

Rechtevergabe

Inhaltsverzeichnis

Aufgabenstellung:	2
Festplatte anlegen:	2
Festplatte formatieren:	4
Benutzer und Gruppen erstellen:	5
Skript:	5
Ordner für die einzelnen OU erstellen:	6
Skript:	7
Im Netzwerk freigeben	7
Rechtevergabe:	8
Alte Rechte von Homedirectory entfernen:	8
Haupt-Verzeichnisse Rechte geben:	9
Neue Rechte vergeben:	10
Skript:	11
Ergebnis im WindowsServer:	11
Ergebnis im WindowsClient:	12

Aufgabenstellung:

- Erstellung eines eigenen Volumes in Virtual-Box + Zuweisung zur Windows-Server-VM
- Datenträger in "Disk Management" mounten (z.B. C:\DATA)
- Ordnerstruktur aufbauen, Userverzeichnisse automatisiert mit Powershell-Skript anlegen. Beispiel:
 - homedirectories
 - lehrer
 - lehrer1
 - lehrer2
 - lehrer3
 - schueler
 - schueler1
 - schueler2
 - schuler3
 - verwaltung
 - verwaltung1
 - verwaltung2
 - verwaltung3
- Analog dazu im Active-Directory OUs für User-Objekte und Gruppen-Objekte anlegen
 - Useranlage per Powershell (3 Userobjekte pro Gruppe)
 - Gruppenanlage (lehrer, schueler, verwaltung) per Powershell
- Berechtigungen definieren
 - Ein User darf nur seinen eigenen Ordner am Share einsehen und Inhalte verändern.
 - Verwende das Commandlet Set-Acl um die Berechtigungen für alle Verzeichnisse automatisiert zu setzen.

Dokumentiere alle notwendigen Schritte in einem Protokoll (2er-Teams erlaubt, Abgabe des Protokolls jedoch von beiden Teammitgliedern!)

Festplatte anlegen:

Unter der Option Tools kann man mit **Create** eine neue Festplatte erstellen.

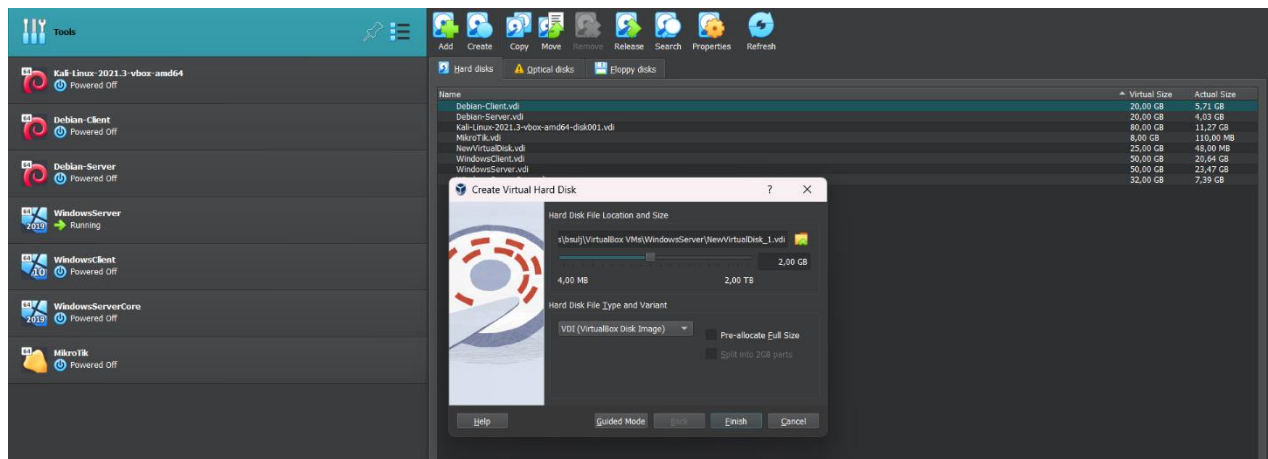


Abbildung 1 - VirtualBox Tool Option mit Erstellung einer neuen Festplatte

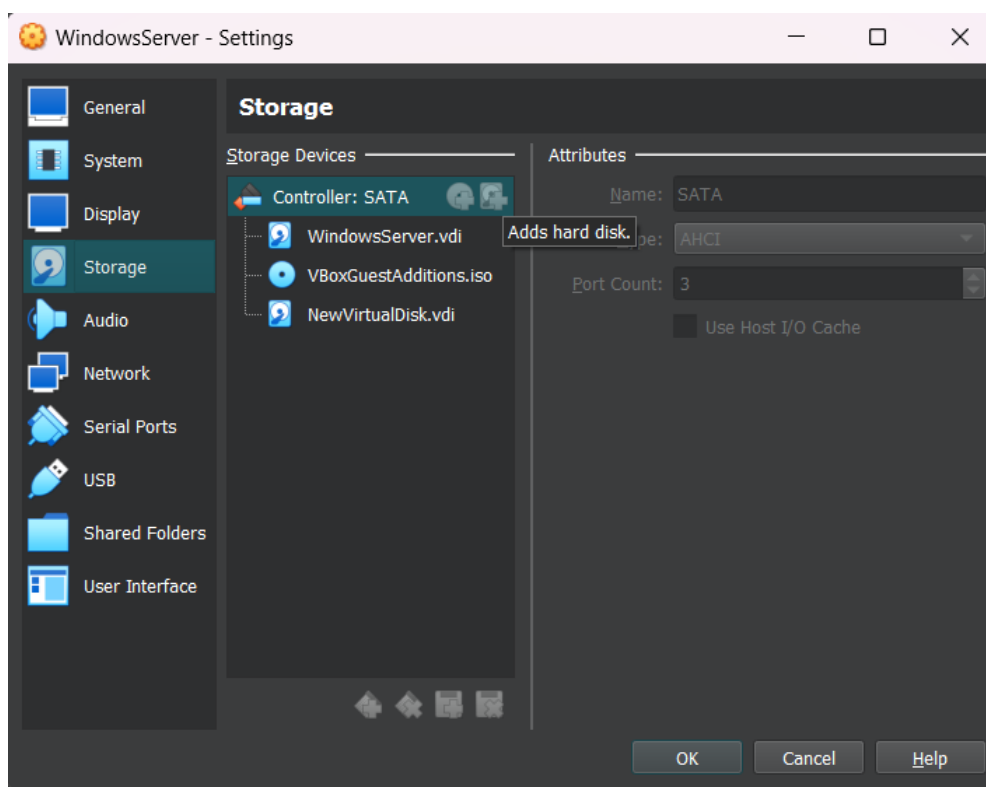


Abbildung 2 - VM Settings im Optionspunkt Storage

Wichtig zu beachten ist, dass **Pre-allocate Full Size** nicht aktiviert ist.

