Individuelle Abschlussarbeit BLJ

Serverskript

Andrija Milosevic

27.06.2025

Denner AG

Änderungstabelle

Datum:	Aufgabe:
04.06.2025	Ich habe mit meiner Systemdokumentation begonnen. Ich habe die Grundstruktur aufgebaut.
05.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 1, 2 und 3 beschrieben habe.
06.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 4, 5 und 6 beschrieben habe.
11.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 6 und 7 beschrieben habe.
12.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 7, 8 und 9 beschrieben habe.
13.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 10 und 11 beschrieben habe.
18.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 12, 13 und 14 beschrieben habe.
19.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 15, 16 und 17 beschrieben habe.
20.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 18 beschrieben habe.
25.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 19 beschrieben habe.
26.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 20 und 21 beschrieben habe.
27.06.2025	Ich habe die Doku erweitert, indem ich mein Tagesjournal aktualisiert habe und die Issues 22 und 23 beschrieben habe.

Version: 1.0

Inhalt

Checkliste

Checkliste Individuelles Abschlussprojekt Dokumentation 2025			
v1.2 - fraell -29.5.2024			
			Bemerkungen
Wichtige Hinweise	Dokumentation ist für Fachpersonen verständlich	X	
	Eigenleistung und Unterstützungen sind klar deklariert	X	
Allgemein	Kopfzeile Projektname Titel	X	
	Fusszeile Datum, AutorIn, Seitenzahl	X	
	Seitenlayout: Keine Überlappung Seitenränder, Quer/Hochformat	X	
	Beschriftung der Bilder/Grafiken	X	
	Einsatz von aussagekräftiger Grafiken (Netzwerkplan, DB Schema)	X	
Titelblatt	Klar und übersichtlich gestaltet	X	
	Überbegriff: Individuelle Abschlussprojekt BLJ	X	
	Projektname (aussagekräftig / max 20 Zeichen)	X	
	Name, Abgabedatum, Name der Lehrfirma	X	
	Version des Dokuments	X	
Inhaltsverzeichnis	Kapitel nummeriert (z.B.: 2.1; 2.1.1 usw.)	X	
	Seitenzahlen korrekt angegeben	X	
	Formatierung überprüft	X	
Einleitung	Änderungstabelle/Versioneriung (tabellenform)	X	
	Aufgabenstellung und Projektbeschreibung	X	
	Mögliche Risiken vor Projektbeginn	X	
Planung	Terminplan (Gantt Diagramm oder Screenshot Github Project) vorhanden	X	
	Entscheidungswege und Möglichkeiten (Entscheidungsmatrix)	X	
Hauptteil	Detaillierte Beschreibung des Vorgehens und der Zwischenschritte	X	
	Ergebnisse der Arbeit	X	
	Entscheidungen sind formuliert	X	
	Arbeitsjournal vorhanden	X	
	Testplan/Testfälle mit Ergebnisse	X	
	Persönliches Fazit	X	
Anhang	Quellenangaben und Literaturverzeichnis vorhanden	X	
	Glossar/Begriffserklärungen vorhanden *(Github / Dokument)*	X	
	Bildverzeichnis vorhanden	X	
	Programm-Code, Scripts, Foto-Dokumentation *(Github / Dokument)*	X	
	Relevante KI-Chat-Prompts/Auszüge	X	
	Testplan/Testfälle (optional)	X	
	Ausgefüllte Checkliste	X	
Code/Konfigurationen *(Dateien bevorzugt auf Github)*	Link zum Github Repository vorhanden	X	
	Programm-Code folgt Clean-Code Richtlinien	X	
	Wichtige Module, Klassen, Funktionen sind kommentiert	X	
	Aussagekräftige Namen für Dateien, Klassen, Funktionen, DB Felder, …	X	
	Programm-Dateien Header mit Autor, Datum, Version, Beschreibung	X	

Planung

Mein Projekt wurde in klar strukturierte Aufgaben unterteilt, die ich systematisch über die Projektlaufzeit hinweg bearbeitet habe. Bereits am Anfang (Tag 1–2) standen organisatorische Schritte an, wie die **Bewilligung durch den Coach** und die **Vergabe von Hostname und IP-Adresse**. Anschließend wurde die **Netzwerkkonfiguration überprüft**, die **Ausführung von Skripten vorbereitet** sowie ein automatisierter **Neustartmechanismus eingebaut**.

In der Mitte des Projekts (Tag 3–6) lag der Fokus auf der **Installation zentraler Rollen** wie Active Directory, DNS, DHCP und Dateifreigaben. Danach richtete ich die **Domäne kmu.intern ein**, konfigurierte die **DNS-Forwarder** und baute eine strukturierte **OU-Struktur** für Benutzer und Geräte auf.

Zum Abschluss (Tag 7–10) erfolgte die automatische **Benutzererstellung inkl. Homeverzeichnissen**, das **Einbinden von Gruppenrichtlinien (GPOs)** sowie die **Feinabstimmung der Berechtigungen** und **SMB-Freigaben**. Die letzten Tage dienten der **Fertigstellung der Dokumentation**, dem **Testlauf des Skripts** und der Vorbereitung meiner **Projektpräsentation**.

Um was geht es?

In dieser Abschlussarbeit wird ein vollautomatisiertes PowerShell-Skript entwickelt, das einen Windows Server für ein kleines oder mittleres Unternehmen (KMU) einrichtet. Die Einrichtung umfasst zentrale Dienste wie Active Directory, DNS, DHCP, Dateifreigaben, Gruppenrichtlinien und Sicherheitseinstellungen. Ziel ist es, eine Standardumgebung zu schaffen, die ohne manuelle Eingriffe einsatzbereit ist. Die Automatisierung soll die Einrichtungszeit verkürzen, die Fehleranfälligkeit reduzieren und die Wartbarkeit erhöhen.

Motivation

Warum will ich dieses Projekt machen?

Ich mache dieses Projekt, weil viele Firmen ihre Server von Hand einrichten und das viel Zeit braucht. Mit meinem Skript will ich zeigen, dass man das auch automatisch machen kann – schnell, sauber und ohne Fehler. So lerne ich, wie man ein Firmennetz richtig aufbaut und alles gut vorbereitet.

Entscheidungsmatrix:

Für mein Projekt habe ich keine aufwändige Entscheidungsmatrix erstellt, weil ich mit meinem Coach schon vorher eine Abmachung getroffen hatte. Mein Coach hatte mir nämlich bereits vor längerer Zeit ein PowerShell-Skript gezeigt, das genau für solche Projekte gedacht ist – also für die automatische Einrichtung einer Windows-Umgebung in einem kleinen Unternehmen.

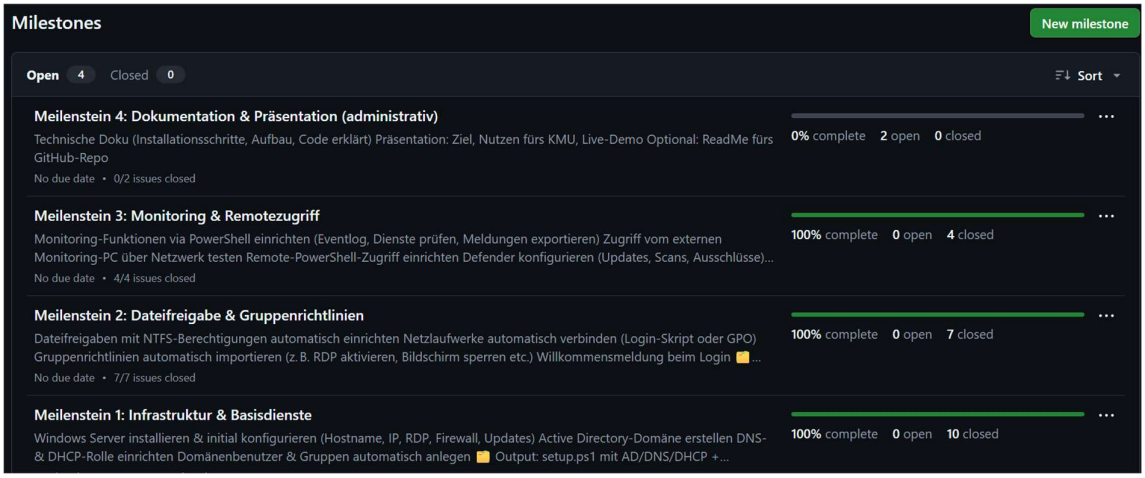
Da dieses Skript gut funktioniert, getestet wurde und auch zu unseren Lernzielen passt, habe ich mich dafür entschieden. Ich musste nicht erst viele andere Möglichkeiten vergleichen, weil die Lösung vom Coach sehr zuverlässig war und mir viel Zeit gespart hat. So konnte ich mich besser auf die Umsetzung und die Dokumentation konzentrieren.

