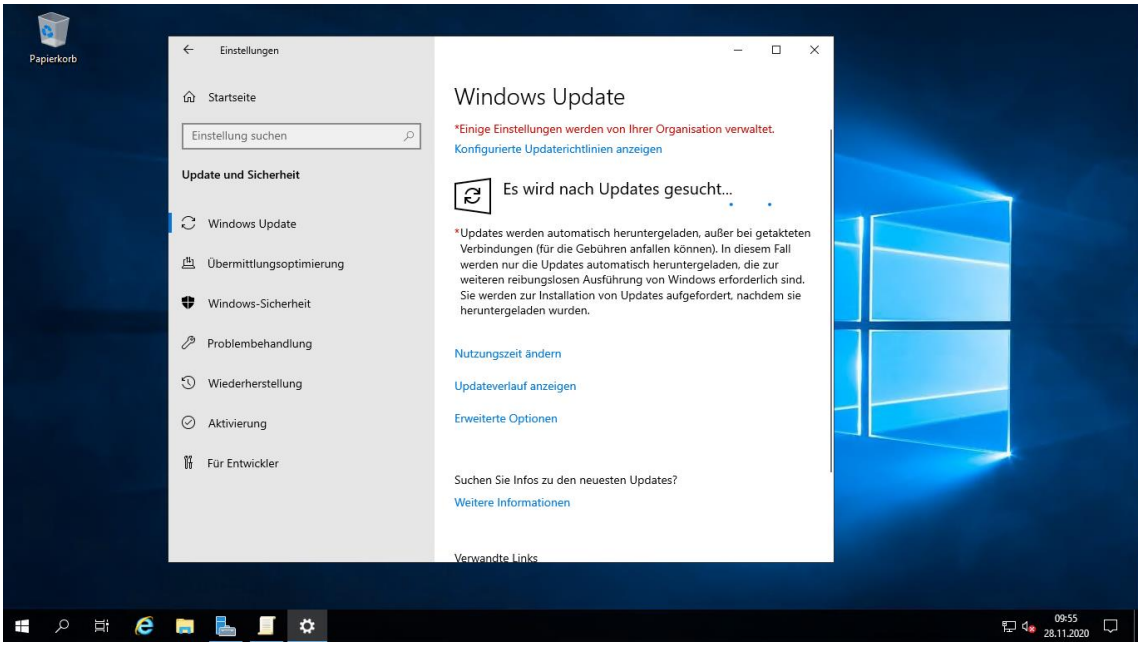
Jetzt startet das System. Weiter geht es im Out-Of-Box-Experience-Mode:



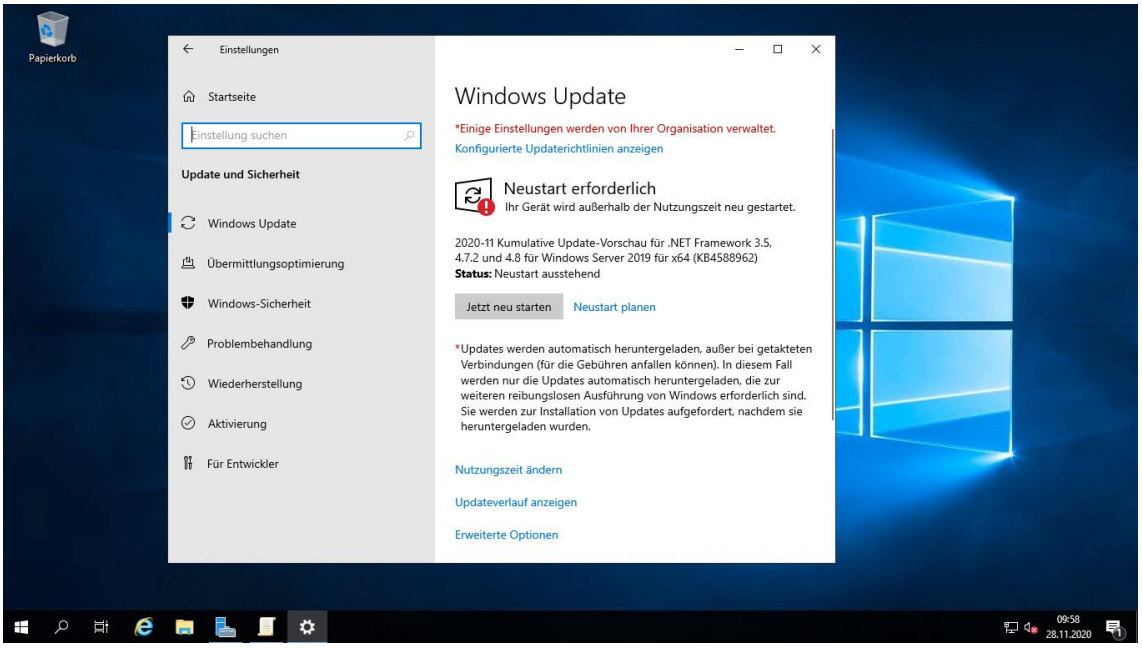
Den neuen Server klemme ich fix ins Client-VLAN:



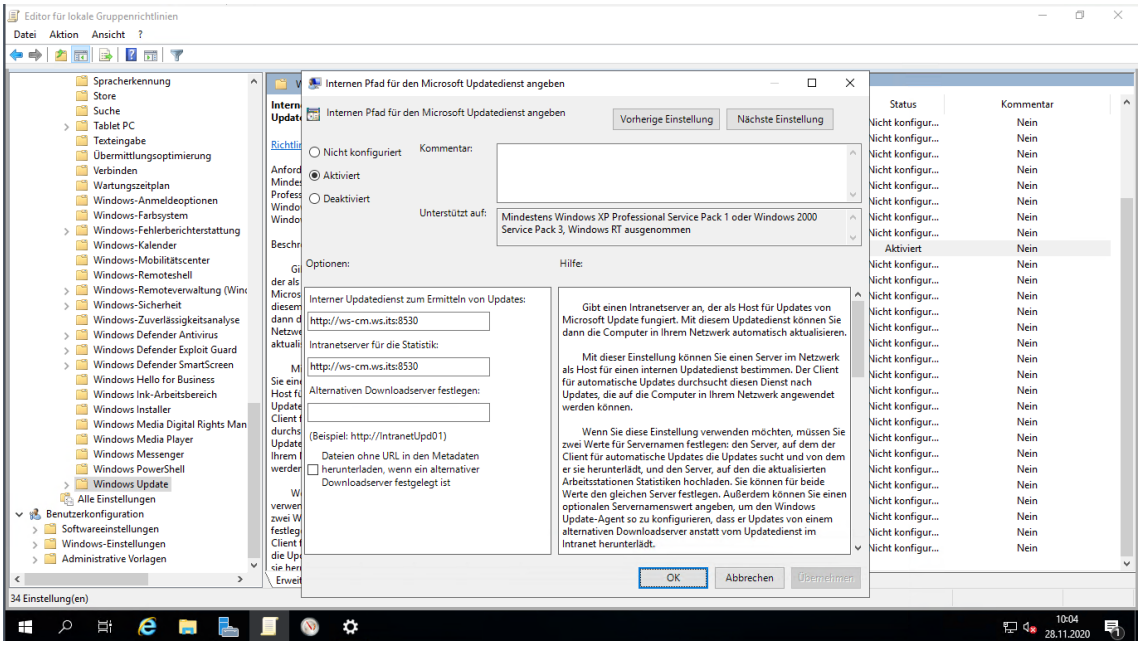
Hier kann er nach Updates im Internet suchen:



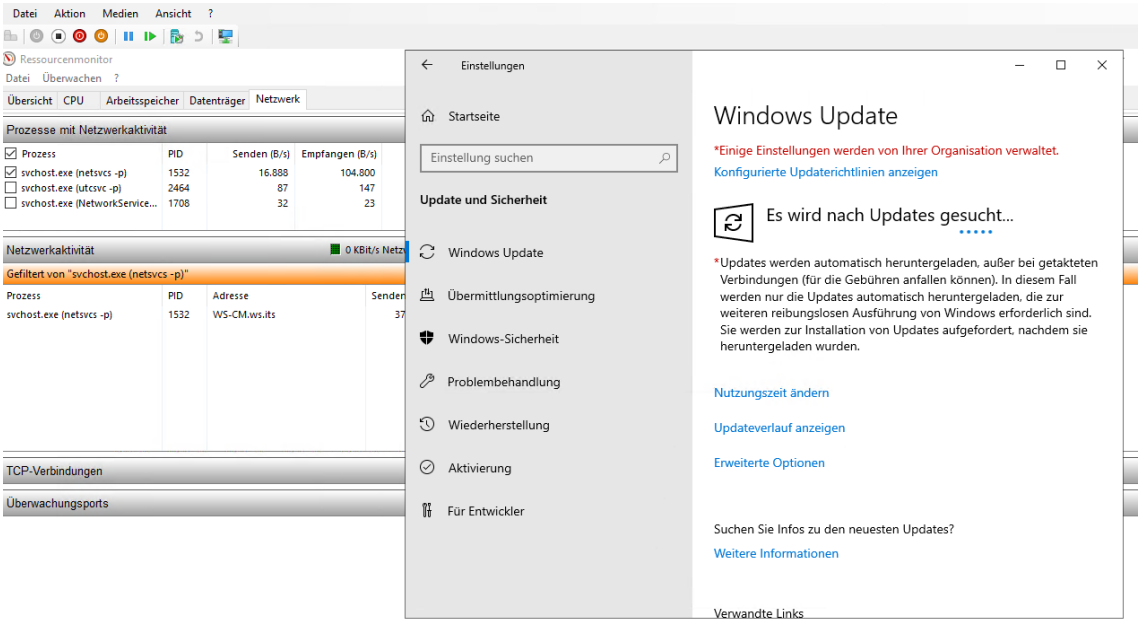
Danach wird der Neustart erforderlich:



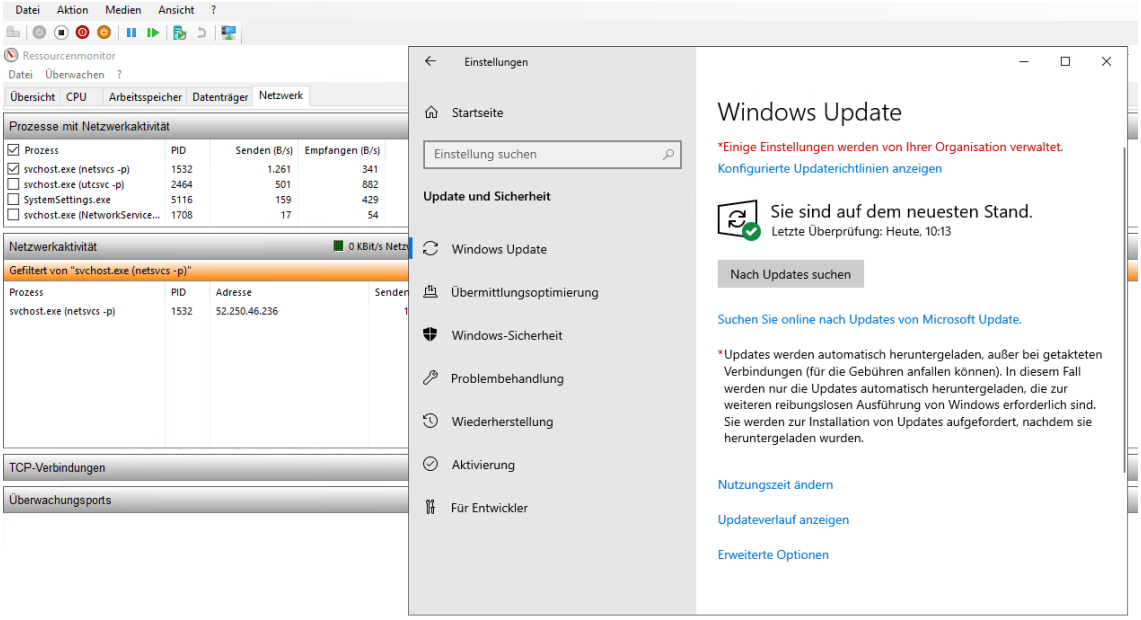
Ich bin aber etwas irritiert: Das Betriebssystem hatte ich aus einer VHDX-Datei mit dem Patchlevel 2020-05 erzeugt. Hier müssten wesentlich mehr Updates installiert werden! Ich passe den Pfad der Windows Updates an. So kann der Standalone-Server mit meinem WSUS kommunizieren:



Dann starte ich die Suche gegen meinen WSUS. Im Resmon sieht man, dass er mit dem WSUS kommuniziert:

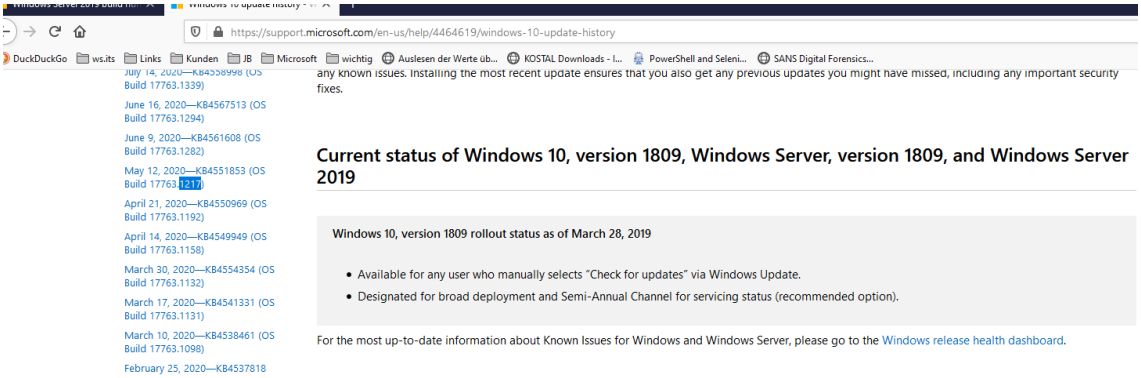


Aber auch hier gibt es keine Updates!



Dann schauen wir mal etwas genauer hin. Die Versionsnummer kann mit winver.exe ermittelt werden:

Mit dieser Info suche ich im Netz nach dem Patch-Level. Es ist wie erwartet 2020-05:

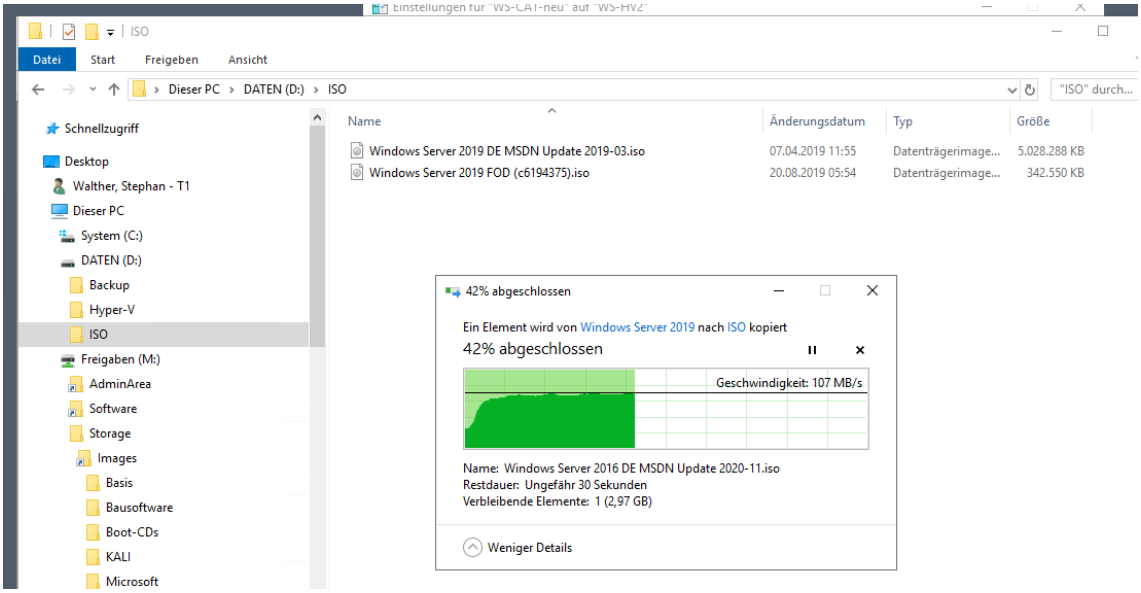


Offensichtlich stimmt hier was nicht! Vielleicht fehlen Voraussetzungen für die aktuelleren Updates. Vielleicht ist der Cache auf dem Server beschädigt. Aber da habe ich bei einem neuen Server keine Lust. Ich installiere lieber neu, denn der neue Server soll ja auch einige Jahre problemfrei laufen!

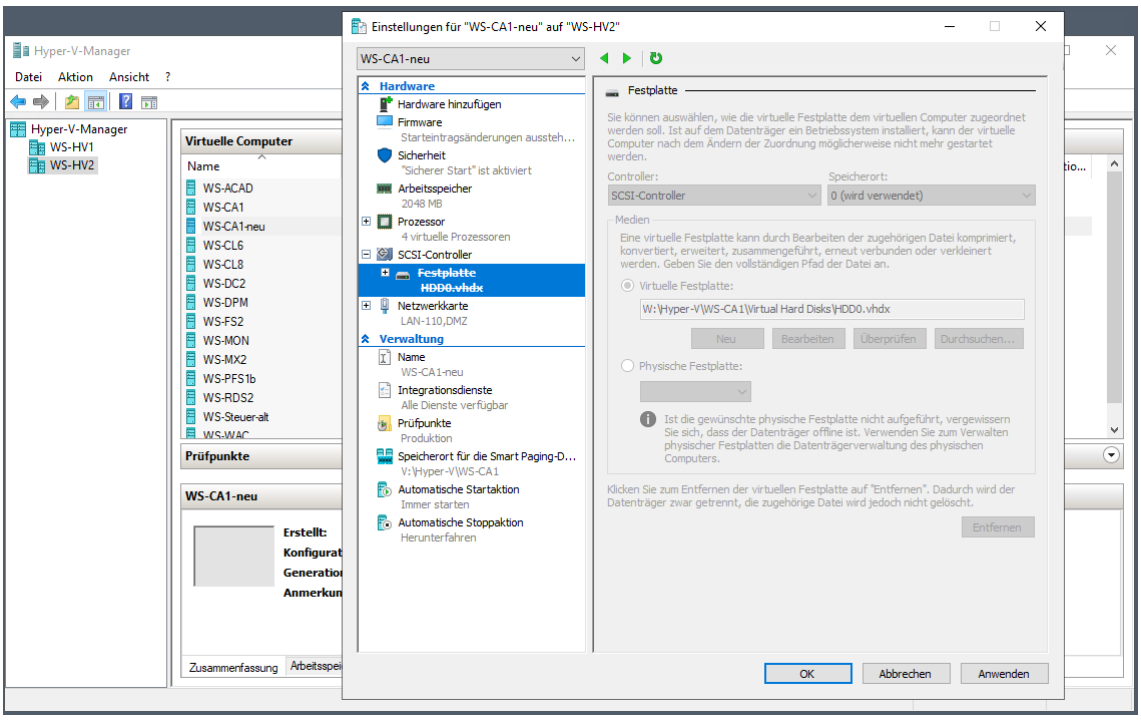
Verlasst euch bitte nicht auf den Update-Dialog. Der kann trügerisch sein!

Aufbau der neuen VM (Neuinstallation)

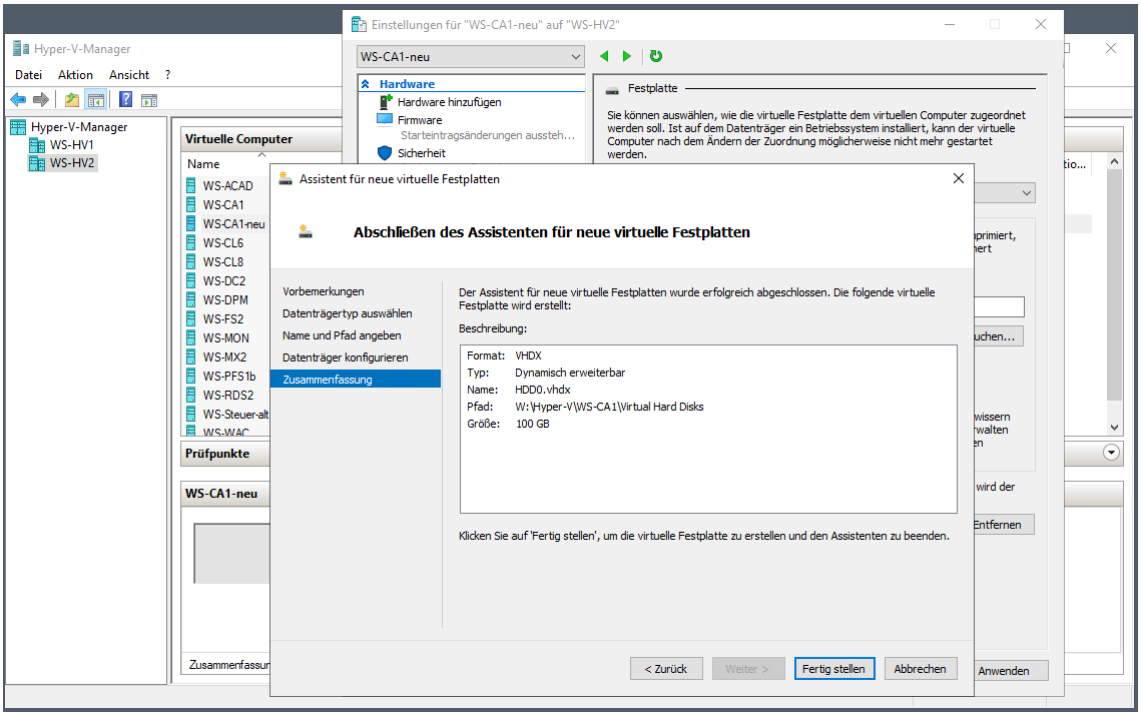
Microsoft stellt in unregelmäßigen Abständen neue ISOs mit integrierten Updates bereit. Ich lade das aktuellste mit dem Patchlevel 2020-11 herunter und kopiere es auf meinen Hyper-V-Server:

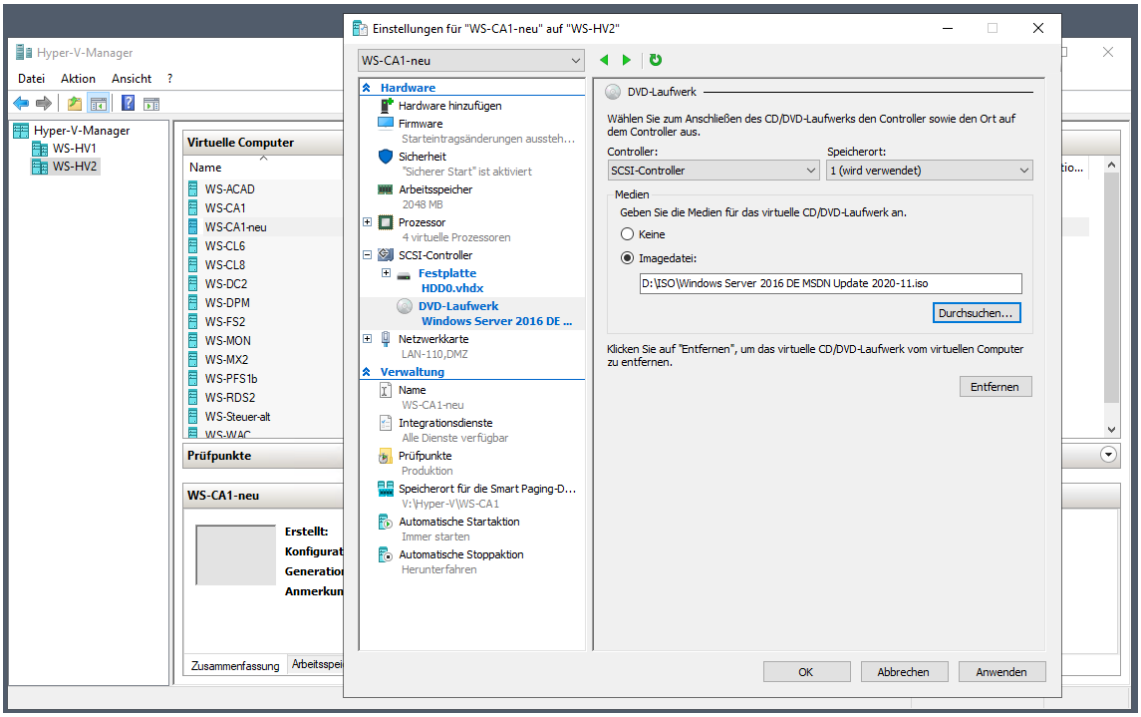


Die defekte VHDX entferne ich:

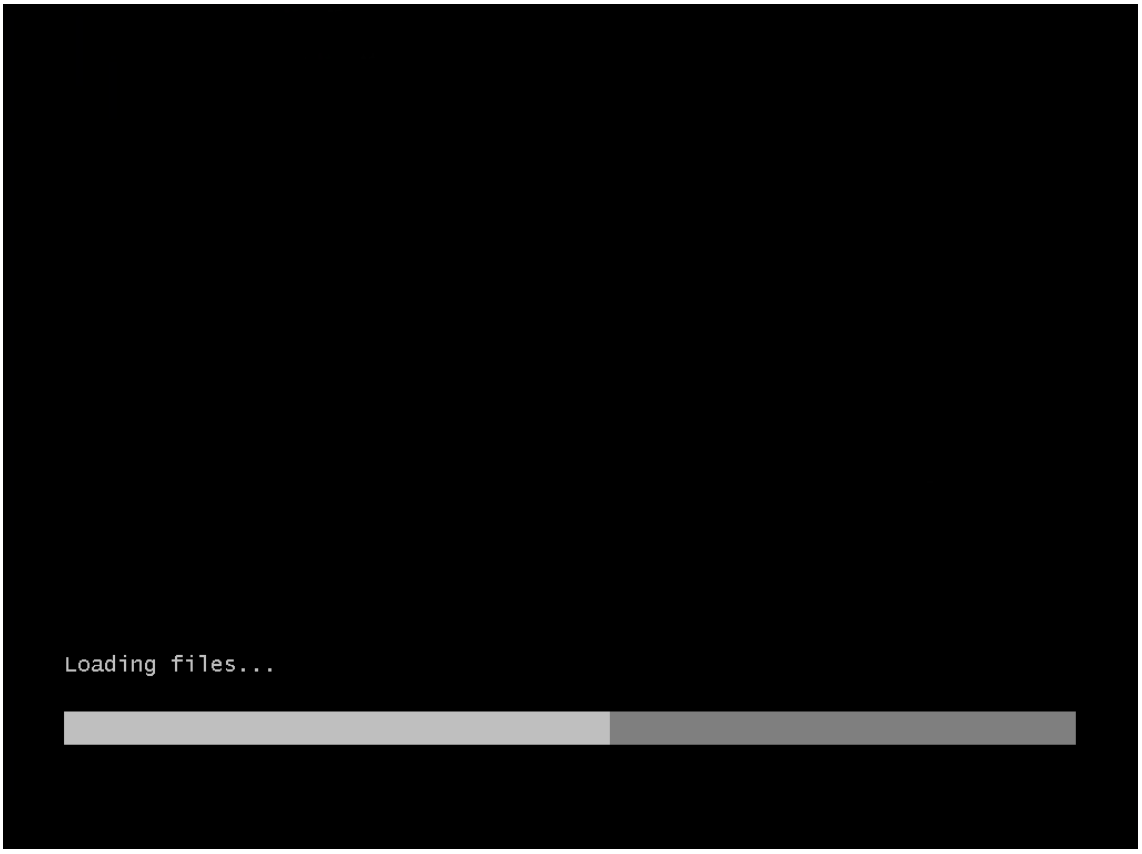


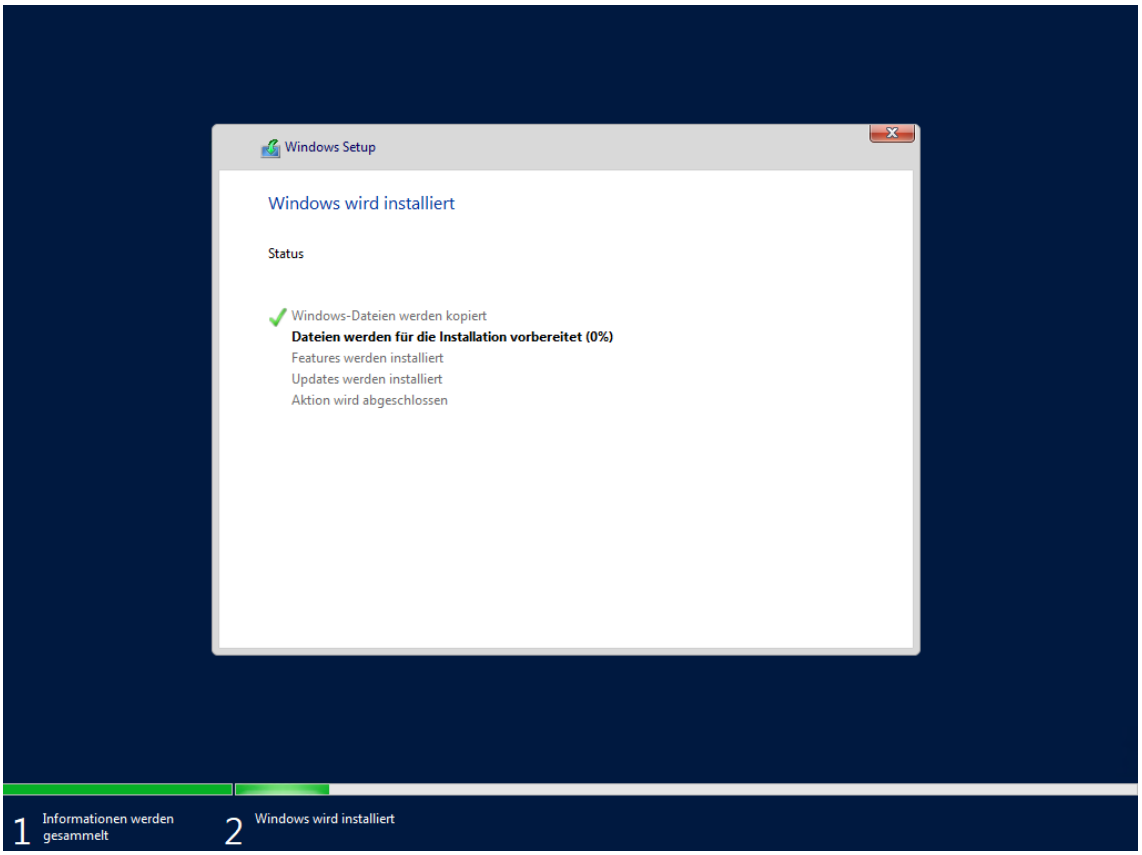
Dann erstelle ich eine leere VHDX:



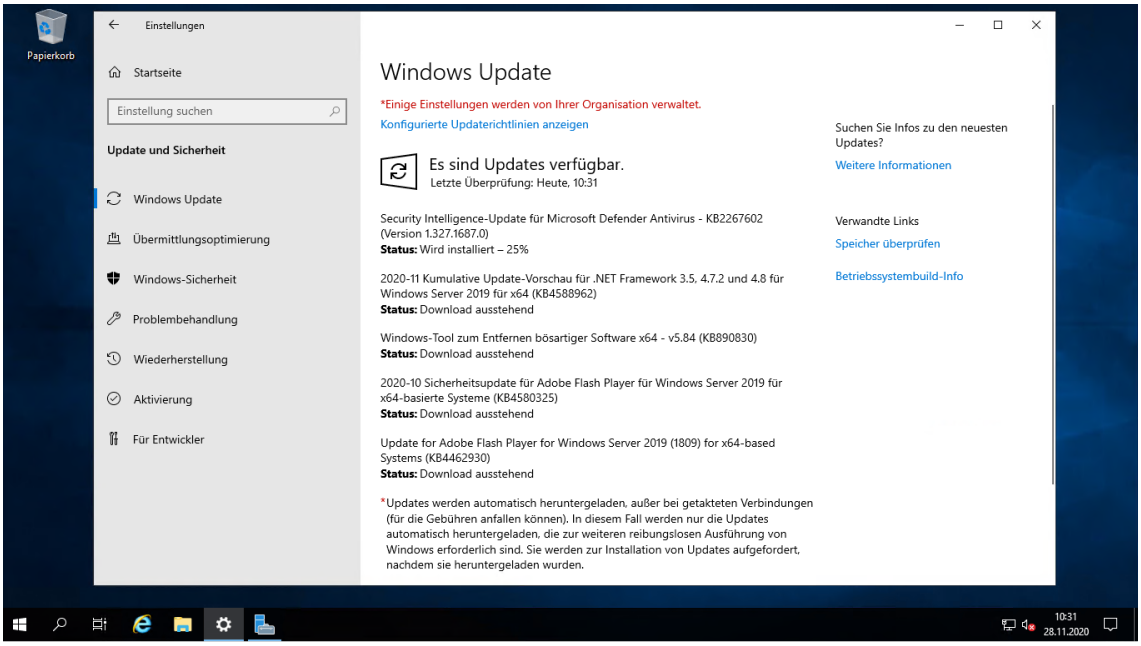


Mit dem ISO und der leeren Festplatte wird die Installation gestartet:

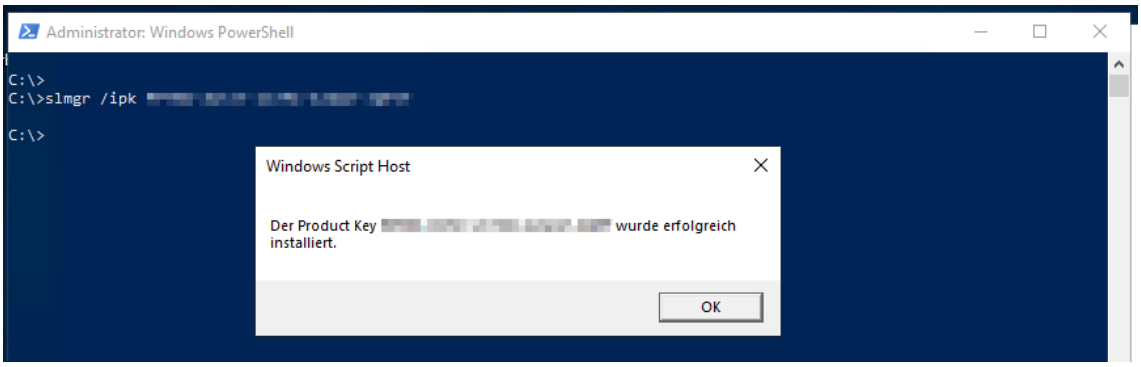


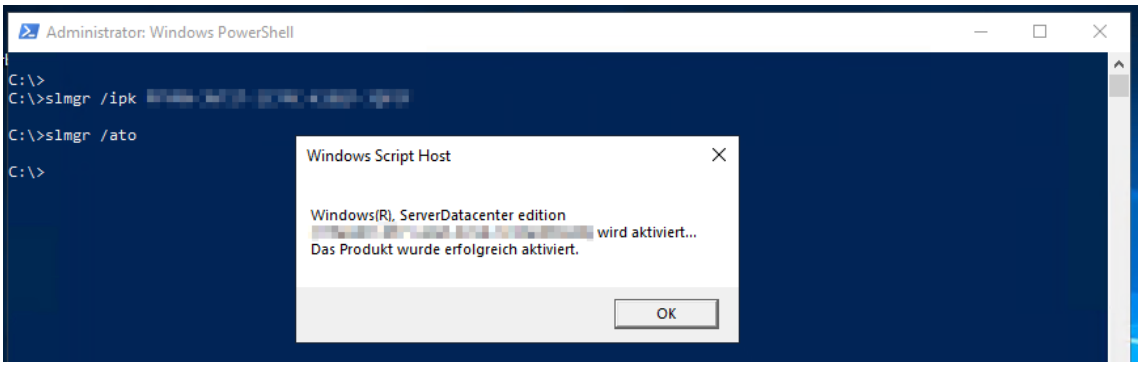


Nach dem Start des neuen Windows Servers suche ich nach Updates. Das sieht viel besser aus:



So macht nun auch eine Aktivierung Sinn:

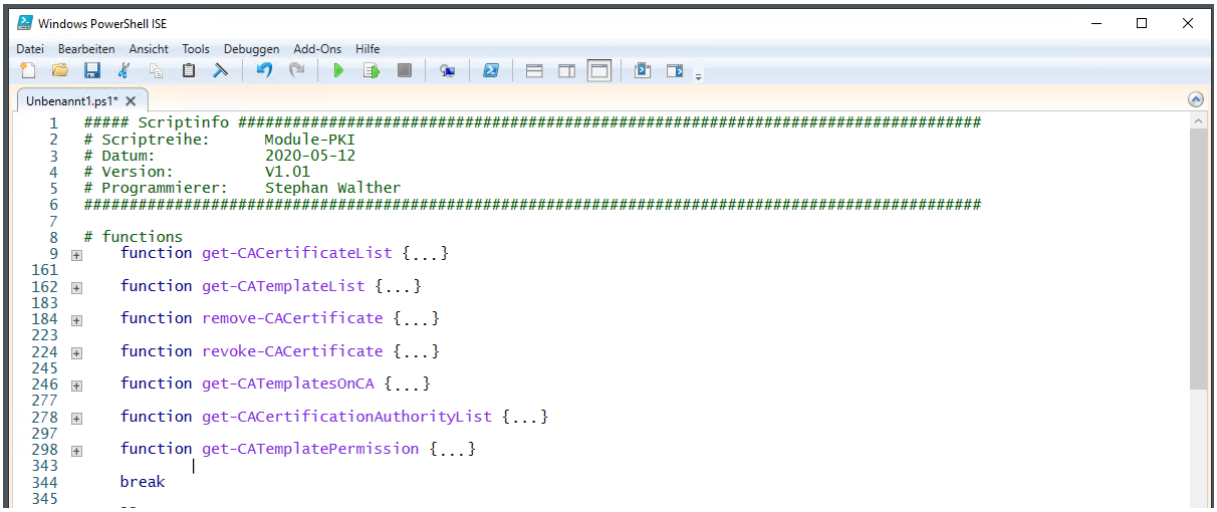




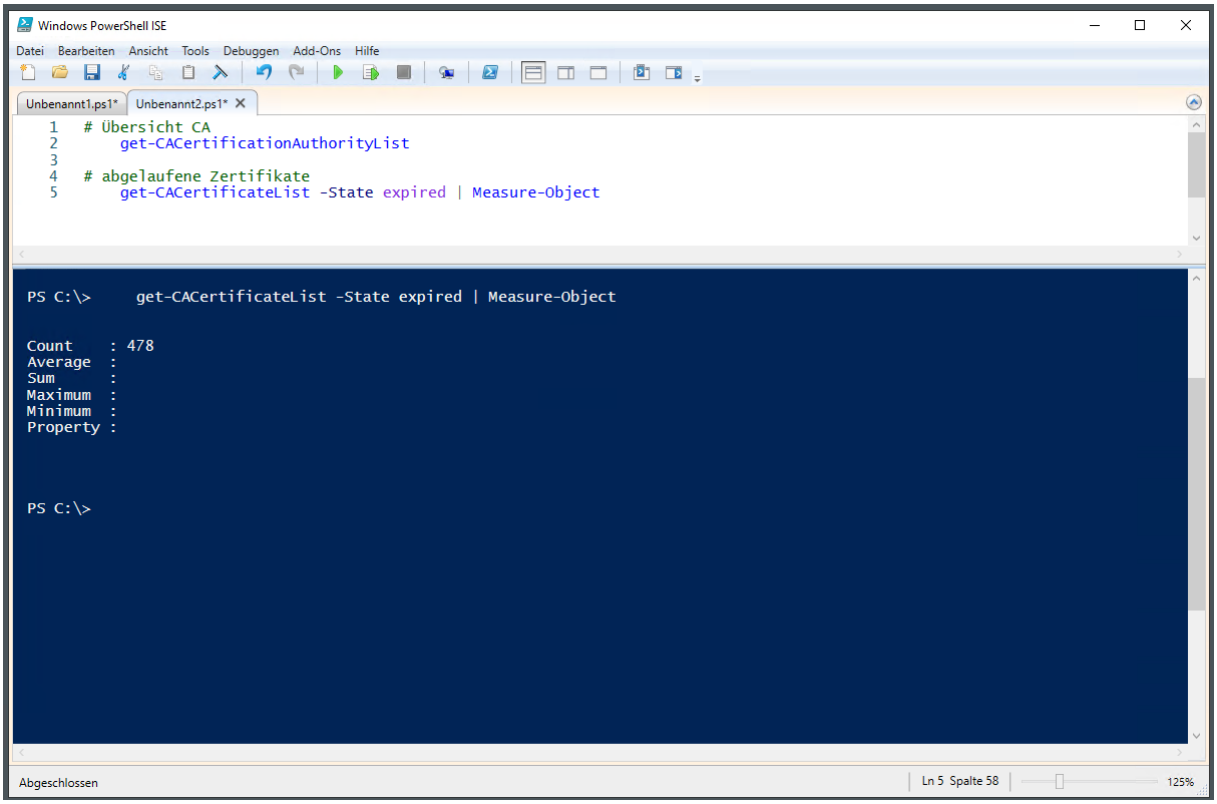
Damit ist der neue Server vorbereitet.

CleanUp PKI

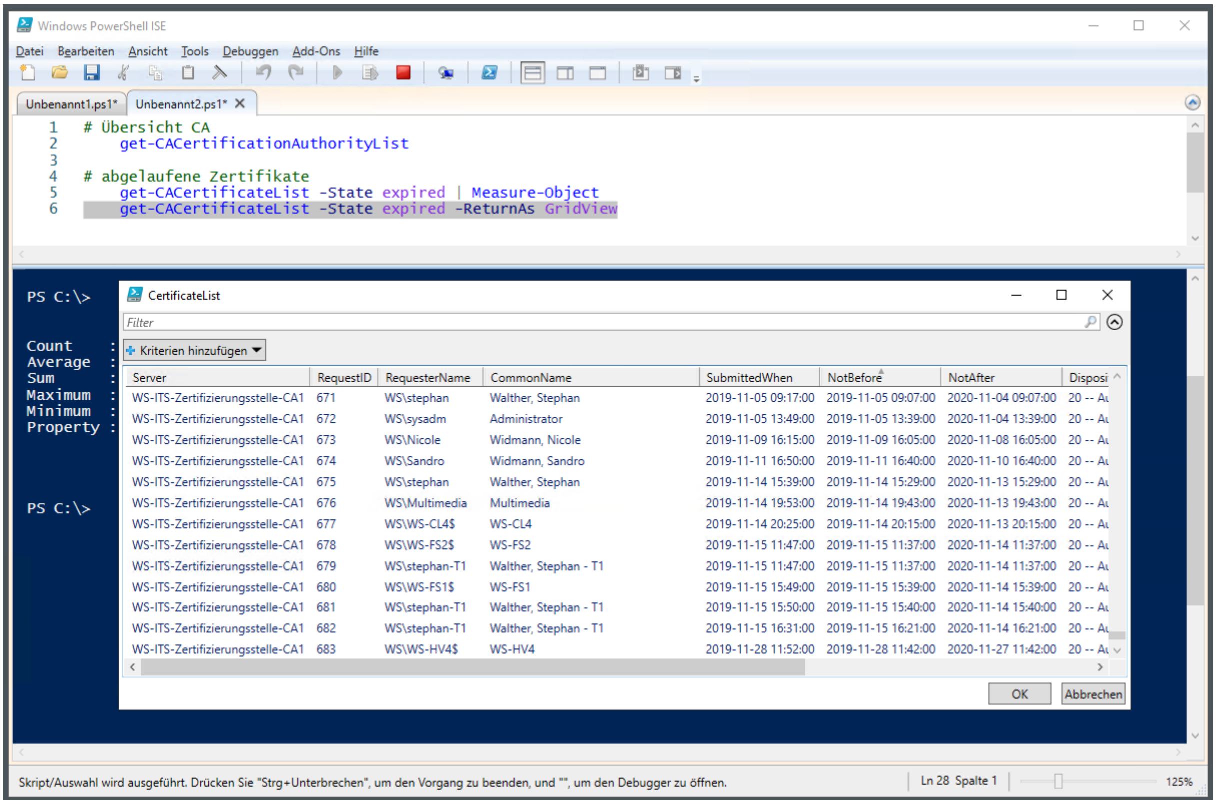
Weiter geht es mit dem Aufräumen in der alten CA. Die grafische Oberfläche bietet hier nur begrenzte Möglichkeiten. Und mit dem Befehl certutil.exe ist das Arbeiten auch nicht immer einfach. Daher hatte ich mir vor einigen Monaten einige PowerShell-Funktionen erstellt. Diese möchte ich heute verwenden:



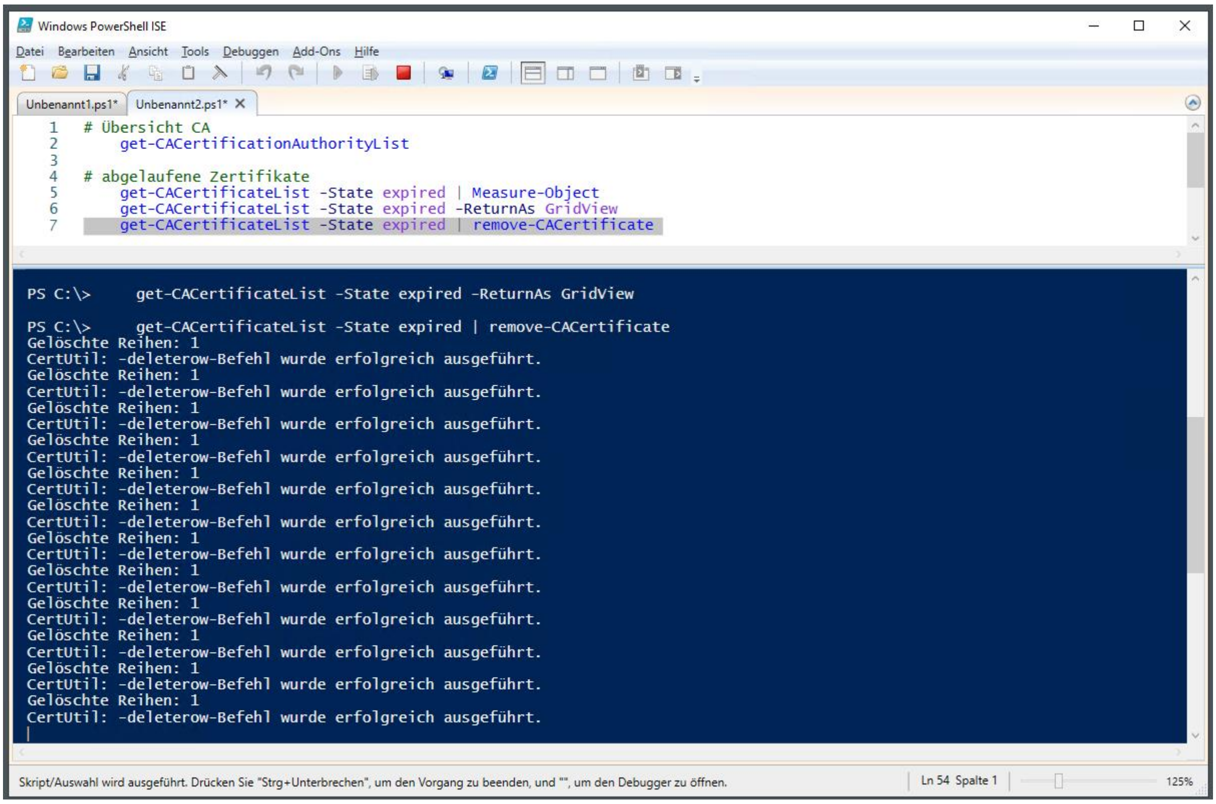
Aktuell sind 478 abgelaufene Zertifikate in der Datenbank der alten CA. Mit meinen PowerShell-Funktionen ist diese Ermittlung echt einfach:



Die Ausgabe kann auch in ein Gridview-Steuerelement umgelenkt werden. So kann ich also einfach und gezielt nach Zertifikaten suchen:

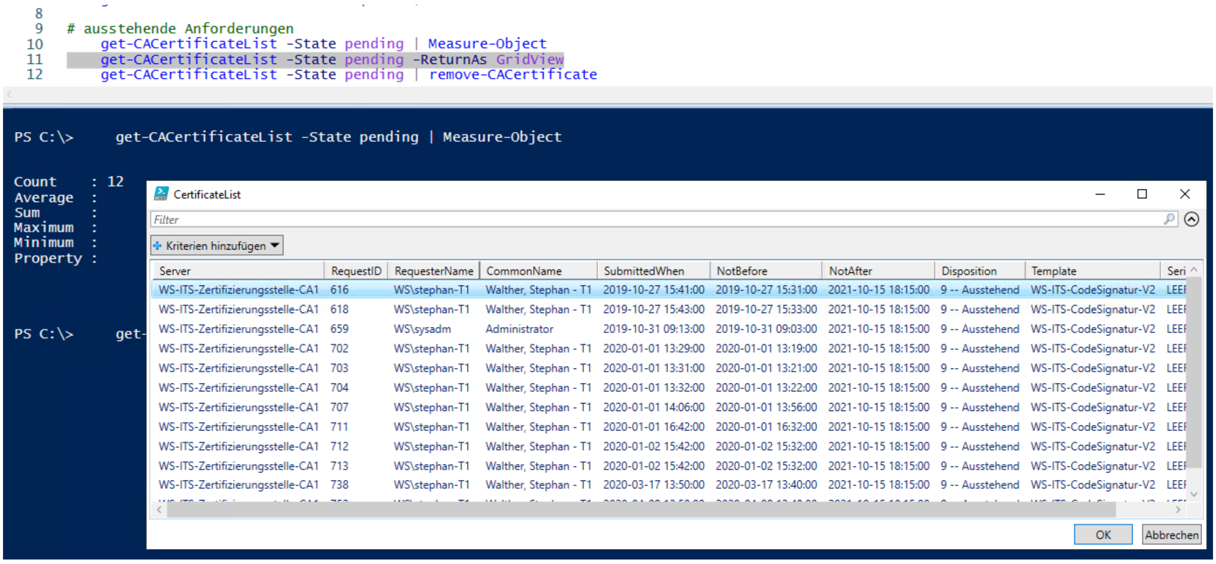


Meine Funktionen sind Pipeline-fähig. So kann ich mir das Ergebnis einer Suche nicht nur ansehen, sondern auch weiterverarbeiten. Hier suche und lösche ich abgelaufene Zertifikate in einer Zeile:

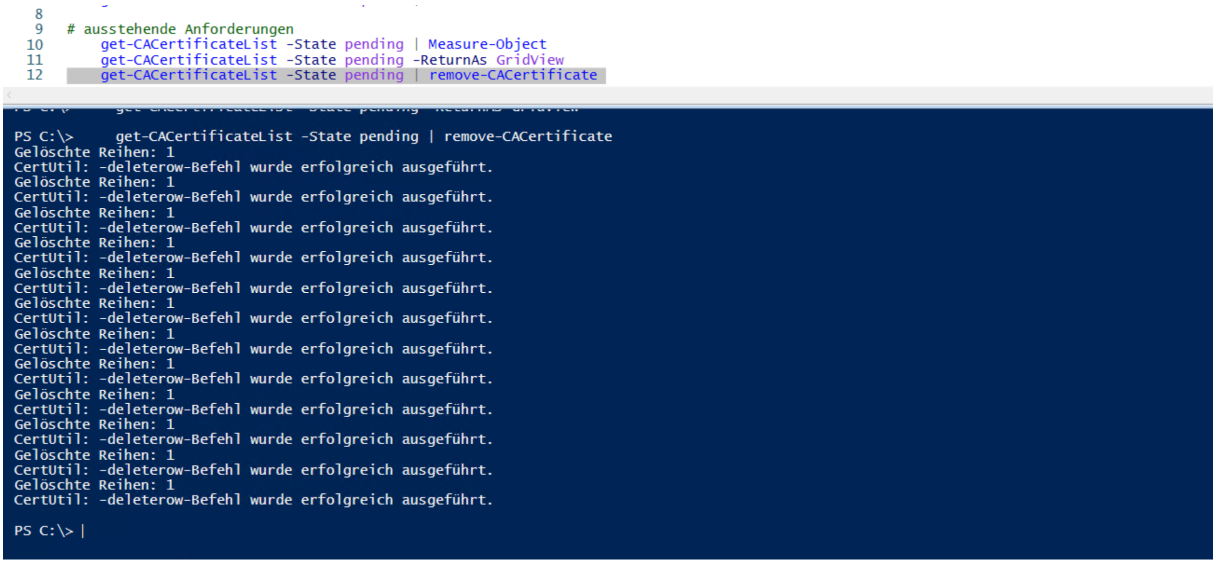


Dabei werden die Zertifikate nacheinander durch certutil-Aufrufe gelöscht. Das kann etwas dauern.

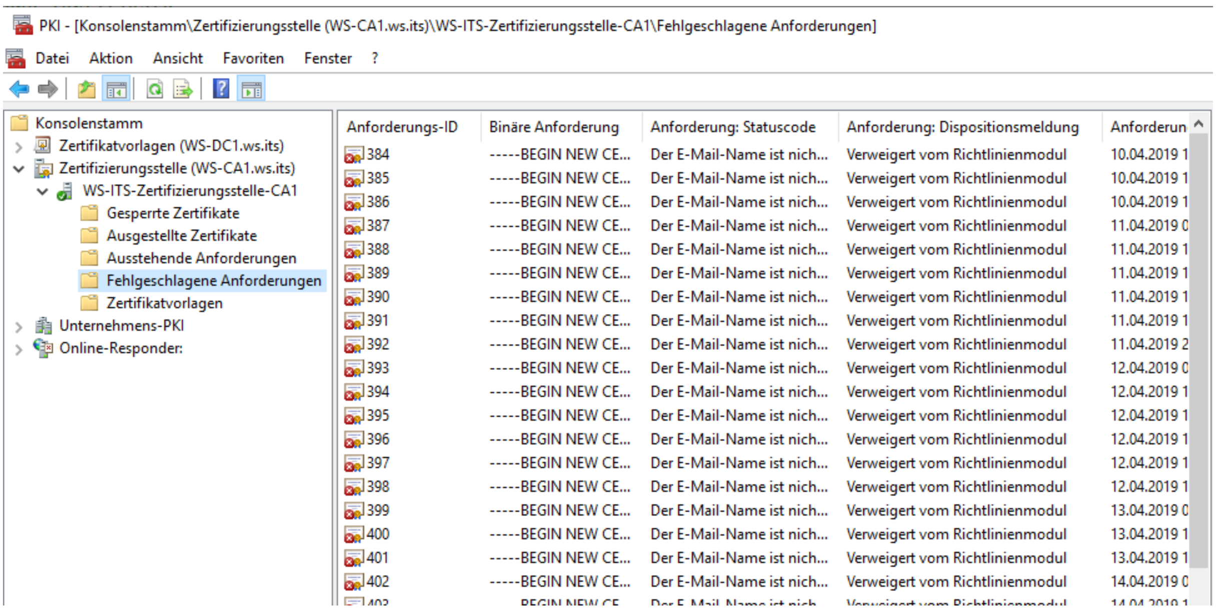
Weiter geht es mit ausstehenden Anfragen, die nie abgeschlossen wurden. Das sind nicht so viele:



Aber auch diese benötige ich nicht länger:



Es wird Zeit für einen Blick in die grafische Oberfläche. Hier stehen noch etliche fehlgeschlagenen Requests:



Aber auch diese kann ich mit der PowerShell-Funktion suchen und löschen:

```
19  # abgelehnte Anforderungen
20      get-CACertificateList -State denied | Measure-Object
21      get-CACertificateList -State denied -ReturnAs GridView
22      get-CACertificateList -State denied | remove-CACertificate
23
```

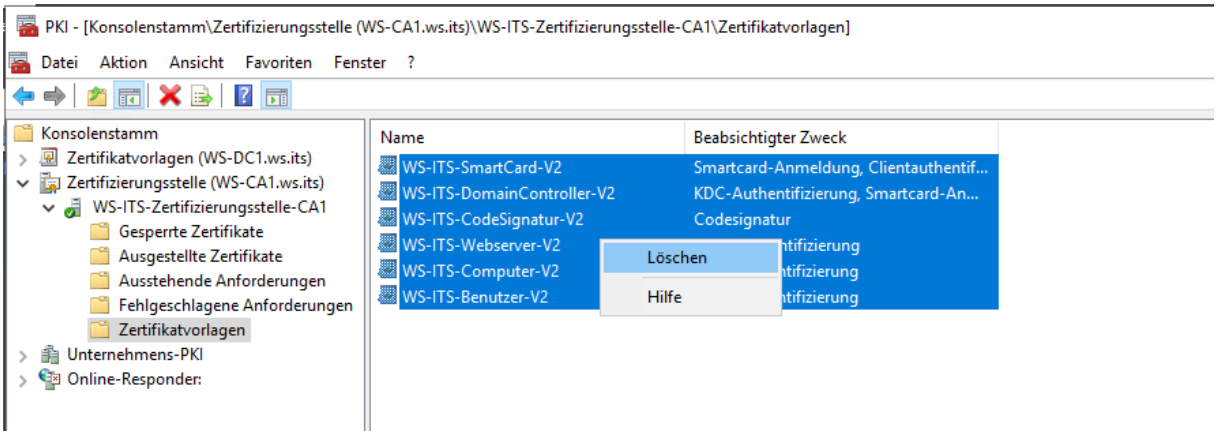
```
CertUtil: -deleterow-Befehl wurde erfolgreich ausgeführt.
Gelöschte Reihen: 1
CertUtil: -deleterow-Befehl wurde erfolgreich ausgeführt.
Gelöschte Reihen: 1
CertUtil: -deleterow-Befehl wurde erfolgreich ausgeführt.
Gelöschte Reihen: 1
CertUtil: -deleterow-Befehl wurde erfolgreich ausgeführt.
Gelöschte Reihen: 1
CertUtil: -deleterow-Befehl wurde erfolgreich ausgeführt.
Gelöschte Reihen: 1
CertUtil: -deleterow-Befehl wurde erfolgreich ausgeführt.

PS C:\Users\stephan-t1>
```

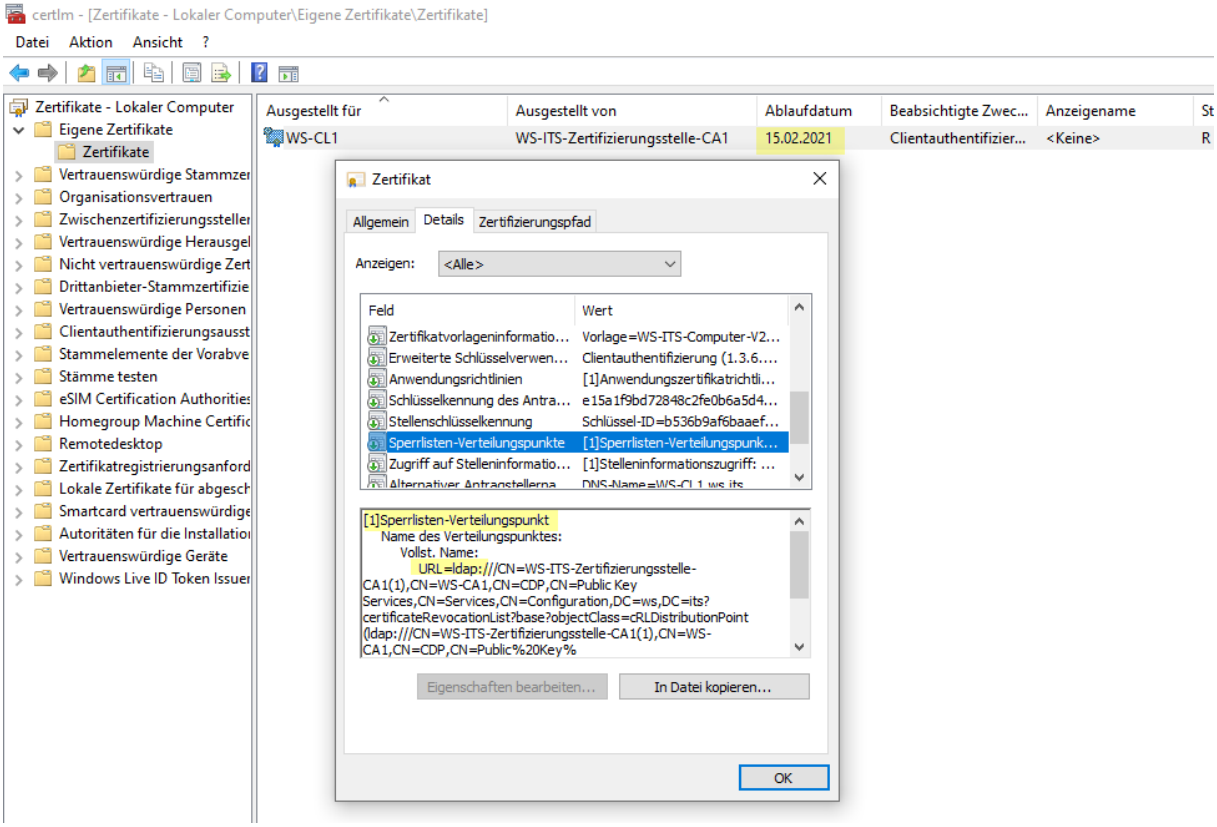
Anpassungen an der PKI

Weiter geht es mit Anpassungen an der PKI. Ab jetzt soll die alte CA keine Zertifikate mehr ausstellen. Damit beginnt also ein Wartungszeitfenster.