Danach in der Virtuellen Maschine unter **Storage** kann man mit **Add Hard Disk** die Festplatte hinzufügen.

Festplatte formatieren:

Um die Festplatte zu formatieren, muss man in den **Disk Management** wechseln

Mit **Run** und **diskmgmt.msc** wechselt man sofort zum **Disk Management**

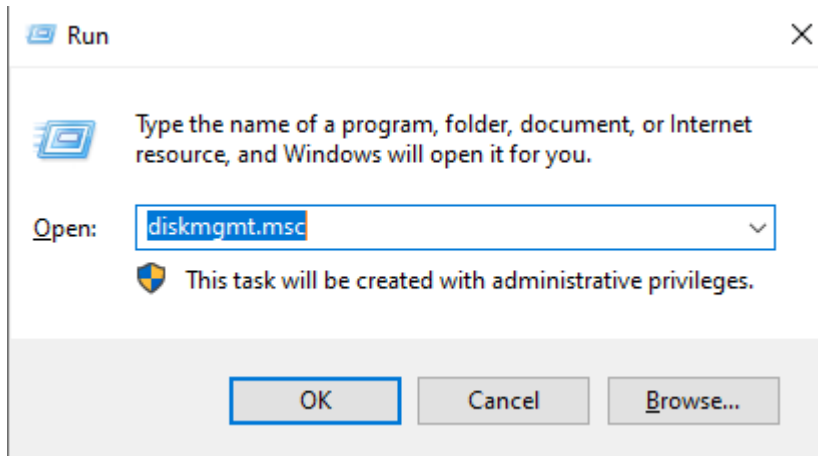


Abbildung 3 - Run Fenster mit diskmgmt.msc

Daraufhin eine neue Disk formatieren die **DATA** genannt wird.

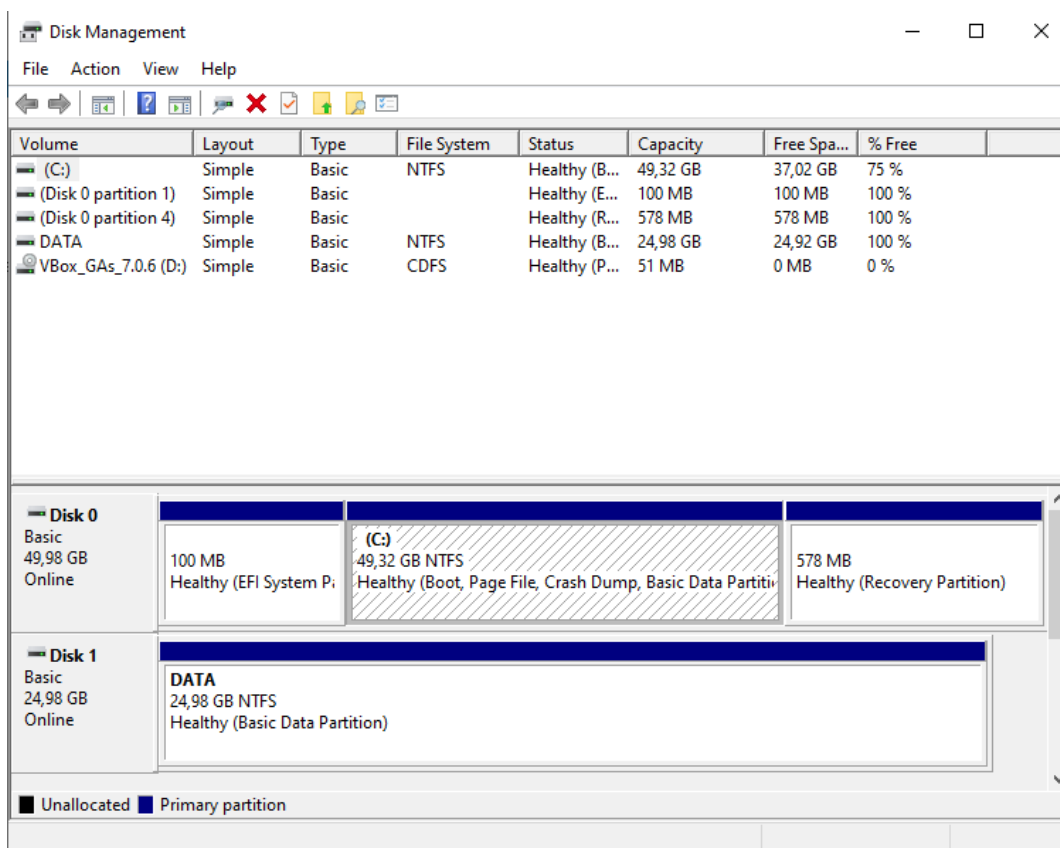


Abbildung 4 - Disk Management mit der DATA Disk

In DATA dann, denn Homedirectories Order erstellen.

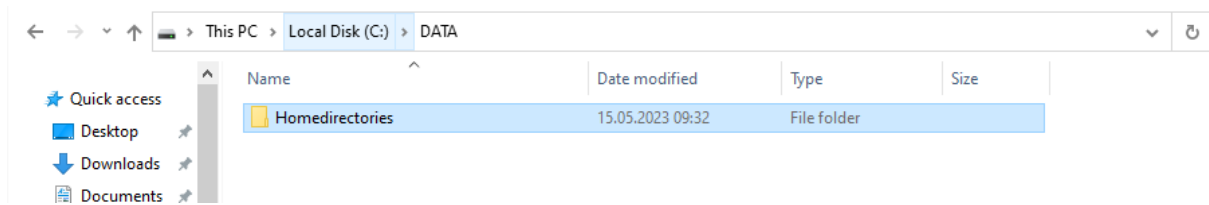


Abbildung 5 - Homedirectories in der DATA Disk

Benutzer und Gruppen erstellen:

Im WindowsServer danach in das **Active Directory Users and Computer** wechseln.