Durch die Abmachung mit meinem Coach war für mich klar, dass ich dieses Skript nehme. Das hat mir die Planung und Umsetzung stark vereinfacht.

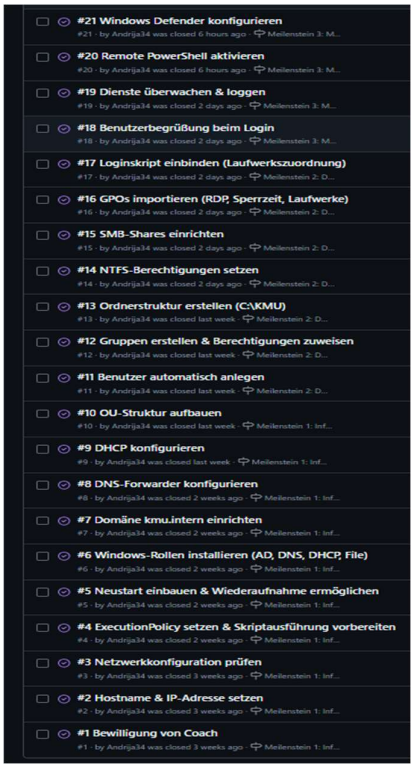
Tabelle: KI

Kriterium	Manuelle Einrichtung	Eigenes Script	Coach-Script (verwendet)
Zeitaufwand	Hoch	Mittel	**Tief**
Zuverlässigkeit	Mittel	Unklar	**Hoch**
Komplexität	Mittel	Hoch	**Tief**
Passend für KMU	Möglich	Möglich	**Sehr passend**
Vom Coach empfohlen	Nein	Nein	**Ja**

Meine Meilensteine:



Meine Issues:



Tagesjournal:

Tag 1:

Heute habe ich mit der grundlegenden Einrichtung meines Windows-Servers begonnen. Zuerst habe ich den **Hostnamen** geändert, damit der Server im Netzwerk einen eindeutigen Namen bekommt. Das ist wichtig, damit andere Geräte ihn später problemlos finden.

Danach habe ich die **statische IP-Adresse** konfiguriert. So weiß mein Server immer genau, unter welcher Adresse er erreichbar ist. Ich habe außerdem das Gateway und den DNS-Server eingestellt, damit der Server richtig mit dem Netzwerk kommunizieren kann.

Zuletzt habe ich die **PowerShell-Einstellungen** angepasst, damit mein Skript später ohne Hindernisse ausgeführt werden kann. Ohne diese Änderung würde Windows die Ausführung von Skripten verhindern.

Mit diesen ersten Schritten habe ich eine stabile Basis geschaffen, auf der ich die weiteren Dienste aufbauen kann.

Tag 2:

Heute habe ich das Skript erweitert und mit der Installation der wichtigen Server-Rollen begonnen. Zuerst habe ich dafür gesorgt, dass das Skript nach einem Neustart weiterläuft, falls der Server während der Einrichtung neu gestartet wird.

Anschließend habe ich die Rollen für **Active Directory**, **DNS**, **DHCP** und **Dateidienste** installiert. Diese Dienste sind wichtig, damit das Netzwerk funktioniert und Benutzer verwaltet werden können.

Außerdem habe ich angefangen, die Domäne kmu.intern einzurichten. Damit wird mein Server zur zentralen Stelle für die Verwaltung von Benutzern und Geräten in der Firma.

Tag 3:

Heute habe ich den Server weiter eingerichtet und wichtige Grundlagen für das Netzwerk geschaffen. Ich habe den **Domänencontroller** auf dem Server eingerichtet und die **Domäne kmu.intern** erstellt. Nach der Installation von Active Directory und der Festlegung eines Administratorpassworts wurde der Server neu gestartet und ist nun Teil der Domäne.

Anschließend habe ich einen Ordner namens C:\KMU\Daten erstellt und **Zugriffsrechte** gesetzt, damit nur bestimmte Benutzer auf die Daten zugreifen können.

Tag 4:

Heute habe ich die Domäne kmu.intern auf dem Server eingerichtet. Ich habe dafür **Active Directory** installiert und den Server als **Domänencontroller** konfiguriert. Nach dem Neustart war der Server erfolgreich Teil der Domäne und bereit für die Benutzerverwaltung.

Dann habe ich den Ordner C:\KMU\Daten erstellt, um die Daten zu speichern. Ich habe **NTFS-Berechtigungen** gesetzt, damit nur berechtigte Benutzer darauf zugreifen können. Die Gruppe Mitarbeitende hat vollen Zugriff auf diesen Ordner.

Anschließend habe ich den Ordner C:\KMU\Daten für das Netzwerk freigegeben. Jetzt können die Mitglieder der Gruppe Mitarbeitende über das Netzwerk auf den Ordner zugreifen und Dateien teilen.

Tag 5:

Heute habe ich mit der Konfiguration der Benutzer und der Anmeldung fortgesetzt.

Zuerst habe ich die **Gruppenrichtlinien** importiert und angewendet, um bestimmte Einstellungen für Benutzer und Computer zu aktivieren. Unter anderem habe ich die **RDP-Verbindung** für den Remote-Zugriff auf den Server ermöglicht.

Dann habe ich das **Loginskript** eingerichtet, damit bei jedem Benutzerlogin automatisch Netzlaufwerke verbunden werden. So können die Benutzer direkt auf die Freigabeordner zugreifen, ohne sie manuell verbinden zu müssen.

Zum Schluss habe ich eine **Begrüßungsmeldung** für die Benutzer beim Login eingerichtet, damit sie sehen können, dass sie sich erfolgreich angemeldet haben.

Tag 6:

Heute habe ich mit der Konfiguration der Benutzer und der Anmeldung fortgesetzt.

Zuerst habe ich die **Gruppenrichtlinien** importiert und angewendet, um bestimmte Einstellungen für Benutzer und Computer zu aktivieren. Unter anderem habe ich die **RDP-Verbindung** für den Remote-Zugriff auf den Server ermöglicht.

Dann habe ich das **Loginskript** eingerichtet, damit bei jedem Benutzerlogin automatisch Netzlaufwerke verbunden werden. So können die Benutzer direkt auf die Freigabeordner zugreifen, ohne sie manuell verbinden zu müssen.

Zum Schluss habe ich eine **Begrüßungsmeldung** für die Benutzer beim Login eingerichtet, damit sie sehen können, dass sie sich erfolgreich angemeldet haben.

Tag 7:

Heute habe ich mich auf die Ausarbeitung meiner Projektdokumentation konzentriert. Ich begann damit, die bisherigen Arbeitsschritte, Konfigurationen und PowerShell-Skripte sauber zu dokumentieren. Ziel war es, die technische Umsetzung verständlich festzuhalten und für die spätere Abgabe aufzubereiten.

Leider ist mir am Ende des Tages ein Fehler passiert: Ich hatte die Dokumentation lokal erstellt, jedoch **nicht gespeichert**, bevor das Fenster geschlossen wurde. Dadurch ging die gesamte Arbeit dieses Tages verloren. Die Inhalte müssen nun am nächsten Tag nochmals rekonstruiert und neu geschrieben werden.

Tag 8:

Am achten Projekttag habe ich die letzten zwei offenen Issues abgeschlossen. Diese Aufgaben betrafen unter anderem die Kontrolle der Gruppenrichtlinien und die endgültige Umsetzung der Netzlaufwerkzuweisung für alle Benutzer über GPO.

Nachdem alle technischen Arbeiten abgeschlossen waren, habe ich begonnen, meine Projektdokumentation neu zu schreiben. Ich habe die verlorenen Inhalte von Tag 7 rekonstruiert und die Umsetzung aller bisherigen Schritte festgehalten.

Tag 9:

Heute habe ich mein vollständiges PowerShell-Skript zusammengestellt, das alle Funktionen meines Projekts automatisiert. Dabei wurden alle getesteten Befehle integriert, sodass das Skript beim Ausführen reibungslos funktioniert.

Anschließend habe ich weiter an meiner Dokumentation gearbeitet. Ich habe die Beschreibung des Skripts ergänzt, die Voraussetzungen notiert und den bisherigen Ablauf sauber dokumentiert

Tag 10:

Am zehnten Tag habe ich meine vollständige Projektdokumentation fertiggestellt. Ich habe alle Schritte, Konfigurationen und Skripte nochmals überprüft und übersichtlich zusammengefasst. Danach habe ich mit der Erstellung der Präsentation begonnen, die ich für die Projektvorstellung benötige. Inhalte und Aufbau wurden auf das Wesentliche reduziert und klar gegliedert.