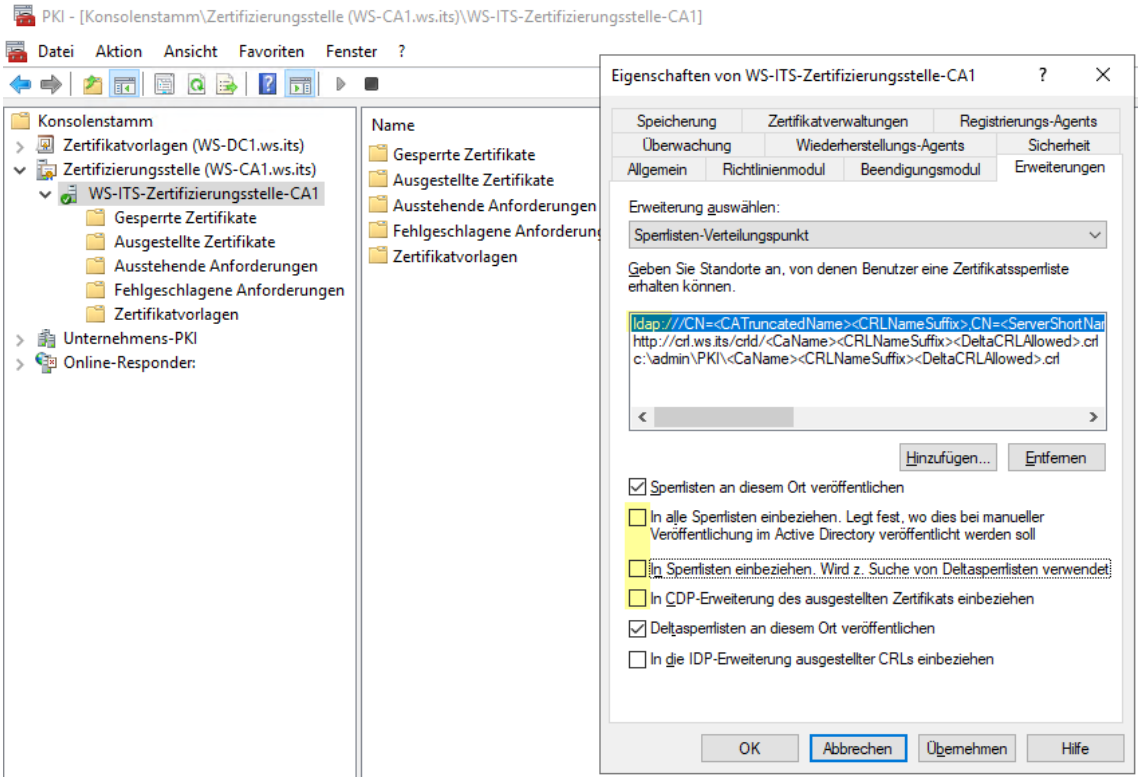
Ich entferne die auszustellenden Vorlagen. Die Vorlagen sind nur Verknüpfungen. Die Definitionen bleiben erhalten und ich kann die Vorlagen jederzeit wieder hinzufügen:



Wenn keine auszustellenden Vorlagen mehr vorhanden sind, dann werden neue Requests abgelehnt. So kann ich nun die Anpassungen eintragen. Ich beginne mit den Sperrlisten-Verteilungspunkten. Zukünftig sollen Sperrlisten nicht mehr über LDAP verteilt werden. Ich kann aber nicht einfach den Record löschen, denn hier hängt nicht nur die Veröffentlichung in den ausgestellten Zertifikaten dran, sondern auch die Veröffentlichung der Sperrlisten selber. In bisher ausgestellten Zertifikaten steht also drin, dass das vorliegende Zertifikat über eine Sperrliste aus dem Active Directory geprüft werden kann:

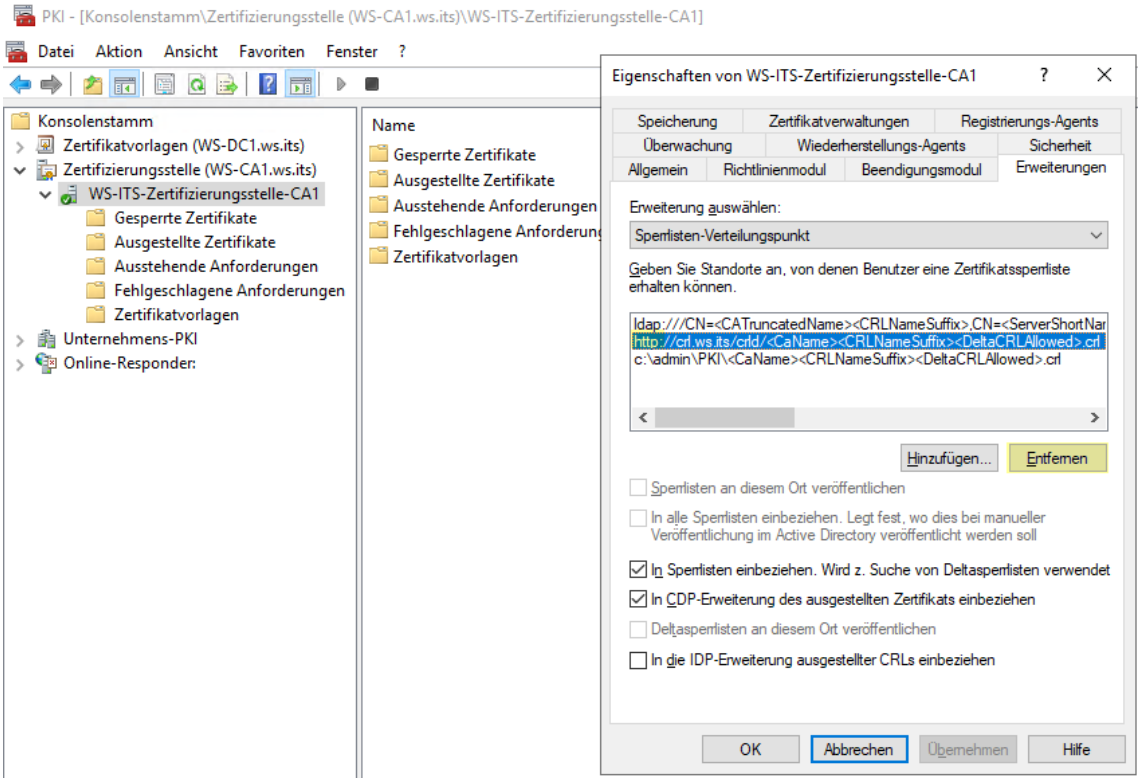


Wenn ich nun den LDAP-Record komplett entferne, dann wird dieser Eintrag in neuen Zertifikaten nicht mehr vorhanden sein. Gleichzeitig wird die CA aber auch keine aktuellen Sperrlisten im AD veröffentlichen und alte Zertifikate sind nicht mehr komplett verifizierbar! Das kann ich so also nicht gebrauchen. Daher entferne ich nur die Haken für die Integration in neue Zertifikate. Die Veröffentlichung der Sperrlisten im Hintergrund läuft einfach weiter:

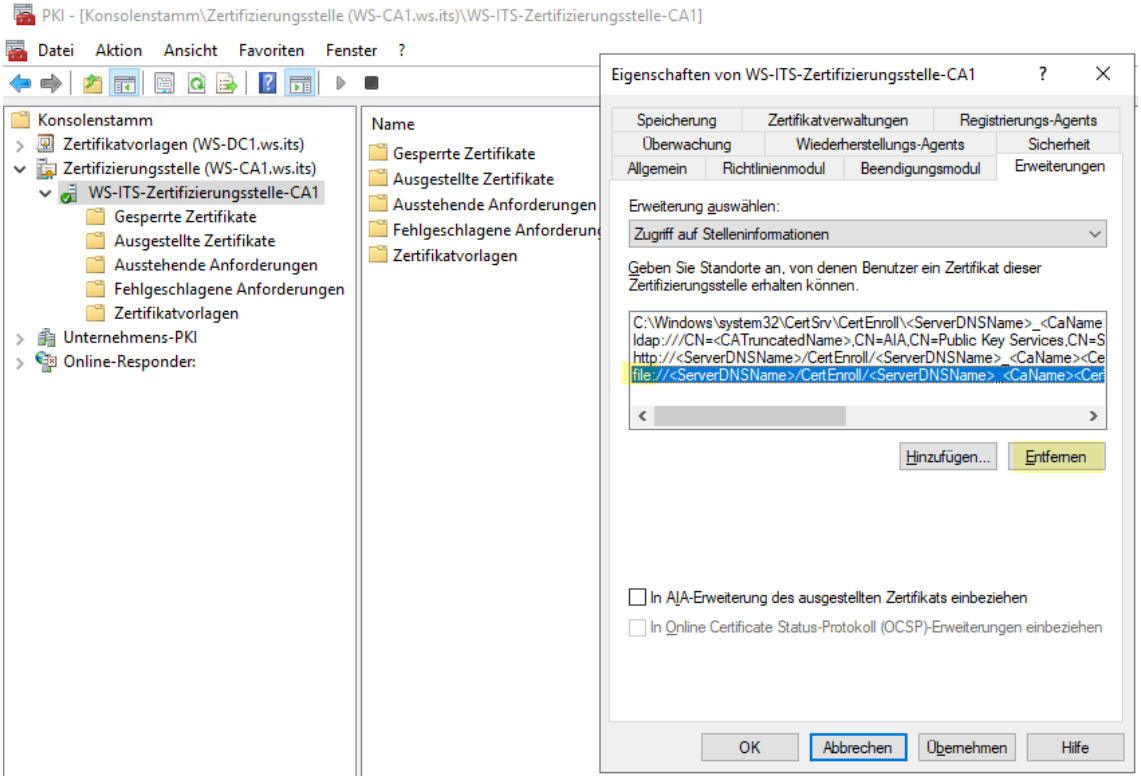


Spätestens zum Ablauf des alten Zertifizierungsstellen-Zertifikates Ende 2021 sind alle alten Zertifikate abgelaufen und durch neue ersetzt. Dann kann ich den LDAP-Record gefahrlos entfernen.

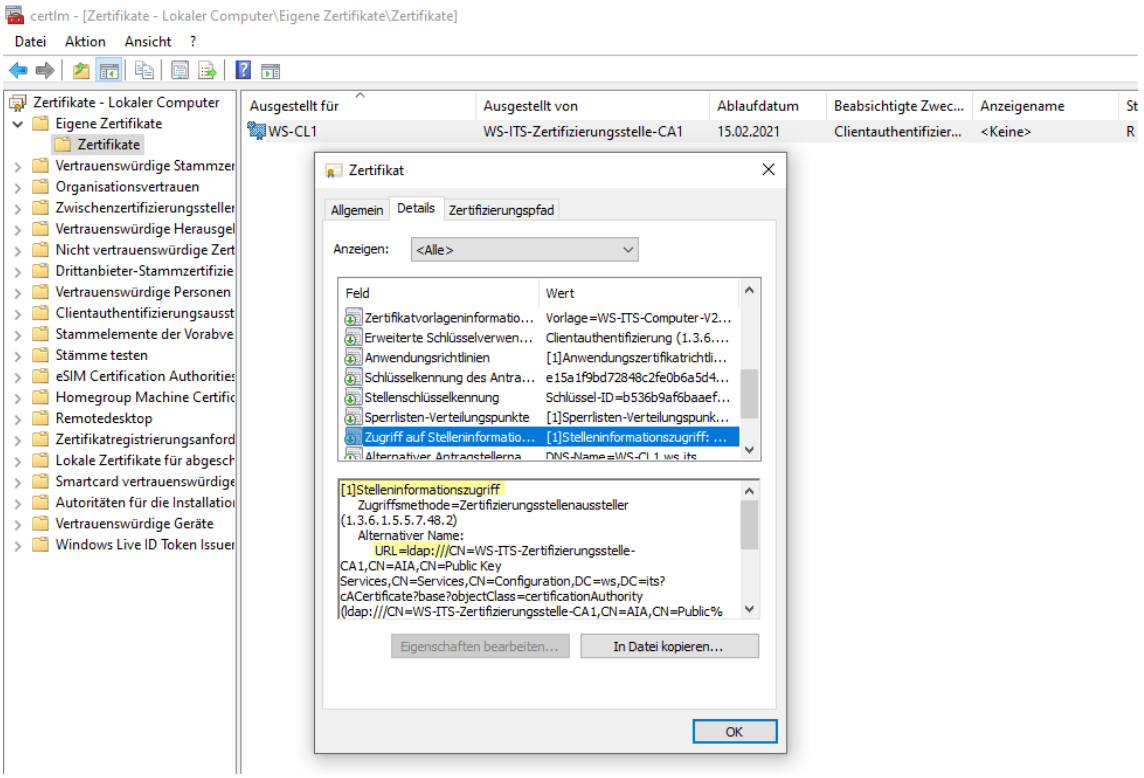
Ich möchte auch keinen Download der Sperrliste über http ermöglichen. So bleibt später nur noch der neue Online Responder, mit dem ich eine Echtzeit-Sperrprüfung erzwingen kann:



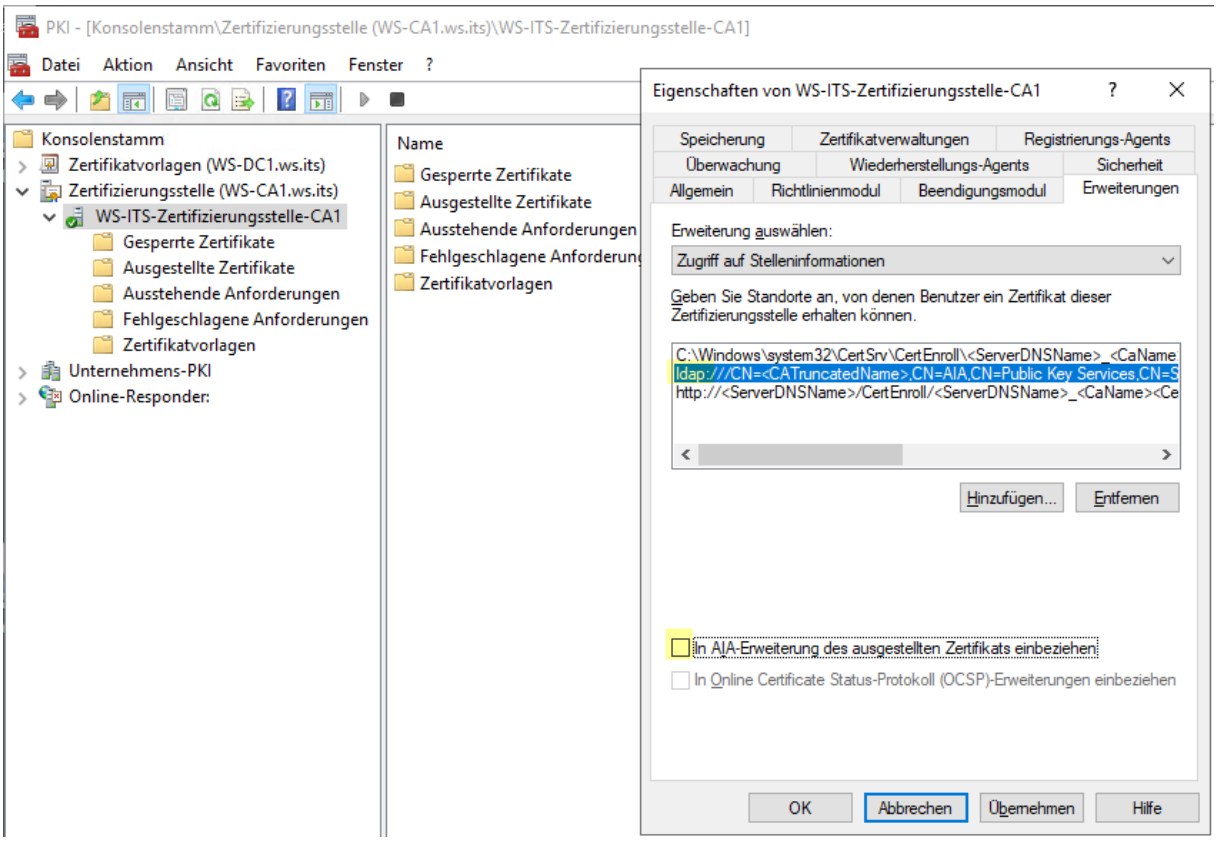
Auch in den Stelleninformationen muss ich aufräumen. Die Defaults enthalten einen File-Record. Den brauche ich nicht:



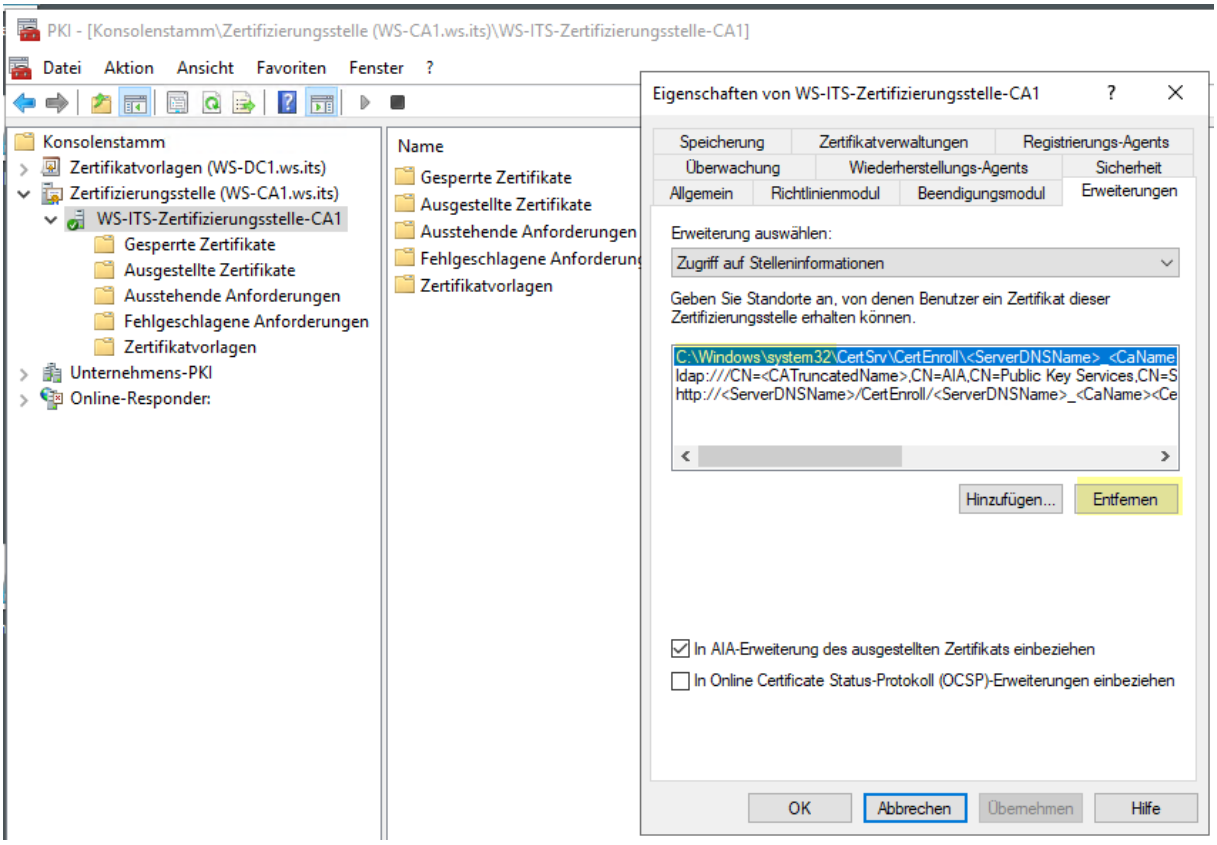
Auch soll es keinen Hinweis in den Zertifikaten mehr geben, dass das Zertifizierungsstellen-Zertifikat im Active Directory gefunden werden kann. Das sind alles Grundvoraussetzungen für eine saubere Veröffentlichung von internen Zertifikaten ins Internet:



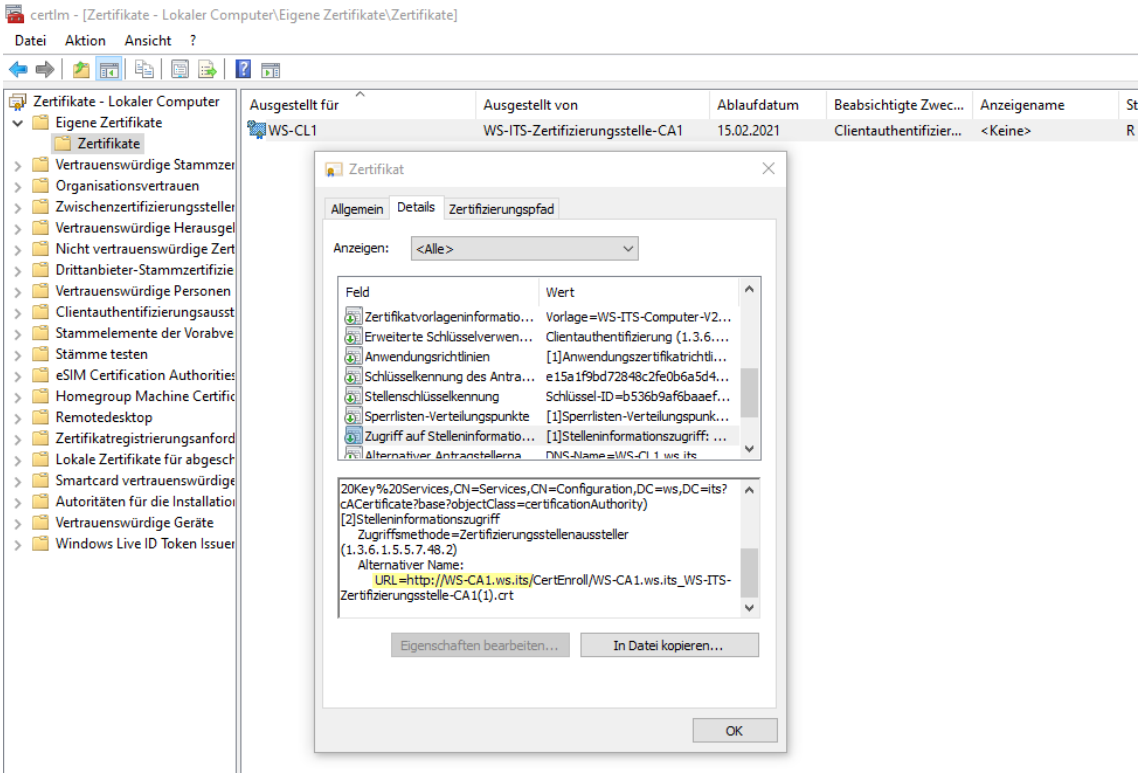
Also fliegt auch dieser Haken raus:



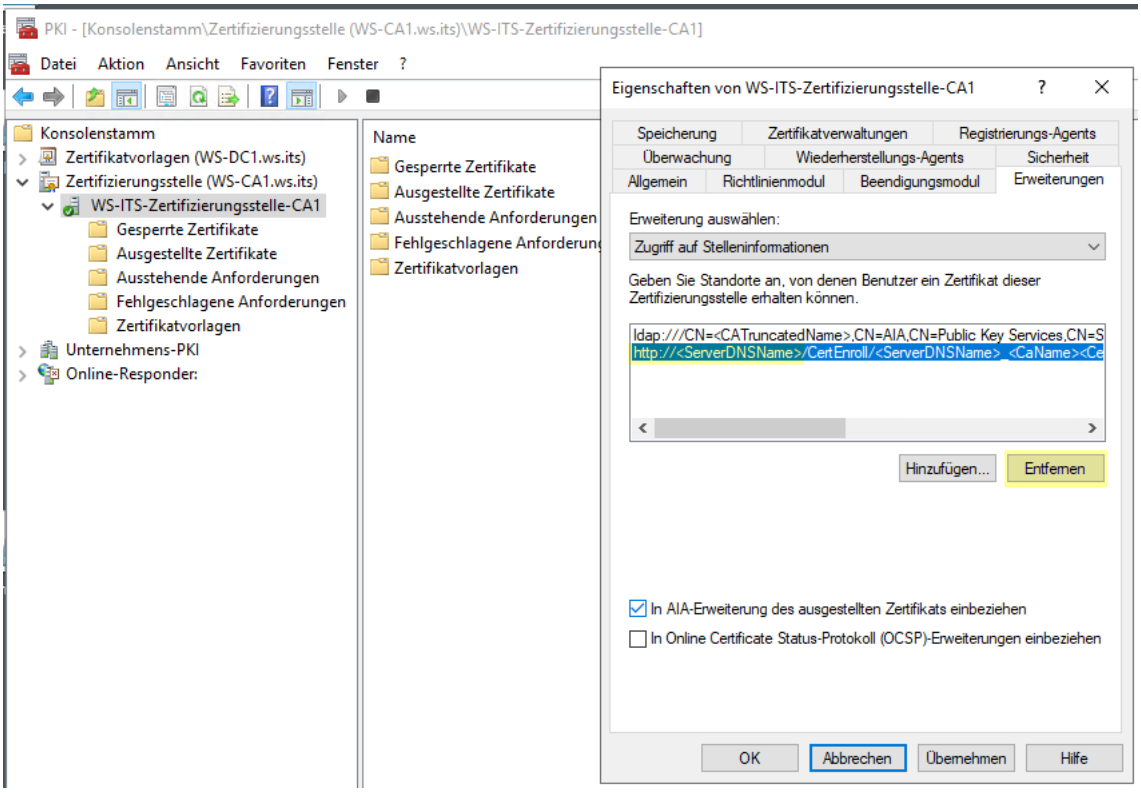
Diesen Default-Record benötige ich ebenfalls nicht:



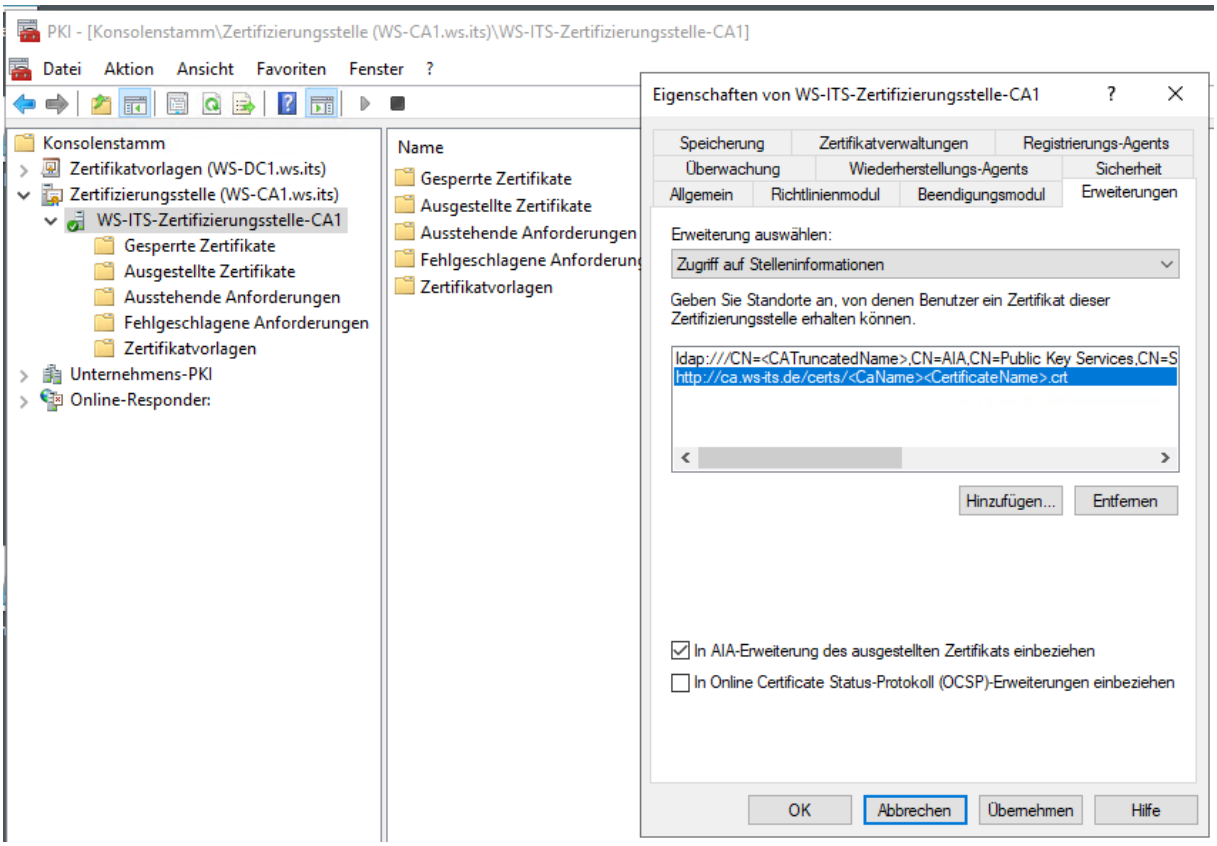
Und dieser Default ist auf den FQDN der CA ausgelegt. Damit wird also immer der interne Name der CA in den Zertifikaten veröffentlicht:



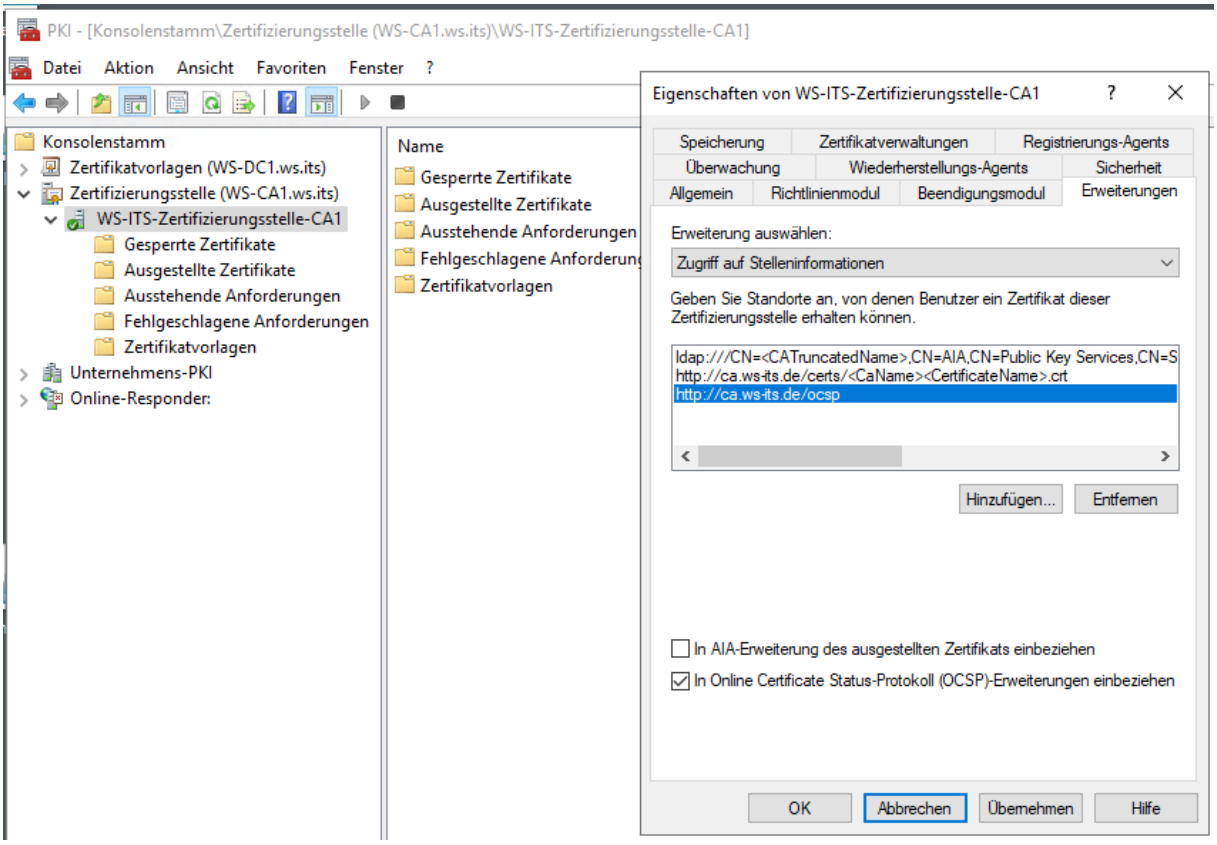
Also raus mit dem Record:



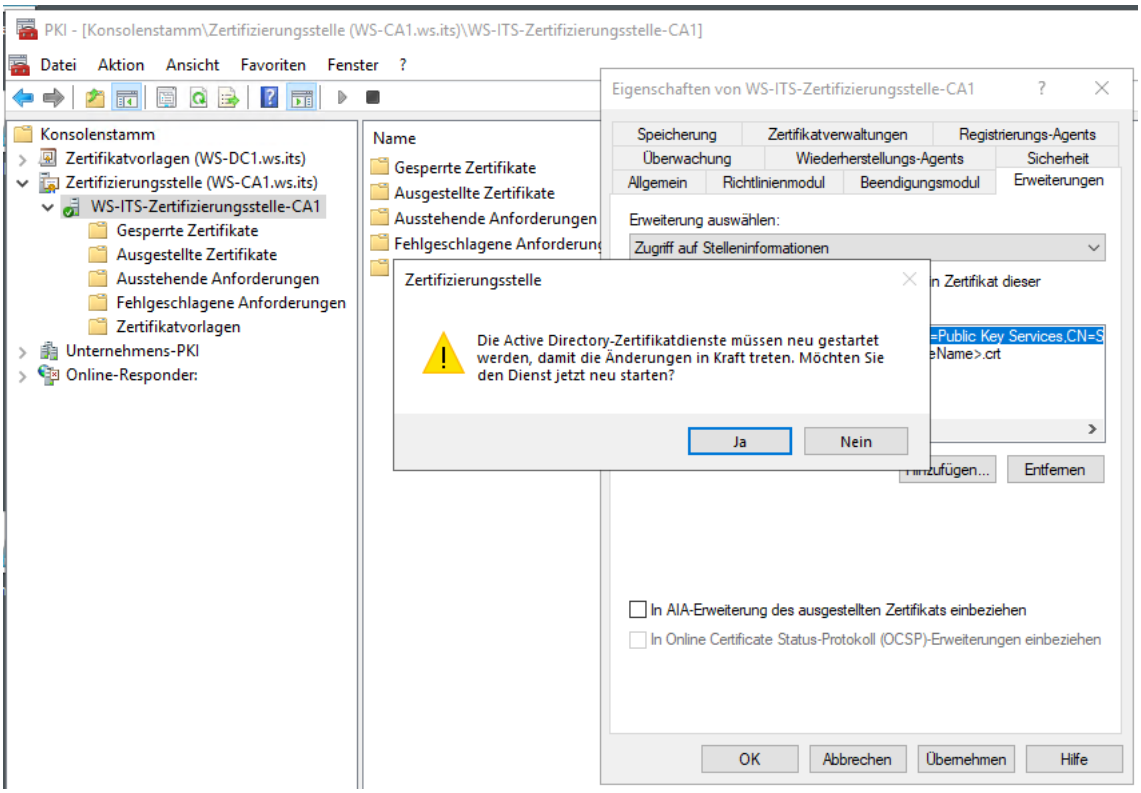
Als Ersatz trage ich nun einen neuen http-Record ein. Hier verwende ich einen neuen CNAME, den ich später auch aus dem Internet heraus erreichbar machen kann:



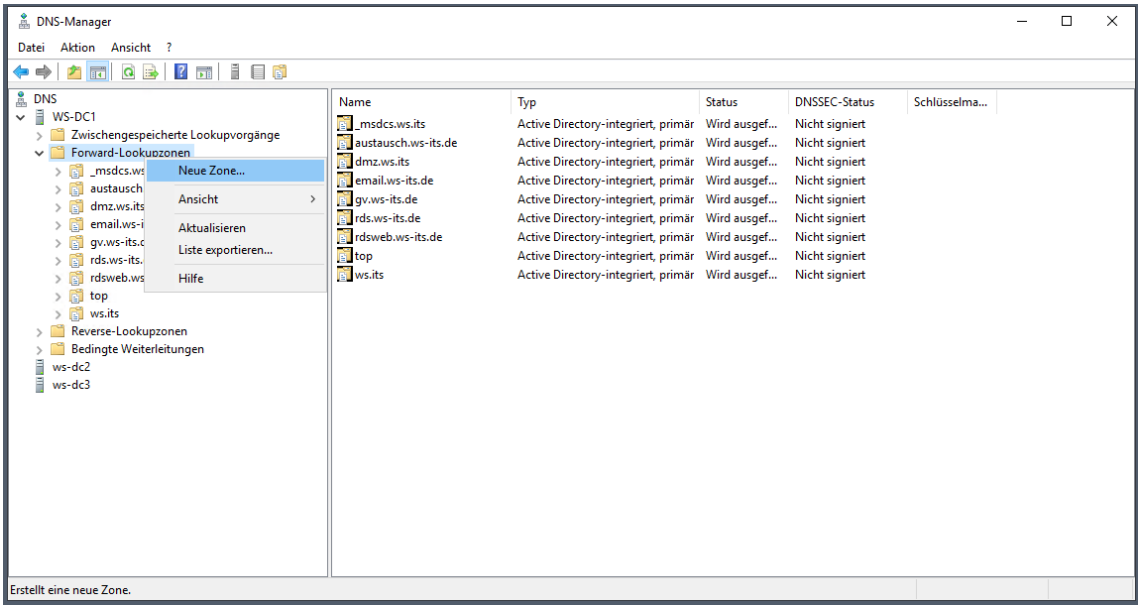
Und ebenso registriere ich hier einen neuen Record für meinen noch nicht vorhandenen Online Responder:

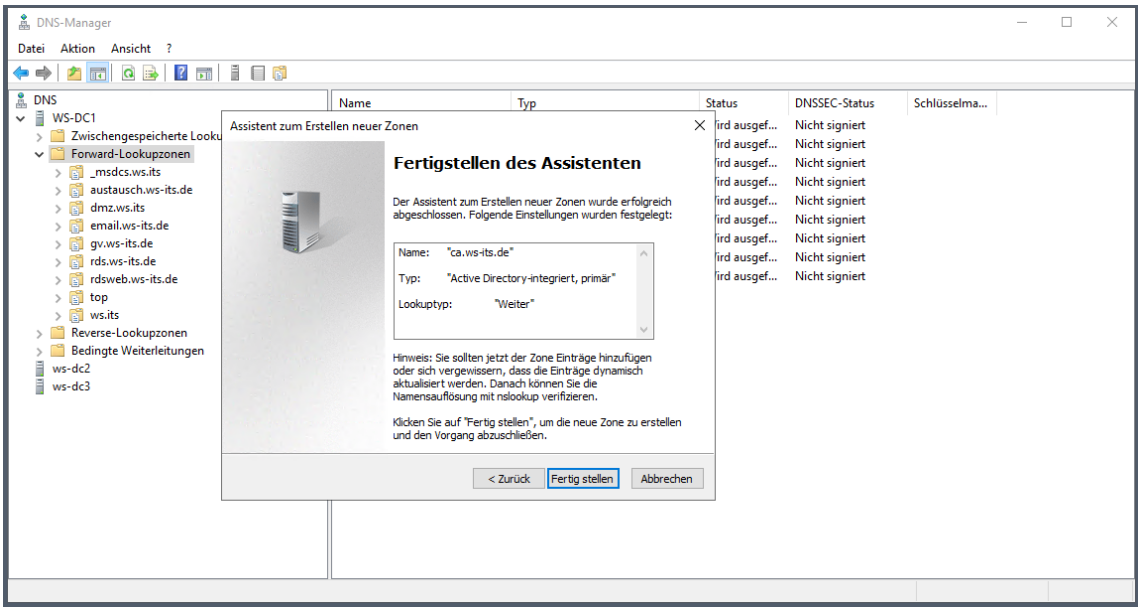


Nach den Anpassungen muss der CA-Service neugestartet werden:

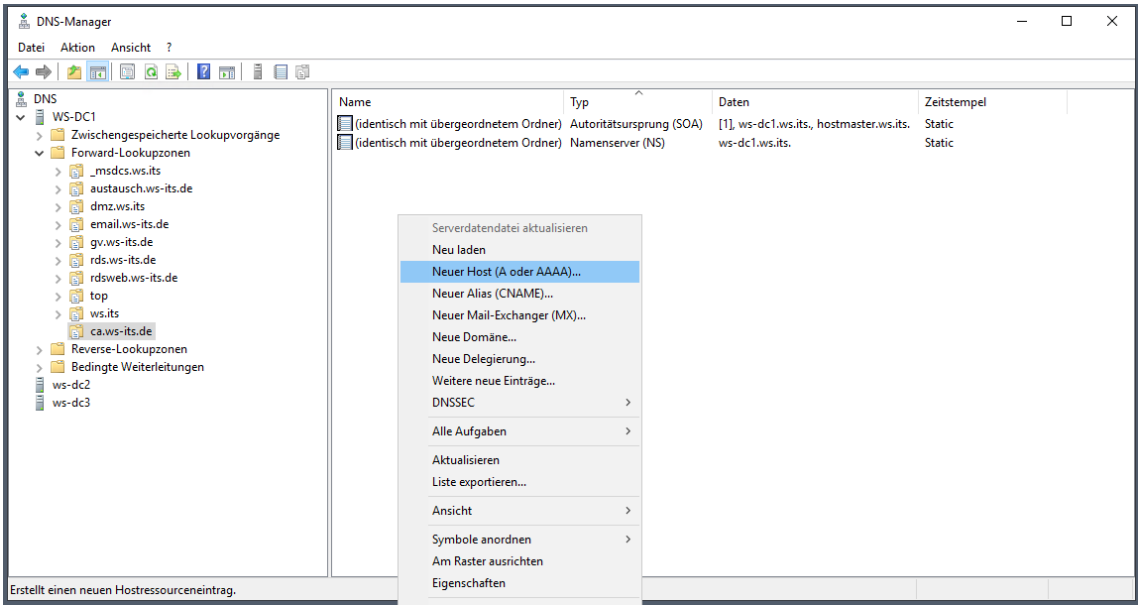


Bevor ich hier was testen kann muss ich den neuen Namen ca.ws-its.de über DNS auflösbar gestalten. Da es aktuell noch keinen Grund für eine Veröffentlichung im Internet gibt, erstelle ich eine interne DNS-Zone mit dem Namen des CNAMEs:

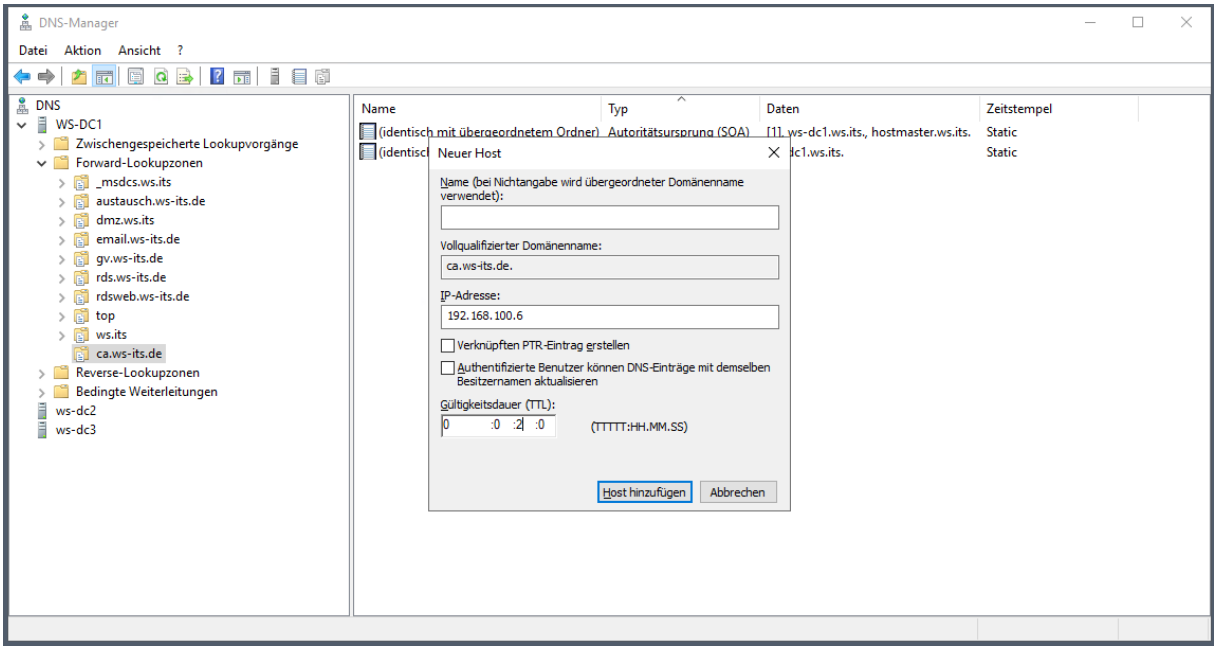




In der neuen DNS-Zone erstelle ich nun einen Host-A-Record:



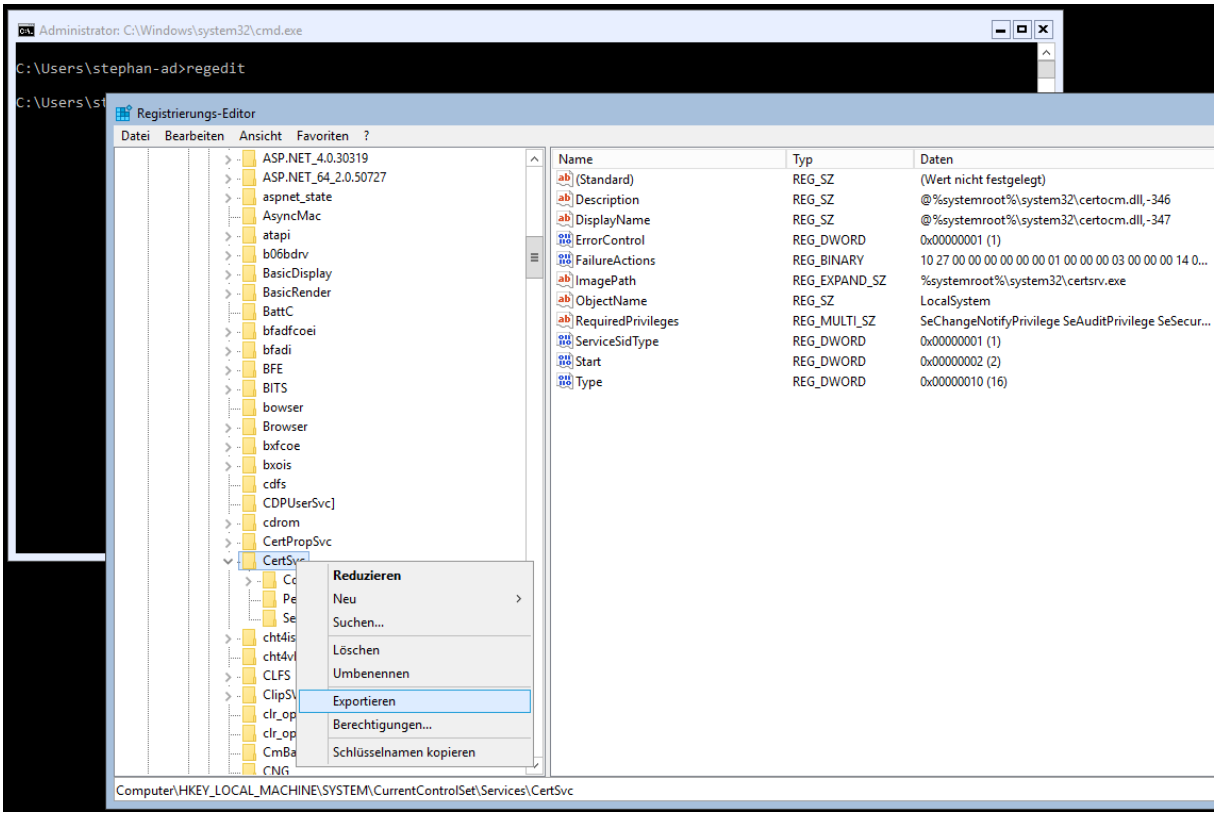
Wichtig ist, dass der neue Record keinen Namen hat. So lenke ich Clients auf die interne IPv4:

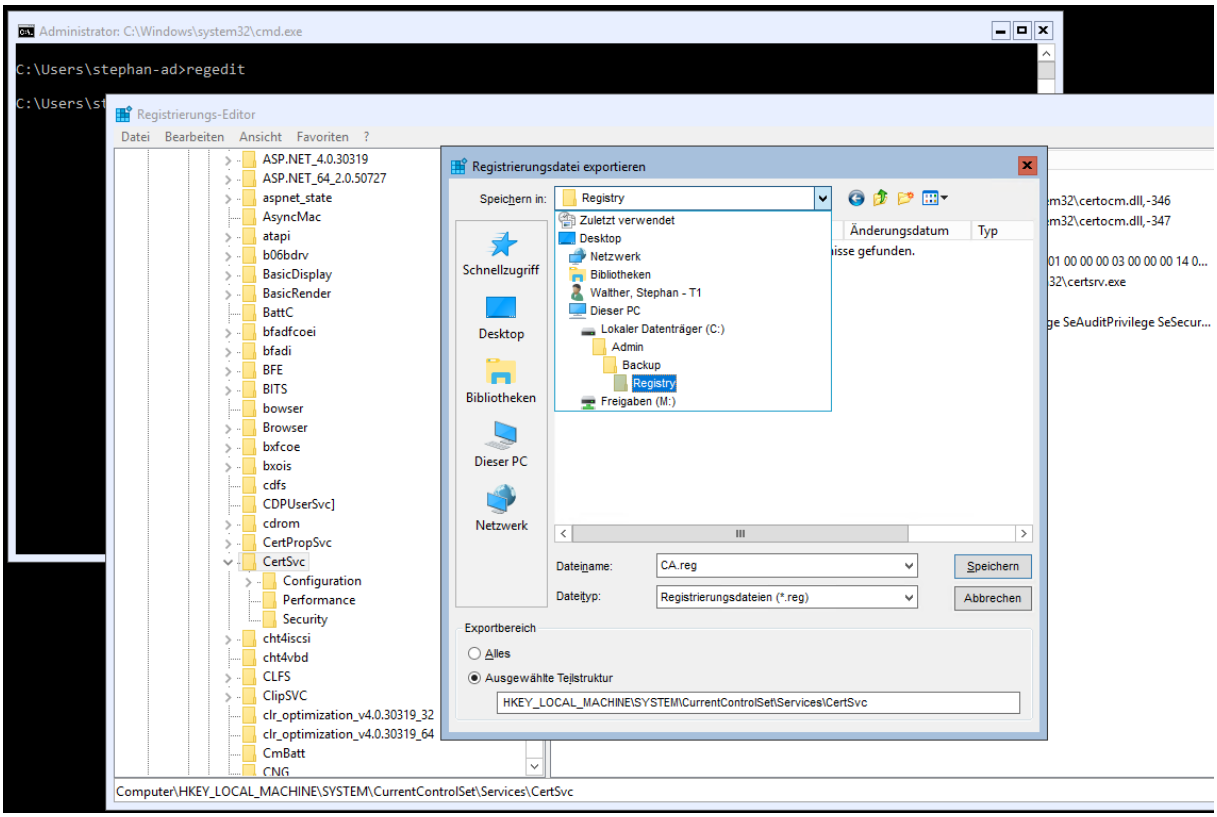


Ein Test von einem Client zeigt den Erfolg:

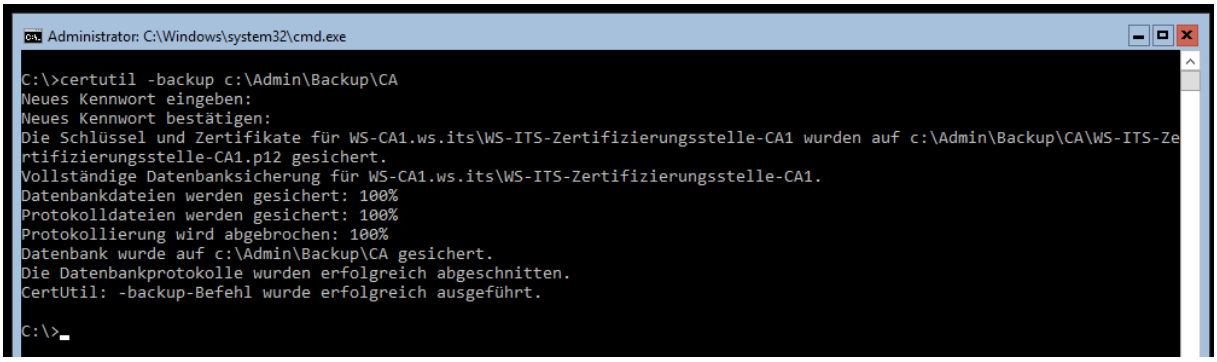
Backup PKI

Die Migration des Services wird durch ein simples Backup & Restore erreicht. Also sichere ich jetzt auf dem alten Server alle Bestandteile. Die Konfiguration des Services liegt in der Registry. Diese exportiere ich in eine Datei:

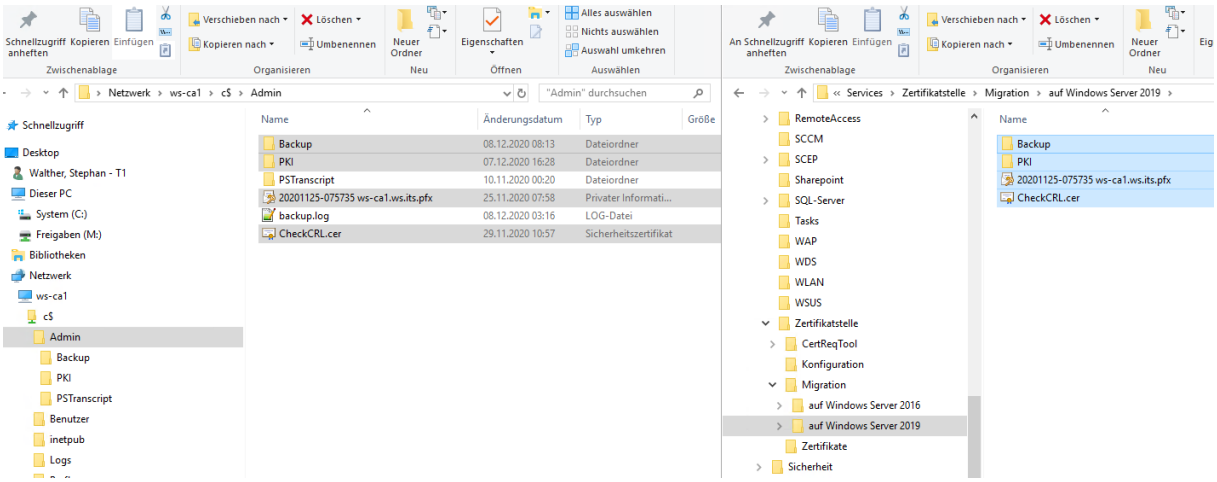




Die eigentliche Datenbank und die Zertifikate kann ich mit certutil sichern:



Die lokal erstellten Sicherungsdateien kopiere ich auf mein AdminShare:



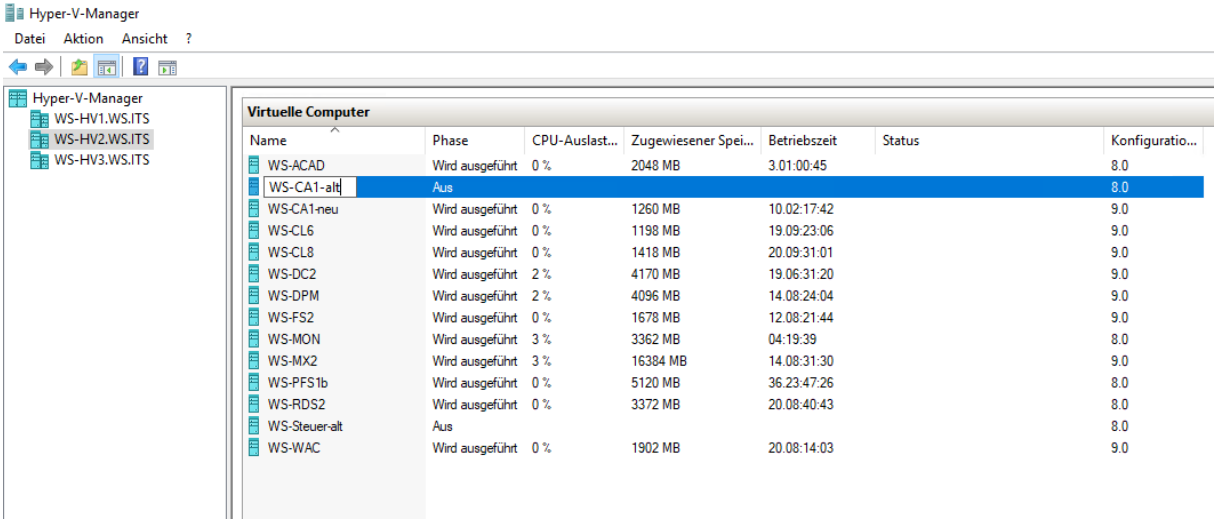
Jetzt kann die Migration starten.

Migration

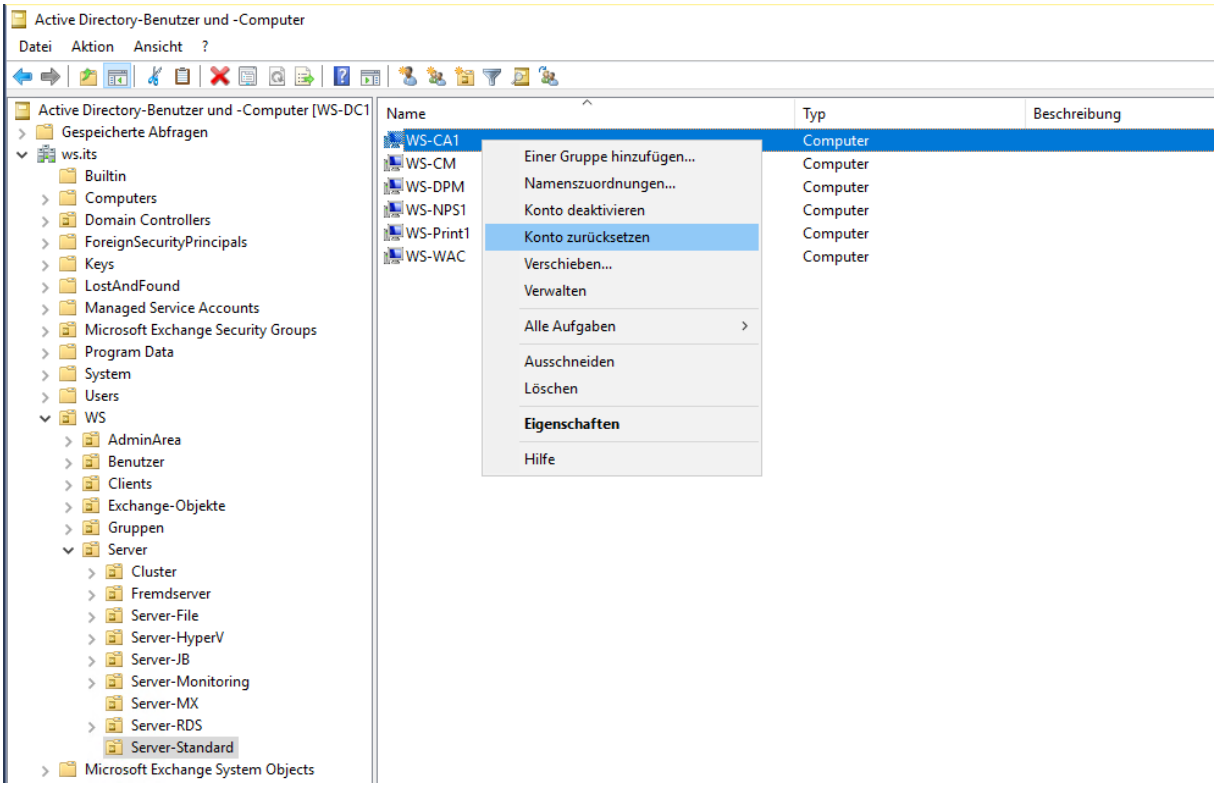
Austausch des Servers

Eine Maintenance für den Service brauche ich nicht einrichten, denn automatische Zertifikatanforderungen werden alle 8 Stunden auf den Clients getriggert. Sollte mal ein Request nicht beantwortet werden, dann kommt der Client eben später wieder.

