Inhalt

Einleitung

Zielsetzung

Meine Serverumstellung auf Windows Server 2019 geht in die nächste Runde. Dieses Mal sind die beiden Server WS-RA1 und WS-RA2 dran. Beide laufen aktuell unter Windows Server 2016 als virtuelle Maschinen. Im folgenden Abschnitt prüfe ich, welche Services auf den Servern laufen und wie ich diese migrieren werde.

Bereitgestellte Services

Web Application Proxy (WAP) & Active Directory Federation Service (ADFS)

Früher nutzte ich sie für die Bereitstellung eines Web Application Proxy (**WAP**) Clusters. Mit diesem konnte ich eingehende Verbindungsanfragen von außen über HTTPS nach dem SNI auf die richtigen, internen Server aufteilen. Dies war notwendig, da ich von meinem Provider nur eine öffentliche IPv4-Adresse bekommen habe, aber mehrere Anwendungen von außen über den Port 443 erreichbar sein sollten.

Beide Server stellten das Frontend des Services bereit. Das Backend sind 2 ADFS-Services, die im Farm-Mode auf meinen Domain Controllern laufen.

Wie vor jeder Migration eines Servers überlege ich auch in diesem Fall, ob die Services so noch benötigt werden. Mittlerweile habe ich WAP durch einen HA-Proxy in meiner Firewall-Appliance unter PFSense abgelöst. Damit würde eine komplette Deinstallation von WAP genügen. Durch den Wegfall wäre dann auch die ADFS-Farm auf meinen Domain Controllern überflüssig. Das erleichtert dann später auch deren Migration.

Die Umstellung auf den HAProxy habe ich in dieses WSHowTo als eigenen Punkt integriert. Die Arbeiten dazu habe ich aber schon im Oktober ausgeführt.

Network Policy Service (NPS)

Dazu stellt der Server WS-RA1 noch einen Network Policy Service (**NPS** – auch als Radius Server bekannt) bereit. Diesen nutzt ein WLAN-Accesspoint für WPA2-Enterprise-Anmeldungen meiner Clients. Die Funktion wird weiter benötigt und muss daher auf einen neuen Server migriert werden. Dabei halte ich mir eine Erweiterung auf eine hochverfügbare Lösung offen.

Die Migration wird mittels Wipe & Load vorgenommen, da ich aktuell keine Hochverfügbarkeitsanforderung gestellt habe. Für den Wechsel ist eine Downtime erforderlich.

Diesen Teil der Migration führe ich separat aus, da der Artikel sonst zu lang wird.

VPN-Service

Die Namen der beiden Server habe ich aus dem Servicenamen RemoteAccess abgeleitet. Ich nutzte die Server als VPN-Server für die Einwahl von extern.

Die Formulierung in der Vergangenheitsform deutet es schon an: Ich nutze seit Ewigkeiten kein VPN mehr für die Arbeiten von außen. Diese Funktion bilde ich über meine Remote Desktop Services dank des RD-Gateways ab. Der Service VPN wird also nicht mehr benötigt und kann einfach entfernt werden.

Diesen Teil der Migration führe ich separat aus, das der Artikel sonst zu lang wird.

Planung der Migration

Dieser Artikel befasst sich mit der Entfernung des Web Application Proxy Cluster und der ADFS-Farm. Dazu habe ich vorher ausgeführte Konfiguration de PFSense HAProxies integriert. Die Migration des NPS wird in einem anderen Artikel beschrieben.

Damit sind die Arbeitsschritte für die komplette Migration klar:

- Schritte in diesem Artikel
    - Zuerst entferne ich alle nicht mehr benötigten Services und deren Konfigurationen in der richtigen Reihenfolge.
- Schritte im nächsten Artikel

- Danach migriere ich den Service NPS auf einen neuen Windows Server 2019 mit dem Namen WS-NPS1.
- Zuletzt entferne ich die beiden alten Server aus meiner Infrastruktur.

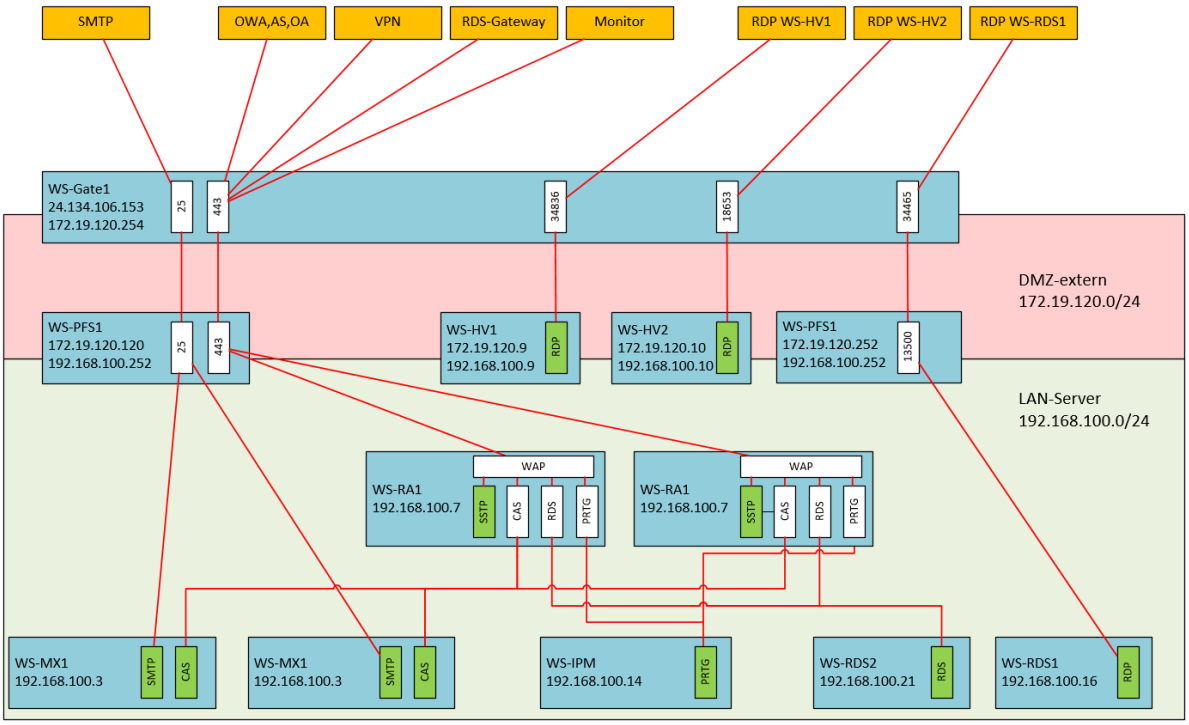
Für die Migration des NPS werde ich den neuen Server neben dem alten synchron aufbauen. Der eigentliche Austausch wird durch die Übergabe der alten IPv4-Konfiguration an den neuen Server vorgenommen. Denn nur über diese IPv4 findet der WLAN-AccessPoint den NPS-Server. Damit spare ich mir die Rekonfiguration des WLAN-AccessPoints und die Anpassung der Firewall-Ausnahmen. Und ich könnte auch schnell wieder auf den alten Server zurückschwenken, indem ich die IP-Änderung wieder zurücknehme. Ein Rollback-Szenario ist immer gut.

Umstellung von Web Application Proxy auf HAProxy (2019-10-27!)

Dieses Kapitel hatte ich bereits vor 2 Monaten geschrieben und administrativ bearbeitet. Damals hat es aber nirgends richtig reingepasst. Daher füge ich es hier an diese Stelle ein.

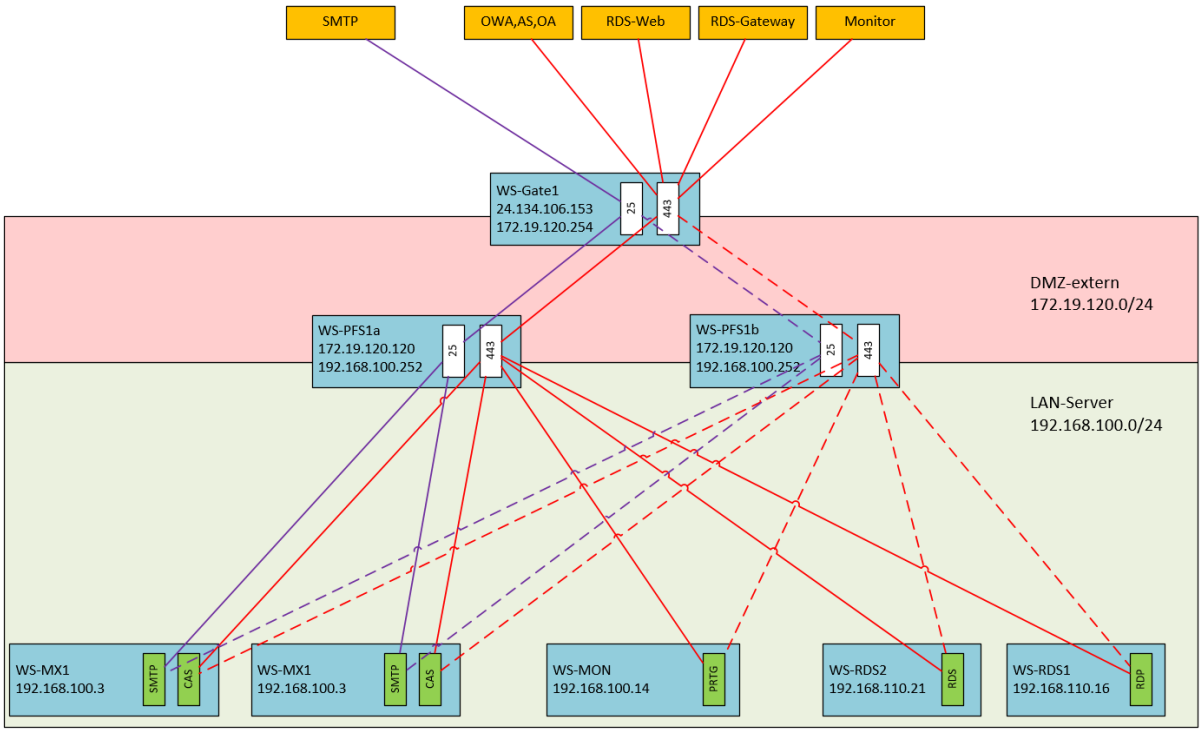
Vorgeschichte und IST-Zustand

Ich wollte meinen Web Application Proxy durch einen HAProxy ablösen. Das Konstrukt ist kompliziert und fehleranfällig geworden. Ursprünglich wollte ich einfach mehrere Webanwendungen mit https auf dem gleichen Port (443) auf der gleichen externen IPv4-Adresse veröffentlichen. Dazu nutzte ich 2 Web Application Proxy Server – beides sind virtuelle Maschinen, verteilt auf 2 Hyper-V-Hosts. Primär arbeiteten beide mit einem Windows Network Loadbalancer Feature unter einer virtuellen IP-Adresse. Für diese erstellte ich in meinen Internetrouter ein Portforwarding. Aber NLB unter Windows ist einfach schlecht. Und da kam mir eine Funktion meiner Linux Firewall gelegen: der HAProxy. Dieser kann als intelligenter Loadbalancer die eingehenden Verbindungen verteilen und Fehler ausgleichen. Das sah dann so aus (diese Zeichnung stammt aus meiner Infrastruktur-Dokumentation ☺). Man erkennt hoffentlich im oberen Bereich die verschiedenen, extern verfügbaren Anwendungen und deren Weg nach intern. Die beiden Server WS-RA1 und WS-RA2 konnten die vom Client angesprochenen Namen auswerten und danach die Verbindung an das richtige Backend leiten. Und vor diesen beiden Servern befindet sich meine PFSene WS-PFS1 und deren HAProxy:



Das geniale an dem HAProxy ist, dass er die Funktion des Web Application Proxies direkt übernehmen kann. Damit wird die Abhängigkeitskette für meine externen Anwendungen deutlich schlanker: ich benötige die beiden RemoteAccess-Server WS-RA1 und WS-RA2 nicht mehr. Und WAP benötigt im Backend ein Active Directory Federation Service. Diesen Service hatte ich ebenfalls hochverfügbar auf 2 Servern installiert. Diese kann ich damit ebenfalls verschlanken.

Das wird dann meine Infrastruktur für externe Services:

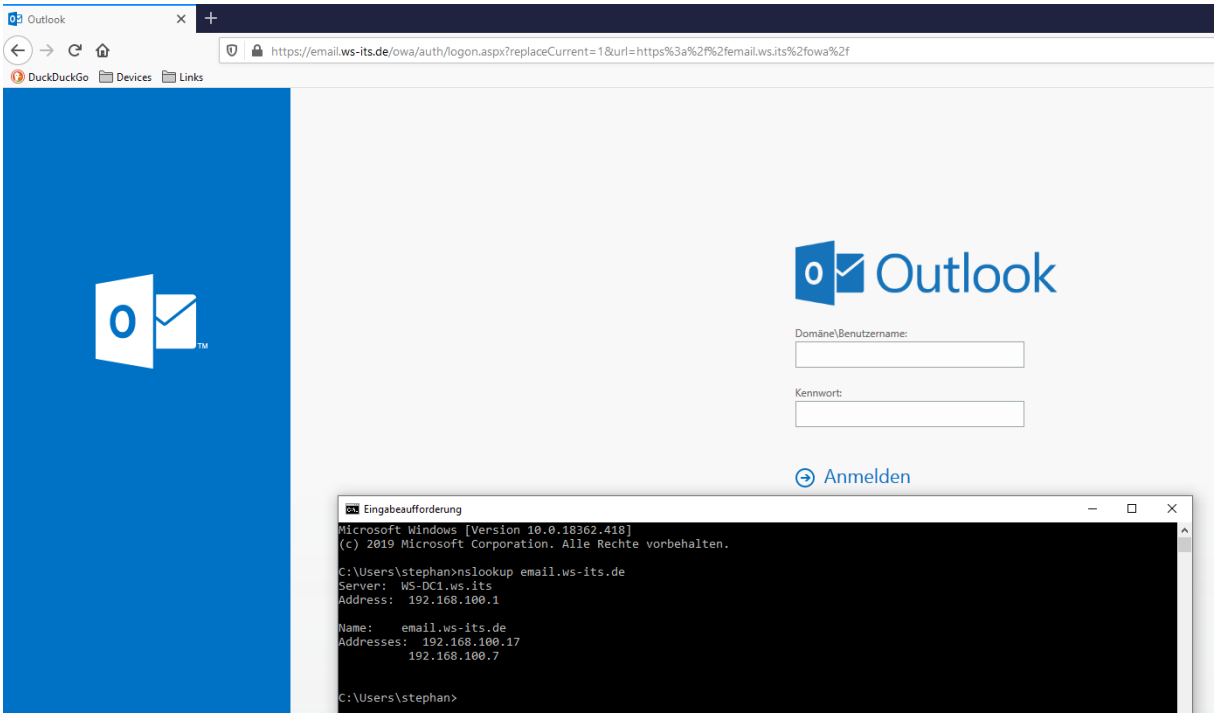


Sieht doch gleich viel einfacher aus, oder? Zwei PFSense-Systeme als virtuelle Maschinen auf unterschiedlichen Hyper-V-Hosts arbeiten als CARP-Cluster und stellen darüber einen hochverfügbaren HAProxy bereit, der vom Internetrouter weitergeleitete Pakete auf Port 443 erhält. Und diese Pakete werden nach ihrem Ziel analysiert und intern an die richtigen Systeme weitergereicht.

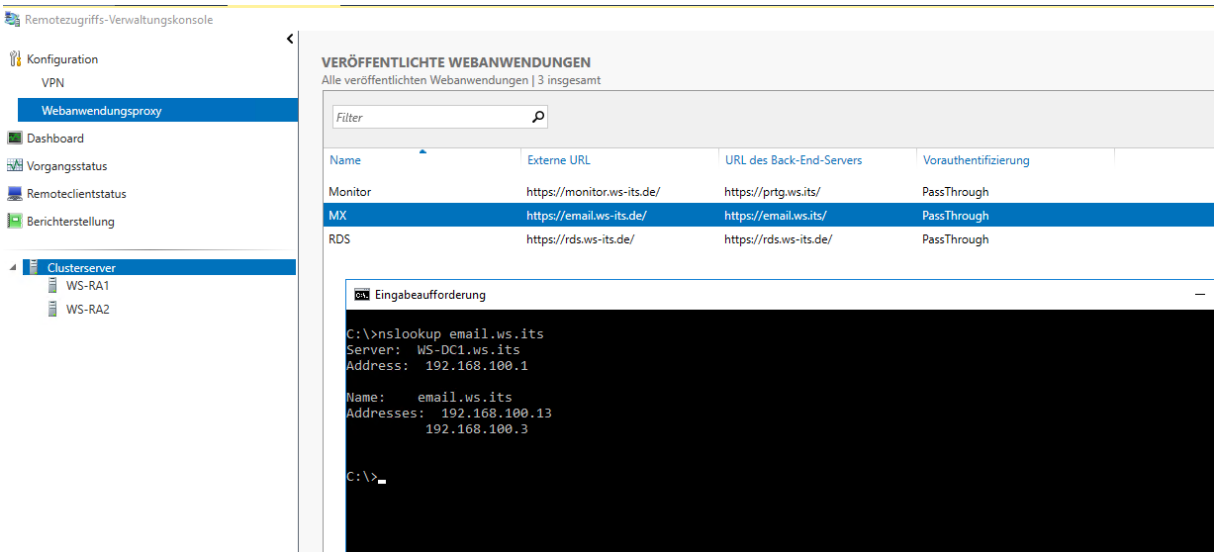
HAProxy für Exchange

IST-Zustand

Ein für mich sehr wichtiger Service ist der Zugriff auf meine Mailserver. Aktuell unterscheide ich zwei unterschiedliche Zugriffswege: den Zugriff von intern und den Zugriff von extern. Für beide verwende ich den gleichen Namespace email.ws-its.de. Meine interne Domain heißt aber ws.its. Ich muss also einen Trick anwenden. Greife ich von intern zu, dann löst mein eigener DNS-Server auf die beiden IP-Adressen der RemoteAccess-Server auf:

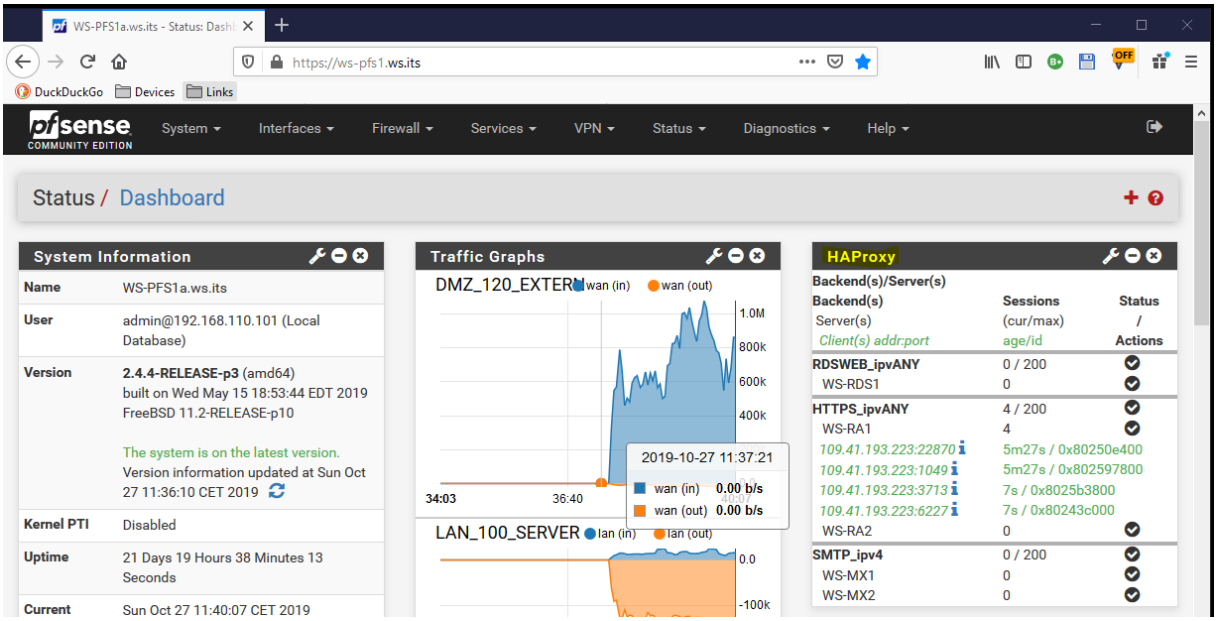


Der Benutzer kommt also erst einmal an einem der beiden Web Application Proxies raus. Die Verteilung läuft dabei über DNS-Rounrobin. Am WAP findet dann der Redirect auf den Namen der beiden Mailserver statt – ebenfalls über DNS-Roundrobin:

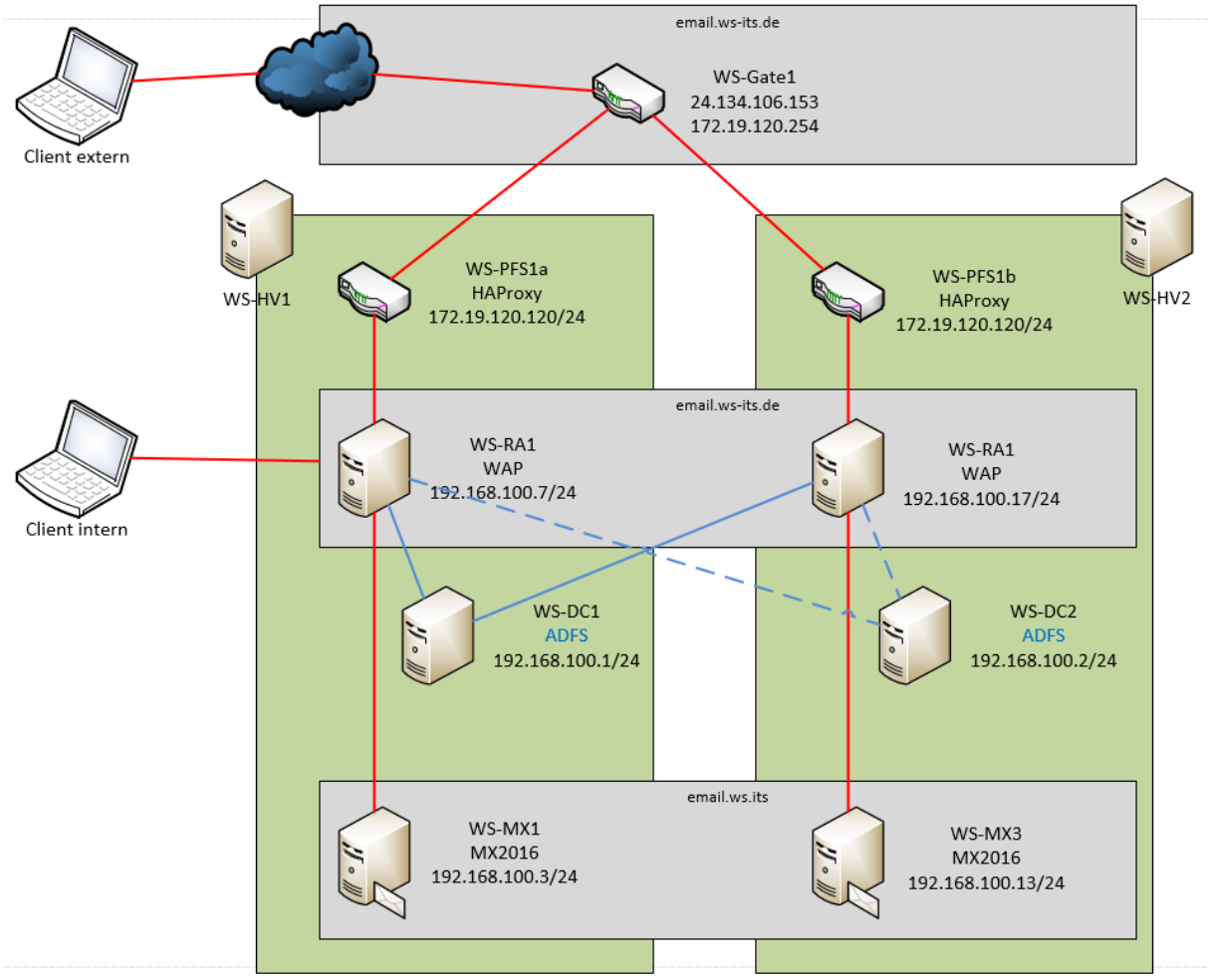


Das hat den Nachteil, dass bei einem Ausfall eines der 4 Servern (WS-MX1, WS-MX2, WS-RA1, WS-RA2) oder beim Ausfall eines darunterliegenden Hyper-V-Hosts ggf. lange Verbindungszeiten zu erwarten sind. Clients benötigen einige Zeit für den DNS-Timeout, bevor sie auf den nächsten DNS-Roundrobin-Wert springen.