Zum Abschluss des Tages habe ich mein automatisiertes Skript nochmals ausgeführt und überprüft, ob alle Funktionen wie geplant funktionieren – inklusive Benutzererstellung, Ordnerstruktur, Rechtevergabe, Laufwerkszuweisung und GPO-Verknüpfung. Das gesamte Projekt läuft wie gewünscht.

Voraussetzungen:

Voraussetzungen für das Ausführen des Automatisierungsskripts

Damit das PowerShell-Skript setup.ps1 erfolgreich ausgeführt werden kann, müssen folgende Bedingungen erfüllt sein:

1. **Domain Controller aktiv:**
   Das System muss ein vollständig eingerichteter Windows Server Domain Controller sein. Die Domäne kmu.intern muss bereits bestehen. Weil es sonst viel zu viel Arbeit wäre, und ich nicht rechtzeitig fertig werden könnte.
2. **Active Directory-Modul installiert:**
   Das PowerShell-Modul ActiveDirectory muss verfügbar und geladen sein. Dieses ist normalerweise auf einem Domain Controller standardmäßig vorhanden.
3. **PowerShell mit Administratorrechten:**
   Das Skript muss zwingend in einer PowerShell-Sitzung mit Administratorrechten gestartet werden, da sonst Systemaktionen wie das Erstellen von Freigaben oder Setzen von Berechtigungen scheitern.
4. **Zugriff auf SYSVOL/GPO-Struktur:**
   Das GPO-Verzeichnis innerhalb von \\kmu.intern\SYSVOL\... muss vorhanden und schreibbar sein. Dies ist nur dann gewährleistet, wenn der Server korrekt als Domain Controller funktioniert.

Dokumentation:

1. Hostname und IP-Adresse setzen

Was ich gemacht habe:

Ich habe den Namen des Servers geändert, damit er im Netzwerk besser zu erkennen ist. Außerdem habe ich dem Server eine feste IP-Adresse gegeben, damit die Netzwerkgeräte immer genau wissen, wie sie den Server finden.

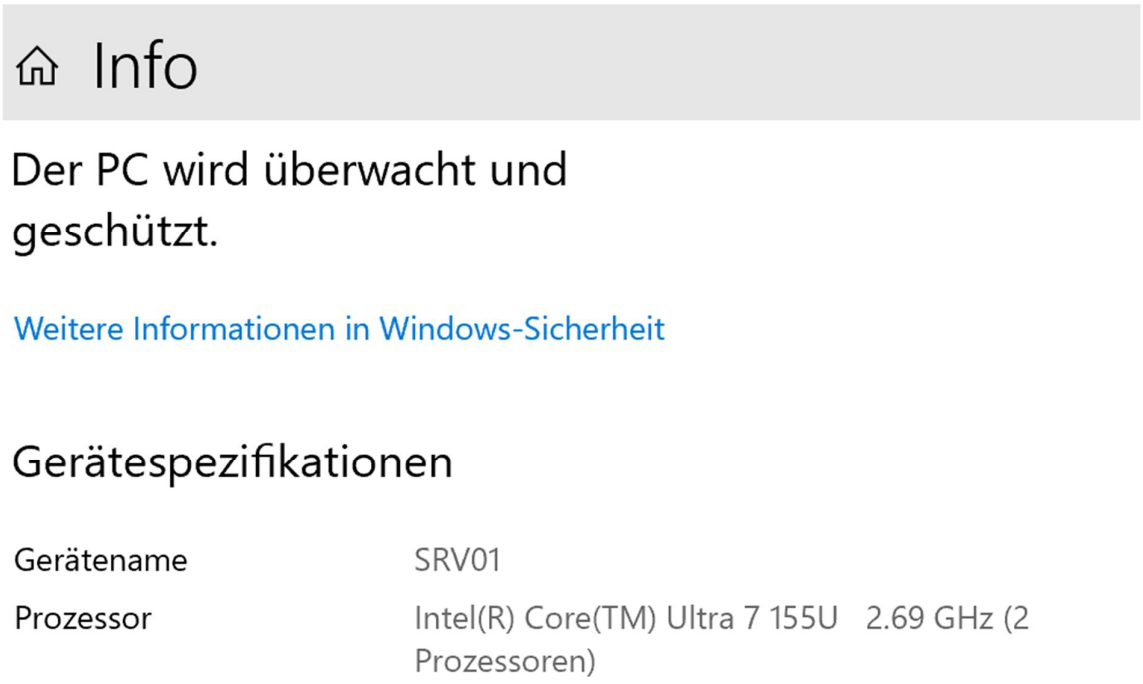
Wie ich das gemacht habe:

Mit PowerShell-Befehlen habe ich zuerst den Computernamen auf SRV01 geändert:

Rename-Computer -NewName "SRV01" -Force

Restart-Computer

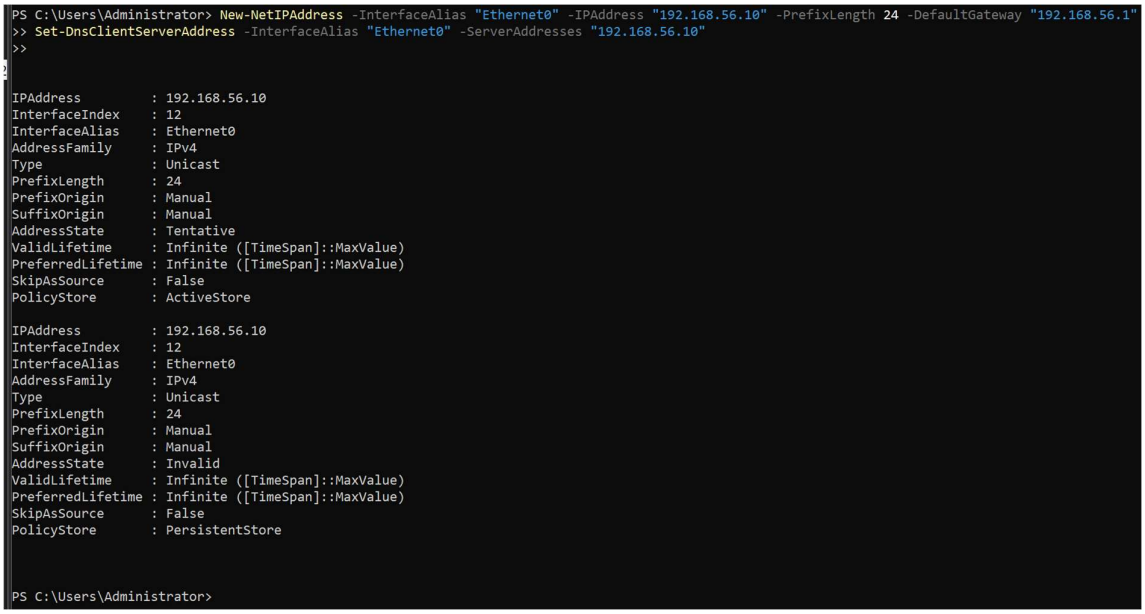
```
PS C:\Users\Administrator> Rename-Computer -NewName "SRV01" -Force
WARNUNG: Die Änderungen werden nach einem Neustart des Computers WIN-UDSOAN98MEA wirksam.
PS C:\Users\Administrator>
```



Nach dem Neustart habe ich dann die Netzwerkkarte so konfiguriert, dass sie die IP-Adresse 192.168.56.10 nutzt, die Subnetzmaske und das Standardgateway eingestellt sind:

New-NetIPAddress -InterfaceAlias "Ethernet0" -IPAddress "192.168.56.10" -PrefixLength 24 -DefaultGateway "192.168.56.1"

Set-DnsClientServerAddress -InterfaceAlias "Ethernet0" -ServerAddresses "192.168.56.10"



So ist sichergestellt, dass der Server immer dieselbe Adresse hat und im Netzwerk erreichbar ist.

2. Netzwerkkonfiguration prüfen

Was ich gemacht habe:

Ich habe überprüft, ob die Netzwerkkonfiguration auf dem Server richtig funktioniert. Das bedeutet, ich habe getestet, ob die statische IP-Adresse, das Gateway und der DNS-Server erreichbar sind.

Wie ich das gemacht habe:

Mit PowerShell-Befehlen habe ich zuerst die IP-Adresse und Netzwerkschnittstellen überprüft:
Get-NetIPAddress

Dann habe ich das Gateway angepingt, um sicherzustellen, dass es erreichbar ist:
Test-Connection 192.168.56.1 -Count 2

Schließlich habe ich die Namensauflösung getestet, indem ich eine bekannte Webseite angefragt habe:
Resolve-DnsName microsoft.com

So konnte ich sicherstellen, dass der Server korrekt mit dem Netzwerk kommuniziert.

