

| Auftrag 0003

- Der Auftrag
- SAMBA Installieren unter Linux
- SAMBA Konfigurieren unter Linux
- Datei- / Verzeichnisberechtigungen
- SMB Freigaben unter Windows einrichten

Der Auftrag

Das Unternehmen **Star Entwicklung AG** möchte die derzeitige Datei-Verwaltung durch eine zentralisierte Server-Lösung ablösen.

Die Lösung soll folgende Möglichkeiten bieten.

- Die Lösung soll direkt Betriebssystem integriert werden.
- Die Mitarbeiter verwenden Windows und Mac Geräte. Diese sollen unterstützt werden.

Diese Funktionen können mit einem SMB-File-Server abgedeckt werden.

- Jeder Benutzer soll ein persönliches Laufwerk für seine Dateien erhalten.
- Es existieren weitere Verzeichnisse, auf jene die Benutzer anhand bestimmter Berechtigungen Zugriff erhalten sollen. Die Geschäftsleitung stellt hierzu weiter unten eine Übersicht bereit.

Der Auftrag erfolgt in drei Schritten:

1. **Konzeptionelles** Erstellen Sie ein Namenskonzept für die Benutzer inkl. eigens definierten Passwörtern und erstellen Sie eine Berechtigungsmatrix für den Dateizugriff.
2. **Installation und Bereitstellung:** Sie installieren die Lösung auf einer AWS-EC2-Instanz, damit diese verwendet werden kann.
3. Erstellung **technische Dokumentation**.

Mitarbeiter und Abteilungen

Nachfolgend finden Sie alle Mitarbeiter im Unternehmen und die Abteilungen, in denen diese arbeiten:

- Dieter Schwartz, Geschäftsleitung
- Stefanie Barth, Geschäftsleitung
- Matthias Köhler, Administration
- Tim Baer, Entwicklung
- Christina Naumann, Entwicklung

- Daniela Hahn, Grafik
- Daniele Hauser, Kundenbetreuung

Es obliegt Ihnen als Unternehmen ein gutes Namens-Konzept auszuarbeiten.

Berechtigungen

Die Geschäftsleitung hat Ihnen wie folgt beschrieben, wie die Berechtigung der einzelnen Benutzer auf die Dateien genau aussehen soll:

- Jeder Benutzer soll die Lese- und Schreiberechtigung auf sein persönliches Verzeichnis haben. Darin sollen die Benutzer die persönlichen Dateien ablegen können. Die Geschäftsleitung hat keinen Zugriff auf die persönlichen Daten der Mitarbeitenden.
- Jede Abteilung hat ein eigenes Verzeichnis in dem die relevanten Daten der Abteilungen liegen. Jede Abteilung soll vollen Zugriff in das eigenen Verzeichnis erhalten. Die Geschäftsleitung hat Lese- und Schreiberechtigung auf alle Abteilungs-Verzeichnisse.
- Die Abteilungen Entwicklung und Grafik haben ein zusätzliches Verzeichnis "projekt_webseite", indem gemeinsam benötigte Dateien abgelegt werden können. Administration und Kundenbetreuung haben auf dieses Verzeichnis keinen Zugriff. Die Geschäftsleitung hat Lese und Schreibezugriff auf diese Freigabe.
- Es existiert das Verzeichnis "info", in de die Geschäftsleitung regelmässig Informationen für die Mitarbeitenden ablegt. Die Mitarbeitenden sollen diese Informationen nur lesen können. Die Geschäftsleitung benötigt natürlich Schreiberechtigung.
- Im Verzeichnis "share" haben alle Benutzer Zugriff vollen Zugriff.

Erwartete Abgaben

- Ein fertig installierter und getesteter SMB-Server.
- Technische Dokumentation gem. Vorlage als Word oder PDF:
 - [Vorlage_technische_Dokumentation_m123.docx](#)
 - [Anleitung zum Testen von Systemen und Objekten für Berufsschulen.docx](#)
 - In der Dokumentation muss das Namenskonzept als Tabelle / Liste der Benutzer, deren Gruppen inkl. Passwörter enthalten sein.
 - In der Dokumentation muss die Berechtigungsmatrix enthalten sein.

Organisatorisches

Bevor Sie mit der Konfiguration des SMB-Servers beginnen, zeigen Sie der Lehrperson Ihr Namenskonzept und die Berechtigungs-Matrix.

Lohn: 8000

Bonus

Falls Sie alle Aufträge umgesetzt haben können Sie folgende Bonus-Aufgabe in Angriff nehmen.

- Das Verzeichnis Share soll automatisch alle 10 Minuten geleert werden. (Tipp cronjob / rm)

SAMBA Installieren unter Linux

Vorbereitung

Sie benötigen eine neue AWS EC2 Instanz.

- 1. Erstellen Sie vorgängig eine Sicherheitsgruppe für den SMB-Dienst.