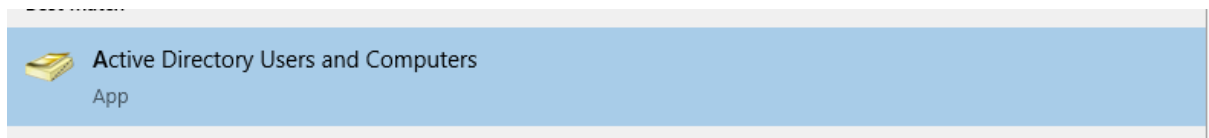


Abbildung 6 - Active Directory am WindowsServer

Skript:

Um die Benutzer und die OU's zu erstellen wird ein Skript benutzt.

```
#-----
$path = "ou=people,dc=suljevic,dc=at"
$groups = "schuler", "lehrer", "verwaltung"

New-ADOrganizationalUnit -Name "group" -Path "$path"

foreach($group in $groups) {

    New-ADGroup -GroupScope universal -Name "$group" -Path "ou=group,$path"
    $group

    New-ADOrganizationalUnit -Name $group -Path "$path"

    for($i = 1; $i -le 3; $i++) {
        $user = $group+$i
        $user

        New-ADUser -Name $user -DisplayName $user -Enabled:$true `
        -AccountPassword (ConvertTo-SecureString -AsPlainText "ww1ssPfuWI" -force) `
        -UserPrincipalName $user@suljevic.at -SamAccountName $user `
        -Path "ou=$group,$path" -Department $group `
        -HomeDirectory "\\winsrv01\DATA\homedirectory\$group\$user" -HomeDrive "H:"

        Add-ADGroupMember -Identity "$group" -Members "$user"
    }
}
```

Abbildung 7 - Skript zum Erstellen von Nutzern

Hier werden nicht nur Benutzer, sondern auch **Groups** und **Organizational Units (OU)** erstellt und alle werden einem **Department** zugeteilt.

Im **Active Directory** werden unter People die OU's für Lehrer, Schüler und Verwaltung angezeigt.

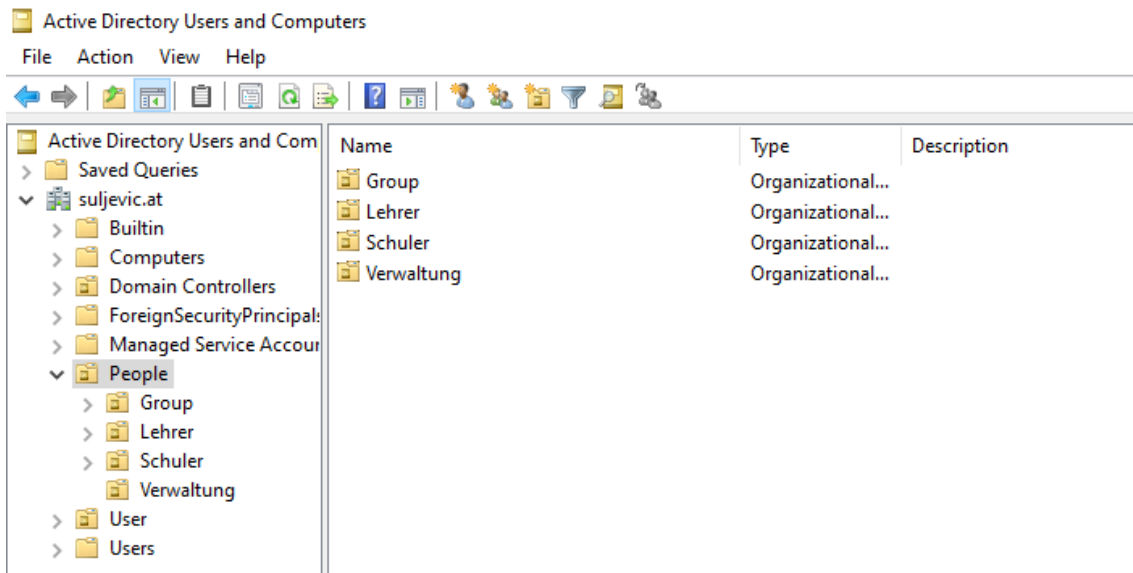


Abbildung 8.1 - Active Directory; People mit den verschiedenen Organizational Unit

In den jeweiligen Organizational Unit sieht man gleich die erstellten Benutzer.

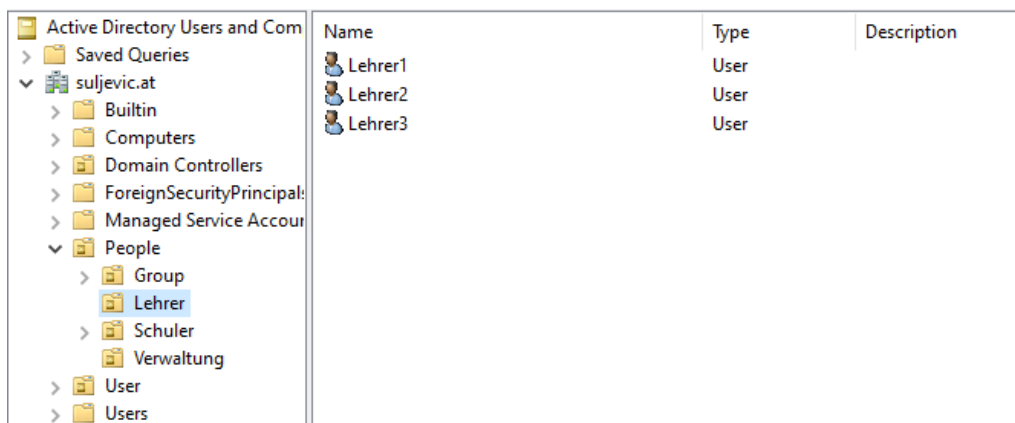


Abbildung 8.2 - Anzeige der Verschiedenen Benutzer im Active Directory

Ordner für die einzelnen OU erstellen:

Um die Order zu erstellen, wurde ein Skript genutzt.