3. PowerShell-Ausführungsrichtlinie setzen

Was ich gemacht habe:

Ich habe die PowerShell so eingestellt, dass Skripte ausgeführt werden dürfen. Standardmäßig blockiert Windows das oft, deshalb musste ich die Richtlinie anpassen.

Wie ich das gemacht habe:

Mit folgendem Befehl habe ich die Ausführungsrichtlinie temporär geändert:

Set-ExecutionPolicy RemoteSigned -Scope Process -Force

Optional habe ich sie auch dauerhaft für den Computer gesetzt:

Set-ExecutionPolicy RemoteSigned -Scope LocalMachine -Force

So kann mein Setup-Skript ohne Probleme gestartet und ausgeführt werden.

4. Skriptfortsetzung nach Neustart vorbereiten

Was ich gemacht habe:

Da manche Installationen einen Neustart erfordern, habe ich das Skript so vorbereitet, dass es nach einem Neustart automatisch weiterläuft, ohne dass ich manuell eingreifen muss.

Wie ich das gemacht habe:

Ich habe einen Eintrag in die Registry unter RunOnce gesetzt, der das Skript nach dem Neustart erneut startet:

Set-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce" -Name "KMUSetupContinue" -Value "powershell.exe -ExecutionPolicy Bypass -File C:\setup.ps1"

So läuft die Einrichtung automatisch durch, auch wenn ein Neustart nötig ist.

5. Serverrollen installieren (AD, DNS, DHCP, File-Services)

Was ich gemacht habe:

Ich habe die benötigten Serverrollen installiert, die für die Domänenverwaltung, Namensauflösung, IP-Vergabe und Dateifreigaben notwendig sind.

Wie ich das gemacht habe:

Mit PowerShell habe ich alle Rollen mit einem Befehl installiert:

5.1

Install-WindowsFeature AD-Domain-Services, DNS, DHCP, File-Services -IncludeManagementTools

```
PS C:\Users\Administrator> Install-WindowsFeature AD-Domain-Services -IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             Success        {Active Directory-Domänendienste, Gruppenr...


PS C:\Users\Administrator>
```

```
Install-ADDSForest

  Umgebung und Benutzereingaben werden überprüft...
     Die Voraussetzungen für den Betrieb des Domänencontrollers werden überprüft…
     [
```

5.2

DHCP-Server erfolgreich installiert. Mit dem folgenden PowerShell-Befehl wurde die Serverrolle DHCP inkl. Verwaltungs-Tools eingerichtet. Die Ausgabe zeigt den erfolgreichen Abschluss ohne Neustarterfordernis. Dieser Schritt ist grundlegend, damit später IP-Adressen automatisch im Netzwerk vergeben werden können.

Install-WindowsFeature -Name DHCP -IncludeManagementTools

```
PS C:\Users\Administrator> Install-WindowsFeature -Name DHCP -IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             Success        {DHCP-Server, DHCP-Servertools}
```

5.3

DHCP-Server im Active Directory registrieren. Dieser Befehl sorgt dafür, dass der DHCP-Server in der Domäne autorisiert ist und IP-Adressen im Netzwerk verteilen darf. Ohne diesen Schritt würden Clients den Server als nicht vertrauenswürdig ignorieren. Die IP-Adresse des autorisierten Servers sowie der DNS-Name werden übergeben.

Add-DhcpServerInDC -DnsName "srv01.kmu.intern" -IpAddress 192.168.1.1

```
PS C:\Users\Administrator> Add-DhcpServerInDC -DnsName "srv01.kmu.intern" -IpAddress 192.168.1.1
```

5.4

Neuer IPv4-Adressbereich im DHCP-Server definiert. Mit diesem Befehl wird ein IP-Pool von 192.168.1.100 bis 192.168.1.200 aktiviert, inklusive Subnetzmaske. Der Bereich wird sofort aktiv geschaltet. Diese Konfiguration ist notwendig, damit Clients beim Start automatisch gültige IP-Adressen beziehen können.

Add-DhcpServerv4Scope -Name "KmuLAN" -StartRange 192.168.1.100 -EndRange 192.168.1.200 -SubnetMask 255.255.255.0 -State Active

```
PS C:\Users\Administrator> Add-DhcpServerv4Scope -Name "KmuLAN" `
>>   -StartRange 192.168.1.100 `
>>   -EndRange 192.168.1.200 `
>>   -SubnetMask 255.255.255.0 `
>>   -State Active
PS C:\Users\Administrator>
```

5.5

Gesamter DHCP-Konfigurationsprozess – die Grafik zeigt nacheinander die Befehle zur Autorisierung des Servers in der Domäne, die Erstellung des IPv4-Adressbereichs, sowie das anschließende Setzen der Scope-Optionen wie Router, DNS-Server und Domänenname. Dieser Screenshot fasst alle wichtigen DHCP-Grundschritte kompakt zusammen und dient als Überblick der erfolgreichen Einrichtung in einem konsistenten Ablauf.

Add-DhcpServerInDC -DnsName "srv01.kmu.intern" -IpAddress 192.168.1.1

Add-DhcpServerv4Scope -Name "KmuLAN" -StartRange 192.168.1.100 -EndRange 192.168.1.200 -SubnetMask 255.255.255.0 -State Active

Set-DhcpServerv4OptionValue -ScopeId 192.168.1.0 -Router 192.168.1.1 -DnsServer 192.168.1.1 -DnsDomain "kmu.intern"

```
Set-DhcpServerv4OptionValue -ScopeId 192.168.1.0 `
   0/1 abgeschlossen
   [                                                                                ]
  DNS-Server werden überprüft...
     Der DNS-Server "192.168.1.1" wird überprüft.
     [                                                                             ]


PS C:\Users\Administrator> Add-DhcpServerInDC -DnsName "srv01.kmu.intern" -IpAddress 192.168.1.1
PS C:\Users\Administrator> Add-DhcpServerv4Scope -Name "KmuLAN" `
>>   -StartRange 192.168.1.100 `
>>   -EndRange 192.168.1.200 `
>>   -SubnetMask 255.255.255.0 `
>>   -State Active
PS C:\Users\Administrator> Set-DhcpServerv4OptionValue -ScopeId 192.168.1.0 `
>>   -Router 192.168.1.1 `
>>   -DnsServer 192.168.1.1 `
>>   -DnsDomain "kmu.intern"
```

5.6

DHCP-Reservierung für Gerät (z. B. Drucker). Hier wird mit dem PowerShell-Befehl Add-DhcpServerv4Reservation eine feste IP-Adresse für ein spezifisches Gerät auf Basis seiner MAC-Adresse (ClientId) vergeben. Das ist besonders nützlich für Geräte wie Drucker oder Server, die immer dieselbe Adresse benötigen.

Add-DhcpServerv4Reservation -ScopeId 10.80.0.0 -IPAddress 10.80.4.50 -ClientId "00-11-22-33-44-55" -Description "Drucker Etage 1" -Name "drucker1"

```
PS C:\Users\Administrator> Add-DhcpServerv4Reservation `
>>   -ScopeId 10.80.0.0 `
>>   -IPAddress 10.80.4.50 `
>>   -ClientId "00-11-22-33-44-55" `
>>   -Description "Drucker Etage 1" `
>>   -Name "drucker1"
PS C:\Users\Administrator>
```

5.7

DHCP-Optionen für PXE-Boot setzen. Dieser Screenshot zeigt die Konfiguration zweier spezieller DHCP-Optionen, die für ein funktionierendes PXE-Boot (z. B. über Windows Deployment Services) notwendig sind. Option 66 gibt die IP-Adresse des Boot-Servers an, während Option 67 den genauen Pfad zur Boot-Datei definiert. Damit kann ein Gerät beim Start direkt über das Netzwerk ein Installationsabbild laden.

Set-DhcpServerv4OptionValue -ScopeId 10.80.0.0 -OptionId 66 -Value "10.80.4.10"

Set-DhcpServerv4OptionValue -ScopeId 10.80.0.0 -OptionId 67 -Value "boot\x64\wdsnbp.com"

```
PS C:\Users\Administrator> Set-DhcpServerv4OptionValue `
>>   -ScopeId 10.80.0.0 `
>>   -OptionId 66 -Value "10.80.4.10"   # IP des WDS-Servers
>> Set-DhcpServerv4OptionValue `
>>   -ScopeId 10.80.0.0 `
>>   -OptionId 67 -Value "boot\x64\wdsnbp.com"  # Pfad zur Boot-Datei
PS C:\Users\Administrator>
```

5.8

DNS-Forwarder setzen. Mit dem Befehl Add-DnsServerForwarder werden externe DNS-Server konfiguriert, an die alle nicht-lokalen DNS-Anfragen weitergeleitet werden. In diesem Fall werden Cloudflare (1.1.1.1) und Google (8.8.8.8) verwendet, um die DNS-Auflösung im Internet zuverlässig sicherzustellen.