Sicherheitsgruppe erstellen

Folgende Einstellungen müssen gesetzt werden:

Typ	Portbereich	Quelle
SSH		0.0.0.0/0
TCP	139	0.0.0.0/0
SMB		0.0.0.0/0

Regeln für eingehenden Datenverkehr

ID der Sicherheitsgruppenregel	Typ	Protokoll	Portbereich	Quelle	Beschreibung – optional	
sgr-014953bfb8e89d811	SSH	TCP	22	Benutz... 0.0.0.0/0		Lösch en
sgr-0efdf9e624f93b9fb	Benutzerdefiniertes TCP	TCP	139	Benutz... 0.0.0.0/0		Lösch en
sgr-04488505cfbdc6274	SMB	TCP	445	Benutz... 0.0.0.0/0		Lösch en

Regel hinzufügen

Nun können Sie eine neue EC2-Instanz mit dieser Sicherheitsgruppe erstellen:

[EC2 Instanz anlegen](#)

3. Weisen Sie anschliessend eine statisch IP-Adresse dem neuen Server zu:

[Eine statische IP-Adresse zuweisen](#)

Installation von Samba

1. Erstellen Sie eine AWS-EC2 Instanz und verbinden Sie sich mit Ihrem SSH-Client (Alternativ können Sie auch den Webzugriff verwenden)
2. Wechseln Sie mit dem Befehl **sudo su** in den Administrator-Modus.
3. Installieren Sie samba mit folgendem Befehl:

```
yum install samba
```

Verwaltung des Dienstes

Wie beim DNS-Dienst named erfolgt die Verwaltung des Dienstes über systemctl

Wenn Sie wollen, dass der smb-Dienst automatisch startet, wenn der Server aufgestartet wird, dann müssen Sie diesen einmalig aktivieren:

```
systemctl enable smb
```

Dienst starten:

```
systemctl start smb
```

Dienst beenden:

```
systemctl stop smb
```

Dienst neu starten:

```
systemctl stop smb
```

Status des Dienstes ausgeben:

```
systemctl status smb
```

SAMBA Konfigurieren unter Linux

Mit die nachfolgende Anleitung beschreibt die Grundkonfiguration von SAMBA, einem SMB Server der unter Linux Betrieben werden kann.

Sicherheitshinweis: SMB ist ein Protokoll, dass es nun schon sehr lange gibt und in den letzten Jahren starke Veränderungen erfahren hat. Ab der SMB Version 3 wurde Verschlüsselung eingeführt. Zuvor war es nicht möglich, den SMB Datenverkehr ausreichend sicher zu verschlüsseln. Eine Nutzung im LAN ist hierbei noch zu vertreten, aber in diesem Setup richten Sie eine SMB-Verbindung über das Internet ein. Dieses Setup darf nicht mit einer SMB Version kleiner als SMBv3.1.1 vorgenommen werden. **SMB1 und SMB2 sind nicht sicher und für den Betrieb über das Internet nicht ausgelegt.**

Die Version SMB3 ist ausreichend sicher und wird auch von Microsoft in der Azure Cloud in dieser Form bereitgestellt. **In diesem Setup wird daher die SMB1 und SMB2 Unterstützung des SAMBA Server deaktiviert. Wird dies vergessen, gehen Sie ein grosses Risiko ein, was entsprechend in der Bewertung der Arbeit sich niederschlagen wird!**

Grundkonfiguration

Sie finden die Konfigurations-Datei unter folgendem Pfad: `/etc/samba/smb.conf`

Die Direktive der SAMBA-Konfiguration sind wie beim BIND Dienst in einzelne Sektionen unterteilt. Diese sind jedoch nicht mit geschweiften Klammern umschlossen. Stattdessen beginne jede Sektion mit einem Titel in eckigen Klammern z. B. die **global**-Sektion:

```
[global]

workgroup = SAMBA
security = user

passdb backend = tdbsam

printing = cups
printcap name = cups
load printers = yes
cups options = raw
```

Sektion [global]

In der globalen Sektion werden die allgemeinen Einstellungen und Sicherheitsoptionen festgelegt. Sie beginnt, wie der Namen vermuten lässt, mit der Bezeichnung **global**. Diese Bezeichnung ist reserviert und darf nur für die globale Konfiguration verwendet werden. (Es kann keine Freigabe mit diesem Namen erstellt werden). Üblicherweise müssen nur wenige Werte in dieser Sektion angepasst werden.

In der Standard-Konfiguration der **global**-Sektion fehlen ein paar wichtige Punkte, welche zwingend gesetzt werden sollten:

protocol = SMB3	Setzt die Protokoll-Version auf SMB Version 3. Damit können sich zwar ältere Clients nicht mehr verbinden, aber die Verbindung kann dafür sauber verschlüsselt werden. SMB Version 3 wird ab Windows 8 unterstützt. SMT Version 3 wird ab dem Linux Kernel 3.12 unterstützt.
lanman auth = no	Deaktiviert eine unsichere Authentifizierungsmethode
smb encrypt = required	Die Verschlüsselung des SMB Traffics wird erzwungen.
encrypt passwords = yes	Stellt sicher, dass die Passwörter verschlüsselt übertragen werden.
access based share enum = yes	Dadurch werden Freigaben ausgeblendet, wenn keine Berechtigung auf diesen Freigaben existieren. Diese Option ist standardmässig deaktiviert. Somit werden standardmässig dem Benutzer alle Freigaben auf dem SMB-Server angezeigt, auf wenn der Benutzer auf gewisse keinen Zugriff hat. Der Benutzer kann dann diese Freigaben öffnen, erhält aber einen "Zugriff verweigert" Fehler.