Skript:

```
1 $users = Get-ADUser -SearchBase "ou=people, dc=suljevic, dc=at" -Filter "samaccountname -like ''" -Properties ""
2
3 foreach($user in $users){
4
5     if ($user.Department -eq $null) {
6         continue
7     }
8
9     try {
10         echo ("C:\DATA\homedirectory" + $user.Department + "" + $user.samAccountName)
11         $item = Get-Item -Path "C:\DATA\homedirectory" + $user.Department + "" + $user.samAccountName -ErrorAction Stop
12     } catch {
13         "Homedirectory does not exist for " + $user.SamAccountName
14         New-Item -Path ("C:\DATA\homedirectory" + $user.Department) -Name $user.SamAccountName -ItemType Directory
15     }
16 }
17
```

Abbildung 9 - Skript zum Erstellen von den Verzeichnissen

Die Ordner kann man dann in Homedirectories sehen.

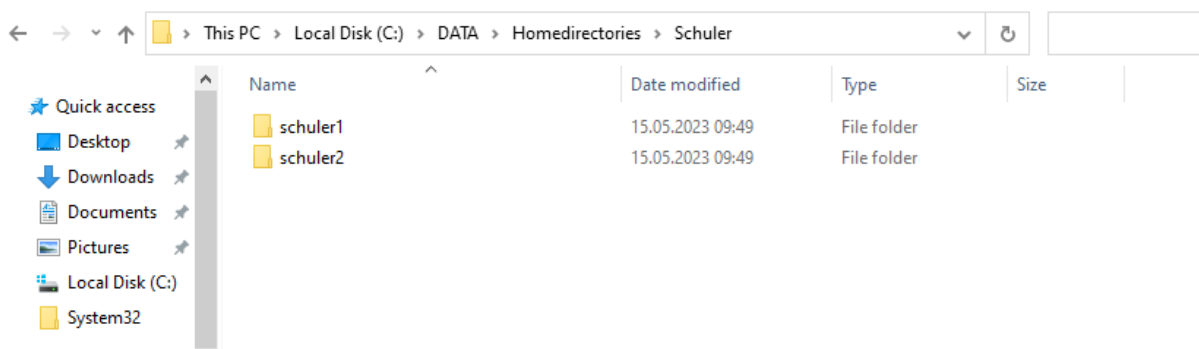


Abbildung 10 - Verzeichnisse die vom Skript erstellt wurden

Im Netzwerk freigeben

Um das Homedirectory im Netzwerk freizugeben, musste man unter **Properties→Sharing→Advanced Sharing→ Share this folder**

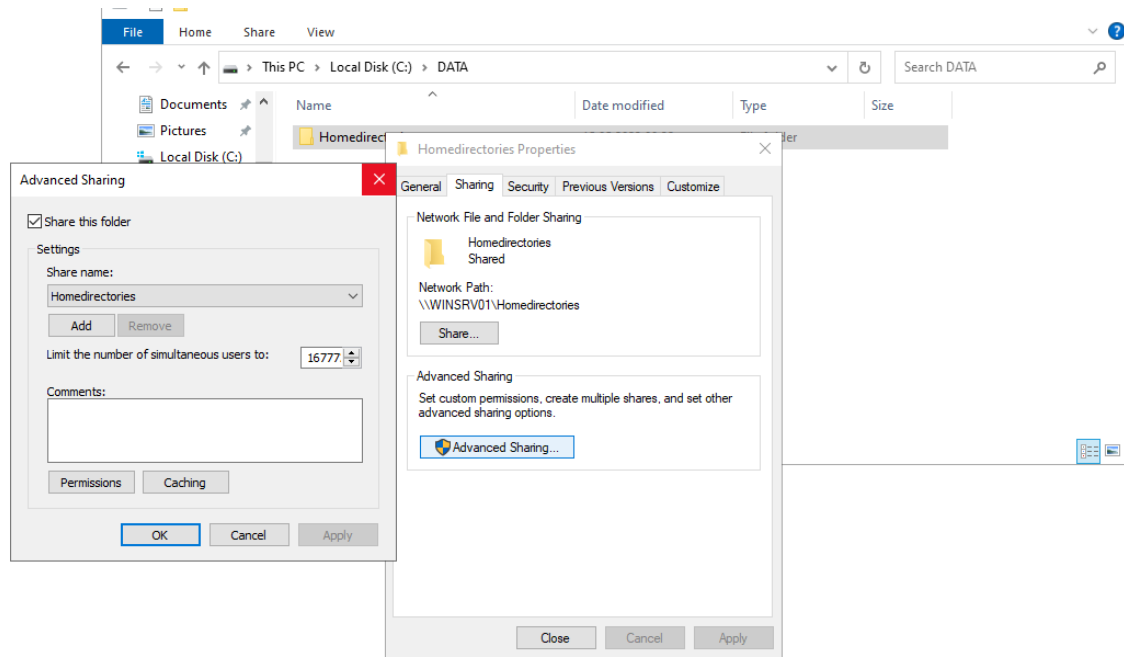


Abbildung 11 - Die Einstellung, um den Ordner im Netzwerk anzuzeigen

Rechtevergabe:

Alte Rechte von Homedirectory entfernen:

Damit die Rechte von Homedirectory nicht weitervererbt werden muss man sie davor löschen.

Mit **Disable inheritance** werden alle Rechte gelöscht (Danach wird **Enable inheritance** angezeigt).