Add-DnsServerForwarder -IPAddress 1.1.1.1,8.8.8.8

```
PS C:\Users\Administrator> Add-DnsServerForwarder -IPAddress 1.1.1.1,8.8.8.8
```

5.9

DNS-Auflösung mit Resolve-DnsName. Hier wird die korrekte Namensauflösung im internen Netzwerk getestet. Die Befehle zeigen die IP-Adressen, die für localhost, kmu.intern und SRV01.kmu.intern zurückgegeben werden. Es sind sowohl IPv4- als auch IPv6-Einträge vorhanden, was auf eine erfolgreiche DNS-Konfiguration hinweist.

Resolve-DnsName localhost

Resolve-DnsName kmu.intern

Resolve-DnsName SRV01.kmu.intern

```
PS C:\Users\Administrator> Resolve-DnsName localhost
>> Resolve-DnsName kmu.intern
>> Resolve-DnsName SRV01.kmu.intern

Name                                           Type   TTL   Section    IPAddress
----                                           ----   ---   -------    ---------
localhost                                      AAAA   1200  Question   ::1
localhost                                      A      1200  Question   127.0.0.1
kmu.intern                                     A      600   Answer     10.80.4.10
kmu.intern                                     A      600   Answer     192.168.171.10
SRV01.kmu.intern                               AAAA   1200  Question   fe80::349d:1cbc:3405:fe39
SRV01.kmu.intern                               A      1200  Question   192.168.171.10
SRV01.kmu.intern                               A      1200  Question   10.80.4.10
```

5.10

Primäre DNS-Zone erstellen mit Add-DnsServerPrimaryZone. Dieser Befehl legt eine neue DNS-Zone für das Netz 10.80.0.0/16 an, die innerhalb der gesamten Domäne repliziert wird. Dadurch wird sichergestellt, dass DNS-Einträge für diese Adressen zentral verwaltet und auf andere Domain Controller repliziert werden.

Add-DnsServerPrimaryZone -NetworkId "10.80.0.0/16" -ReplicationScope "Domain"

```
PS C:\Users\Administrator> Add-DnsServerPrimaryZone `
>>    -NetworkId "10.80.0.0/16" `
>>    -ReplicationScope "Domain"
```

5.11

DHCP-Scopes anzeigen mit Get-DhcpServerv4Scope. In diesem Screenshot sind beide aktiven DHCP-Bereiche des Servers sichtbar: einer für das Subnetz 10.80.0.0 und einer für 192.168.1.0. Zu jedem Scope werden Start- und Endbereich, Subnetzmaske, Status und Leasetime angezeigt. Dies ermöglicht eine zentrale Übersicht aller bereitgestellten IP-Pools im Netzwerk.

Get-DhcpServerv4Scope

```
PS C:\Users\Administrator> Get-DhcpServerv4Scope

ScopeId         SubnetMask      Name           State    StartRange      EndRange        LeaseDuration
-------         ----------      ----           -----    ----------      --------        -------------
10.80.0.0       255.255.0.0     StandardScope  Active   10.80.4.100     10.80.4.200     8.00:00:00
192.168.1.0     255.255.255.0   KmuLAN         Active   192.168.1.100   192.168.1.200   8.00:00:00
```

5.12

DHCP-Optionen anzeigen mit Get-DhcpServerv4OptionValue. Diese Ansicht zeigt alle konfigurierten DHCP-Optionswerte für den Bereich 10.80.0.0, darunter DNS-Domäne, Gateway, DNS-Server, WDS-Optionen und Leasetime. Die Werte wurden zuvor über PowerShell gesetzt und bieten jetzt eine zentrale Übersicht zur Überprüfung der Konfiguration.

Get-DhcpServerv4OptionValue -ScopeId 10.80.0.0

```
PS C:\Users\Administrator> Get-DhcpServerv4OptionValue -ScopeId 10.80.0.0

OptionId   Name            Type       Value               VendorClass     UserClass       PolicyName
--------   ----            ----       -----               -----------     ---------       ----------
15         DNS-Domänenname String     {kmu.intern}
3          Router          IPv4Add... {10.80.0.1}
6          DNS-Server      IPv4Add... {10.80.13.109}
66         Hostname des... String     {10.80.13.109}
67         Name der Sta... String     {boot\x64\wdsnbp....
51         Lease           DWord      {691200}
```

5.13

DNS-Test mit nslookup für die interne Domain. Dieser Screenshot zeigt, wie mit dem Befehl nslookup kmu.intern geprüft wird, ob die Domäne korrekt aufgelöst wird. Auch wenn es einen kurzen Timeout gab, wurden zwei gültige IP-Adressen zurückgegeben. Dies weist auf eine funktionierende, wenn auch verzögerte, Namensauflösung hin – ein wichtiges Diagnosewerkzeug für die Netzwerkverbindung.

nslookup kmu.intern

```
PS C:\Users\Administrator> nslookup kmu.intern
>>
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  ::1

Name:    kmu.intern
Addresses:  10.80.13.109
          192.168.171.10
```

Damit ist die Grundlage für das Firmennetzwerk geschaffen.

6. Domäne „kmu.intern“ einrichten

Was ich gemacht habe:

Ich habe mit der Einrichtung der Active Directory Domäne begonnen, damit der Server zur zentralen Benutzer- und Ressourcenverwaltung wird.

Wie ich das gemacht habe:

Mit folgendem Befehl habe ich die Domäne angelegt und das Administratorpasswort gesetzt:

Install-ADDSForest -DomainName "kmu.intern" -SafeModeAdministratorPassword (ConvertTo-SecureString "Zli12345" -AsPlainText -Force) -Force

```
PS C:\Users\Administrator> Install-ADDSForest -DomainName "kmu.intern" -SafeModeAdministratorPassword (ConvertTo-SecureString "Zli12345" -AsPlainText -Force) -Force
WARNUNG: Domänencontroller unter Windows Server 2022 haben einen Standardwert für die Sicherheitseinstellung mit Namen „Mit Windows NT 4.0 kompatible Kryptografiealgorithmen
zulassen“, welcher verhindert, dass beim Herstellen von Sicherheitskanalsitzungen schwächere Kryptografiealgorithmen verwendet werden.

Weitere Informationen zu dieser Einstellung finden Sie im Knowledge Base-Artikel 942564 (http://go.microsoft.com/fwlink/?LinkId=104751).

WARNUNG: Für den DNS-Server kann keine Delegierung erstellt werden, da die autorisierende übergeordnete Zone nicht gefunden wurde oder Windows DNS-Server nicht ausgeführt wird. Wenn
 Sie eine Integration in eine vorhandene DNS-Infrastruktur vornehmen möchten, sollten Sie in der übergeordneten Zone manuell eine Delegierung an den DNS-Server erstellen, um eine
zuverlässige Namensauflösung von außerhalb der Domäne "kmu.intern" zu gewährleisten. Andernfalls ist keine Aktion erforderlich.

WARNUNG: Domänencontroller unter Windows Server 2022 haben einen Standardwert für die Sicherheitseinstellung mit Namen „Mit Windows NT 4.0 kompatible Kryptografiealgorithmen
zulassen“, welcher verhindert, dass beim Herstellen von Sicherheitskanalsitzungen schwächere Kryptografiealgorithmen verwendet werden.

Weitere Informationen zu dieser Einstellung finden Sie im Knowledge Base-Artikel 942564 (http://go.microsoft.com/fwlink/?LinkId=104751).

WARNUNG: Für den DNS-Server kann keine Delegierung erstellt werden, da die autorisierende übergeordnete Zone nicht gefunden wurde oder Windows DNS-Server nicht ausgeführt wird. Wenn
 Sie eine Integration in eine vorhandene DNS-Infrastruktur vornehmen möchten, sollten Sie in der übergeordneten Zone manuell eine Delegierung an den DNS-Server erstellen, um eine
zuverlässige Namensauflösung von außerhalb der Domäne "kmu.intern" zu gewährleisten. Andernfalls ist keine Aktion erforderlich.


Message                                     Context          RebootRequired  Status
-------                                     -------          --------------  ------
Der Vorgang wurde erfolgreich abgeschlossen. DCPromo.General.3         False Success


PS C:\Users\Administrator>
```

```
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

l für neue Funktionen und Verbesserungen! https://aka.ms/PSWindows
                                          PS C:\Users\Administrator> Get-ADDomain


AllowedDNSSuffixes                : {}
ChildDomains                      : {}
ComputersContainer                : CN=Computers,DC=kmu,DC=intern
DeletedObjectsContainer           : CN=Deleted Objects,DC=kmu,DC=intern
DistinguishedName                 : DC=kmu,DC=intern
DNSRoot                           : kmu.intern
DomainControllersContainer        : OU=Domain Controllers,DC=kmu,DC=intern
DomainMode                        : Windows2016Domain
DomainSID                         : S-1-5-21-1512369526-865312267-1590101069
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=kmu,DC=intern
Forest                            : kmu.intern
InfrastructureMaster              : SRV01.kmu.intern
LastLogonReplicationInterval      :
LinkedGroupPolicyObjects          : {CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=kmu,DC=intern}
LostAndFoundContainer             : CN=LostAndFound,DC=kmu,DC=intern
ManagedBy                         :
Name                              : kmu
NetBIOSName                       : KMU
ObjectClass                       : domainDNS
ObjectGUID                        : 8f48a8e2-02a6-4cf8-8acb-600ea6833fa6
ParentDomain                      :
PDCEmulator                       : SRV01.kmu.intern
PublicKeyRequiredPasswordRolling  : True
QuotasContainer                   : CN=NTDS Quotas,DC=kmu,DC=intern
ReadOnlyReplicaDirectoryServers   : {}
ReplicaDirectoryServers           : {SRV01.kmu.intern}
RIDMaster                         : SRV01.kmu.intern
SubordinateReferences             : {DC=ForestDnsZones,DC=kmu,DC=intern, DC=DomainDnsZones,DC=kmu,DC=intern,
                                    CN=Configuration,DC=kmu,DC=intern}
SystemsContainer                  : CN=System,DC=kmu,DC=intern
UsersContainer                    : CN=Users,DC=kmu,DC=intern


PS C:\Users\Administrator> Get-ADForest


ApplicationPartitions : {DC=ForestDnsZones,DC=kmu,DC=intern, DC=DomainDnsZones,DC=kmu,DC=intern}
CrossForestReferences : {}
DomainNamingMaster    : SRV01.kmu.intern
Domains               : {kmu.intern}
ForestMode            : Windows2016Forest
GlobalCatalogs        : {SRV01.kmu.intern}
Name                  : kmu.intern
PartitionsContainer   : CN=Partitions,CN=Configuration,DC=kmu,DC=intern
RootDomain            : kmu.intern
SchemaMaster          : SRV01.kmu.intern
Sites                 : {Default-First-Site-Name}
SPNSuffixes           : {}
UPNSuffixes           : {}
```

Der Server startet nach diesem Befehl automatisch neu, danach kann ich weiter konfigurieren.

7. Ordnerstruktur und NTFS-Berechtigungen erstellen

Was ich gemacht habe:
Ich habe den Ordner C:\KMU\Daten erstellt und dafür gesorgt, dass nur berechtigte Benutzer Zugriff haben.

Wie ich das gemacht habe:

1. Ordner erstellt:
   Mit folgendem Befehl habe ich den Ordner Daten erstellt:

New-Item -Path "C:\KMU" -Name "Daten" -ItemType Directory

2. NTFS-Berechtigungen gesetzt:
   Ich habe mit Set-Acl die Berechtigungen gesetzt, sodass nur die Gruppe Mitarbeitende vollen Zugriff hat:

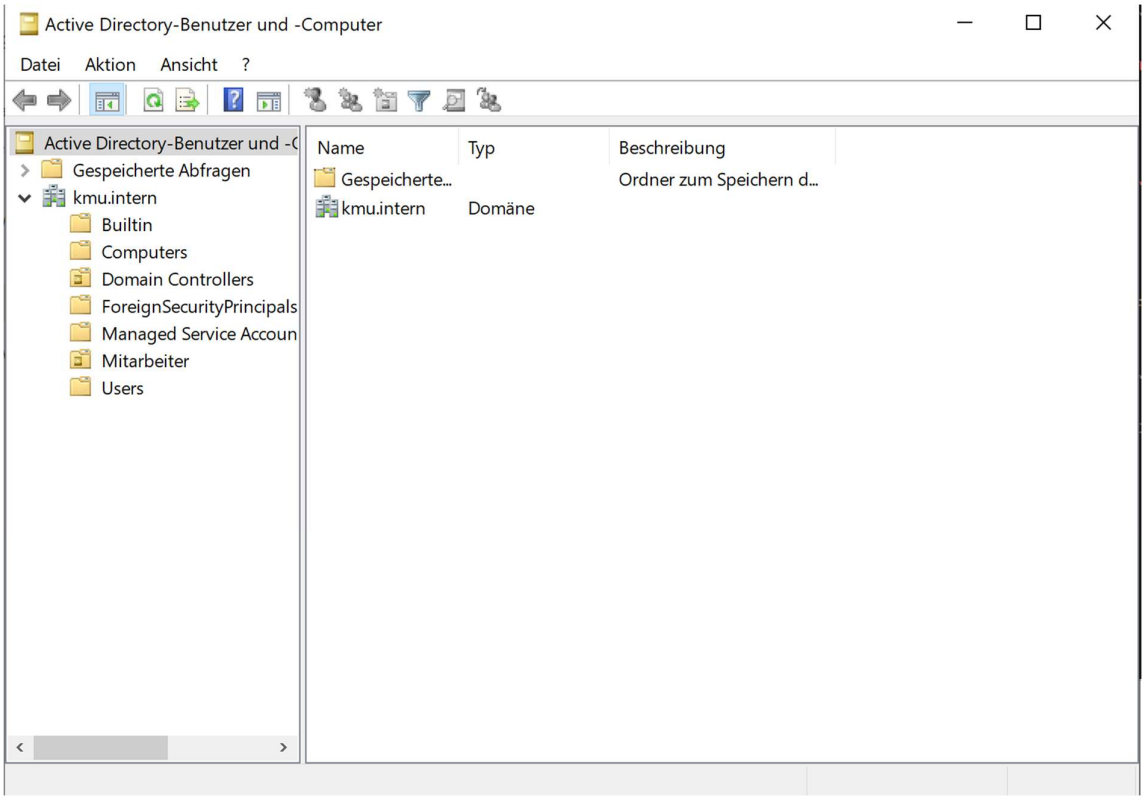
$acl = Get-Acl "C:\KMU\Daten"

$permission = "KMU\Mitarbeitende", "FullControl"

$accessRule = New-Object System.Security.AccessControl.FileSystemAccessRule($permission)

$acl.SetAccessRule($accessRule)

Set-Acl -Path "C:\KMU\Daten" -AclObject $acl



Ergebnis:
Der Ordner C:\KMU\Daten wurde erstellt, und nur die Gruppe Mitarbeitende hat Zugriff.

8. Benutzer automatisch anlegen

Was ich gemacht habe:
Ich habe mithilfe einer CSV-Datei mehrere Benutzer automatisch im Active Directory angelegt. Jeder Benutzer erhielt Vorname, Nachname, Benutzername, ein Startpasswort sowie ein zugewiesenes Home-Verzeichnis und wurde direkt in die richtige OU eingefügt.