Dann gibt es ein paar Punkte, welche wir in der global-Sektion existieren und nicht angepasst werden müssen. Es ist aber wertvoll die Bedeutung dieser Punkte zu verstehen.

workgroup = SAMBA	Wenn der SMB-Server in einer Arbeitsgruppe verwendet wird, müsste hier die korrekte Arbeitsgruppe definiert werden. Dies ist in unserem Fall nicht nötig.
security = user	Definiert, dass die Authentifizierung des SMB-Server gegenüber (vom Client) mittels Benutzername und Passwort erfolgt. Es gibt hier weitere mögliche Optionen: • share: Es wird kein Benutzername benötigt, aber ein Passwort pro Share. • domain: Die Authentifizierung wird über einen Domänencontroller abgewickelt. Der Benutzer muss dann keine Anmeldung vornehmen. • ADS: Die Authentifizierung wird über einen Active Directory Server vorgenommen.
passwd backend = tdbsam	Diese Direktive steuert wo Samba die Passwörter und Benutzerdaten speichert. Diese speichert Samba standardmässig in einer " TDB " (Trivial Database). Die Verwendung dieses Backends erfordert keine zusätzliche Konfiguration. Dieses Backend wird für neue Installationen empfohlen, die kein LDAP benötigen. Wird die Option ldapsam verwendet, dann werden die Benutzerdaten in einem ldap Verzeichnis gespeichert.

Sektion [homes]

Über die homes-Sektion können die Heim-Verzeichnisse der Benutzer aktiviert werden. Jeder Benutzer erhält dadurch automatisch Zugriff auf das Verzeichnis mit seinem Namen.

Die Homes-Verzeichnisse liegen in Linux unter /home und werden automatisch angelegt, wenn ein neuer Benutzer mit **adduser** hinzugefügt wird. Dieser Vorgang ist im Abschnitt "Benutzer hinzufügen" genauer beschrieben.

Standardmässig sieht diese Sektion wie folgt aus:

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
inherit acls = Yes
```

Nachfolgend die Erklärung zu den einzelnen Direktiven:

comment = Home Directories	Dies ist ein Kommentar der dieser Freigabe mitgegeben wird.
valid users = %S	Definiert, welche Benutzer auf Ihre Heim-Verzeichnisse zugreifen können. Der Parameter %S wird bei einer Anmeldung automatisch mit dem Benutzername ersetzt. Alternativ kann auch die folgende Option gesetzt werden: %S, %D%w%S D: Name der Domäne oder Arbeitsgruppe des aktuellen Benutzers. w: Winbind Separator. S: Benutzername Dies wird benötigt, wenn die Anmeldung mit dem vollen Domänen-Namen erfolgen soll. In unserem Fall ist dies aber nicht relevant.
browseable = No	Diese Direktive verhindert nicht, wie angenommen werden könnte, dass der Benutzer in seiner Freigabe browsen kann, sondern, dass die Freigabe doppelt sichtbar ist. Einmal unter /homes und einmal unter /benutzername
read only = No	Der SMB-Benutzer erhält Schreibrechte auf diese Freigabe Achtung: alternativ kann auch die Variante writeable = yes verwendet werden.
inherit acls = Yes	Sie kann verwendet werden, um sicherzustellen, dass, wenn Standard-Zugriffsrechte auf übergeordneten Verzeichnissen existieren, diese immer beachtet werden, wenn eine neue Datei oder ein Unterverzeichnis in diesen übergeordneten Verzeichnissen angelegt wird.

Verzeichnis für die Freigabe erstellen

Definieren Sie ein Verzeichnis, in dem Sie die Unterverzeichnisse für die Freigaben einrichten. Erstellen Sie das Verzeichnis /shared im Verzeichnis /var

```
mkdir -p /var/shared
```

Neue Freigabe einrichten

Die oben besprochene Home-Freigabe (Benutzer / Heimatverzeichnis) ermöglichen es dem Benutzer, seine Daten zentral auf dem Server zu organisieren, jedoch bietet diese keine Möglichkeit zum Datenaustausch unter mehreren Benutzern. Die Verzeichnisse sind für andere Benutzer unsichtbar und unerreichbar. Eine Lösung hierfür sind

Eine Freigabe wird eingerichtet, indem ein weitere Block der Konfiguration hinzugefügt wird:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
```

Standardmässig erhalten alle Benutzer des File-Servers Zugriff, da der Zugriff nicht auf bestimmte Benutzer eingeschränkt ist. Dies kann beispielsweise verwendet werden, um eine unternehmensweite Freigabe zu erstellen.

Gruppenfreigabe

Unter valid users können eine oder mehrere berechtigte Gruppen angegeben werden.