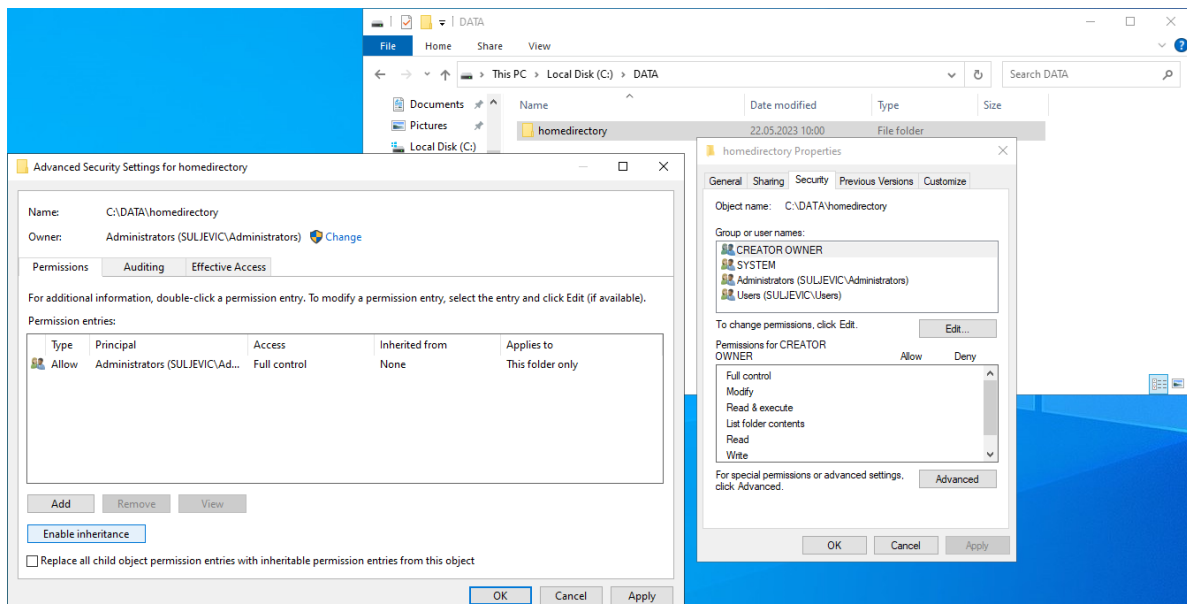


Abbildung 12 - Löschen der alten Rechte im Verzeichnis Homedirectory

Danach im Homedirectory die Zugriffsrechte für die einzelnen groups hinzufügen.

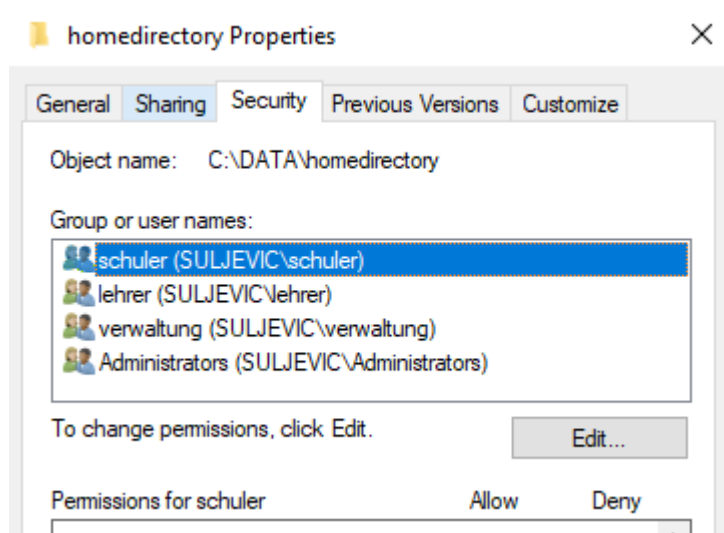


Abbildung 13 - Zugriffsrechte für Homedirectory

Haupt-Verzeichnisse Rechte geben:

Lehrer, Schuler und Verzeichnisse die Rechte geben, um auf ihre Unter-Verzeichnisse zuzugreifen.

Bei einem Ordner auf **Properties**→**Security**→**Edit**→**Add**→**Enter the object names to select (den Namen der jeweiligen groups eingeben)**