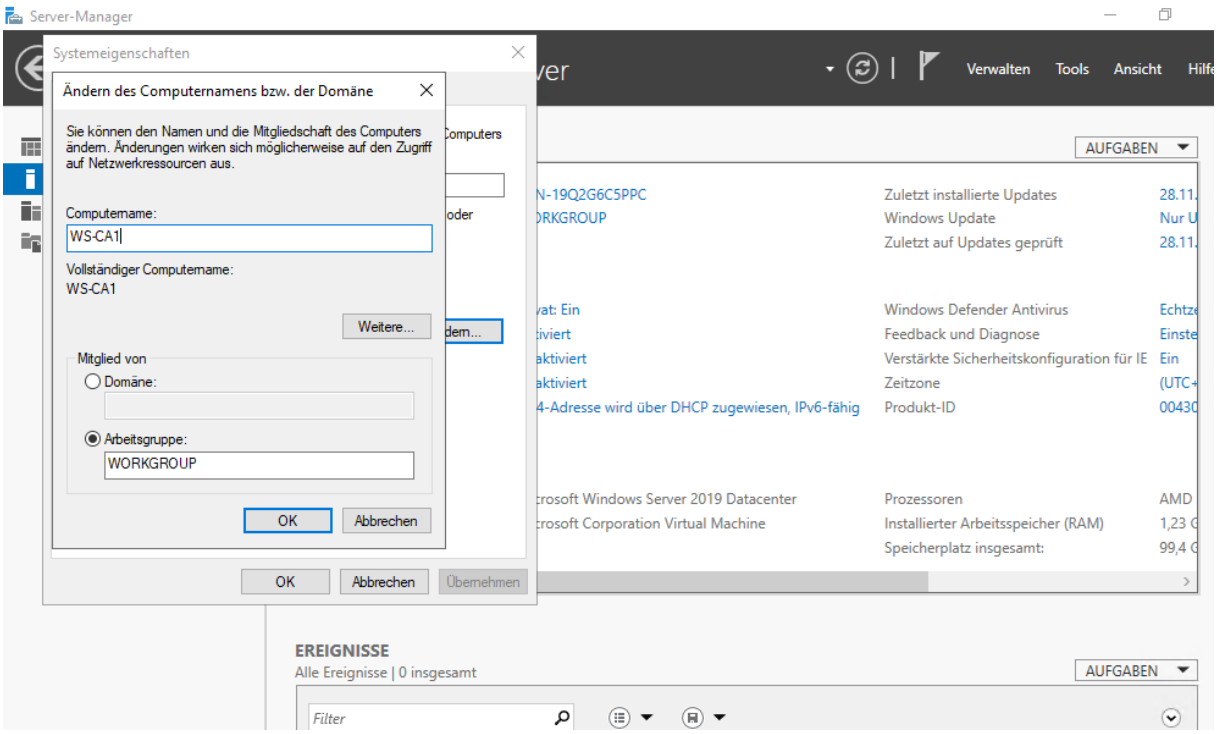
Weiter geht es also mit dem Abschalten der alten Windows Server 2016 Maschine:



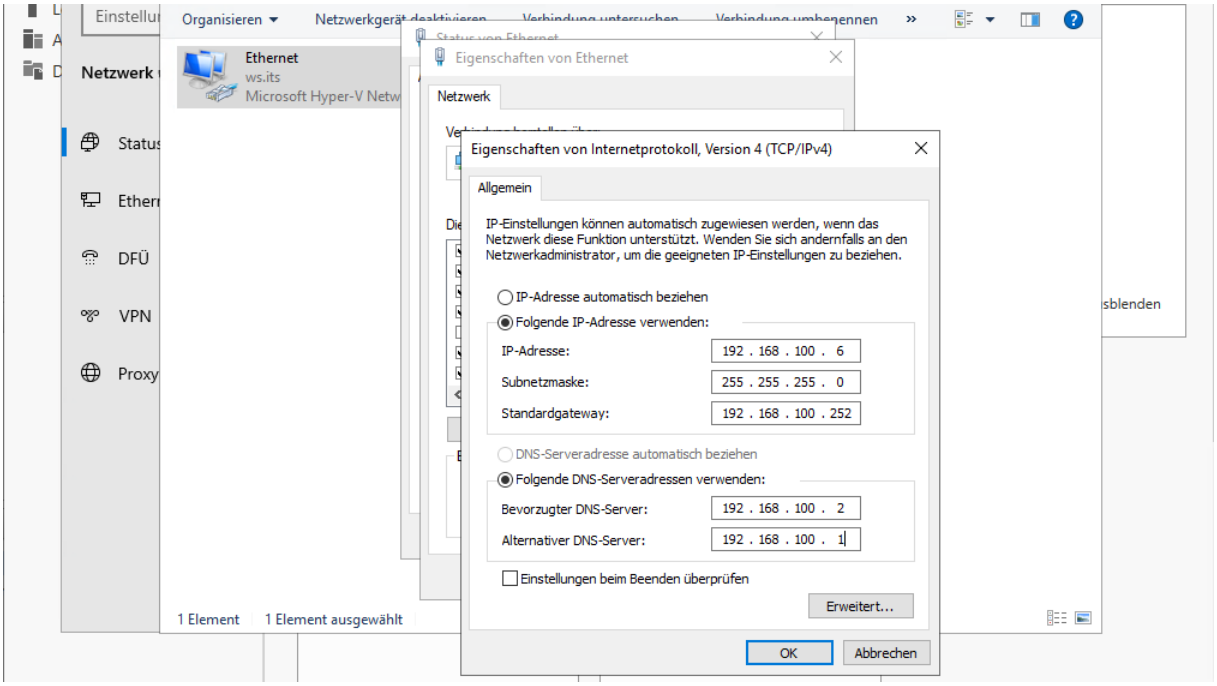
Damit ich die Identität des Computerkontos übertragen kann, setze ich im Active Directory das Konto zurück:



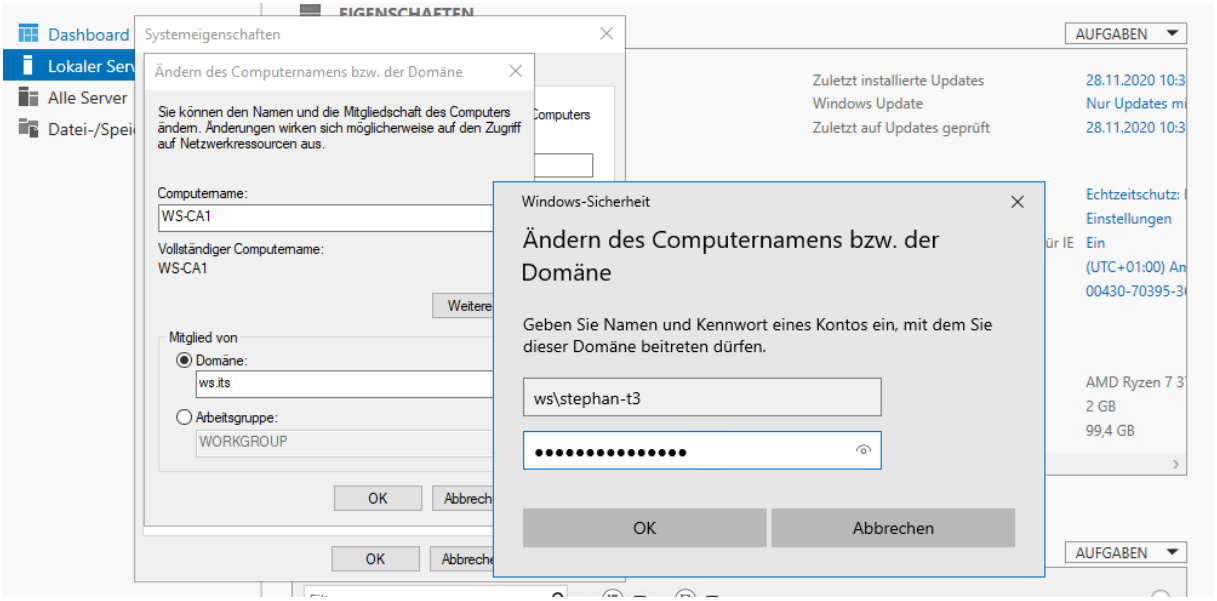
Auf dem neuen Server ändere ich den Computernamen, ohne die Domain zu betreten:



Ebenso passe ich die Netzwerkkonfiguration an und trage die alte IPv4 ein. Der Server ist jetzt im Server-VLAN:



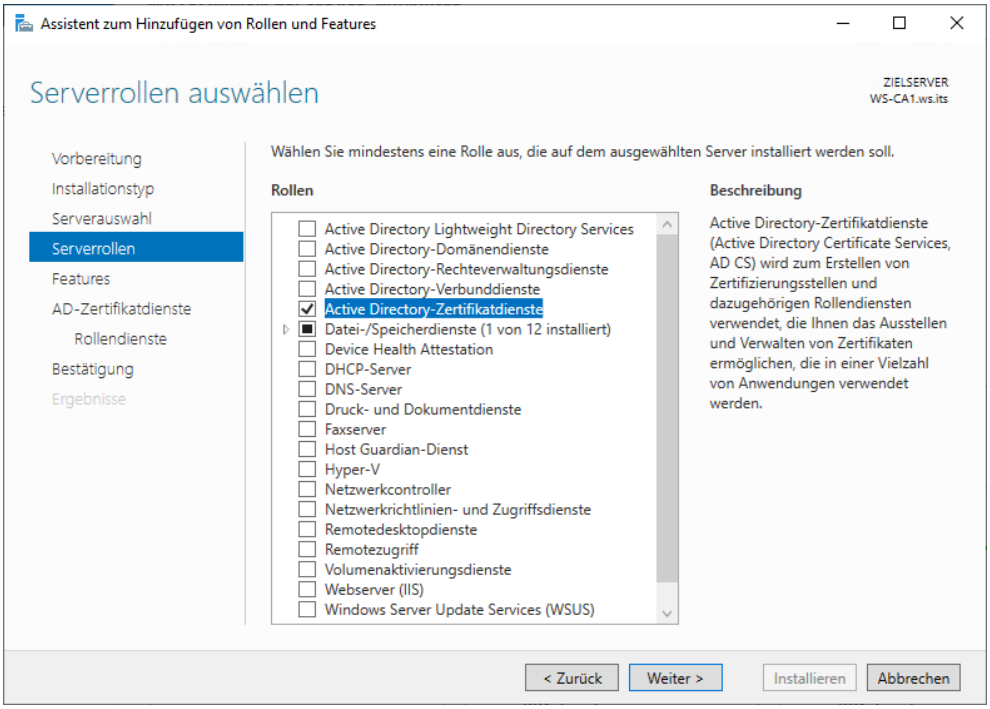
Nach einem Neustart nehme ich den Server in die Domain auf:

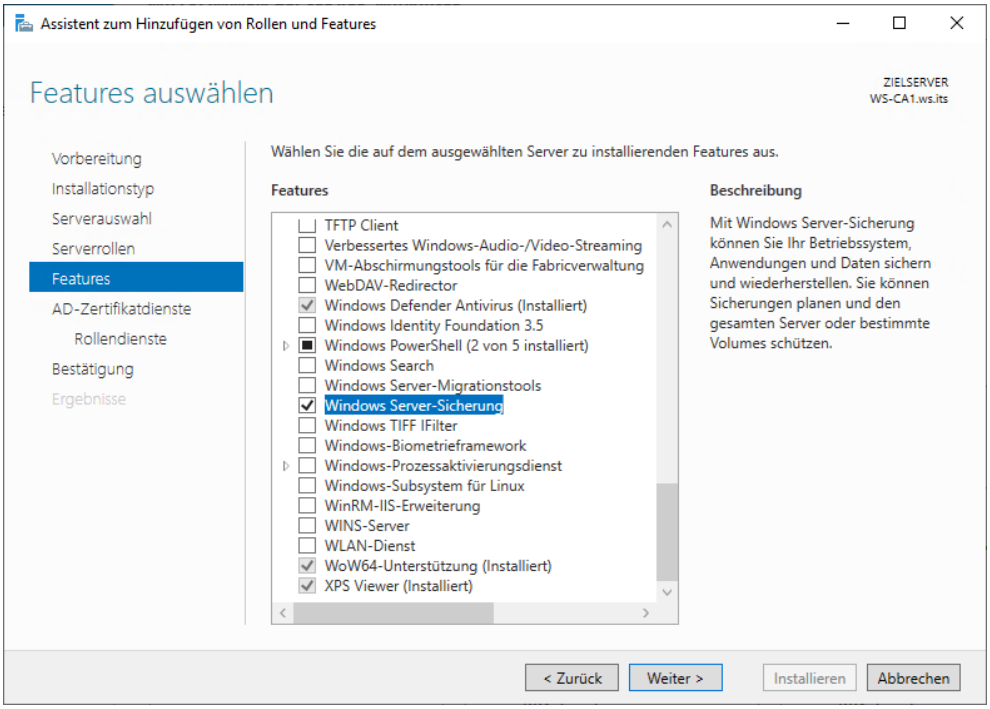


Damit ist das Betriebssystem ausgetauscht.

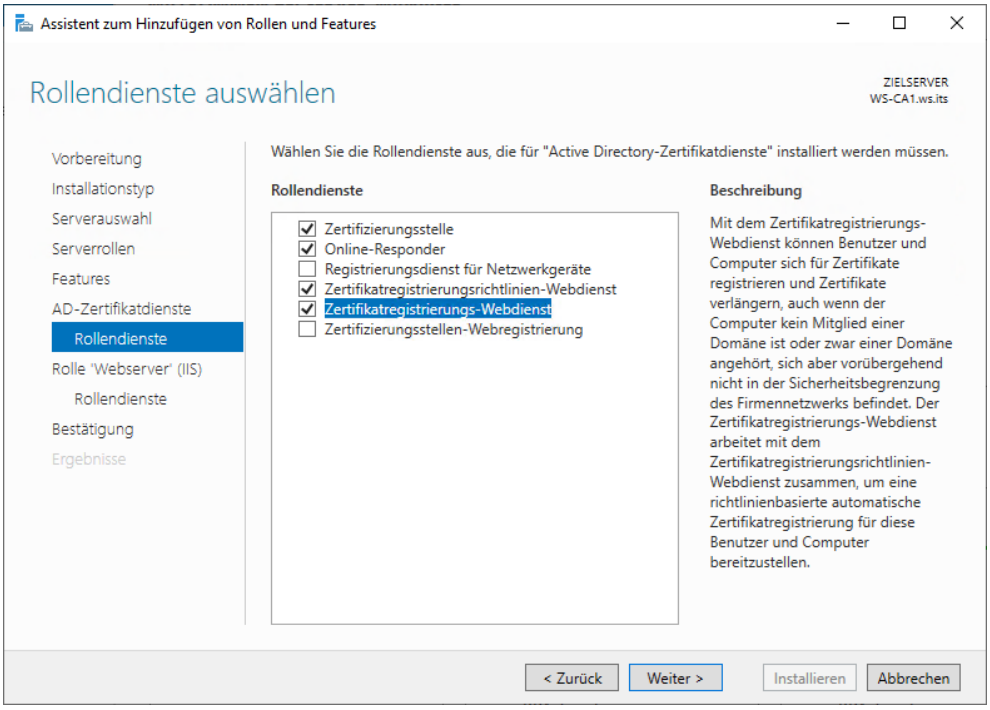
Rolleninstallation

Jetzt installiere ich die erforderlichen Rollen und Features:

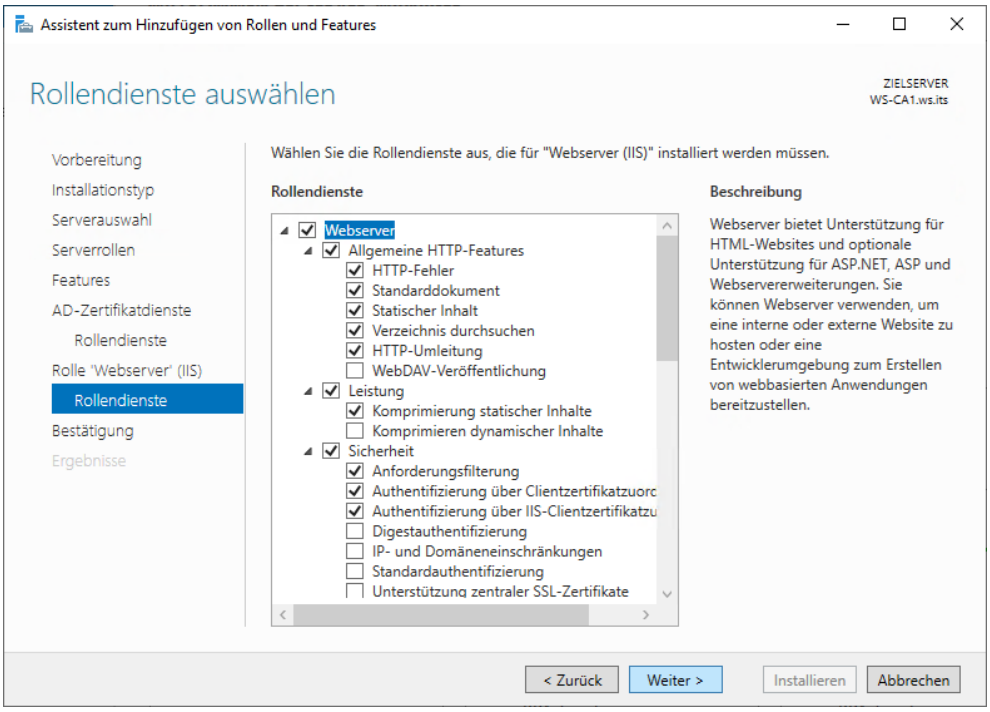




In den Details der Rolle „Active Directory Zertifikatdienste“ wähle ich zusätzlich noch den Online Responder und CEP-CES aus. Der Zertifikatregistrierungsrichtlinien-Webdienst ist CEP, der Zertifikatregistrierungs-Webdienst ist CES:



Der Rest passt:



Problem mit 802.1x

Meine eigenen Migrationen führe ich in meiner Freizeit aus. Da kann es schon einmal vorkommen, dass ich nicht am Stück arbeiten kann, auch wenn es beim Lesen meiner WSHowTo`s vielleicht so scheinen mag. So ist es auch bei dieser Migration. Zwischen den bisherigen Arbeitsschritten und jetzt sind leider einige Tage vergangen. In dieser Zeit war die alte Windows CA nicht mehr erreichbar und der neue Server führt den Service noch nicht aus. Ich dachte mir, das wird schon kein Problem sein und im Vorfeld habe ich natürlich auch nach demnächst ablaufenden Zertifikaten Ausschau gehalten. Da gab es aber keine. Leider hatte ich aber eine andere Abhängigkeit vergessen: Meinen Netzwerkschutz mit 802.1x…

Mein WLAN-Segment für meine internen Clients ist mit einer zertifikatbasierten Authentifizierung konfiguriert. Ein Client, der sich am WLAN-Accesspoint anmelden möchte, muss also ein gültiges Clientzertifikat verwenden. Der Accesspoint leitet

diese Information weiter an meinen WS-NPS1 – das ist ein Radiusserver. Und dieser Server beweist seine Identität ebenfalls mit einem Serverzertifikat.

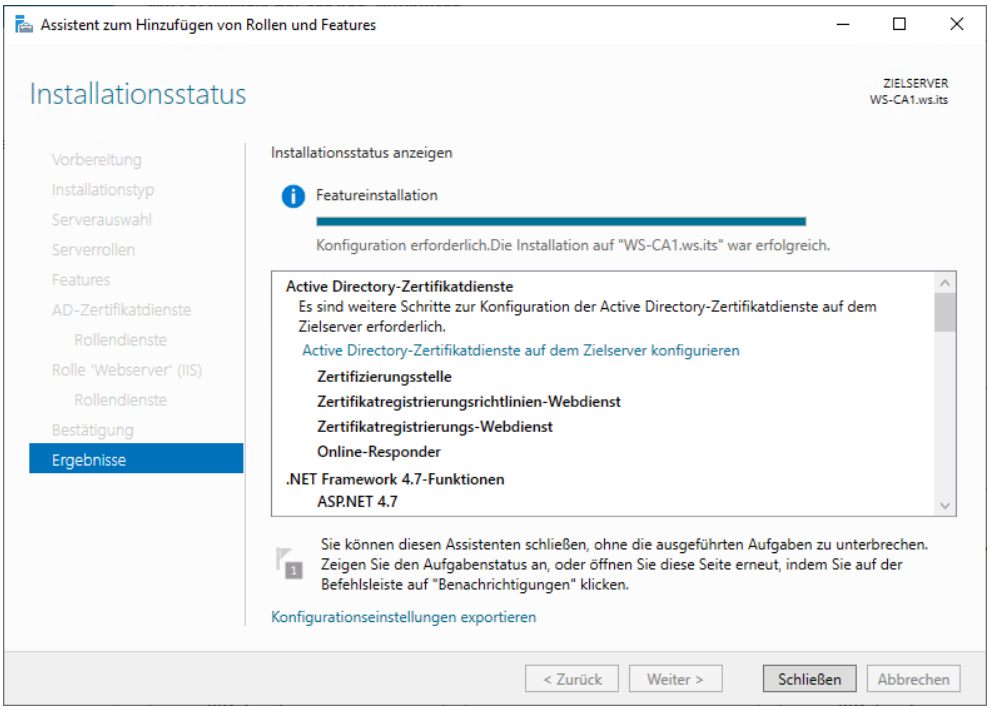
Der NPS (Network Policy Server) prüft die Identität des Clients anhand der Gültigkeit des Client-Zertifikates. Dabei verwendet er auch eine Sperrprüfung. Nur in meinem aktuellen Fall ist die Gültigkeit der auf dem NPS zwischengespeicherten Sperrliste abgelaufen, da die für die Erneuerung zuständige Zertifizierungsstelle offline ist. Daher verweigert der NPS alle Anfragen für eine WLAN-Anmeldung. Das wäre bei einer zügigen Bereitstellung der neuen CA nicht passiert.

Für meinen Fall ist es einfach: Ich ignoriere das WLAN-Problem, da es nur einen betroffenen Client gibt (mein Notebook ist meist verkabelt ans Netzwerk angeschlossen). In der realen Welt wäre jetzt eine Verlängerung der Sperrliste mit manueller Veröffentlichung sinnvoll.

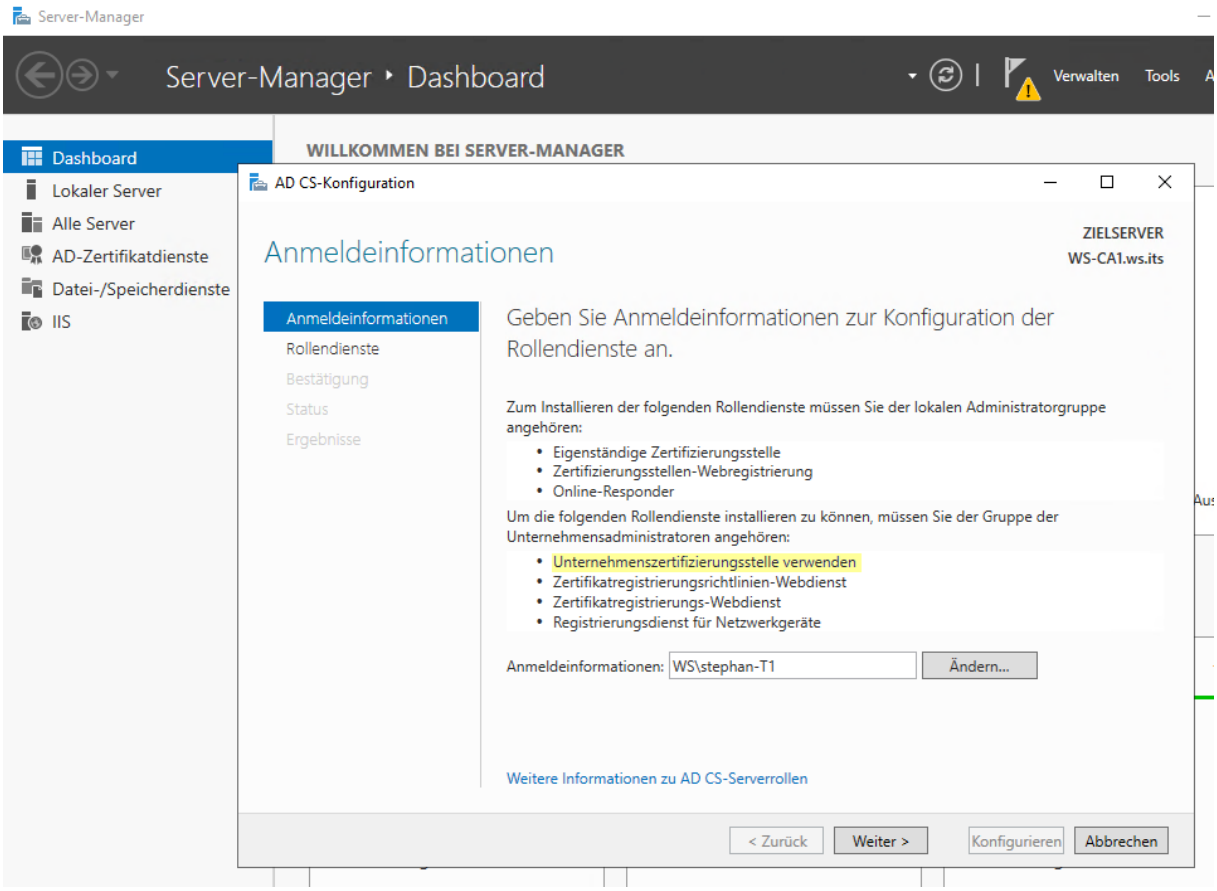
Es geht also weiter mit dem Austausch.

Migration der ADCS

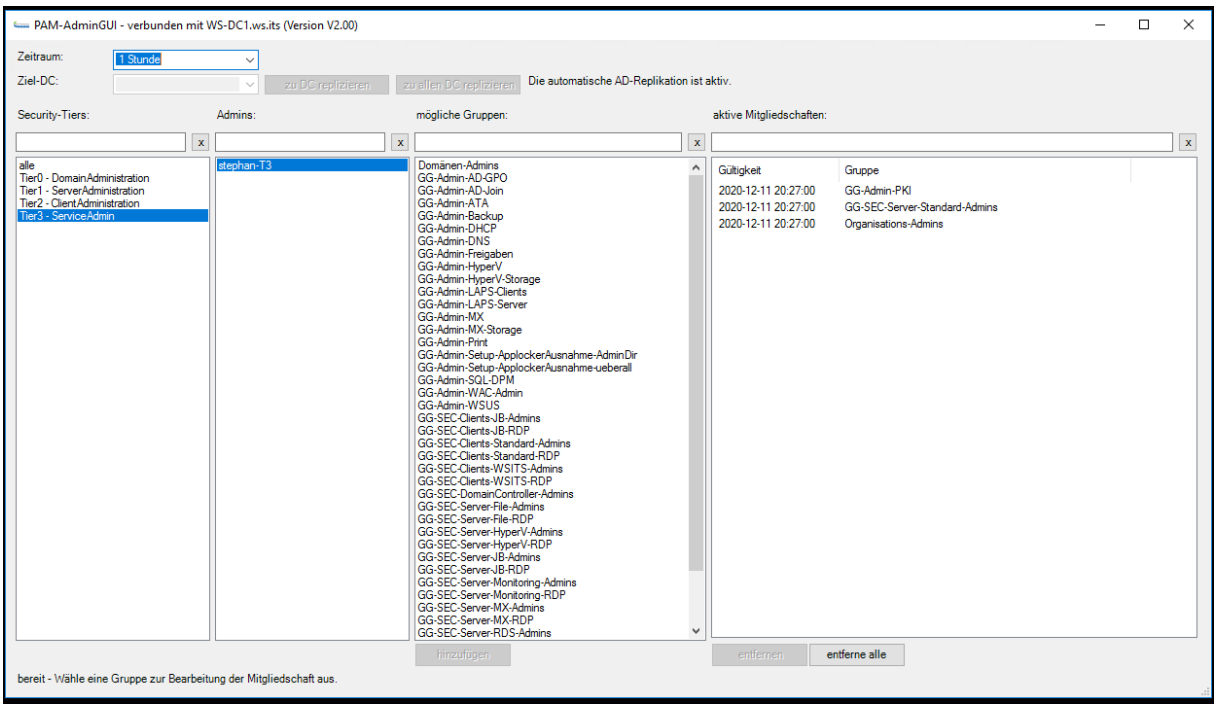
Nun starte ich das Post-Deployment der ADCS:



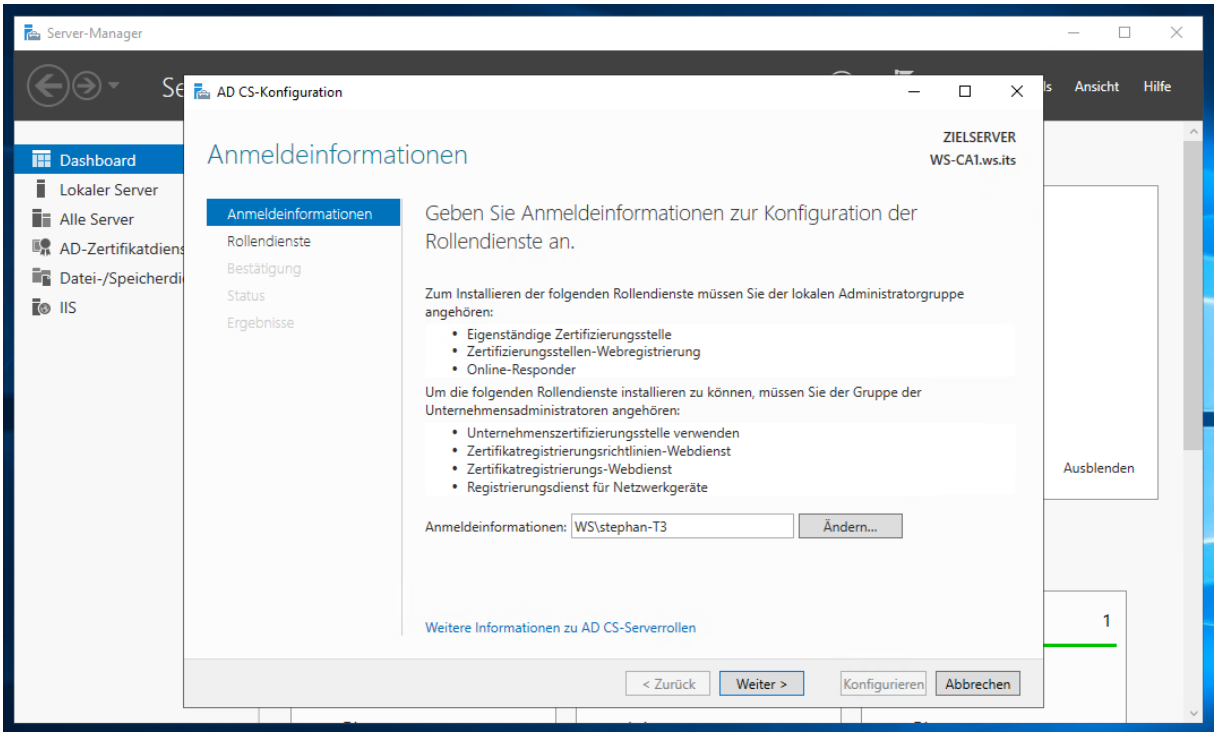
Wichtig ist hier, dass die AD-Integration nur von einem Enterprise-Administrator vorgenommen werden kann:



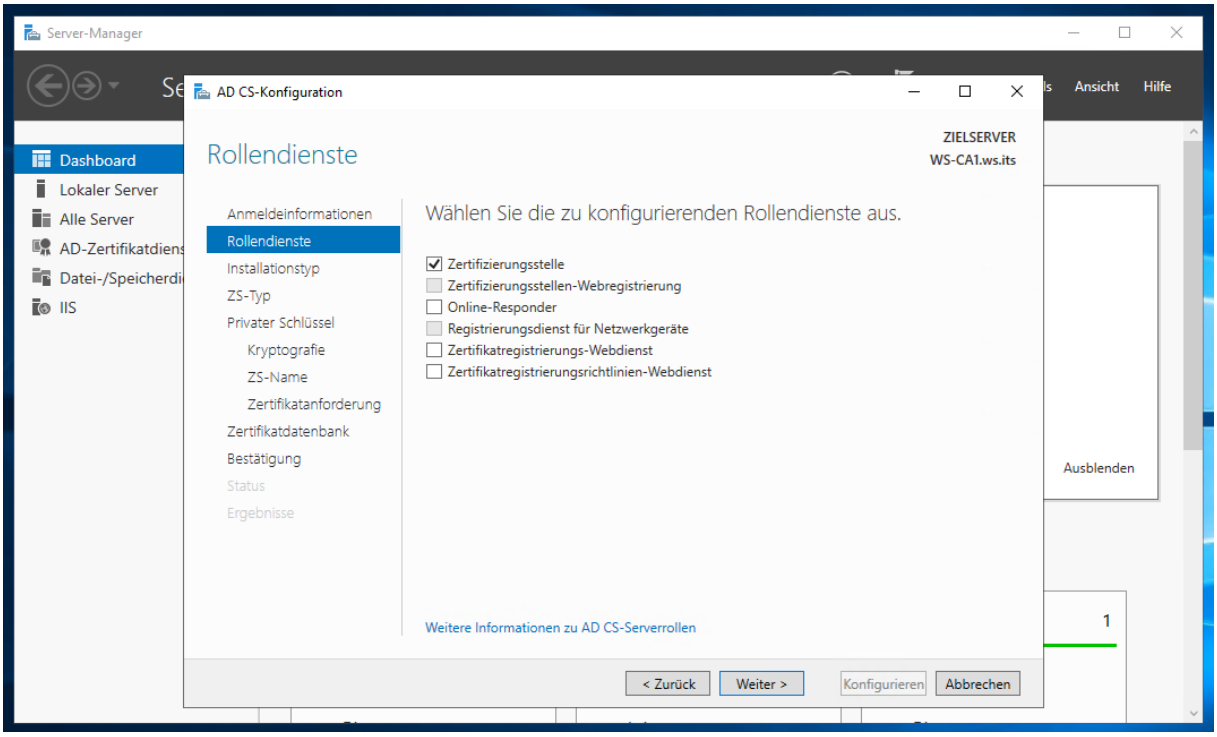
Mein Admin stephan-t1 ist aber nur Memberserver-Admin. Also bereite ich meine T3-Kennung vor:



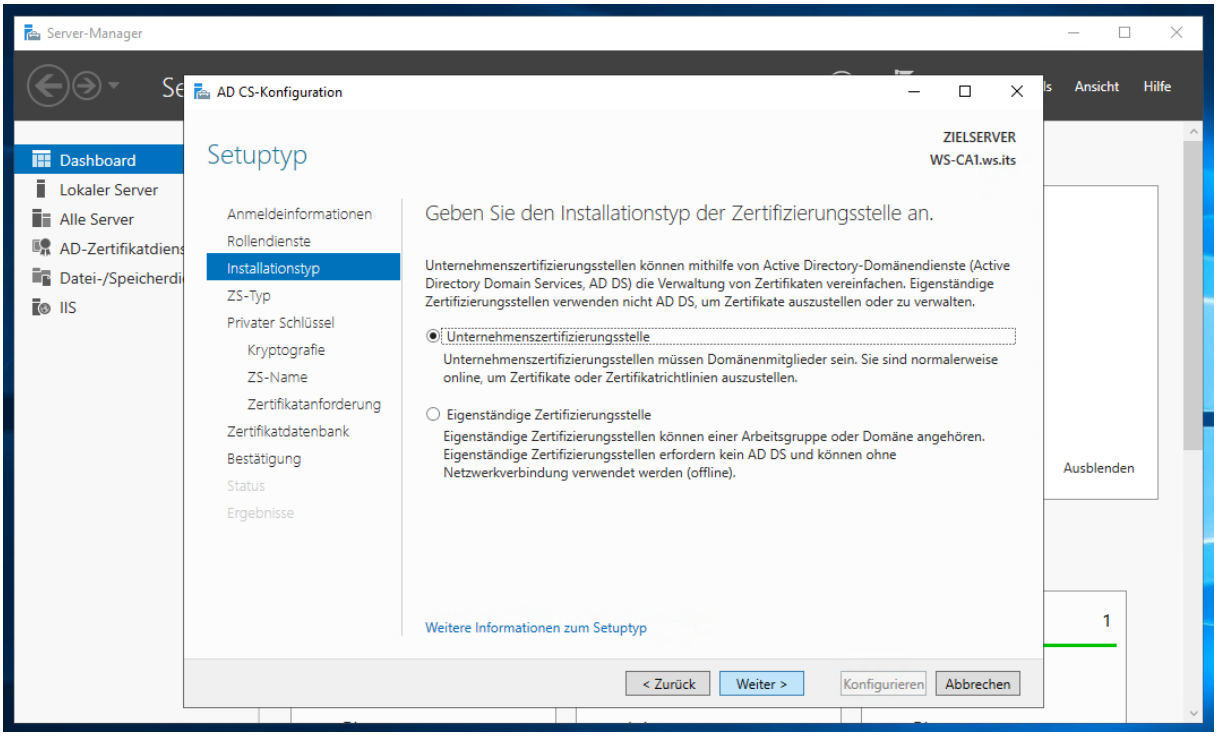
Anschließend wechsle ich den Sicherheits-Kontext:

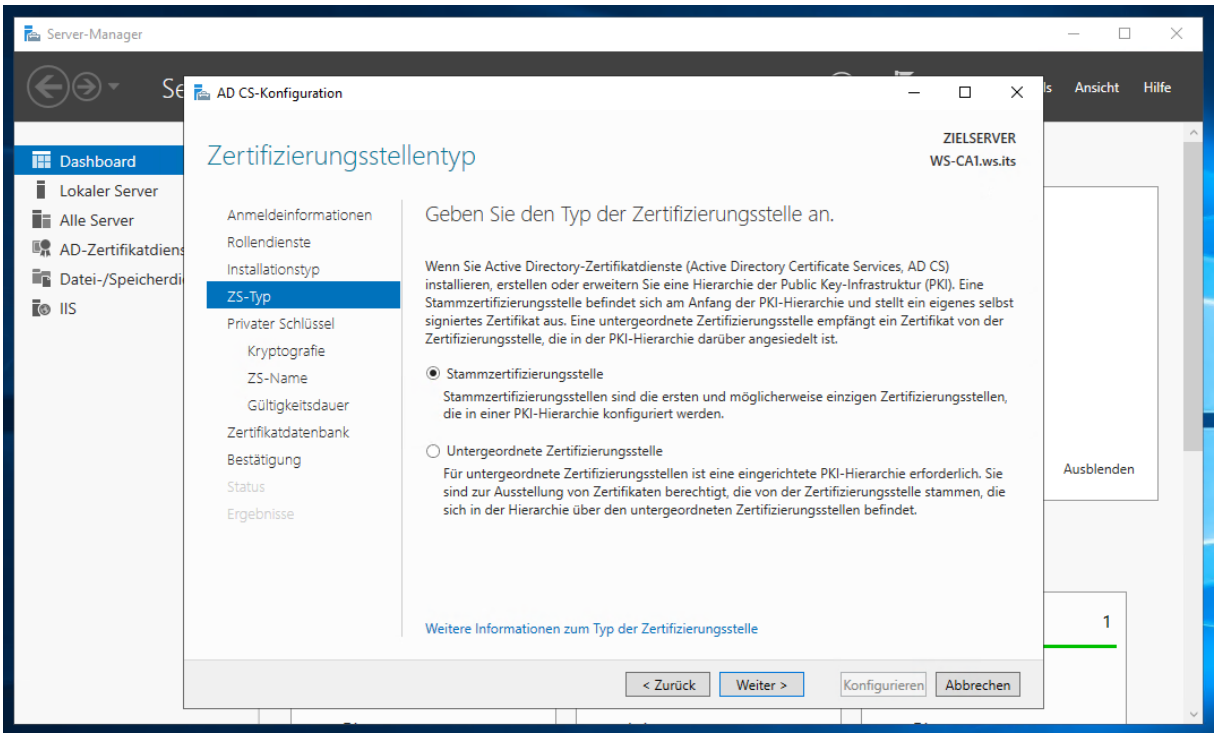


Ich installiere nur die Rolle ADCS:

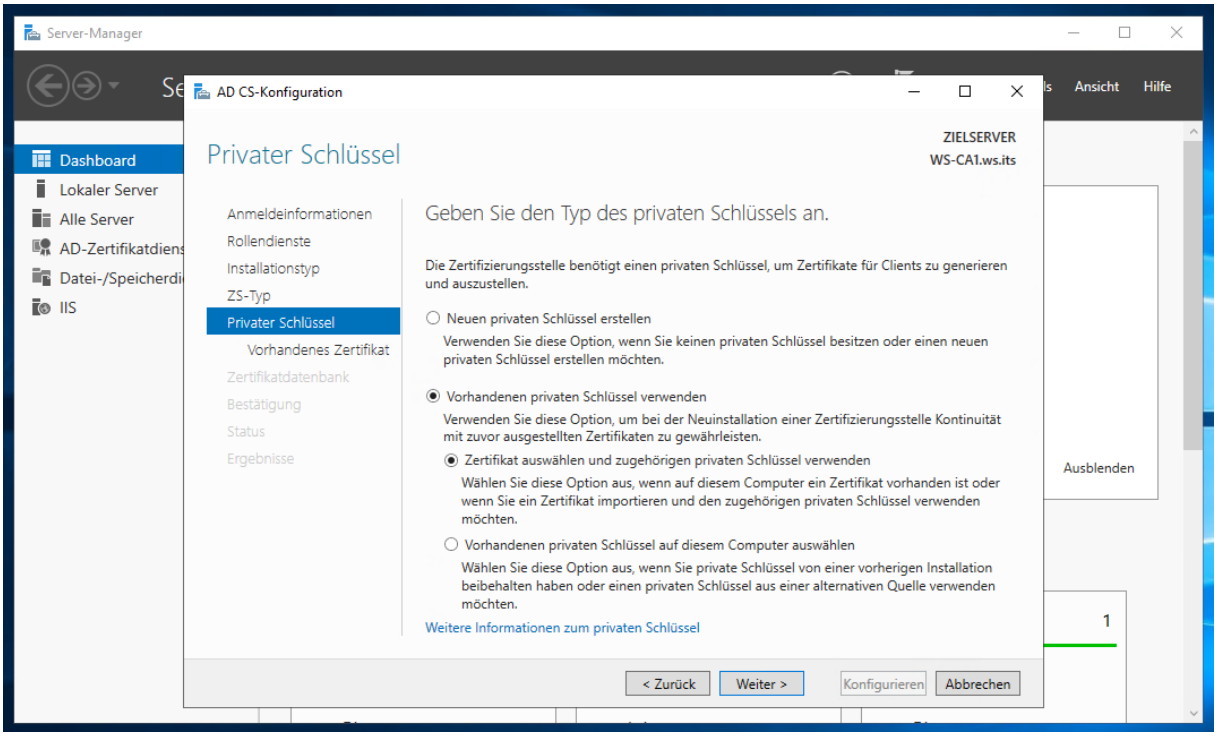


Der neue Server soll eine Enterprise Root-CA werden – so wie vorher:

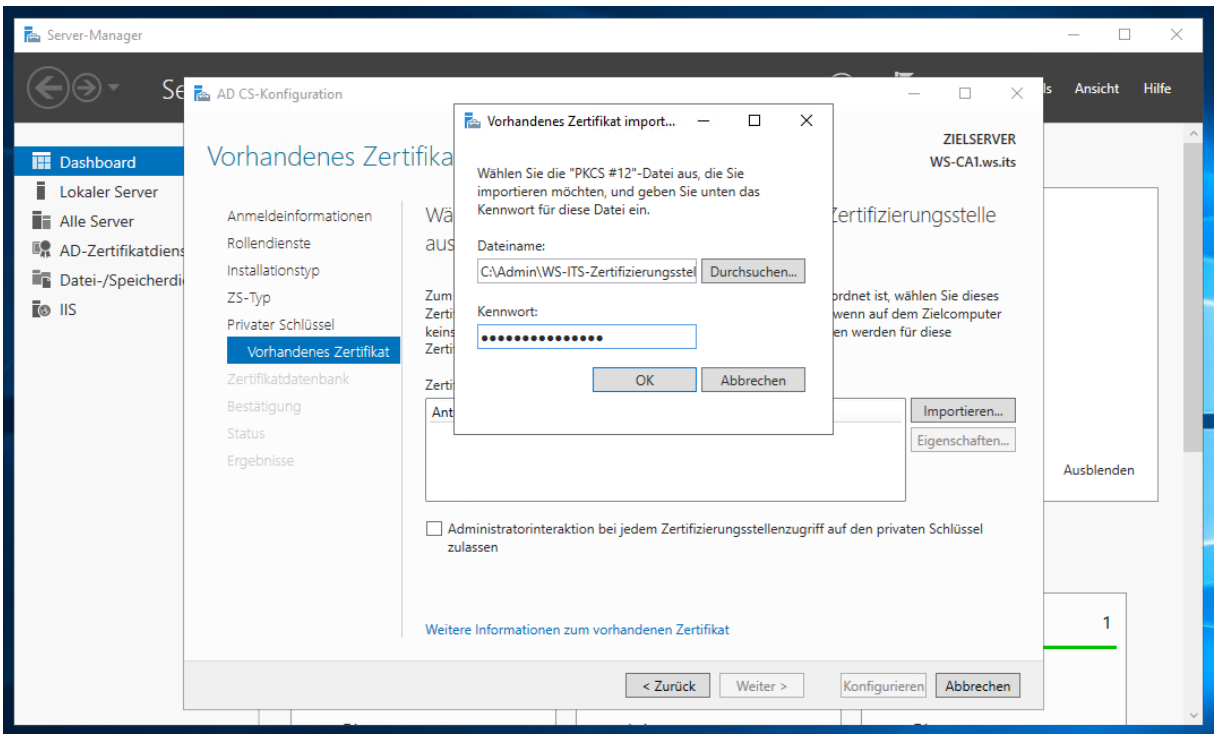


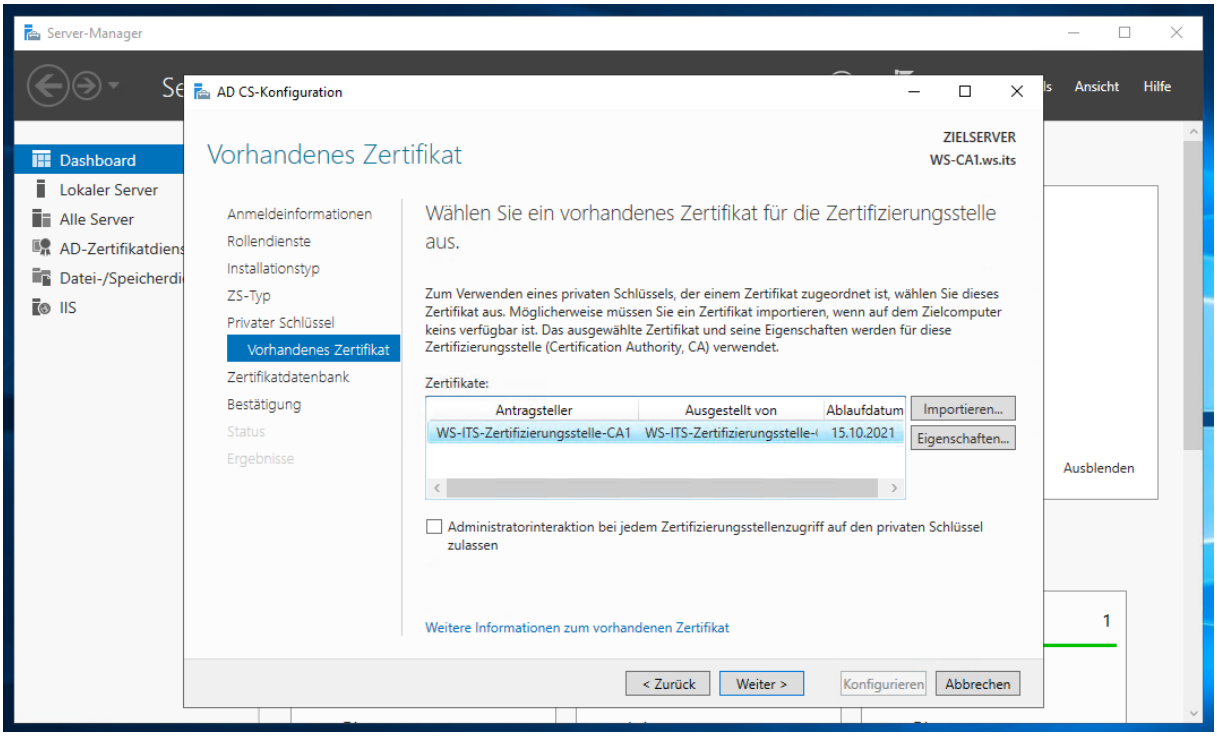


Aber anders als bei einer Neuinstallation muss ich hier die Zertifikate des alten Servers verwenden. Diese definieren die Identität:

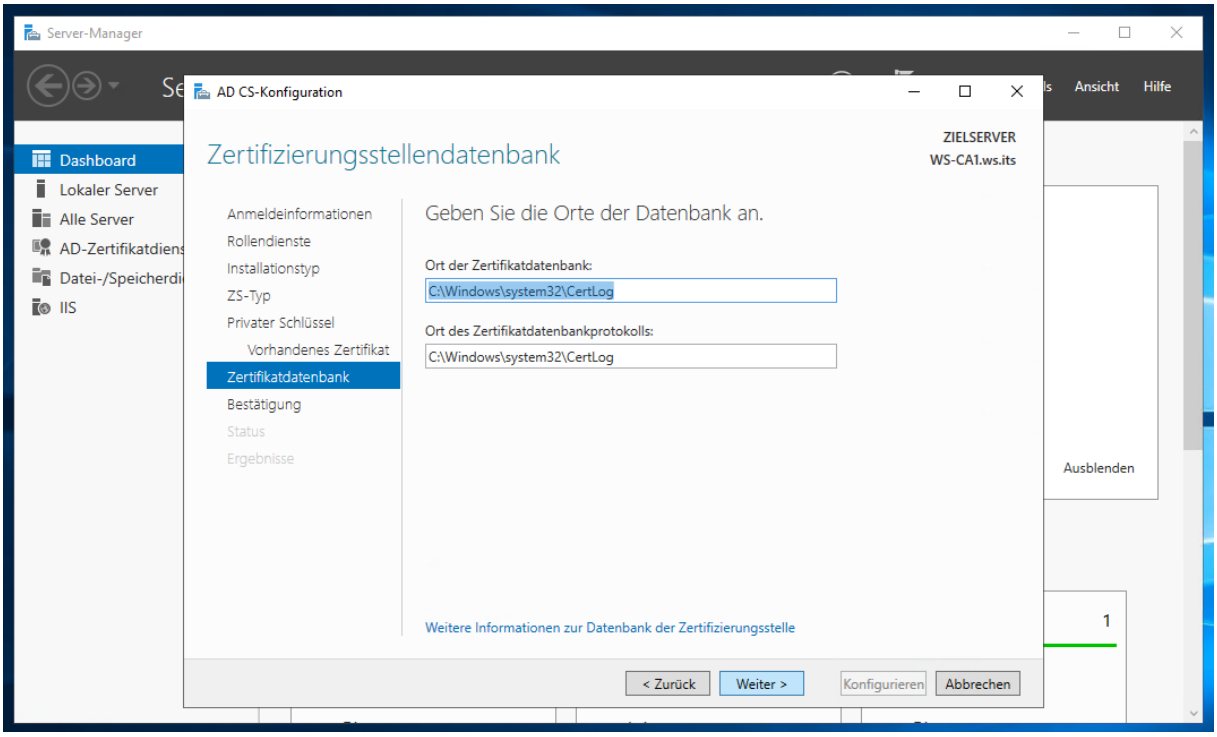


Die Zertifikate wurden durch das Backup mit Certutil in PFX-Dateien exportiert. Ich muss aber nur noch das aktuell gültige auswählen und importieren:

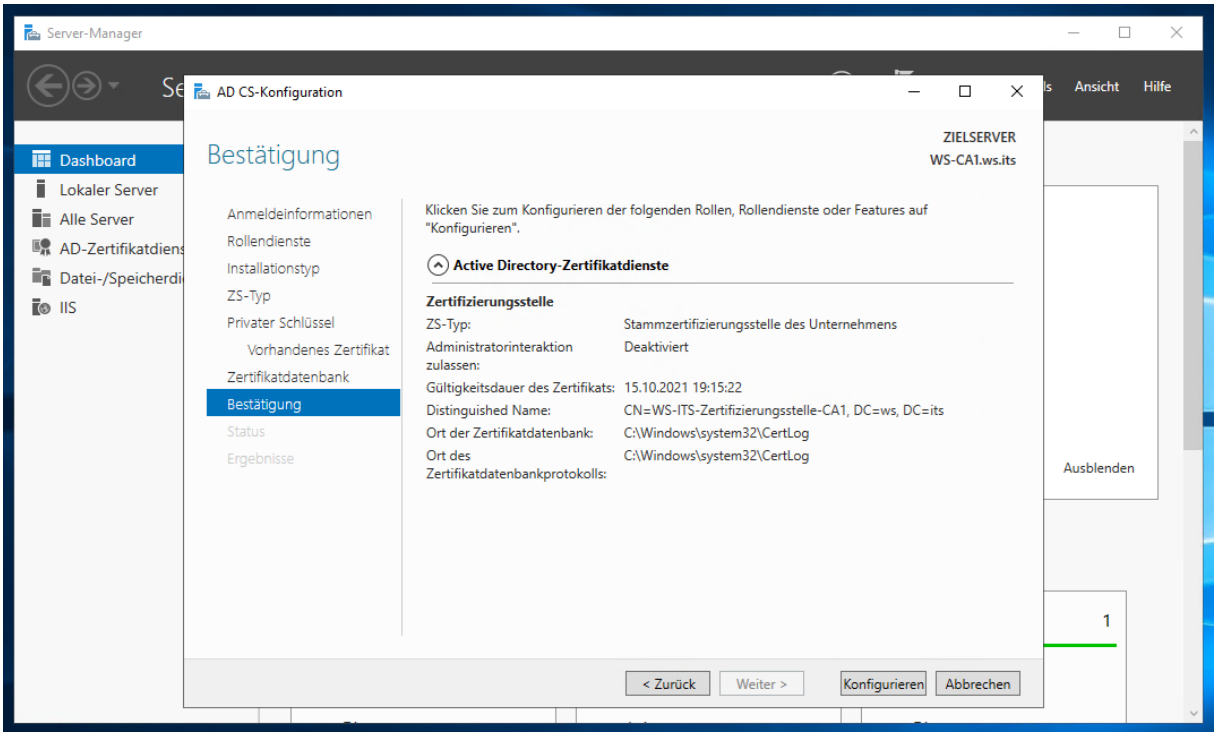




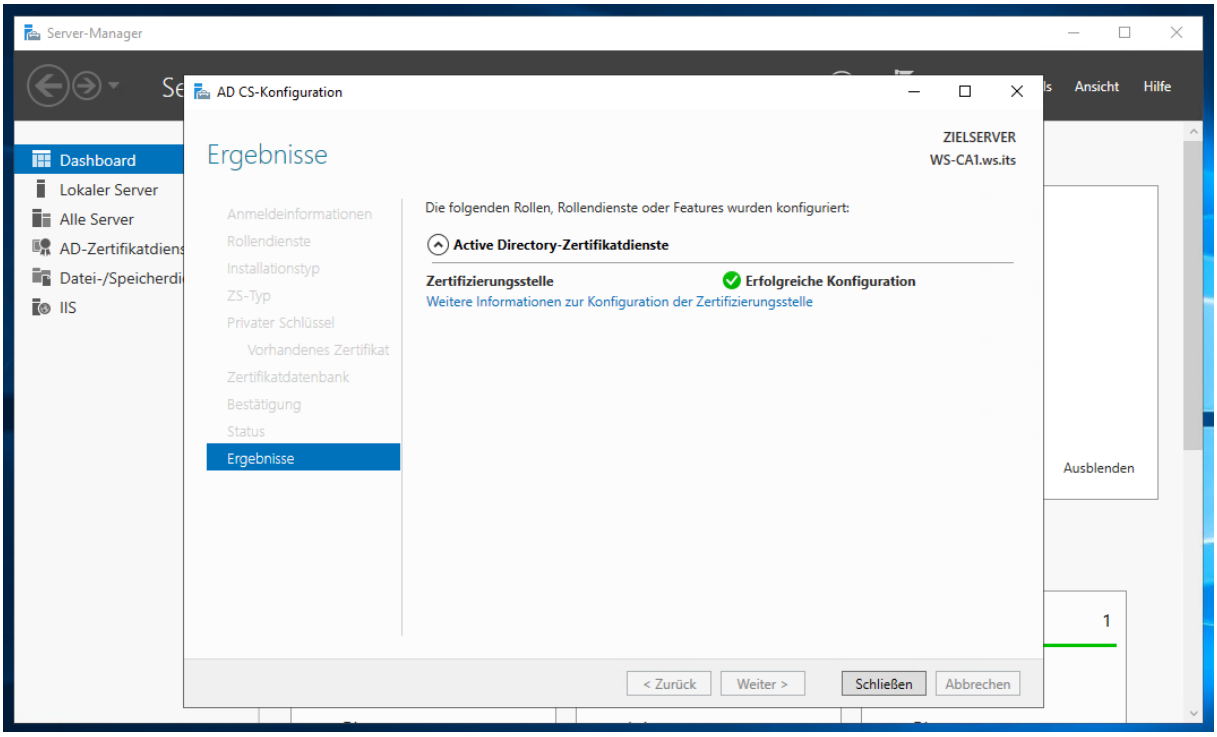
Die Pfade belasse ich im Default:



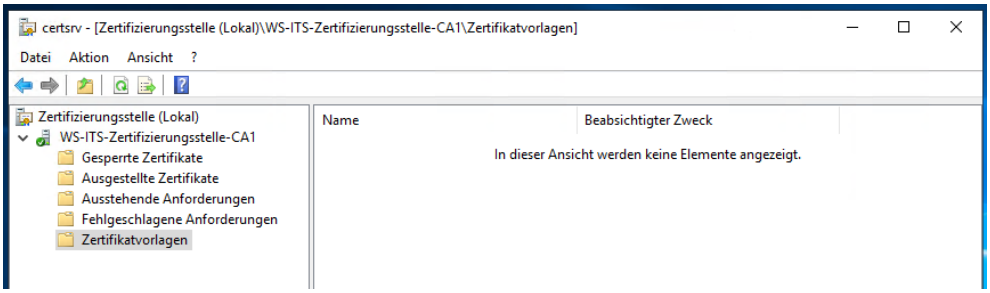
Das sieht gut aus. Der alte Name der CA wurde über das alte Zertifikat gefunden:



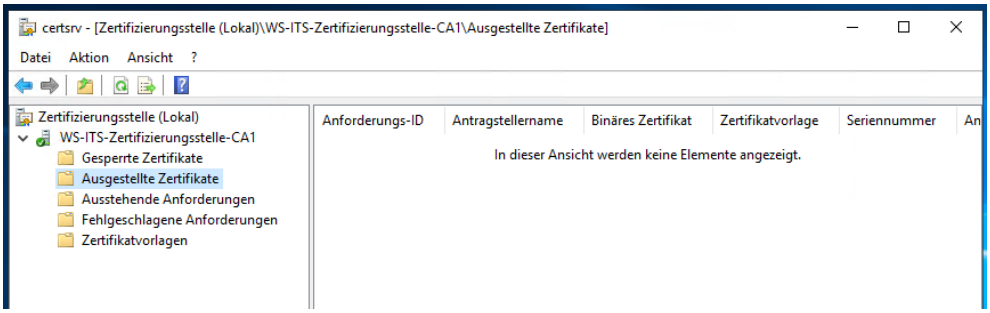
Das hat funktioniert:



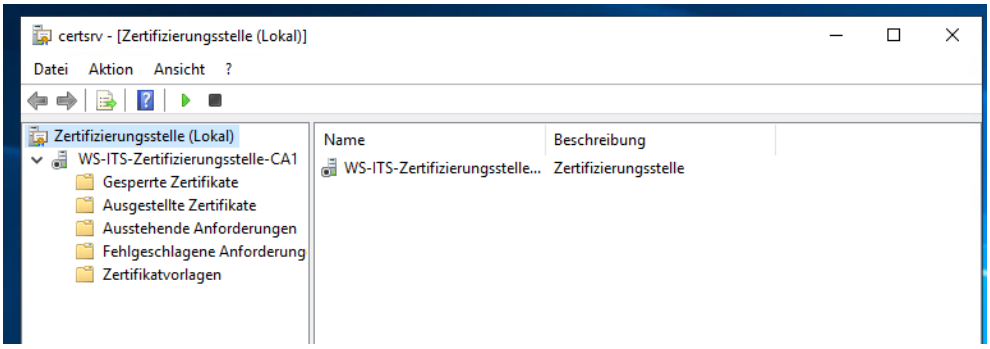
Bei Neuinstallationen werden ohne weitere Anpassungen die Standardvorlagen aktiviert. Da ich aber eine Migration ausführe und die Vorlagen vorab auf der alten Maschine entfernt hatte und die Identität wiederverwende, ist die Liste der auszustellenden Vorlagen leer. So werden dann keine Zertifikate versehentlich ausgestellt:



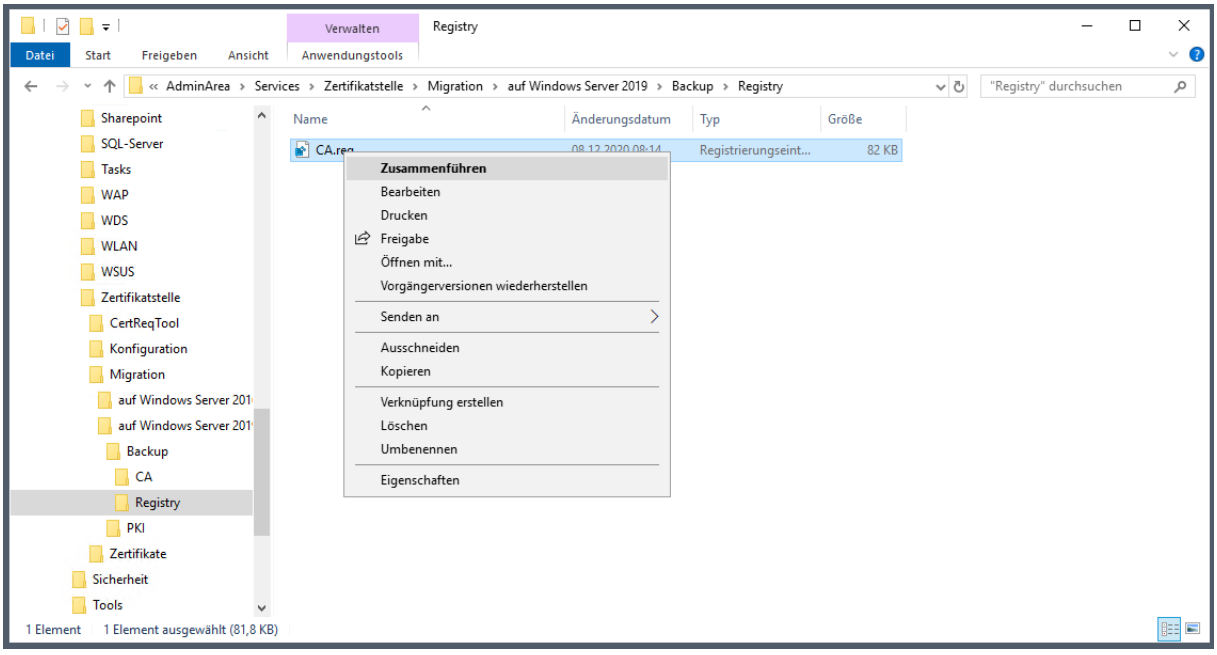
Die Datenbank der Windows CA wurde neu erstellt. Deshalb gibt es noch keine aktiven Zertifikate:

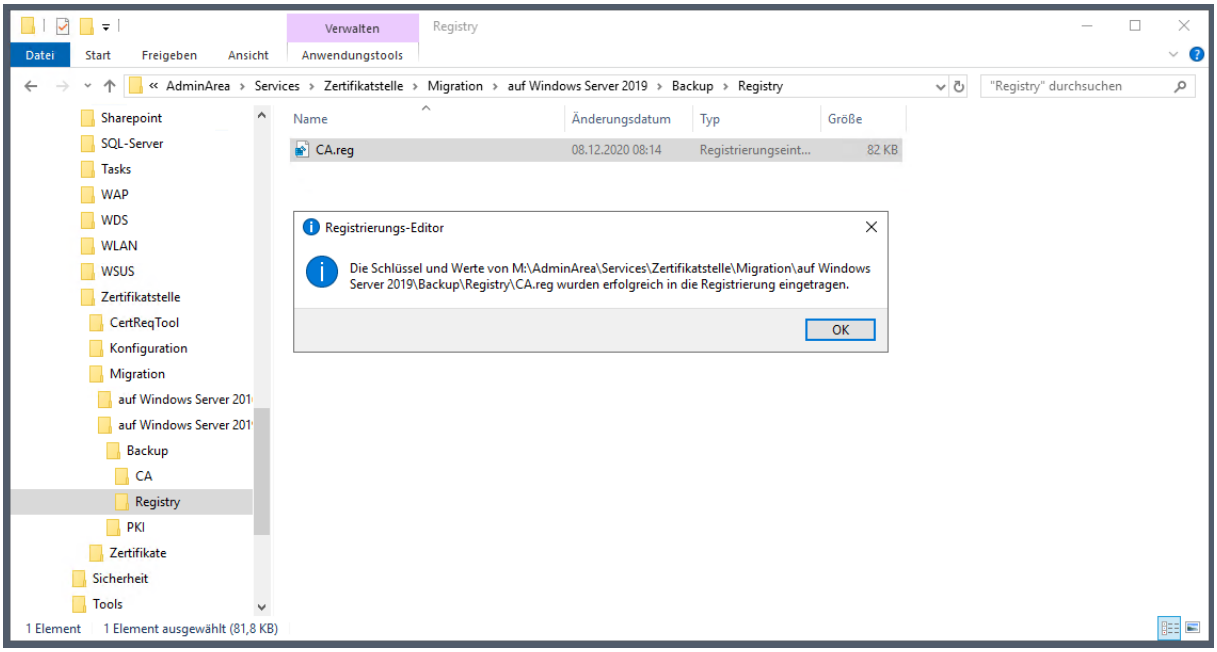


Nun deaktiviere ich den Service für die Wiederherstellung:

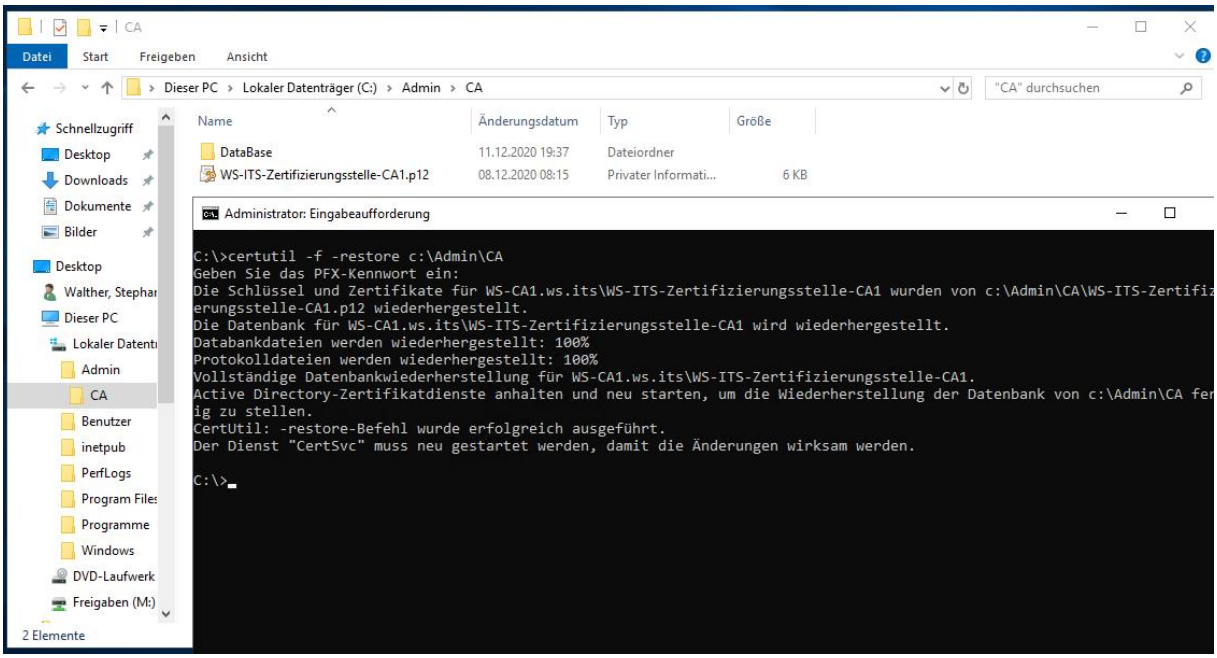


Die auf dem alten Server angepasste Konfiguration importiere ich nun auf dem neuen Server:

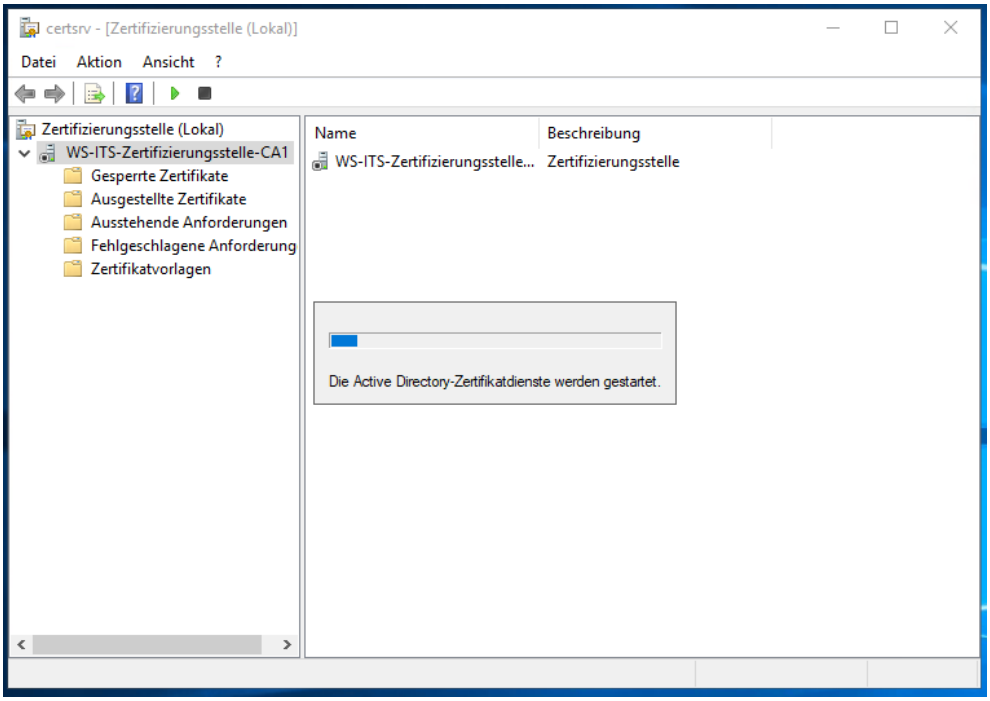




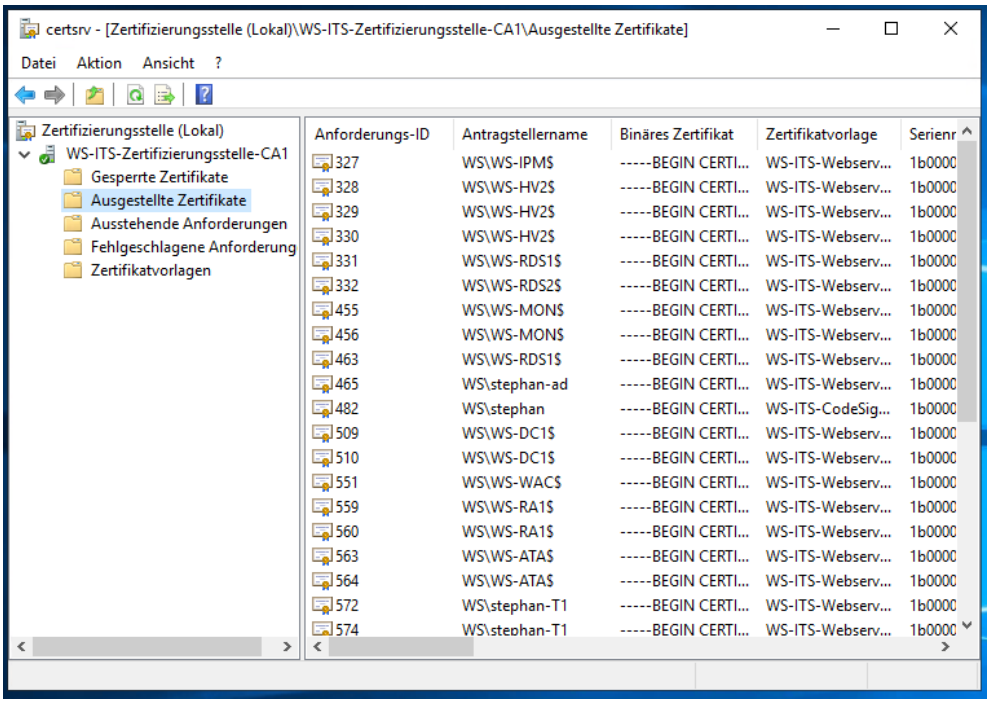
Die gesicherte Datenbank spiele ich mit certutil ein:



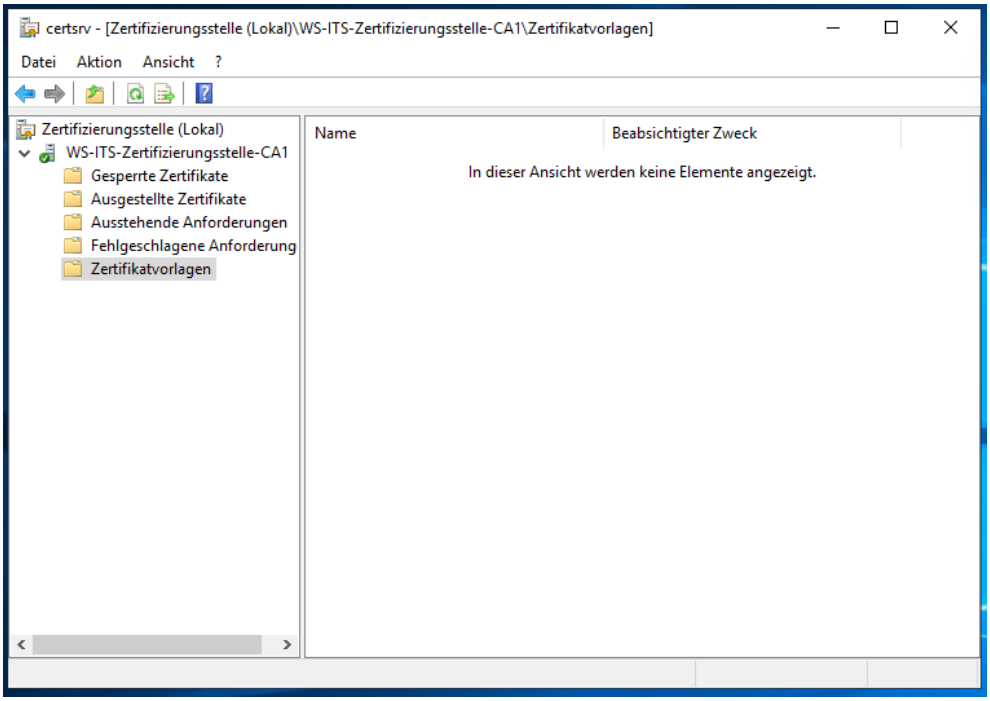
Jetzt kann der Services wieder starten:



Durch die Wiederherstellung wurde die leere Datenbank durch die alte DB ausgetauscht. Somit sind auch wieder alle alten Zertifikate im Bestand enthalten:

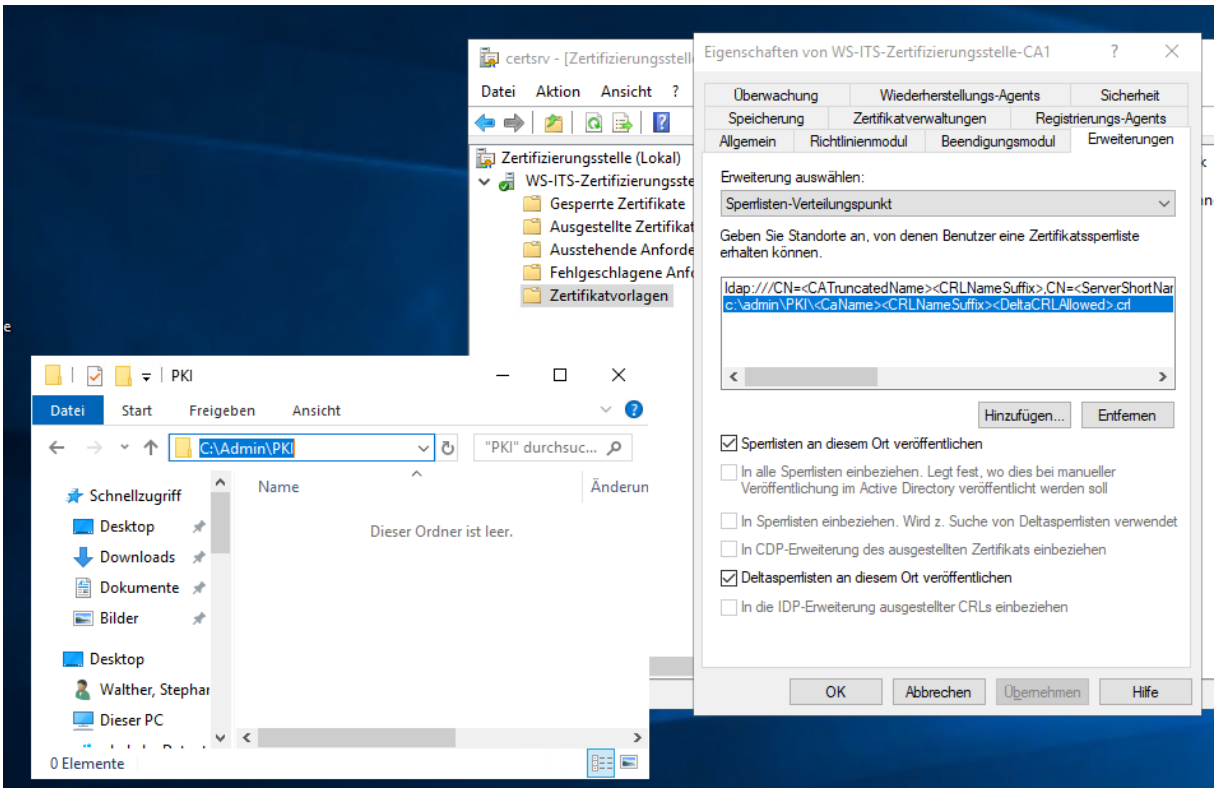


Die auszustellenden Vorlagen werden über ein Attribut im Active Directory zugewiesen. Die Wiederherstellung ist aber eine rein lokale Angelegenheit. Daher ist der Speicher der auszustellenden Vorlagen immer noch leer:



Anpassung und Überarbeitung der Sperrlistenverteilung

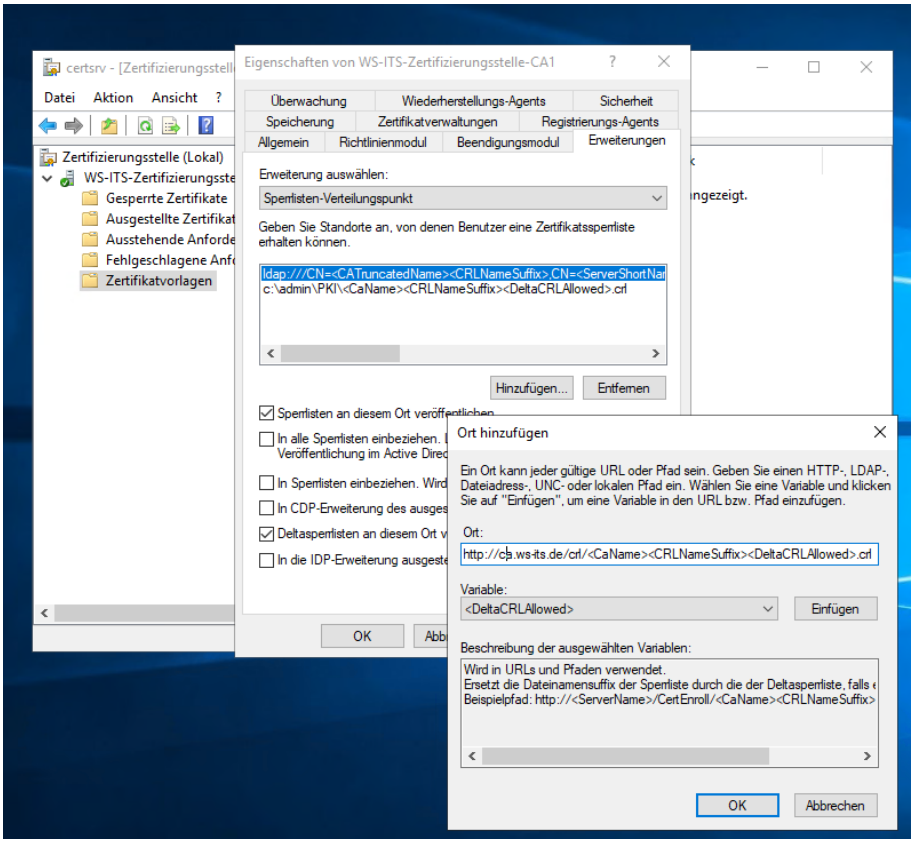
Nun kommen wir zu den Nacharbeiten und dem Feintuning. Für die Sperrlistenveröffentlichung habe ich in der Konfiguration ein lokales Verzeichnis angegeben. Dieses muss ich noch erstellen:



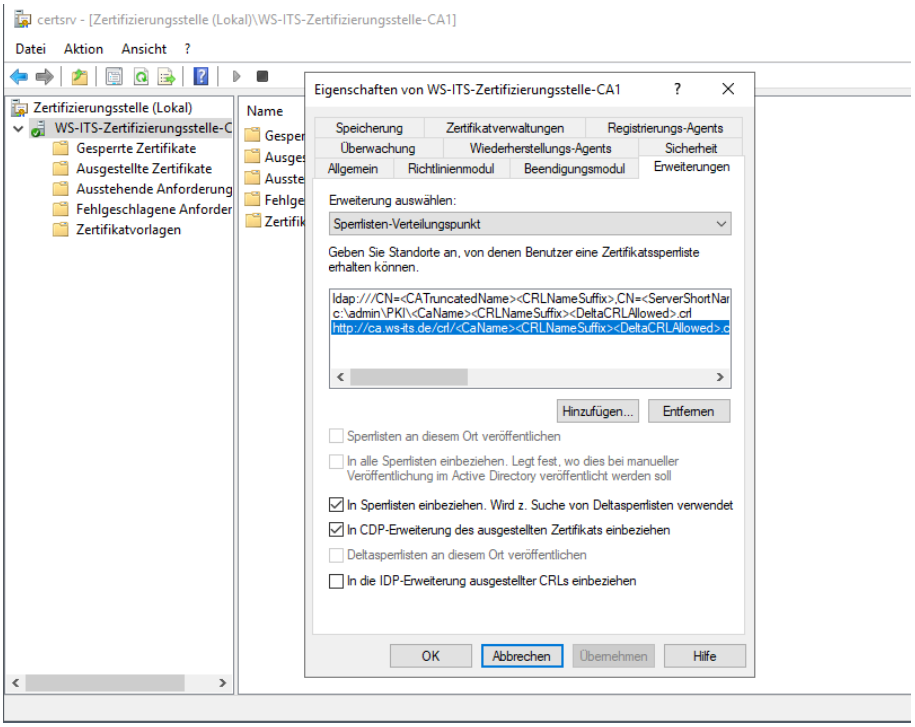
Der Verteilungspunkt LDAP ist weiter mit dabei, denn die bisher ausgestellten Zertifikate haben einen Verweis auf das LDAP. Nur neue Zertifikate werden nicht mehr über LDAP verifizierbar sein: Die Option „In CDP-Erweiterung des ausgestellten Zertifikates Einbeziehen“ ist nicht mehr aktiv.

Dennoch habe ich mit meinem 802.1x-Problem während der Downtime der CA festgestellt, dass ich auf eine Sperrlistenprüfung angewiesen bin. Bis hier wollte ich diese Funktion nur noch über OCSP bereitstellen. NPS wäre dazu auch in der Lage. Aber der OSCP-Server muss dafür immer online sein, denn der NPS wird die Antworten nicht cachen. In

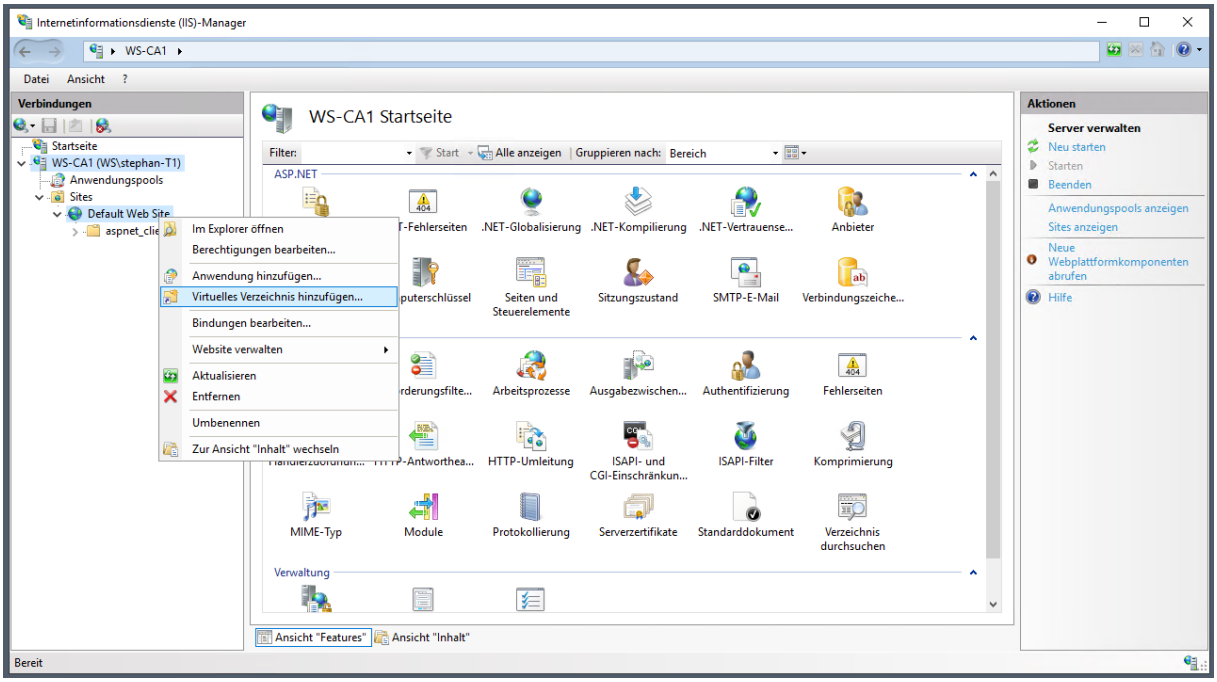
meinem Fall ist der OCSP-Service aber nicht hochverfügbar. Ein Ausfall würde also WLAN-Anmeldungen verhindern. Ich möchte jetzt aber keinen zweiten Server aufbauen. Also werde ich einen Sperrlistenverteilungspunkt über http erstellen. Dort kann sich mein NPS-Server die Sperrliste herunterladen, zwischenspeichern und somit WLAN-Zugriffsanfragen auf bei ausgefallener CA validieren. Den Verteilungspunkt lenke ich auf einen „öffentlichen FQDN“:



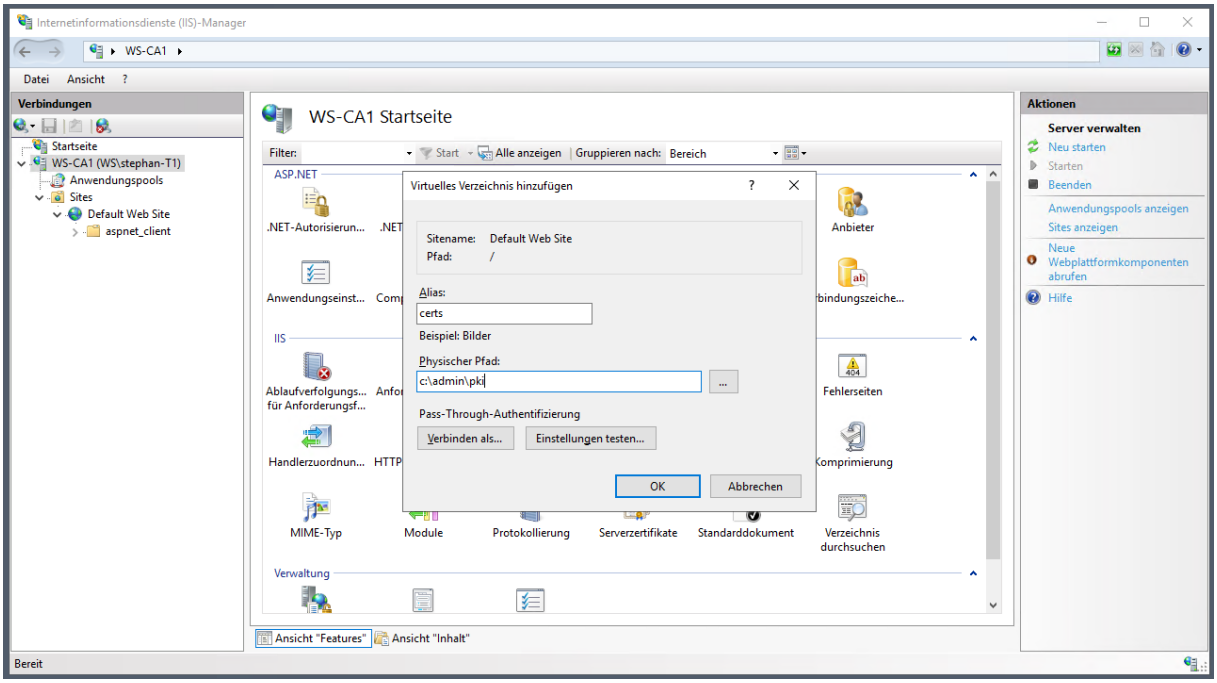
Die Optionen verankern den Verteilungspunkt in den neu ausgestellten Zertifikaten:



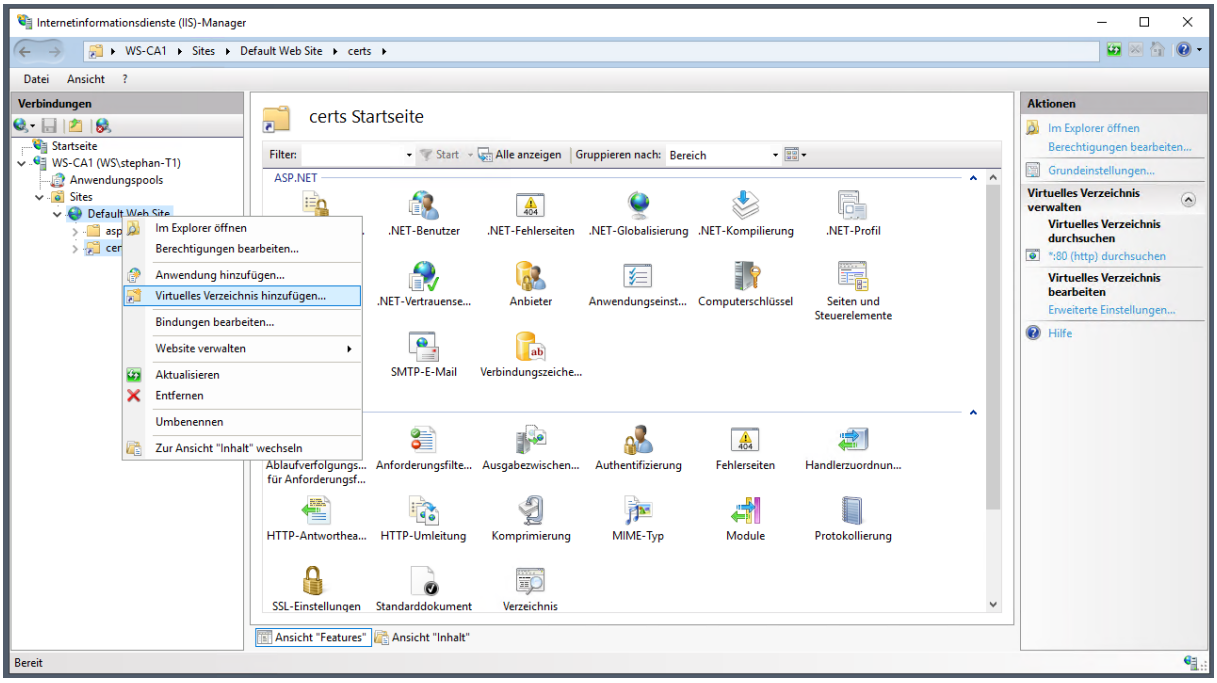
Die neue URL ca.ws-its.de soll auf meiner Windows CA aufsetzen. Daher erstelle ich im IIS-Manager ein neues virtuelles Verzeichnis „certs“ für die AIA-Stelleninformationen:



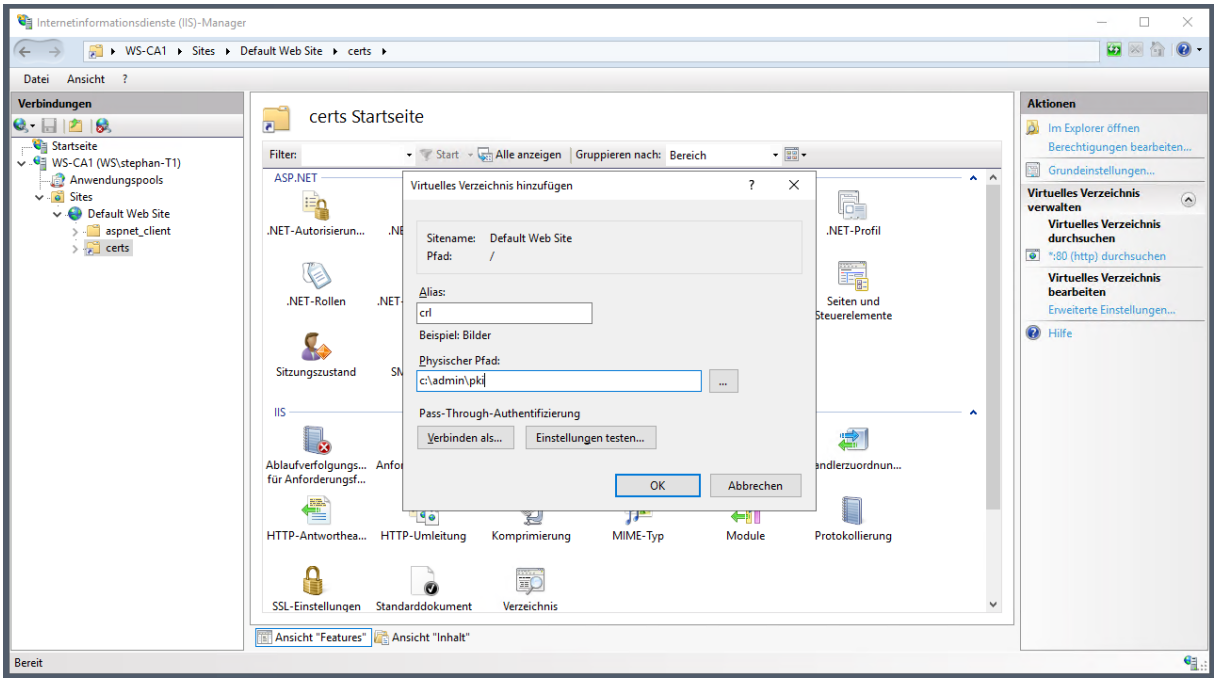
Dieses lasse ich auf den lokalen Speicherpfad zeigen, in dem meine CA ihre Sperrlistendateien und die Zertifikate bei der Veröffentlichung ablegt:



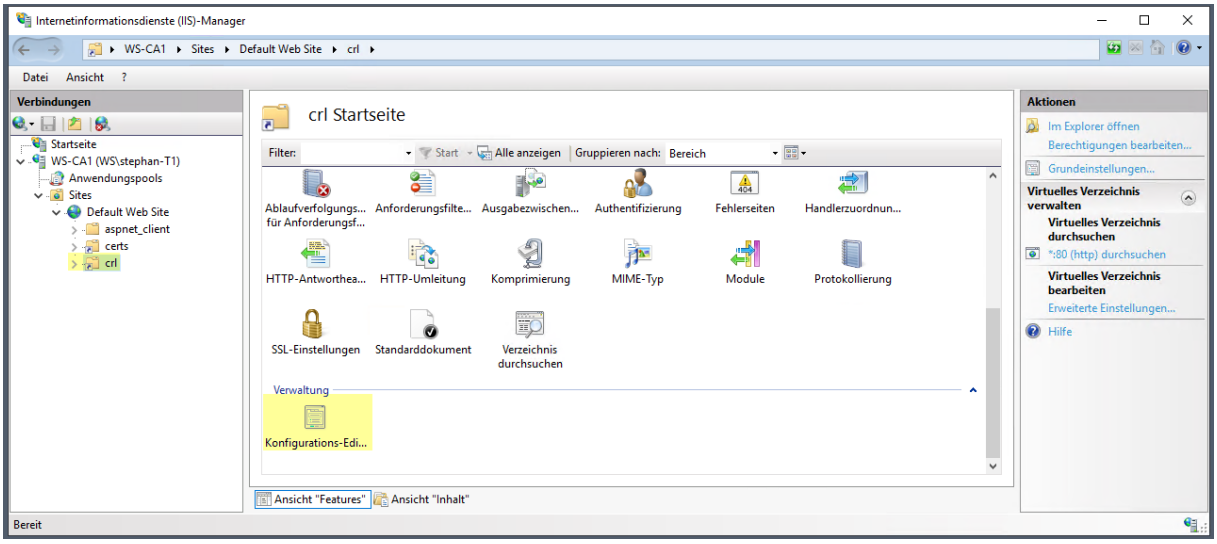
Ein weiteres virtuelles Verzeichnis „crl“ wird für die Sperrlistenveröffentlichung über http://ca.ws-its.de/crl benötigt:

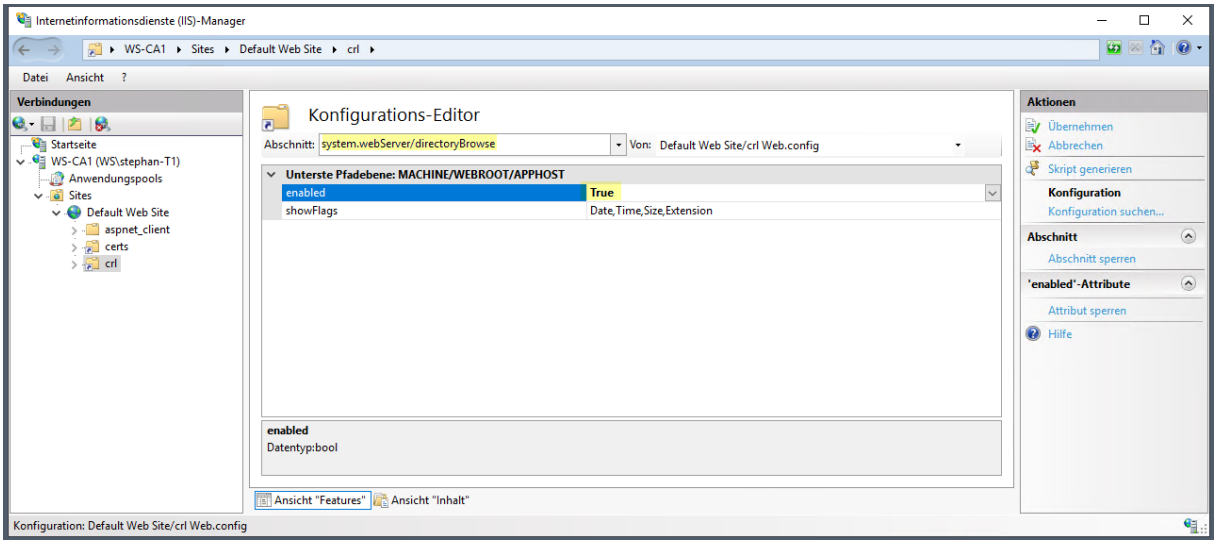


Auch dieses Verzeichnis zeigt auf den lokalen Ordner c:\admin\pki:

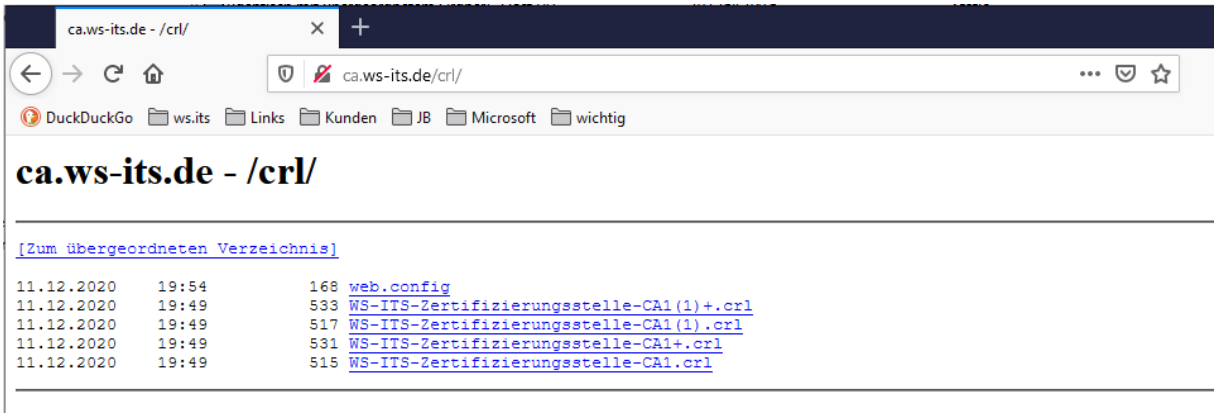


Ich aktiviere das Directory Browsing, damit ich den Inhalt des Verzeichnisses direkt im Browser ansehen kann:



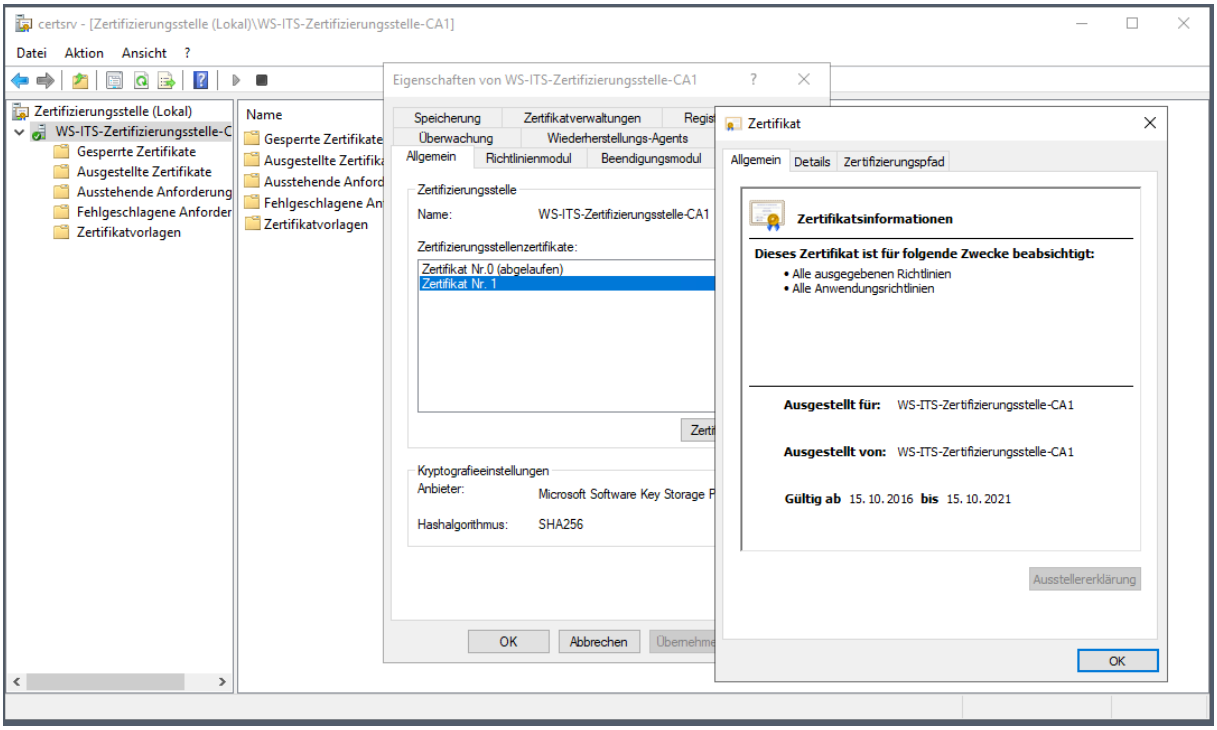


Da der FQDN in meinem DNS-Server bereits erstellt wurde und auf die interne IPv4 meiner Windows CA zeigt, kann ich das Verzeichnis einfach im Browser aufrufen und testen:

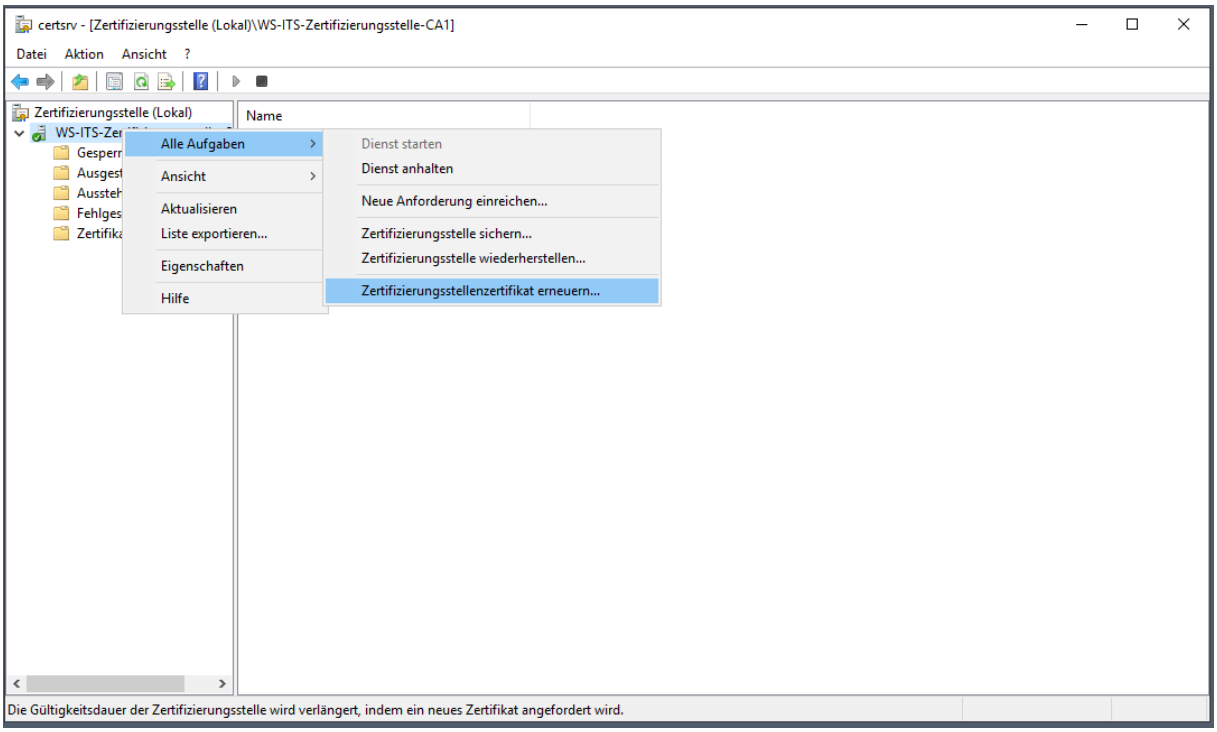


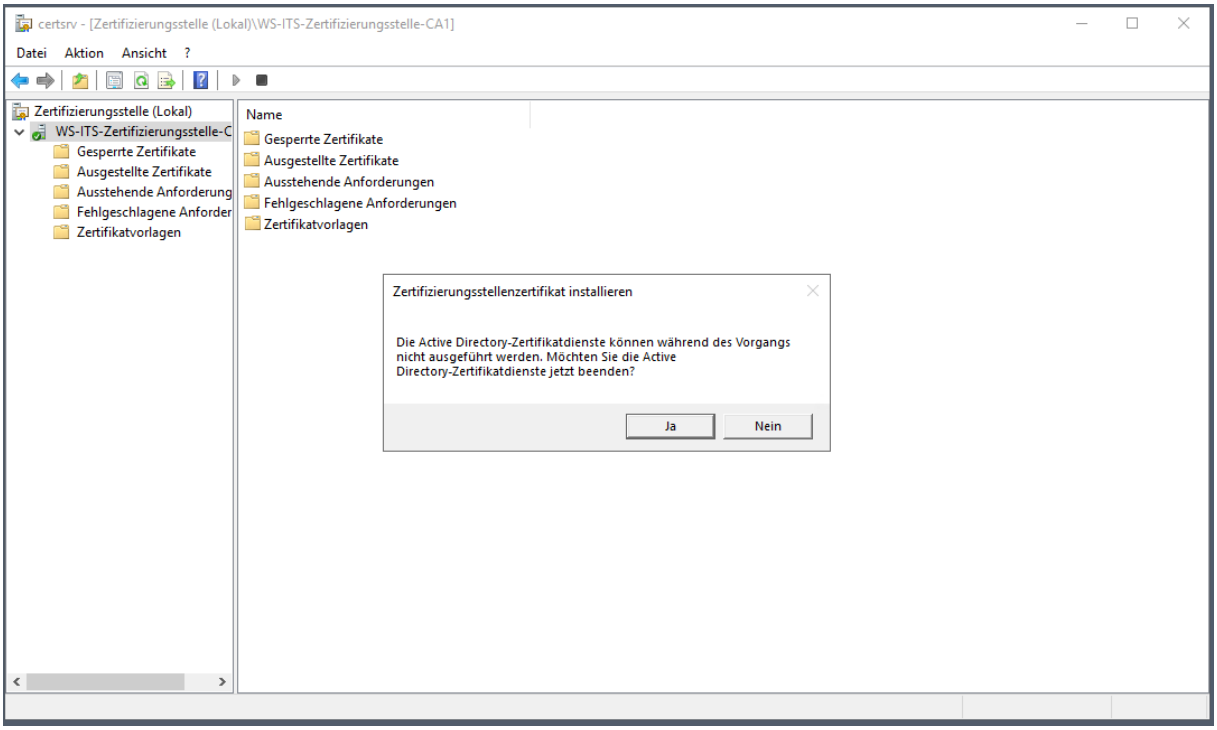
Grundkonfiguration und Ausstellung eines neuen Zertifizierungsstellen-Zertifikates

Es wird Zeit für ein neues Zertifizierungsstellen-Zertifikat, denn das bisherige läuft bald ab:

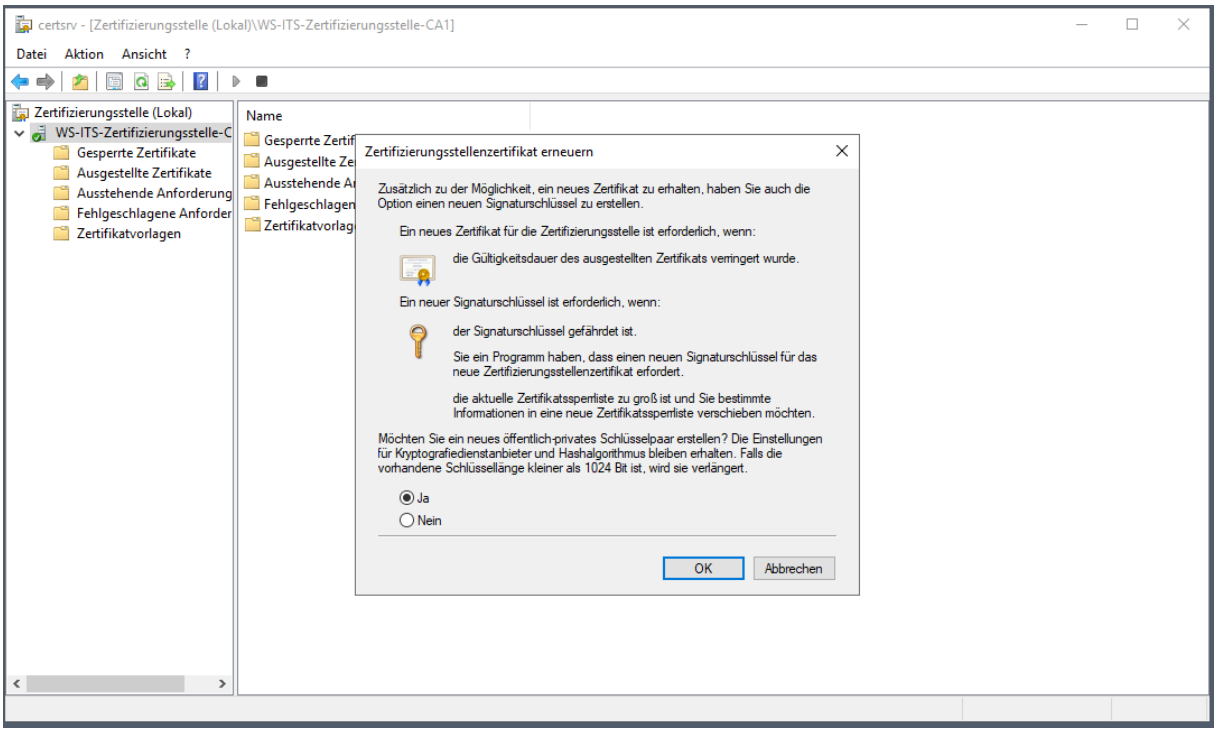


Der Vorgang kann einfach weil grafisch durchgeführt werden:

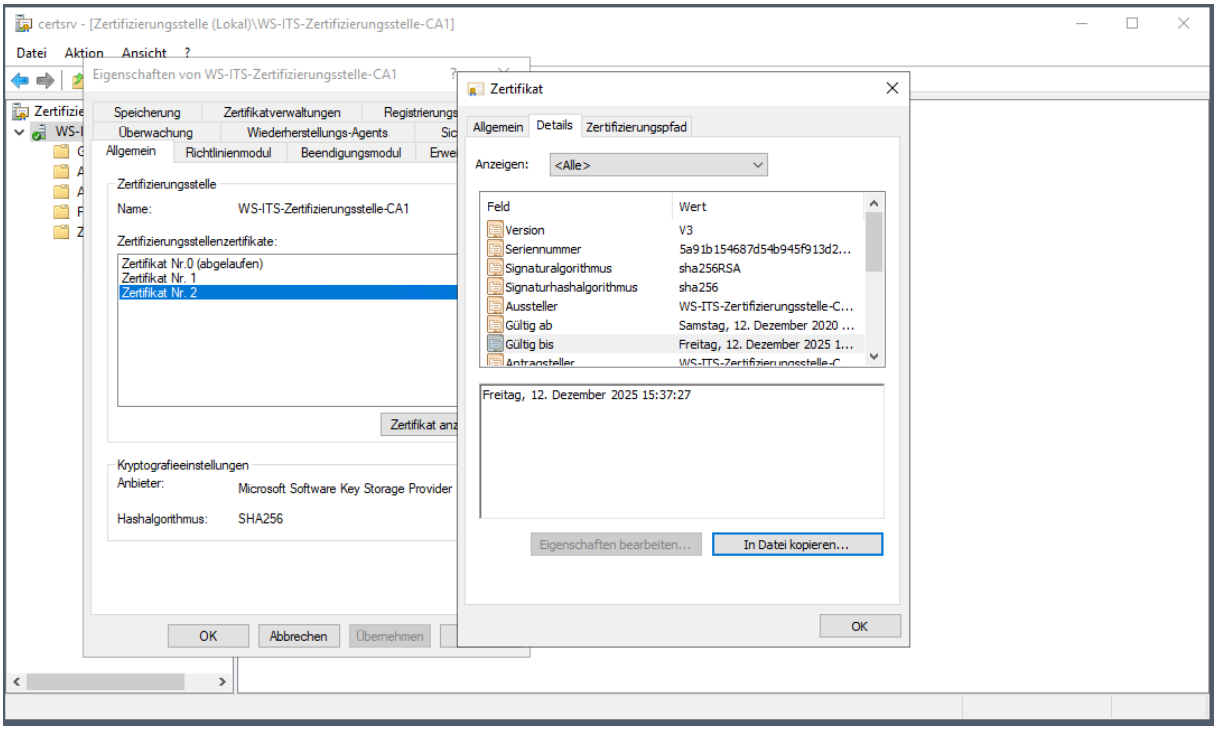




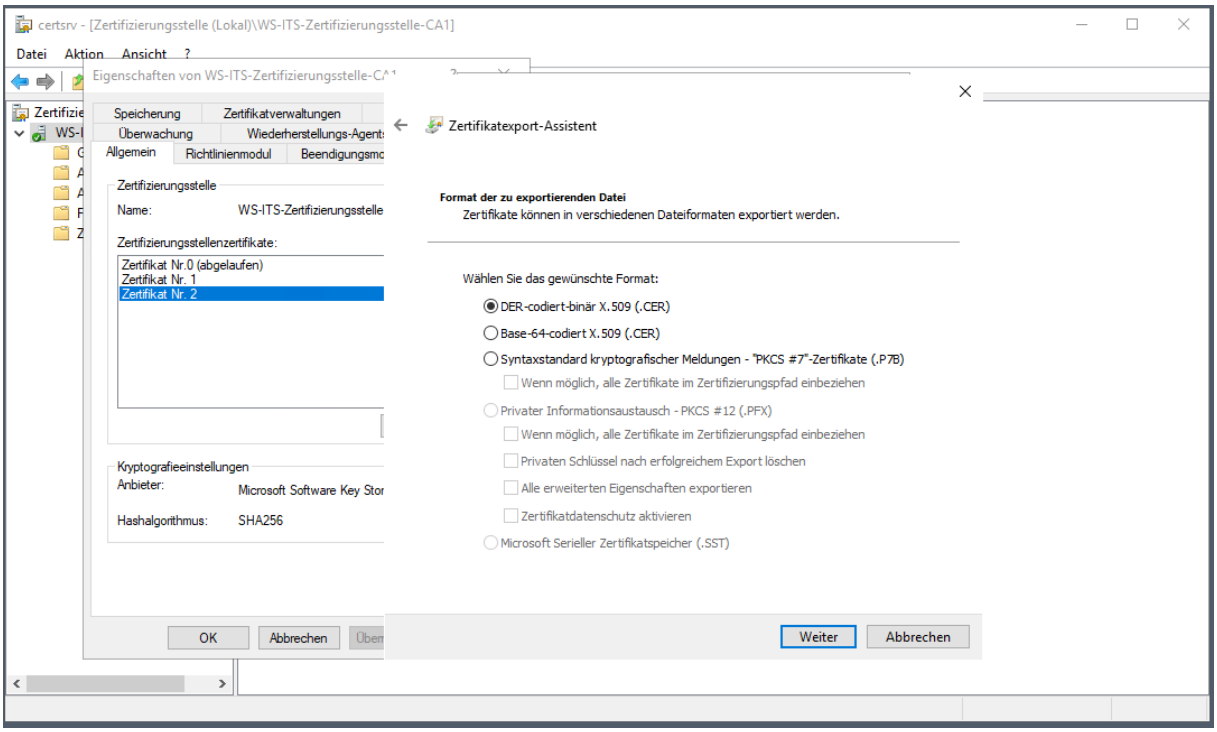
Ich erstelle dabei gleich ein neues Schlüsselpaar:



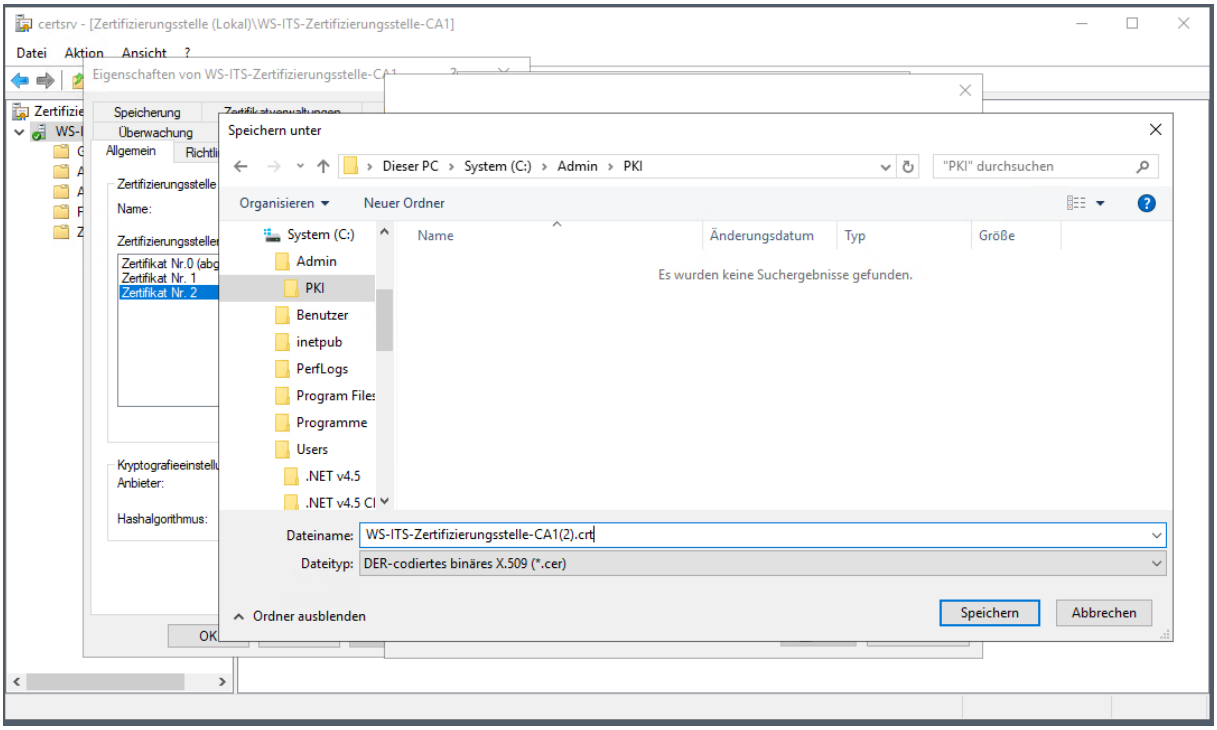
Nach Abschluss ist das neue Zertifikat im Speicher sichtbar. 2025 ist für eine interne Windows CA nicht viel, aber ich bin damit zufrieden.