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname
```

Will man mehreren Gruppen den Zugriff erlauben, dann müssen die Gruppen kommasepariert hinterlegt werden:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname1, @gruppenname2
```

Wenn mehrere Gruppen auf eine Freigabe Zugriff haben, muss sichergestellt werden, dass wenn eine neue Datei erstellt wird, die Berechtigung von der Hauptgruppe übernommen wird. (Die Gruppe der auch das Verzeichnis auf dem File-Server gehört), dies wird mit der Direktiven **force group** erreicht. Mit create mask und Directory mask wird zudem sichergestellt, dass die Schreib und Leseberechtigung korrekt gesetzt wird beim Anlegen einer neuen Datei oder eines neuen Verzeichnisses:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname1, @gruppenname2
    force group = +gruppenname1
    create mask = 0660
    directory mask = 0770
```

Sollen nur bestimmte Gruppen oder Benutzer Schreibberechtigung auf ein Verzeichnis erhalten, dann muss das Verzeichnis auf readonly gesetzt werden. Jene Benutzer der nun Schreibberechtigung erhalten

soll, muss separat in der Direktive **write list** freigeschaltet werden.

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = no
    write list = @gruppenname1
    valid users = @gruppenname1, @gruppenname2
    force group = +gruppenname1
    create mask = 0660
    directory mask = 0770
```

Benutzerverwaltung

Samba verwendet die Berechtigungen der eingerichteten Benutzer auf dem Betriebssystem. Nachfolgend wird beschrieben, wie die Benutzer unter Linux verwaltet werden.

Linux Benutzer hinzufügen

Als erstes muss auf dem Betriebssystem ein Benutzer angelegt werden. Dies wird mit dem Befehl **adduser** erledigt:

```
adduser BENUTZERNAME
```

Nun muss dem neuen Benutzer noch ein Samba-Passwort zugewiesen werden

```
smbpasswd -a BENUTZERNAME
```

Linux Gruppe hinzufügen

Als erstes muss auf dem Betriebssystem eine Gruppe angelegt werden. Dies wird mit dem Befehl **groupadd** erledigt:

```
groupadd BENUTZERNAME
```

Nun kann auf dem Betriebssystem ein Verzeichnis angelegt werden, in dem die Daten dieser Gruppe abgelegt werden:

```
mkdir -p /var/shared/GRUPPE_VERZEICHNIS
```

Anschliessend muss die Gruppe diesem Verzeichnis zugewiesen werden:

```
chgrp GRUPPE /var/shared/GRUPPE_VERZEICHNIS
```

Damit existiert auf dem Betriebssystem nun eine Gruppe mit einem eigenen Verzeichnis. Damit dieses Verzeichnis auch als Freigabe existiert, muss dieses nun noch als Freigabe eingerichtet werden. Dazu wird ein weitere Block eingefügt:

```
[ GRUPPE]  
    path= /var/shared/GRUPPEN_VERZEICHNIS  
    writeable = yes  
    valid users = @GRUPPE
```

Nun kann die Datei gespeichert und der Dienst neu gestartet werden.

Linux Benutzer einer Gruppe hinzufügen:

Um einem Gruppe einen Benutzer hinzuzufügen, wird der Befehl **usermod** verwendet:

```
usermod -a -G gruppenname benutzername
```

Linux Benutzer aus einer Gruppe löschen:

Um einen Benutzer aus einer Gruppe zu löschen, wird der Befehl **gpasswd** verwendet:

```
gpasswd --delete gruppenname benutzername
```

Gruppen eines Benutzers anzeigen:

Wenn Sie wissen möchten, in welchen Gruppen ein Benutzer ist, können Sie den Befehl **groups** verwenden:

```
groups benutzername
```

Konfiguration testen

Mit dem Befehl **testparm** kann die Samba-Konfiguration getestet werden. Ungültige Direktiven und Probleme werden im Bericht ausgegeben:

```
testparm
```


Datei- / Verzeichnisberechtigungen

Damit der Zugriff über die Freigaben auch ordnungsgemäss funktioniert, müssen die freigegebenen auch die korrekten Berechtigungen aufweisen. Dazu müssen Sie wissen, wie Berechtigungen unter Linux aufgebaut sind.

Eine Datei oder ein Verzeichnis gehört grundsätzlich einem **Benutzer** und einer **Gruppe**. Beim Benutzer spricht man auch vom Besitzer oder englisch Owner. Nachfolgend sehen Sie ein die Ausgabe, wen Sie den Befehl **ls -la** ausführen.

```
[root@ip-172-31-68-46 shares]# ls -la
drwxrw-r--+ 4 miten gruppenname 46 Jan 17 23:20 verzeichnis_name
  ^      ^      ^
  |      |      |
  |  |      |      o- Name des Verzeichnisses oder der Datei
  |  |      |      o----- Name der Gruppe, welche dem Verzeichnis
  |  |      |      zugeordnet ist
  |  |      |      o----- Name des Benutzers, welchem das Verzeichnis
  |  |      |      zugrordnet ist
```

Jedes Verzeichnis und jede Datei hat immer zwingend ein Benutzer und eine Gruppe. Wurde diese vom Root-Benutzer mit der Root-Gruppe erstellt, so sieht die Ausgabe beispielsweise so aus:

```
[root@ip-172-31-68-46 var]# ls -la
```

drwxr-xr-x	2	root	root	19	Dec	15	21:54	account
drwxr-xr-x	2	root	root	6	Apr	9	2019	adm
drwxr-xr-x	6	root	root	63	Dec	15	21:54	cache

Die Berechtigung lässt sich nun über drei Dimensionen steuern:

- Besitzer
- Gruppe

- Andere

Betrachten wir die drei Dimensionen etwas genauer:

Besitzer

Der Besitzer einer Datei oder einem Verzeichnis ist in der Regel der Ersteller der Datei. Die Datei ist primär diesem Benutzer zugeordnet, sprich gehört diesem Benutzer. **Der Besitzer kann immer die Berechtigung steuern / verändern.**

Es kann nun gesteuert werden, was der Benutzer mit dieser Datei oder Verzeichnis machen kann:

Datei:

- **Lesen:** Der Besitzer kann den Inhalt der Datei lesen / einsehen.
- **Schreiben:** Der Besitzer kann den Inhalt der Datei verändern.
- **Ausführen:** Der Besitzer kann eine Datei, welche ein Programm ist, ausführen.