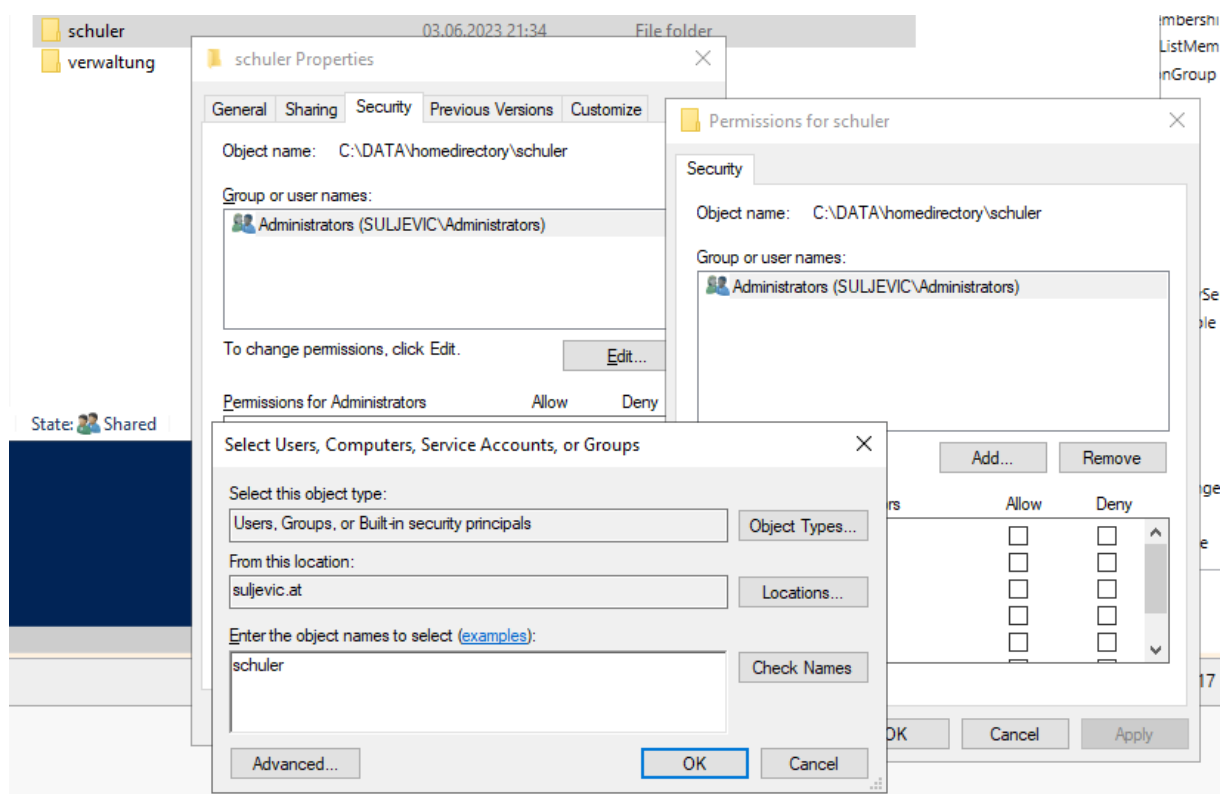


Abbildung 14 - Berechtigungen für das Verzeichnis Schuler setzen

Die Group danach auswählen.

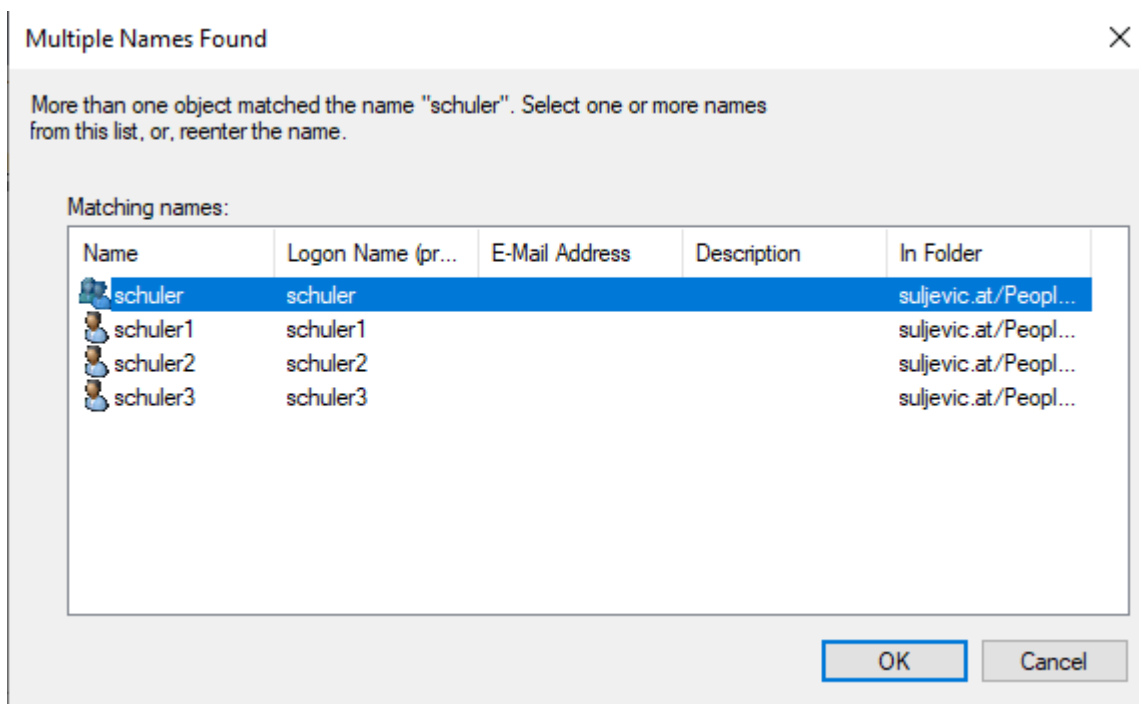


Abbildung 15 - Die jeweilige group auswählen

Die hinzugefügte group kann man dann in Security sehen.

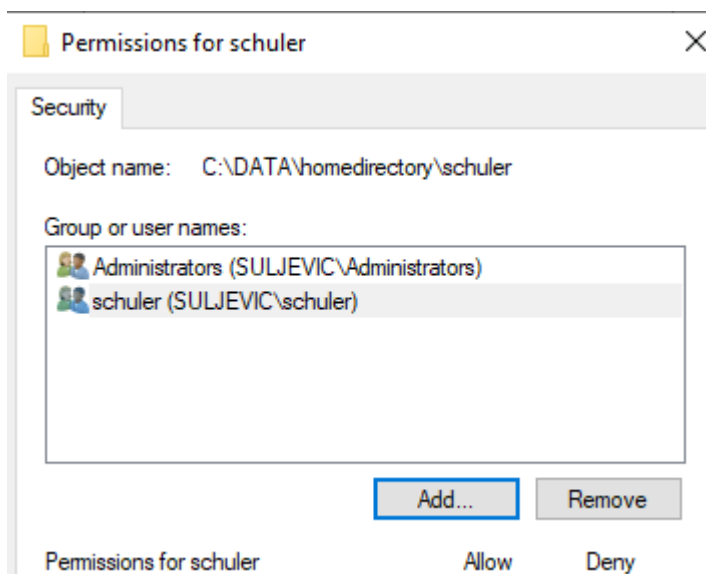


Abbildung 16 - Die Anzeige der group in Security

Neue Rechte vergeben:

Die Rechte werden mit einem Skript übergeben.

Skript:

```

19
20 $users = Get-ADUser -SearchBase "ou=people, dc=suljevic, dc=at" -Filter "samaccountname -like '*'" -Properties "*"
21
22 foreach($user in $users){
23
24     if ($user.Department -eq $null) {
25         continue
26     }
27
28     $path = ("C:\DATA\Homedirectory\" + $user.Department + "\" + $user.SamAccountName)
29
30     $acl = Get-Acl -Path $path
31
32     $accessRule = New-Object System.Security.AccessControl.FileSystemAccessRule($user.SamAccountName, 'FullControl', 'None', 'None', 'Allow')
33
34     $acl.AddAccessRule($accessRule)
35
36     Set-Acl -Path $path -AclObject $acl
37 }
38

```

Abbildung 17 - Skript um Rechte für die Unterordner zu vergeben

Ergebnis im WindowsServer:

Im WindowsServer kann man in, denn einzelnen Verzeichnissen sehen wer „**Full control**“ hat.

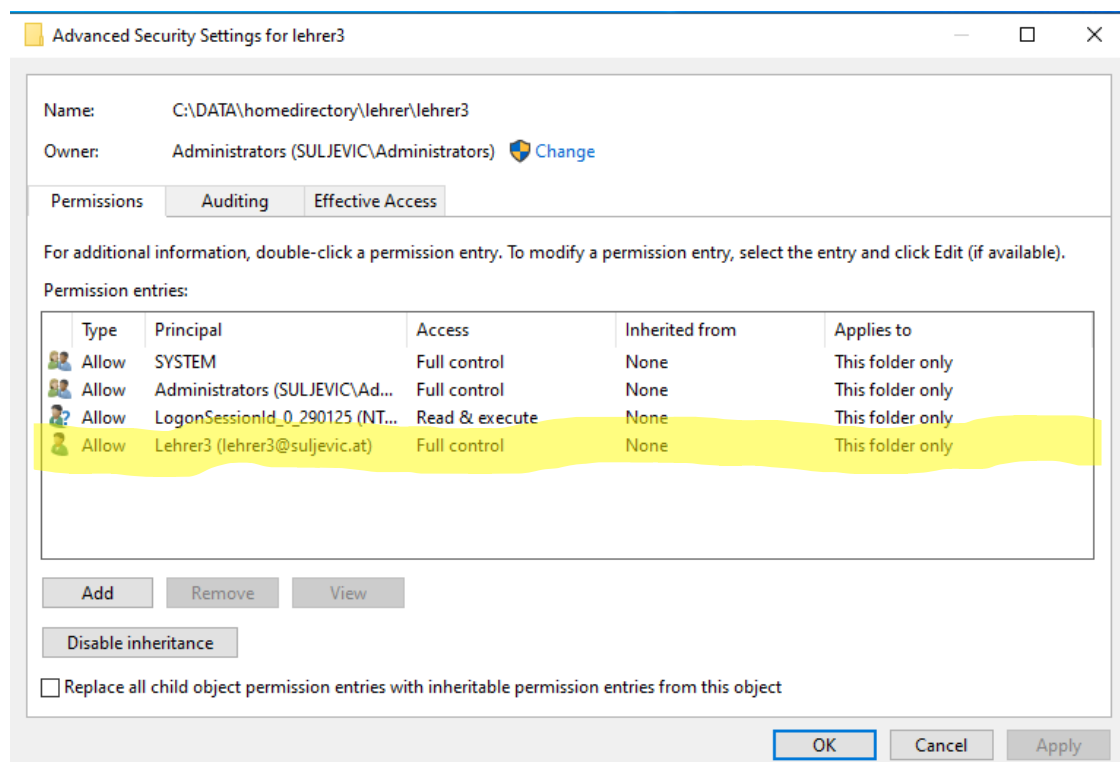


Abbildung 18 - Der Benutzer Lehrer 3 hat nur auf seinem Ordner volle Rechte

Ergebnis im WindowsClient:

Danach sich am WindowsClient mit einem der erstellten Benutzer einloggen

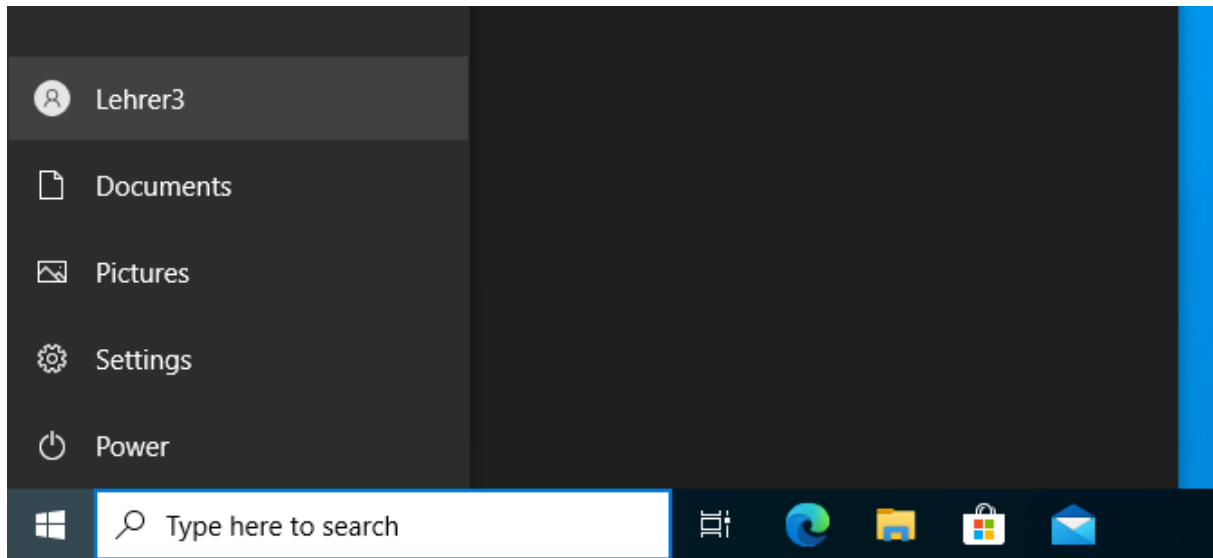


Abbildung 19 - Der Benutzer "Lehrer3" am WindowsClient

Im **Verzeichnis Explorer** in die URL <\\WINSRV01\homedirectory> eingeben.

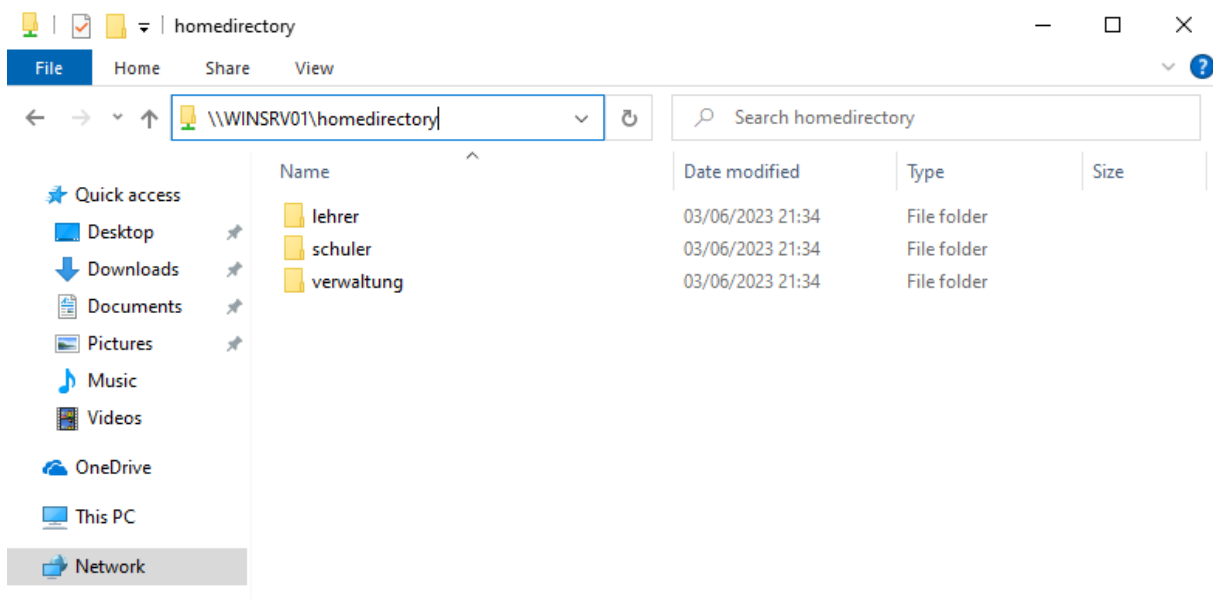


Abbildung 20 - Homedirectory am WindowsClient

Nachdem ist es möglich die Verzeichnisse zu sehen.

Lehrer 3 kann nur auf seinen eigenen Ordner zugreifen, da er für die anderen keine Berechtigungen besitzt.

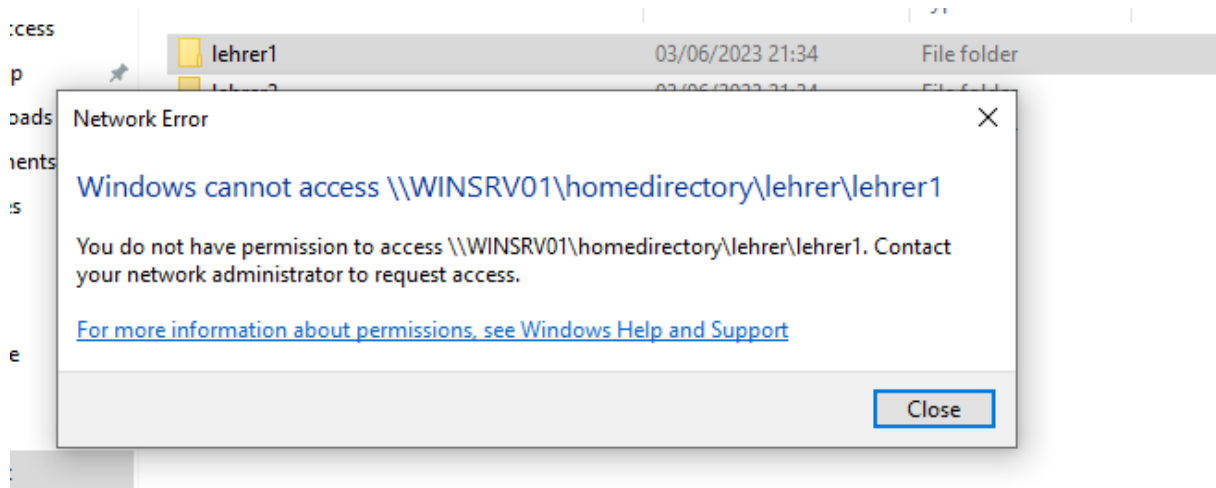


Abbildung 21 - Der Benutzer Lehrer3 hat keine Rechte um auf Lehrer1 zuzugreifen

In Ordner von Lehrer3 ist es dem Benutzer möglich Dateien zu erstellen, abzuspeichern, bearbeiten oder zu löschen.

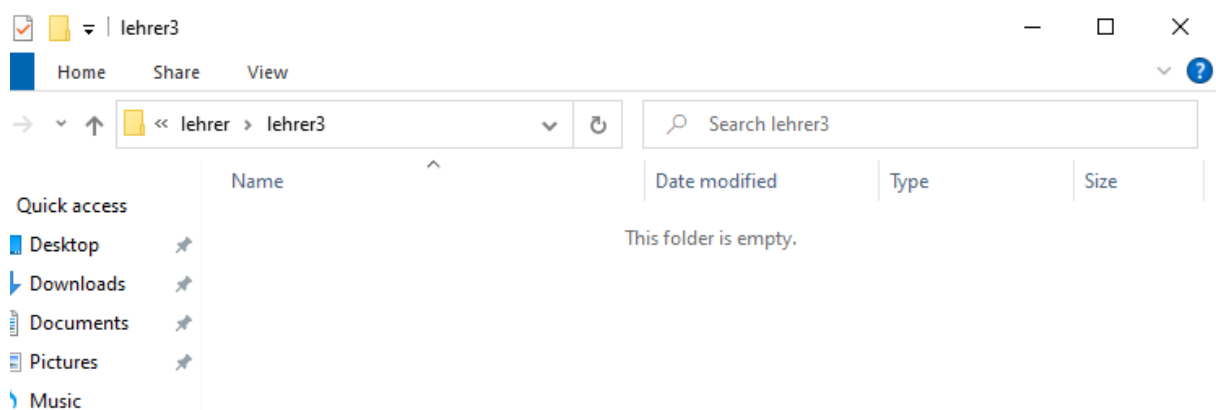


Abbildung 22 - Der eigene Ordner für den Benutzer