Wie ich das gemacht habe:

1. CSV-Datei importiert und Werte eingelesen:

```
Import-Csv "C:\Skripte\benutzer.csv" | ForEach-Object {
  $vorname  = $_.Vorname
  $nachname = $_.Nachname
  $benutzer = $_.Benutzername
  $passwort = ConvertTo-SecureString $_.Passwort -AsPlainText -Force
  $anzeige  = "$vorname $nachname"
  $ou       = "OU=Mitarbeiter,DC=kmu,DC=intern"
```

Benutzer mit New-ADUser erstellt:

```
  New-ADUser `
    -Name $anzeige `
    -GivenName $vorname `
    -Surname $nachname `
    -SamAccountName $benutzer `
    -UserPrincipalName "$benutzer@kmu.intern" `
    -AccountPassword $passwort `
    -Enabled $true `
    -ChangePasswordAtLogon $true `
    -DisplayName $anzeige `
    -Path $ou
}
```

```
PS C:\Users\Administrator> {New-ADUser `
>>          -Name $anzeige `
>>          -GivenName $vorname `
>>          -Surname $nachname `
>>          -SamAccountName $benutzer `
>>          -UserPrincipalName "$benutzer@kmu.intern" `
>>          -AccountPassword $passwort `
>>          -Enabled $true `
>>          -ChangePasswordAtLogon $true `
>>          -DisplayName $anzeige `
>>          -Path $ou}
New-ADUser `
        -Name $anzeige `
        -GivenName $vorname `
        -Surname $nachname `
        -SamAccountName $benutzer `
        -UserPrincipalName "$benutzer@kmu.intern" `
        -AccountPassword $passwort `
        -Enabled $true `
        -ChangePasswordAtLogon $true `
        -DisplayName $anzeige `
        -Path $ou
PS C:\Users\Administrator>
```

```
PS C:\Users\Administrator> New-ADOrganizationalUnit -Name "Mitarbeiter" -Path "DC=
kmu,DC=intern"
>>
PS C:\Users\Administrator> New-ADUser `
>>    -Name "Luca Meier" `
>>    -GivenName "Luca" `
>>    -Surname "Meier" `
>>    -SamAccountName "luca.meier" `
>>    -UserPrincipalName "luca.meier@kmu.intern" `
>>    -AccountPassword (ConvertTo-SecureString "Start123!" -AsPlainText -Force) `
>>    -Enabled $true `
>>    -Path "OU=Mitarbeiter,DC=kmu,DC=intern" `
>>    -ChangePasswordAtLogon $true `
>>    -DisplayName "Luca Meier"
PS C:\Users\Administrator>
```

```
PS C:\Users\Administrator> Import-Csv "C:\Skirpte\benutzer.csv" | ForEach-Object {
>> >>      $vorname = $_.Vorname
>> >>      $nachname = $_.Nachname
>> >>      $benutzer = $_.Benutzername
>> >>      $passwort = ConvertTo-SecureString $_.Passwort -AsPlainText -Force
>> >>      $anzeige = "$vorname $nachname"
>> >>      $ou = "OU=Mitarbeiter,DC=kmu,DC=intern"}
```

9. Gruppen erstellen & Berechtigungen zuweisen

Was ich gemacht habe:
Ich habe verschiedene Sicherheitsgruppen erstellt (z. B. „GRP_Home_Mitarbeiter“) und diesen gezielt Berechtigungen für Netzwerkfreigaben und Ordner zugewiesen. Anschliessend habe ich Benutzer diesen Gruppen zugeordnet, um die Rechtevergabe zentral zu steuern.

Wie ich das gemacht habe:

1. **Gruppe erstellt:**
   Mit folgendem Befehl habe ich eine globale Sicherheitsgruppe in der OU "Mitarbeiter" erstellt:

New-ADGroup -Name "GRP_Home_Mitarbeiter" -GroupScope Global -GroupCategory Security -Path "OU=Mitarbeiter,DC=kmu,DC=intern"

```
PS C:\Users\Administrator> }
>>
>> # CSV einlesen und Homeverzeichnisse anlegen
>> Import-Csv "C:\Skripte\benutzer.csv" | ForEach-Object {
>>     $login = $_.Benutzername
>>     $userPath = Join-Path $freigabePfad $login
>>     $uncPfad = "\\$(hostname)\$freigabeName\$login"
>>
>>     # Ordner anlegen
>>     New-Item -Path $userPath -ItemType Directory -Force | Out-Null
>>
>>     # NTFS-Rechte setzen
>>     icacls $userPath /inheritance:r
>>     icacls $userPath /grant "$login:(OI)(CI)F"
>>
>>     # Homeverzeichnis im AD zuweisen
>>     Set-ADUser $login -HomeDirectory $uncPfad -HomeDrive "H:"
>> }
In Zeile:1 Zeichen:1
+ }
+ ~
Unerwartetes Token "}" in Ausdruck oder Anweisung.
In Zeile:14 Zeichen:30
+     icacls $userPath /grant "$login:(OI)(CI)F"
+                              ~~~~~~~
Ungültiger Variablenverweis. Nach ":" folgte kein Zeichen, das für einen Variablennamen gültig ist. Verwenden Sie ggf. "${}", um den Namen zu begrenzen.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : UnexpectedToken

PS C:\Users\Administrator>
```

Es gab hier und da natürlich ein paar Fehler.

2. Benutzer der Gruppe hinzugefügt:
   Ich habe beispielsweise den Benutzer „luca.meier“ der Gruppe zugewiesen:

Add-ADGroupMember -Identity "GRP_Home_Mitarbeiter" -Members "luca.meier"

```
PS C:\Users\Administrator> New-ADGroup -Name "GRP_Home_Mitarbeiter" -GroupScope Global -GroupCategory Security -Path "OU=Mitarbeiter,DC=kmu,DC=intern"
PS C:\Users\Administrator>
```

3. **Berechtigungen mit icacls gesetzt:**
   Die Gruppe hat über NTFS-Vergabe Zugriff auf den zugeordneten Freigabeordner erhalten:

icacls "C:\Gruppen\Home_Mitarbeiter" /grant "GRP_Home_Mitarbeiter:(OI)(CI)F"

```
PS C:\Users\Administrator> if (-not (Get-SmbShare -Name $freigabeName -ErrorAction SilentlyContinue)) {
>>     New-SmbShare -Name $freigabeName -Path $freigabePfad -FullAccess "Domänen-Admins"
>>     icacls $freigabePfad /grant "Domänen-Admins:(OI)(CI)F"}

D:\Home: Das System kann die angegebene Datei nicht finden.
0 Dateien erfolgreich verarbeitet, bei 1 Dateien ist ein Verarbeitungsfehler aufgetreten.
Name  ScopeName Path    Description
----  --------- ----    -----------
Home$ *         D:\Home


PS C:\Users\Administrator>
```

10. Ordnerstruktur erstellen

Was ich gemacht habe:
Ich habe eine strukturierte Verzeichnisstruktur für Gruppen, Benutzer und Abteilungen erstellt. Dazu gehören zentrale Ordner wie C:\Gruppen, C:\HomeVerzeichnisse sowie individuelle Benutzerordner. Diese Struktur ist die Basis für Netzlaufwerke, Homeverzeichnisse und Zugriffskontrollen.

Wie ich das gemacht habe:

Basisordner erstellt:
Zuerst habe ich über PowerShell die Hauptverzeichnisse für Homeverzeichnisse und Gruppenfreigaben angelegt:

New-Item -Path "C:\Gruppen\Home_Mitarbeiter" -ItemType Directory -Force

New-Item -Path "C:\HomeVerzeichnisse" -ItemType Directory -Force

Benutzerverzeichnisse automatisch generiert:
Mit einer Schleife wurden individuelle Ordner pro Benutzer erstellt:

$benutzer = "luca.meier"

$pfad = "C:\HomeVerzeichnisse\$benutzer"

New-Item -Path $pfad -ItemType Directory -Force

SMB-Freigabe eingerichtet:

Die erstellten Ordner wurden als Freigabe im Netzwerk sichtbar gemacht, mit Zugriff für eine bestimmte Gruppe:

New-SmbShare -Name "Home_Mitarbeiter" -Path "C:\HomeVerzeichnisse" -FullAccess "GRP_Home_Mitarbeiter"

```
PS C:\Users\Administrator> if (-not (Get-SmbShare -Name $freigabeName -ErrorAction SilentlyContinue)) {
>>     New-SmbShare -Name $freigabeName -Path $freigabePfad -FullAccess "Domänen-Admins"
>>     icacls $freigabePfad /grant "Domänen-Admins:(OI)(CI)F"}

D:\Home: Das System kann die angegebene Datei nicht finden.
0 Dateien erfolgreich verarbeitet, bei 1 Dateien ist ein Verarbeitungsfehler aufgetreten.
Name  ScopeName Path    Description
----  --------- ----    -----------
Home$ *         D:\Home


PS C:\Users\Administrator>
```

11. GPOs importieren

Was ich gemacht habe:

Ich habe Gruppenrichtlinien importiert und verlinkt, die zentrale Einstellungen im Netzwerk steuern: automatische Laufwerkszuweisung (H:), Sperrzeit nach Inaktivität sowie das Aktivieren von RDP für Benutzergruppen. Die Zuweisung erfolgt über XML-Dateien, die in den SYSVOL-Pfad der Domäne geschrieben wurden.

Wie ich das gemacht habe:

1. **GPO erstellt:**
   Zuerst habe ich per PowerShell eine neue Gruppenrichtlinie erstellt:

New-GPO -Name "KMU_Laufwerkszuweisung"

```
PS C:\Users\Administrator> New-GPO -Name "KMU_Basisrichtlinie" -Comment "Standardrichtlinie für Benutzer"