Und damit nicht genug: Von extern kommt die Verbindung über meinen HAPoxy auf die WAP-Server rein. Also ein weiterer Hop bzw. eine weitere Technologie, welche die Business Continuity nicht gerade verbessern:

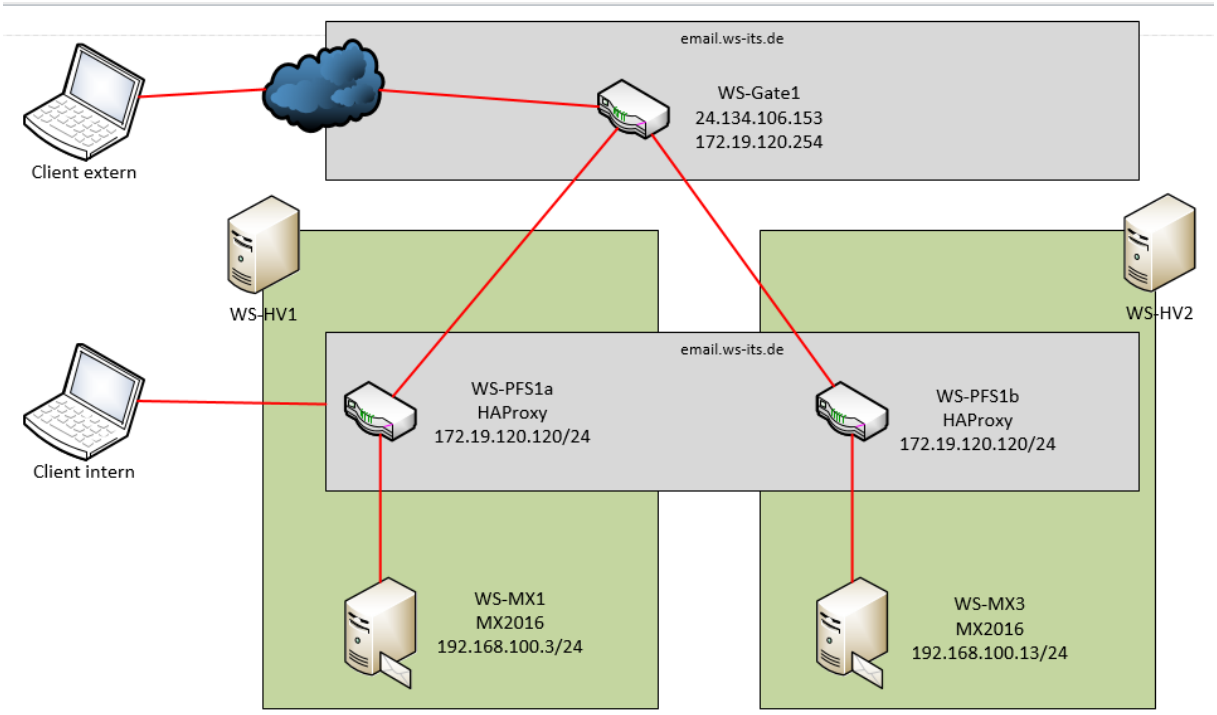


So schaut die Konstruktion schematisch aus. Und die ADFS-Server hab ich als Abhängigkeit mal mit dazu genommen:



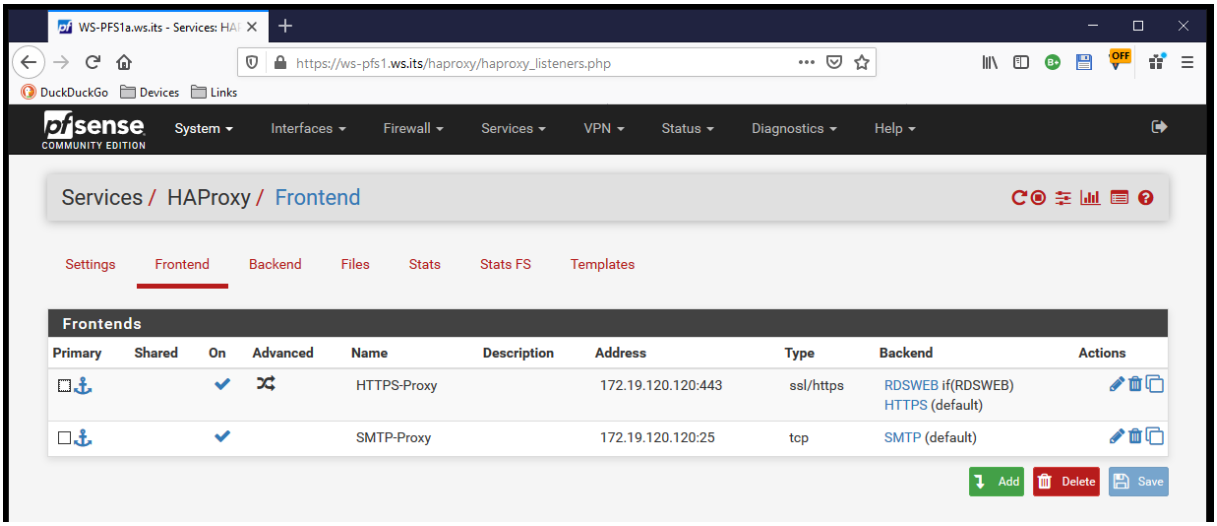
Umbau

Und so wird die Konstruktion nach dem Umbau aussehen:

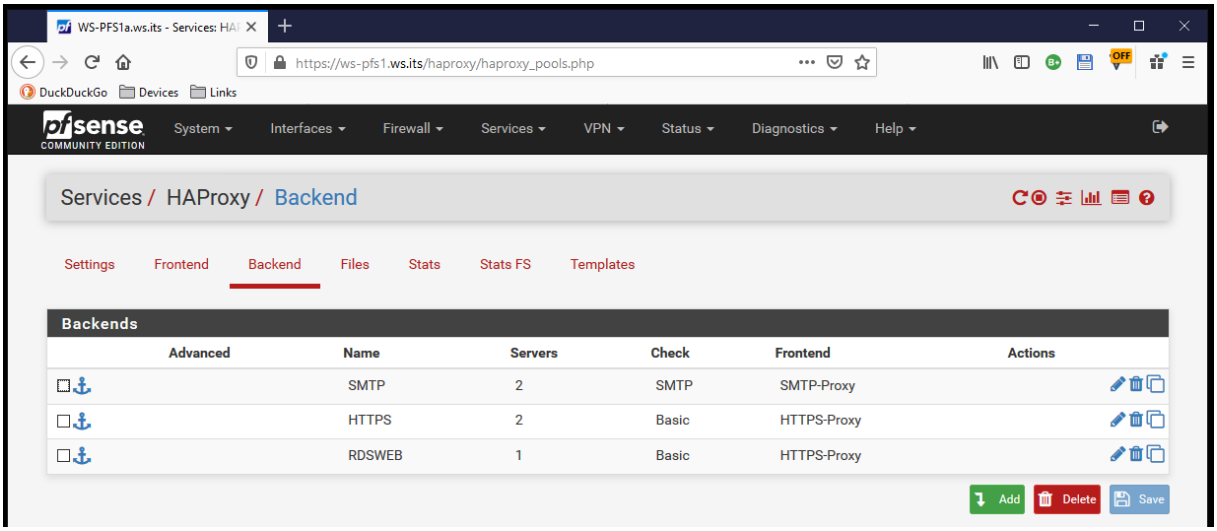


Das ist eine deutliche Vereinfachung, oder?

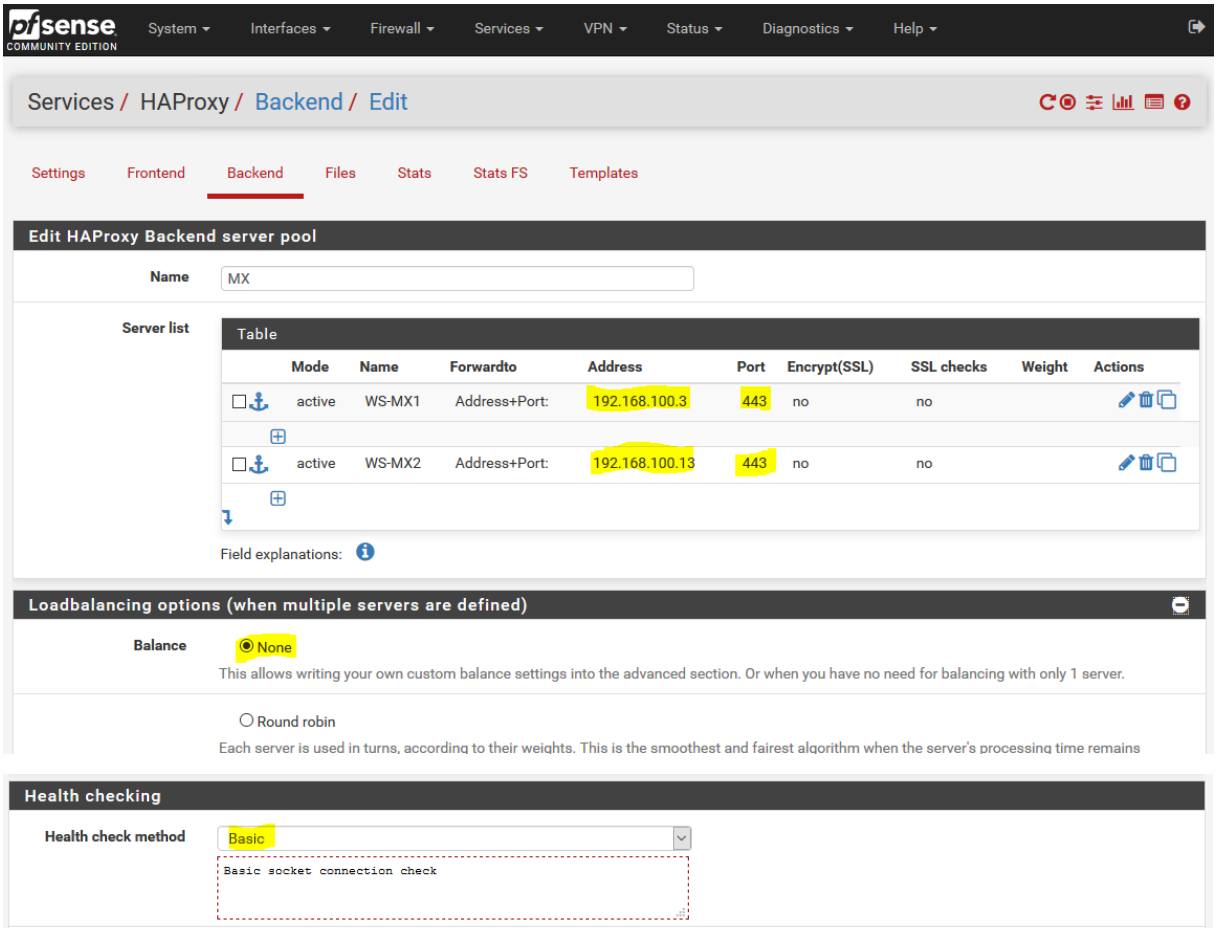
Zuerst editiere ich in meiner primären PFSense das Modul HAProxy. Von der Hauptseite mit den Frontends geht es zu den Backends:



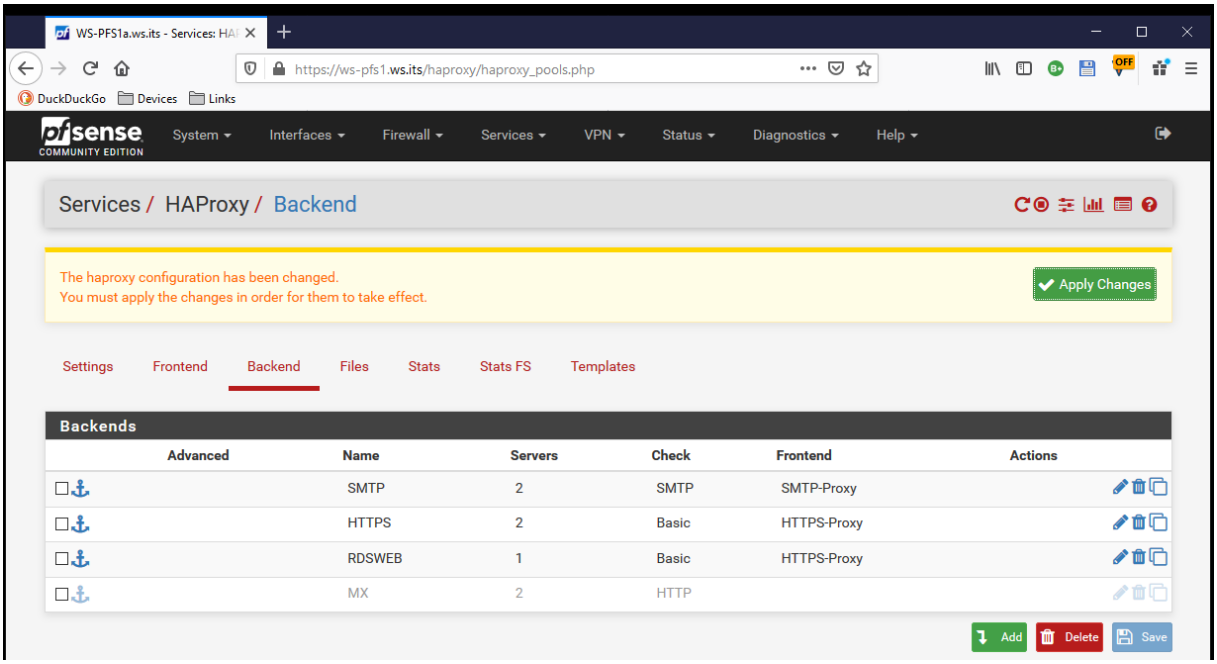
Dort erstelle ich ein neues Backend für die Kommunikation mit meinen beiden Mailservern. Über den Schalter add ist das recht einfach:



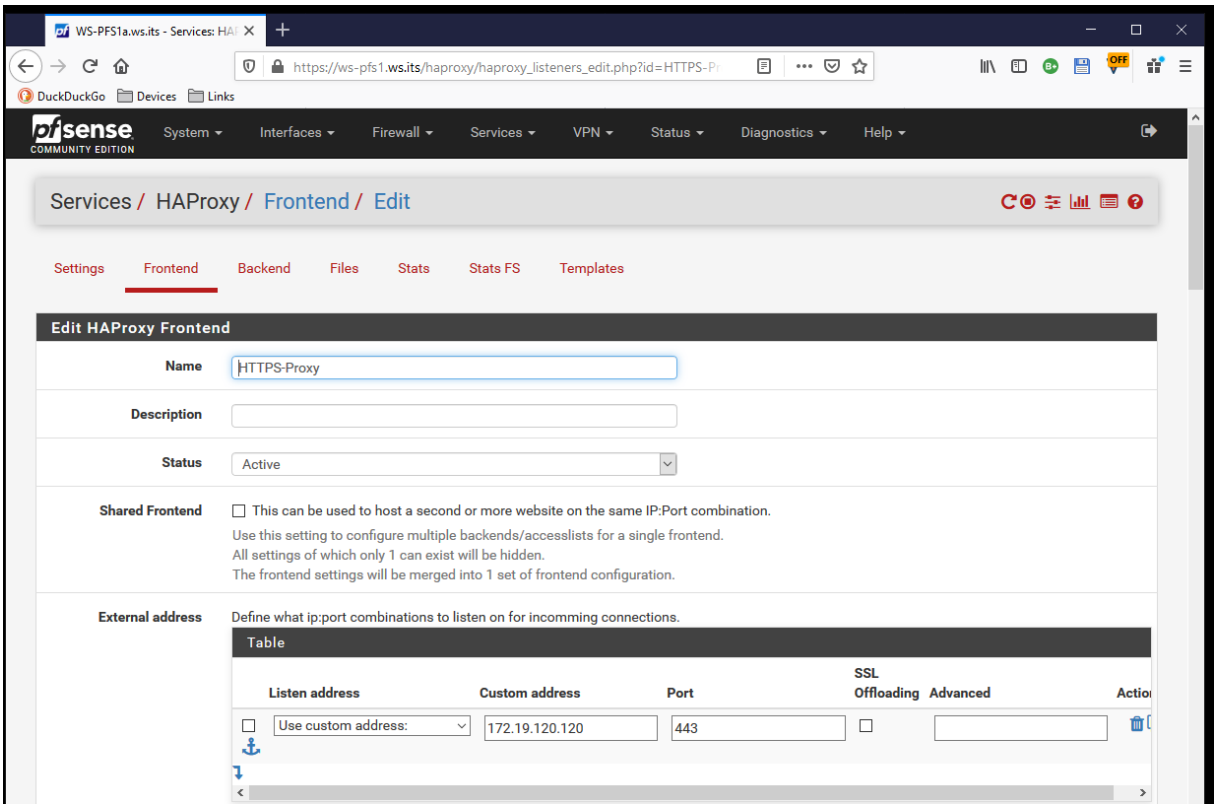
Die Mailserver werden von den Clients über https angesprochen. Daher wähle ich eine eine passende Validierung für die Verfügbarkeit aus. So kann mein HAProxy erkennen, wenn ein Exchange Server offline geht und die Clients auf den anderen Server umleiten:



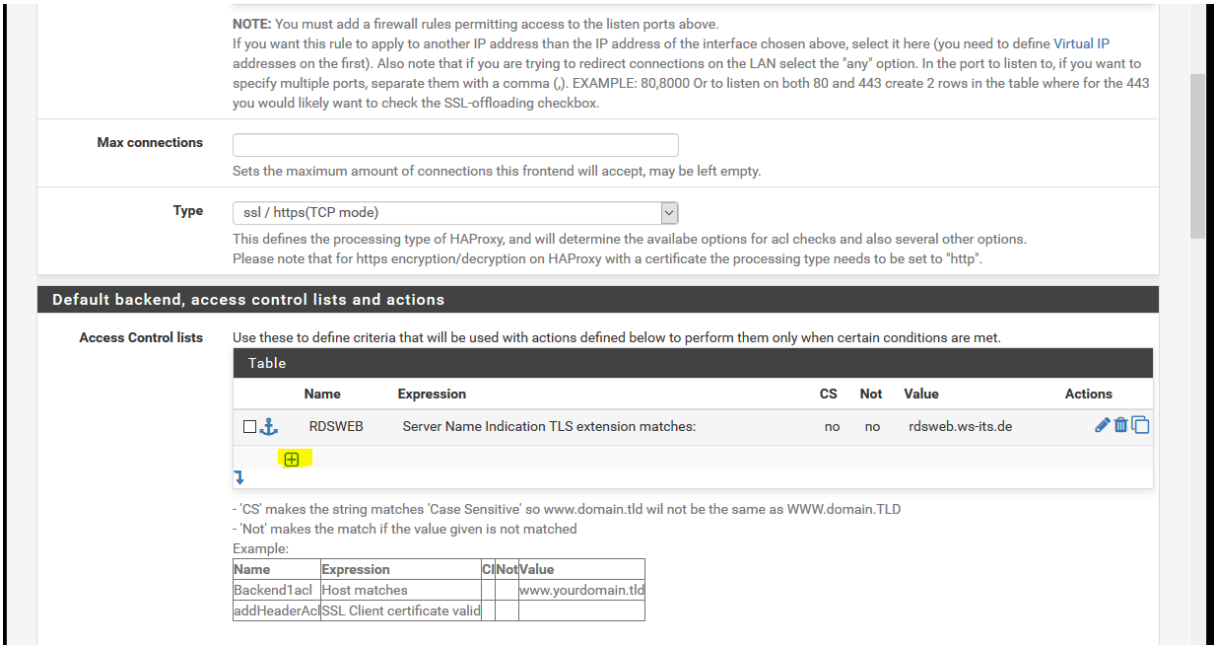
Das Backend ist fertig, wird aber von keinem Frontend verwendet:



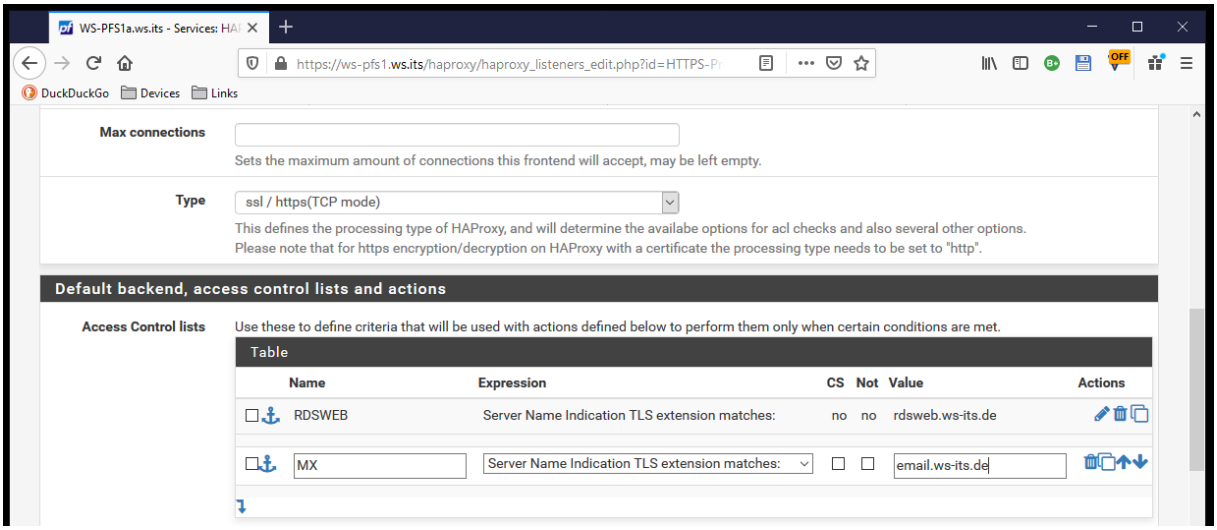
Das Frontend existiert ja schon. In meinem Fall reagiert der HAProxy auf eingehende Verbindungen auf der IPv4-Adresse 172.19.120.120 und dem Port 443:



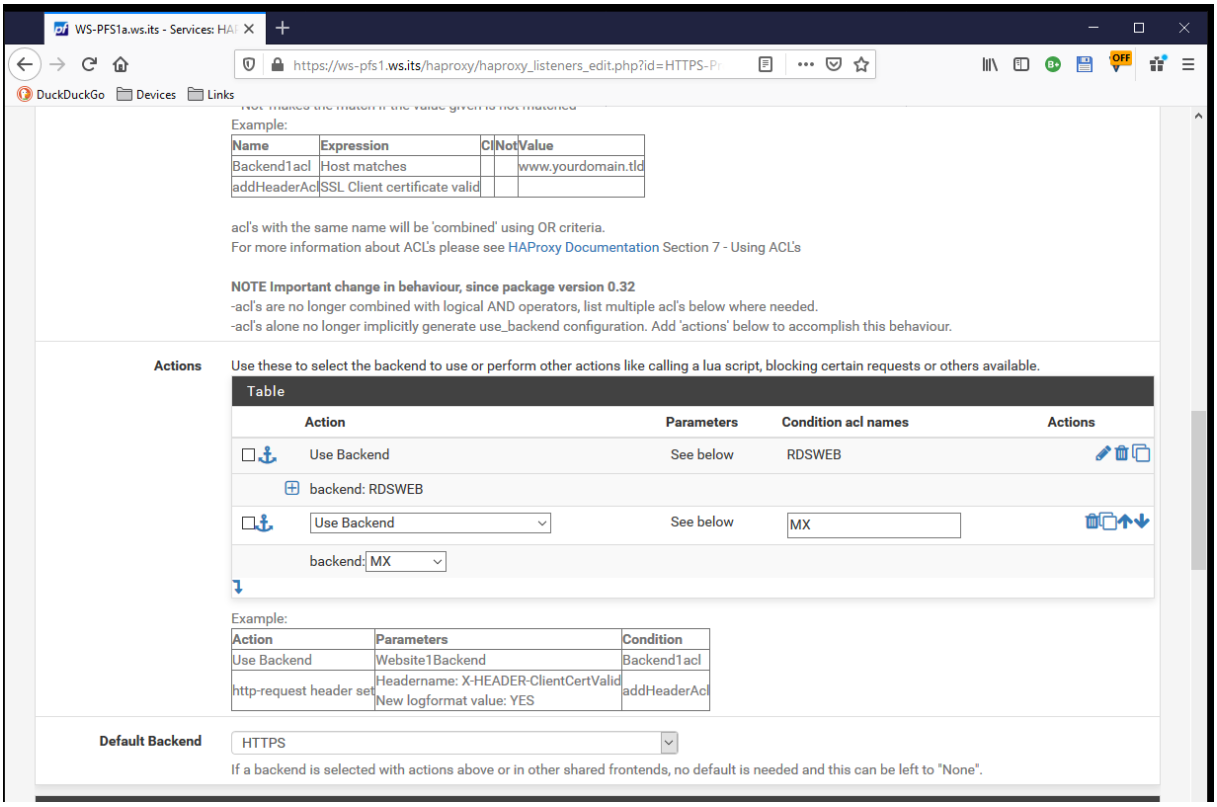
Aber nun muss er noch eine Differenzierung zur Backend-Weiterleitung erhalten. Dazu nutze ich den SNI (Server Name Indikation) – also den FQDN, den ein Client anspricht. Meine Smartphones und Outlooks verwenden den Namen email.ws-its.de. Erkennt der HAProxy diesen SNI, dann soll er an das neue Backend weiterleiten. Gesteuert wird das Verhalten im Frontend in so genannten Access Control Lists. Hier füge ich eine neue hinzu:



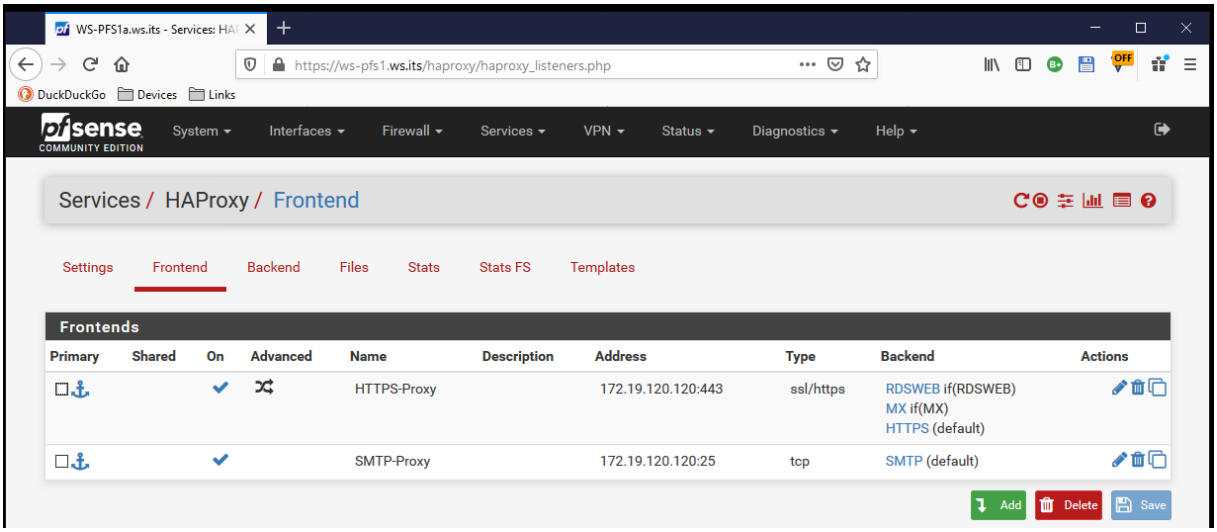
Die neue ACL bekommt einen passenden Namen bzw. ein Kürzel und natürlich die Regel für die Bedingung:



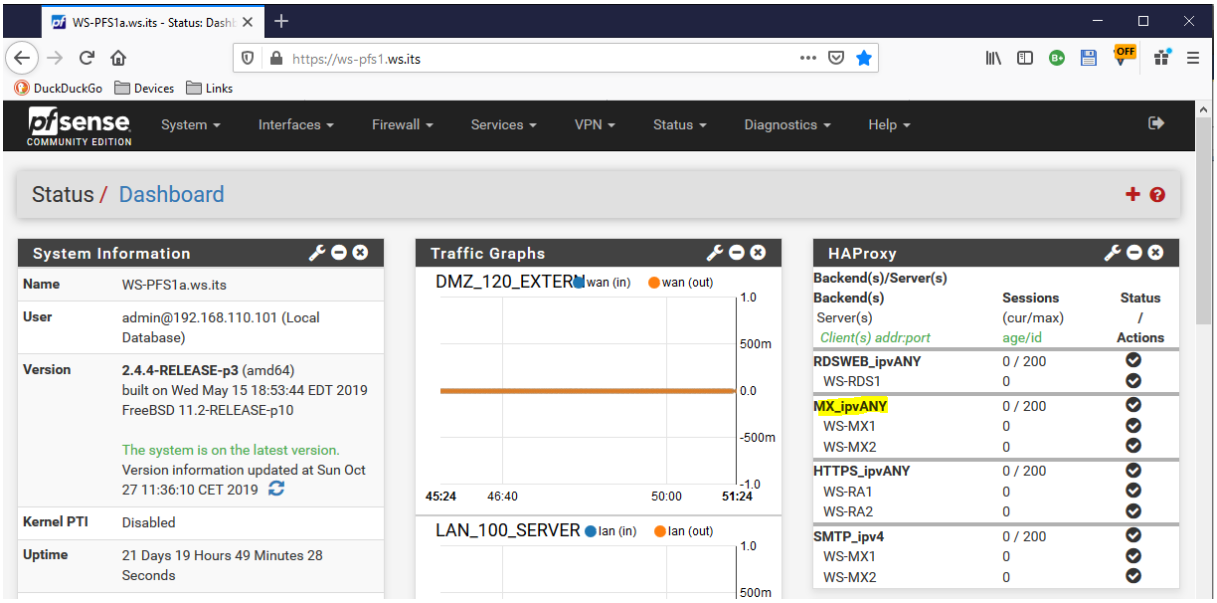
Die ACL ist aber nur ein Bestandteil. Zusätzlich muss weiter unten im Frontend noch die action für eine positive Bedingungsprüfung definiert werden. In meinem Fall soll das Backend „MX“ angesprochen werden:



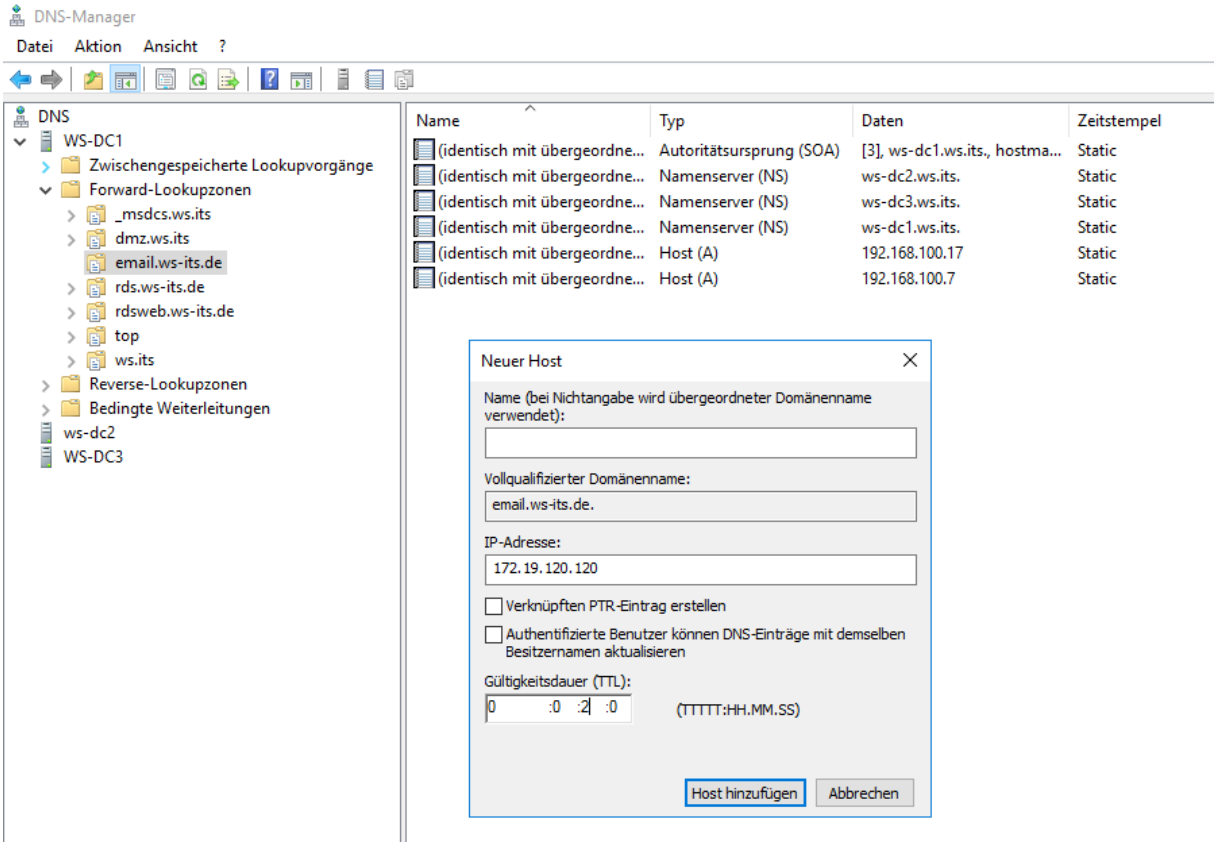
Ein Apply später ist diese Regel aktiv:



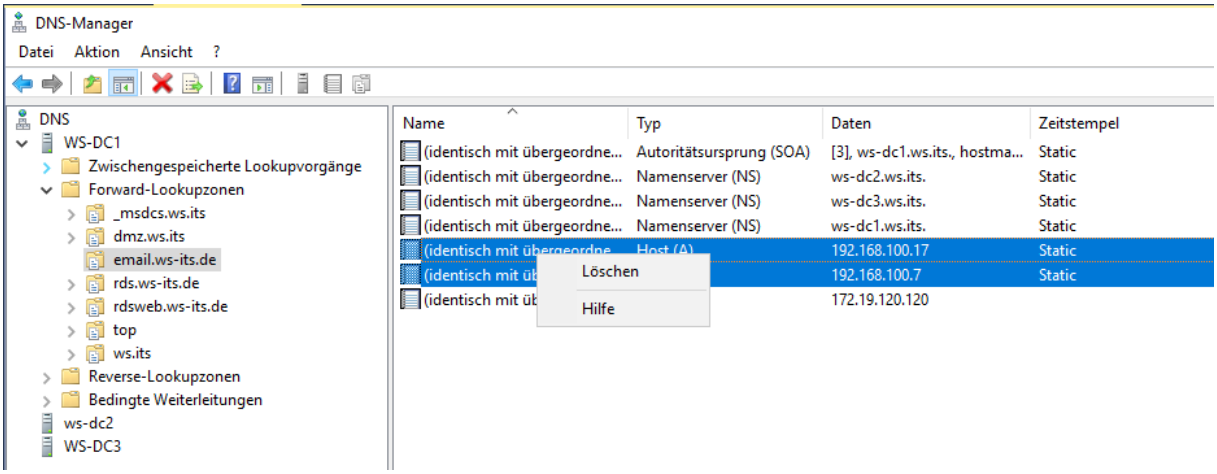
Auf dem Dashboard meiner PFSense sieht man den neuen Eintrag. Bisher ging der Traffic durch das https_ipvany-Frontend an die beiden WAP-Server WS-RA1 und WS-RA2. Ab jetzt wird vorher direkt auf die Exchange Server umgeleitet:



Damit funktioniert der externe Aufruf ohne weitere Konfiguration. Intern haben meine Clients aber die Exchange Server über den Namespace email.ws-its.de angesprochen. Im internen DNS hatte ich dazu eine Zone erstellt und direkt auf beide WAP-Server verwiesen. Jetzt kommen die Clients direkt zum HAProxy. Also erstelle ich einen neuen Record. Wichtig dabei: ich gebe keinen Namen an. Damit ist der Record direkt für die Zone gültig – also für email.ws-its.de:

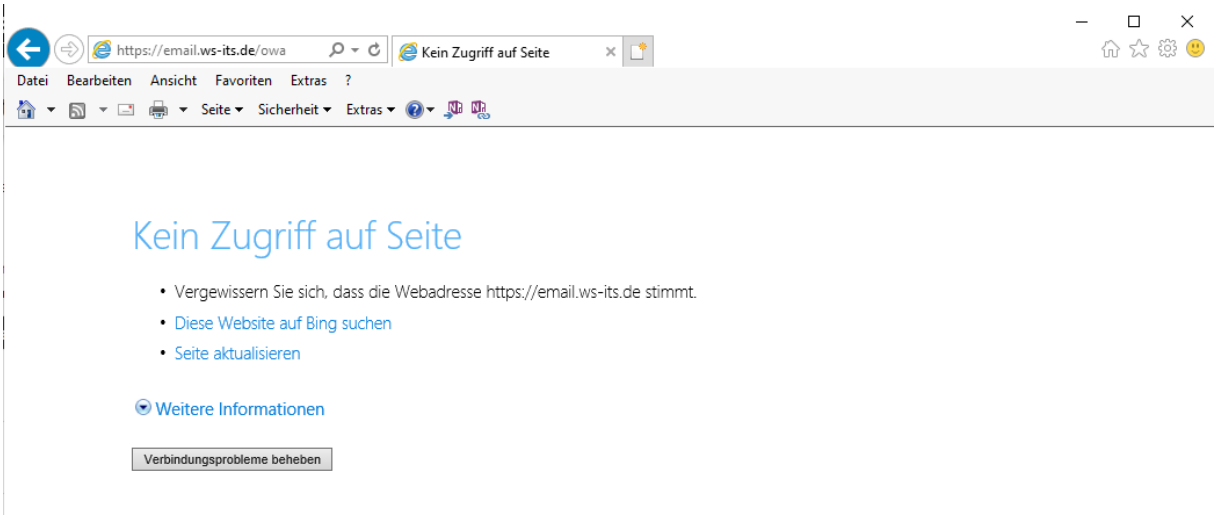


Danach kann ich die beiden alten Records zu den WAP-Servern löschen:

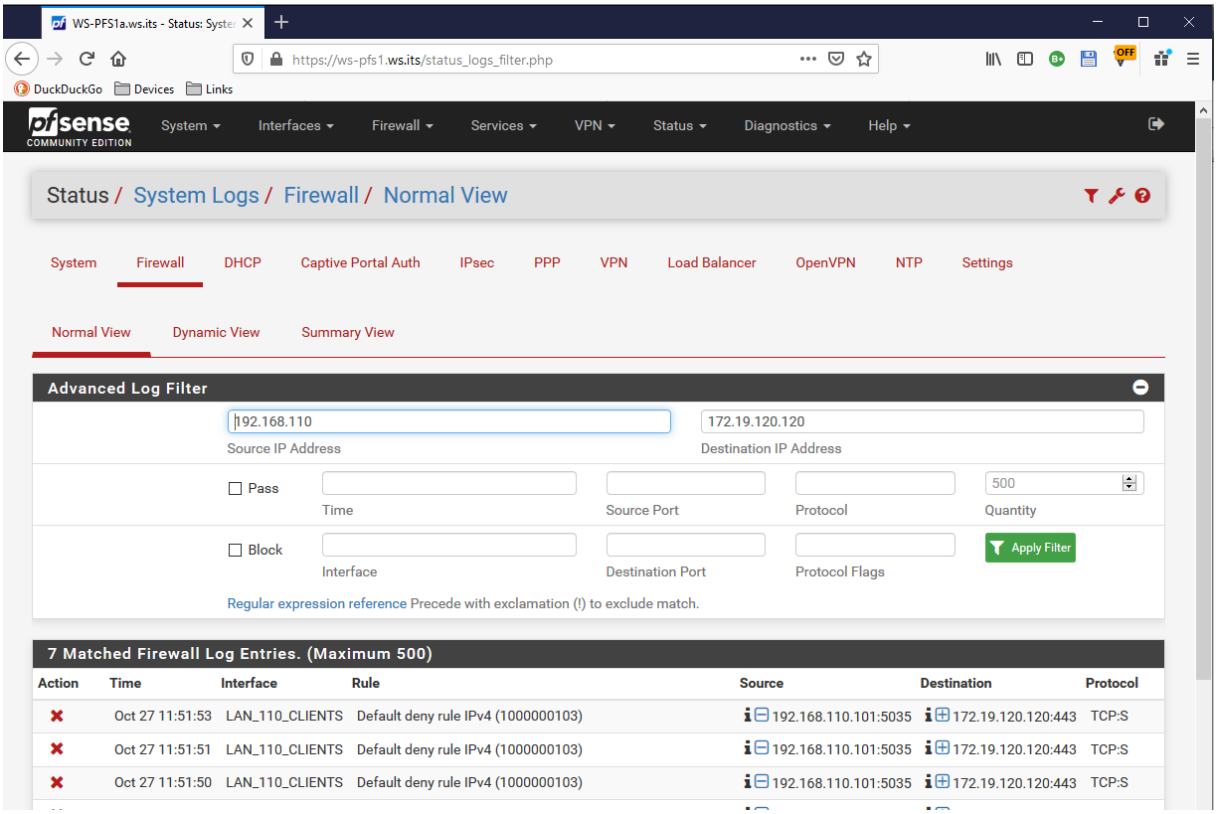


Die Clients lernen bei mir recht schnell diese Änderung, da ich alle DNS-Record mit 2 Minuten TimeToLive erstellt habe:

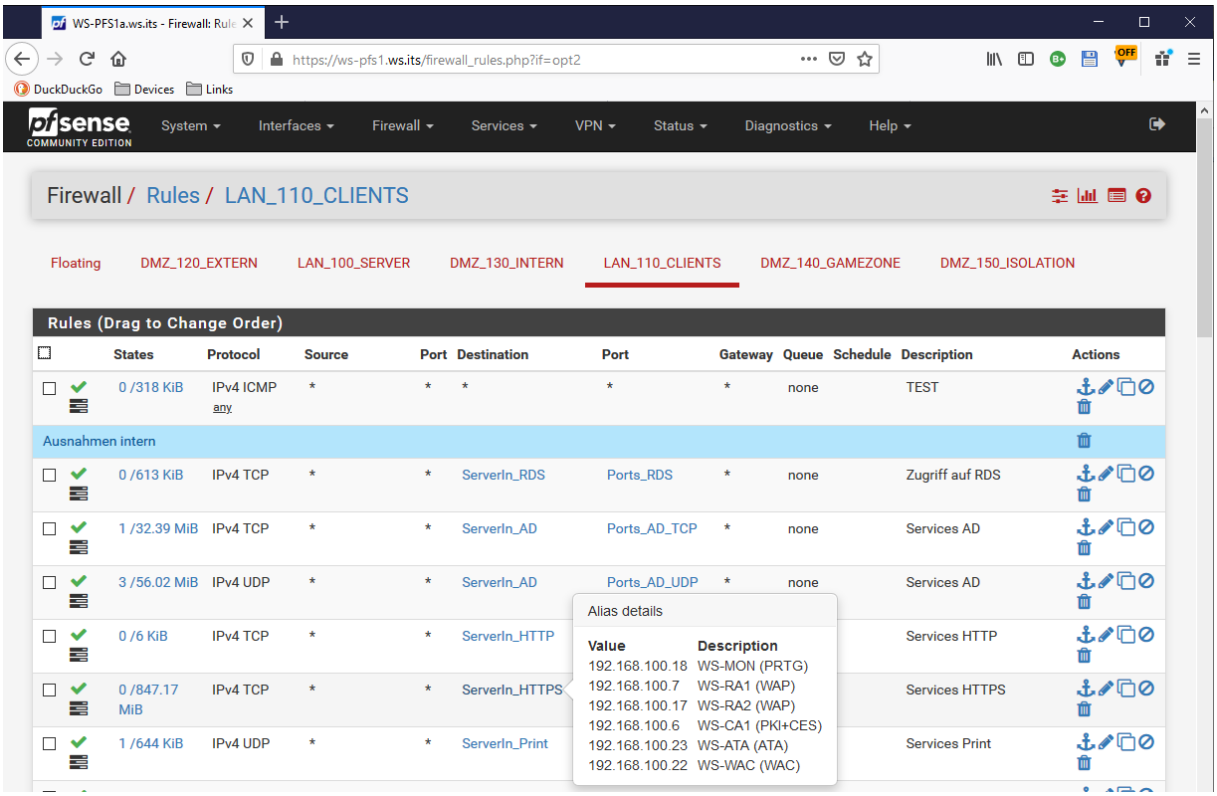
Also wird es Zeit für einen Test. Ich öffne einen Browser auf einem internen Client und navigiere zum OWA-Portal des Exchange Servers. Aber das scheint noch nicht durchzugehen:



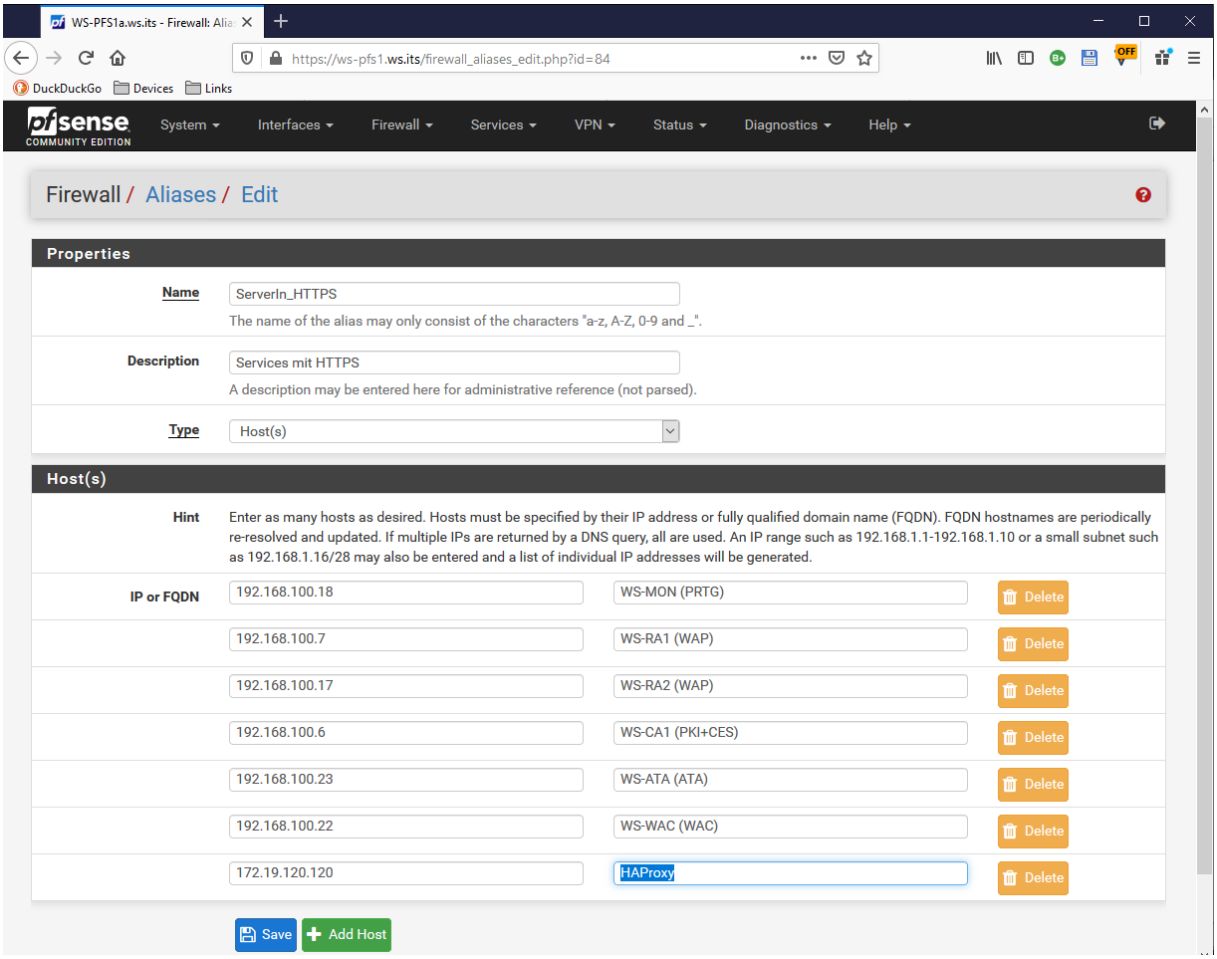
Die Ursache des Problems ist schnell gefunden: Meine Clients haben nicht das Recht, den HAProxy von intern anzusprechen. Das verhindert die Firewall der PFSense. Bisher war das ja auch nicht erforderlich. Im Firewall-Log sieht man sehr schön die Blocks:



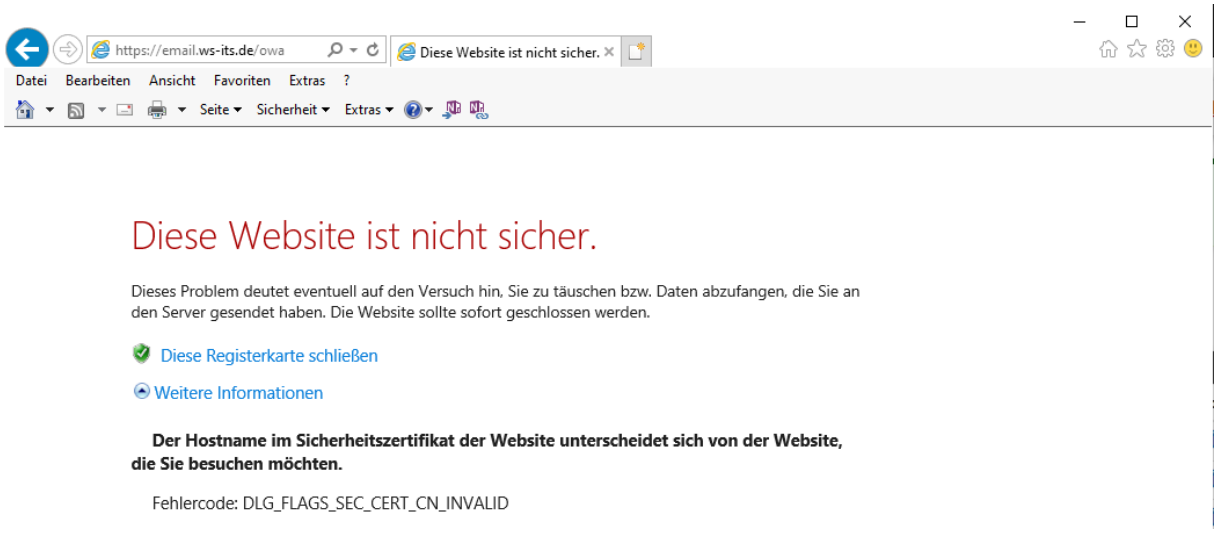
Meine Regeln habe ich durch Alias-Definitionen etwas strukturiert. So kann ich sehr bequem die Erweiterung vornehmen. Meine Clients dürfen HTTPS nur zu folgenden internen Servern verwenden. Natürlich stehen hier die beiden alten WAP-Server drin:



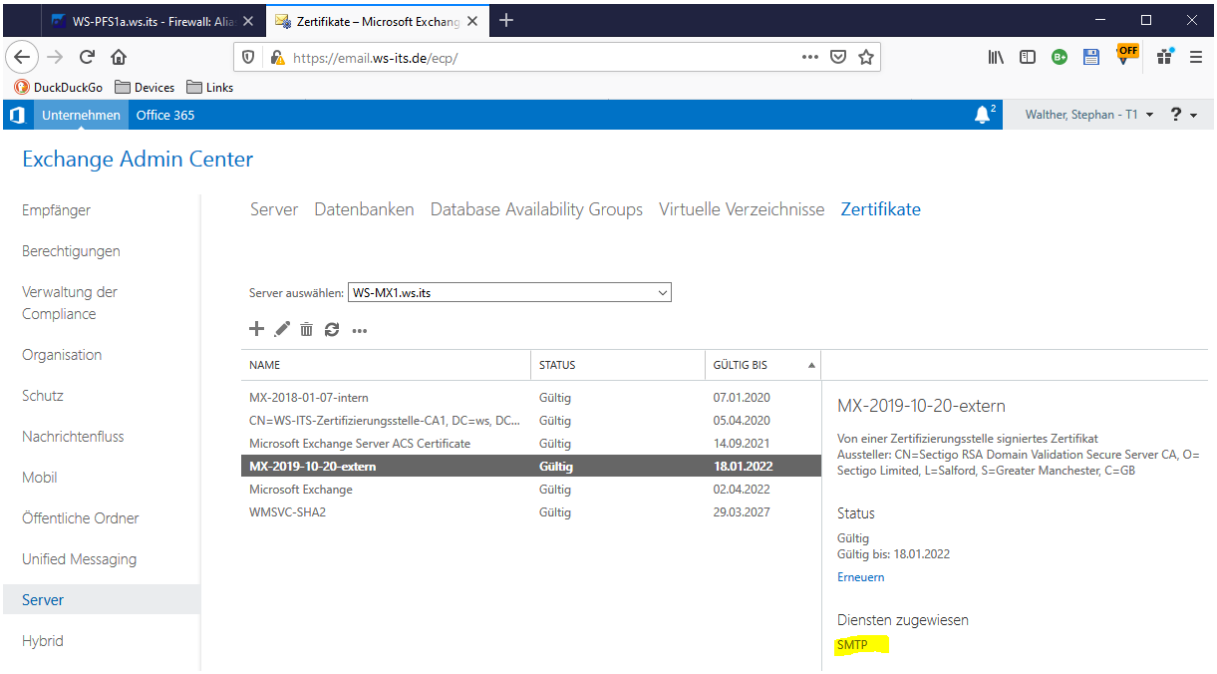
Ein Klick auf den Hyperlink des Alias bringt mich zur Konfiguration. Ich nehme die IPv4 des HAProxy mit auf:



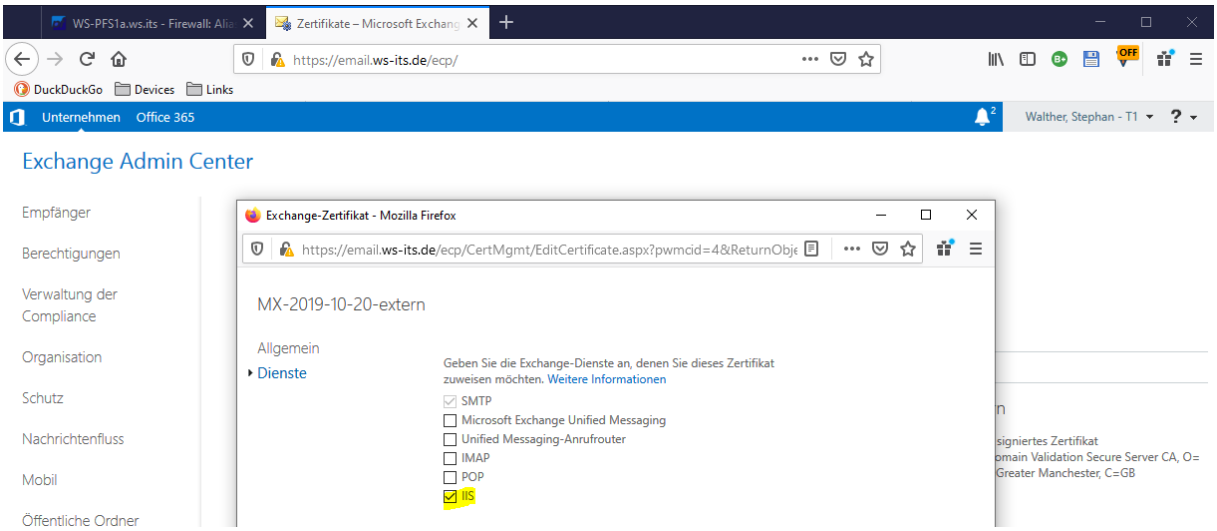
Die beiden WAP-Server belasse ich noch, da es noch weitere Anwendungen gibt, die ich vorab umstellen muss. Die Firewall-Ausnahme greift sofort. Mein Browser kann eine Verbindung aufbauen. Aber die Fehlermeldung zeigt ein weiteres Problem:



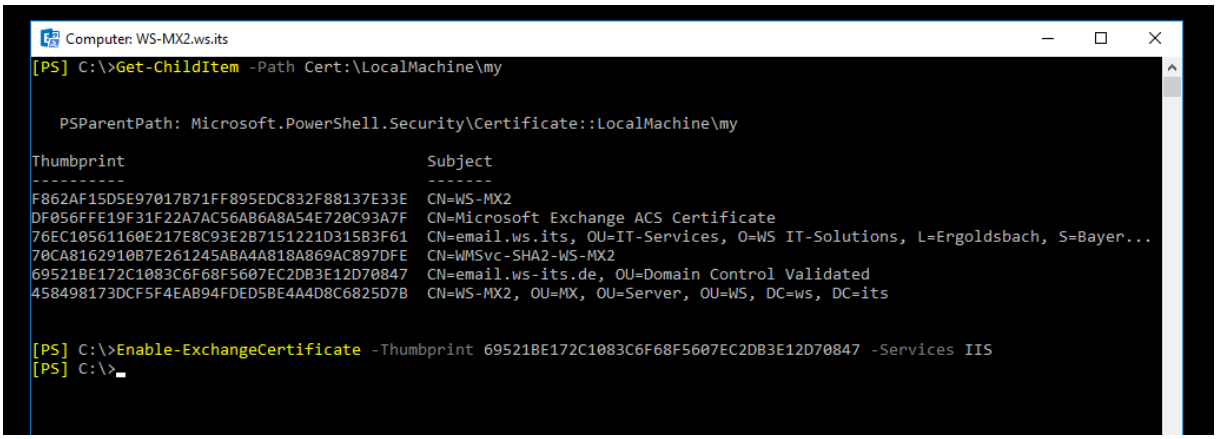
Bisher war mein öffentliches Zertifikat für email.ws-its.de auf den WAP-Servern installiert. Die Mailserver hatten nur ein internes Zertifikat. Dessen Name passt natürlich nicht mehr. Also editiere ich noch die Zertifikatverwendung auf beiden Mailservern. Das öffentliche Zertifikat hatte ich bereits für SMTP-TLS installiert. Ich muss also nur noch die IIS-Bindung nachtragen:



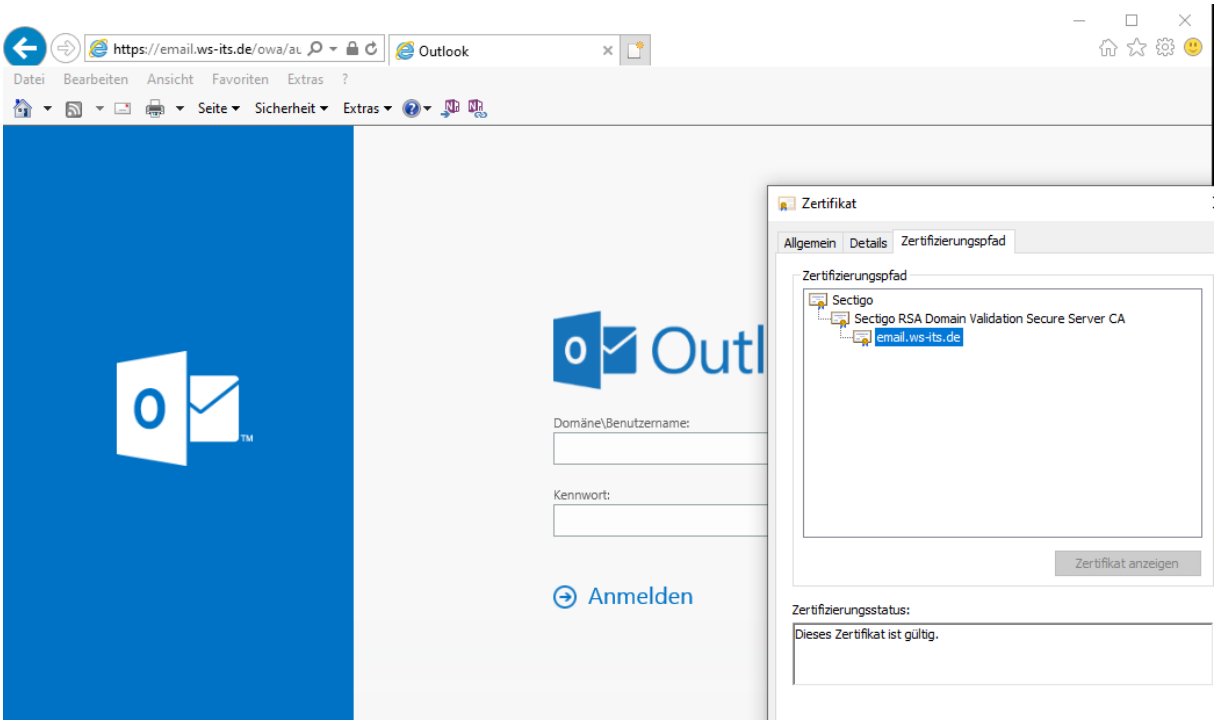
Das geht sehr einfach:



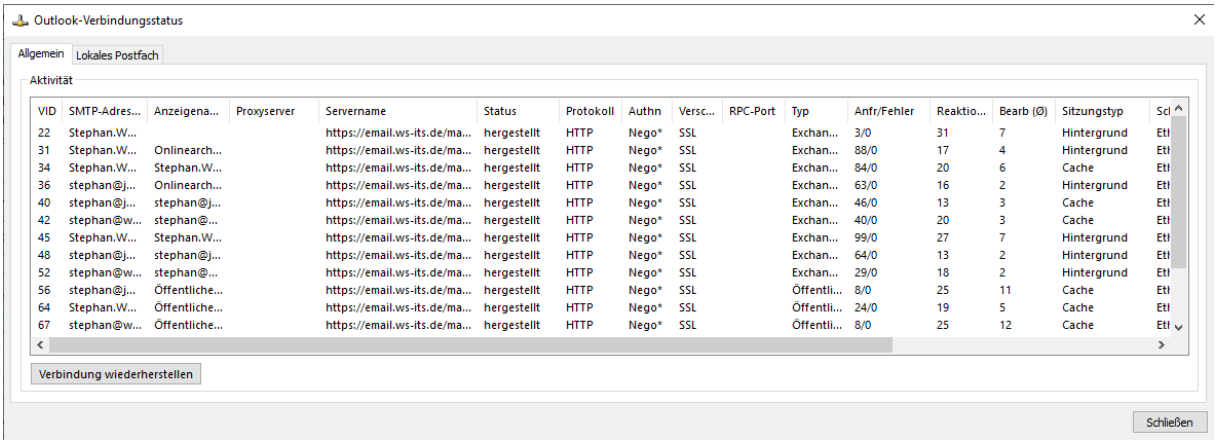
Alternativ kann das auch mit der PowerShell erledigt werden. Den zweiten Server stelle ich mit diesen 2 Zeilen um:



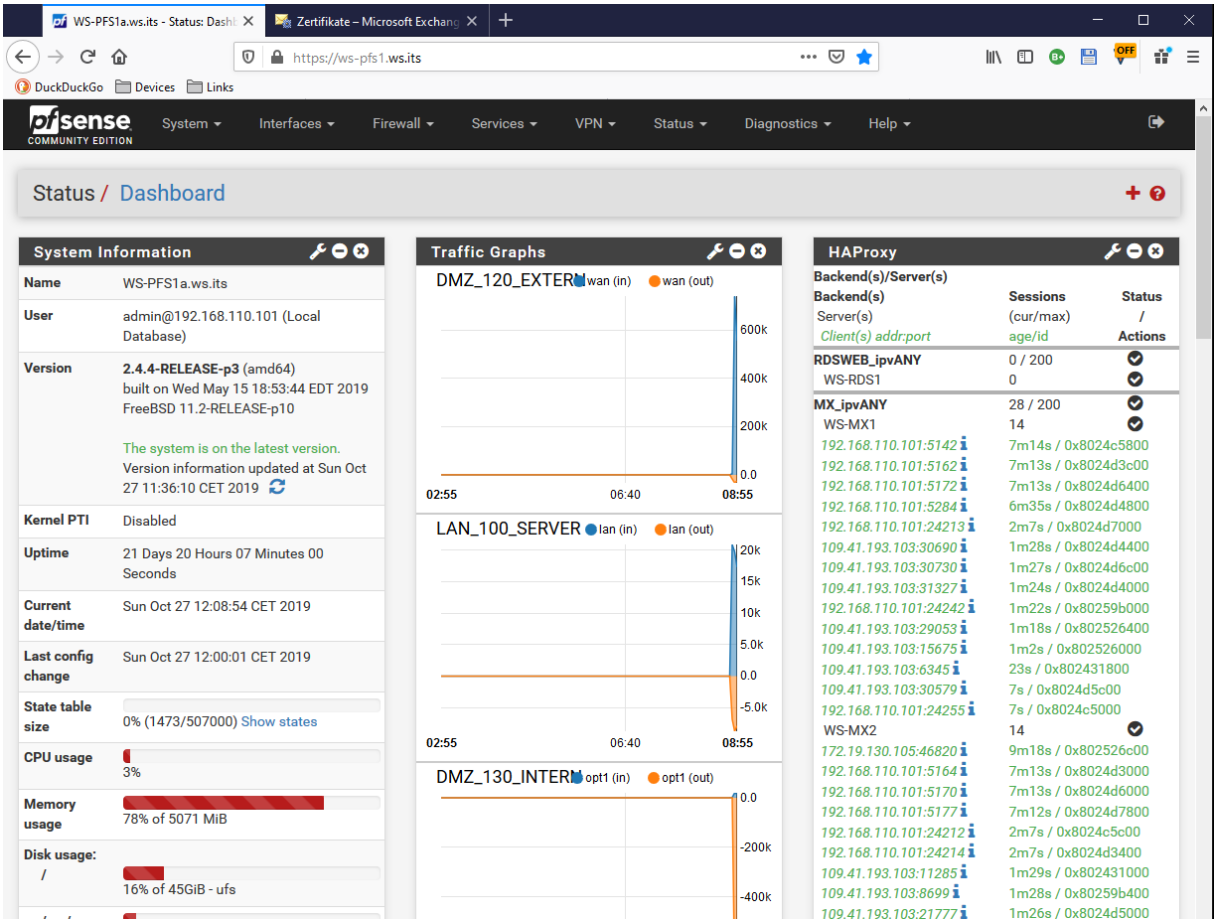
Ein weiterer OWA-Test wird nun zu einem der Mailserver durchgereicht. Der Client möchte mit https://email.ws-its.de sprechen. Und beide Server präsentieren dafür das richtige Zertifikat:



Natürlich geht es einem internen Outlook gleich. In der Verbindungsanzeige von Outlook kann man schön die Servernamen erkennen. Und natürlich die erfolgreiche Verbindung über den HAProxy zum Mailserver:



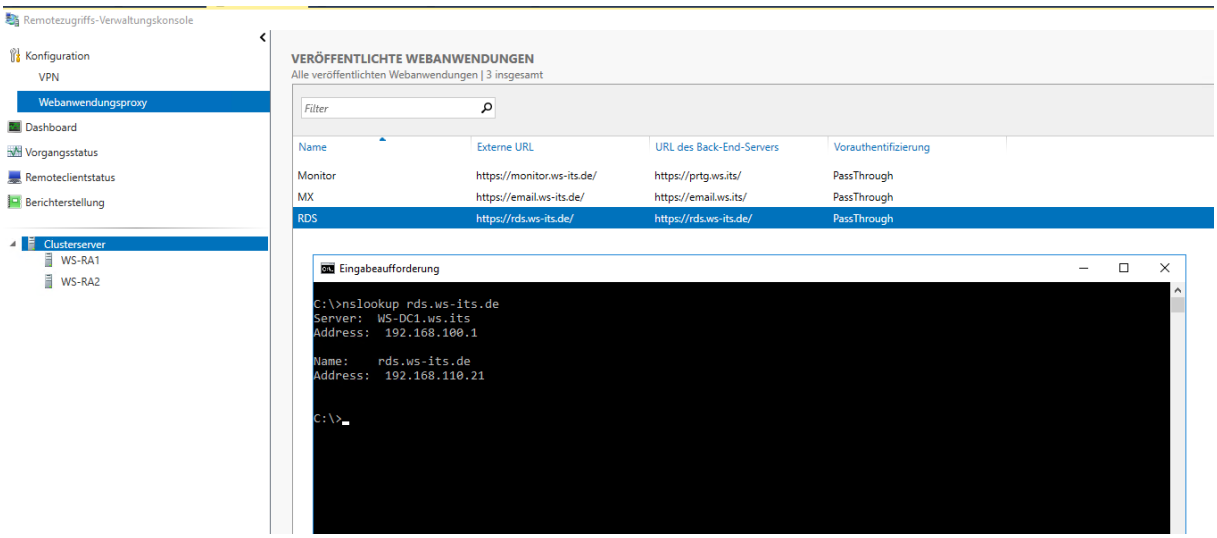
Doch mit welchem Server reden meine Clients aktuell? Der HAProxy zeigt die Verbindungen im Dashboard der PFSense an:



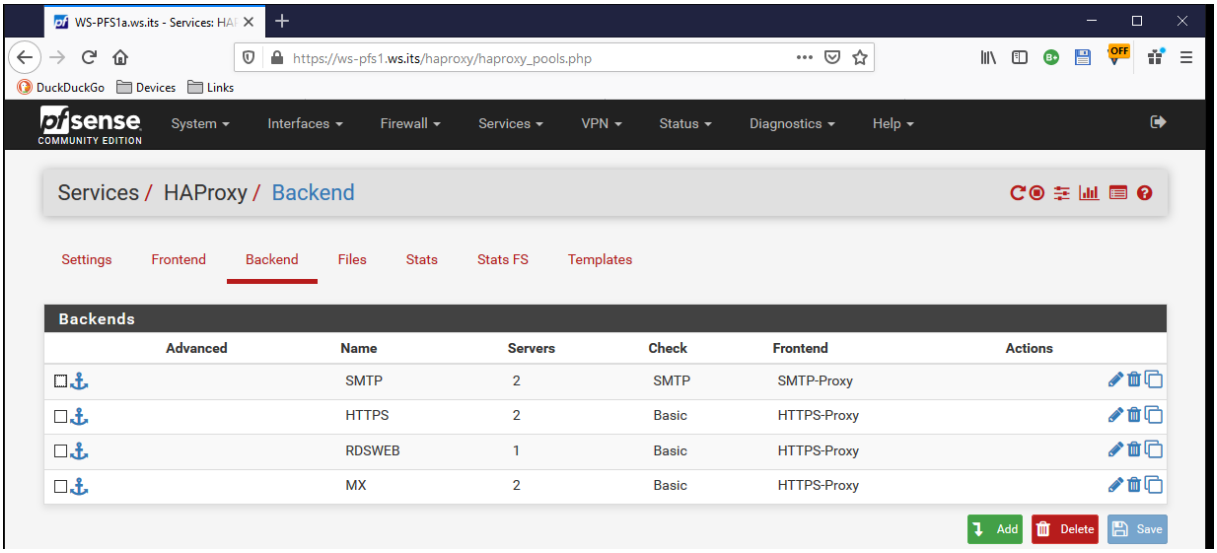
Damit benötige ich WAP nicht mehr für den Zugriff auf meine Exchange Server.

HAProxy für RDS

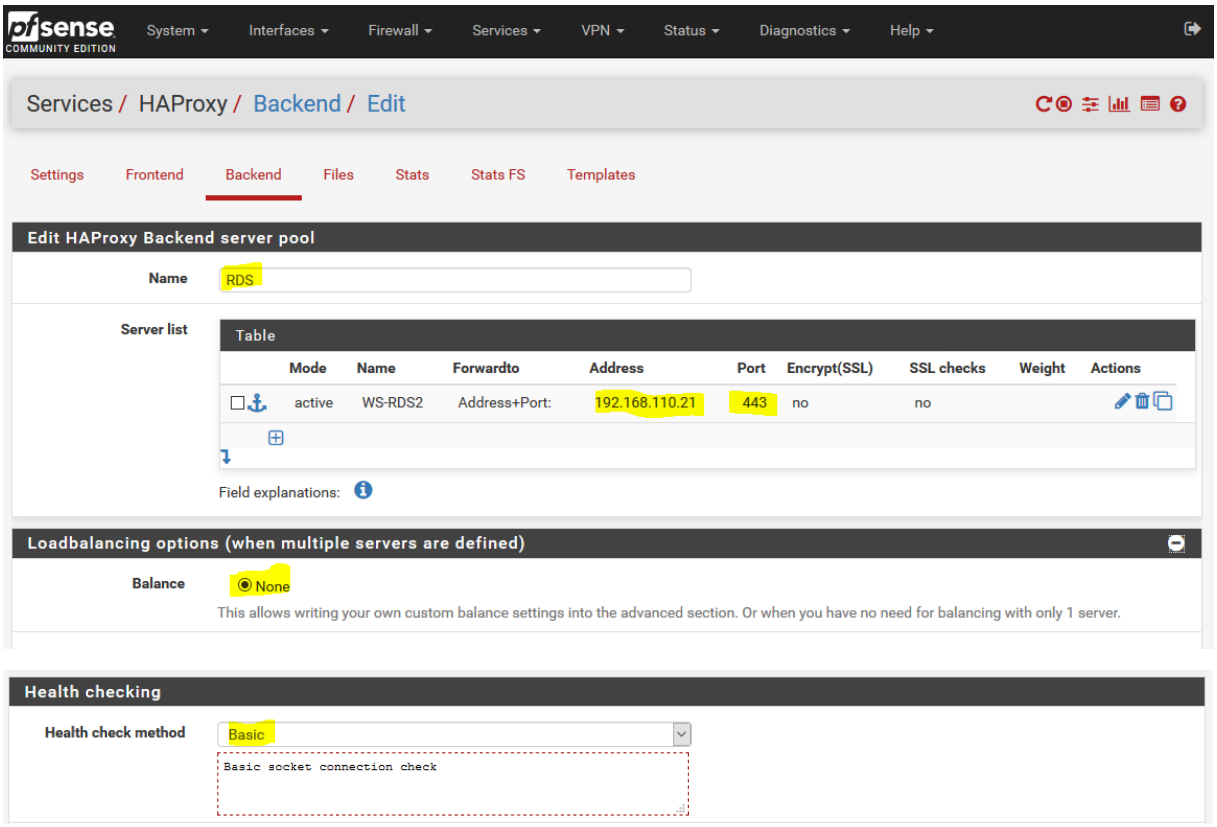
Die WAP-Server leiten externe Anfragen auf den SNI rds.ws-its.de auf meinen RD-Gateway weiter. Dieser verwendet mit dem gleichen DNS-Trick wie beim Exchange Server intern den gleichen Namen:

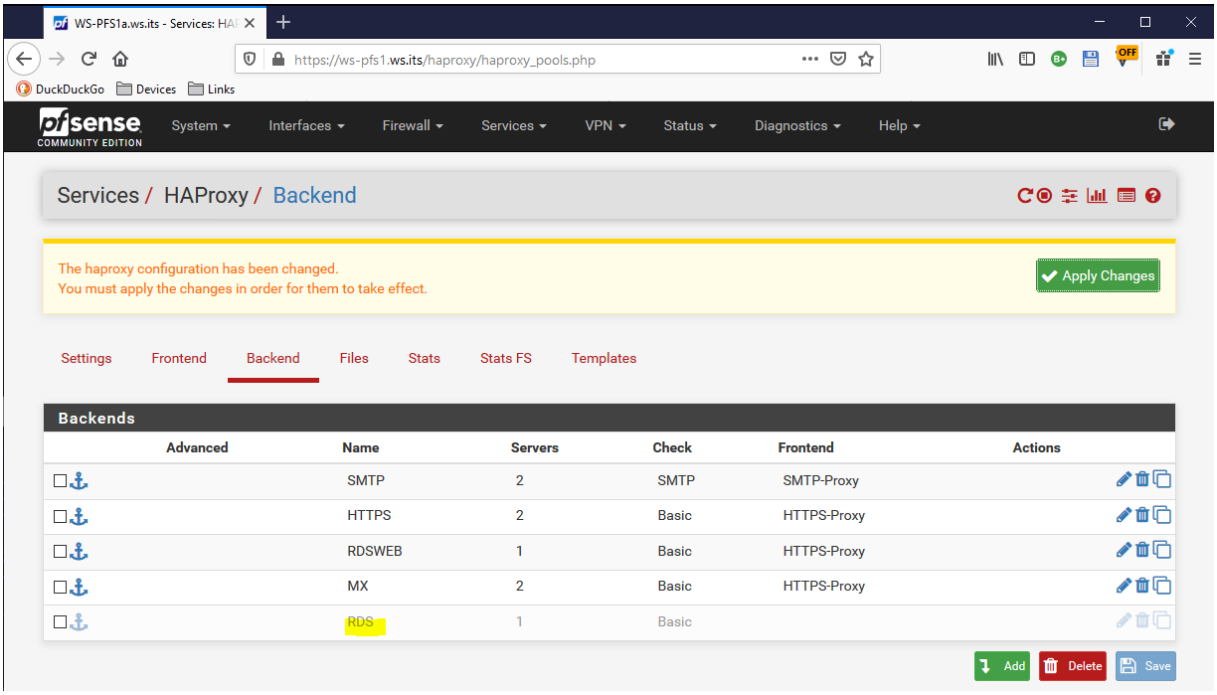


Damit ist die Rekonfiguration fast identisch zu der vom Exchange Service. Im HAProxy erstelle ich zuerst ein passendes Backend, dass auf meinen RD-Gateway verweist. Dieses ist nicht zu verwechseln mit dem Backend rdsweb. Dieses leitet zu einem anderen RDS-Server mit dem HTML5-Client um:

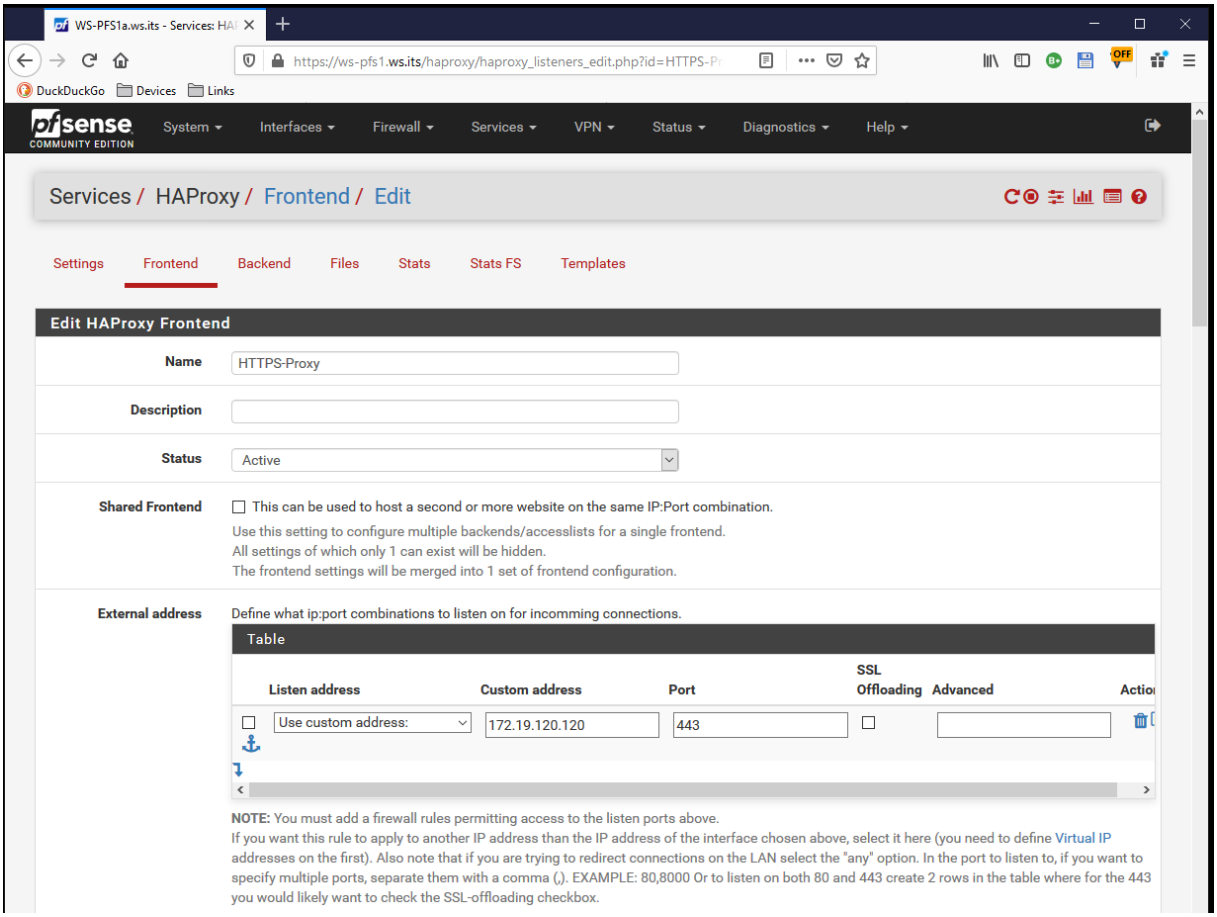


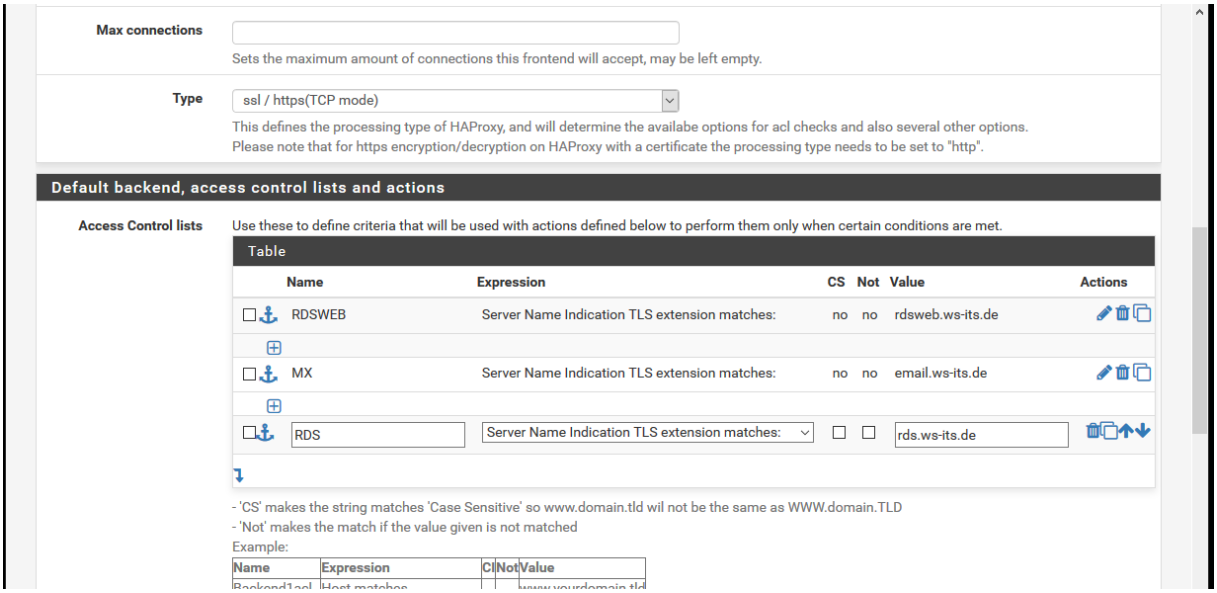
Das neue Backend hat nur ein einziges Ziel. RDS ist damit nicht hochverfügbar:



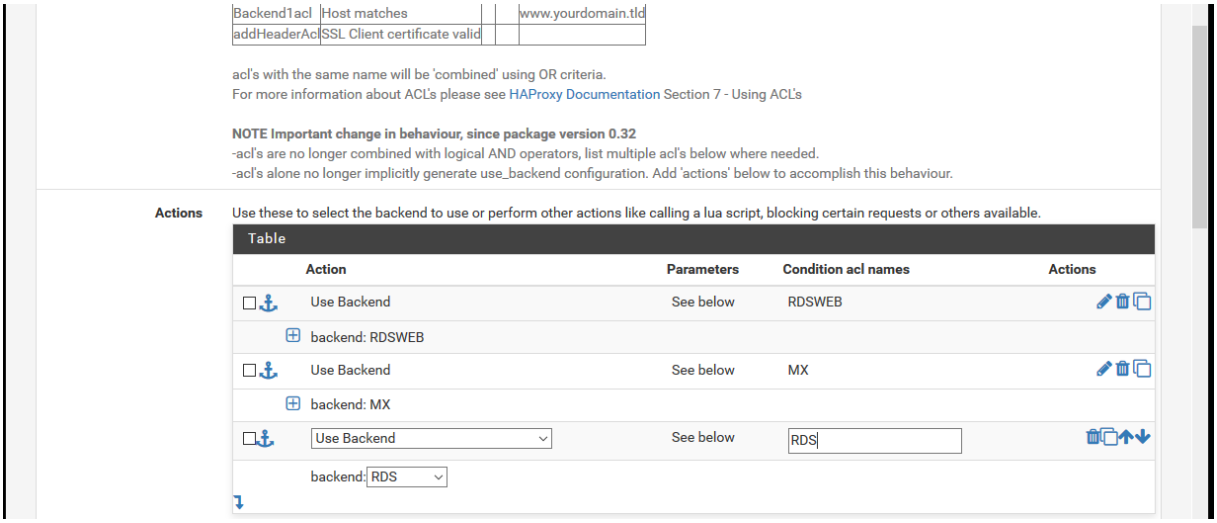


Im HTTPS-Frontend erstelle ich wieder eine ACL mit dem SNI-Filter für die neue Anwendung:

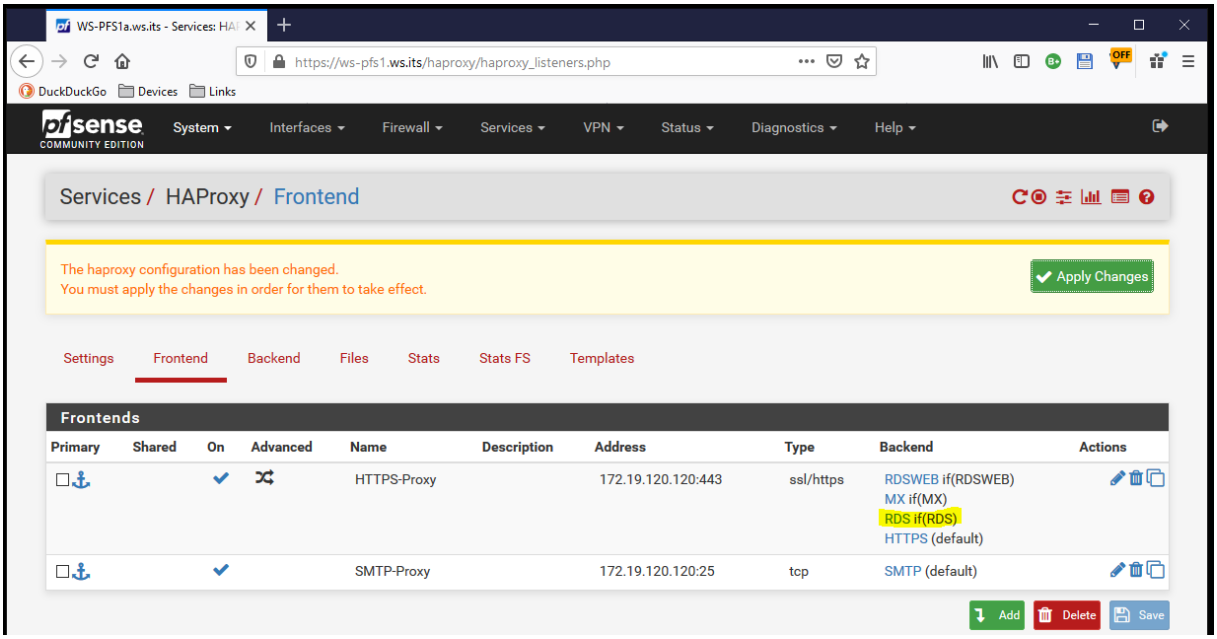




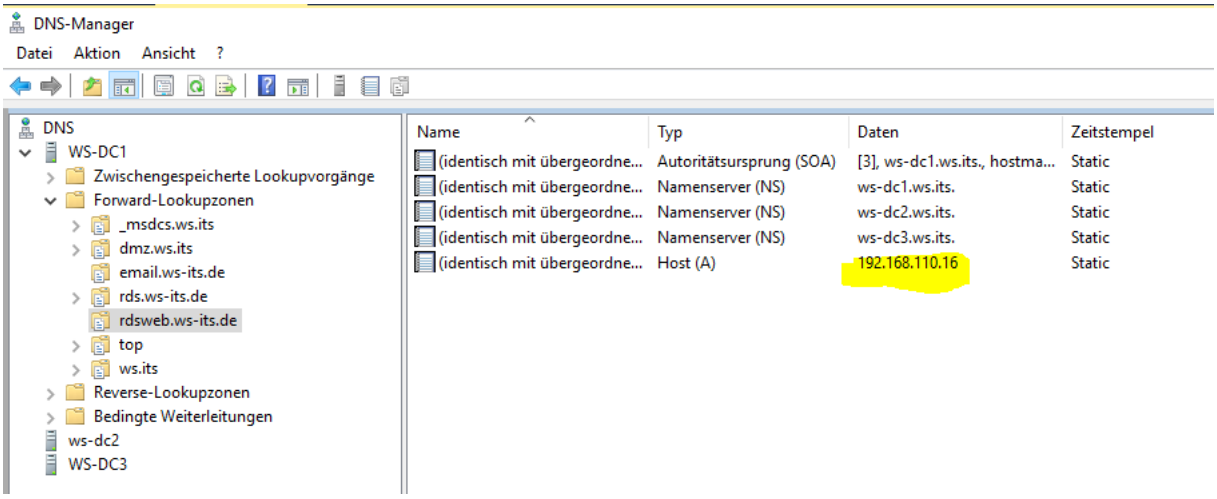
Die neue ACL leitet dann auf das neue Backend:



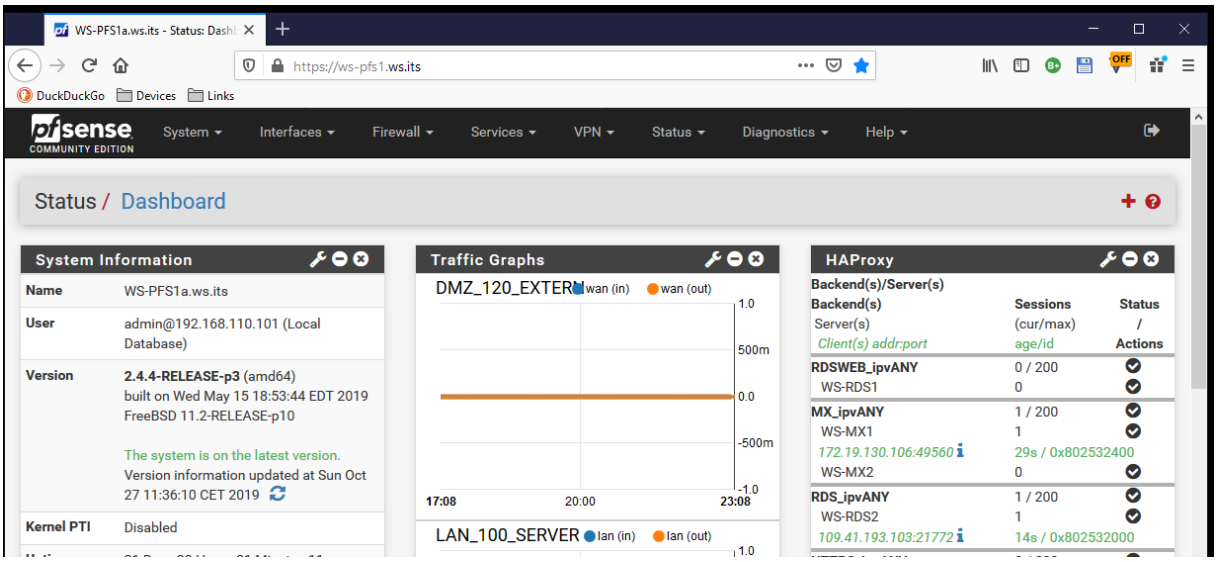
Ein Apply später ist die Anwendung bereit:



Intern dürfen meine Clients weiter direkt mit dem RDS-Broker sprechen und so den HAProxy umgehen. Keep it simple, oder?

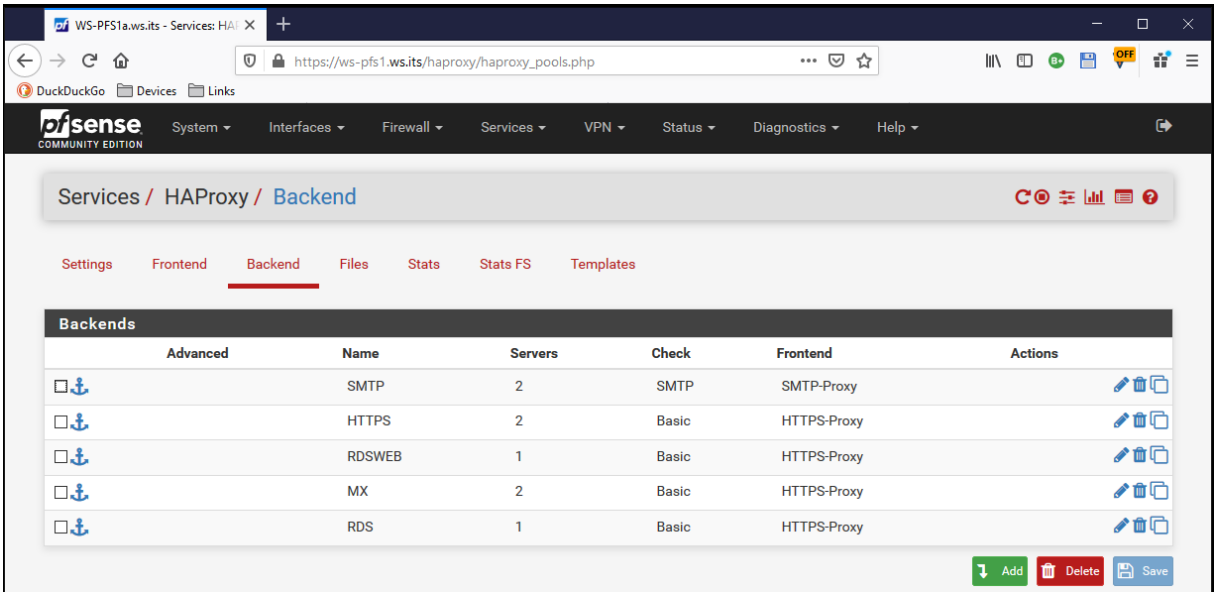


Die neue App zeigt sich wie gewohnt im Dashboard der PFSense. Und ein Einwahlversuch von außen wird erfolgreich auf meinen RDS-Server geleitet:

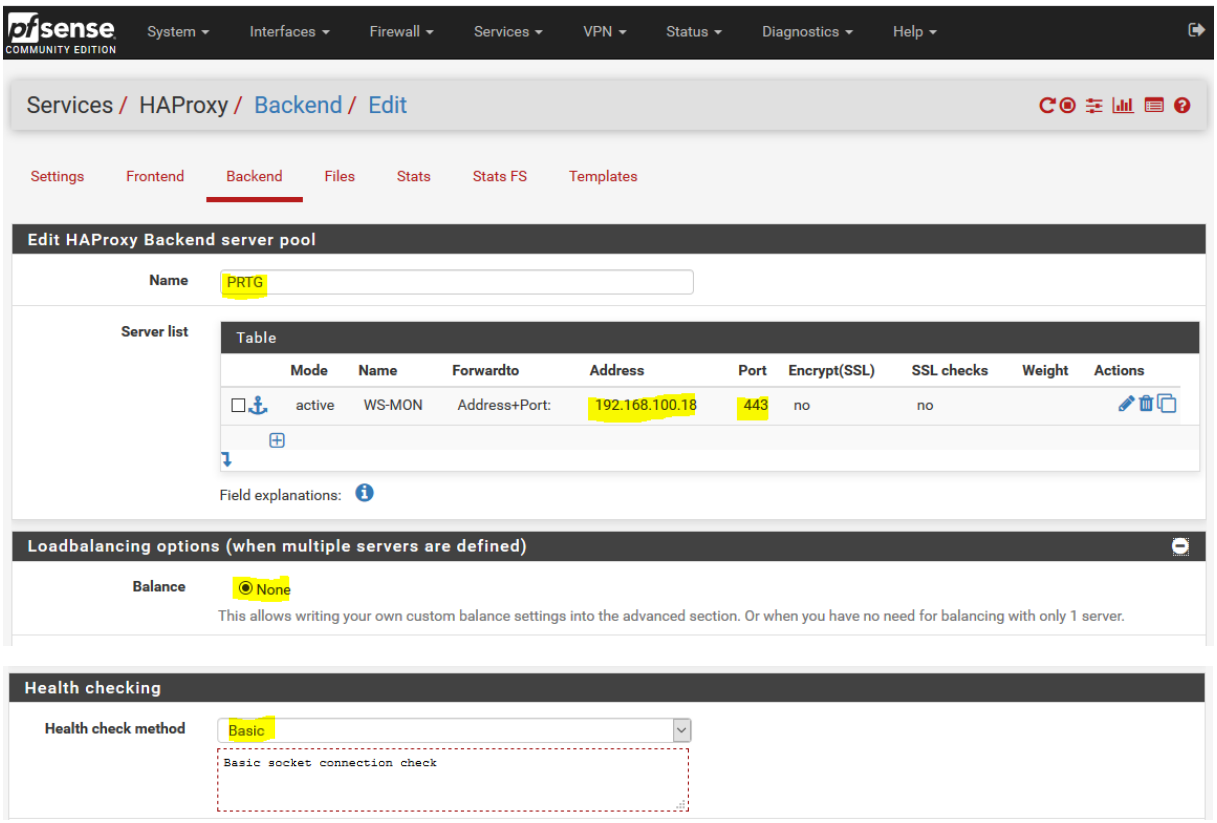


HAProxy für PRTG

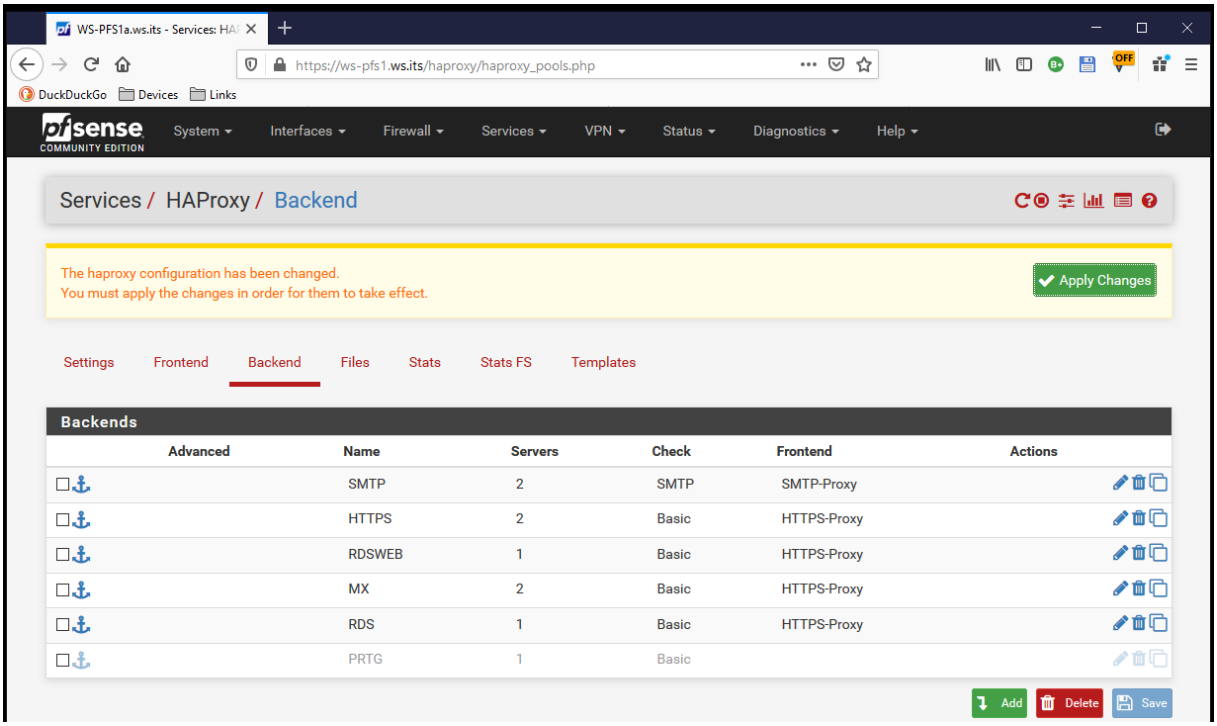
Meine letzte Anwendung im Web Application Proxy ist mein PRTG-Monitor. Mit diesem Zugriffspunkt erhalte ich Push-Benachrichtigungen auf mein Smartphone, wenn es Probleme in meiner Infrastruktur gibt. Die Vorgehensweise ist gleich mit der meines RDS-Servers. Ich erstelle wieder ein Backend im HAProxy. Ein Klick auf add und es geht los:



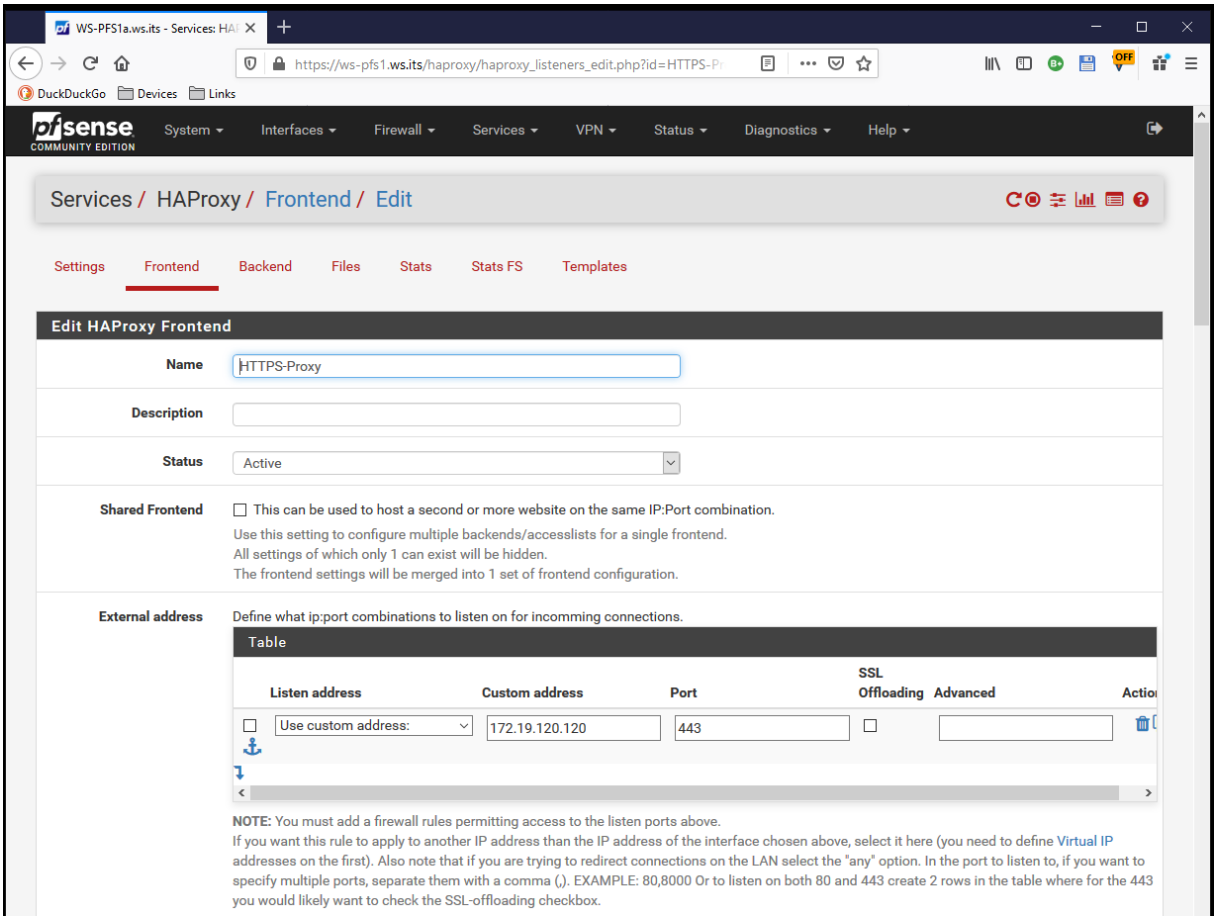
Das Ziel ist mein WS-MON, auf dem die PRTG-Installation läuft. Ich benötige kein Load Balancing und als Prüfmechanismus genügt der Standardtest:

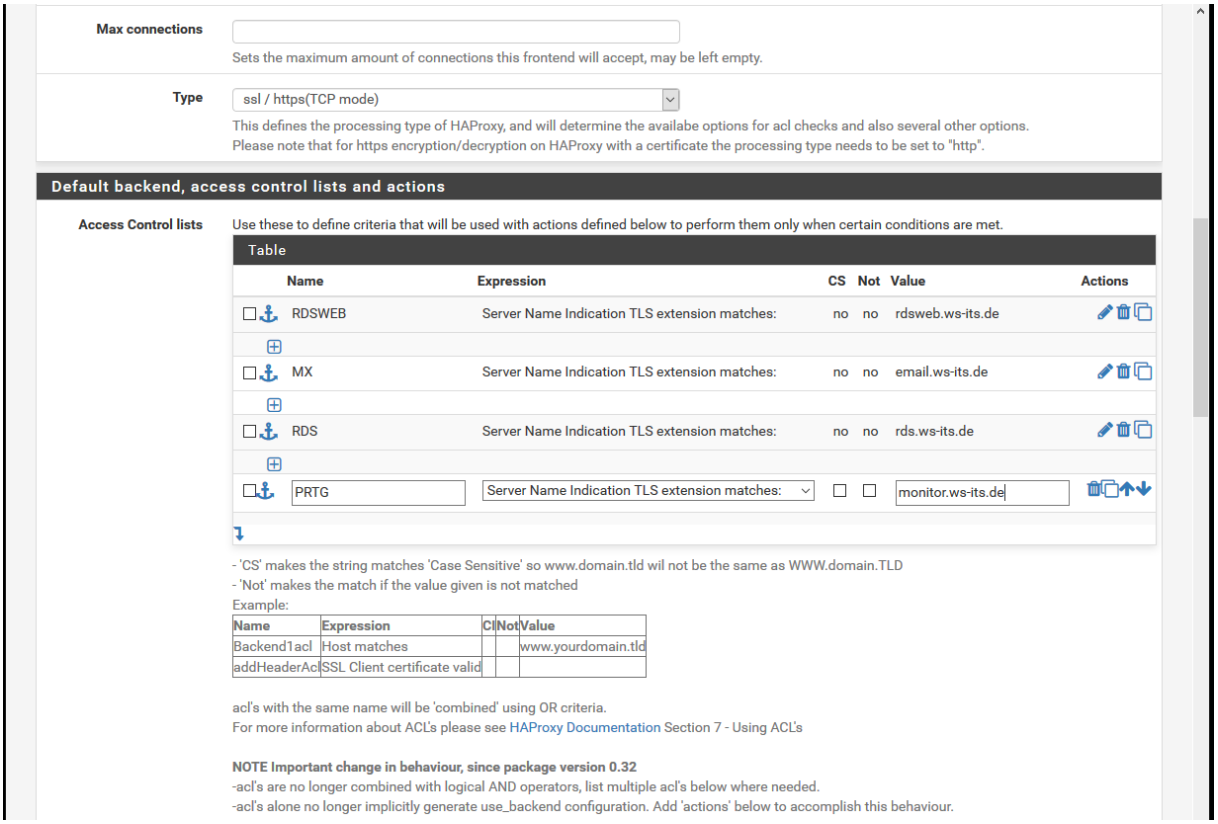


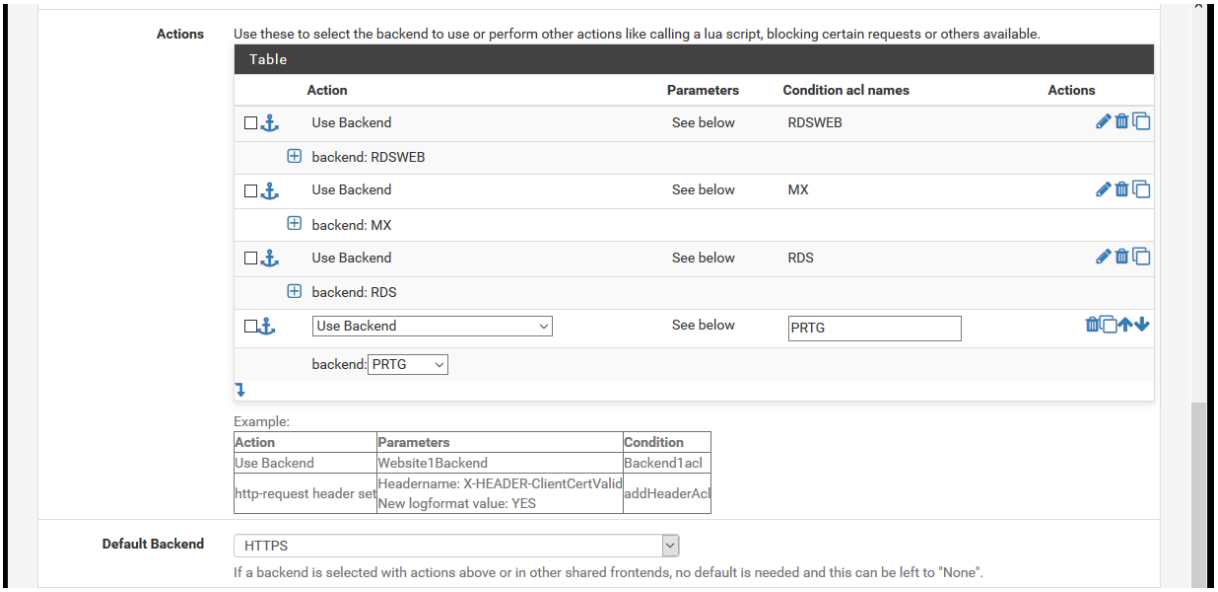
Ein Apply später ist das Backend einsatzbereit:



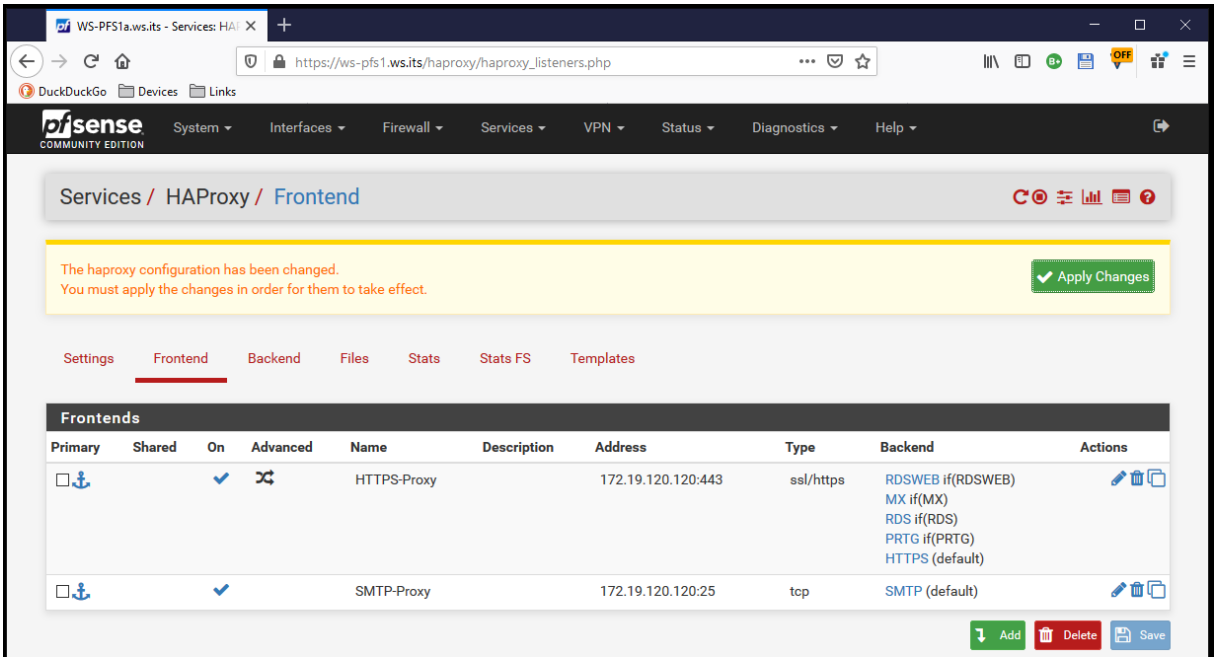
Und ein letztes mal editiere ich mein HAProxy-Frontend und erstelle die ACL und die Weiterleitung:



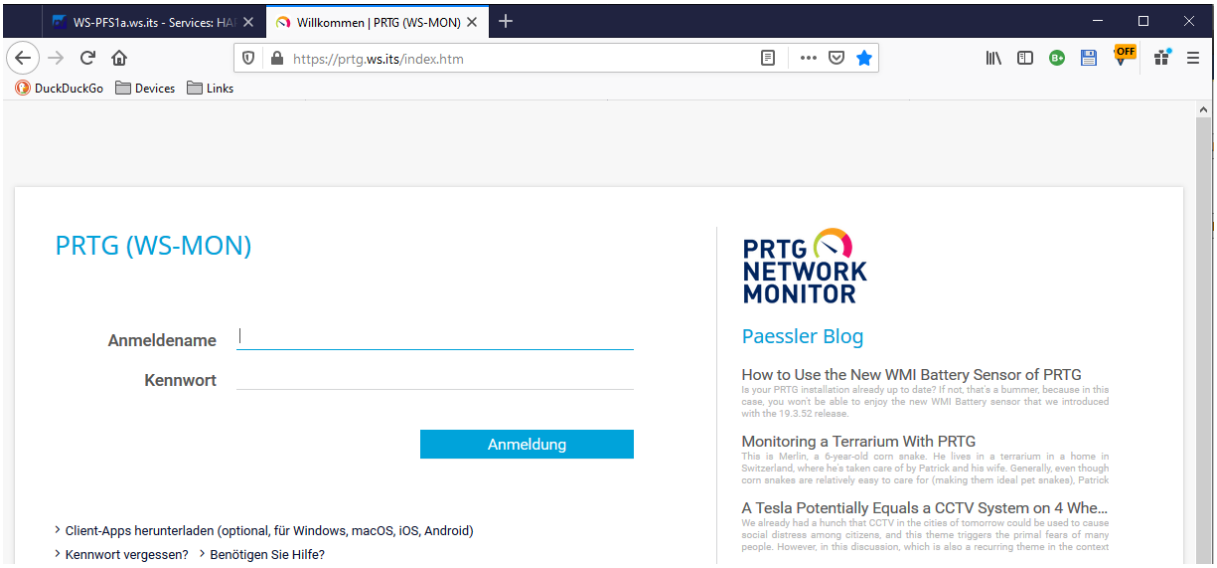




Ein finales Apply später ist auch diese Anwendung umgestellt:



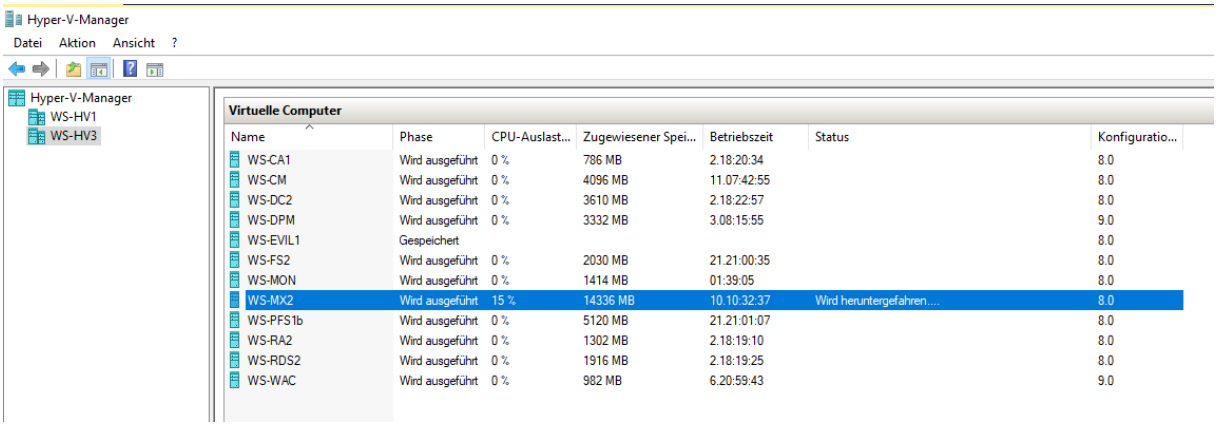
Intern spreche ich meine PRTG-Installation direkt an. Das funktioniert davon unabhängig. Meine App im Smartphone zeigt eine kurze Zertifikatbestätigung an und ist danach wieder verbunden:



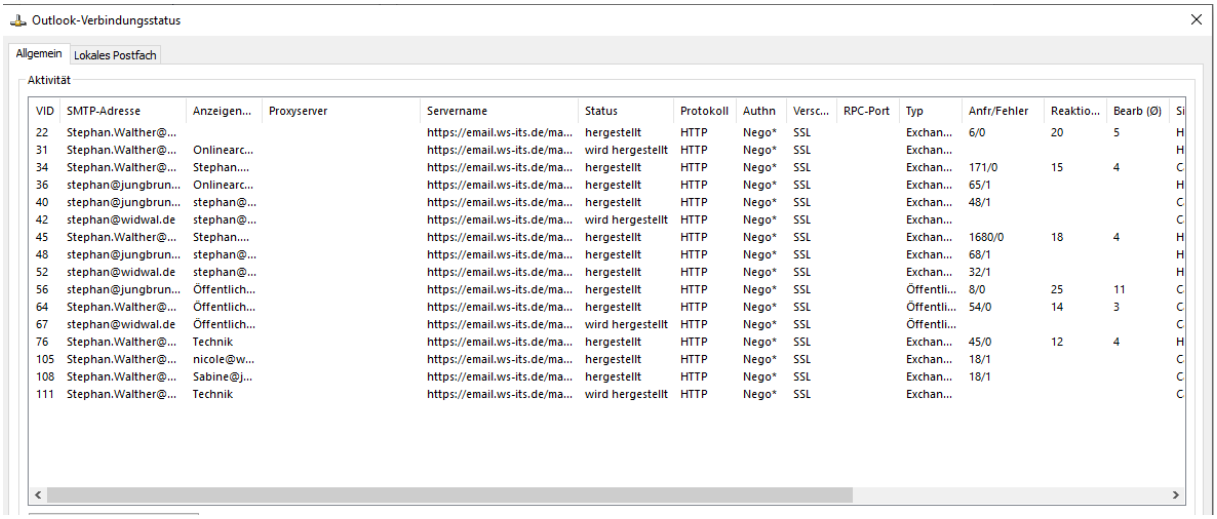
Testlauf HA

Bevor ich meinen Web Application Proxy abreiße möchte ich die neue Lösung gerne testen. Dafür werde ich nun verschiedene Server ausschalten und danach bzw. währenddessen von der zugehörigen Anwendung aus prüfen, ob der Schwenk funktioniert.

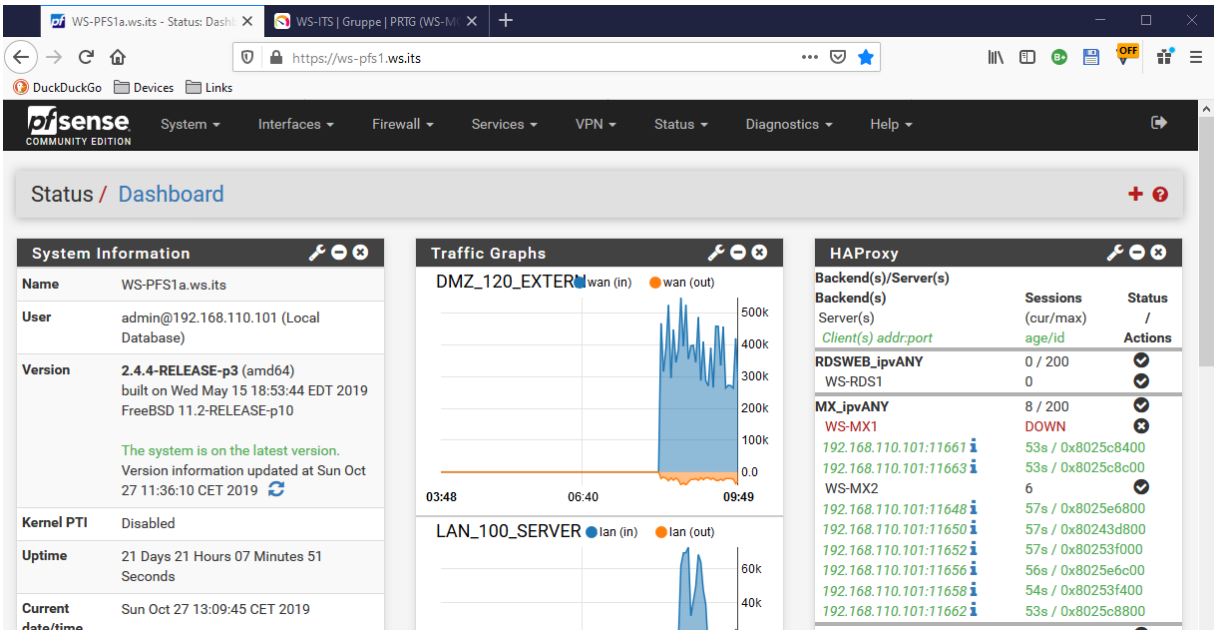
Zuerst fahre ich einen der Exchange Server herunter:



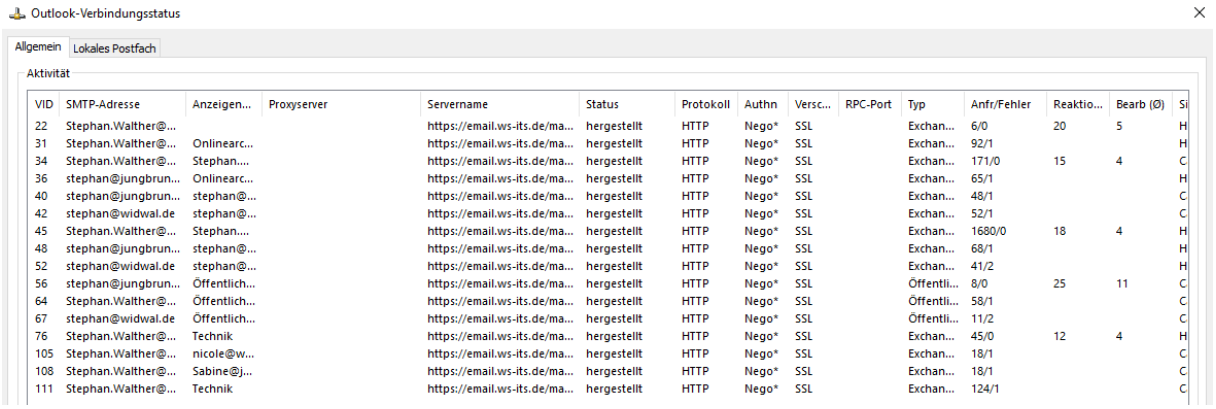
Dabei beobachte ich in meinem Outlook-Verbindungsstatus, wie einige Verbindungen schwenken:



Die gleiche Information erhalte ich auch in der PFSense. Das HAProxy-Modul hat erkannt, dass der Server nicht mehr einsatzbereit ist und die Verbindungen schwenken zum anderen Server:



Der Prozess dauert nur wenige Sekunden. In der Outlook-Verbindungsanzeige sind alle Verbindungen wieder hergestellt. Ohne die Anzeige hätte ich als Benutzer nichts bemerkt:

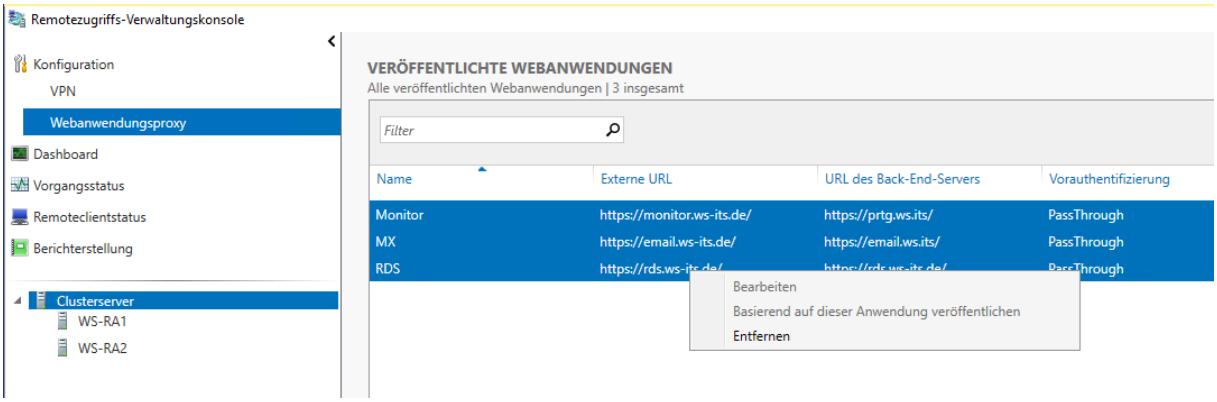


VID	SMTP-Adresse	Anzeigen...	Proxyserver	Servername	Status	Protokoll	Authn	Versc...	RPC-Port	Typ	Anfr/Fehler	Reaktio...	Bearb (Ø)	Si
22	Stephan.Walther@...			https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	6/0	20	5	H
31	Stephan.Walther@...	Onlinearc...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	92/1			H
34	Stephan.Walther@...	Stephan....		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	171/0	15	4	C
36	stephan@jungbrun...	Onlinearc...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	65/1			H
40	stephan@jungbrun...	stephan@...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	48/1			C
42	stephan@widwal.de	stephan@...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	52/1			C
45	Stephan.Walther@...	Stephan....		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	1680/0	18	4	H
48	stephan@jungbrun...	stephan@...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	68/1			H
52	stephan@widwal.de	stephan@...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	41/2			H
56	stephan@jungbrun...	Öffentlich...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Öffentli...	8/0	25	11	C
64	Stephan.Walther@...	Öffentlich...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Öffentli...	58/1			C
67	stephan@widwal.de	Öffentlich...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Öffentli...	11/2			C
76	Stephan.Walther@...	Technik		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	45/0	12	4	H
105	Stephan.Walther@...	nicole@w...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	18/1			C
108	Stephan.Walther@...	Sabine@j...		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	18/1			C
111	Stephan.Walther@...	Technik		https://email.ws-its.de/ma...	hergestellt	HTTP	Nego*	SSL		Exchan...	124/1			C

Auch der Schwenk des HAProxy auf die zweite PFsense geschieht nahezu transparent. Das sieht gut aus.

Bereinigung WAP

Also kann ich jetzt die im WAP veröffentlichten Anwendungen entfernen:



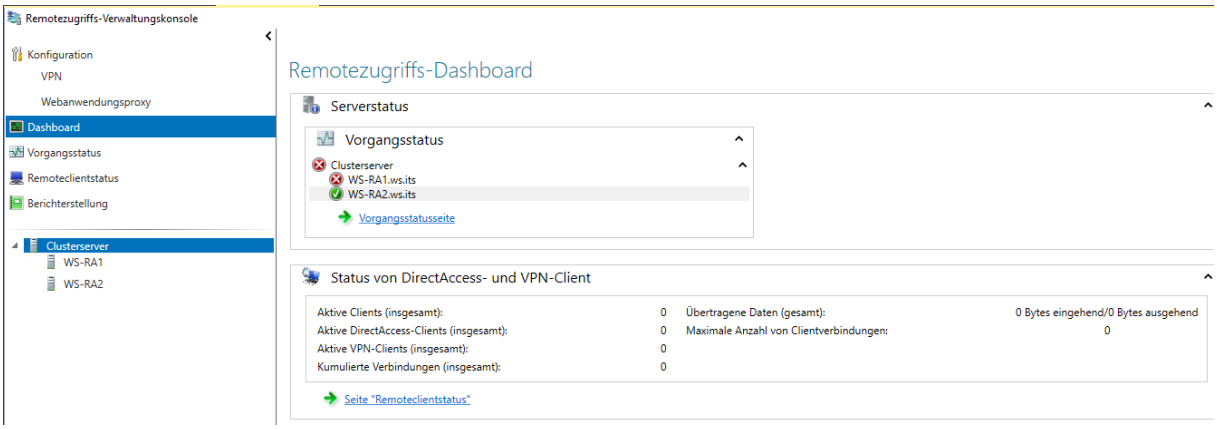
Damit wird mein Web Application Proxy Cluster nicht länger verwendet.

Dieses Kapitel habe ich bereits im Oktober geschrieben. Es gehört aber thematisch in diesen Artikel. Ab jetzt geht es wieder im Dezember 2019 weiter…

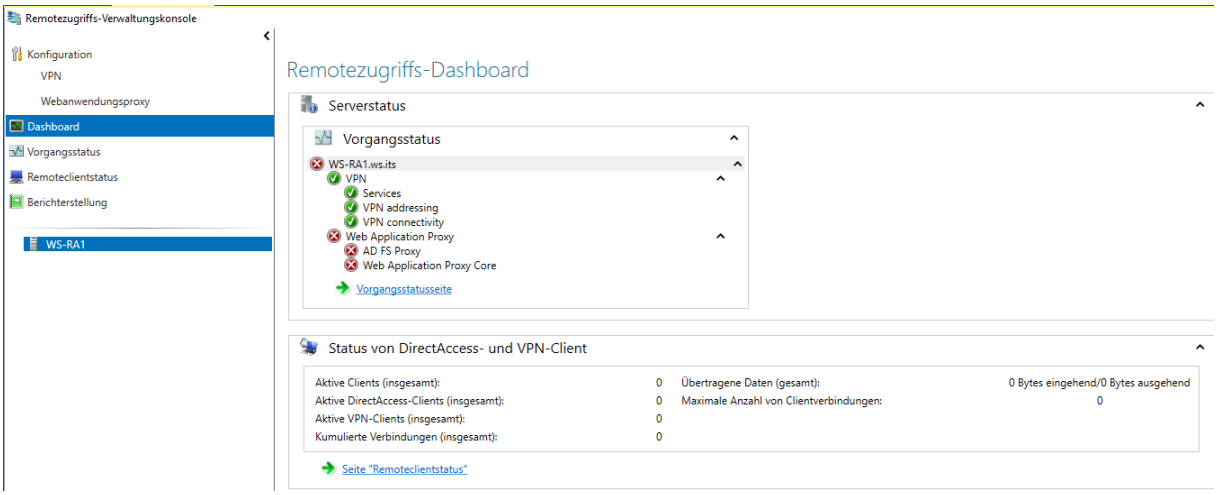
Entfernung von ADFS und WAP

Vorbereitung

Beide WAP-Server bilden einen WAP-Cluster. Dieser ist aber seit einigen Tagen gestört:

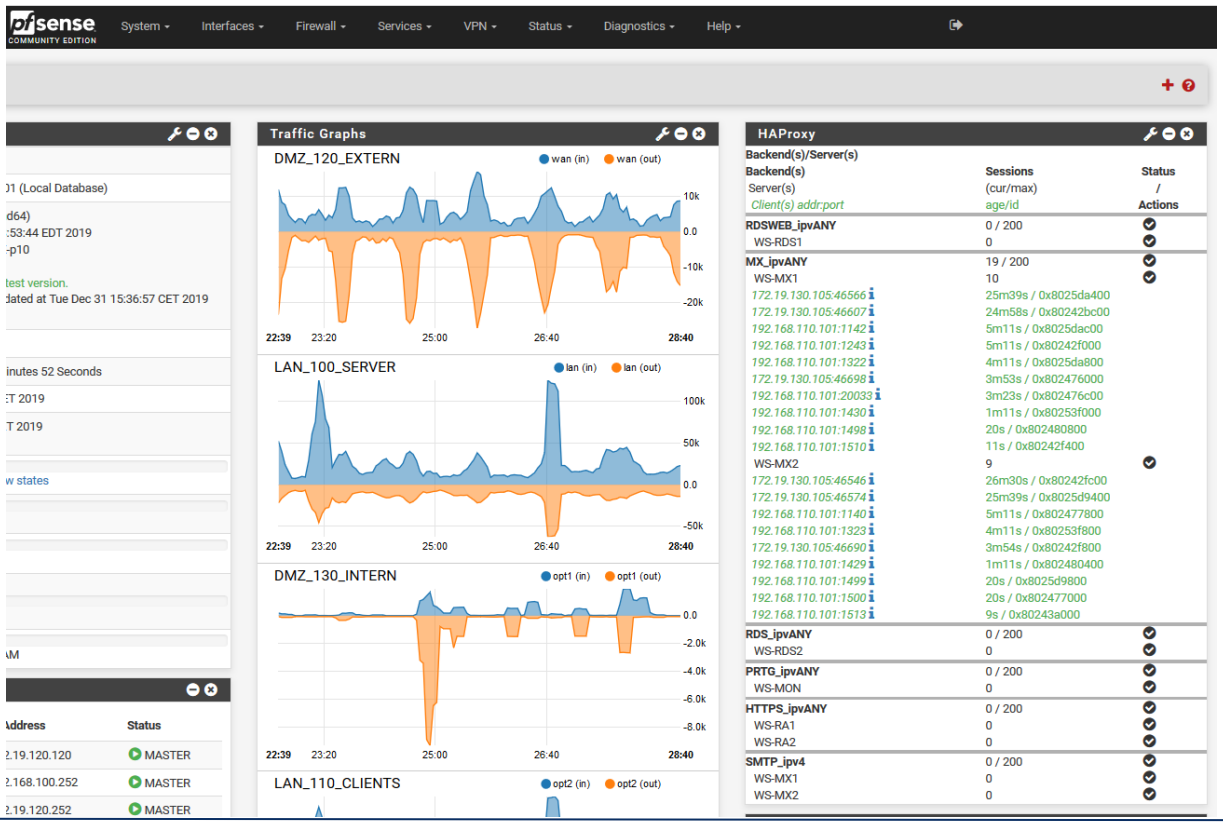


In den Details der Administrationsoberfläche sieht man, dass die Services nicht laufen. Diese lassen sich auch nicht mehr starten. Das Eventlog des Servers ist voll mit Fehlermeldungen. Die Ursache ist mir aber nach dem Entschluss der Service-Entfernung egal:

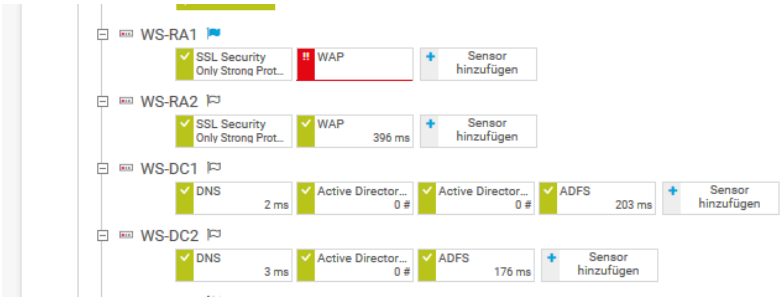


Es sind im WAP-Cluster keine Webanwendungen mehr veröffentlicht. Die habe ich alle in meinen HA-Proxy der PFSense integriert. Sonst gäbe es an dieser Stelle noch ein paar offene Löschaktionen:

Hier sieht man rechts im Bild meinen Nachfolger des WAP-Clusters:



Natürlich wurde ich von meinem Monitoring über den Ausfall des Services auf WS-RA1 informiert:

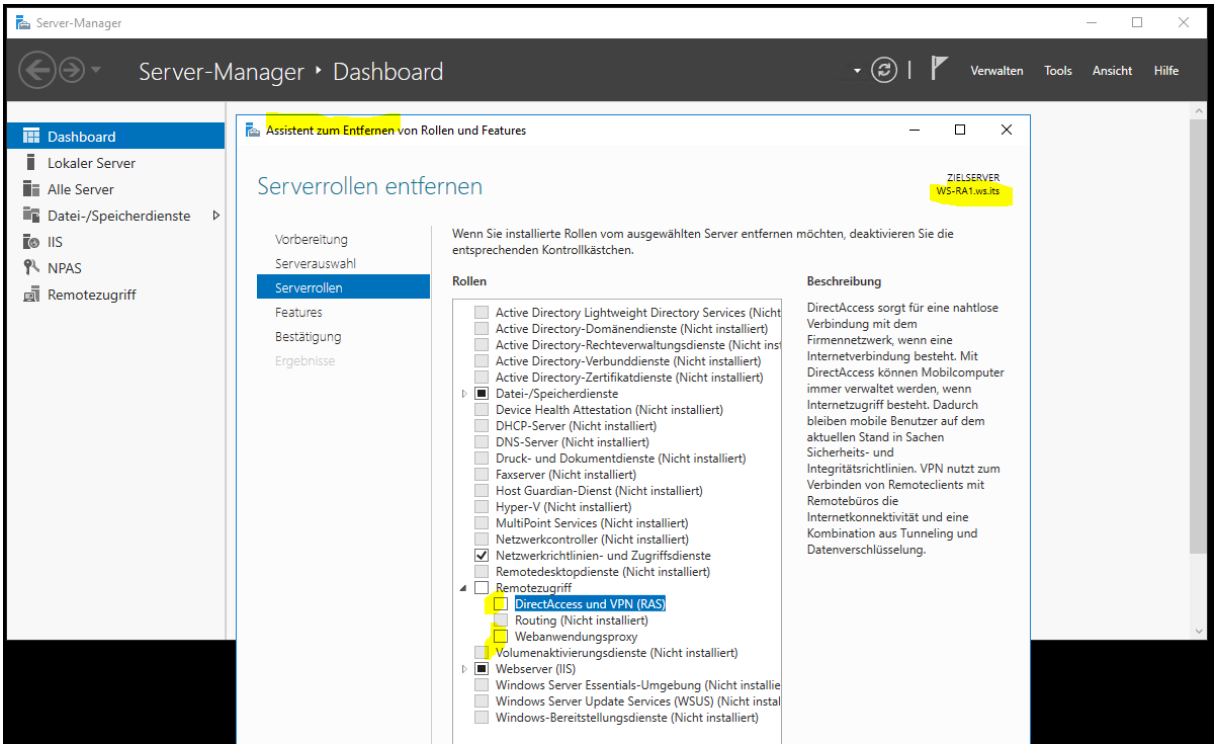


Eine Entfernung der Services WAP und ADFS sind problemlos möglich, da es keine Abhängigkeiten mehr gibt. Bleibt nur die Reihenfolge der Deprovisionierung:

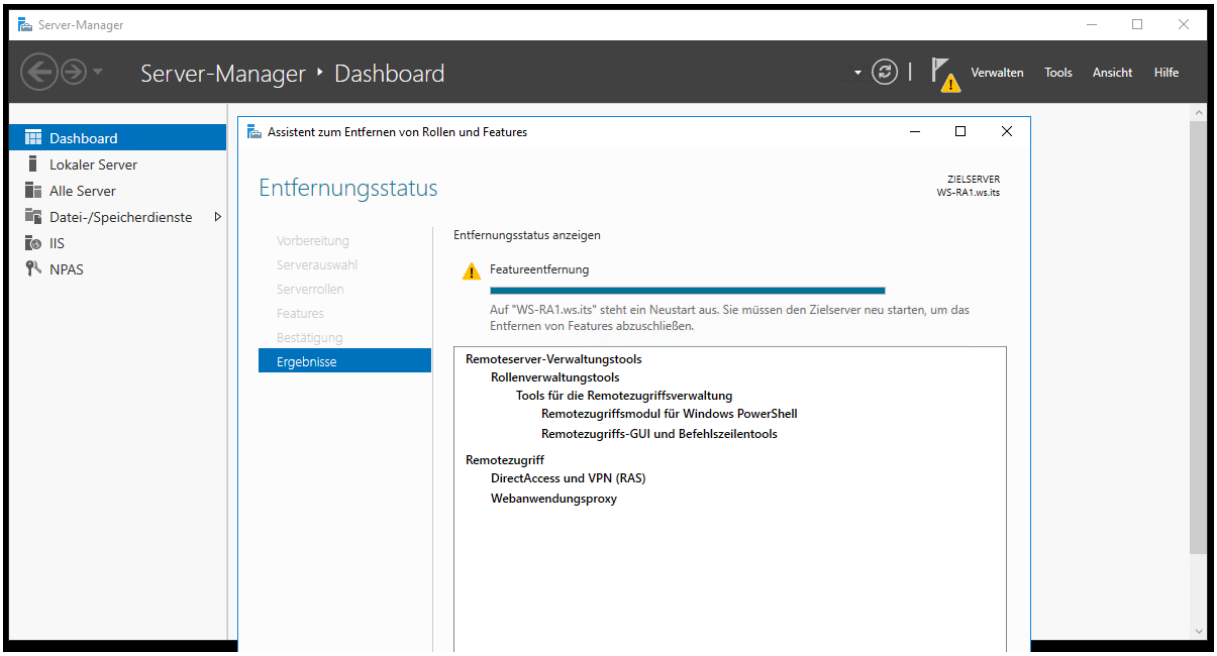
- Ich werden zuerst den defekten WAP-Service auf WS-RA1 löschen
- Dann kann ich WAP auf WS-RA2 korrekt im ADFS löschen.
- ADFS besteht bei mir aus 2 Farm-Mitgliedern: WS-DC1 und WS-DC2. Dabei ist ein Server der Master, alle anderen sind im Slave-Mode. Zuerst entferne ich den Slave.
- Zuletzt entferne ich den ADFS-Master.

Entfernen von WAP auf WS-RA1

Den defekten WAP-Server WS-RA1 kann ich nicht sauber deregistrieren. Daher entferne ich die Rolle und hoffe, dass damit ein positiver Effekt erzielt wird. Nebenbei entferne ich auch das Feature für die VPN-Services:

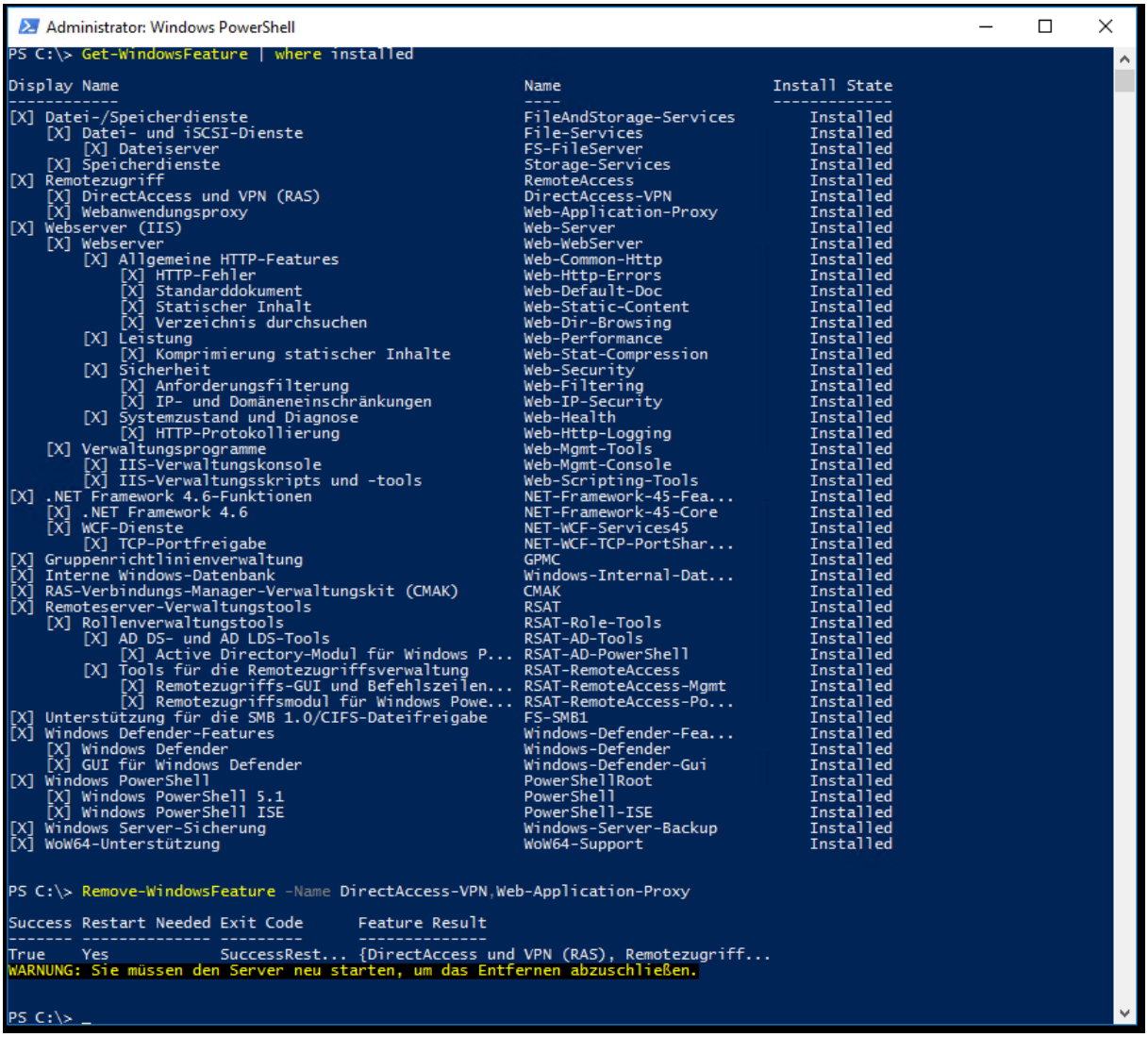


Der Abschluss ist ein einfacher Neustart:



Entfernen von WAP auf WS-RA2

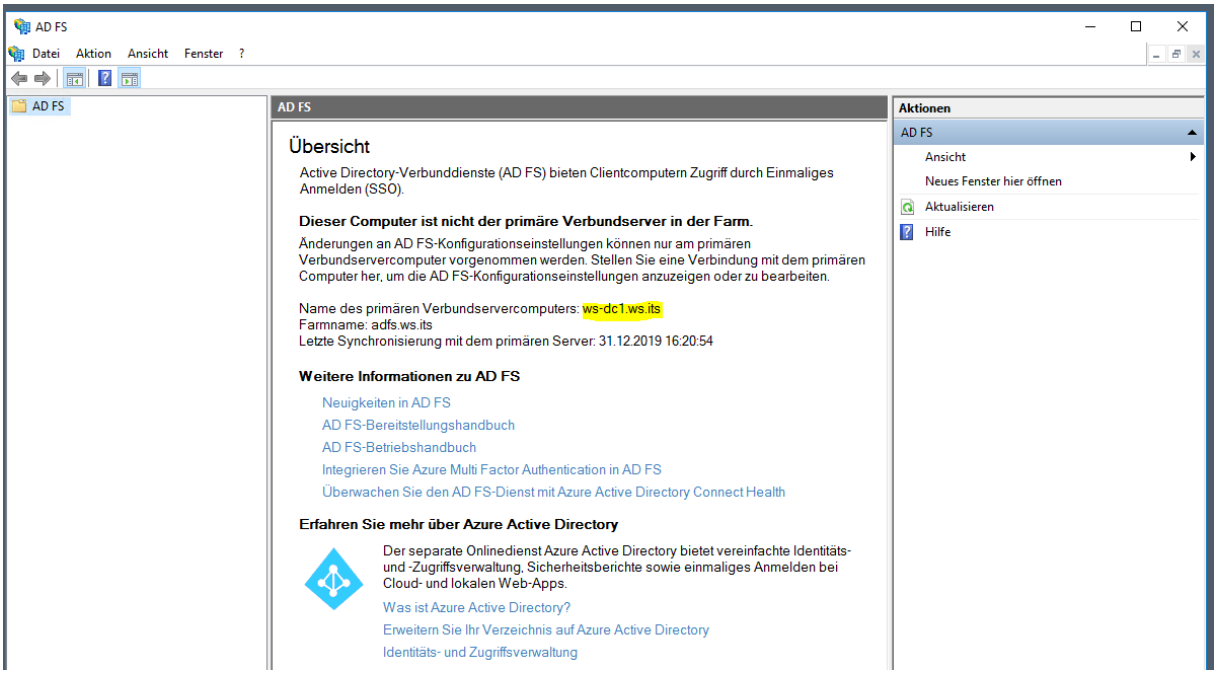
Auf dem aktiven Server WS-RA2 entferne ich die Rolle mit der PowerShell. Warum? Weil ich es kann!



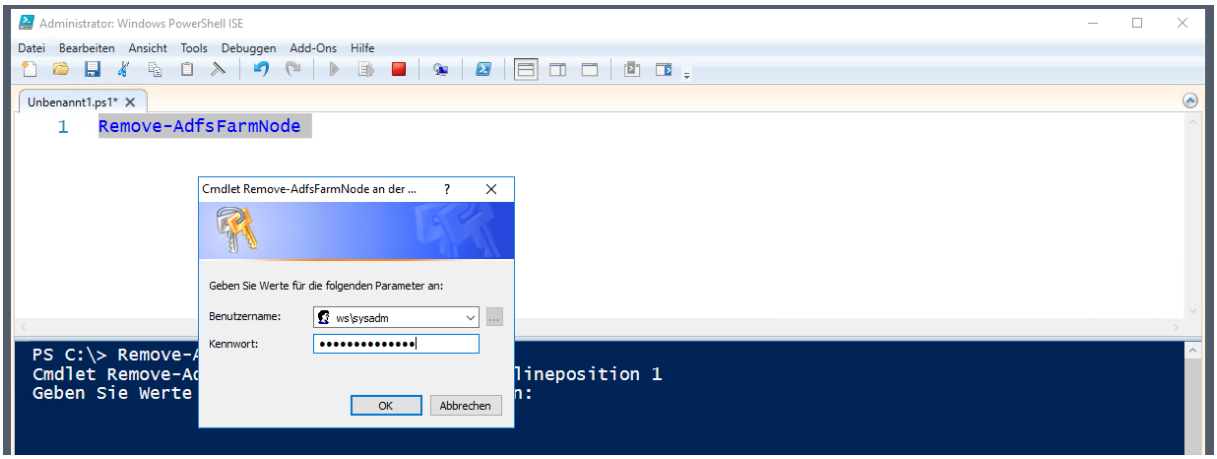
Damit ist der letzte WAP bereinigt.

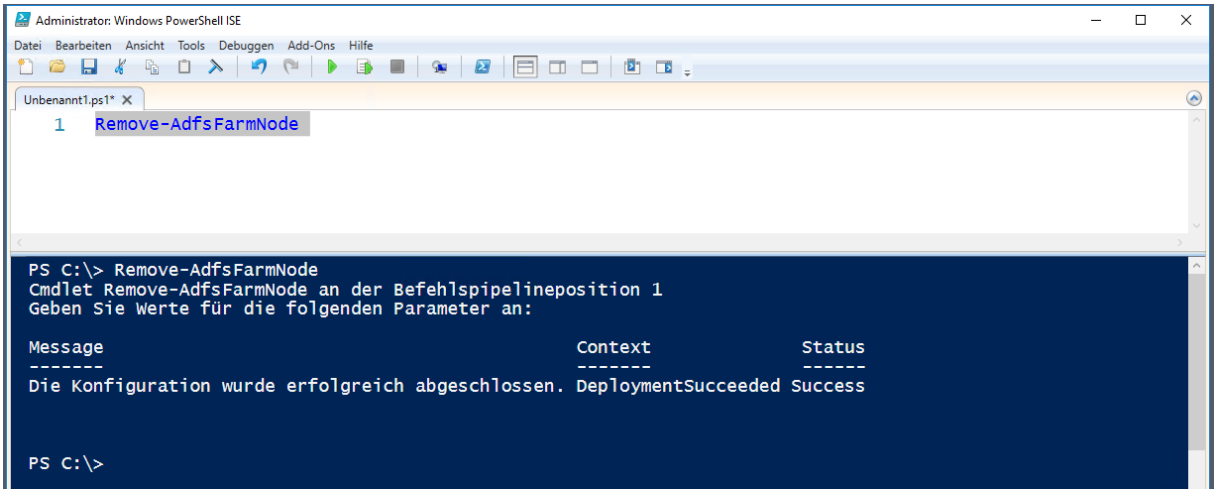
Entfernen von ADFS auf WS-DC2 (Slave)

Weiter geht es im ADFS. Der Slave-Server ist mein Domain Controller WS-DC2. Ein ADFS auf einer solchen Maschine ist alles andere als optimal. Aber „damals“ hatte ich kaum noch Systemressourcen frei… Das würde ich heute nicht mehr so umsetzen. In der ADFS-Konsole kann ich den Mode des Servers prüfen:

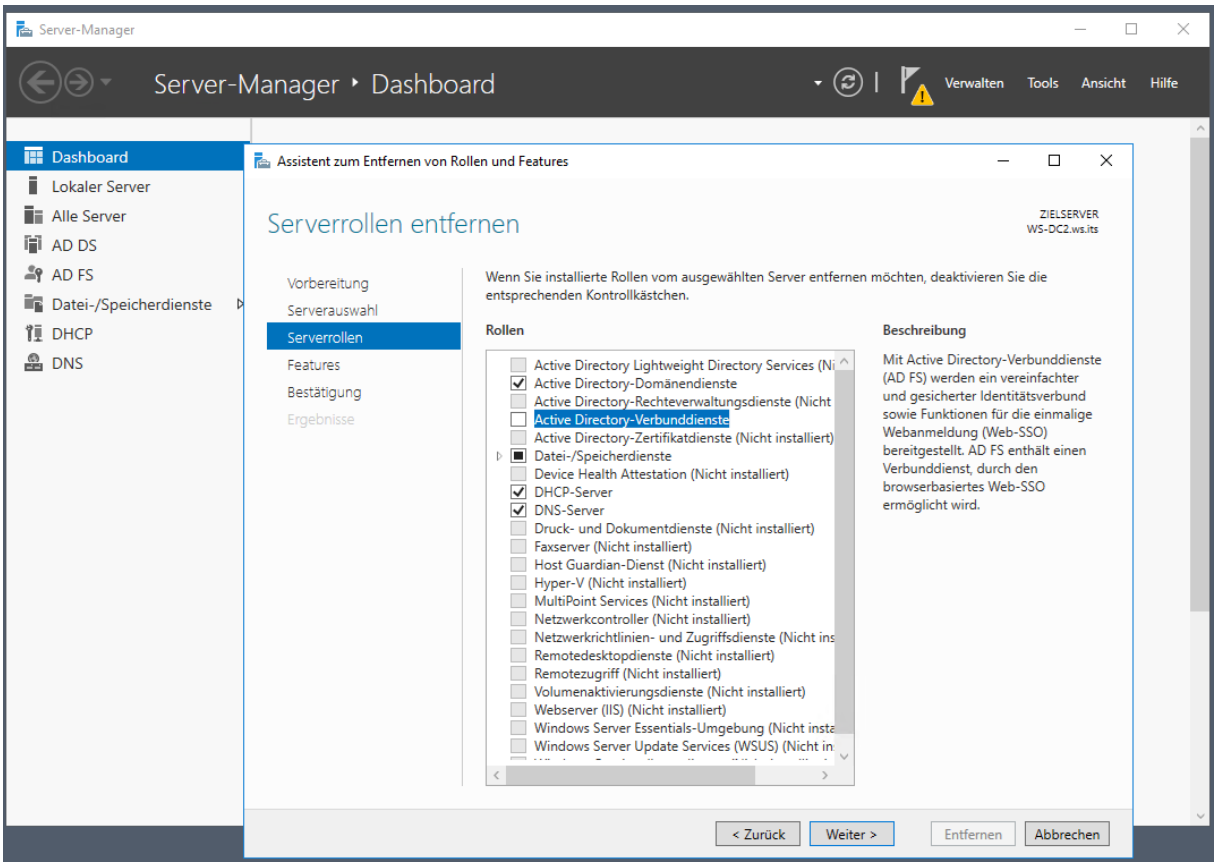


Bevor ich die Rolle deinstallieren kann, entferne ich WS-DC2 als FarmNode aus der ADFS-Farm. Das geht mit der PowerShell:

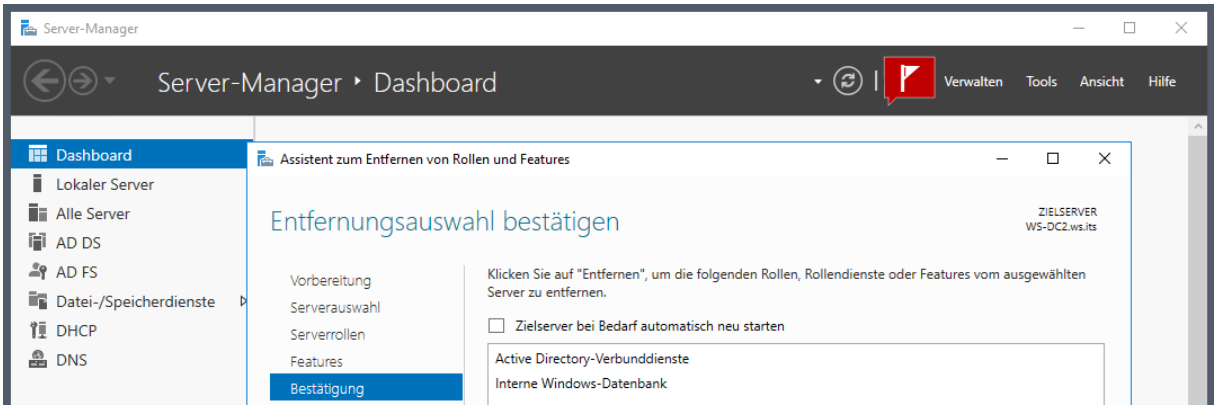




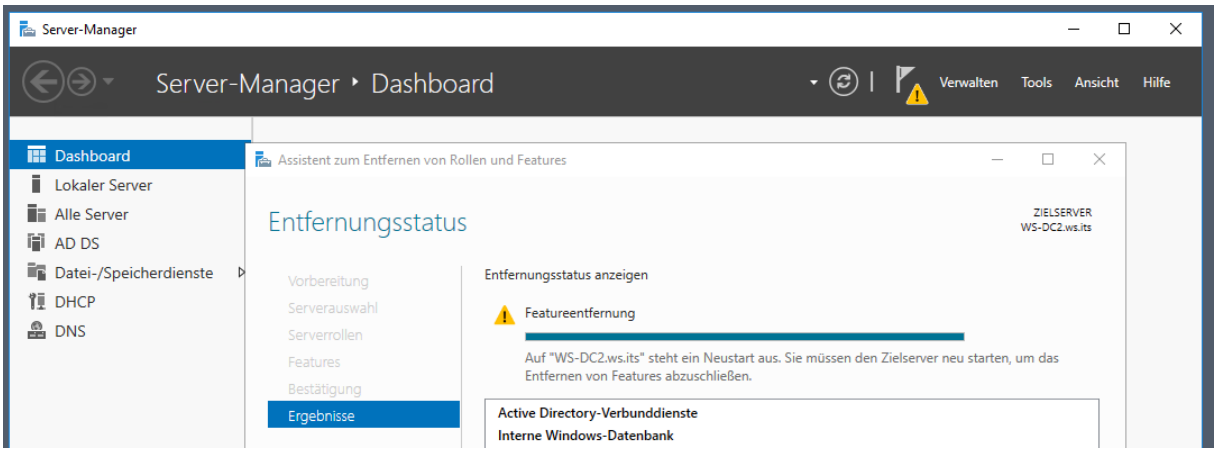
Nun ist die Rolle kein Problem mehr. Ich wähle die Deinstallation im Server Manager aus:



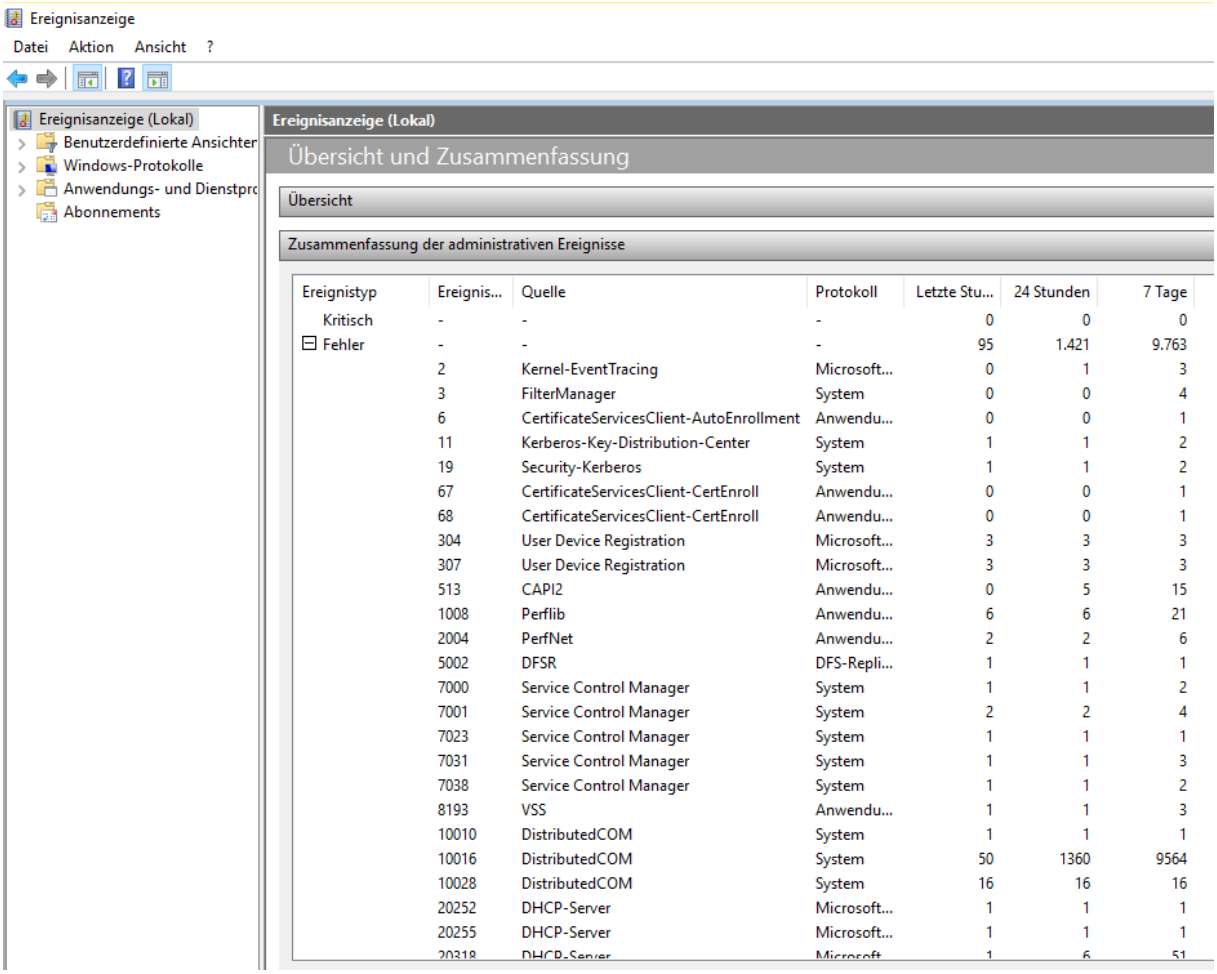
Auch die Windows Internal Database des ADFS wird nicht mehr benötigt:



Die Entfernung muss mit einem Neustart abgeschlossen werden. Da ich 2 Domain Controller einsetze, kann ich einen davon einfach durchstarten:



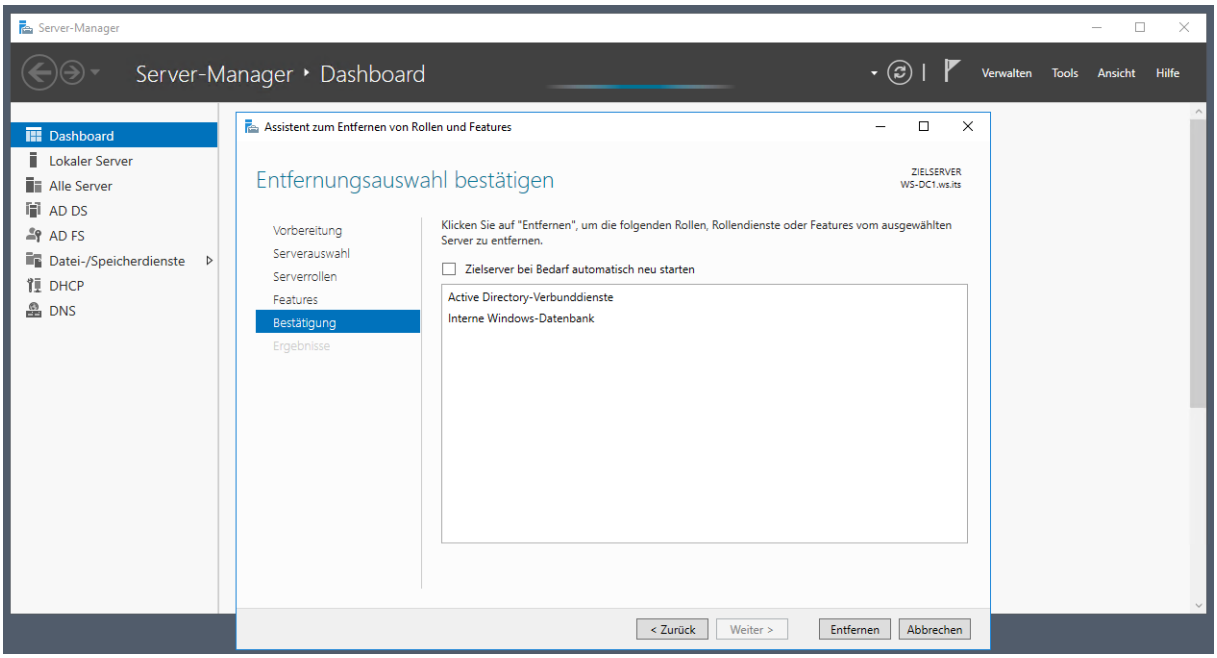
Ein kurzer Blick in die Ereignisprotokolle nach dem Neustart zeigt keine Fehler oder Warnungen. Das hat funktioniert:



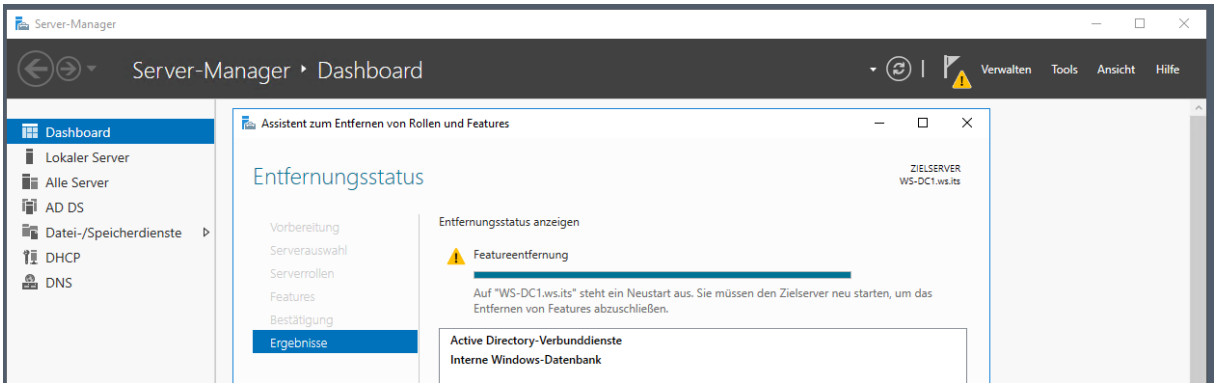
Ereignistyp	Ereignis...	Quelle	Protokoll	Letzte Stu...	24 Stunden	7 Tage
Kritisch	-	-	-	0	0	0
⊟ Fehler	-	-	-	95	1.421	9.763
	2	Kernel-EventTracing	Microsoft...	0	1	3
	3	FilterManager	System	0	0	4
	6	CertificateServicesClient-AutoEnrollment	Anwendu...	0	0	1
	11	Kerberos-Key-Distribution-Center	System	1	1	2
	19	Security-Kerberos	System	1	1	2
	67	CertificateServicesClient-CertEnroll	Anwendu...	0	0	1
	68	CertificateServicesClient-CertEnroll	Anwendu...	0	0	1
	304	User Device Registration	Microsoft...	3	3	3
	307	User Device Registration	Microsoft...	3	3	3
	513	CAPI2	Anwendu...	0	5	15
	1008	Perflib	Anwendu...	6	6	21
	2004	PerfNet	Anwendu...	2	2	6
	5002	DFSR	DFS-Repli...	1	1	1
	7000	Service Control Manager	System	1	1	2
	7001	Service Control Manager	System	2	2	4
	7023	Service Control Manager	System	1	1	1
	7031	Service Control Manager	System	1	1	3
	7038	Service Control Manager	System	1	1	2
	8193	VSS	Anwendu...	1	1	3
	10010	DistributedCOM	System	1	1	1
	10016	DistributedCOM	System	50	1360	9564
	10028	DistributedCOM	System	16	16	16
	20252	DHCP-Server	Microsoft...	1	1	1
	20255	DHCP-Server	Microsoft...	1	1	1
	20318	DHCP-Server	Microsoft...	1	6	51

Entfernen von ADFS auf WS-DC1 (Master)

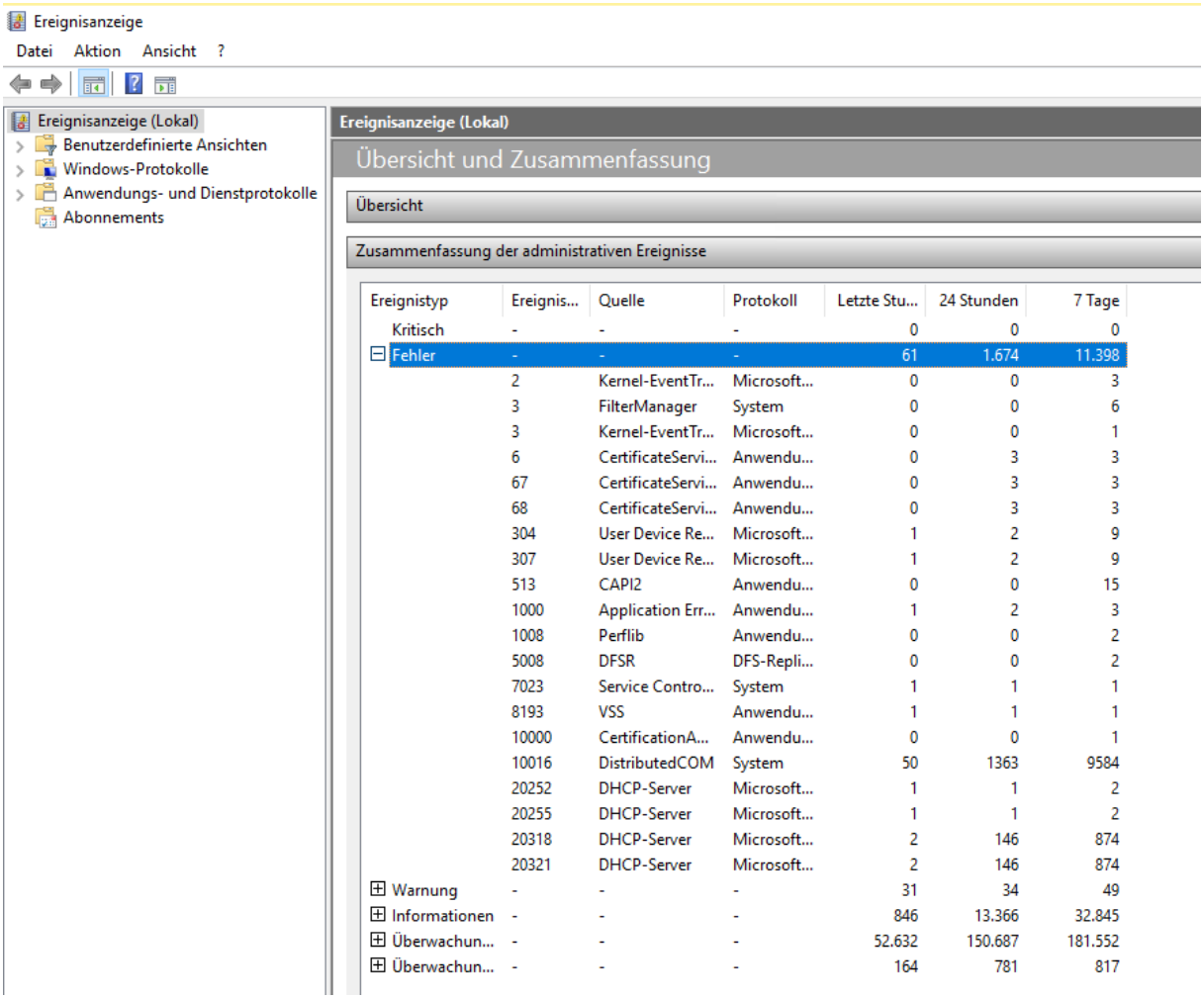
So bleibt nur noch der ADFS-Masternode über. Also geht's auf zum WS-DC1. Hier kann ich die Deinstallation direkt starten. Der letzte ADFS-Node macht sprichwörtlich das Licht aus:



Der Neustartwunsch kommt erwartet und wird umgesetzt:



Nach dem Neustart kontrolliere ich wieder die Eventlogs. Auch hier gibt es keine Probleme im Bezug auf die vorherige Entfernung:

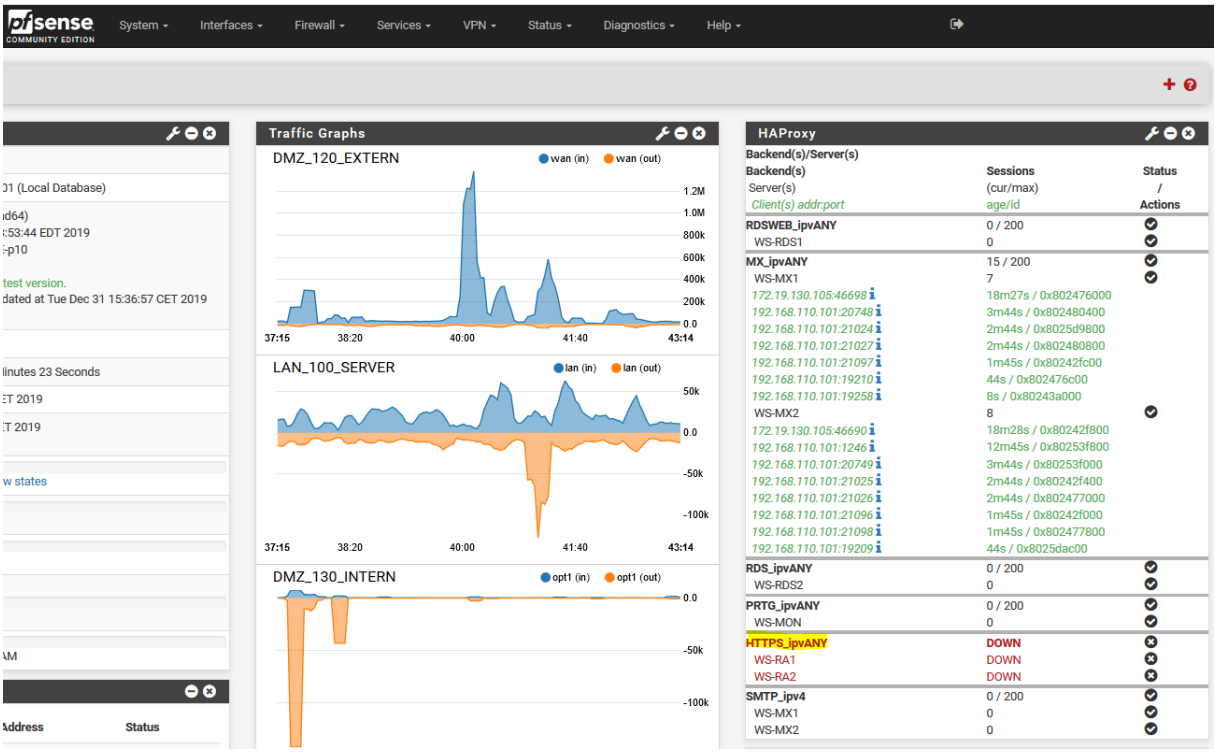


Ereignistyp	Ereignis...	Quelle	Protokoll	Letzte Stu...	24 Stunden	7 Tage
Kritisch	-	-	-	0	0	0
Fehler	-	-	-	61	1.674	11.398
	2	Kernel-EventTr...	Microsoft...	0	0	3
	3	FilterManager	System	0	0	6
	3	Kernel-EventTr...	Microsoft...	0	0	1
	6	CertificateServi...	Anwendu...	0	3	3
	67	CertificateServi...	Anwendu...	0	3	3
	68	CertificateServi...	Anwendu...	0	3	3
	304	User Device Re...	Microsoft...	1	2	9
	307	User Device Re...	Microsoft...	1	2	9
	513	CAPI2	Anwendu...	0	0	15
	1000	Application Err...	Anwendu...	1	2	3
	1008	Perflib	Anwendu...	0	0	2
	5008	DFSR	DFS-Repli...	0	0	2
	7023	Service Contro...	System	1	1	1
	8193	VSS	Anwendu...	1	1	1
	10000	CertificationA...	Anwendu...	0	0	1
	10016	DistributedCOM	System	50	1363	9584
	20252	DHCP-Server	Microsoft...	1	1	2
	20255	DHCP-Server	Microsoft...	1	1	2
	20318	DHCP-Server	Microsoft...	2	146	874
	20321	DHCP-Server	Microsoft...	2	146	874
Warnung	-	-	-	31	34	49
Informationen	-	-	-	846	13.366	32.845
Überwachun...	-	-	-	52.632	150.687	181.552
Überwachun...	-	-	-	164	781	817

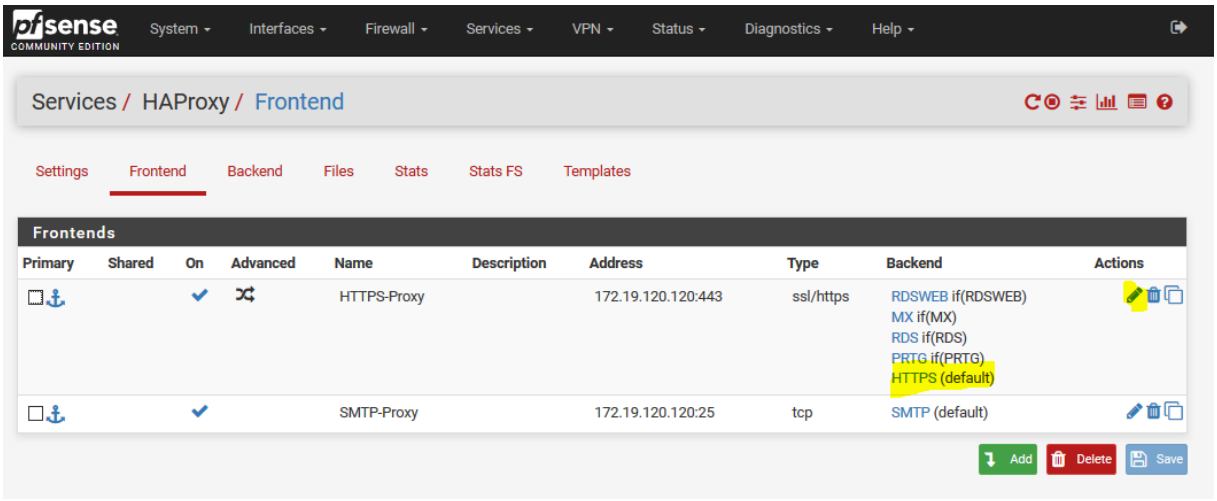
Damit sind WAP und ADFS entfernt. Falls ich diese Services morgen wieder benötige, dann installiere ich auf separaten Servern neu.

Bereinigung in der PFSense

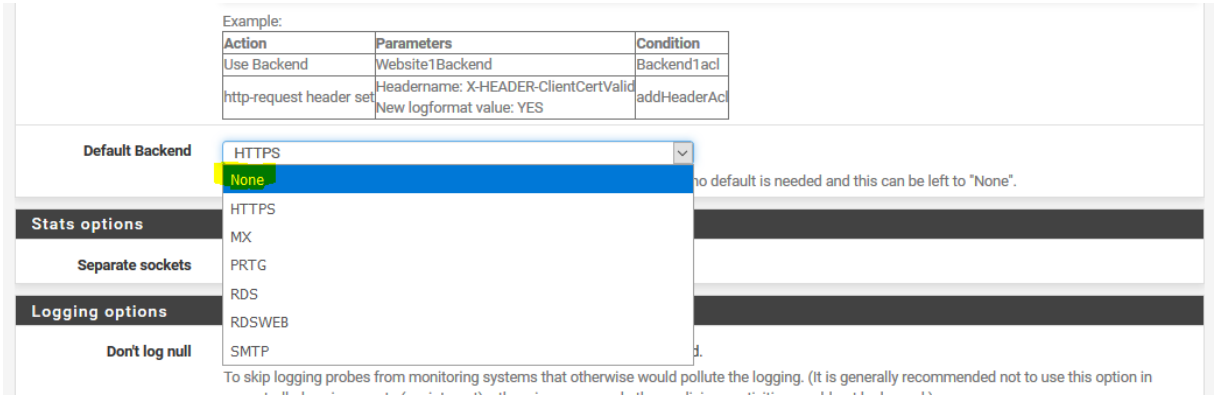
Ich habe WAP zwar nicht mehr verwendet, aber der Endpunkt war noch in meiner PFSense registriert. Der HA-Proxy, der die Funktion des WAP übernahm, hat die WAP-Clusternodes damals vorgelagert angesprochen. Diesen Endpunkt benötige ich nicht mehr:



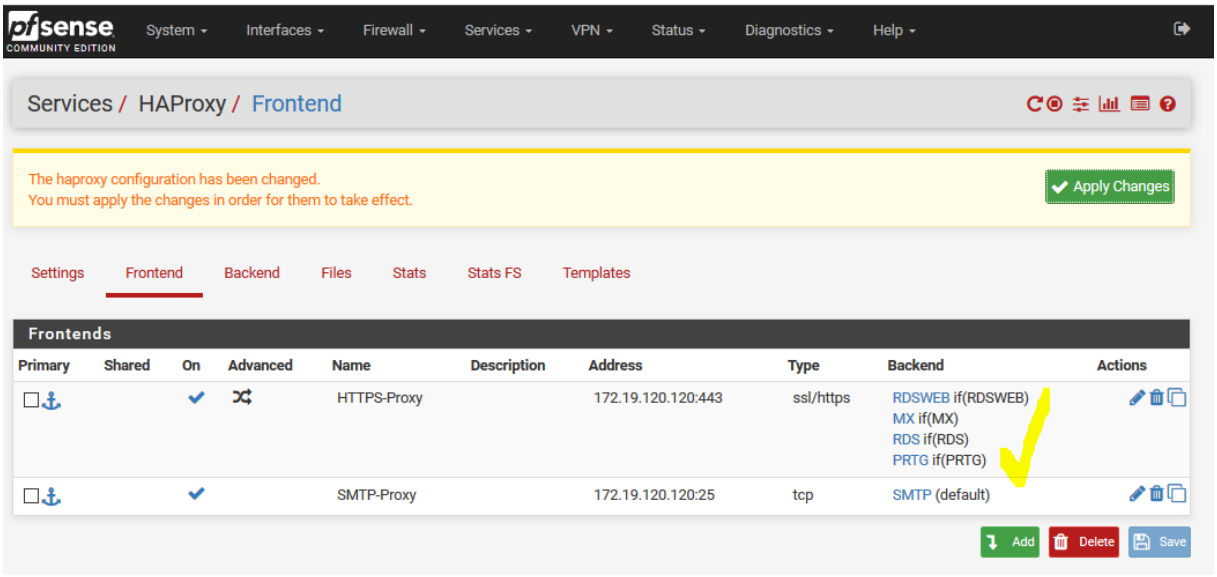
Also entferne ich das hinterlegte Backend für den WAP-Cluster aus dem Frontend des HAProxy:



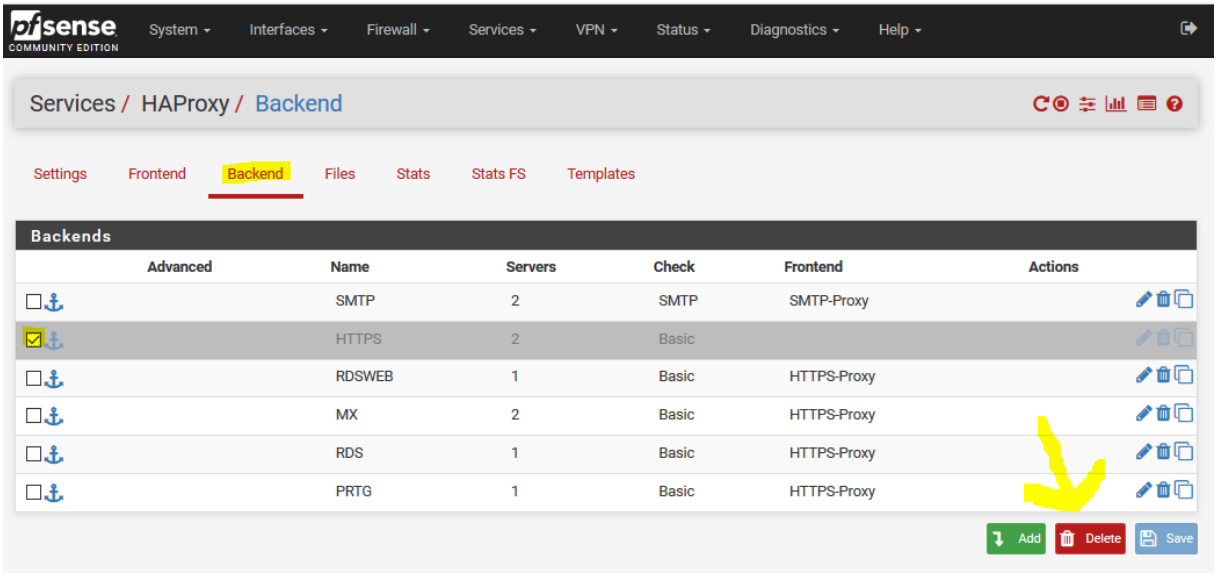
WAP war bis zu diesem Zeitpunkt der Default-Endpunkt. Da alle Verbindungen gezielt umgeleitet werden, setze ich den Default auf NONE:



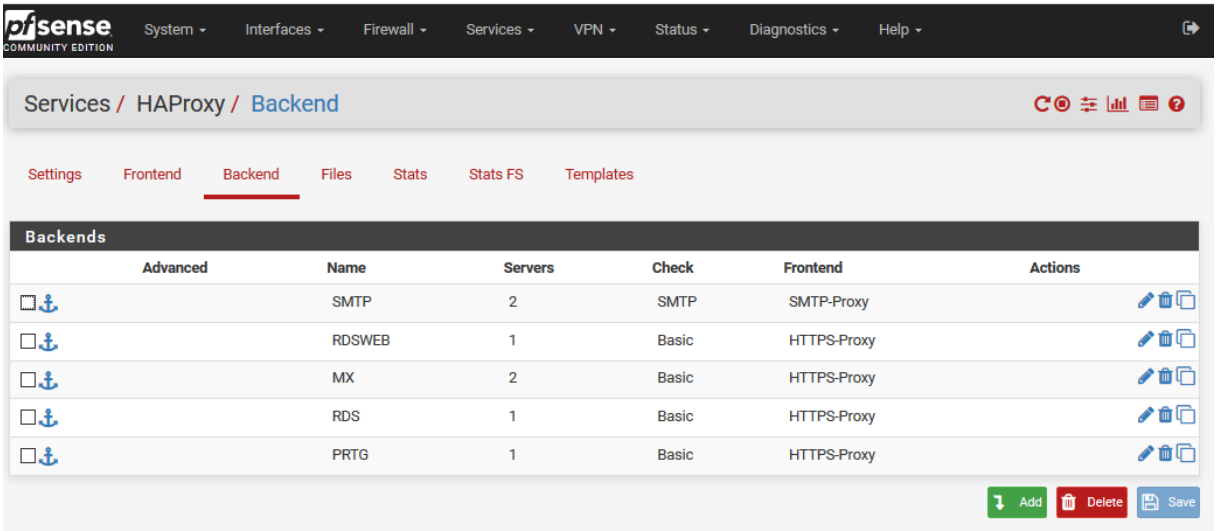
Die Konfiguration muss in der PFSense bestätigt werden:

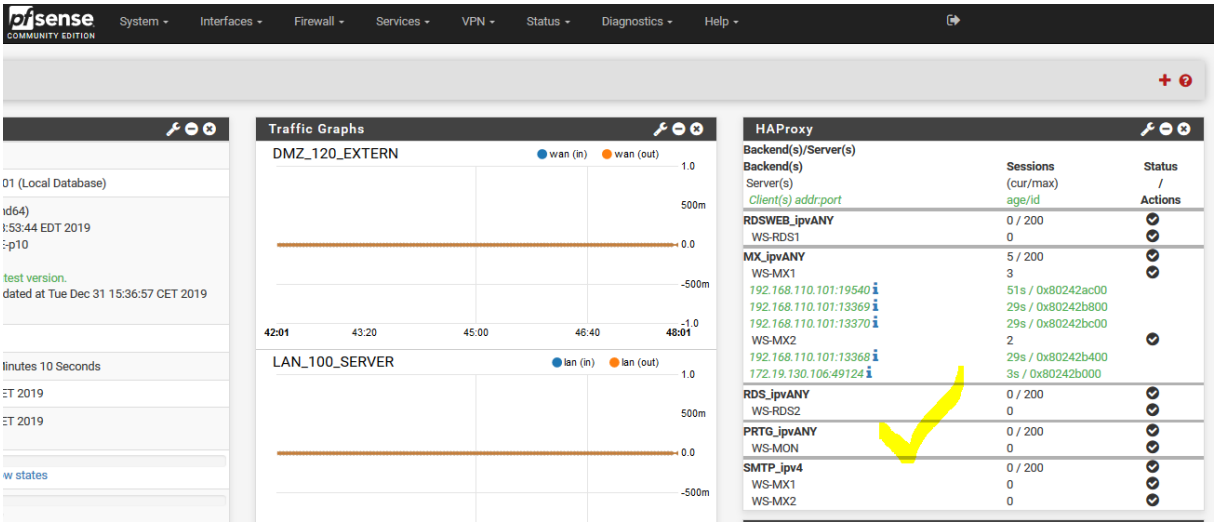


Nun ist das HAProxy-Backend frei und kann ebenfalls gelöscht werden:



Und dann ist auch hier nichts mehr vom WAP über:




pfsense COMMUNITY EDITION
System
Interfaces
Firewall
Services
VPN
Status
Diagnostics
Help
01 (Local Database)
test version.
dated at Tue Dec 31 15:36:57 CET 2019
linutes 10 Seconds
w states
Traffic Graphs
DMZ_120_EXTERN
wan (in)
wan (out)
1.0
500m
0.0
-500m
-1.0
42:01
43:20
45:00
46:40
48:01
LAN_100_SERVER
lan (in)
lan (out)
HAProxy
Backend(s)/Server(s)
Backend(s)
Server(s)
Client(s) addr:port
Sessions
(cur/max)
age/id
Status
/
Actions
RDSWEB_ipvANY 0 / 200
WS-RDS1 0
MX_ipvANY 5 / 200
WS-MX1 3
192.168.110.101:19540 51s / 0x80242ac00
192.168.110.101:13369 29s / 0x80242b800
192.168.110.101:13370 29s / 0x80242bc00
WS-MX2 2
192.168.110.101:13368 29s / 0x80242b400
172.19.130.106:49124 3s / 0x80242b000
RDS_ipvANY 0 / 200
WS-RDS2 0
PRTG_ipvANY 0 / 200
WS-MON 0
SMTP_ipv4 0 / 200
WS-MX1 0
WS-MX2 0