Verzeichnis:

- **Lesen:** Der Besitzer kann den Inhalt in einem Verzeichnis sehen.
- **Schreiben:** Der Besitzer kann im Verzeichnis neue Unterverzeichnisse oder Dateien erstellen
- **Ausführen:** Der Besitzer kann in das Verzeichnis wechseln

Sie können sich als Besitzer selber die Leserechte auf ein Objekt nehmen. Dann können Sie die betreffende Datei nicht mehr öffnen. Als Inhaber können Sie die Berechtigung aber jederzeit ändern, somit können Sie sich bei eigenen Dateien nicht aussperren.

Gruppe

Eine Datei kann auch einer speziellen Gruppe zugeordnet werden. Alle Benutzer, welche sich in dieser Gruppe befinden, haben analog den gesetzten Berechtigungen Zugriff auf die Dateien oder Verzeichnisse. **Ein Gruppenmitglied kann aber die Berechtigung einer Datei nicht anpassen, insofern das Gruppenmitglied nicht auch Besitzer ist.**

Jeder Benutzer hat auch immer eine Gruppe, welche auf seinen Namen lautet. Alle seine Dateien sind standardmässig in dieser Gruppe, wenn keine andere Gruppe zugewiesen ist.

Die möglichen Berechtigungen eines Gruppenmitglieds sind identisch mit den Besitzer-Berechtigungen:

- **Lesen**
- **Schreiben**
- **Ausführen**

Andere

Zu guter Letzt, kann noch definiert werden, wie sich eine Datei bei einem Benutzer verhält, welcher nicht Besitzer der Datei oder ein Mitglied der Gruppe ist. Dies kann über die Andere-Berechtigung

gesteuert werden. Es gelten wieder die selben Berechtigungen wie beim Inhaber einer Datei oder eines Verzeichnisses:

- **Lesen**
- **Schreiben**
- **Ausführen**

Berechtigungen setzen

Diese Berechtigungen können **beliebig** zusammengesetzt werden. Es gibt hier keine Einschränkung.

Ich kann als Besitzer einer Datei z. B. folgende Berechtigung einer Datei setzen:

- **Besitzer** : Kann die Datei weder lesen, noch bearbeiten oder ausführen.
- **Gruppe**: Ein Gruppenmitglied kann die Datei nicht lesen, aber den Inhalt verändern.
- **Andere**: Jeder andere, welcher nicht Besitzer oder Mitglied der Gruppe ist, kann die Datei lesen.

Wie äussert sich dies nun unter realen Bedingungen?

Der Besitzer kann die Datei weder öffnen noch mit Inhalt beschreiben. Er wird immer ein **Permission denied** (Zugriff verweigert) erhalten.

Ein Mitglied der Gruppe kann die Datei beschreiben, aber nicht lesen. Öffnen das Gruppenmitglied z. B. eine Datei mit nano, dann wird die Datei leer angezeigt, (obwohl eigentlich Inhalt darin ist), das Gruppenmitglied kann nun Inhalt in die Datei schreiben. Beim Speichern wird dann der bereits enthaltene Inhalt mit dem neuen überschrieben.

Ein Benutzer, welcher nicht Besitzer oder Mitglied der Gruppe ist, kann die Datei lesen. Öffnet er die Datei z. B. mit dem Editor nano, dann sieht der Benutzer den Inhalt, kann aber keinen neuen Inhalt hinzufügen.

Wenn Sie nun für diese drei Dimensionen die möglichen Berechtigungen einsetzen, erhalten Sie drei Gruppen, welche jeweils die Berechtigung **L**esen, **S**chreiben und / oder **A**usführen haben können.

Besitzer	Gruppe	Andere
L S A	L S A	L S A

Wenden wir nun die Englischen begriffe für **R**ead, **W**rite und **E**xecute an, dann entsteht folgendes Bild:

Besitzer	Gruppe	Andere
R W X	R W X	R W X

Nun möchten Sie dem Besitzer, der Gruppe und den Anderen die Schreibeberechtigung wegnehmen. Wir streichen also überall das **W**rite.

Besitzer	Gruppe	Andere
R W X	R W X	R W X

In der reinen Darstellung als Text wird eine gestrichene Berechtigung, bez eine nicht gesetzte Berechtigung mit einem Bindestrich dargestellt.

Besitzer	Gruppe	Andere
R - X	R - X	R - X

Geben wir nun dem Besitzer die Schreibeberechtigung wieder zurück, nehmen dafür der Gruppe die Ausführberechtigung und den Anderen alle Berechtigungen, so würde dies folgendes Bild ergeben:

Besitzer	Gruppe	Andere
R W X	R - -	- - -

Nun wurde weiter oben beschrieben, dass sich die Berechtigungen auf Verzeichnisse etwas anders auswirken als auf Dateien. Daher markieren Sie ein Verzeichnis mit einem **D** für Directory und eine Datei mit einem Bindestrich.