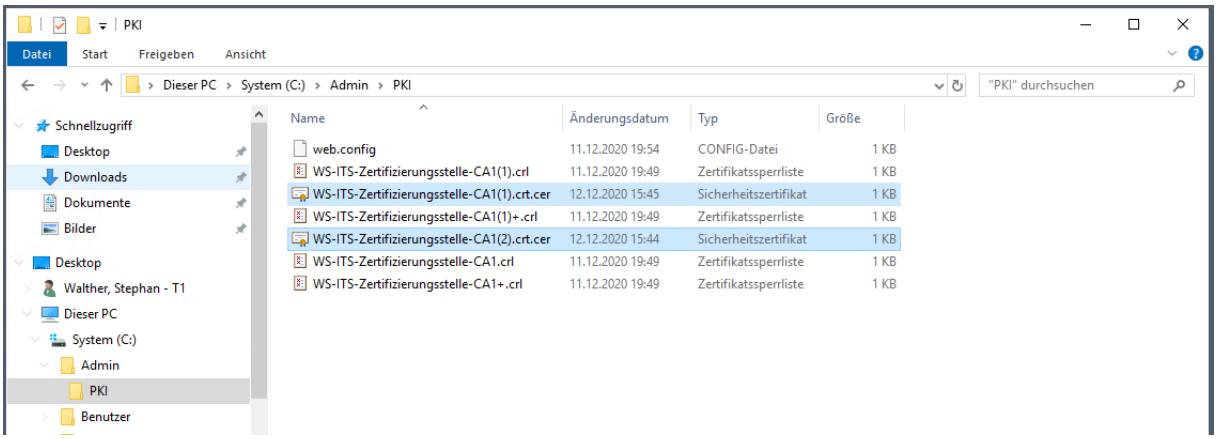
Das Zertifikat hat einen neuen öffentlichen Schlüssel. Diesen exportiere ich in eine Datei:



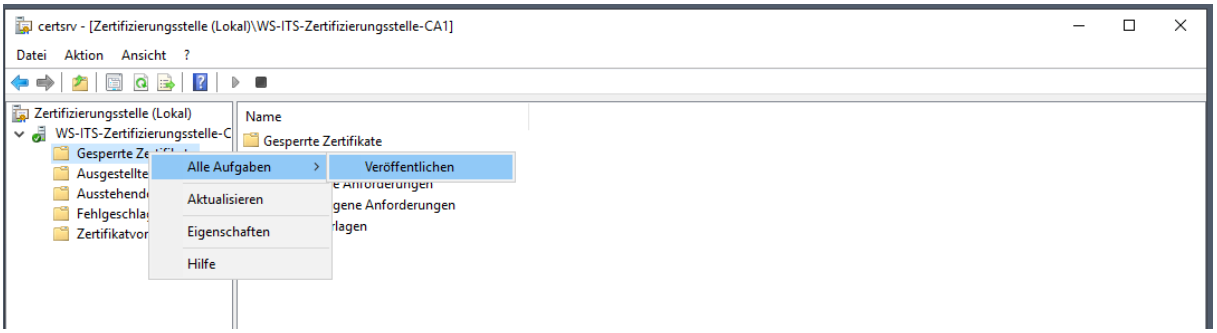
Diese Datei lege ich im AIA-Verzeichnis ab, denn hier könnte ein Client danach suchen:

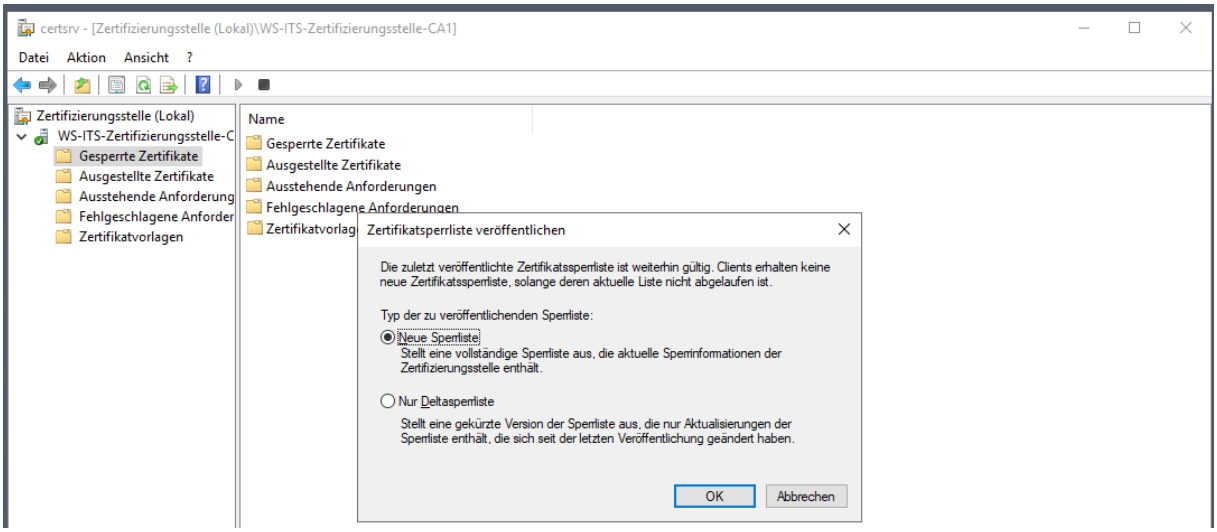


Achtung: Die Dateiendung wurde nicht korrekt erstellt und muss korrigiert werden:

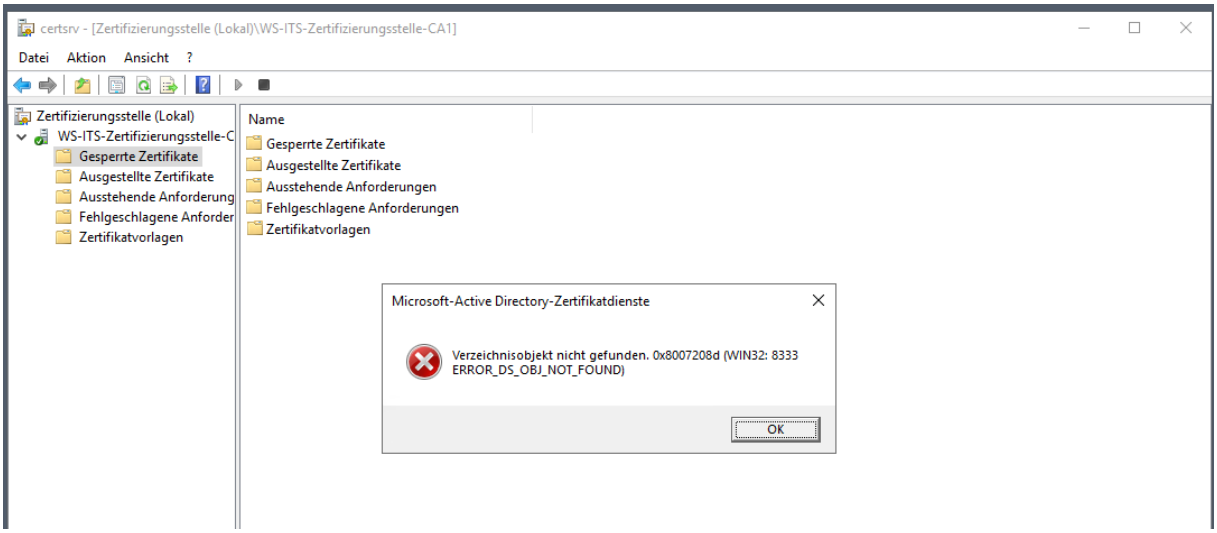


Nun erstelle ich eine neue Sperrliste für die beiden gültigen Zertifizierungsstellen-Zertifikate:

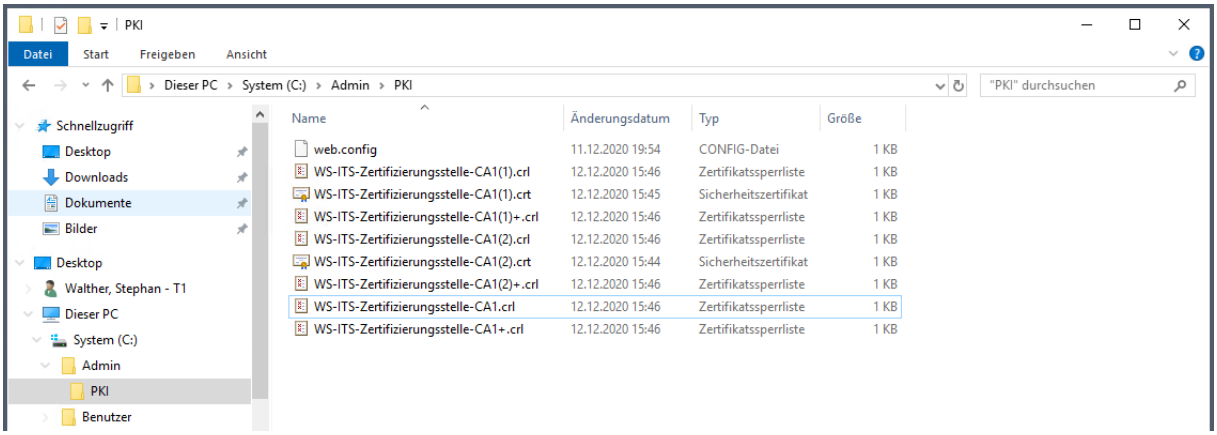




OK, das scheint noch nicht zu funktionieren. Im Active Directory gab es keine Veröffentlichung:



Im Dateisystem dagegen ist die neue Sperrliste erstellt worden:

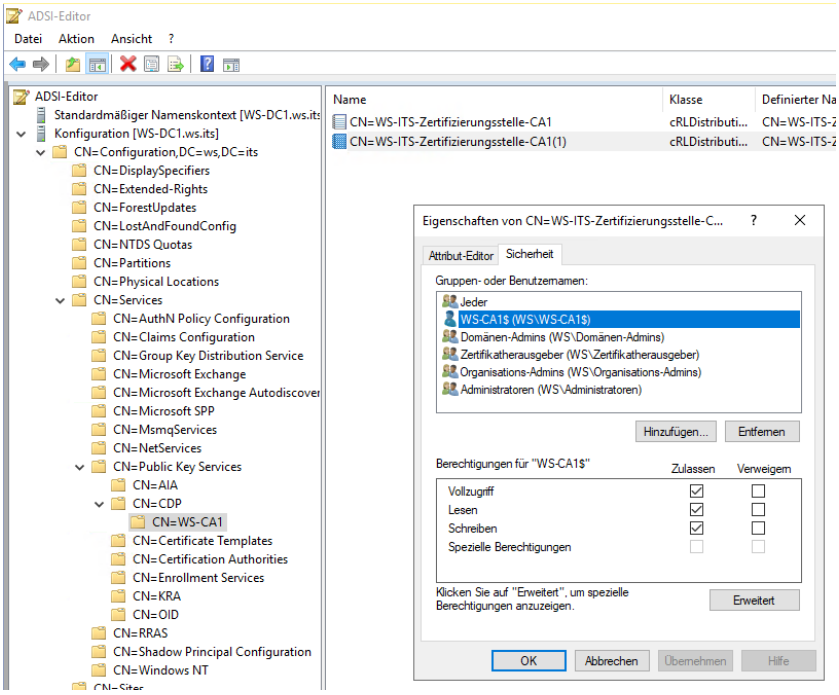


Also ist es ein reiner Fehler im Active Directory…

TroubleShooting – Korrektur des LDAP-Sperrlistenverteilungspunktes

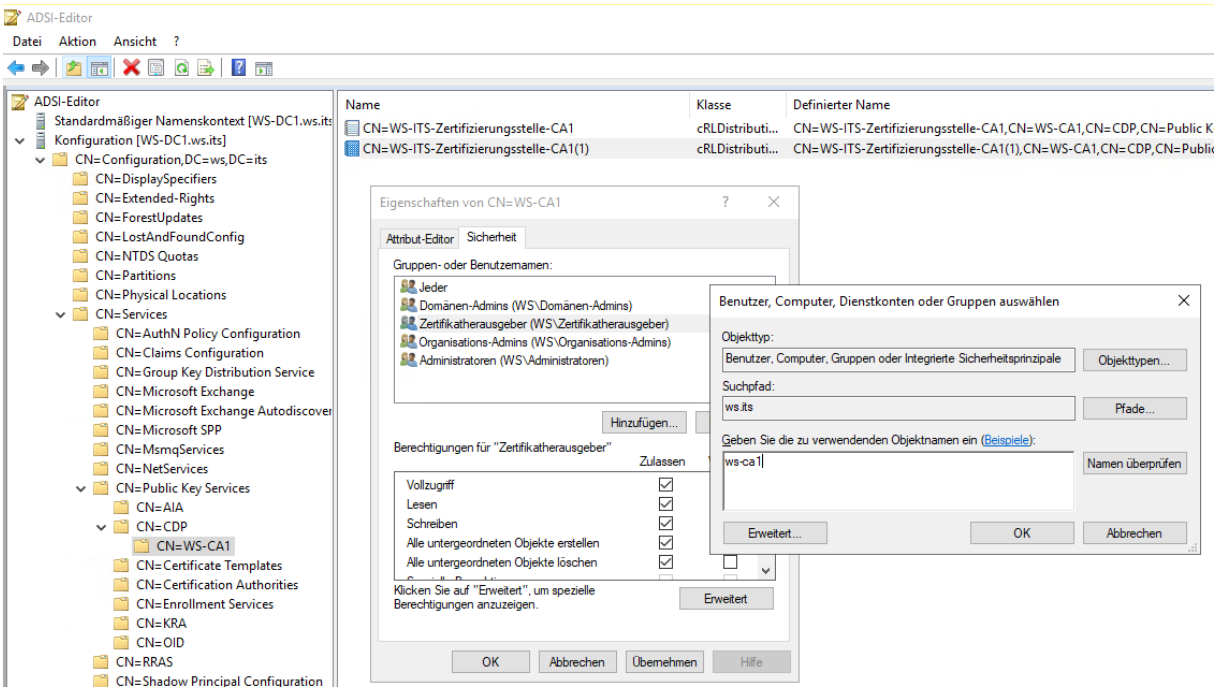
Im Active Directory veröffentlicht meine Windows Zertifizierungsstelle in der Konfigurationspartition ihre Sperrlisten. Hier finde ich 2 Sperrlisten – je eine pro altes Zertifizierungsstellen-Zertifikat. Aber für das neue Zertifikat fehlt der Eintrag! Stimmen hier vielleicht die Berechtigungen nicht?

Die Berechtigung für das Bearbeiten der beiden alten Sperrlisten ist an den Computer-Account gebunden:

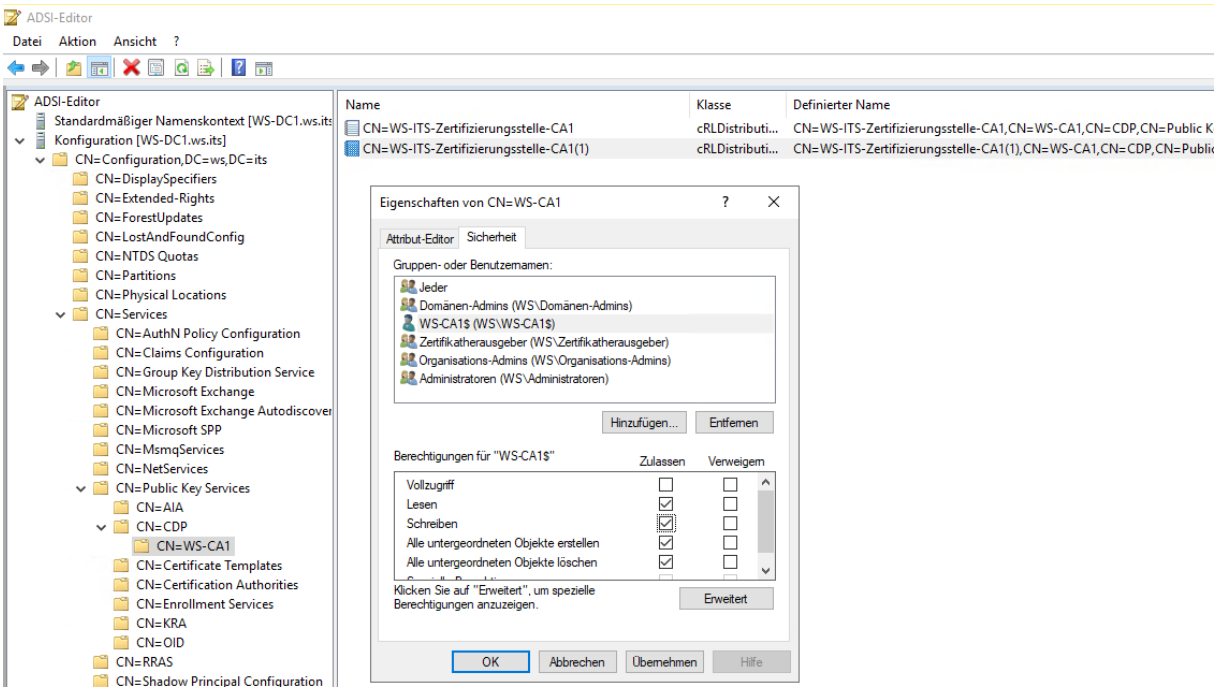


Aber auf dem darüberliegenden Container „CN=WS-CA1“ fehlt die Berechtigung! Normal wird der Record einer neuen Sperrliste durch den Benutzer-Account des Administrators automatisch erzeugt und dann wird das Recht automatisch an den Computer-Account delegiert. Aber meine Admin-Kennung ist kein Mitglied der Gruppe Enterprise-Admins, wie es eigentlich üblich ist. Daher wurde der Record nicht erstellt und die Windows CA kann ihn danach auch nicht aktualisieren.

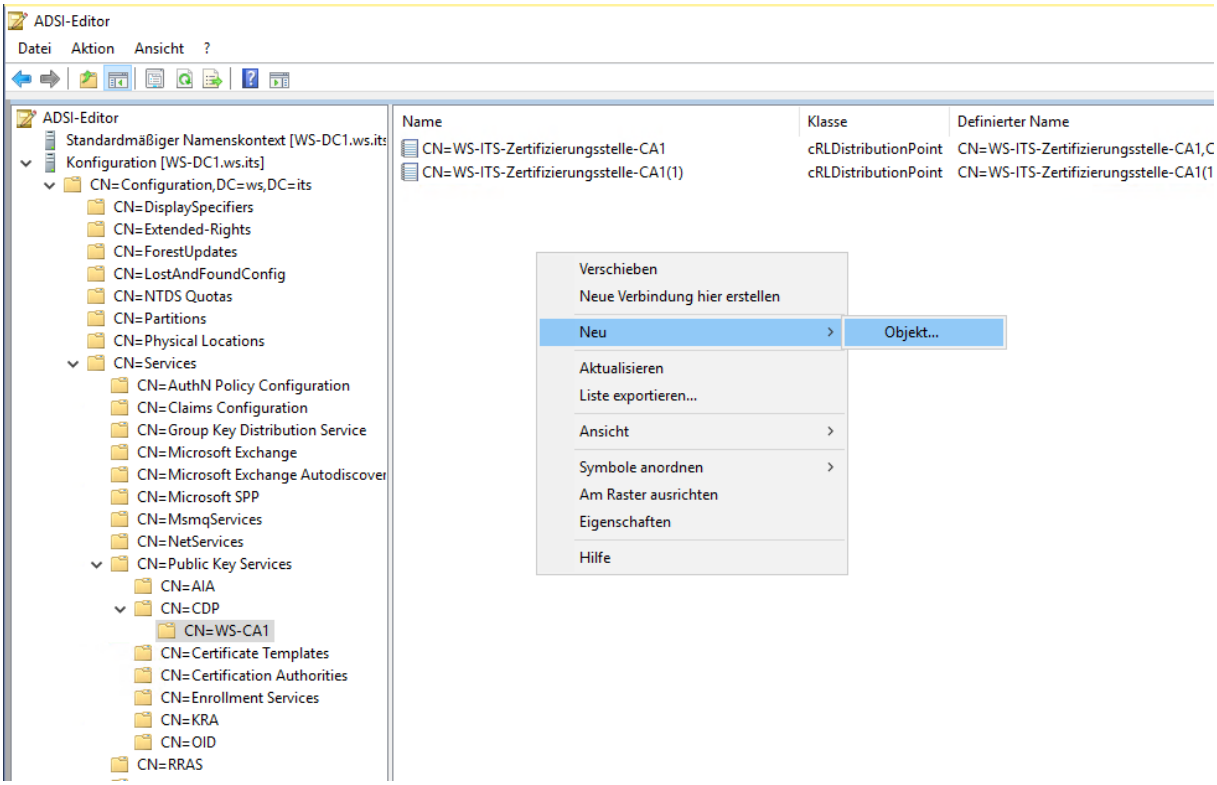
Der Versuch, die Berechtigung auf dem Container direkt an die Windows CA bzw. an den Computer-Account zu delegieren wird nicht ausreichen. Aber dennoch wage ich einen Versuch und nehme den Computer-Account in die ACL auf:



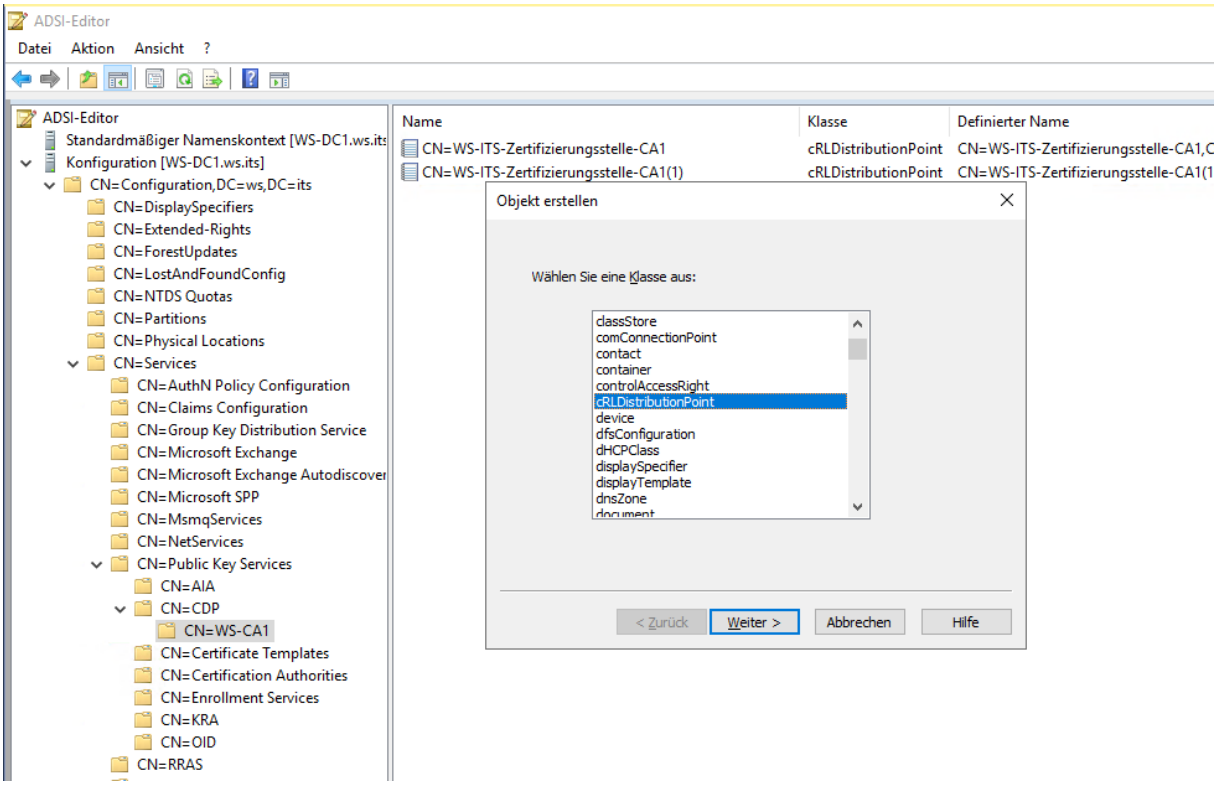
Denn hier kann ich auch das Recht auf untergeordnete Objekte delegieren:



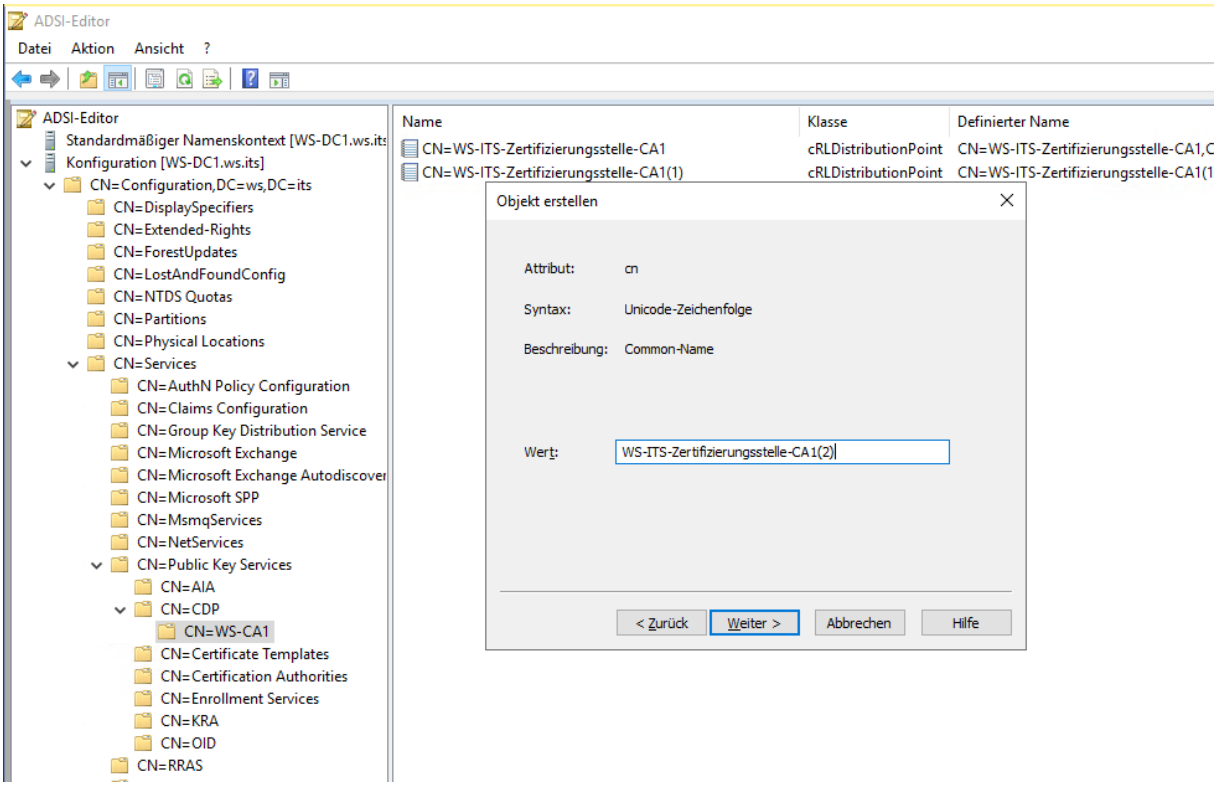
Nun muss ich nur noch ein cRLDistributionPoint-Object erzeugen – das würde ein Admin mit Enterprise-Permission automatisch bei Ausstellen eines neuen Zertifizierungsstellen-Zertifikates in der Windows CA mit erledigen:



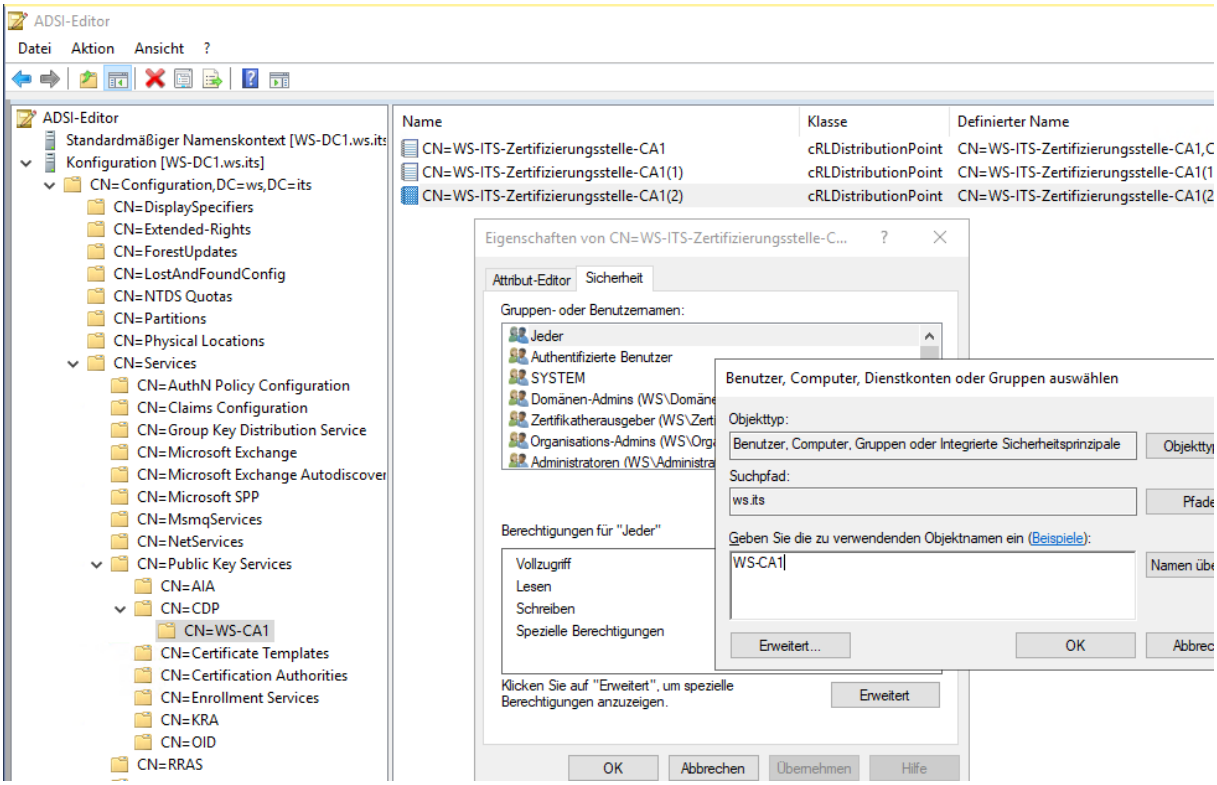
Der Datentyp ist leicht zu finden:

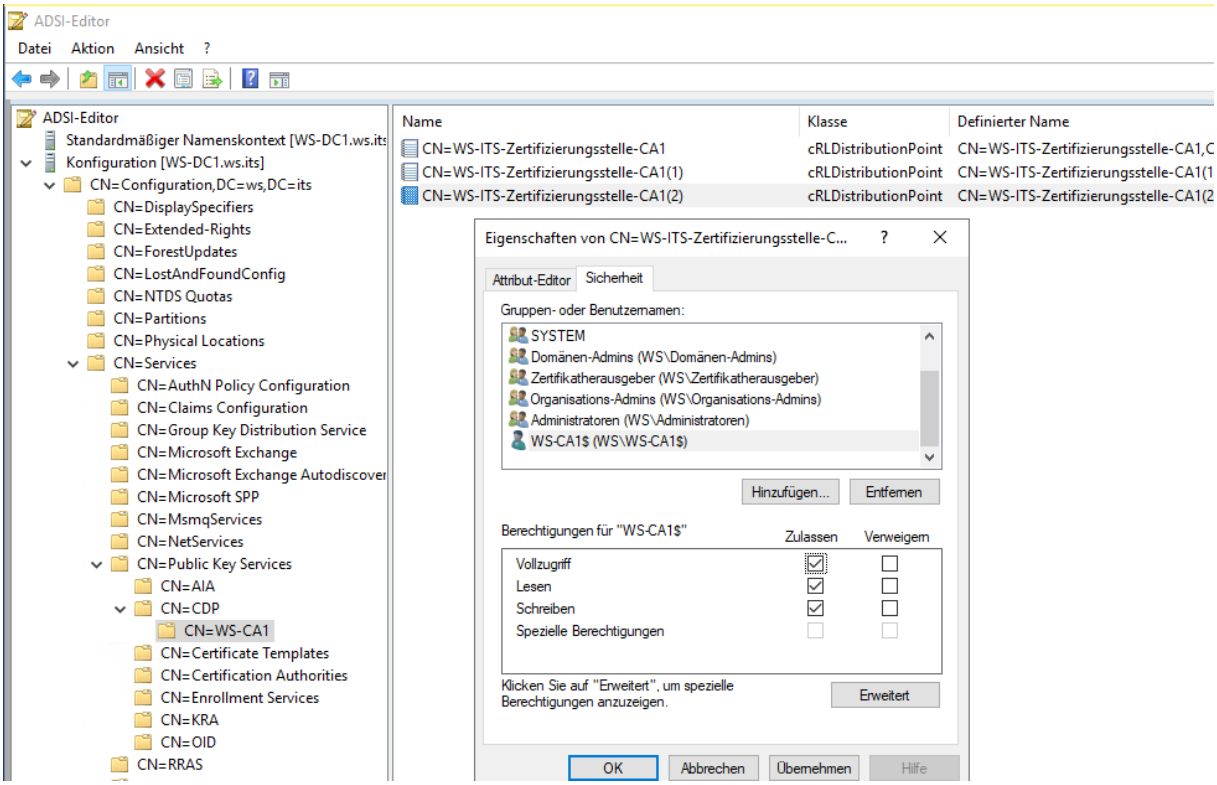


Der Name des neuen CRL-Objektes muss sich an dem Schema der Benennung orientieren. Das neue Zertifizierungsstellen-Zertifikat hat den Zähler (2). Der muss hier mit aufgenommen werden:

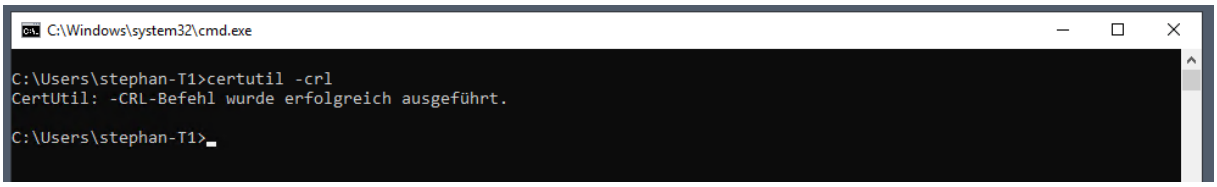


Ein Blick in die ACL der neuen CRL zeigt, dass meine Windows CA nicht berechtigt wurde. Egal, dann füge ich den erforderlichen Eintrag manuell ein:



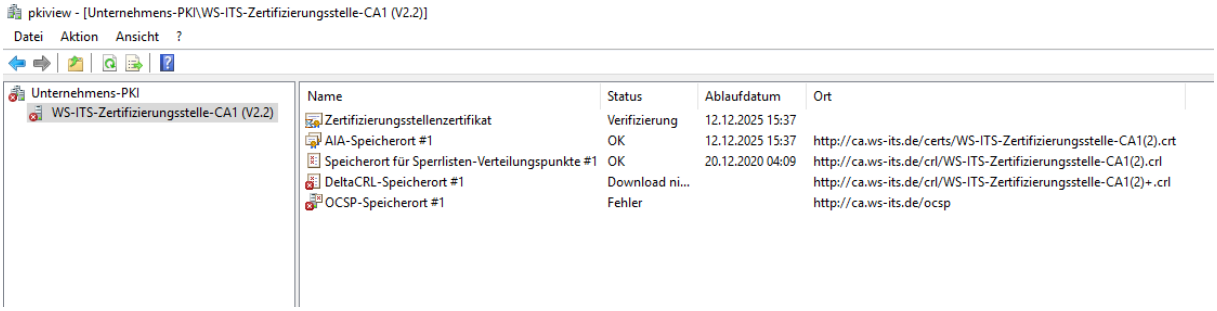


Nun starte ich auf meinem CA-Server die CRL-Generierung erneut – das geht auch über die cmd. Und jetzt ist der Vorgang erfolgreich:

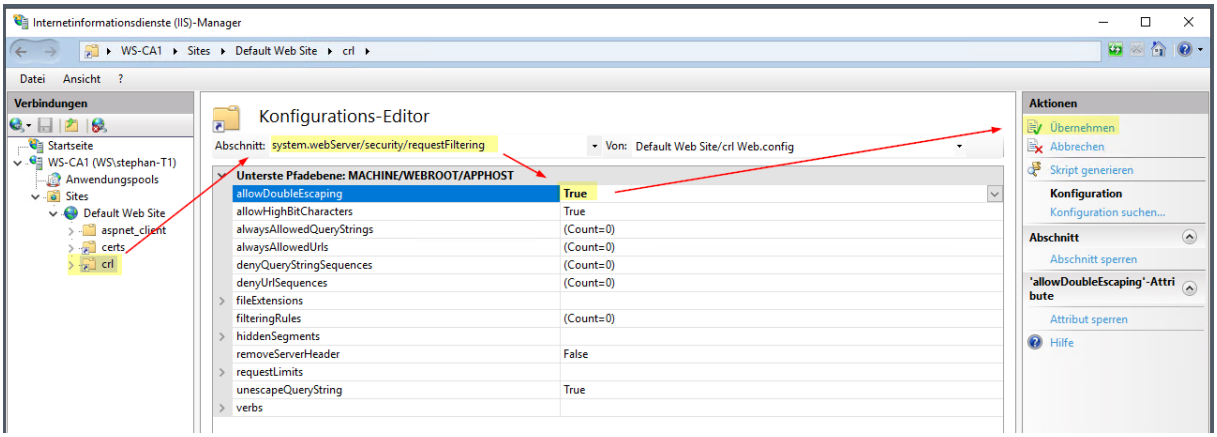


Fortsetzung der Grundkonfiguration

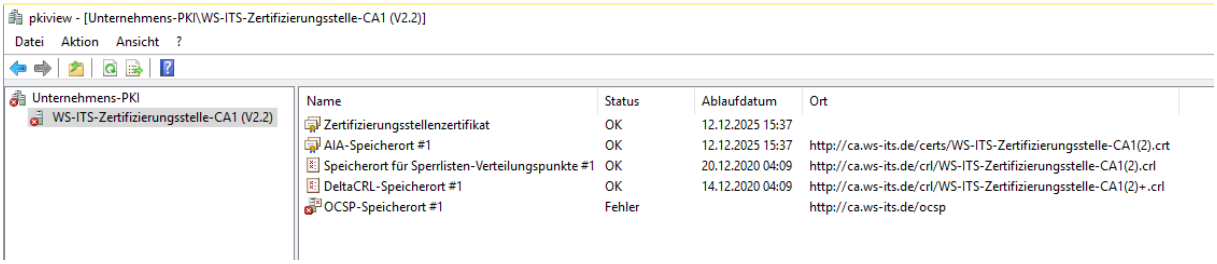
Jede Anpassung an den Zertifizierungsstellen-Zertifikaten, den AIA-Positionen oder den Sperrlisten sollte mit PKIVIEW kontrolliert werden. Ich starte die msc. Das Ergebnis überrascht mich nicht. Es gibt noch Korrekturbedarf bei der Deltasperrliste und OCSP ist noch nicht installiert:



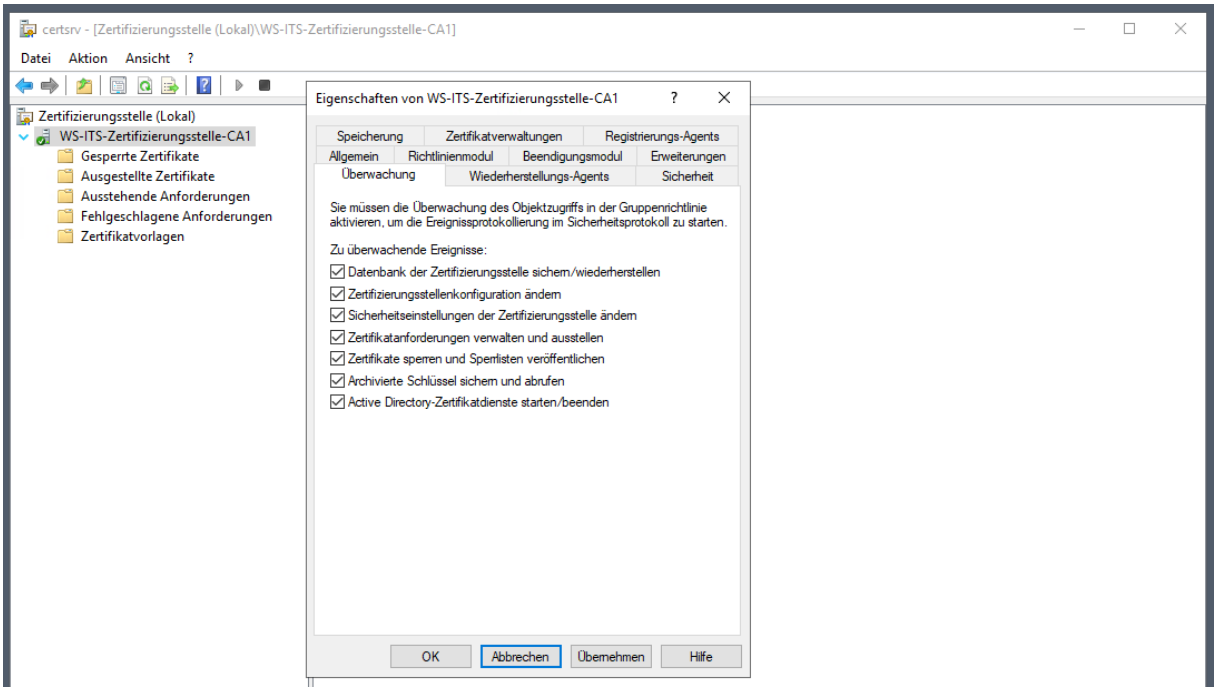
Die Deltasperrliste enthält ein plus-Zeichen. Dieses Sonderzeichen ist in einer URL nicht gerne gesehen. Ein IIS kann das nur mit dem sogenannten „Double-Escaping“. Das Feature muss manuell im IIS-Manager aktiviert werden:



Nach einer PKIVIEW-Aktualisierung ist die Deltasperrliste erreichbar:



Nun schließe ich noch einige andere Konfigurationen ab. Dazu gehört die Aktivierung des Loggings. Eine erforderliche Voraussetzung ist die Aktivierung und die Konfiguration des Advanced Audits. Diese habe ich global über Gruppenrichtlinien bereits abgeschlossen. So fehlen nur noch einige Haken in den Properties meiner neuen CA:

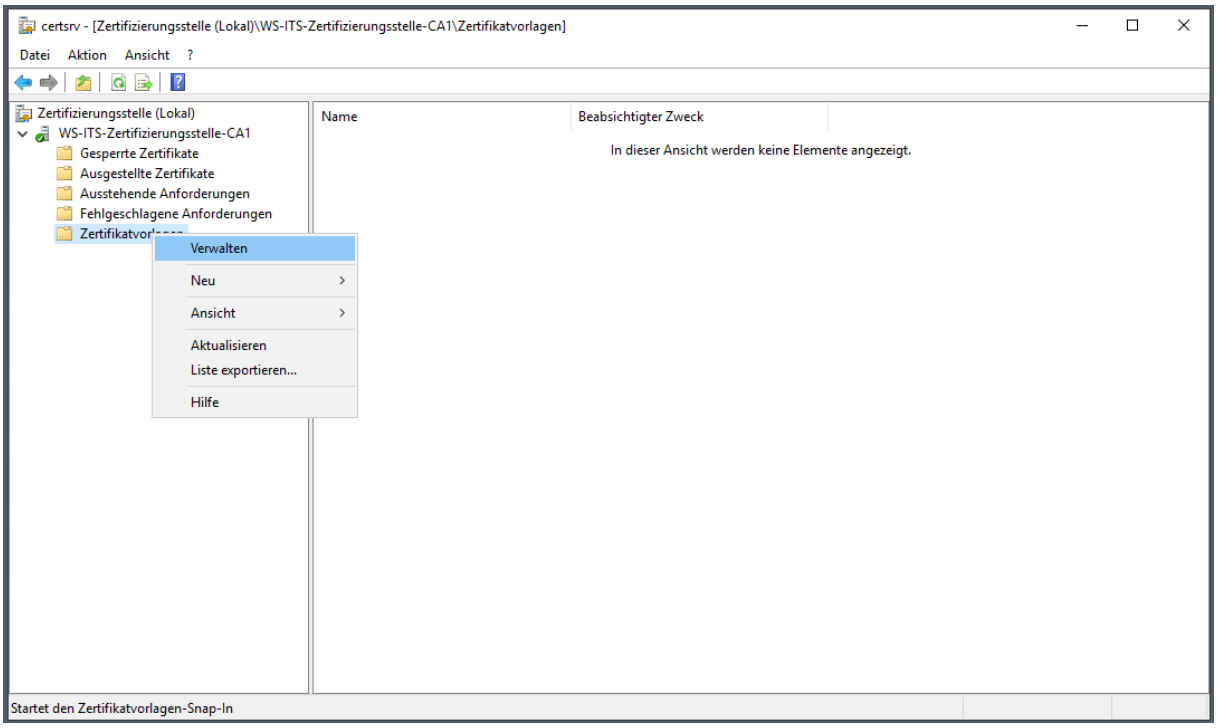


Damit ist die Grundkonfiguration abgeschlossen.

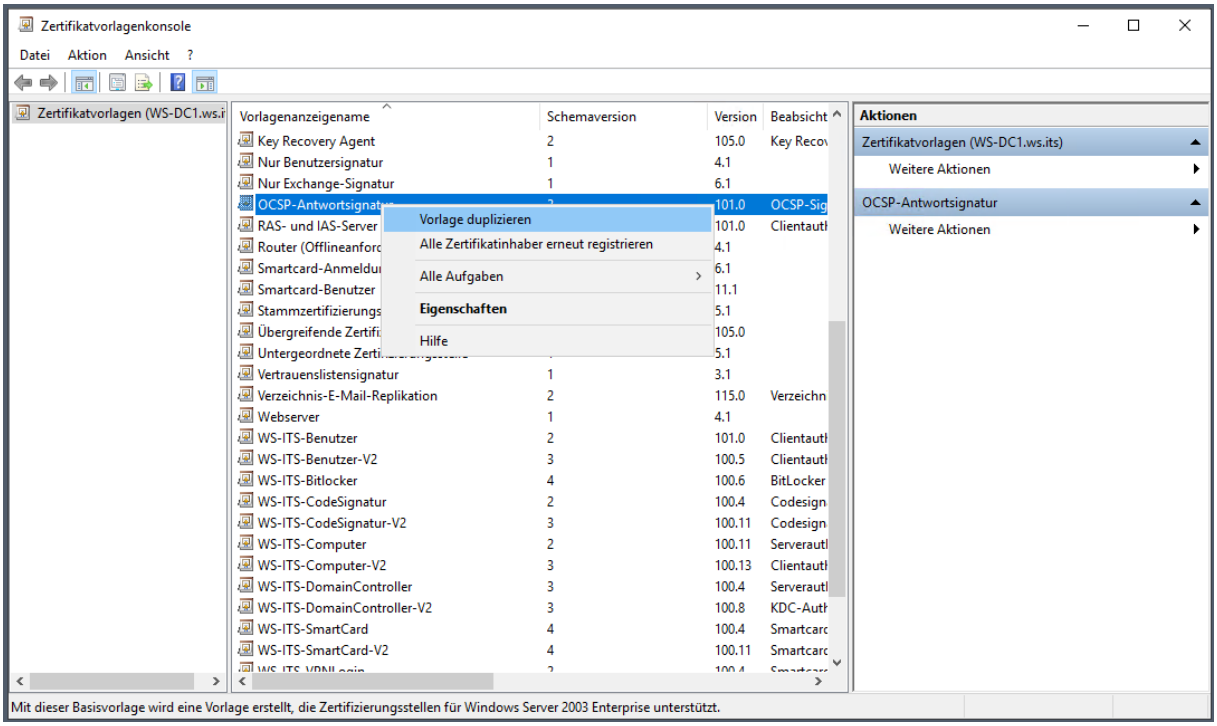
Bereitstellung des Online Responders

Weiter geht es mit den Zusatzdiensten. Der Online Responder ist eine Web-Anwendung, mit der ein Client ein einzelnes Zertifikat durch seine Seriennummer auf eine Sperrung überprüfen kann, ohne dabei selbst die komplette Sperrliste herunterladen und durchsuchen zu müssen. Ohne den Download der Sperrliste umgehe ich das Abwarten der Cache-Dauer der Sperrliste im Falle einer Zertifikatsperrung. Oder kurz gesagt: Jeder Client kann Zertifikate live und in „Echtzeit“ auf Sperrung validieren lassen.

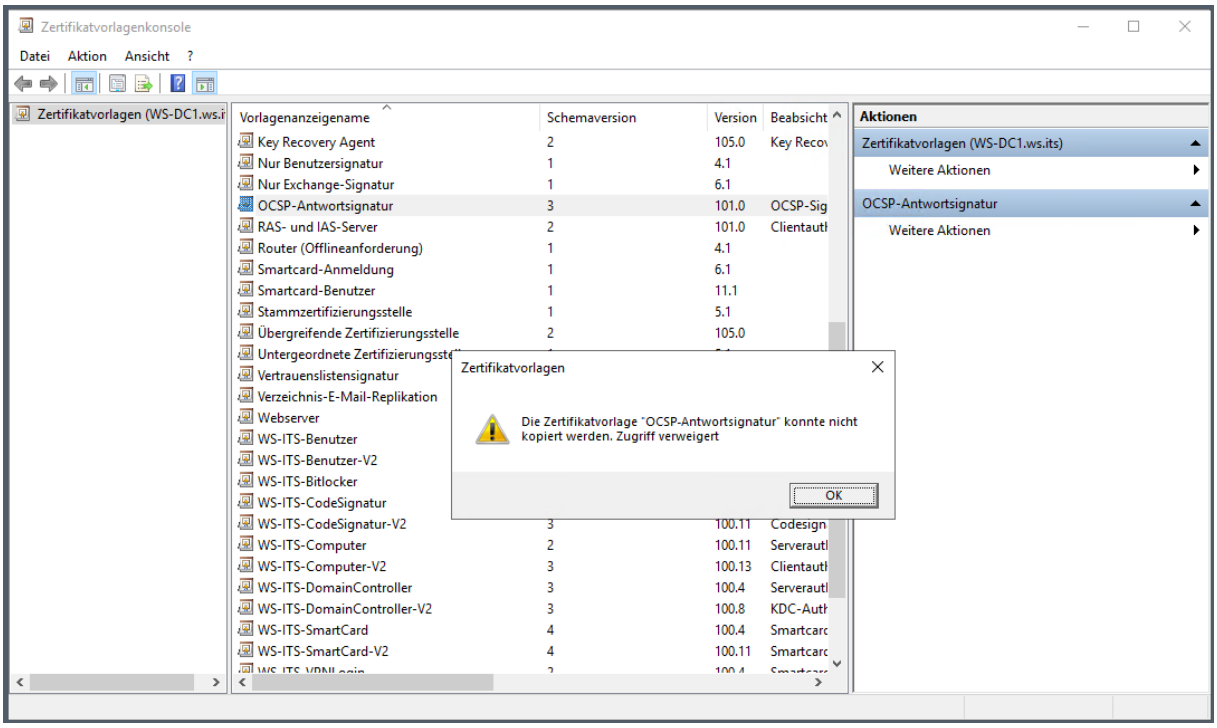
Der Online Responder muss seine Antworten digital signieren, damit der Client ihm vertraut. Dafür benötige ich eine spezielle Zertifikatvorlage. Diese erstelle ich in meiner Zertifizierungsstelle:



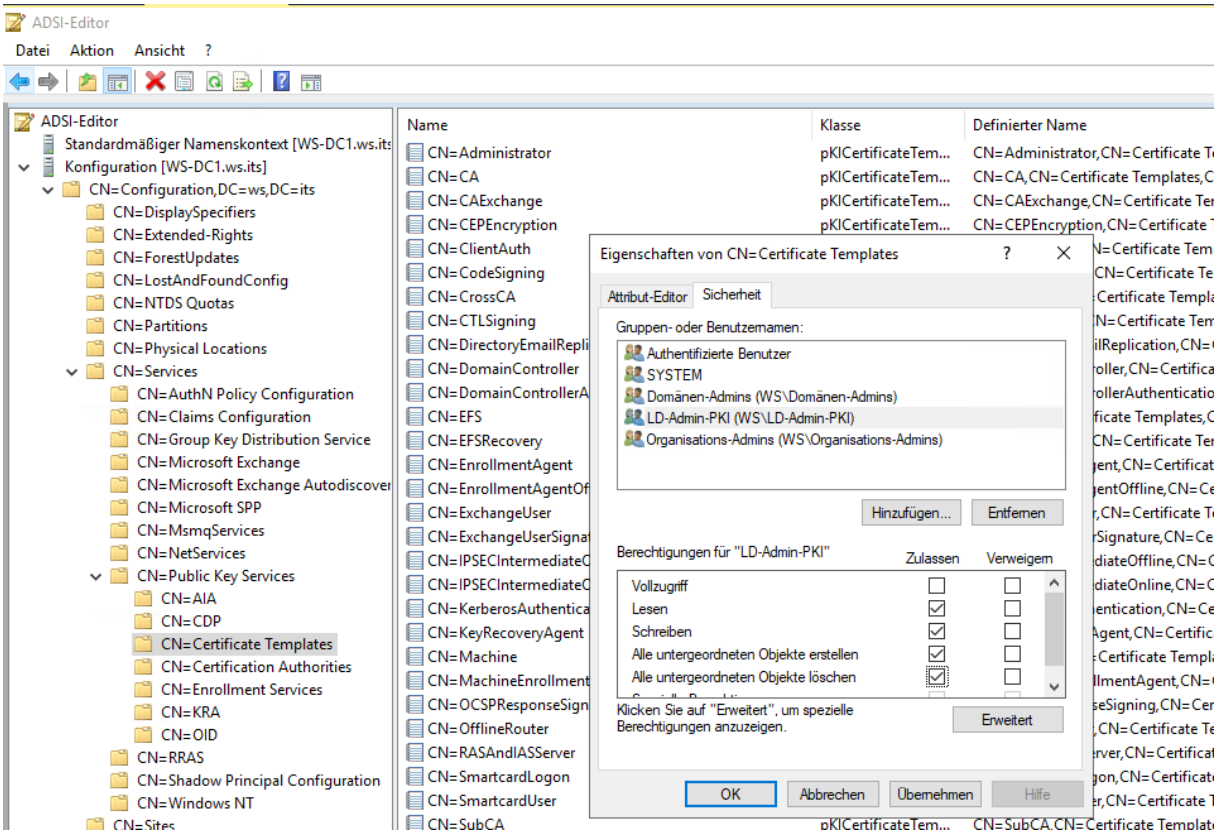
Ich kopiere das Original:



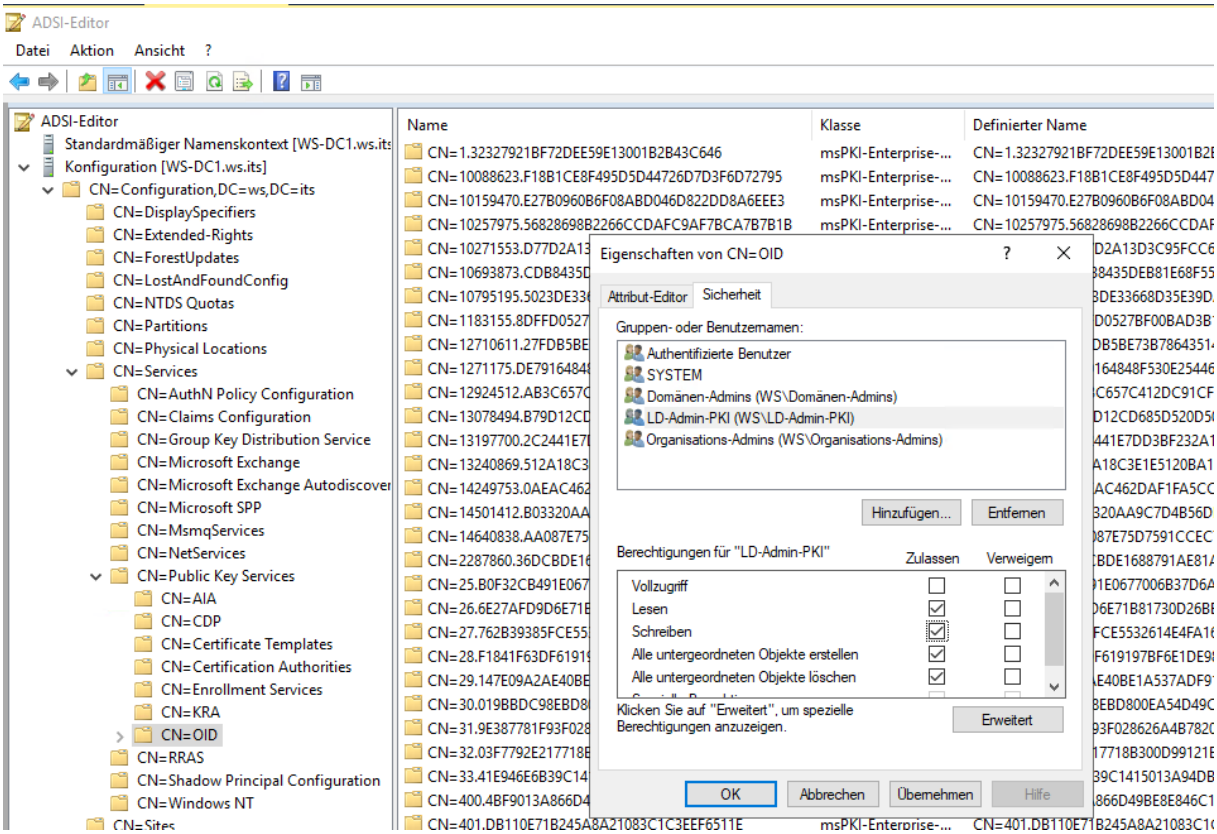
Oder auch nicht. Denn auch hierfür sind Enterprise-Adminrechte erforderlich…



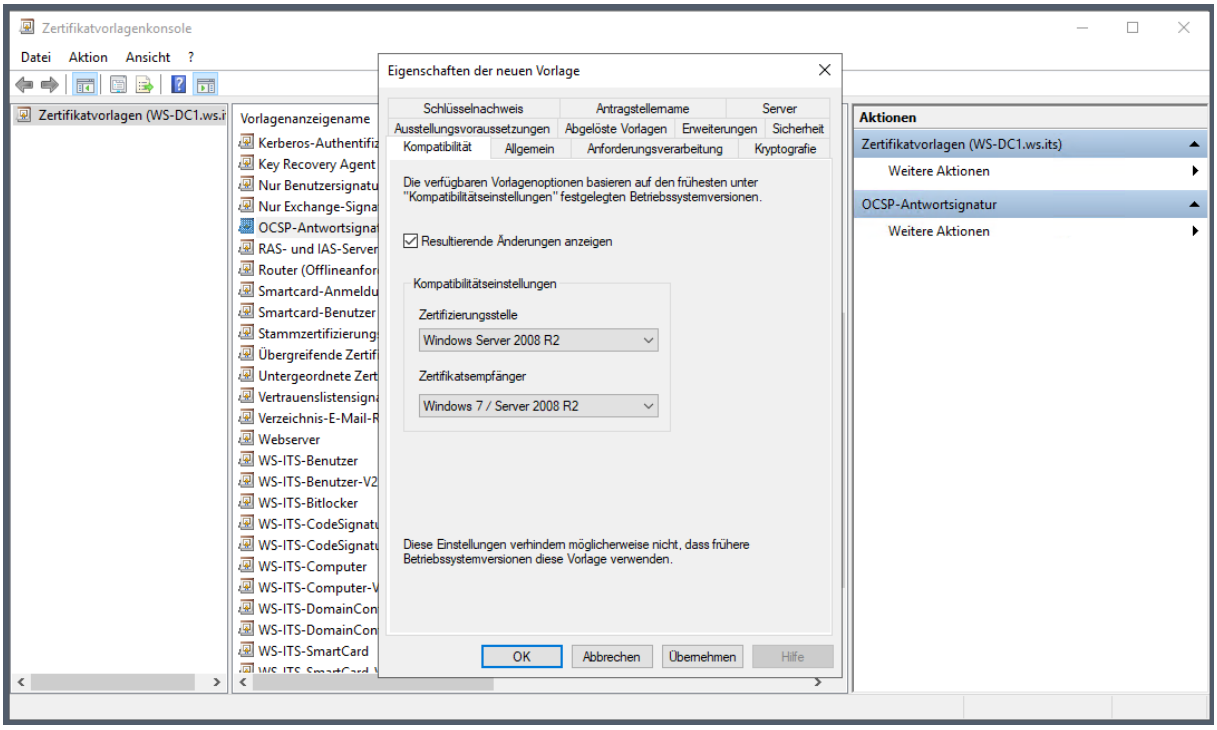
Das hatte ich für meine PKI nie angepasst. Aber heute ist dafür der richtige Zeitpunkt. Ich wechsle auf meinen Domain Controller mit meiner T0-Kennung. Diese nehme ich zuvor in die Gruppe Enterprise-Admins auf. Im ADSIEdit navigiere ich zum Container, in dem die Templates liegen. Hier editiere ich die ACL und nehme meine eigene Gruppe LD-Admin-PKI mit Schreibrechten auf:



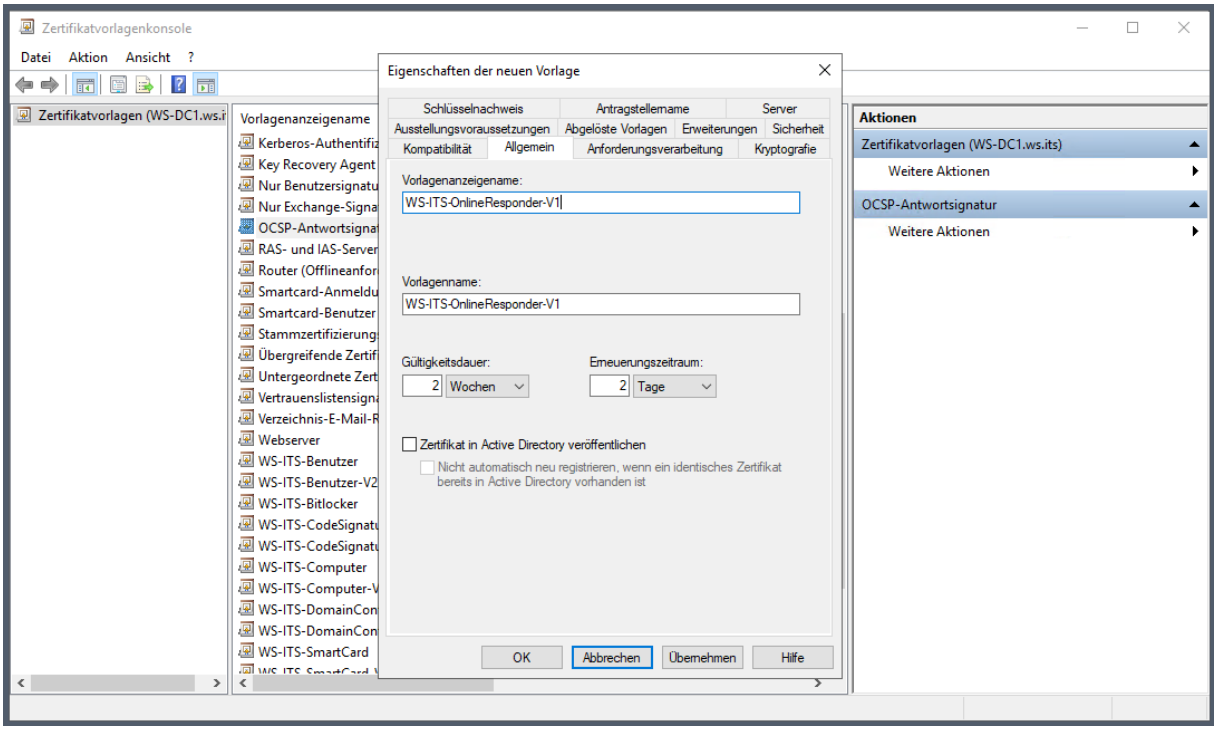
Die Aktion wiederhole ich auf dem dazugehörigen Container OID:



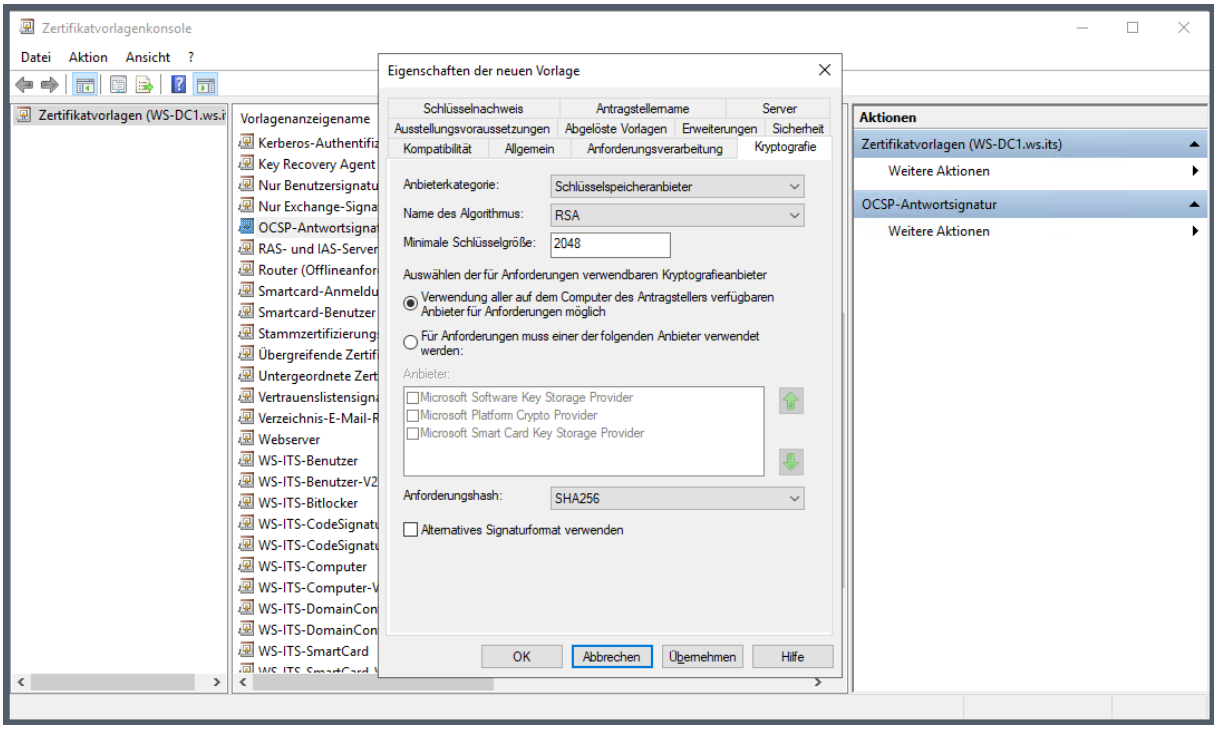
Anschließend dupliziere ich erfolgreich die Vorlage in meiner Zertifizierungsstelle. Nun kann ich noch einige Parameter anpassen. Dazu gehört die Kompatibilitätsebene:



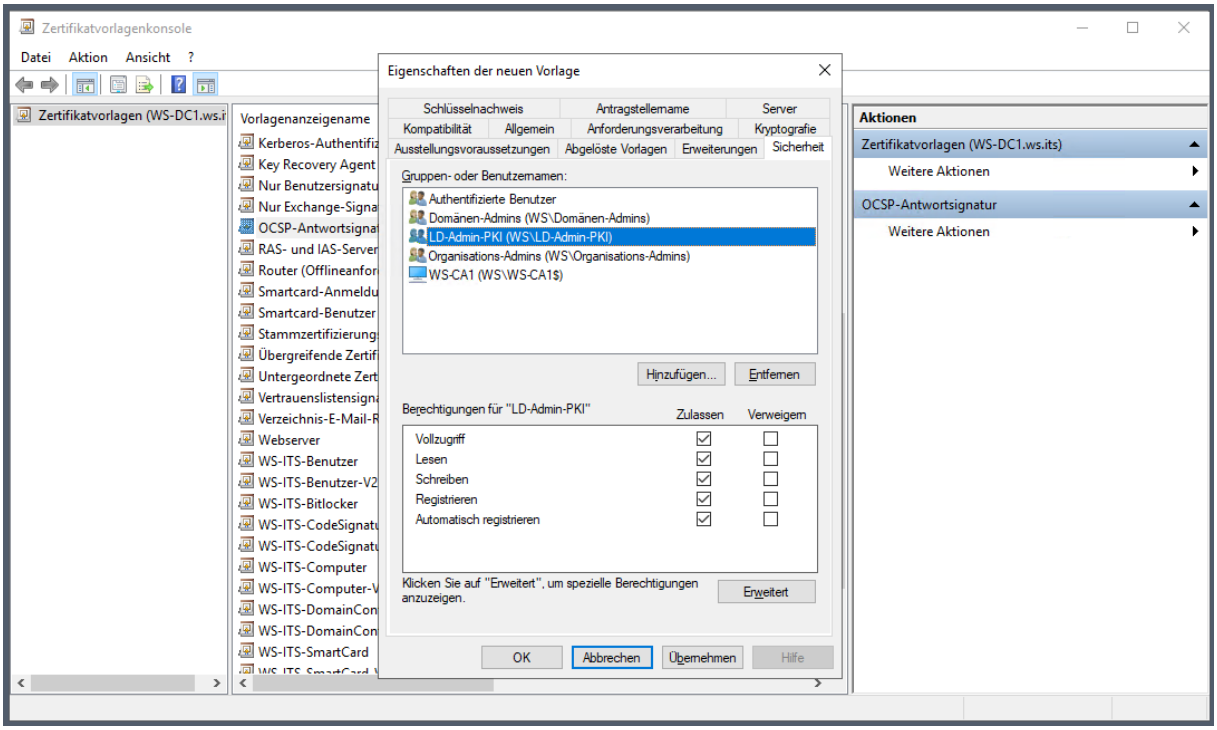
Auch der Name darf sich in mein Namensschema einfügen:



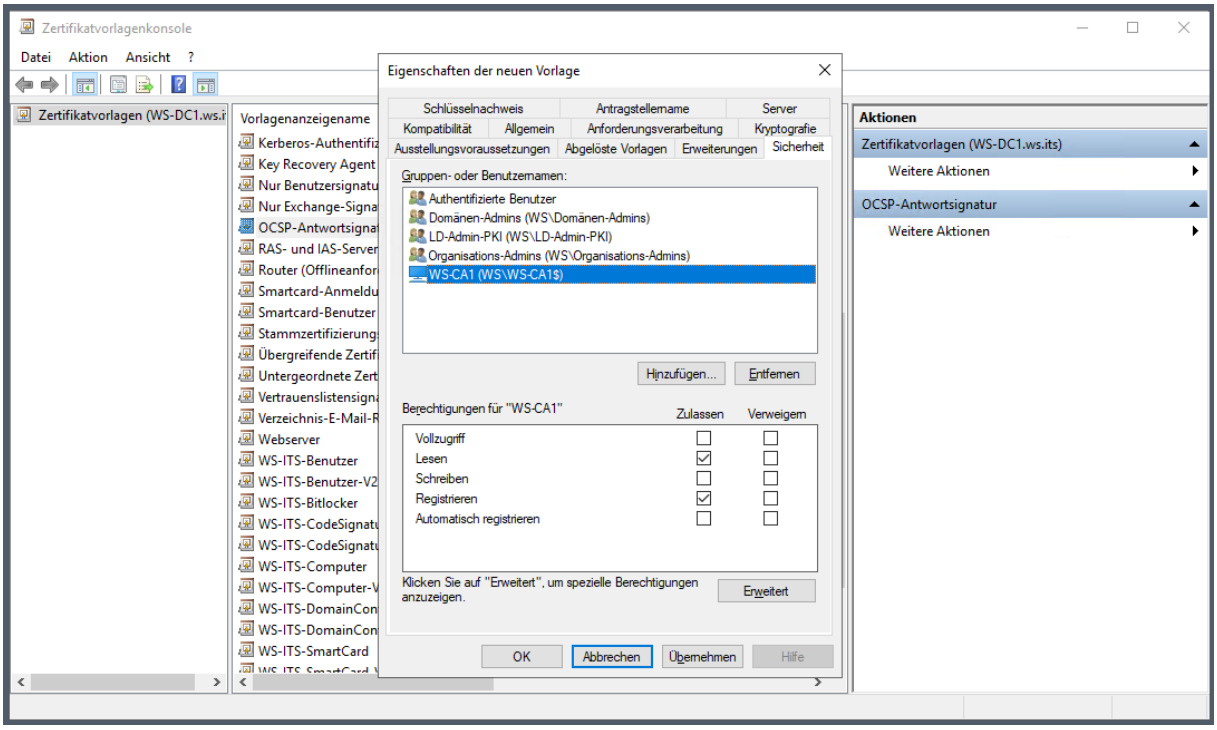
Die Kryptografie passt für den Anwendungsfall:



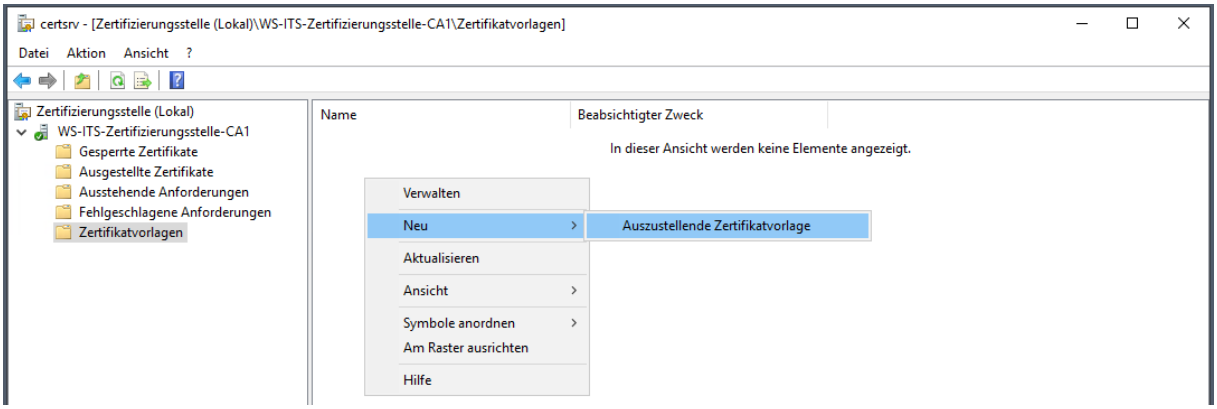
Das Recht zur Editierung passe ich in der Sicherheit an. Hier kommt meine Admin-Gruppe wieder dazu:



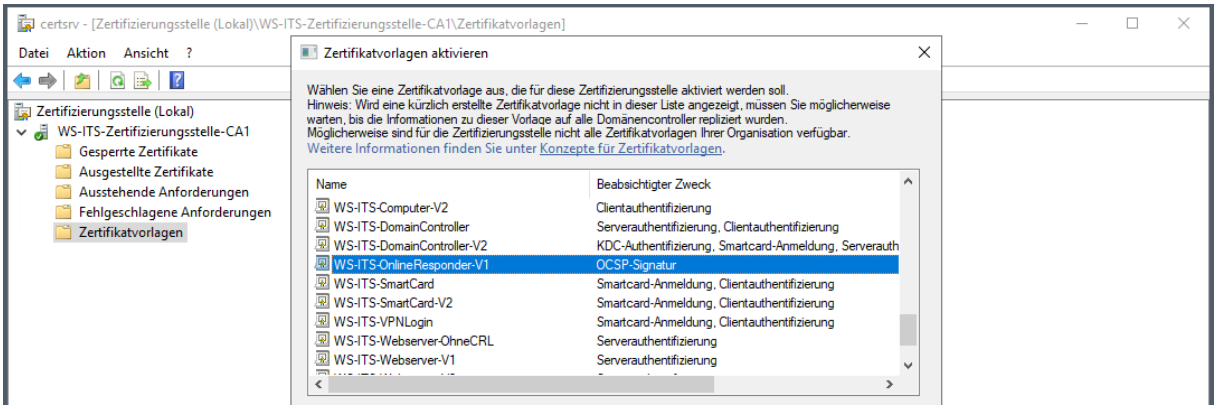
Der Online Responder wird in einem Sicherheitskontext ausgeführt. Die dazugehörige Identität muss in der Lage sein, Zertifikate auf Basis der neuen Vorlage zu ziehen. Mein Online Responder wird im System-Kontext der Windows CA laufen. Also berechtige ich den Computer-Account für ein Enrollment:

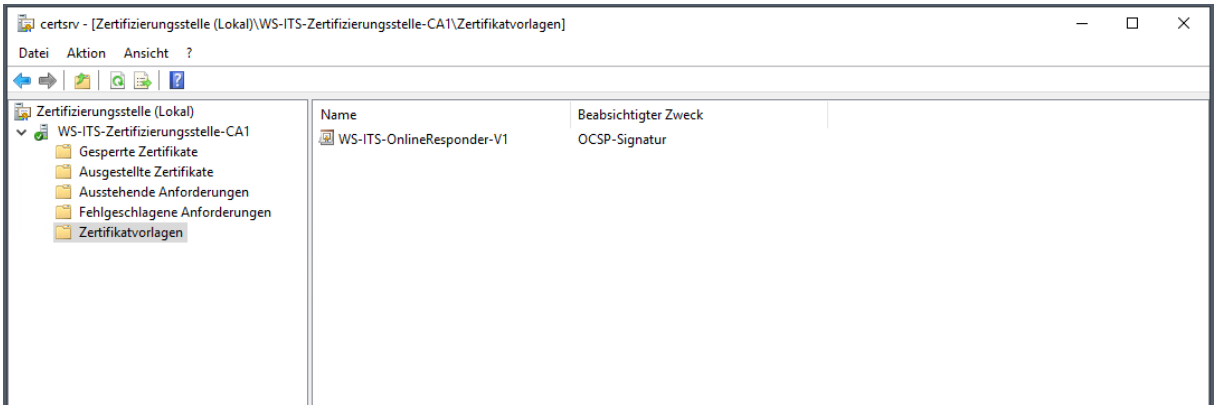


Damit ist die Vorlage fertig. Nun nehme ich die Vorlage in die auszustellenden Vorlagen auf. Erst danach können passende Zertifikate angefragt werden:

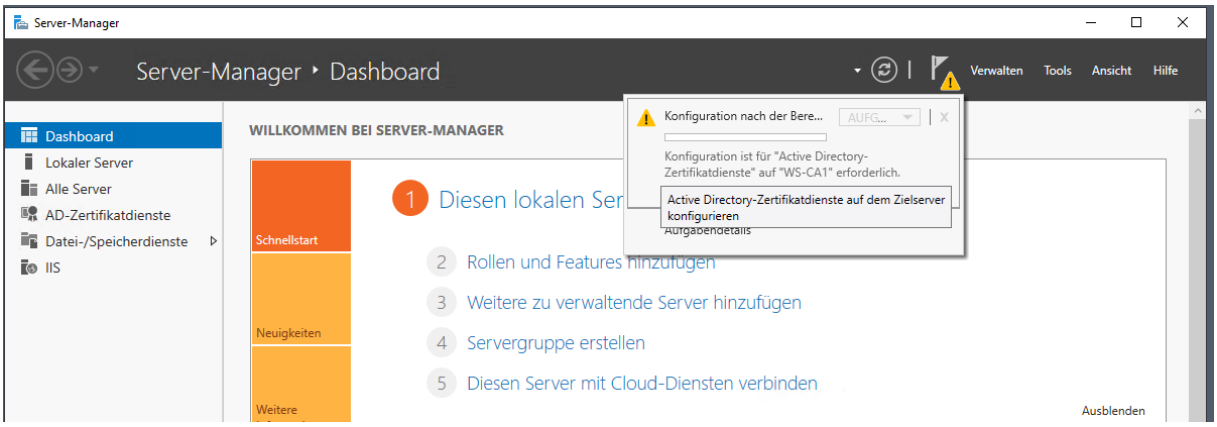


Eventuell wird die Vorlage nicht sofort angezeigt. Dann muss noch eine Domain Controller Replikation abgewartet werden. Bei mir hat der zeitliche Versatz ausgereicht:

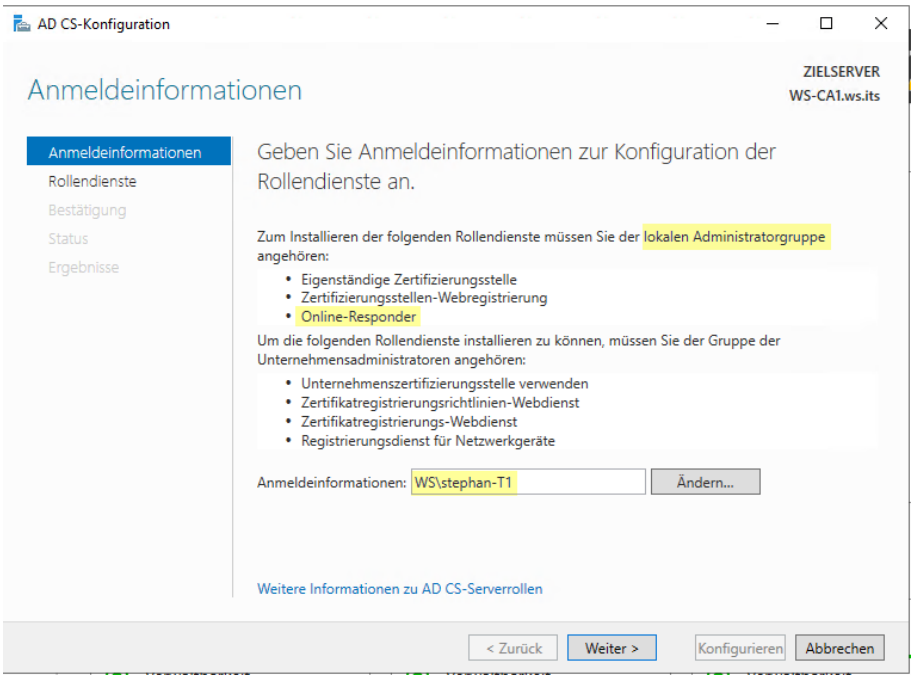




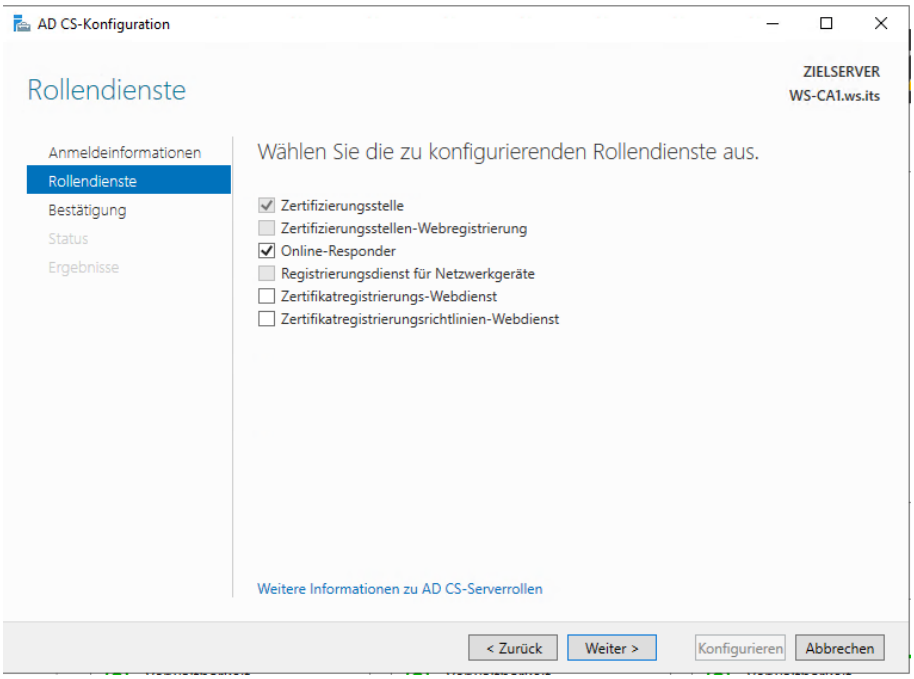
Weiter geht es mit dem Einrichten der Rolle. Die Rolleninstallation ist bereits abgeschlossen. Es fehlt noch das Post-Deployment. Dieses kann im Server Manager gestartet werden:



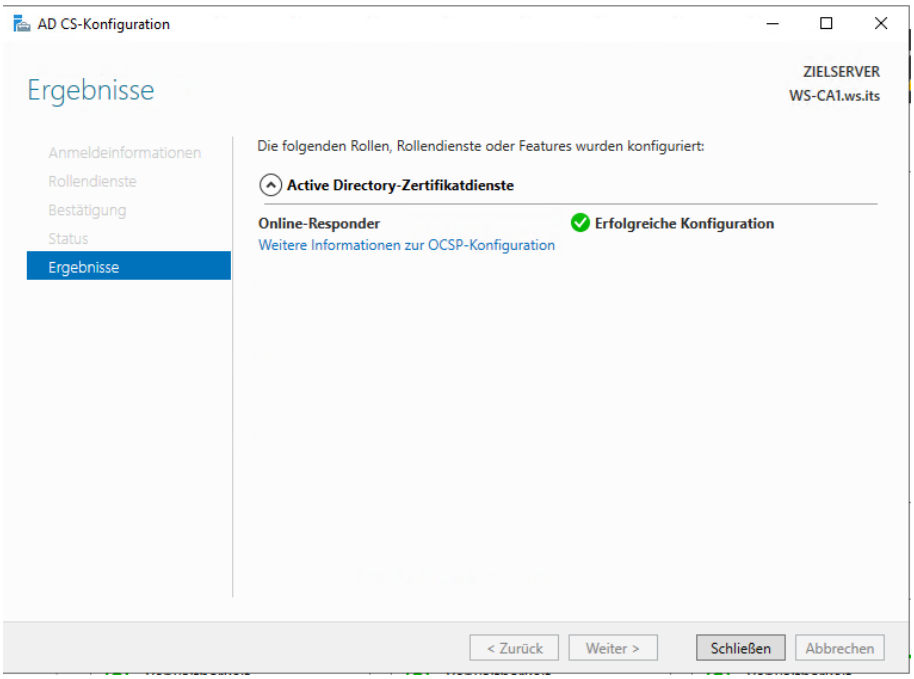
Der Online Responder benötigt nur lokal administrative Rechte. Meine T1-Kennung ist Mitglied in den lokalen Admins:



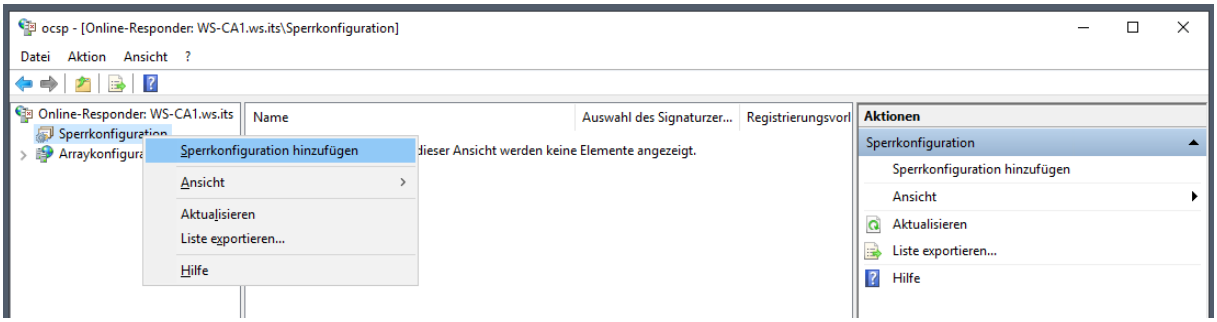
Ich aktiviere nur den Online Responder. Die anderen Rollen kommen später dran:



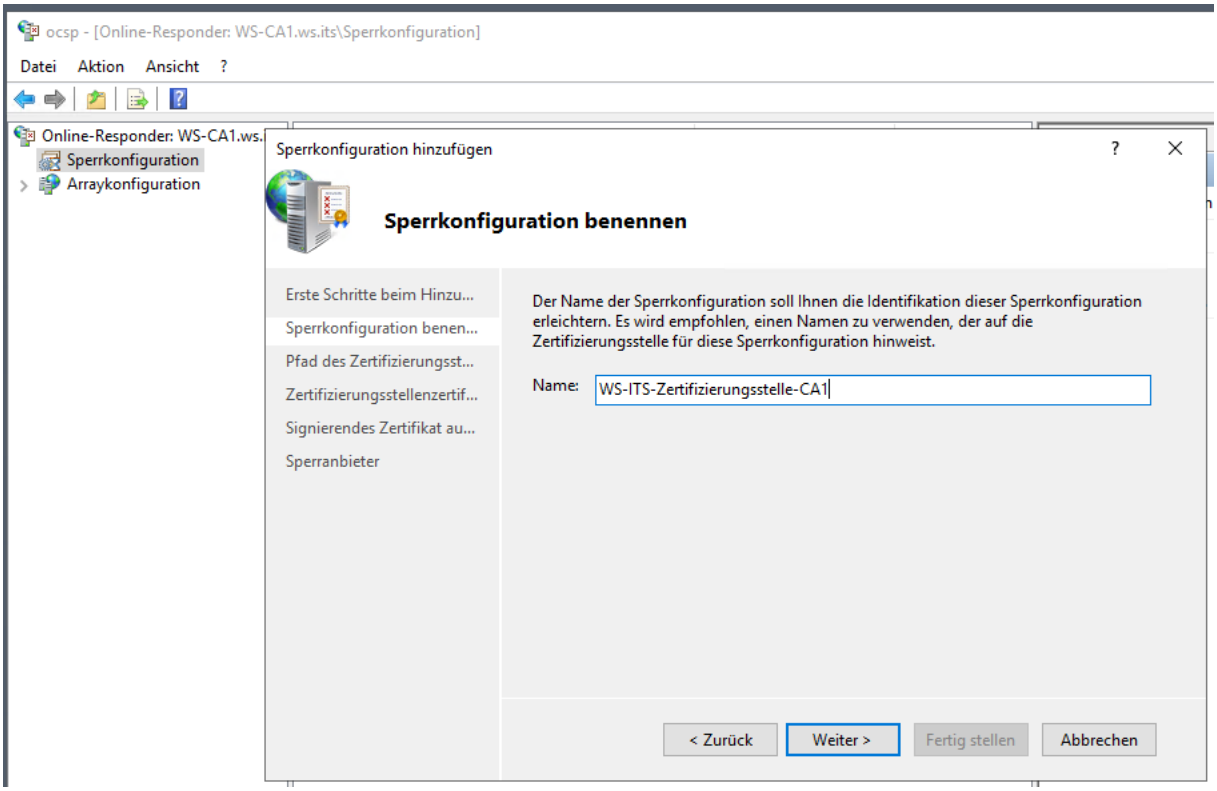
Mehr ist hier nicht erforderlich:



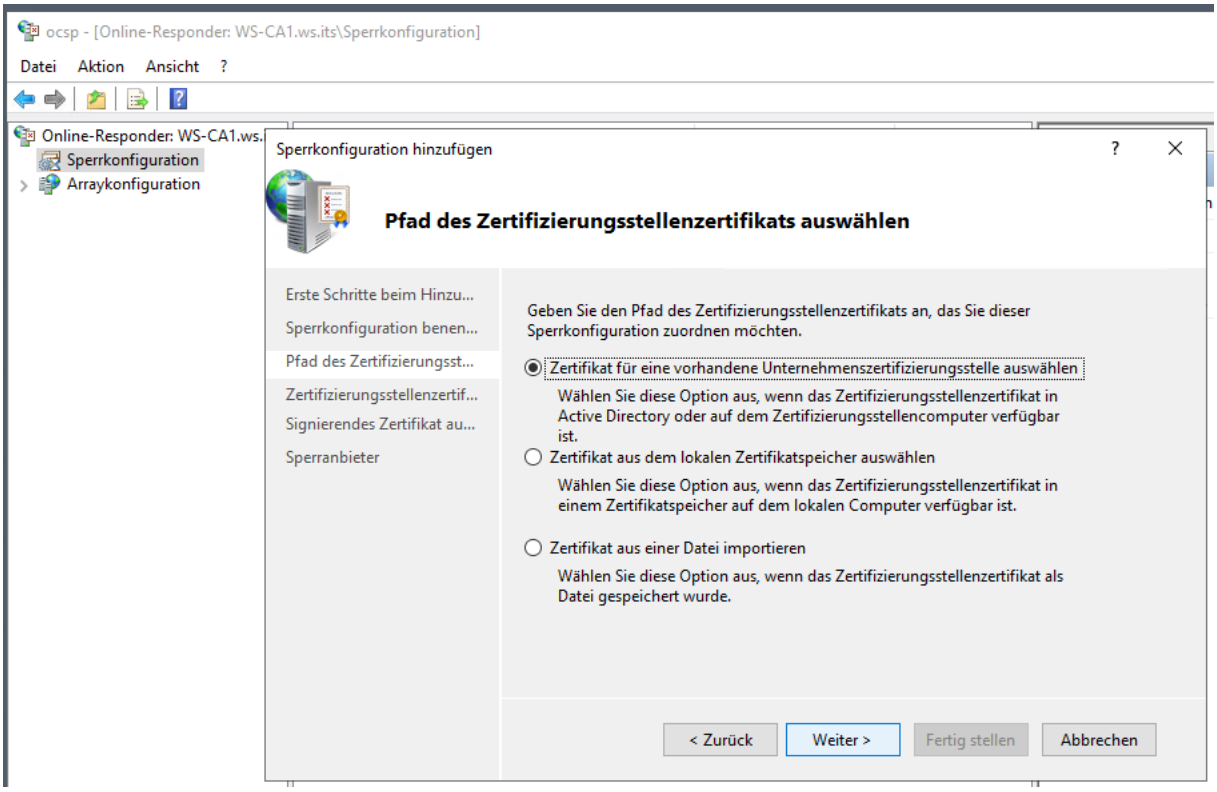
Der Online Responder hat eine eigene Management-Konsole. Hier muss ich nun noch die Sperrlisten zuweisen, die er bedienen soll:



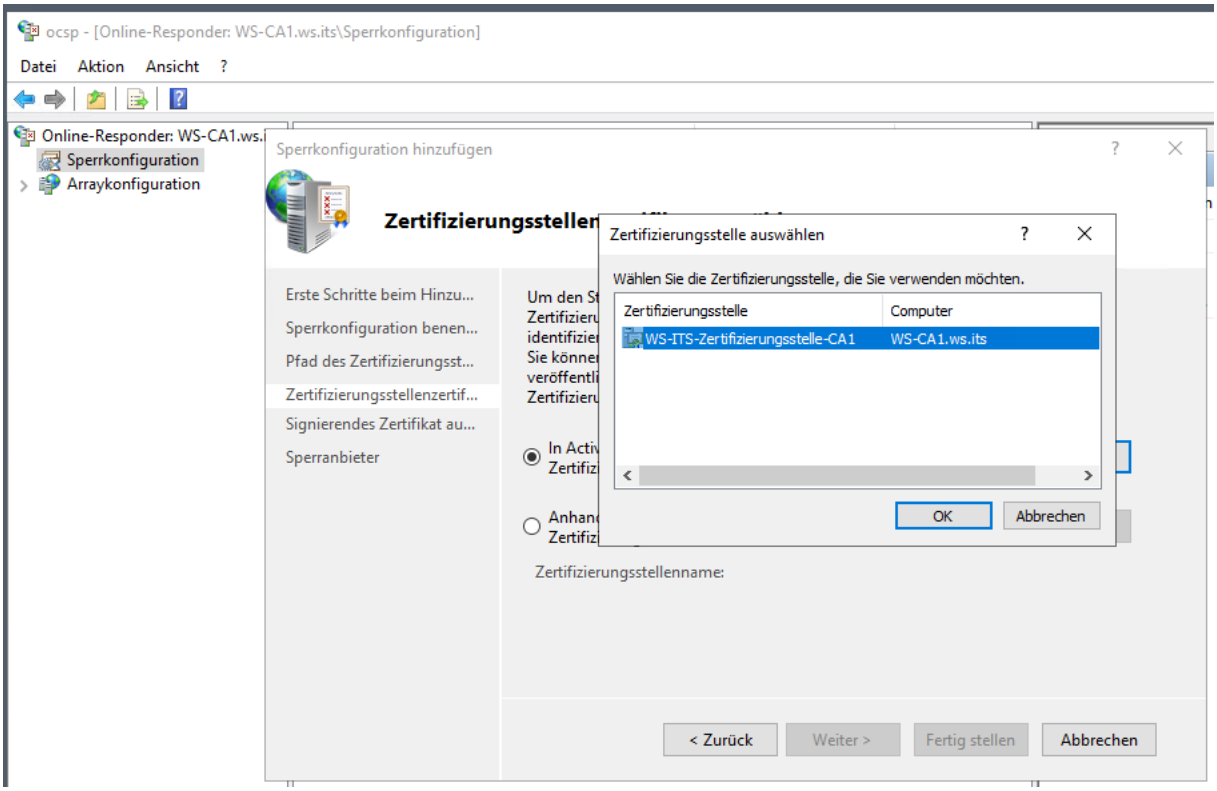
Jede Sperrliste wird durch einen Namen gekennzeichnet. Ich nehme hier den Namen meiner Zertifizierungsstelle:



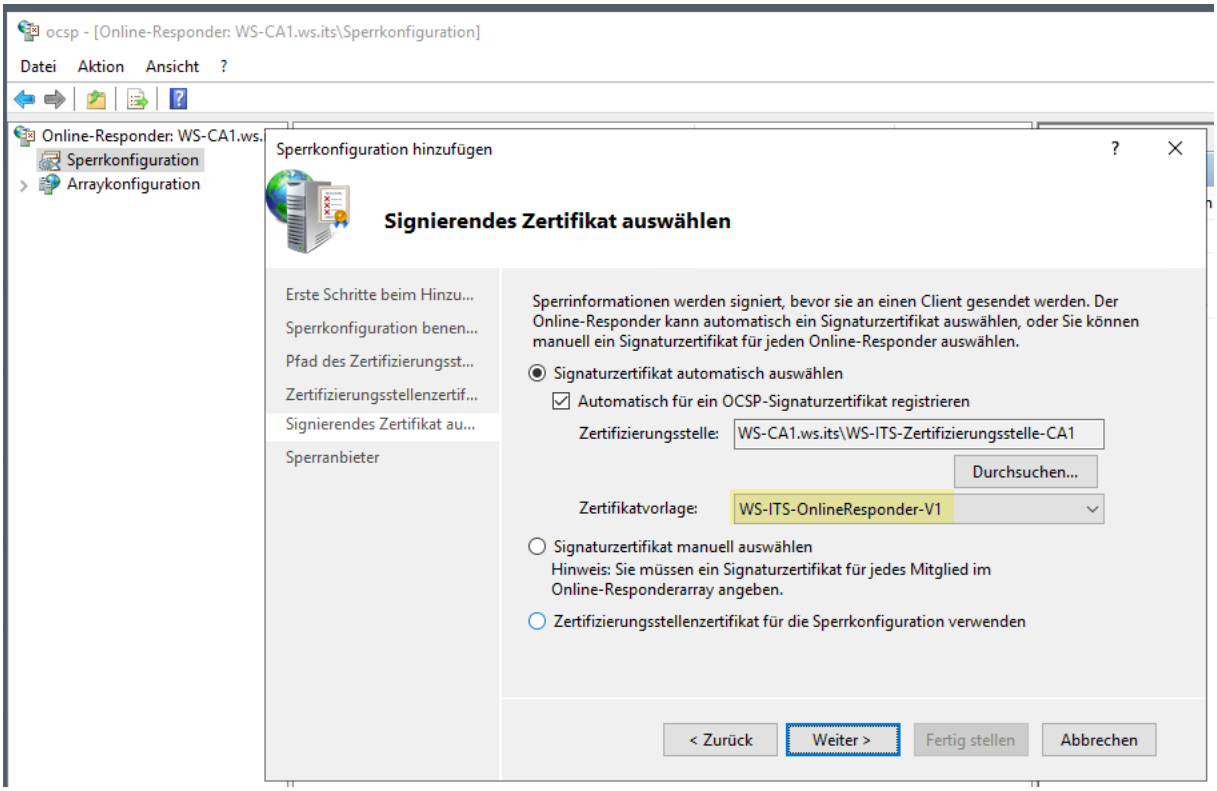
Im nächsten Schritt wird die dazugehörige Zertifizierungsstelle gesucht. Ich kann das durch meine AD-Integration einfach halten:



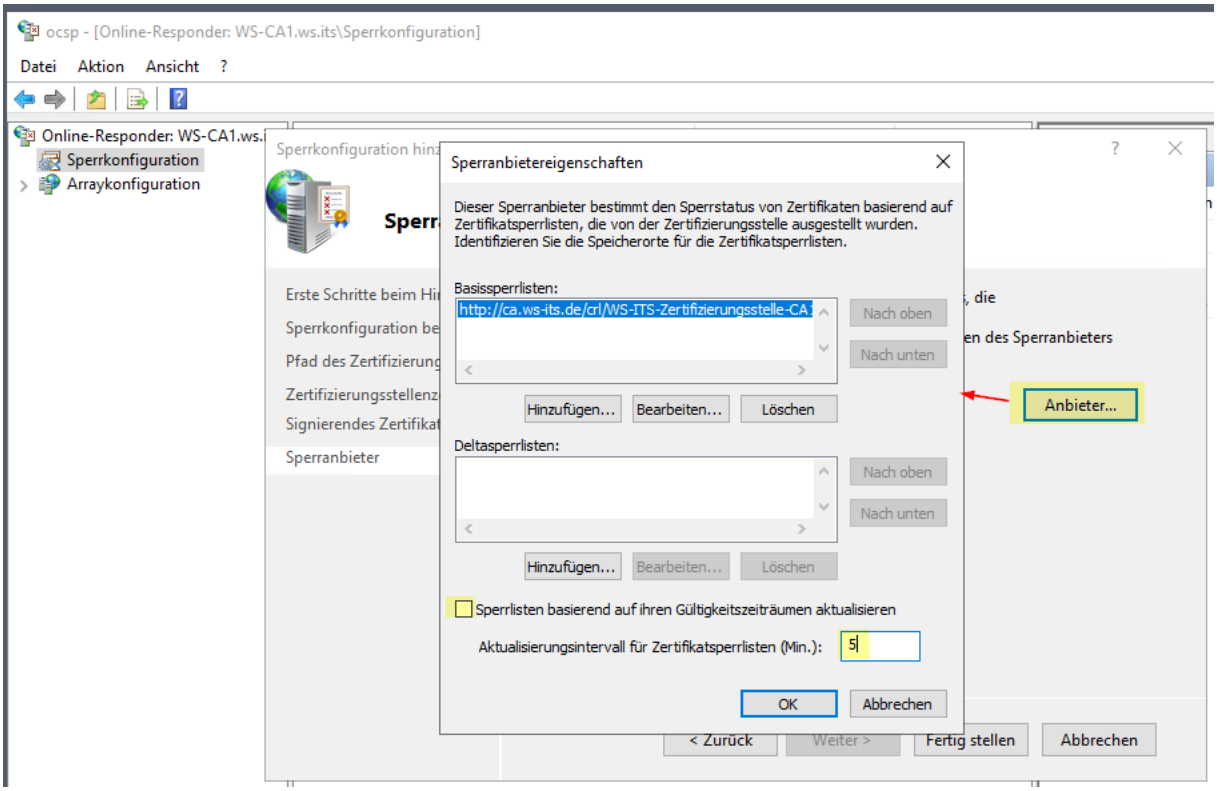
Der Auswahldialog zeigt meine interne Windows CA:



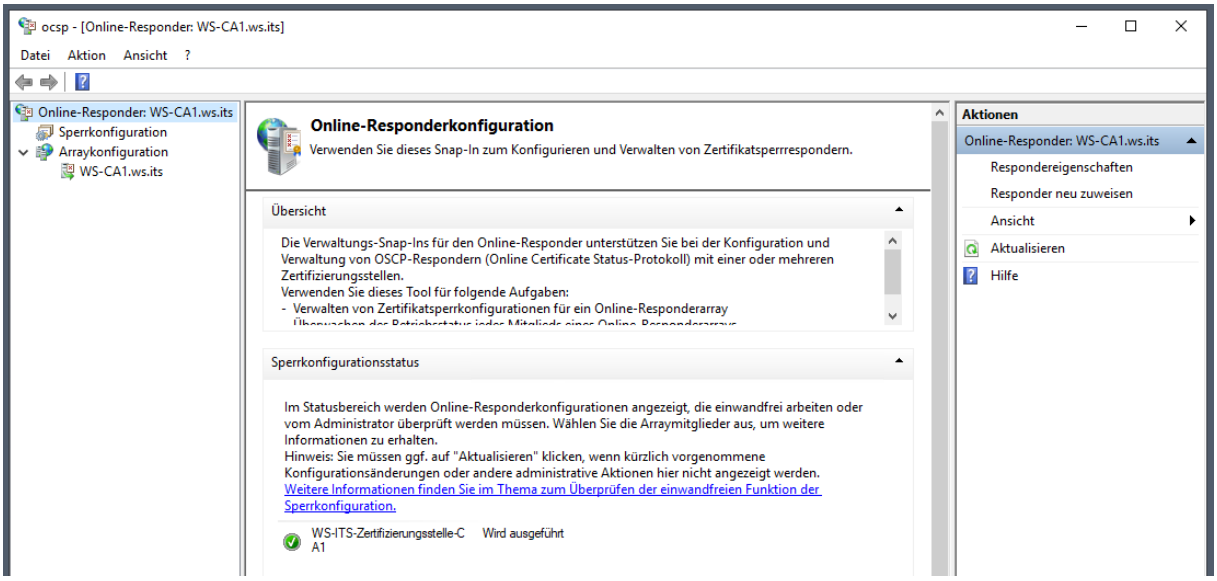
Für diese CA und ihre Sperrliste muss ein Signaturzertifikat gewählt werden. Dieses wird sich mein Online Responder auf der neuen Zertifikatvorlage basierend automatisch ausstellen. Die Vorlage wurde bereits automatisch gefunden und zugewiesen:



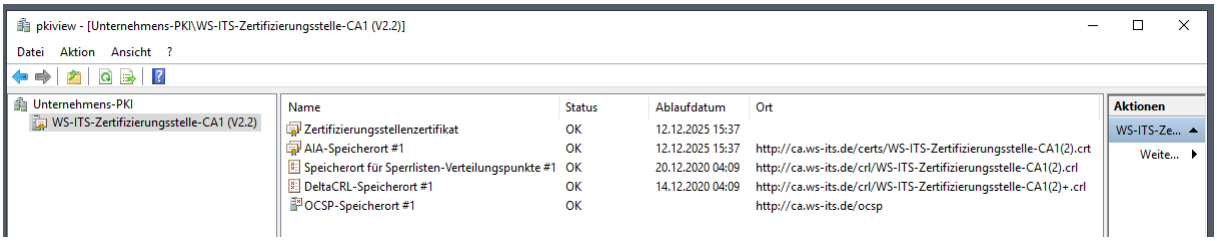
Nach einem Kontakt zur Zertifizierungsstelle werden die möglichen Sperrlistenverteilungspunkte gelistet und zur Auswahl angeboten. Der Online Responder lädt sich die Listen selber herunter und prüft damit die Anfragen der Clients. Bei mir ist nur noch der neue Verteilungspunkt über http sichtbar. Das soll genügen. Aber die Aktualisisierungsfrequenz darf gerne wesentlich kleiner sein als als der Wert, der auf den Sperrlisten steht. So erhalte ich ein „Echtzeit-Sperrverhalten“:



Nach der Bestätigung organisiert sich der Online Responder ein OSCP-Signaturzertifikat und meldet seine Einsatzbereitschaft:



Ein abschließender Blick im PKIVIEW zeigt nun alle Services fehlerfrei an:

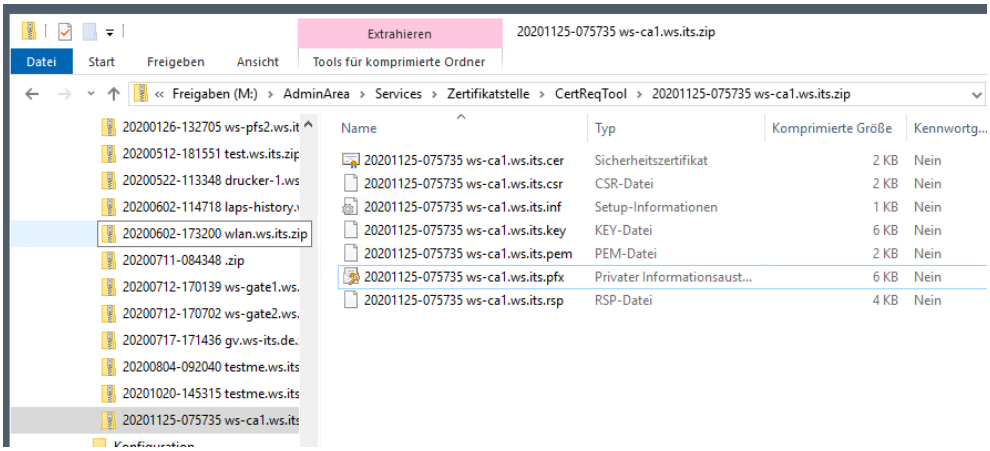


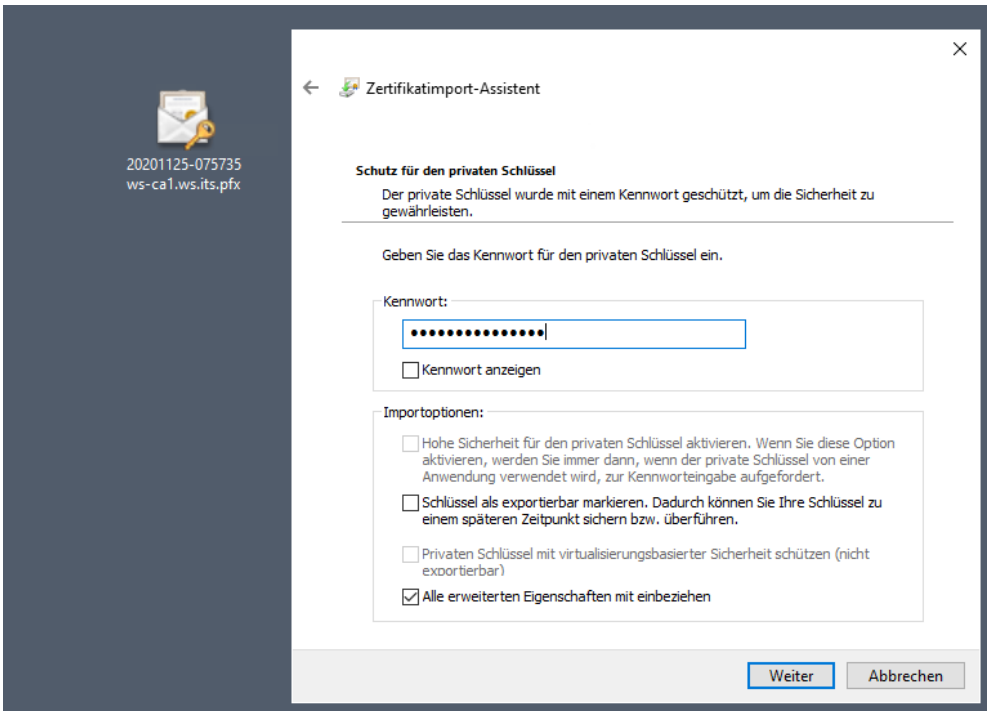
Einen realen Testlauf werde ich später ausführen.

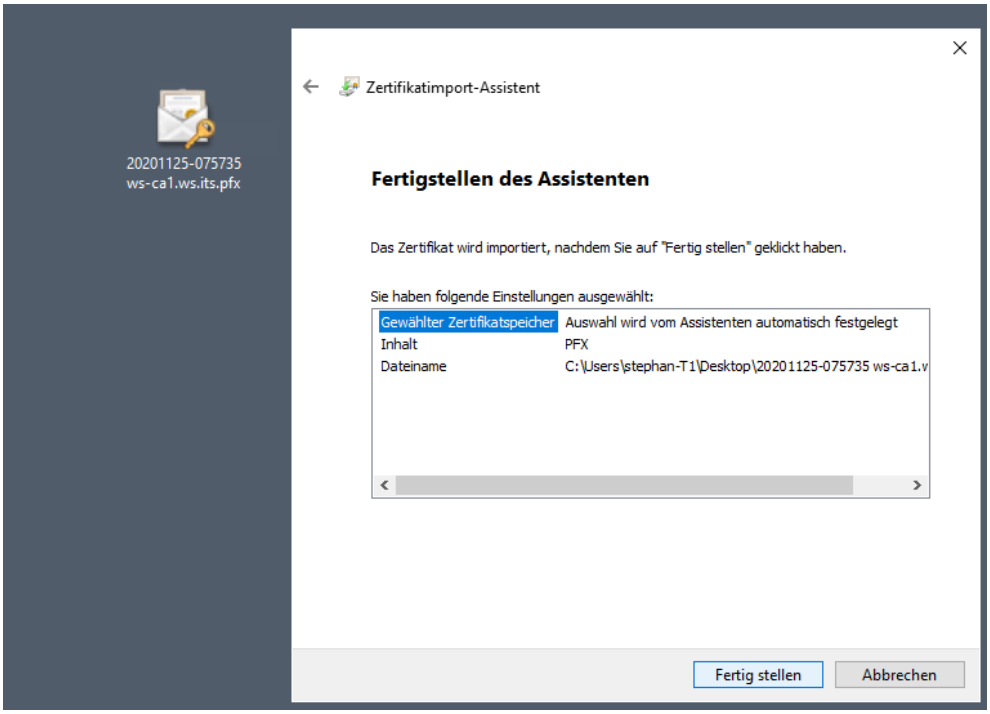
Bereitstellung von CEPCES

Vorher möchte ich einen weiteren Service integrieren: CEPCES. Die Idee dahinter habe ich bereits in der Zielsetzung erklärt.

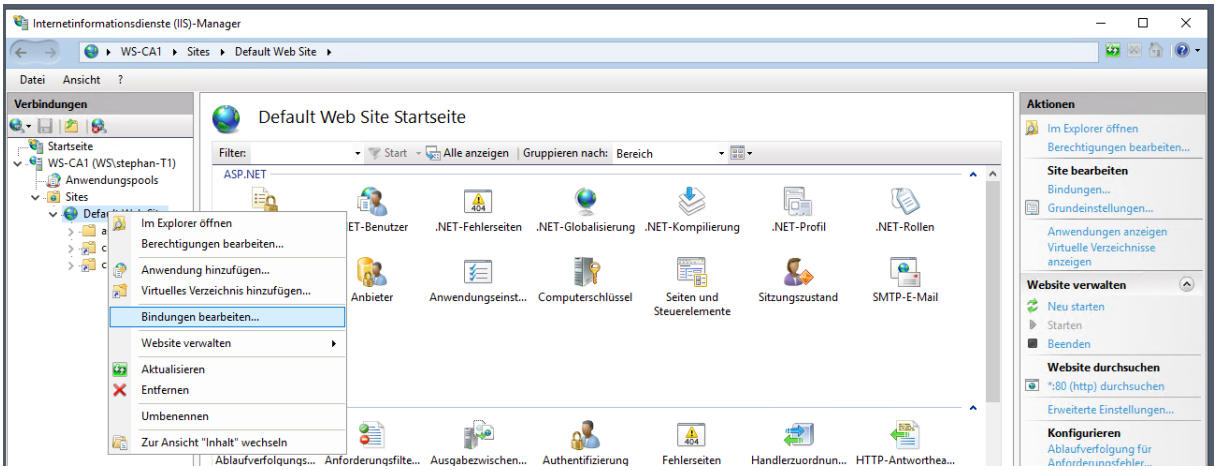
Für die Anfrage und die Ausstellung von Zertifikaten über https benötigt mein CEPCES ein Webserver-Zertifikat. Das alte Zertifikat habe ich im PKCS12-Format mit privatem Schlüssel auf meinem AdminShare liegen. Ich importiere das Zertifikat auf meinen neuen WS-CA1:

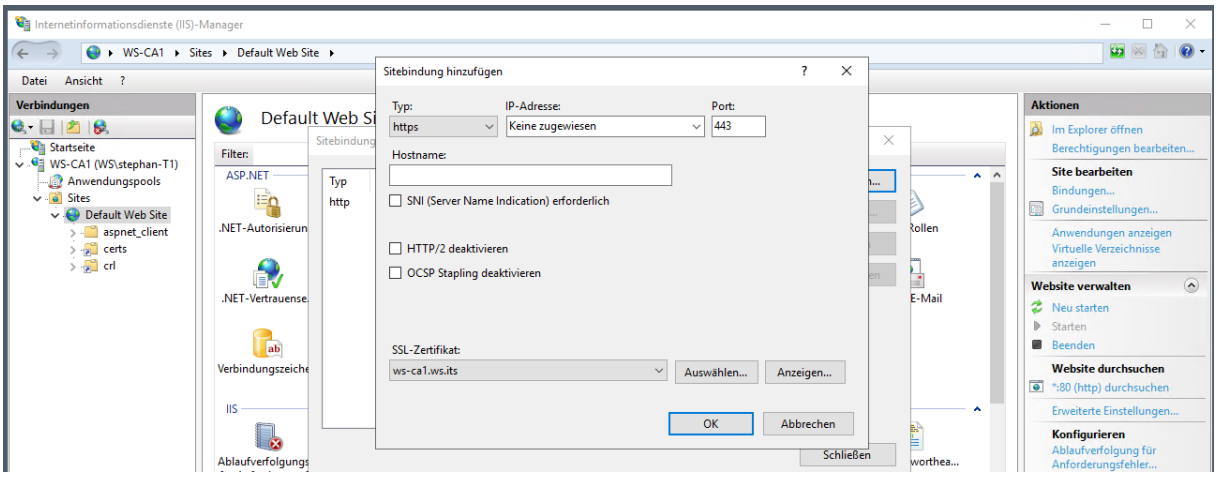




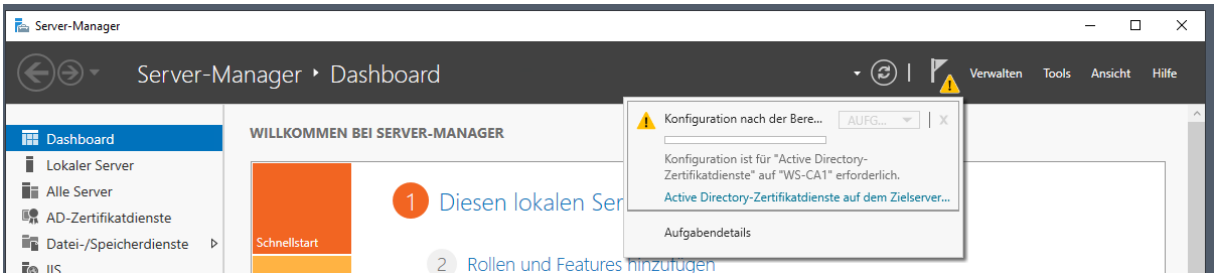


Das Zertifikat binde ich nun im IIS-Manager an den Port 443 meiner Default Website:

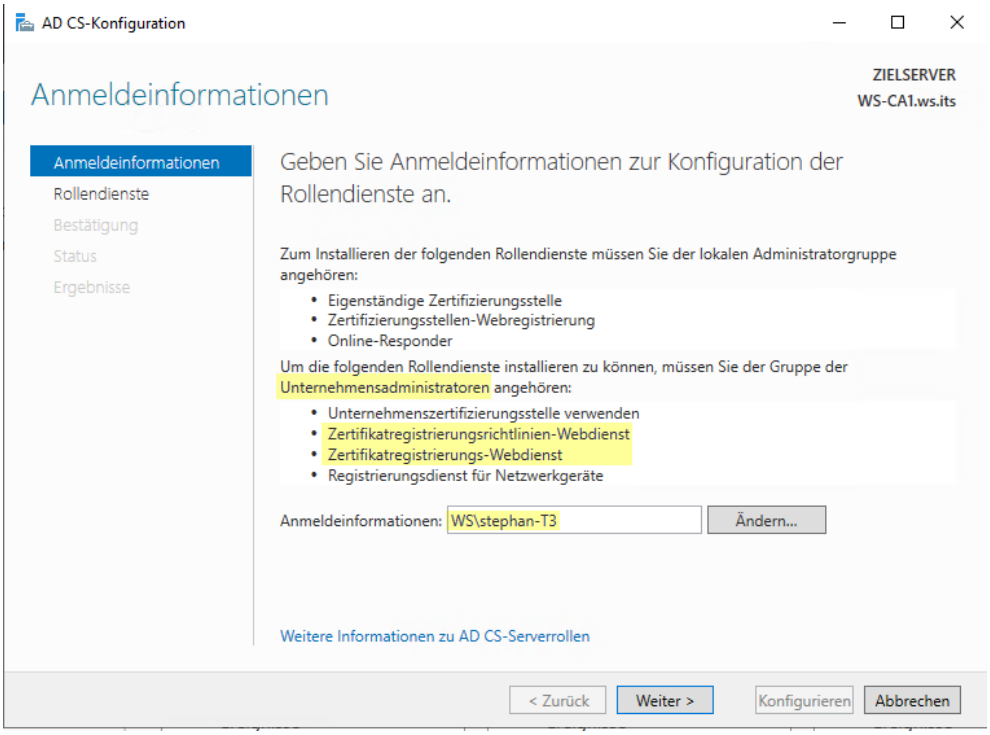




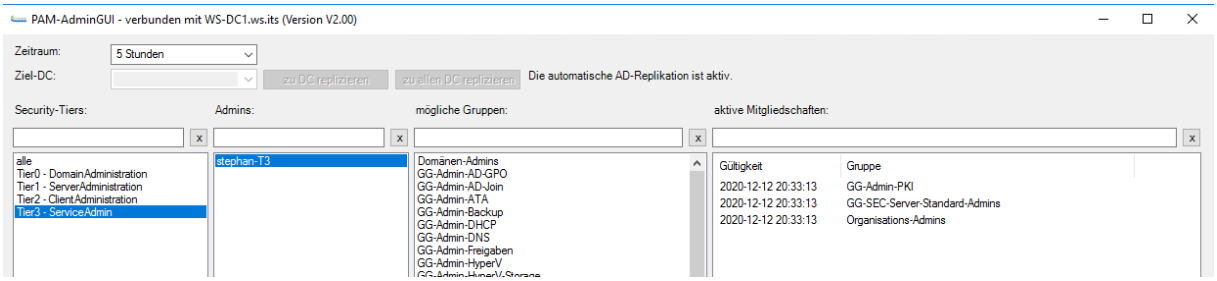
Danach starte ich das Post-Deployment der Rolle im Server Manager. Die Rolleninstallation hatte ich ja bereits ausgeführt:



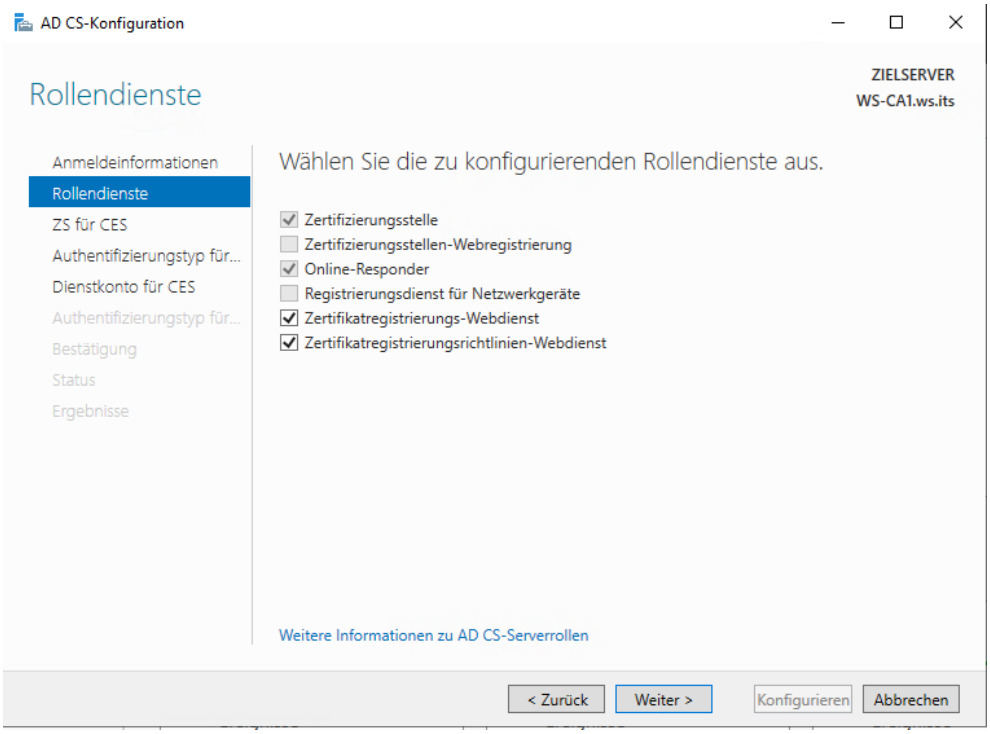
CEPCES wird im Active Directory in der Konfigurationspartition als Endpunkt registriert. Das kann nur ein Enterprise-Admin durchführen. Daher privilegiere ich meine T3-Kennung und wechsle den Sicherheitskontext im Assistenten:



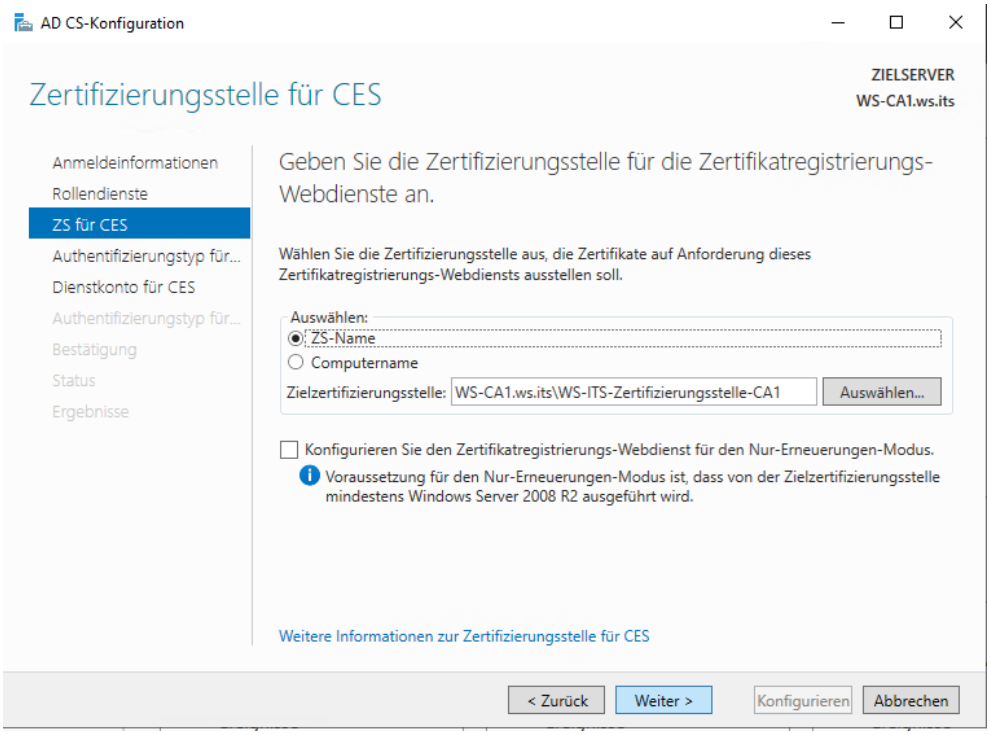
Die Privilegierung übernimmt mein PAM-Script:



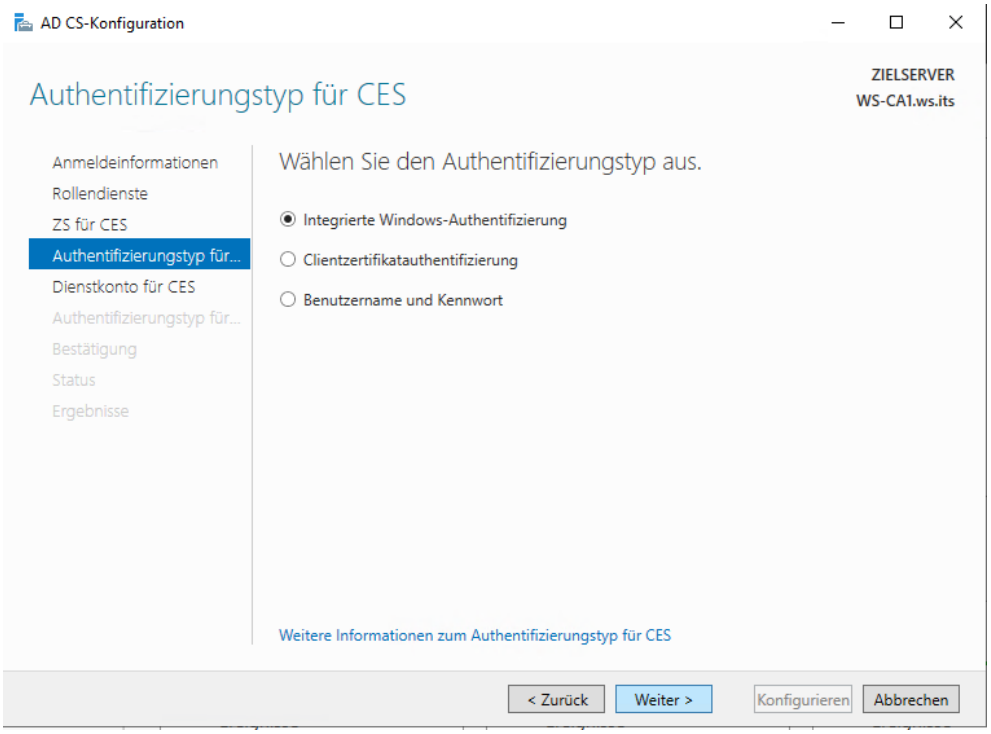
CEP und CES gehören hier zusammen. Daher starte ich das Post-Deployment gemeinsam:



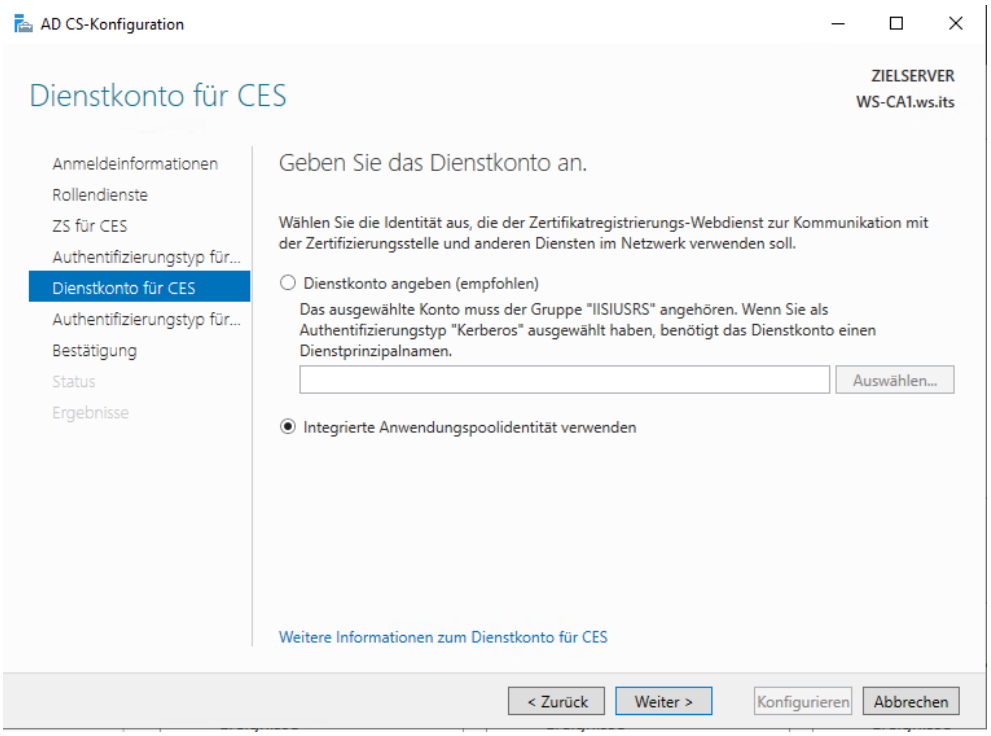
Zuerst ist CES an der Reihe. CES muss mit einer Windows Zertifizierungsstelle „verheiratet“ werden. Wird CES auf einem Zertifizierungsstellen-Server installiert, dann muss dieser gewählt werden. Ich hab es einfach, denn es gibt nur eine Windows CA:



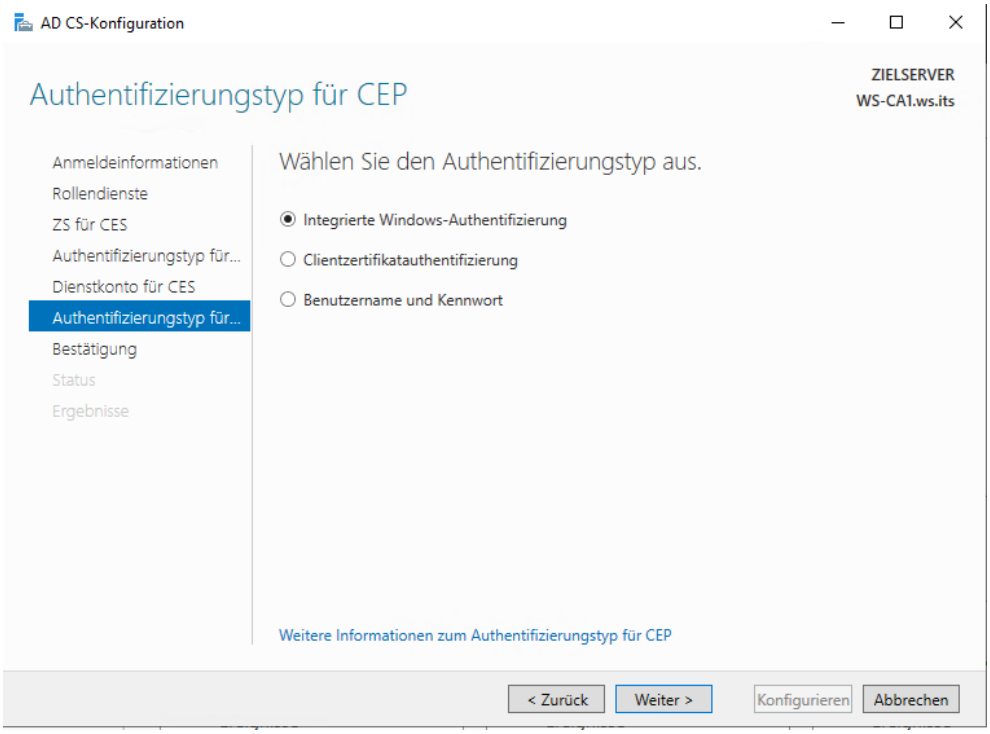
Clients und Benutzer sollen sich mit Kerberos am Webservice vom CES anmelden. So gibt’s Sicherheit und Single-Sign-On:



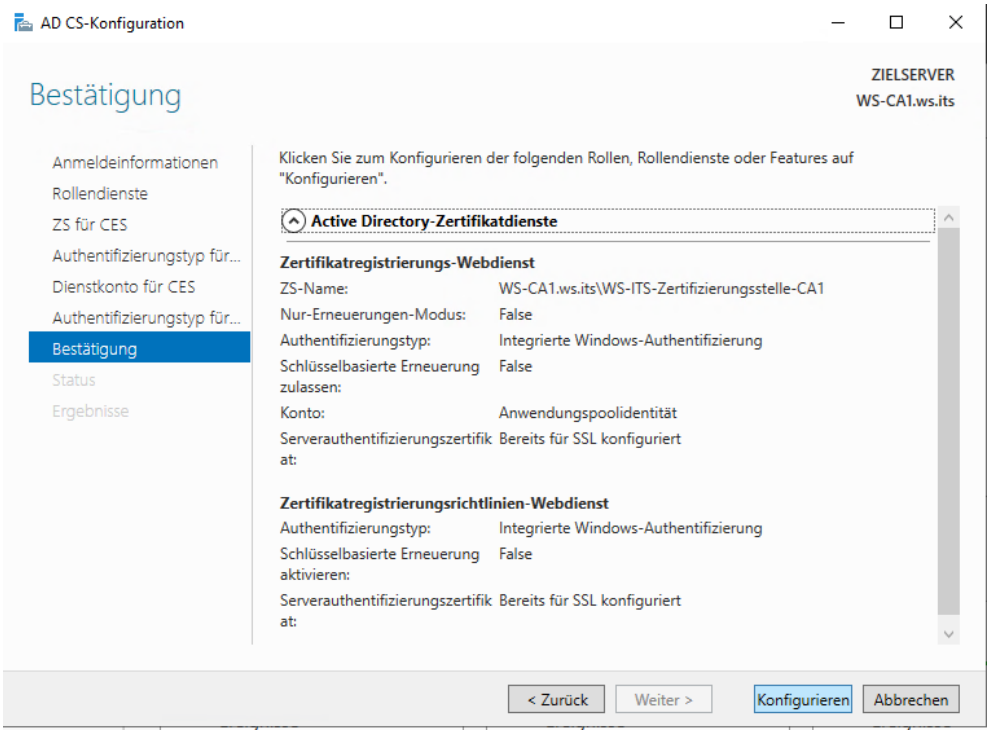
CES selber läuft im einfachen Sicherheitskontext des Application Pools im IIS. Damit spare ich mir einige Anpassungen rund um das Thema SPN:

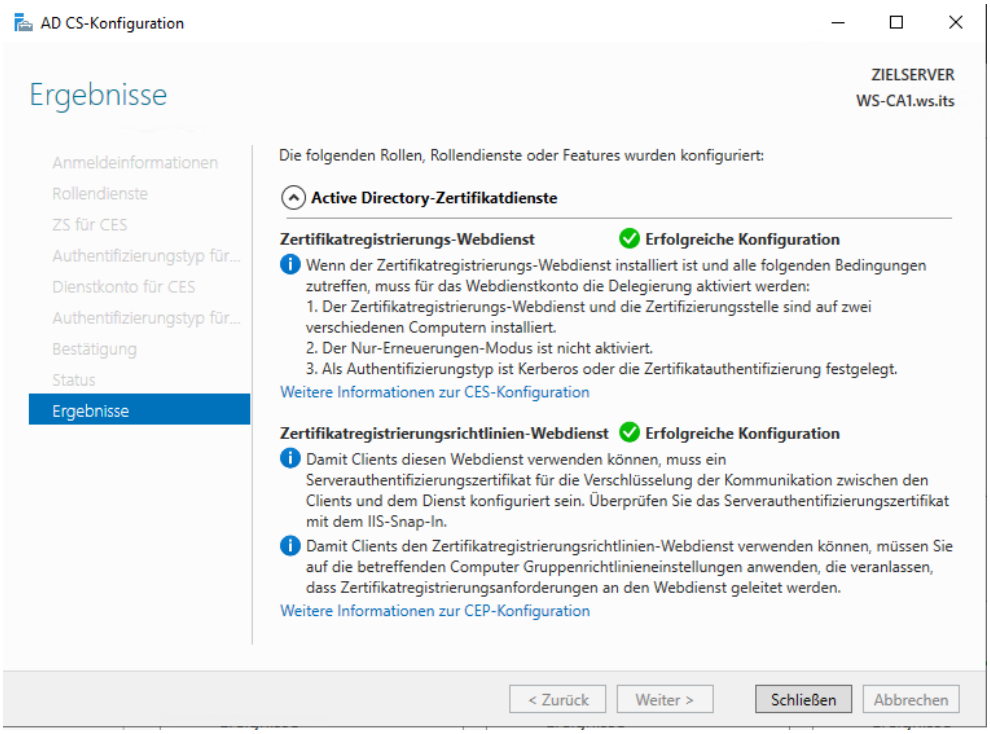


Mehr braucht mein CES nicht. Nahtlos geht es mit CEP weiter. Hier muss ich die gleiche Authentifizierung wie beim CES wählen:

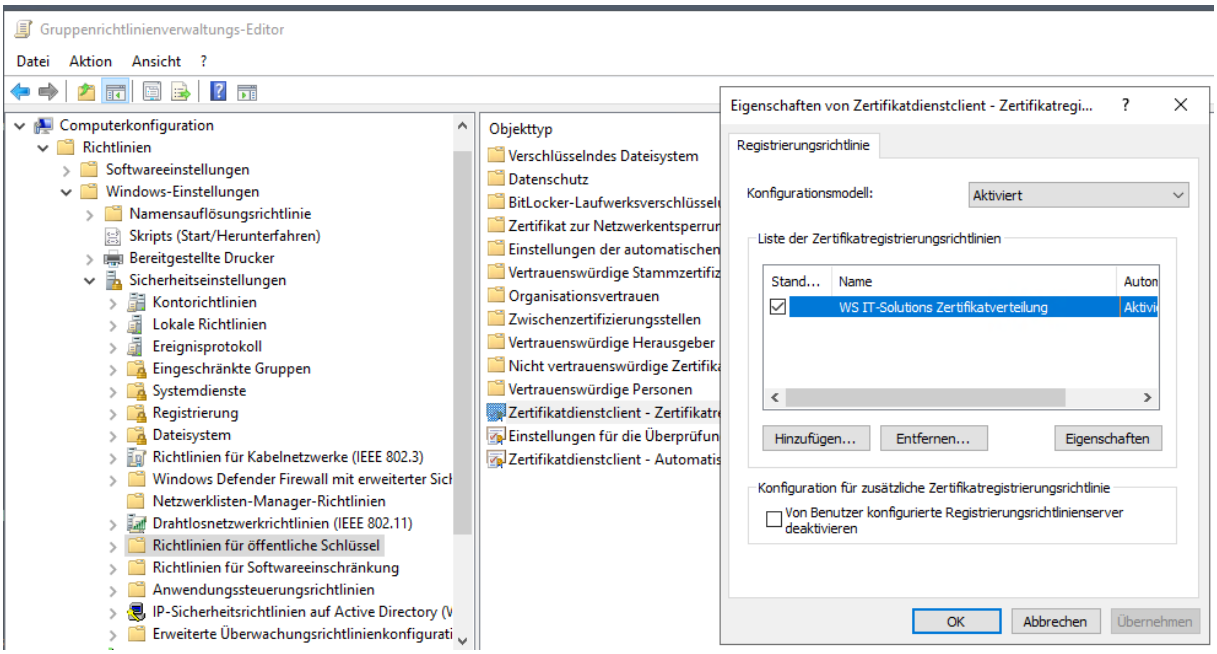


Dann ist alles angegeben. Der Assistent richtet beide Webservices ein:

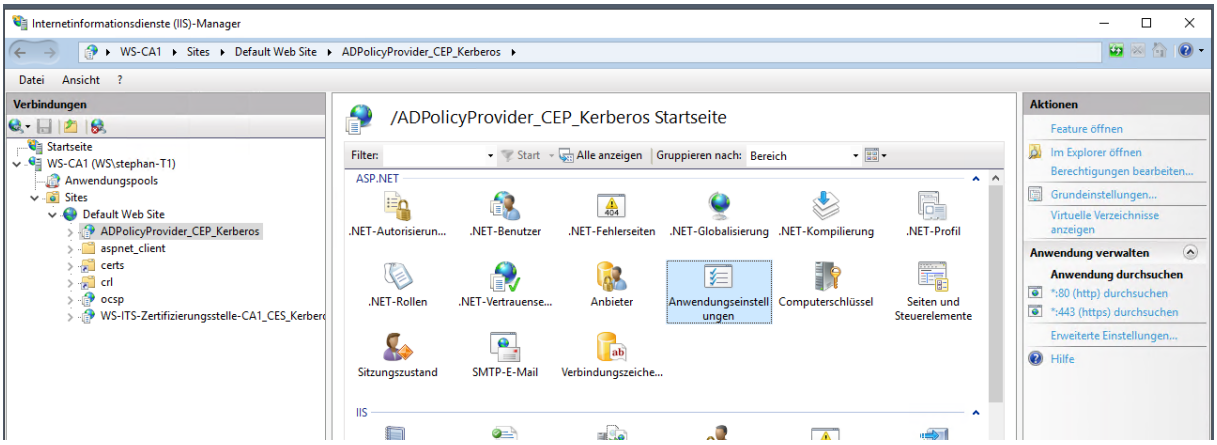




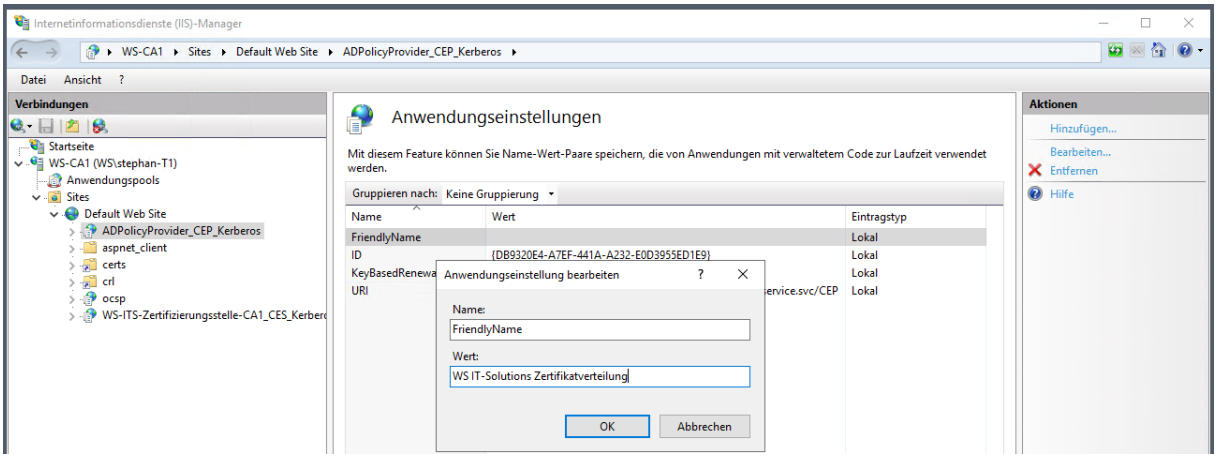
Ich hatte bereits vorher einen CEPCES im Einsatz. Daher muss ich meine Clients und Benutzer nicht mehr neu informieren. Die bestehende Gruppenrichtlinie passt unverändert:



CEPCES benötigt aber noch einige Anpassungen im IIS, die nicht vom Post-Deployment vorgenommen werden. Also geht es in den IIS in die Webanwendung vom CEP. Hier brauche ich die Anwendungseinstellungen:



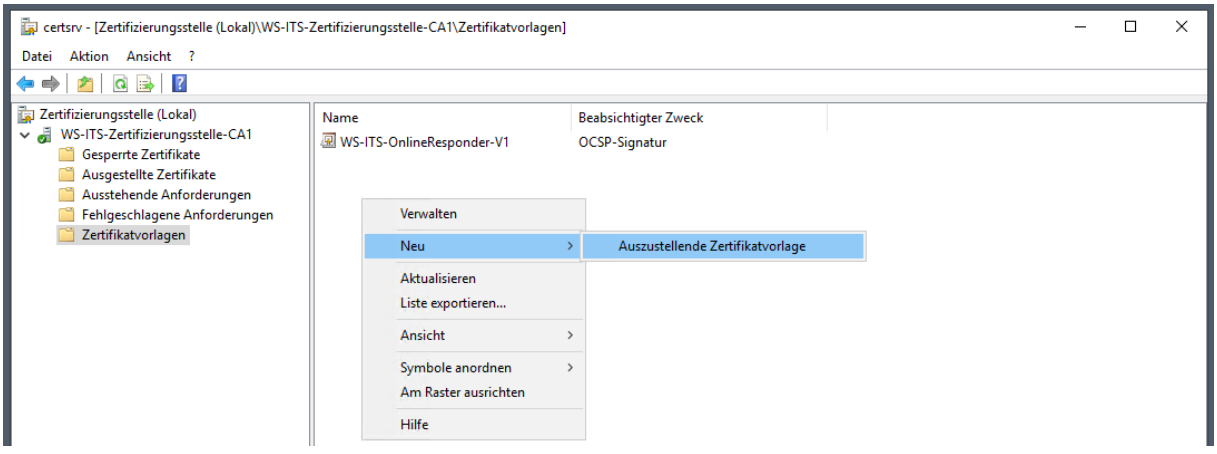
Das Feld Friendly Name darf nicht leer sein. Das ist aber leider der Default. Damit meine Gruppenrichtlinie weiter passt, trage ich den gleichen Text ein, den auch meine vorherige Zertifizierungsstelle verwendete:



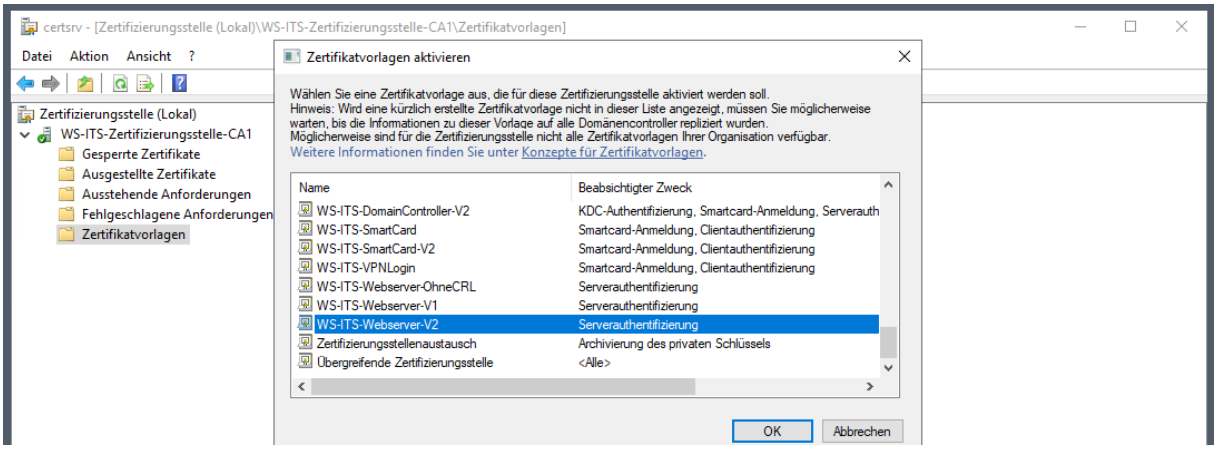
Damit ist der CEPCES einsatzbereit.

Testphase

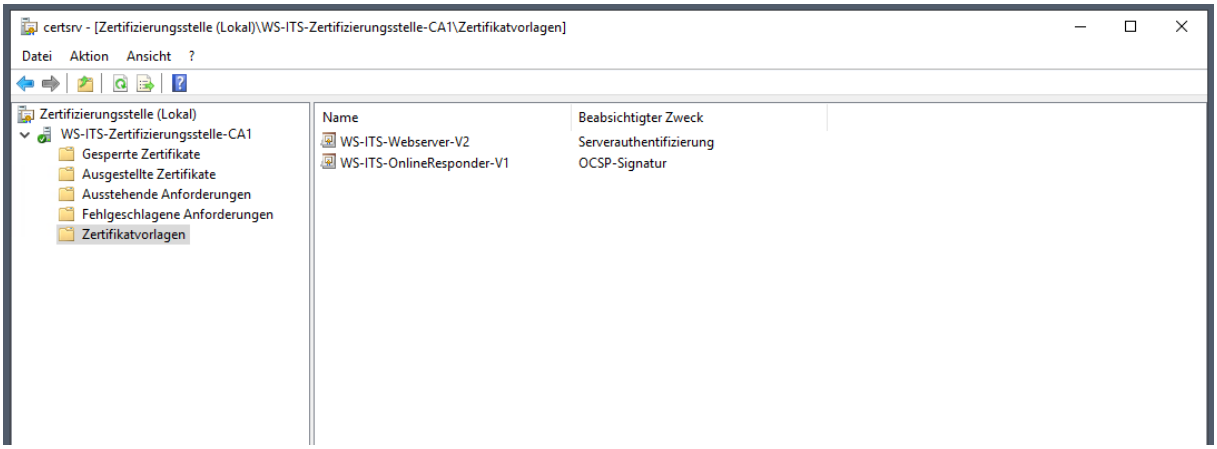
Bevor ich den Service testen kann, muss ich meiner Windows Zertifizierungsstelle noch weitere Zertifikatvorlagen zuweisen:



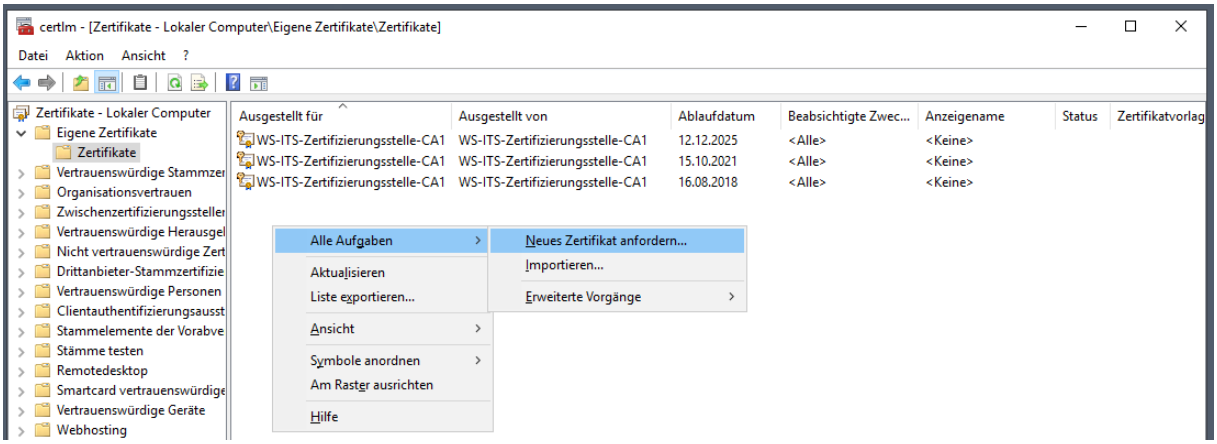
Für einen Testlauf nehme ich nur meine Webserver-Vorlage dazu. Diese kann kein AutoEnrollment und ich vermeide somit ungewollte Zertifikate, bevor meine Tests abgeschlossen sind:



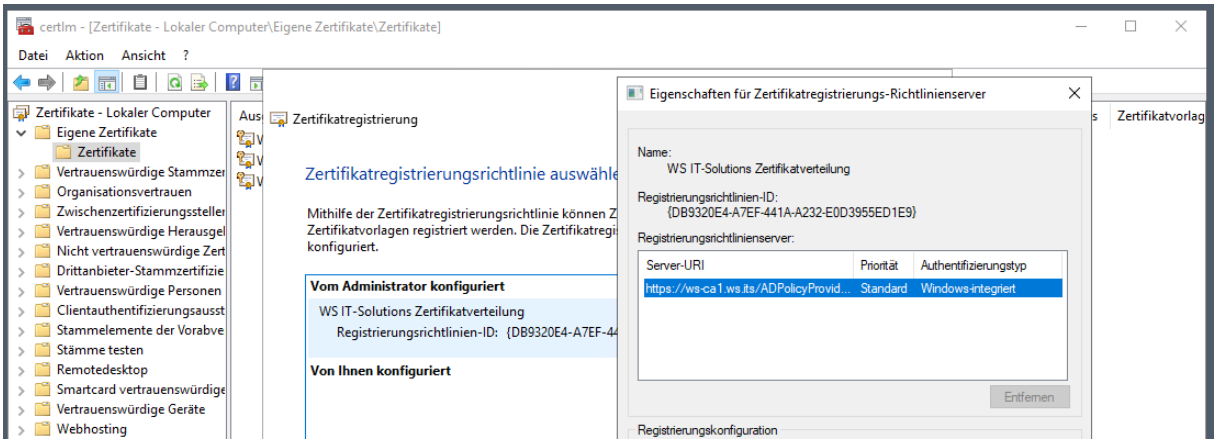
Jetzt sind 2 Vorlagen aktiv:



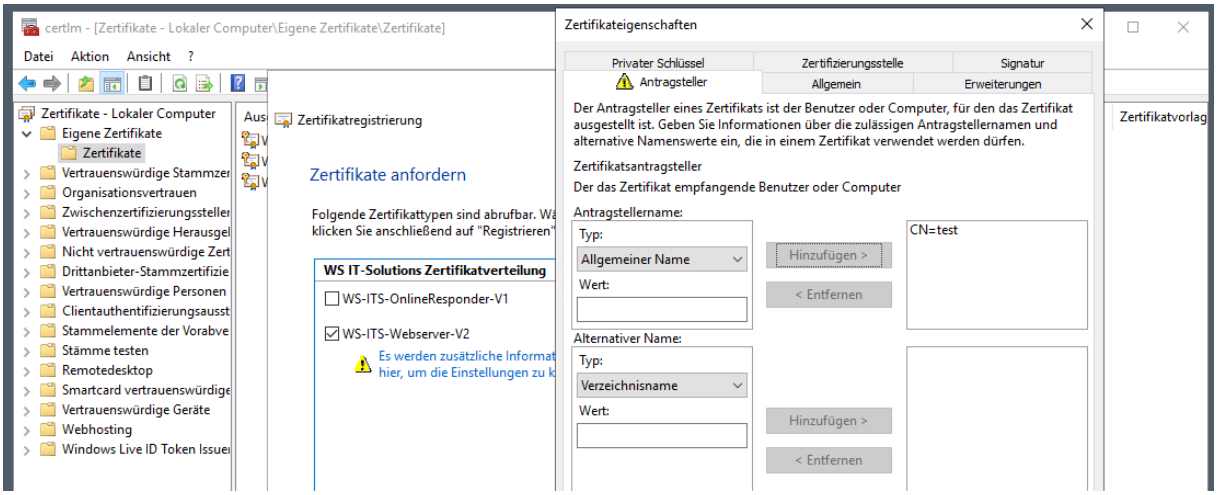
Es wird Zeit für einen Testlauf. Ich starte eine certlm.msc und frage ein neues Zertifikat an:



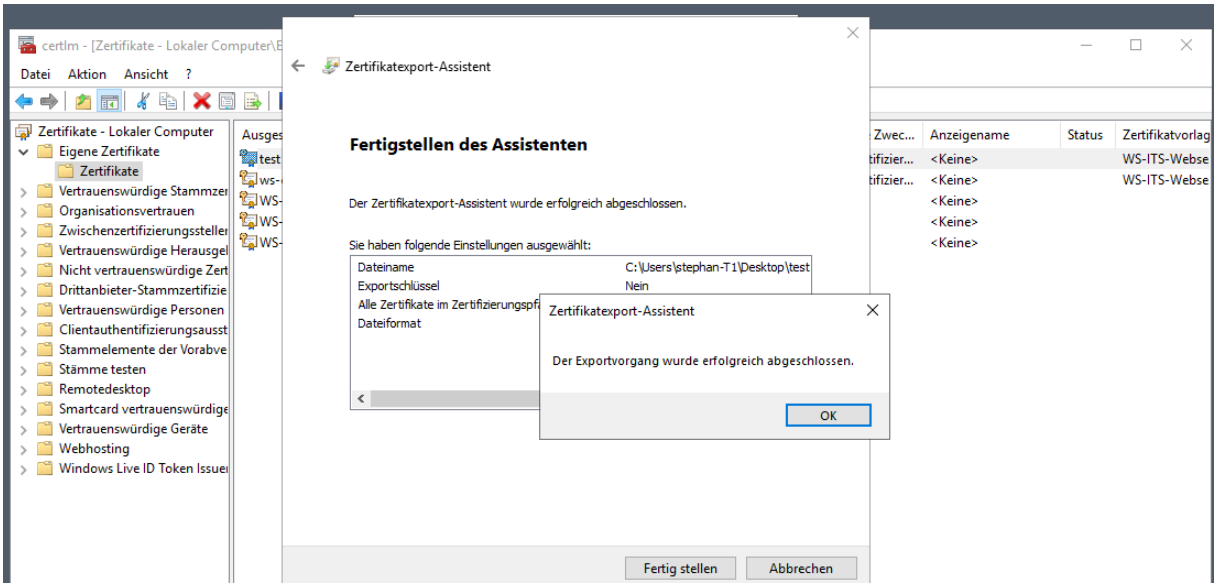
Die Management-Konsole befragt das Active Directory, welche Zertifizierungsstellen existieren. Da wird dann mein CEP mit seiner URL genannt:



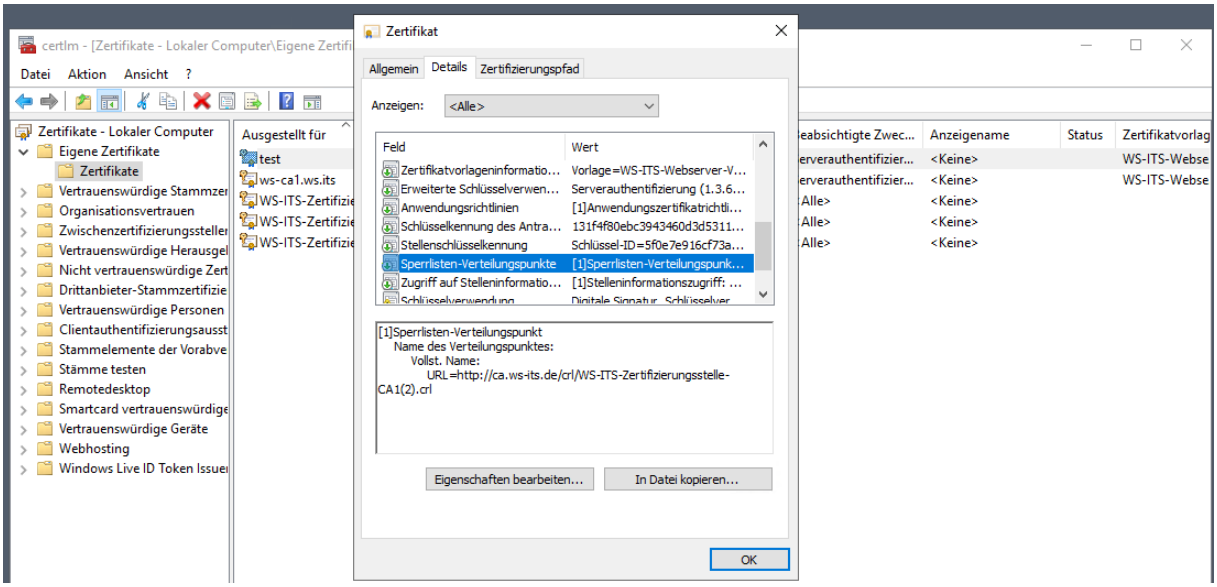
Der CEP listet mir nun die möglichen Zertifikatvorlagen auf. Ich wähle das Template für ein Webserver-Zertifikat. Da muss ich eine Angabe zum Antragstellernahmen vornehmen. Ich erstelle ein Test-Zertikat:



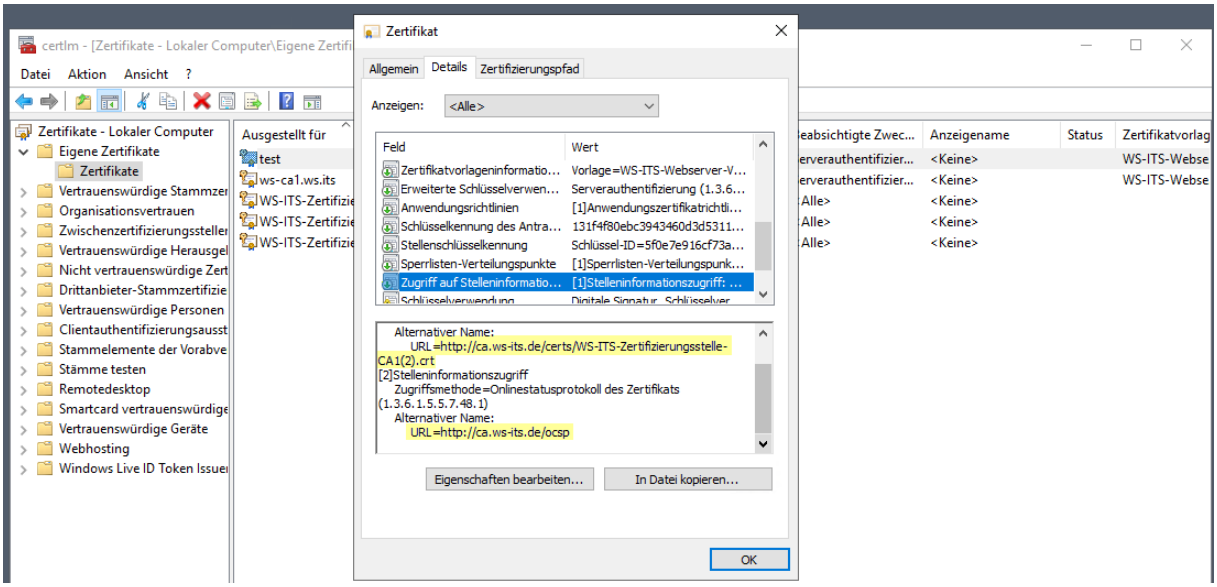
Die Registrierung läuft und ist wenige Sekunden später abgeschlossen:



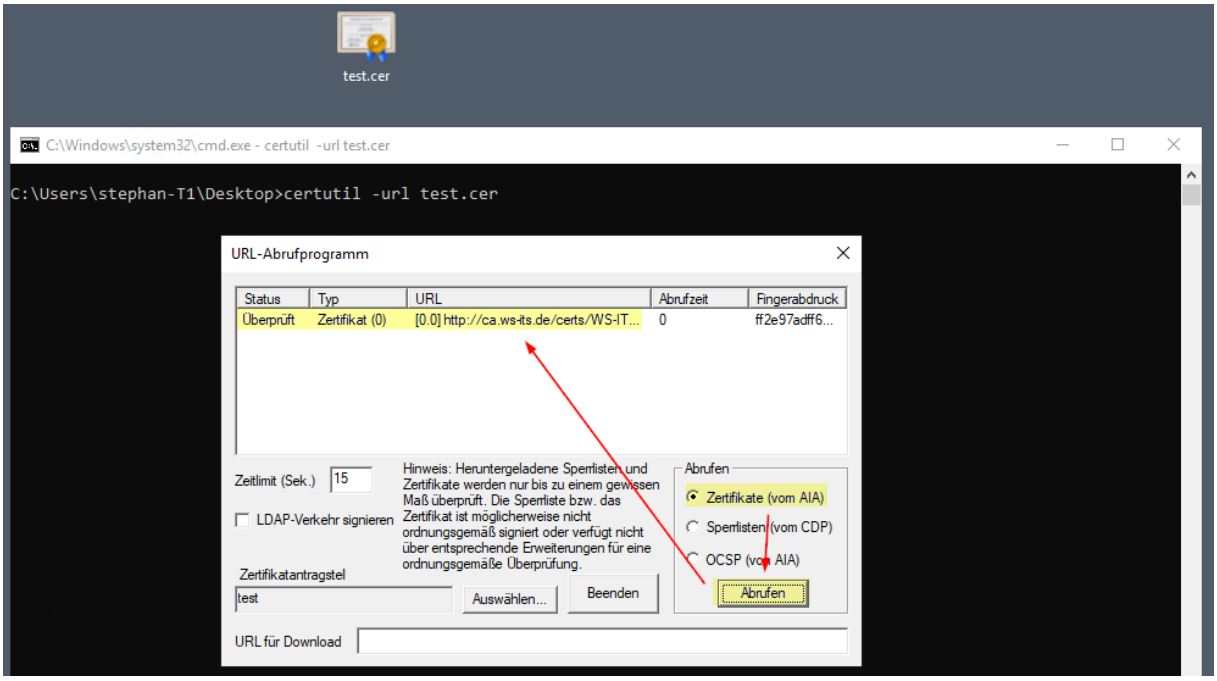
Danach kontrolliere ich die CDP und AIA-Informationen, die beim Ausstellen von der Zertifizierungsstelle übergeben werden. Das neue Zertifikat ist nur noch über eine Sperrliste über http verifizierbar:



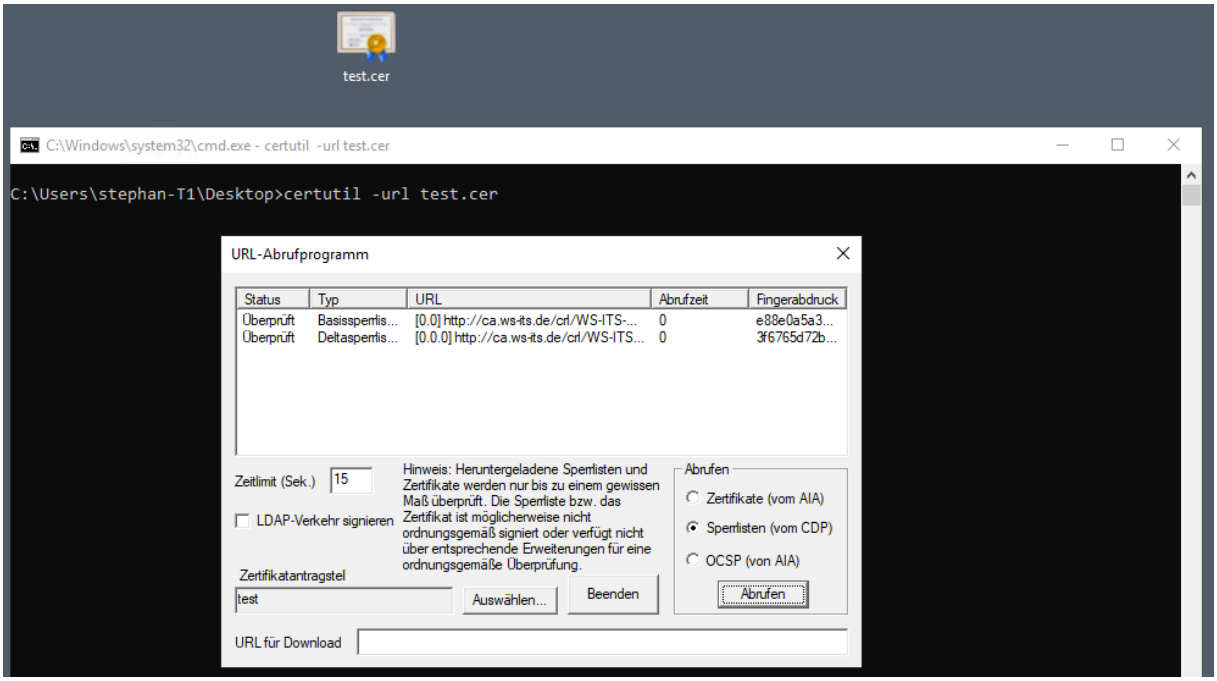
In den Stelleninformationen finde ich 2 Einträge: einer zeigt, wo ein Client das Zertifizierungsstellen-Zertifikat organisieren kann. Und der andere Link gibt die Position meines Online-Responders an:



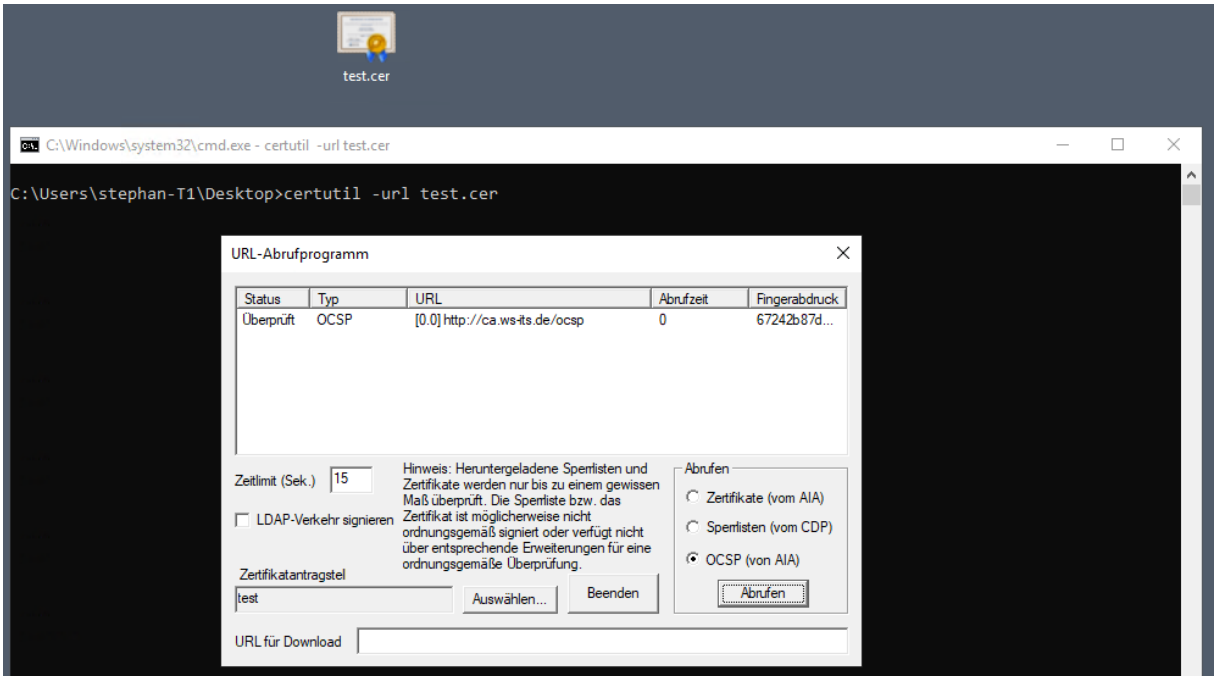
Die Ausstellung über CEPCES funktioniert also. Ebenso werden die Erweiterungsinformationen korrekt ausgegeben. Dann bleibt nun noch die Kontrolle des Online-Responders. Diese Funktion kann mit certutil grafisch verprobt werden. Dazu exportiere ich das eben ausgestellte Test-Zertifikat in eine cer-Datei und rufe certutil mit dem Parameter URL auf. In der grafischen Oberfläche kann ich nun die einzelnen Stelleninformationen und Sperrlistenoptionen prüfen. Ich beginne mit dem Download des Zertifizierungsstellen-Zertifikates. Das funktioniert wie erwartet:



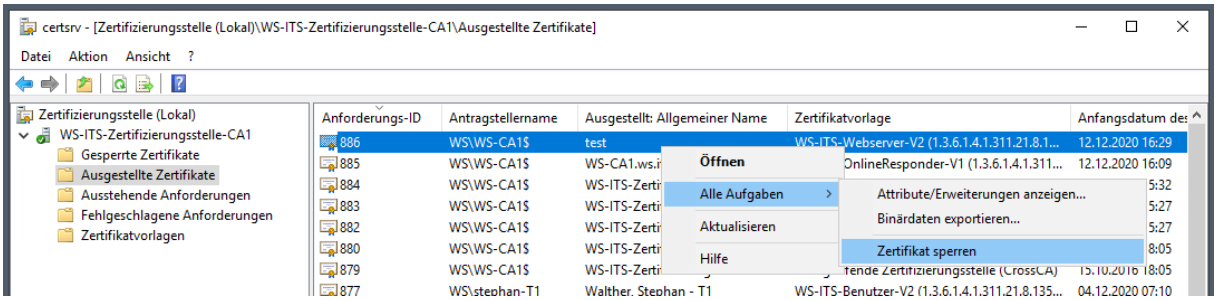
Auch der Download der klassischen Sperrlisten über http funktioniert einwandfrei:



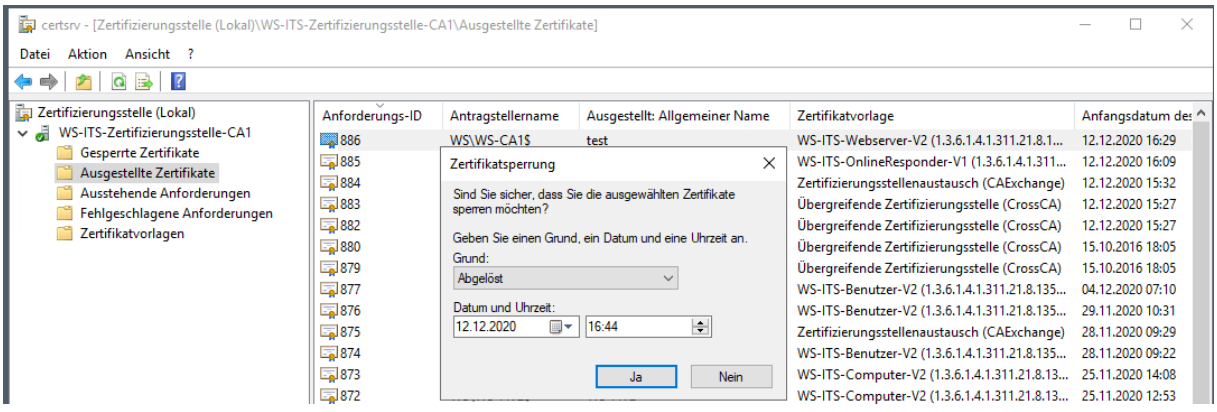
Und auch der Online Responder reagiert wie gewünscht:



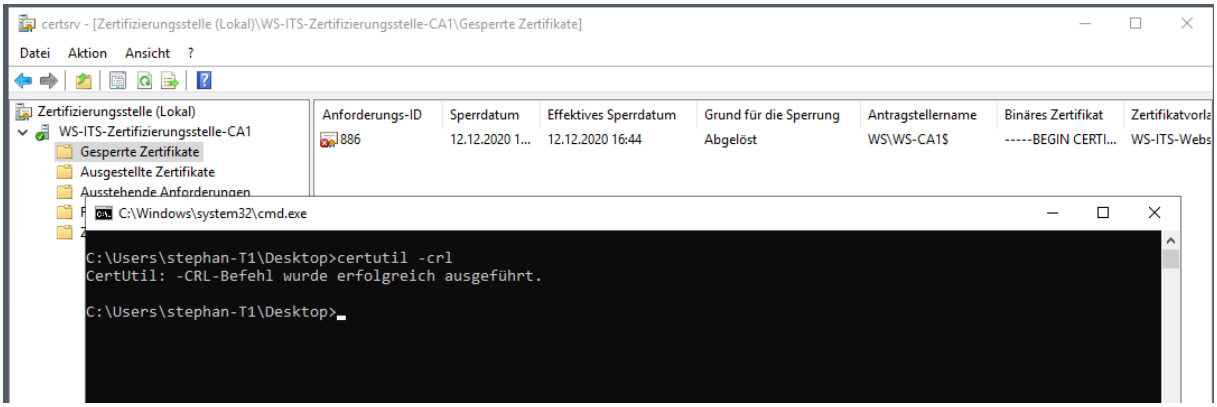
Aber ich möchte auch das „Echtzeit“-Sperrverhalten testen. Daher sperre ich in der Zertifizierungsstelle das Test-Zertifikat:



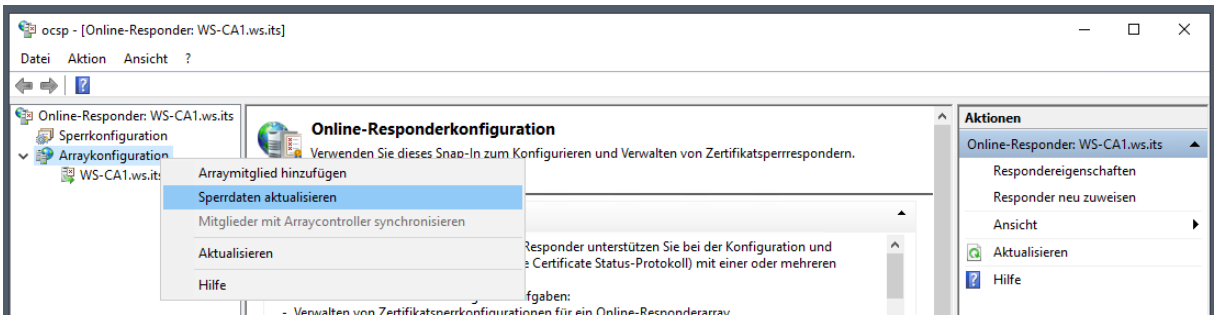
Üblicherweise wird hierfür eine Begründung mit angegeben:



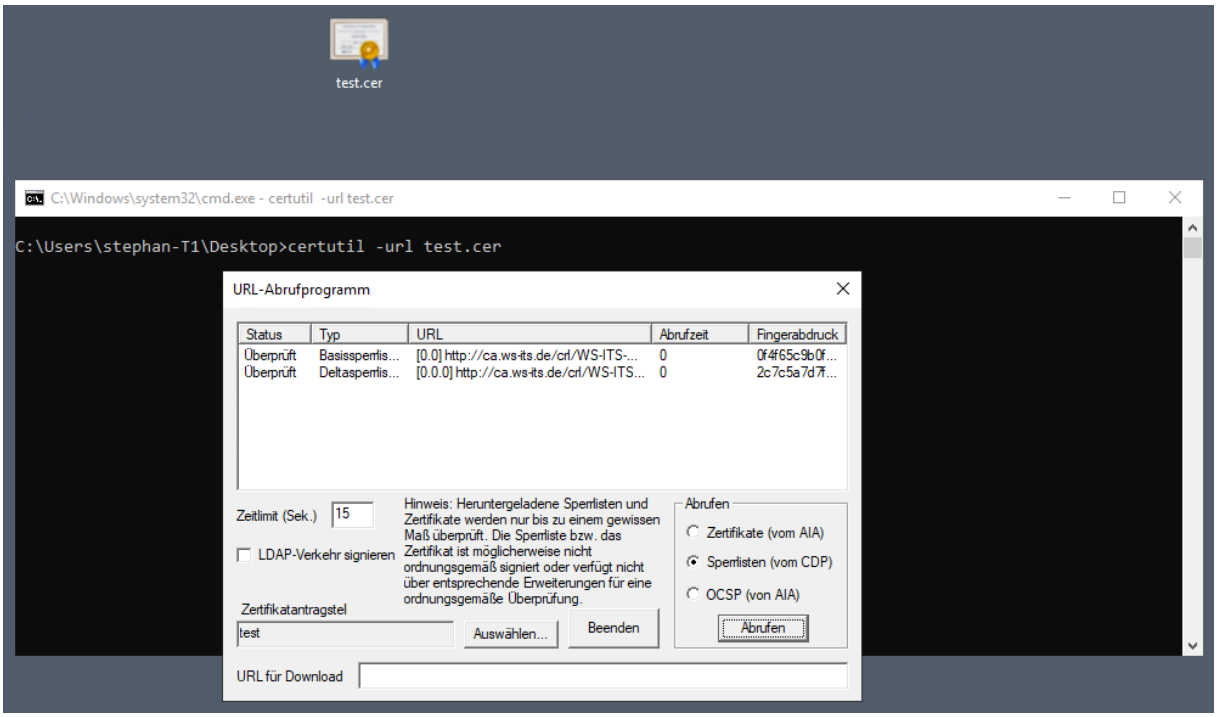
Nach dieser Aktion muss die Sperrliste außerhalb ihrer automatischen Veröffentlichung manuell erstellt werden. Das geht mit der cmd recht schnell:



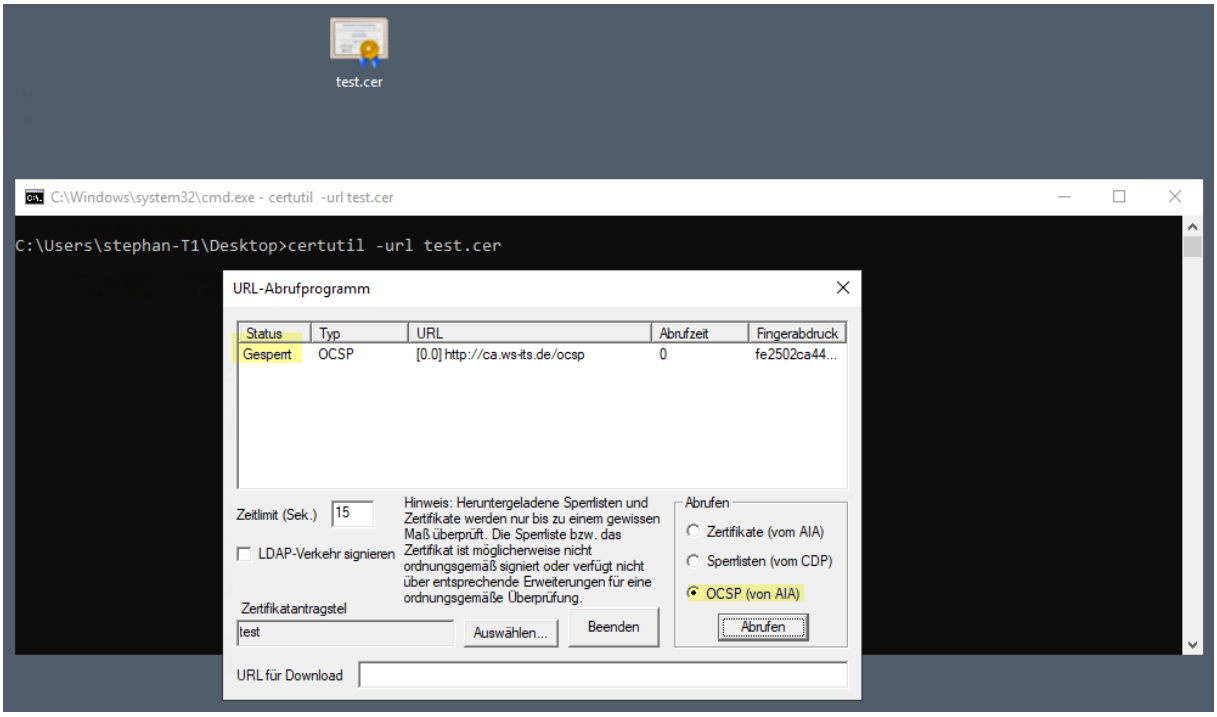
Beim Einsatz eines Online Responders hängt es nun vom Aktualisierungsinterval ab. Ich hatte 5 Minuten angegeben. Die könnte ich nun noch abwarten. Aber ich kann es mit der Management Konsole auch beschleunigen:



Ein Sperrlistentest mit certutil zeigt, dass die klassische Sperrlistenfunktion das Zertifikat noch nicht als zurückgezogen deklariert, denn mein Client hat sich beim ersten Test die Datei über http heruntergeladen und die Datei im Cache ist noch gültig. Er wird sich also die neue Version noch nicht herunterladen:



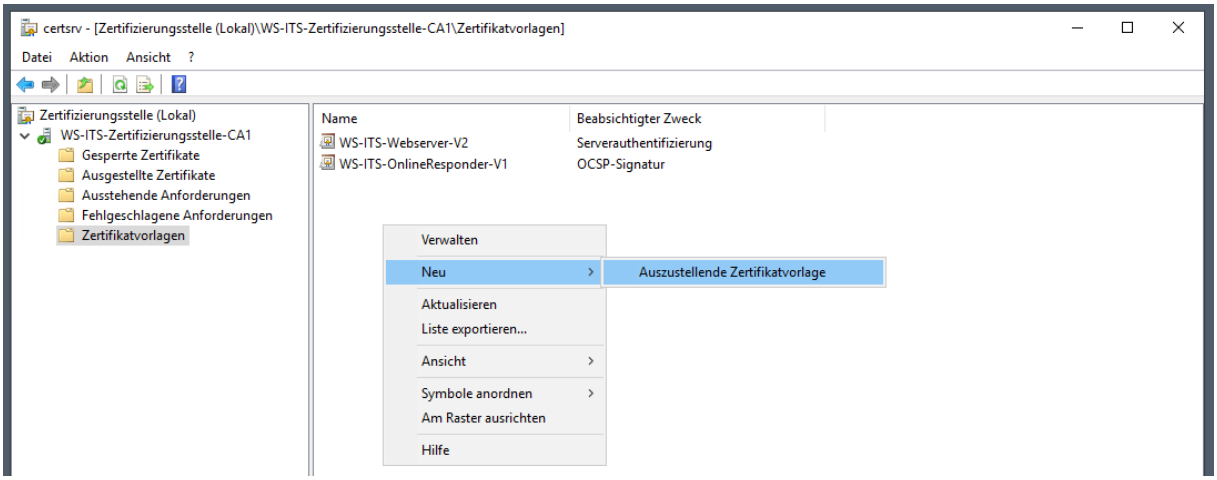
Aber der Online Responder hat diese neue Sperrliste bereits verarbeitet und reagiert dementsprechend mit einem Sperrhinweis. Damit ist diese Sperrfunktion wesentlich schneller und agiler als die Verwendung der klassischen Sperrlisten:



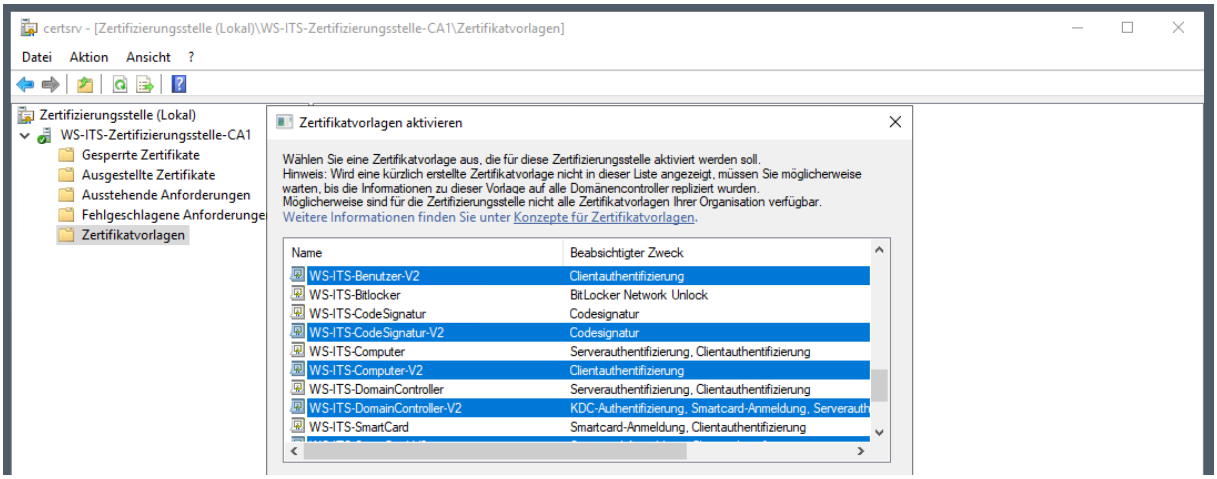
Meine Zertifizierungsstelle hat alle Tests bestanden. Es kann also mit der Aktivierung der Standardfunktionen weiter gehen.