DisplayName      : KMU_Basisrichtlinie
DomainName       : kmu.intern
Owner            : KMU\Domänen-Admins
Id               : 8c88fe22-dd4d-41db-b787-b7a643d1f18b
GpoStatus        : AllSettingsEnabled
Description      : Standardrichtlinie für Benutzer
CreationTime     : 27.06.2025 08:24:10
ModificationTime : 27.06.2025 08:24:10
UserVersion      : AD-Version: 0, SysVol-Version: 0
ComputerVersion  : AD-Version: 0, SysVol-Version: 0
WmiFilter        :


PS C:\Users\Administrator>
```

2. **GPO mit einer OU verknüpft:**
   Die Richtlinie wurde an die OU „Mitarbeiter“ gebunden:

New-GPLink -Name "KMU_Laufwerkszuweisung" -Target "OU=Mitarbeiter,DC=kmu,DC=intern"

```
PS C:\Users\Administrator> New-GPLink -Name "KMU_Basisrichtlinie" -Target "OU=Mitarbeiter,DC=kmu,DC=intern"


GpoId       : 8c88fe22-dd4d-41db-b787-b7a643d1f18b
DisplayName : KMU_Basisrichtlinie
Enabled     : True
Enforced    : False
Target      : OU=Mitarbeiter,DC=kmu,DC=intern
Order       : 1



PS C:\Users\Administrator>
```

3. **Laufwerk H: per XML konfiguriert:**
   Ich habe eine XML-Datei für die Laufwerkszuordnung generiert und in das entsprechende SYSVOL-Verzeichnis kopiert:

$gpo = Get-GPO -Name "KMU_Laufwerkszuweisung"

$gpoId = $gpo.Id

$domain = (Get-ADDomain).DNSRoot

$gpoPath = "\\$domain\SYSVOL\$domain\Policies\{$gpoId}\User\Preferences\Drives"

New-Item -Path $gpoPath -ItemType Directory -Force | Out-Null

$xml = @'

<Drive clsid="{...}" name="H Drive" status="OK">

<Properties action="U" thisDrive="H:" useLetter="1" userName="%USERNAME%" path="\\SRV01\Home$\%USERNAME%" />

</Drive>

'@

Set-Content -Path "$gpoPath\Drives.xml" -Value $xml -Encoding UTF8

```
PS C:\Users\Administrator> # GPO definieren
>> $gpoName = "KMU_Laufwerkzuweisung"
>> $gpo = Get-GPO -Name $gpoName
>> $gpoId = $gpo.Id
>> $domain = (Get-ADDomain).DNSRoot
>> $gpoPath = "\\$domain\SYSVOL\$domain\Policies\{$gpoId}\User\Preferences\Drives"
>>
>> # Ordner erstellen
>> New-Item -Path $gpoPath -ItemType Directory -Force | Out-Null
>>
>> # XML für Laufwerk H:
>> $xml = @'
>> <Drive clsid="{C631DF4C-088F-48E3-A6F0-13A016BFC0C3}" name="H Drive" status="OK">
>>   <Properties action="U" thisDrive="H:" useLetter="1" userName="%USERNAME%" path="\\SRV01\Home$\%USERNAME%" persistent="0" mapAs="0" />
>> </Drive>
>> '@
>>
>> # XML schreiben
>> $xmlPath = Join-Path $gpoPath "Drives.xml"
>> $xml | Set-Content -Path $xmlPath -Encoding UTF8
PS C:\Users\Administrator> _
```

12. Benutzerbegrüßung beim Login

Was ich gemacht habe:
Ich habe ein PowerShell-Skript geschrieben, das beim Login ein Popup-Fenster mit dem Namen des Benutzers anzeigt – als persönliche Begrüssung.

Wie ich das gemacht habe:

$msg = "Willkommen $env:USERNAME"

[System.Windows.MessageBox]::Show($msg)

Dieses Skript wird über die Registry im Autostart eingebunden oder direkt über die GPO-Logon-Einstellungen ausgelöst.

```
PS C:\Users\Administrator> $msg = "Willkommen $env:USERNAME"
>> [System.Windows.MessageBox]::Show($msg)
```

13. Dienste überwachen & loggen

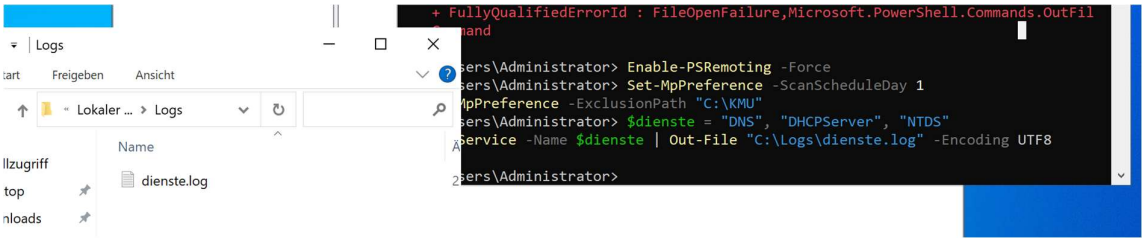
Was ich gemacht habe:
Ich habe ein Skript erstellt, das prüft, ob wichtige Dienste wie DNS, DHCP und AD laufen, und deren Status in eine Logdatei schreibt. Das hilft bei der Fehleranalyse und Systemüberwachung.

Wie ich das gemacht habe:

$dienste = "DNS", "DHCPServer", "NTDS"

Get-Service -Name $dienste | Out-File "C:\Logs\dienste.log" -Encoding UTF8

Das Skript wird regelmäßig per Taskplaner oder manuell ausgeführt.



14. Remote PowerShell aktivieren

Was ich gemacht habe:
Ich habe PowerShell-Remoting aktiviert, damit der Server später remote verwaltet werden kann – z. B. von Admin-PCs oder zur automatischen Wartung per Skript. Das ist besonders wichtig in produktiven Netzwerken, um physische Zugriffe zu vermeiden.

Wie ich das gemacht habe:

Enable-PSRemoting -Force

Damit wird auf dem Server der WinRM-Dienst gestartet und konfiguriert, inklusive Firewallregel. Ich habe anschliessend mit folgendem Befehl die Verbindung von einem anderen Gerät getestet:

Enter-PSSession -ComputerName SRV01

```
PS C:\Users\Administrator> Enable-PSRemoting -Force
PS C:\Users\Administrator> _
```

16. Windows Defender konfigurieren

Was ich gemacht habe:
Ich habe den integrierten Virenschutz Windows Defender so eingestellt, dass automatische wöchentliche Scans durchgeführt werden. Gleichzeitig habe ich Ausnahmen für freigegebene Verzeichnisse konfiguriert, um Konflikte und Performance-Probleme zu vermeiden.

Wie ich das gemacht habe:

Set-MpPreference -ScanScheduleDay 1

Set-MpPreference -ExclusionPath "C:\KMU"

Damit wird jeden Montag ein Scan ausgeführt, und der Ordner C:\KMU (z. B. mit Homeverzeichnissen) von Echtzeitscans ausgenommen.

```
PS C:\Users\Administrator> Set-MpPreference -ScanScheduleDay 1
>> Set-MpPreference -ExclusionPath "C:\KMU"
PS C:\Users\Administrator> _
```

Persönliches Fazit:

Das Projekt war für mich sehr lehrreich und spannend. Ich konnte mein Wissen in PowerShell, Active Directory und der Serverkonfiguration stark vertiefen. Besonders hilfreich war der strukturierte Aufbau mit klaren Issues und Meilensteinen, wodurch ich Schritt für Schritt vorgehen konnte. Obwohl es an manchen Tagen technische Herausforderungen gab – zum Beispiel mit Rechten oder fehlerhaften Befehlen – konnte ich diese mit Geduld und gezielter Fehlersuche lösen.

Rückblickend bin ich sehr zufrieden mit dem Ergebnis: Das gesamte System funktioniert wie geplant und die automatisierte Benutzerverwaltung spart viel Zeit. Ich habe gemerkt, wie wichtig sorgfältige Planung und Testen sind – besonders bei Skripten, die auf mehreren Ebenen eingreifen. Ich habe auch gelernt, wie schnell Daten verloren gehen können, wenn man nicht regelmäßig speichert (Tag 7 war da eine wichtige Lektion).

Insgesamt hat mir das Projekt gezeigt, dass ich in der Lage bin, eine vollständige IT-Lösung für ein kleines Unternehmen selbständig zu planen, umzusetzen und zu dokumentieren. Dieses Erfolgserlebnis hat mein Selbstvertrauen im IT-Bereich gestärkt.

Quellenangabe:

Keine

Github Repo Link:

https://github.com/Andrija34/Abschlussprojekt_2025_PLA-2_andmil_Board_Server-Skript