	Besitzer	Gruppe	Andere
D	R W X	R - -	- - -
	Besitzer	Gruppe	Andere
-	R W X	R - -	- - -

Nehmen Sie nun nur den Text und Sie haben eine einfache Möglichkeit, die Berechtigung einer Datei oder eines Verzeichnisses abzulesen.

-RWXR--
-RWXR--

Schauen Sie sich nun den **ls -la** vom Anfang nochmals genau an. Sie können nun ablesen, was für eine Berechtigung das Verzeichnis `verzeichnis_name` hat:

```
[root@ip-172-31-68-46 shares]# ls -la
drwxrw-r--  4 miten gruppenname 46 Jan 17 23:20 verzeichnis_name
```

- **Typ** = d = Verzeichnis
- **Besitzer** = rwx = Lesen, Schreiben und Ausführen
- **Gruppe** = rw- = Lesen, Schreiben
- **Andere** = r-- = Lesen

Berechtigungen setzen

Es gibt verschiedene Möglichkeiten eine Berechtigung zu setzen. In jedem Fall wird aber der Befehl **chmod** (**Ch**ange **Mod**us) verwendet. Wir schauen uns in diesem Modul nur die Möglichkeit über rwx die Berechtigung zu setzen an.

Angenommen Sie haben eine Datei mit dem Namen **datei**, welche noch gar keine Berechtigungen hat.

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-----  1 miten miten      0 Jan 18 22:52 datei
```

Nun möchten Sie dem Besitzer die Berechtigung zum Lesen, Schreiben und Ausführen geben, also **rwX**. Verwenden Sie nun den Befehl **chmod** und geben Sie als erstem Parameter die gewünschte Berechtigung an.

u = user = Besitzer

rwX = Berechtigung Lesen, Schreiben und Ausführen

Als zweiten Parameter wird die gewünschte Datei angegeben. Somit sieht der vollständige Befehl wie folgt aus:

```
chmod u=rwX datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwx-----  1 miten miten      0 Jan 18 22:52 datei
```

Wie sie sehen, hat der Besitzer nun die rwx (Lesen, Schreiben, Ausführen) Berechtigungen erhalten. Nun soll die Gruppe die Berechtigung zum Lesen und Schreiben erhalten.

Dafür wird wieder der Befehl **chmod** verwendet:

g = group = Gruppe

rw = Berechtigung Lesen und Schreiben

```
chmod g=rw datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwxrw--- 1 miten miten      0 Jan 18 22:52 datei
```

Zu guter Letzt, sollen alle anderen auf die Datei die Berechtigung Lesen und Ausführen erhalten:

o = other= Andere

rx = Berechtigung Lesen und Ausführen

```
chmod o=rx datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwxrw-r-x 1 miten miten      0 Jan 18 22:52 datei
```

Um etwas schneller vorwärts zu kommen, können Sie die Berechtigungen auch kombinieren:

```
chmod u=rwx,g=rw,o=rx datei
```

Um alle Berechtigungen komplett zu entfernen, setzen Sie einfach keine Berechtigung, also z. B.

```
$ chmod u=rwx,g=,o= datei
$ ls -la
-rwx----- 1 miten miten      0 Jan 18 22:52 datei
```

Typische Berechtigungen

Innerhalb von Samba werden vor allem folgende Berechtigungen benötigt:

Freigabe	Beschreibung	chmod-Befehl
Share (Alle haben Zugriff)	Alle Berechtigungen (rwx) werden für den Besitzer, Gruppe und die Anderen gesetzt.	chmod u=rwx,g=rwx,o=rwx verzeichnis
Gruppen-Freigabe	Der Besitzer und die Gruppen erhalten die volle Berechtigung (rwx). Die anderen erhalten keine Berechtigung (---).	chmod u=rwx,g=rwx,o= verzeichnis
Persönliche Freigabe	Nur der Besitzer erhält die volle Berechtigung (rwx). Die Gruppe und die Anderen erhalten keine Berechtigung (---)	chmod u=rwx,g=,o= verzeichnis

Besitzer ändern / zuweisen

Mit dem Befehl **chown** (**Ch**ange **O**wner) können Sie eine Datei oder ein Verzeichnis einem neuen Benutzer zuordnen. Achtung, eine Datei kann immer nur einem Benutzer gehören. Nur der Root-Benutzer kann die Inhaberschaft einer Datei oder eines Verzeichnisses ändern.

```
# ls -la
drwxrwxrwx 2 miten miten 19 Jan 18 22:55 Unterverzeichnis
```

Angenommen Sie möchten den Inhaber vom Unterverzeichnis auf smeier ändern, dann setzen Sie folgenden Befehl ab:

```
chown smueller Unterverzeichnis
```

Nun hat der Besitzer geändert. Achtung die Gruppe bleibt so unverändert.

```
# ls -la
drwxrwxrwx 2 smueller miten 19 Jan 18 22:55 Unterverzeichnis
```

Gruppe ändern / zuweisen

Mit dem Befehl **chgrp** (**Ch**ange **G**roup) können Sie eine Datei oder ein Verzeichnis einem neuen Benutzer zuordnen.

Nehmen Sie als Ausgangslage das Unterverzeichnis.