Aktivierung der PKI

Meine Computer und Benutzer werden über Gruppenrichtlinien angewiesen, die PKI regelmäßig zu besuchen, um dort nach Zertifikatvorlagen mit Auto Enrollment zu suchen. Solange meine Zertifizierungsstelle also keine passenden auszustellende Vorlagen hat, passiert nichts. Das kann ich nach meinen erfolgreichen Tests nun ändern. Ich nehme weitere Vorlagen zur Ausstellung dazu:



Mein Vorlagenkatalog ist recht überschaubar:

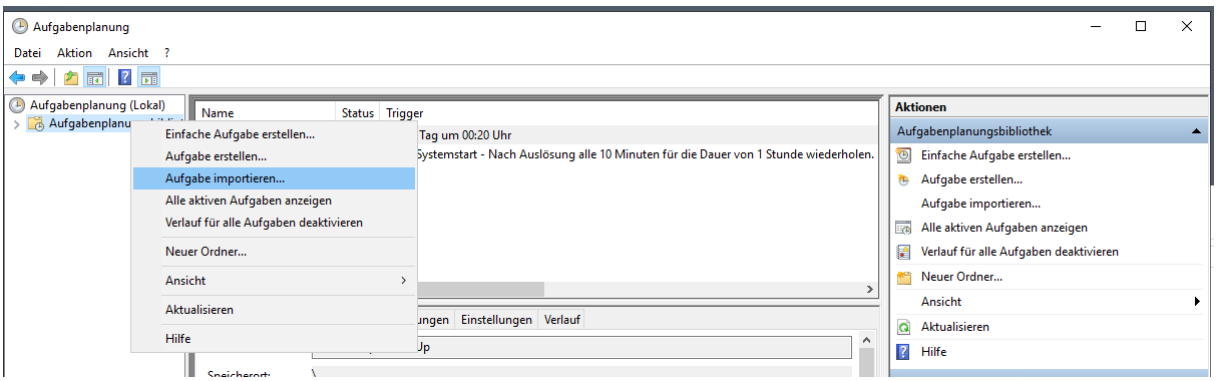


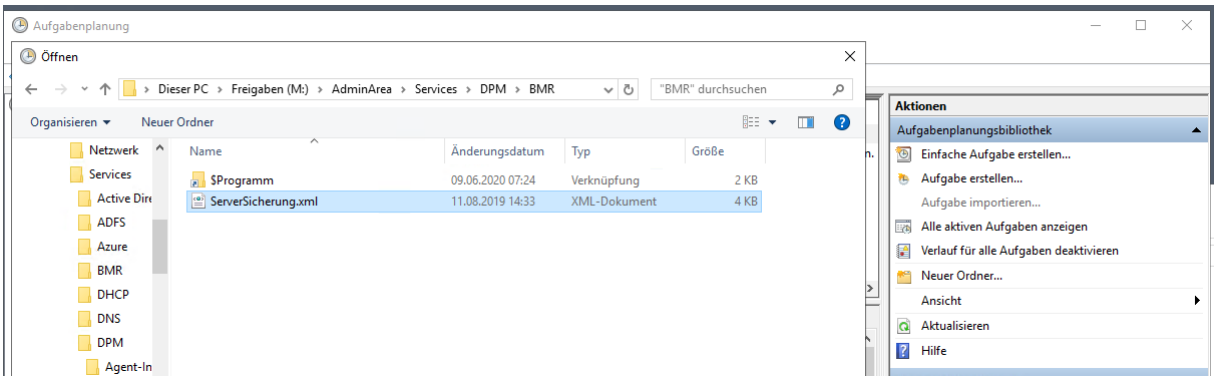
Damit kann meine PKI ihre Arbeit aufnehmen.

Nacharbeiten

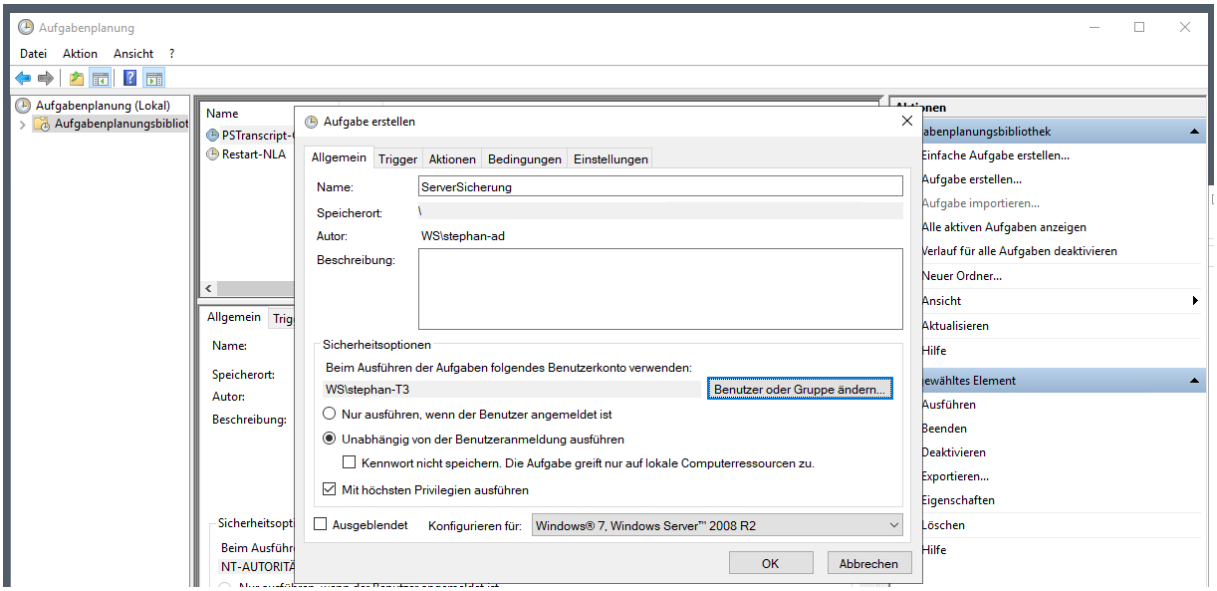
Konfiguration der Datensicherung

Es folgen die üblichen Standardarbeiten. Eine wichtige ist die Konfiguration der Datensicherung. Die Zertifizierungsstelle werde ich wieder mit der Windows Serversicherung sichern. Dafür registriere ich wieder meine Standardaufgabe in der Aufgabenplanung:

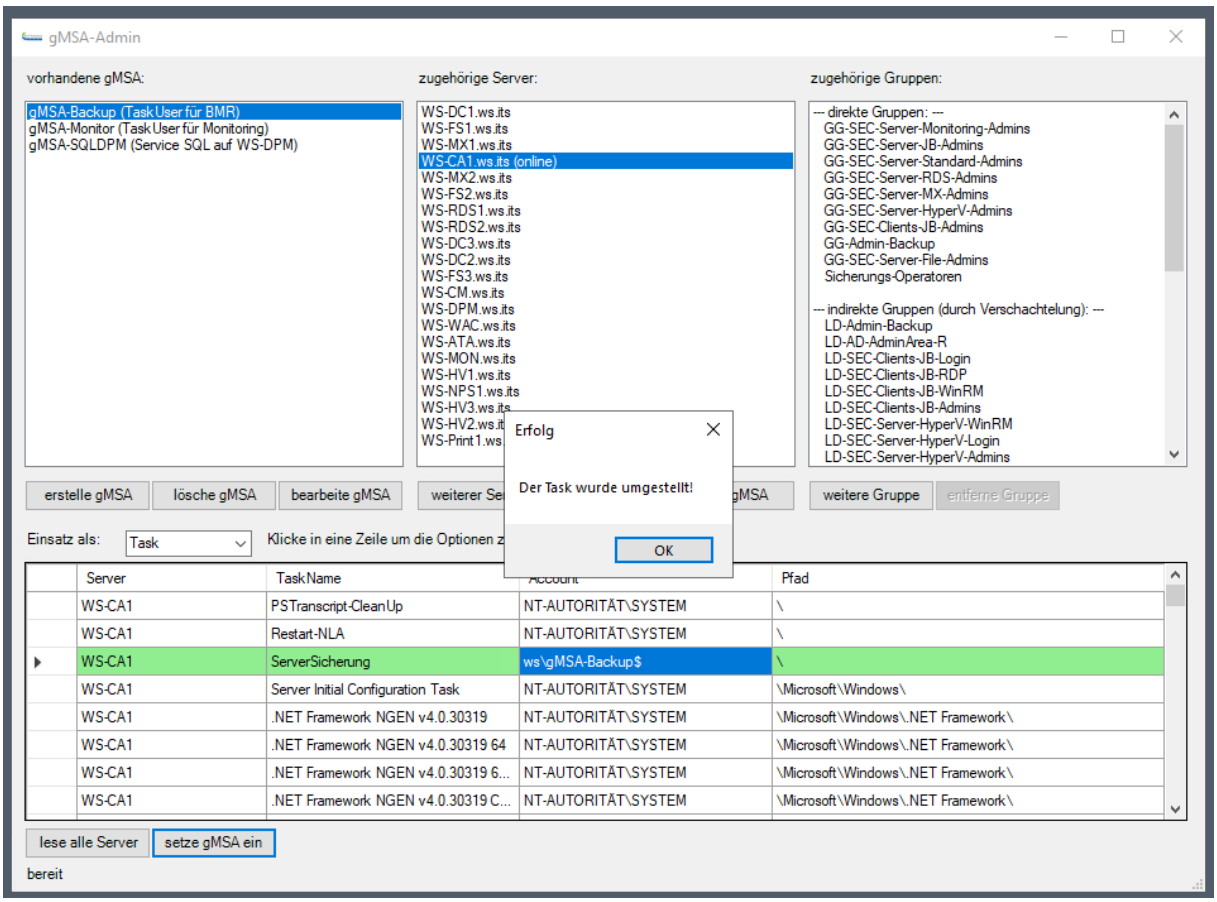




Der Account ist wieder nur als Dummy eingetragen:



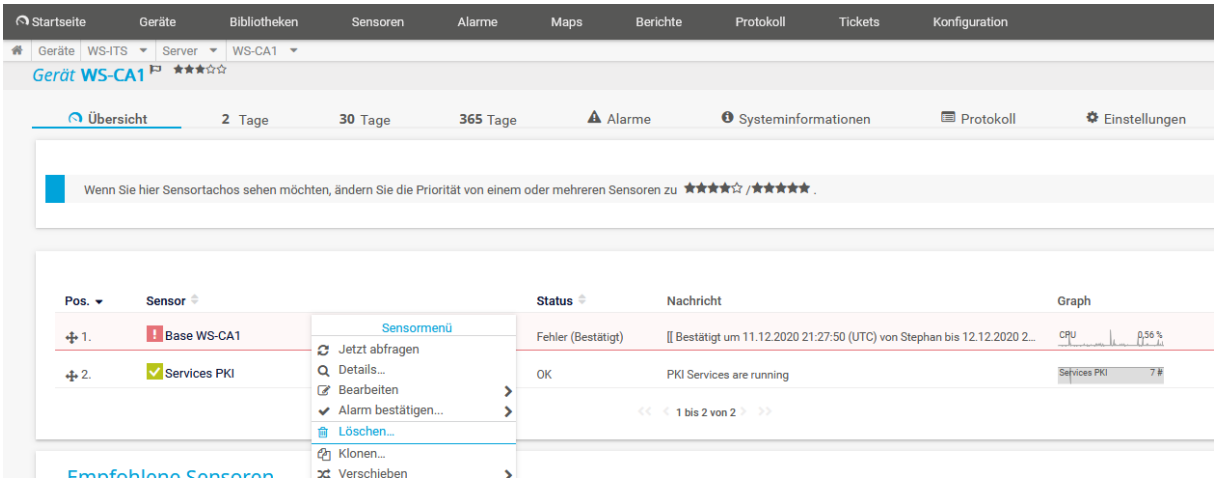
Über mein gMSA-Script kann ich nun vom Domain Controller aus den Sicherungsaccount durch einen Group Managed Service Account ersetzen:



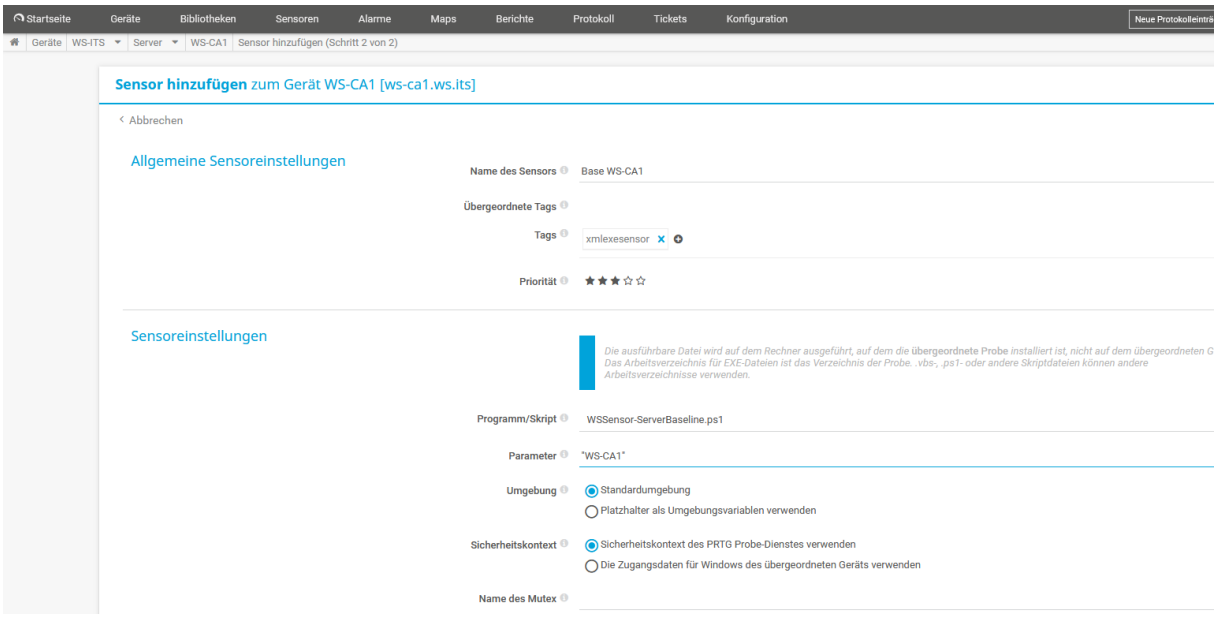
Damit ist alles erledigt. Die Sicherungsdefinition liegt ja zentral in meiner Konfiguratiosdatei. Das genügt mir.

Monitoring

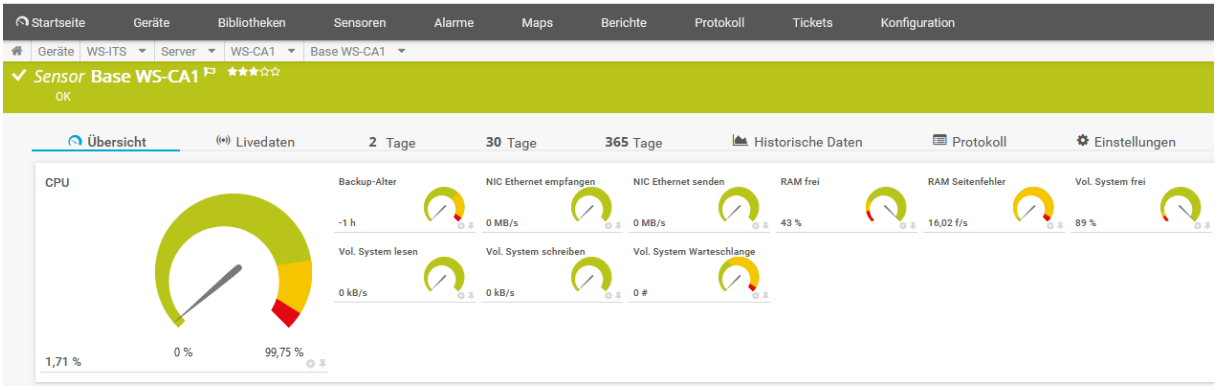
Weiter geht es mit dem Monitoring. Die aktuellen Sensoren zeigen meinen Base-Sensor im Fehlerstatus. Das liegt am Fehlen der alten Systemfestplatte, die über ihre GUID referenziert wurde. Ich lösche diesen alten Sensor im Webportal:



Danach kann ich ihn neu erstellen. Der Sensor ist ein eigens programmierter. Er gehört nicht zum Standard vom PRTG:

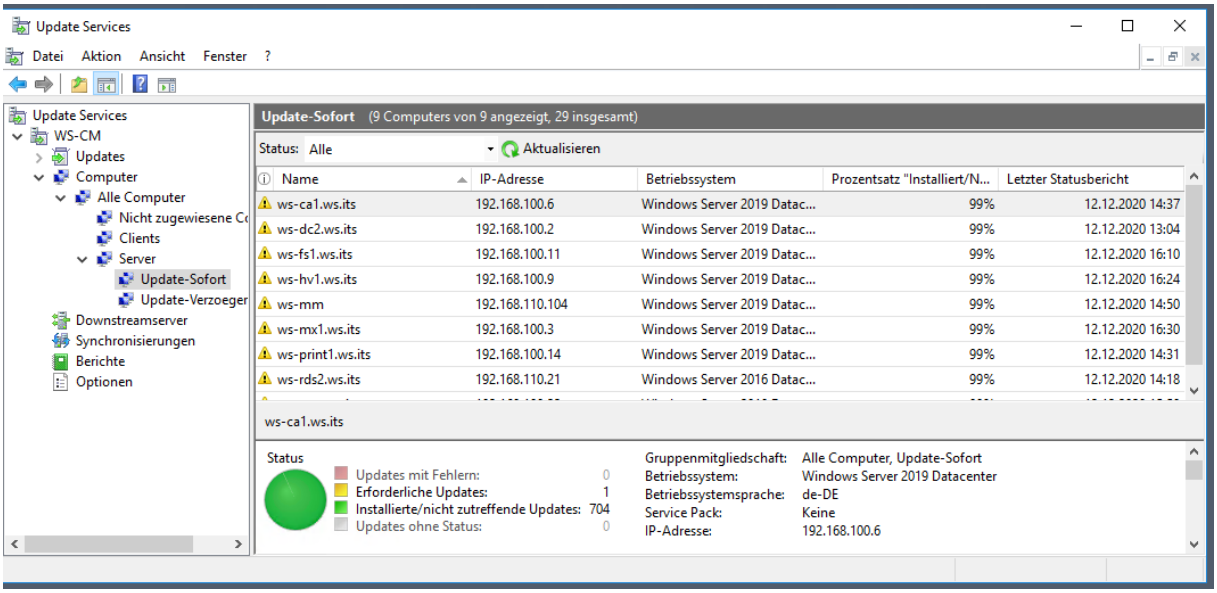


Nach wenigen Sekunden sind die Counter im Sensor geladen:



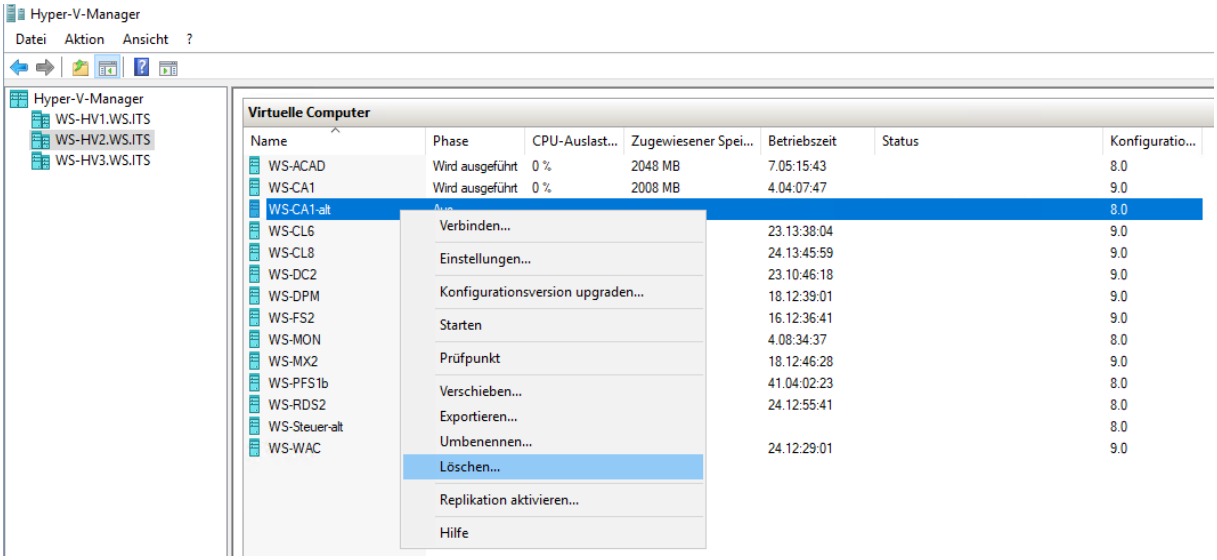
Updates

Im WSUS verschiebe ich den neuen Server noch in den passenden Update-Container. Den Rest erledigen meine Gruppenrichtlinien:

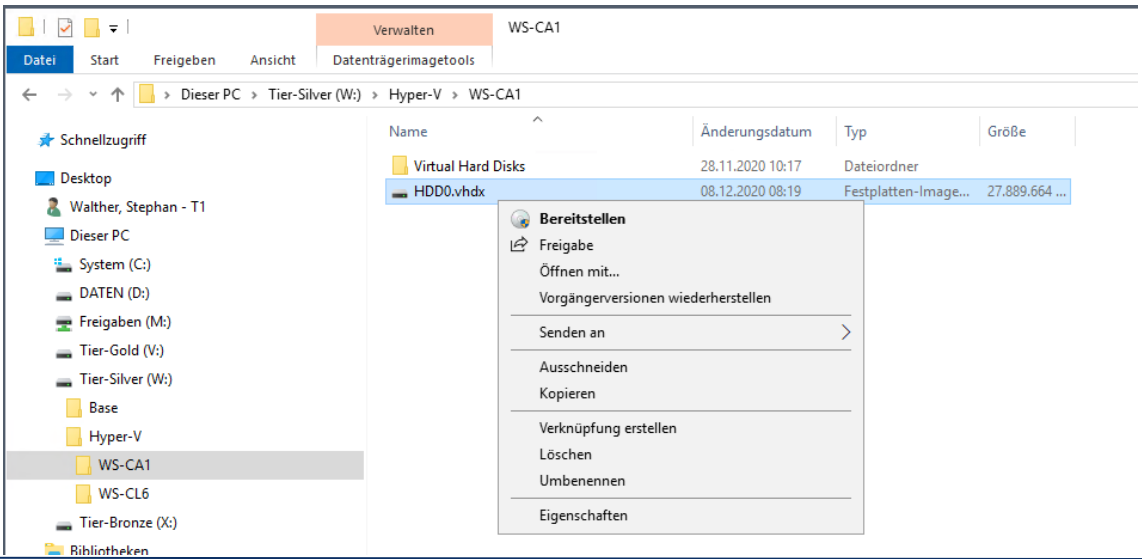


Bereinigungen

Im Hyper-V steht noch der alte VM-Eintrag. Diese VM kann ich nun löschen:



Die dazugehörige Festplattendatei muss manuell bereinigt werden:



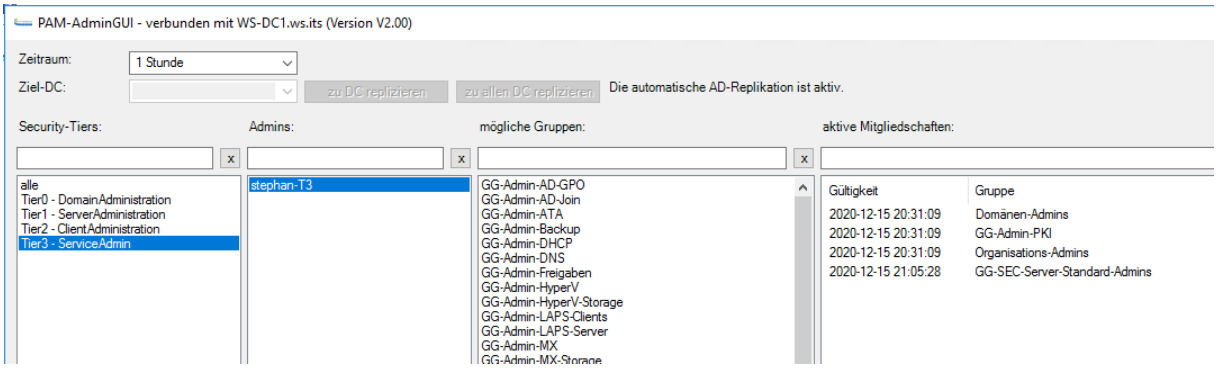
Problem Smartcard Logon

Ich verwende auf einem System eine Smartcard für die Anmeldung. Weil ich heute zufällig vor Ort bin, tausche ich das Zertifikat gleich manuell aus. Aber nach dem Austausch hängt die Anmeldung mit der neuen Smartcard mit dieser Fehlermeldung:

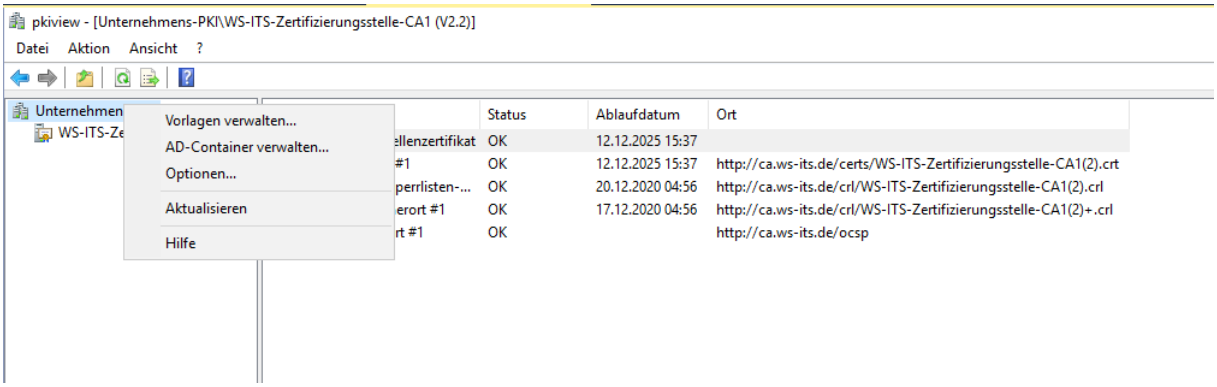
Da hat etwas bei der Neuausstellung des Zertifizierungsstellen-Zertifikates nicht geklappt. Für eine Smartcard-Anmeldung muss der Domain Controller dem Aussteller des Smartcard-Zertifikates vertrauen. Das ist üblicherweise vollautomatisch und wird unsichtbar im Hintergrund konfiguriert. Ich nutze aber ein angepasstes Administrationsmodell. Und bei dem hatte ich ja auch schon Probleme bei der CRL-Veröffentlichung im Active Directory. Offenbar wurde hier noch etwas im AD vergessen…

Ich führe das Problem also auf mein administratives Tier-Management zurück, in dem ich eben nicht alles als Domain Admin oder Enterprise Admin durchführe.

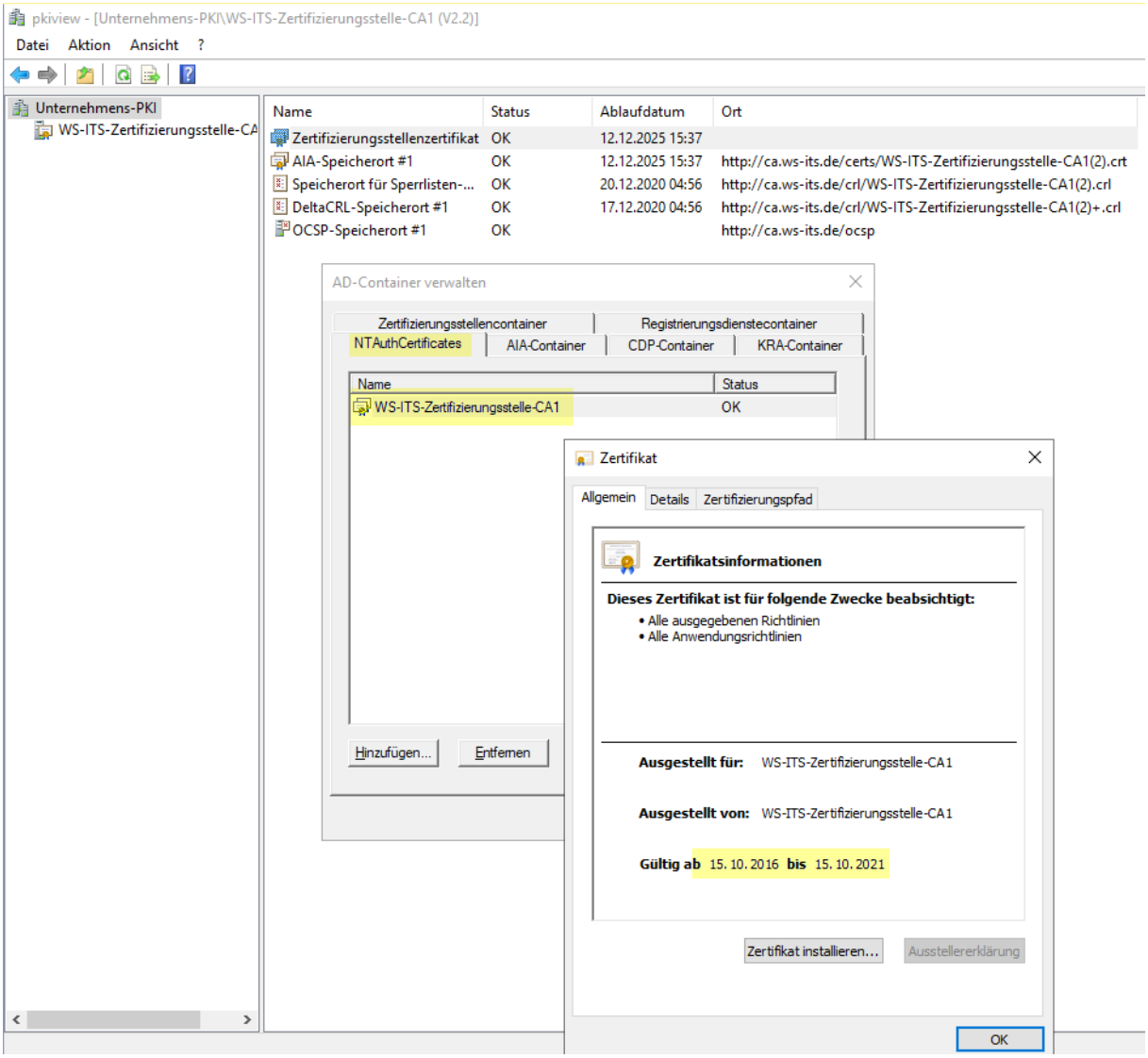
Aber ich kenne die notwendigen Schritte: Meine Domain Controller müssen das neue Zertifizierungsstellen-Zertifikat als NTAuth-Zertifikat konfiguriert bekommen. Um das zu konfigurieren, berechtige ich meinen T3-Admin als PKI-Admin und als Enterprise Admin:



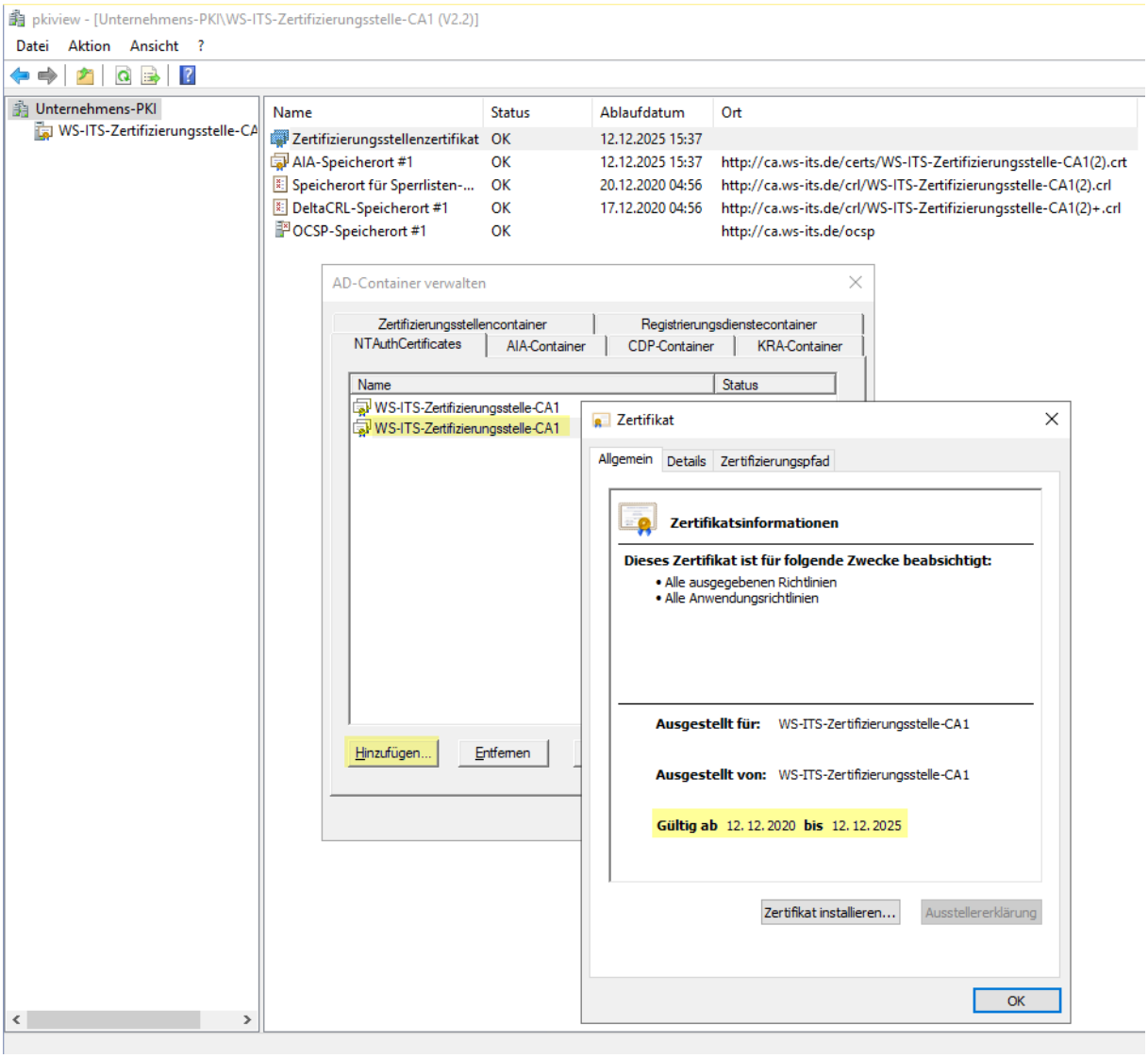
Mit diesen Rechten ausgestattet starte ich eine PKIVIEW-Konsole. Hier kann man nicht nur die PKI überprüfen, sondern auch Anpassungen im AD-Kontext vornehmen:



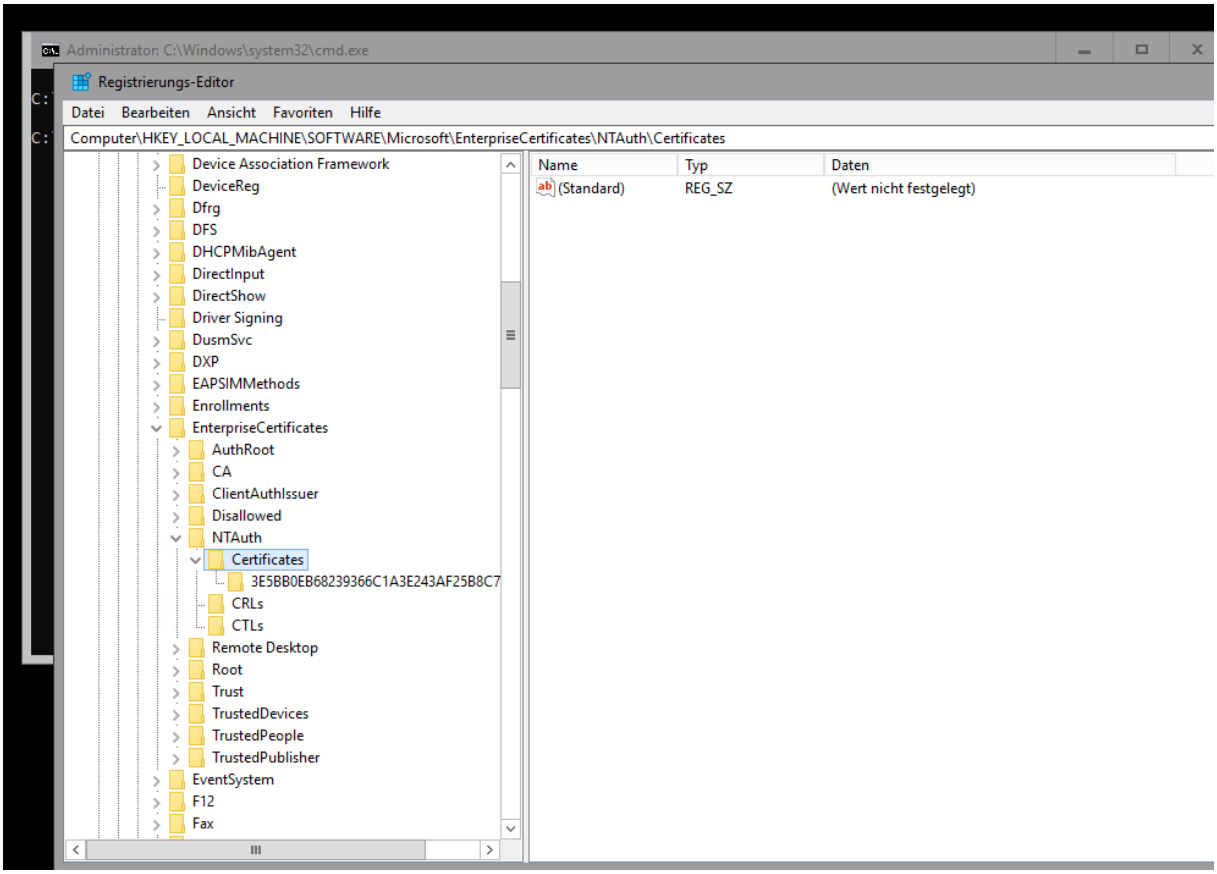
Für eine Smartcard-Anmeldung werden vertrauenswürdige NTAuthCertificates verwendet. Und genau hier liegt das Problem: Das neue Zertifizierungsstellen-Zertifikat fehlt in der Auflistung! Gelistet ist nur das alte:



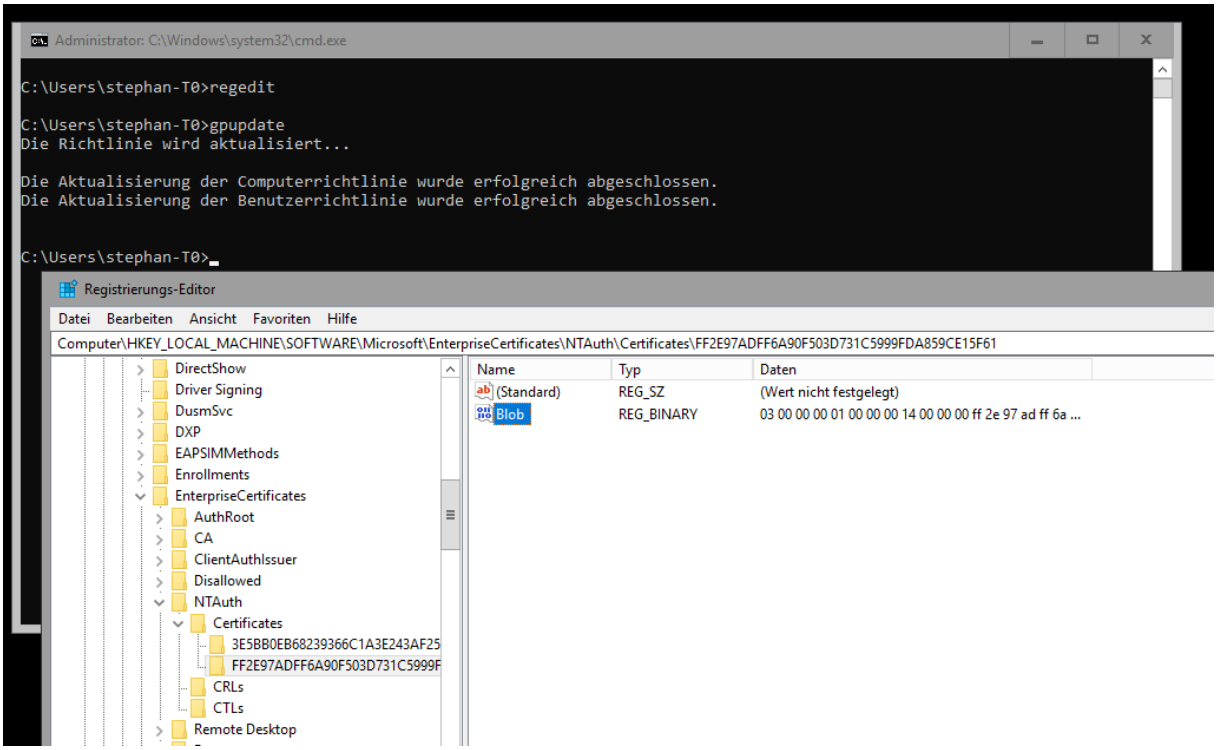
Über den Schalter „Hinzufügen“ kann ich nun das neue Zertifikat zuweisen:



Dennoch wird die Anmeldung mit einer Smartcard erst funktionieren, wenn der prüfende Domain Controller über diese Veränderung informiert wurde. Denn jeder DC wird in seiner Registry die Zertifikate cachen. Bei mir ist es einfach, denn der Client mit der Smardcard steht in meiner Außenstelle und dort gibt es nur einen Domain Controller. In der Registry finde ich nur einen Zertifikateintrag für das alte Zertifizierungsstellen-Zertifikat:



Mit einem simplen gpupdate kann ich den Domain Controller dazu bringen, die Daten neu zu laden. Das hat nichts mit Gruppenrichtlinien zu tun. Aber der erzwungene Kontakt zu einem Domain Controller aktualisiert eben auch diese Daten. Und auch Domain Controller verhalten sich hier eben wie alle anderen Server und Clients. Nach dem gpupdate wird auch das neue Zertifikat lokal gelistet:

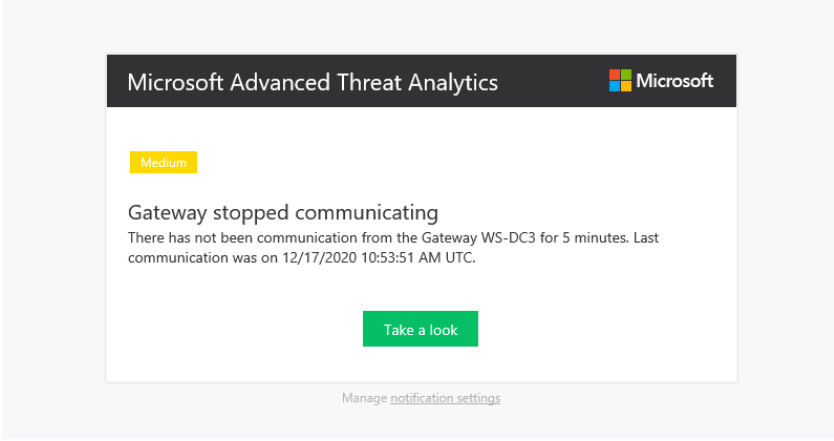


Nachdem der Domain Controller das neue Zertifizierungsstellen-Zertifikat gespeichert hat, akzeptiert er auch die Smartcard-Anmeldung vom Client:

Damit wäre auch dieses Problem behoben.

Problem Snort

Das nächste Problem lässt nicht lange auf sich warten. Mein Advanced Threat Analytics (ATA) meldet mir per Mail, dass es meinen WS-DC3 nicht mehr erreichen kann. Das sieht erst einmal nicht nach einem zertifikatbasierten Problem aus:



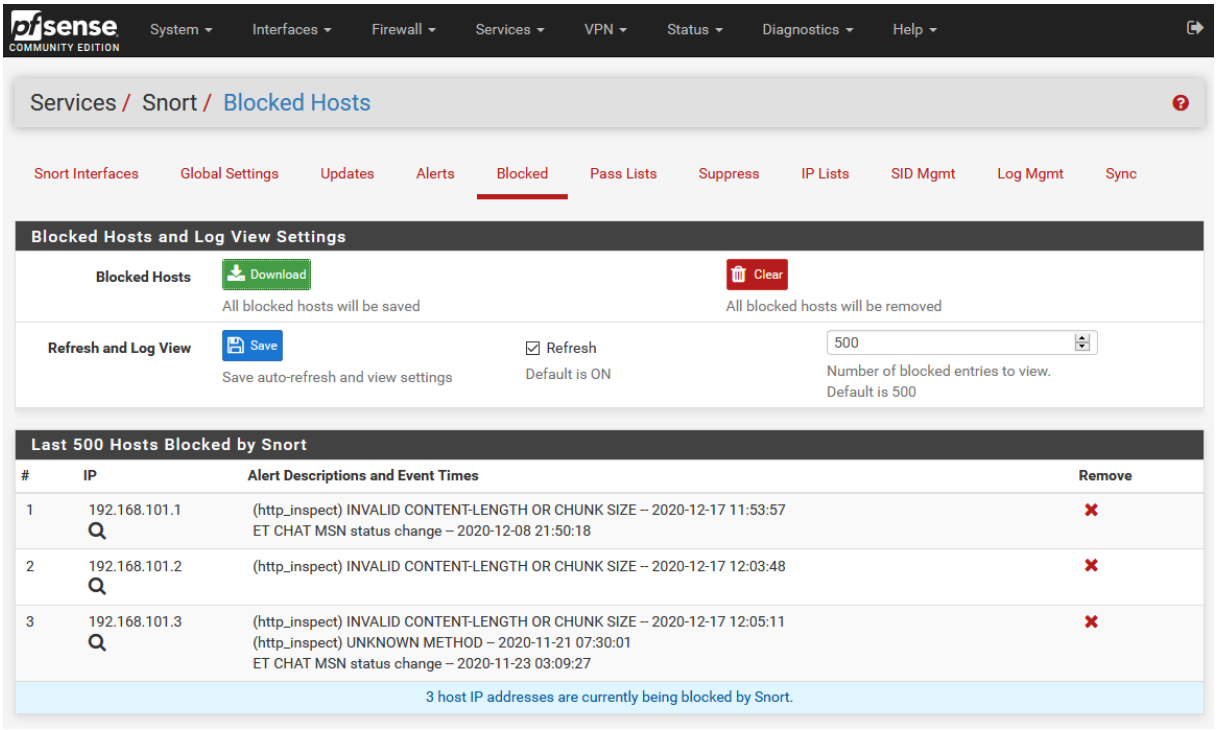
Aber direkt danach erhalte ich eine weitere Mail von meinem Monitor-Server. Dort läuft ein PowerShell-Script, dass die Logfiles im dort installierten SYSLOG analysiert. Im SYSLOG protokolliert meine Firewall und mein IPS diverse Meldungen. Und eine davon zeigt meinen WS-DC3:

neue IPS-Alerts

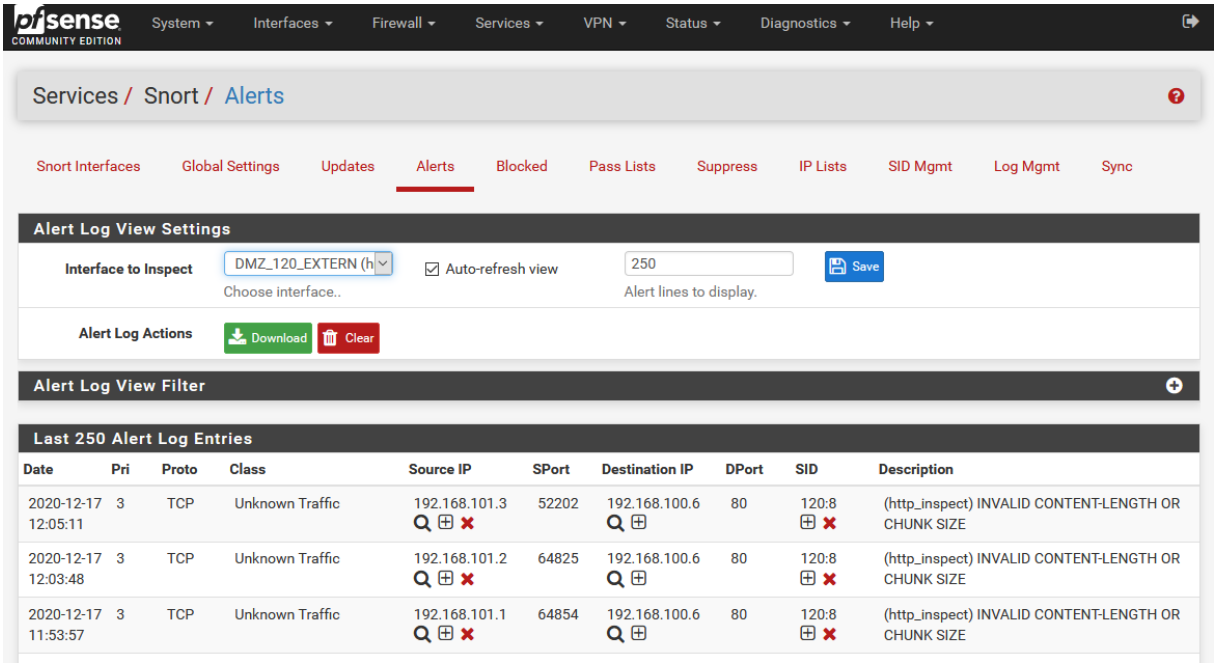
count	TotalCount	SourceIP	DestinationIP	SourcePort	DestPort	FirstSeen	LastSeen	Classification	AlertMessages	SourceName	DestinationName
2	4	167.248.133.24, 167.248.133.37	172.19.120.120	56194, 61407	443	11:43:06	11:43:06	Misc Attack	ET DROP Dshield Block Listed Source group 1	scanner-08.ch1.censys-scanner.com	HAProxy.dmz.ws.its
1	3	192.168.101.1	192.168.100.6	64854	80	11:53:58	11:53:58	Unknown Traffic	(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	WS-DC3.ws.its	WS-CA1.ws.its

Das sehe ich mir im IPS genauer an. Mein IPS ist ein Snort, dass in meiner PFSense mitläuft und den erlaubten Traffic regelbasiert analysiert und ggf. die Verbindungen blockiert. Da mein WS-DC3 in meinem Außenstandort läuft und daher

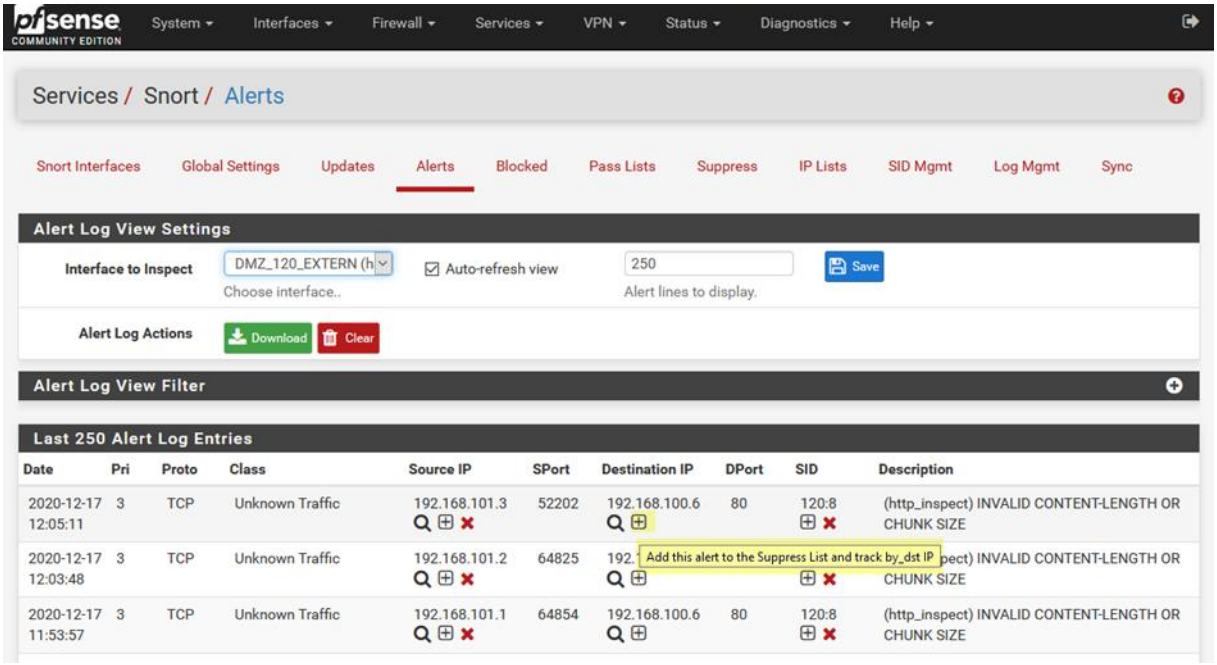
über ein VPN angebunden ist, muss die Kommunikation durch den „äußeren Filter“. Hier sind die blockierten Hosts sichtbar. Das sind alles System im Außenstandort:



In den Alerts kann ich die Ursache finden. Es war eine Kommunikation zwischen den blockierten Hosts und meiner neuen Zertifizierungsstelle. Besonders interessant ist dabei der Port 80:



Auf diesem Port läuft mein neuer OCSP. Ebenso kann hier aber auch der Sperrlisten-Download drüber laufen! Anscheinend werden die Verbindungen falsch bewertet. Daher erstelle ich eine neue Ausnahme:



Danach entferne ich noch die geblockten Hosts und die Verbindungen funktionieren wieder. An diesem Live-Beispiel kann man sehen, wie ein aktives und intelligentes Monitoring beim Lösen von Problemen helfen kann.

Zusammenfassung

Auch diese Migration verlief nicht ohne Probleme. Einige wären vermeidbar gewesen. Dennoch ist jede Problemlösung ein Lieferant für wertvolles Wissen.

Nachdem dieser Server nun mit meinem Ziel-Betriebssystem läuft, bereite ich mich auf den nächsten vor. Viele sind ja nicht mehr über…