```
# ls -la
drwxrwxrwx 2 smueller miten 19 Jan 18 22:55 Unterverzeichnis
```

Nun möchten Sie neu die Gruppe Testgruppe hinterlegen.

```
chgrp testgruppe Unterverzeichnis
```

```
# ls -la
drwxrwxrwx 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Spezialfall: Mehrere Gruppen pro Verzeichnis

Standardmässig hat eine Datei oder ein Verzeichnis nur eine Gruppe. Diese kann mit `chgrp` geändert werden.

Da aber z. B. mit Samba mehrere Freigaben mit mehreren Gruppen erstellt werden können, benötigen wir eine Möglichkeit mehrere Gruppen einem Verzeichnis zuzuweisen. Dafür können wir die Funktion ACL (Access Control List) verwenden. Damit ist es möglich, einem Verzeichnis mehre Gruppen zuzuweisen:

Hinweis: Definieren sie immer eine Hauptgruppe, welche Sie dem Verzeichnis zuweisen. Die restlichen Gruppen werden anschliessend über die ACL hinzugefügt.

Beispiel: Das Unterverzeichnis gehört der Gruppe **testgruppe**. Diese wurde mit `chgrp` hinzugefügt.

```
# ls -la
drwxrwxrwx 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Es benötigt nun aber noch eine zweite Gruppe mit dem Namen **video** Zugriff auf das Unterverzeichnis. Diese wird nun mit dem Befehl `setfacl` hinzugefügt:

```
setfacl -m g:video:rwx Unterverzeichnis
```

Erklärung des vollen Befehls:

setfacl	
-m	modify = verändern , damit möchten wir die ACL eines Verzeichnisses verändern.
g:video:rwx	g = group (Wir möchten eine Gruppe hinzufügen) video = Name der Gruppe die hinzugefügt werden soll rwx = Berechtigung die gesetzt werden soll (Lesen, Schreiben, Ausführen)
Unterverzeichnis	Name des Verzeichnisses, welches wir verändern wollen.

Wenn wir nun einen ls -la machen, sieht die Ausgabe immer noch fast identisch aus:

```
# ls -la
drwxrwxrwx+ 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Die Gruppe hat sich nicht verändert. Es hat jedoch ein + gegeben bei den Berechtigungen. Dies bedeutet, dass dieses Verzeichnis über eine ACL verfügt. Wir können diese mit dem Befehl **getfacl** einsehen:

```
getfacl Unterverzeichnis
# file: Unterverzeichnis
# owner: smueller
# group: testgruppe
user::rwx
group::rwx
group: video: rwx
mask::rwx
other::rwx
```

Man sieht nun, dass neben der Hauptgruppe testgruppe, auch eine Gruppe video mit der Berechtigung rwx existiert.

Löschen einer ACL:

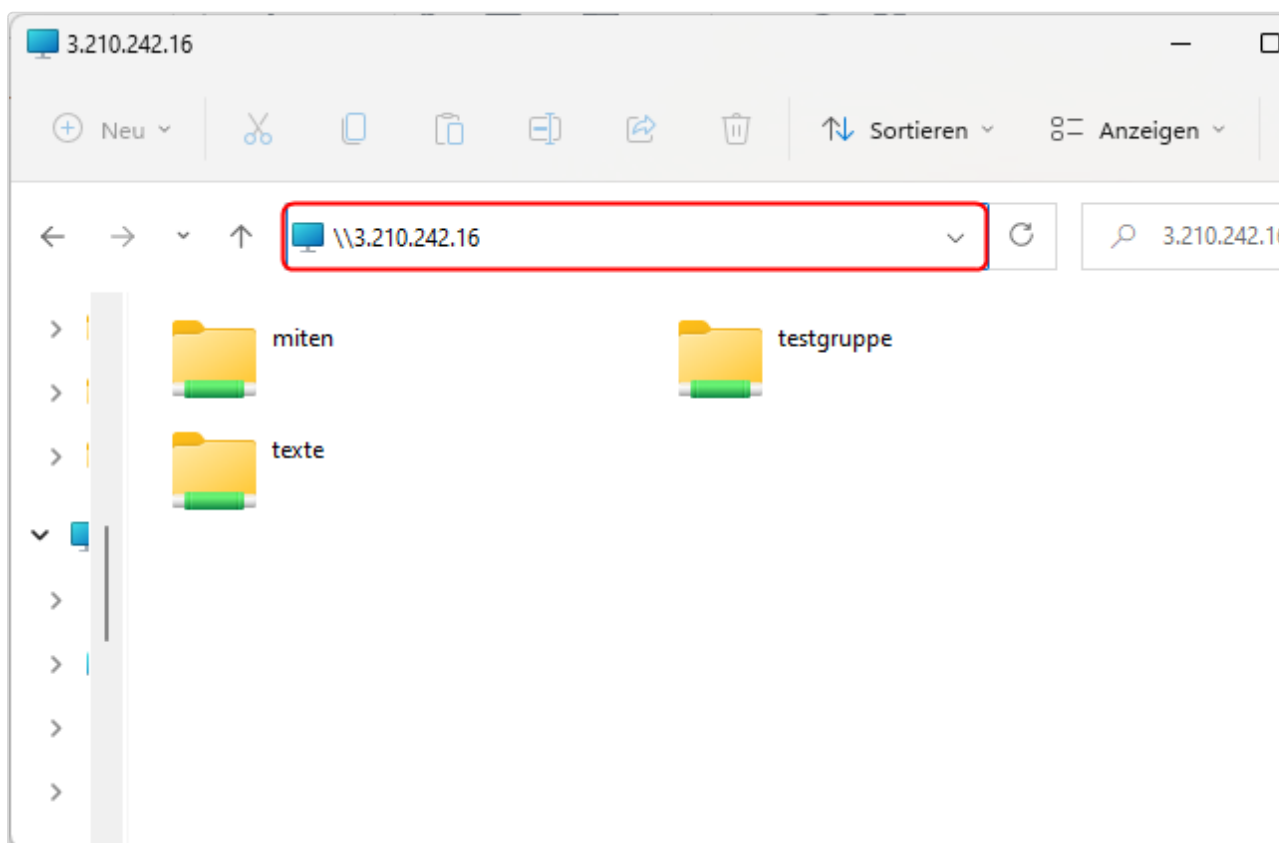
```
setfacl -x g: video Unterverzeichnis
```


SMB Freigaben unter Windows einrichten

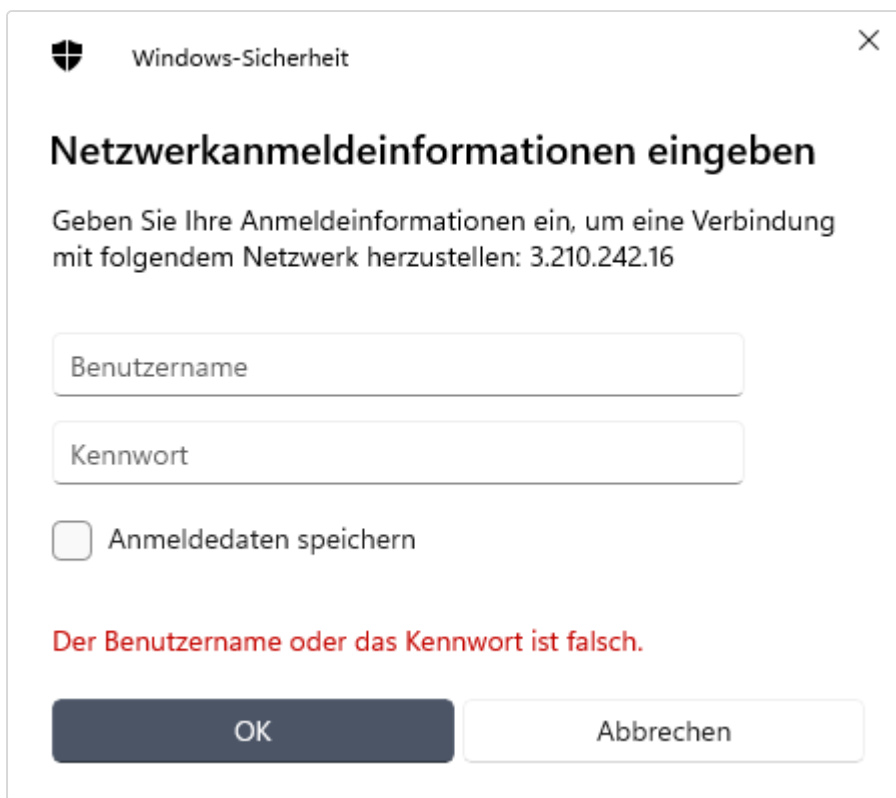
Falls Sie nur kurz eine Freigabe testen möchten, können Sie diese direkt im Windows Explorer öffnen:

Tragen Sie hierzu einfach die IP-Adresse des Servers nach folgendem Schema in der Adresszeile des Explorers ein:

```
\\ip-des-servers
```



Beim ersten Aufruf erscheint eine Aufforderung, dass Passwort und den Benutzernamen zu hinterlegen:



Windows-Sicherheit

Netzwerkanmeldeinformationen eingeben

Geben Sie Ihre Anmeldeinformationen ein, um eine Verbindung mit folgendem Netzwerk herzustellen: 3.210.242.16

Benutzername

Kennwort

☐ Anmeldedaten speichern

Der Benutzername oder das Kennwort ist falsch.

OK Abbrechen

Freigabe schliessen:

Wenn Sie das Explorer-Fenster schliessen und wieder öffnen werden Sie feststellen, dass Sie weiterhin mit dem bereits angemeldeten Benutzer angemeldet sind. Damit Windows die Verbindung sauber trennt, ist ein CMD-Befehl nötig:

```
net use * /delete
```

Danach muss das Explorer-Fenster geschlossen und einige Sekunden abgewartet werden. Danach können Sie die Freigabe wieder aufrufen und es erscheint ein neues Login-Fenster.

Um alle aktiven SMB-Verbindungen aufzulisten, können Sie den nachfolgenden Befehl verwenden:

```
net use
```

Freigabe als Laufwerk einbinden:

Beim Kunden wird die Freigabe schlussendlich als eigenes Laufwerk eingebunden:

