Inhaltsverzeichnis

Die Anforderungen ... 1
Verwendete Software ... 2
Die Hardware ... 3
Die Kosten ... 4
Systeminstallation ... 4
Ubuntu 18.04 Server herunterladen ... 5
ISO-Datei auf USB-Stick übertragen ... 5
Installation von Ubuntu 18.04 starten ... 6
Partitionieren der Festplatten ... 11
Automatische Partitionierung der Systemplatte ... 18
Abschluss der Installation von Ubuntu Server 18.04 ... 22
Exkurs: Remote Login via SSH ... 23
Login auf dem Homeserver mit Linux oder MacOS ... 24
Login auf dem Homeserver via SSH mit Windows 10 ... 24
Login auf dem Homeserver via SSH mit Windows 7 ... 26
Das System aktuell halten ... 28
Homeserver mit DynDNS aus dem Internet erreichbar machen ... 30
DynDNS mit No-IP.com ... 30
DynDNS Dienst in der FritzBox aktivieren ... 31
Dateifreigabe im Heimnetz ... 33
Schnellzugriff auf Freigaben mit Windows 10 ... 36
Freigaben unter Windows 10 dauerhaft als Netzlaufwerk einbinden ... 37
Schnellzugriff auf die Freigaben mit Ubuntu 18.04 ... 40
Dauerhaftes Einbinden der Freigaben auf dem Ubuntu Desktop-System ... 40
Installtion von NextCloud ... 42
Installation der MariaDB-Datenbank ... 43
Anlegen der Datenbank für Nextcloud ... 43
Installation des Apache Webserver ... 44
Einrichten des Let's Encrypt https-Zertifikats (Auch Eingenzertifizierungen funktionieren!) ... 45
Automatische Erneuerung der Zertifikate einrichten ... 46
Installation von Nextcloud und Konfiguration des Apache Webserver ... 46
Apache konfigurieren ... 47
Nextcloud einrichten ... 48
Media Streaming mit Plex ... 50
Die Installation von Plex Mediaserver unter Ubuntu 18.04 ... 50
SSL-Zertifikat für den Mediaserver erstellen ... 51
Plex Mediaserver konfigurieren ... 53
Backups mit Duplicati und Rsnapshot ... 54
Cloudbackup mit Duplicati ... 54
Die Installation von Duplicati ... 55
Die Konfiguration von Duplicati ... 56
Lokales Backup mir Rsnapshot ... 56
USBMount konfigurieren ... 57
Rsnapshot konfigurieren ... 57
Aktivieren der Rsnapshot Backups und erstellen des ersten Backups ... 59

Diese Anleitung ist eine Abschrift von der Seite https://www.techgrube.de/tutorials/homeserver-nas-mit-ubuntu-18-04-teil-1-einleitung-hardware-und-kosten und dient leidiglich der Sicherung der Daten. Weiterfürende Links:

https://www.pcwelt.de/ratgeber/Netzwerken_mit_Samba_-_so_geht_s-Linux-8530128.html
https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-18-04
https://wiki.ubuntuusers.de/Heimnetzwerk/ https://wiki.ubuntuusers.de/Samba_Client_cifs/
https://wiki.ubuntuusers.de/fstab/ https://wiki.ubuntuusers.de/mount/#Parameter
https://wiki.ubuntuusers.de/rm/ https://wiki.ubuntuusers.de/Festplattenstatus/
https://askubuntu.com/questions/58404/how-to-start-and-stop-a-service
https://wiki.ubuntuusers.de/systemd/Service_Units/ https://wiki.ubuntuusers.de/systemd/Units/
https://www.thomas-krenn.com/de/wiki/Festplattenbelegung_unter_Linux_in_der_Konsole_mit_df_und_du_anzeigen
https://www.thomas-krenn.com/de/wiki/Md5sum_und_sha1sum_zum_%C3%9Cberpr%C3%BCfen_von_Dateidownloads_verwenden

---

Die Anforderungen

Diesmal basiert die Artikelreihe auf Ubuntu Server 18.04. Die grundlegenden Anforderungen an das System sind dabei nahezu unverändert.

- Möglichst geringer Stromverbrauch, da das System 24/7 laufen soll.
- Schutz der Daten vor einem Festplattenausfall durch Speicherung auf einem RAID-Verbund.
- Bereitstellen von Netzwerkfreigaben zum einfachen Zugriff auf die Daten mit Windows und Linux.
- Synchronisieren bestimmter Daten über mehrere Geräte (so dass diese auch lokal vorliegen).
- Bereitstellen von zentralem Adressbuch und Kalender und Synchronisierung mit Mobilgeräten Zugriff auf die Daten von außer Haus.
- Streamen von Audio und Video über das Internet.
- Tägliches automatisches und verschlüsseltes Backup der wichtigsten Daten auf einen Cloudspeicher außer Haus.
- Zusätzliches Backup sämtlicher Daten auf einen externen Datenträger.

Noch ein paar Worte zu den Anforderungen an den eigenen Internet Anschluss: Um den Homeserver über das Internet erreichbar zu machen, muss der eigene Internetanschluss zwingend über eine öffentliche IP-Adresse verfügen. Bei DSL-Anschlüssen ist dies, soweit mir bekannt ist, immer der Fall. Durch die Knappheit der IPv4Adressen, setzten allerdings gerade die Kabelanbieter auf einen sogenannten Dual-Stack Lite (DS-Lite) Anschluss. Hier erhält man nur eine öffentliche IPv6-Adresse und eine private IPv4-Adresse, welche nicht über das Internet erreichbar ist.

Der Zugriff von außen wäre in einem solchen Fall nur über eine IPv6-Verbindung möglich. Dies scheitert in der Regel aber daran, dass man nicht immer eine IPv6-Adresse bekommt. Gerade in Mobilfunknetzen und auch bei vielen Internetprovidern ist das nicht unbedingt der Fall.

An einem Kabelanschluss mit DS-Lite wird der Zugriff über das Internet auf den Homeserver leider nicht funktionieren. Hat man einen solchen Anschluss kann evtl. ein kostenpflichtiger Dienst wie der DS-Lite / IPv6 Portmapper von feste-ip.net helfen. Damit verbindet man sich via IPv4 zum Portmapper, welcher sich wiederum via IPv6 zum heimischen Router verbindet.

Aber auch wenn kein Zugriff von außen möglich ist, kann der Server innerhalb des Heimnetzes trotzdem als Datenspeicher verwendet werden.

Verwendete Software

Als Betriebssystem kommt wie bereits erwähnt Ubuntu Server 18.04 zum Einsatz. Auch wenn Ubuntu auf dem Desktop in den vergangenen Jahren viel Kritik einstecken musste und auch ich Ubuntu nicht mehr als Desktop Betriebssystem einsetze, halte ich die Serverversion nach wie vor für ein hervorragendes OS. Ich setze das System nicht nur auf meinem Homeserver ein, sondern z.B. auch auf dem Webserver der dir gerade diese Seite ausliefert. Ubuntu Server hat sich dabei seit Jahren als sehr zuverlässiges und sehr stabiles System bewährt.

Ein Problem das sich bei der alten Artikelreihe gezeigt hat ist, dass es nahezu unmöglich ist eine allgemeingültige Anleitung zum partitionieren einer Festplatte zu geben. Dies hängt von zu vielen Faktoren ab, vor allem davon ob das UEFI im legacy/BIOS-Modus betrieben wird oder nicht. Dies zu vermitteln ist glaube ich nur teilweise gelungen.

Daher wird in dieser Artikelreihe das Betriebssystem nicht mehr auf einem Raid-Verbund installiert, sondern auf einer einzelnen SSD. Für die eigentlichen Daten die auf dem Homeserver gespeichert werden steht nach wie vor ein Raid1-Verbund aus zwei Festplatten zur Verfügung. Die geringere Ausfallsicherheit der Systemplatte ist meiner Meinung nach zu verschmerzen, da auf diesem Datenträger keine wertvollen Daten liegen. Konfigurationsdateien können bei einem Plattendefekt und einer Neuinstallation des Betriebssystems einfach aus dem Backup wiederhergestellt werden. Dafür hat dies den Vorteil, dass die automatische Partitionierung des Installationsprogramms verwendet werden kann, so dass der Benutzer hier keine Entscheidung mehr treffen muss und außerdem sichergestellt ist dass das System auch bootet.

Netzwerkfreigaben im lokalen Netzwerk werden wieder durch SAMBA bereitgestellt.

Das synchronisieren von Daten über mehrere Rechner und Mobilgeräte, der mobile Dateizugriff sowie das bereitstellen von Kalender und Adressbuch übernimmt Nextcloud, welche mit einem Apache Webserver und MariaDB Datenbank installiert wird.

Das Mediastreaming wurde mit dem Plex Mediaserver realisiert. Seit einigen Jahren nutze ich die kostenpflichtige Version dieses Projekts und bin sehr zufrieden damit.

Auch die Gründung von Let’s Encrypt hat einiges verändert. So werden in dieser Artikelreihe Let’s Encrypt Zertifikate verwendet, während in der vorherigen Version selbstsignierte Zertifikate genutzt wurden. Damit sind die Fehlermeldungen im Browser und Probleme mit Drittsoftware die Schwierigkeiten mit selbstsignierten Zertifikaten hat Vergangenheit.

Für das Cloudbackup wird wieder Duplicati eingesetzt, das erfreulicherweise nach wie vor ein sehr aktives Softwareprojekt ist.

Das zweite Backup auf eine lokale USB-Festplatte wurde diesmal über Rsnapshots realisiert. Alles in allem hoffe ich dass ein Großteil der Stolpersteine aus der alten Artikelreihe beseitigt sind.

Die Hardware

Ich setzte nach wie vor meine alte Hardware aus dem Projekt von 2015 ein und bin mit meiner damaligen Wahl sehr zufrieden. Das System hat sich wie erwartet als stromsparend und sehr zuverlässig herausgestellt. Auch die Leistung des kleinen Celeron ist mehr als ausreichend, sofern man keine FullHD Videostreams live umwandeln muss. Meine Videodateien sind mittlerweile alle in h264 kodiert, was alle meine Geräte direkt wiedergeben können, so dass hier keine Umwandlung von Seite des Homeservers nötig ist.

Grundsätzlich eignet sich jede Hardware für einen Homeserver. Meine Priorität lag bei der Hardwareauswahl auf einem stromsparenden System. In der Regel liegt man mit einer Kombination aus einer kleinen Mainboard mit relativ schwachem aber stromsparenden Prozessor leistungsmäßig immer noch deutlich über den üblichen fertig NAS Geräten.

Dementsprechend würde ich wieder mit den gleichen Ansprüchen an die Hardwareauswahl herangehen wie bereits 2015.

Mainboard und CPU. Für eine geeignete Mainboard/CPU Kombination halte ich beispielsweise das ASRock J5005-ITX* mit derzeit aktueller Intel Gemini Lake Architektur. Wichtig ist dass das Board vier SATA-Anschlüsse besitzt. Da wir eine SSD und zwei Festplatten anschließen wollen, reichen die zwei SATA-Anschlüsse die viele Mini-ITX-Boards mitbringen nicht aus. Die maximale Leistungsaufnahme des Boards liegt bei 10 Watt.

RAM. Ausgestattet wird das Board mit maximal zwei DDR4-2133 oder DDR4-2400 RAM-Riegeln, die zusammen maximal 8GB Speicher haben dürfen. Beispielsweise die Crucial CT2K4G4SFS824A 8GB*

Festplatten. Aufgrund der guten Erfahrungen die ich mit Festplatten der Western Digital Red Serie* gemacht habe, würde ich wieder Festplatten aus dieser Reihe kaufen. Benötigt werden zwei gleiche Platten für den Raid Verbund.

SSD. Für das Betriebssystem wird eine kleine SSD genutzt. Ubuntu Server belegt nur wenige Gigabyte und da auf der SSD ansonsten keine großen Datenmengen gespeichert werden, reicht hier eine kleine und günstige SSD. Beispielsweise eine SanDisk SSD Plus 120GB*.

Gehäuse. Hier entscheidet vor allem das Auge und der Geldbeutel. Das Gehäuse muss ein Mini-ITX Mainboard unterbringen und sollte Standard-ATX Netzteile unterstützen.Ich bin zufrieden mit meinem Sharkoon CA-M black*.

Netzteil. Computernetzteile arbeiten mit dem besten Wirkungsgrad, wenn sie gut ausgelastet sind. Dementsprechend sollte ein hochwertiges Netzteil mit relativ geringer Leistung gekauft werden. Ich habe gute Erfahrungen mit Netzteilen von Be Quiet! gemacht. Ein 300Watt Netzteil ist für unsere Anforderungen völlig überdimensioniert, hochwertige Netzteile mit geringerer Maximalleistung sind allerdings kaum noch zu finden.Somit könnte man z.B. ein Be Quiet! Pure Power 10 ATX 300W* kaufen.

Kleinteile: ggf. werden noch Kleinteile benötigt, wie z.B.

- 3x SATA-Kabel zum Anschließen der Datenträger am Mainboard. Wer ebenfalls das oben angegebene Gehäuse nutzt, muss Kabel mit geraden Steckern kaufen.
- Adapter für Stromstecker. Wer das oben genannte Gehäuse und Netzteil einsetzt, benötigt einen Adapter für für die Stromstecker. Die abgewinkelten Stecker des Netzteils können nicht

direkt verwendet werden , da die Festplatten zu dicht über dem Gehäuseboden angebracht sind.
- Klettband zum befestigen der SSD an der Gehäusewand.

Die Kosten

Die Kosten unseres selbstgebauten Homeservers/NAS setzen sich aus zwei Faktoren zusammen. Die einmaligen Kosten, die bei der Anschaffung anfallen und den laufenden Kosten die durch den Stromverbrauch entstehen. Die Anschaffungskosten hängen natürlich sehr davon ab welche Hardware gekauft wird, bzw. ob vielleicht schon Hardware da ist, die einfach weitergenutzt werden kann. Im folgenden Rechenbeispiel wird die oben beschriebene Hardware gekauft.

Bauteil	**Kosten €**
Mainboard inkl. Prozessor: ca.	125.00
RAM: ca.	80.00
2x WD Red Festplatte 4TB:	235.00
SSD:	35.00
Gehäuse	60.00
Netzteil	50.00
Total	**585.00 €**

Dies scheint im Vergleich zu einem fertig NAS-System relativ viel Geld zu sein. Allerdings muss man berücksichtigen dass bei einem fertigen NAS die Festplatten nicht enthalten sind. Rechnet man die Platten heraus, bleibt ein Betrag von 350€, der zum Vergleich mit einem anderen Gerät herhalten muss. Ein NAS-System in einer ähnlichen Leistungsklasse ist definitiv auch preislich in der selben Kategorie angesiedelt.

Allerdings geht einem mit einem fertigen System der Bastelspaß, der Lerneffekt und die Flexibilität verloren.

Der zweite Faktior ist der Stromverbrauch. Mein derzeitiges und Hardwaremäßig vergleichbares System verbraucht im Leerlauf und mit drehenden Festplatten ca. 26Watt. Wenn sich das Gerät im Dauerbetrieb befindet belaufen sich die Stromkosten bei einem Strompreis von 0,28€/kWh auf ca. 5,30€/Monat oder 64€/Jahr. Man wird also auch bei den Stromkosten nicht arm. Besonders wenn der Homeserver z.B. ein Dropbox Abo für 10€ im Monat ersetzt.

Systeminstallation

Im ersten Teil der Reihe selbstgebauter Homeserver/NAS mit Ubuntu 18.04 geht es um die Installation des Betriebssystems. Für viele ist dies wahrscheinlich der einfachste Teil und in weiten Teilen ist die Installation selbsterklärend.

Allerdings hat Ubuntu für die Serverversion mit 18.04. ein neues Installationsprogramme eingeführt, leider ist dieses noch etwas unfertig und deutlich eingeschränkter als die bisherige Variante. So ist es mit der neuen Version derzeit noch nicht möglich bei der Installation einen Raidverbund aus mehreren

Festplatten zu erstellen.

Dies haben wir in diesem Artikel allerdings vor. Aus diesem Grund bietet Ubuntu weiterhin eine Version mit dem klassischen Installer an, die in diesem Artikel auch verwendet wird. Ubuntuusers.de verlinkt auf seiner Downloadseite nur diese Version mit dem klassischen Installer. Wenn man das Installationsmedium von einer anderen Seite herunterlädt, z.B. der offiziellen Seite ubuntu.com, muss man darauf achten die richtiger Version herunterzuladen.

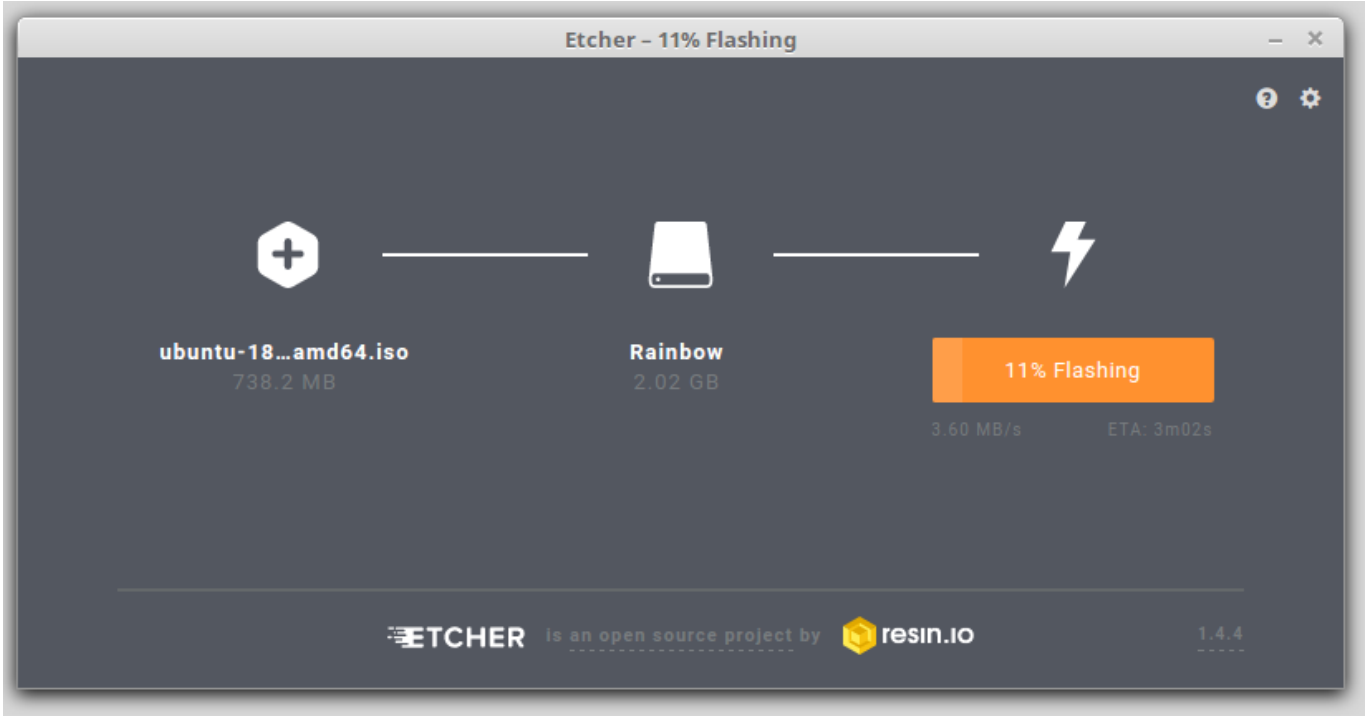
Besonders zur Partitionierung der Festplatten gab es immer wieder Fragen und Unklarheiten, außerdem soll auch dieses Tutorial wieder einsteigerfreundlich sein, so dass ich wieder eine Schritt-für-Schritt Anleitung zur Installation schreibe.

Ubuntu 18.04 Server herunterladen

Ubuntu ist freie Software und kann einfach heruntergeladen werden. Im deutschsprachigen Raum am besten von ubuntuuser.de. Auf der Downloadseite wählt man die 64-Bit Version der Server Edition. Der Download besteht aus einer .iso Datei, welche mit einem geeigneten Programm auf einen USB-Stick kopiert werden kann. Anschließend kann man den Homeserver von diesem USB-Stick booten und das System kann installiert werden.

ISO-Datei auf USB-Stick übertragen

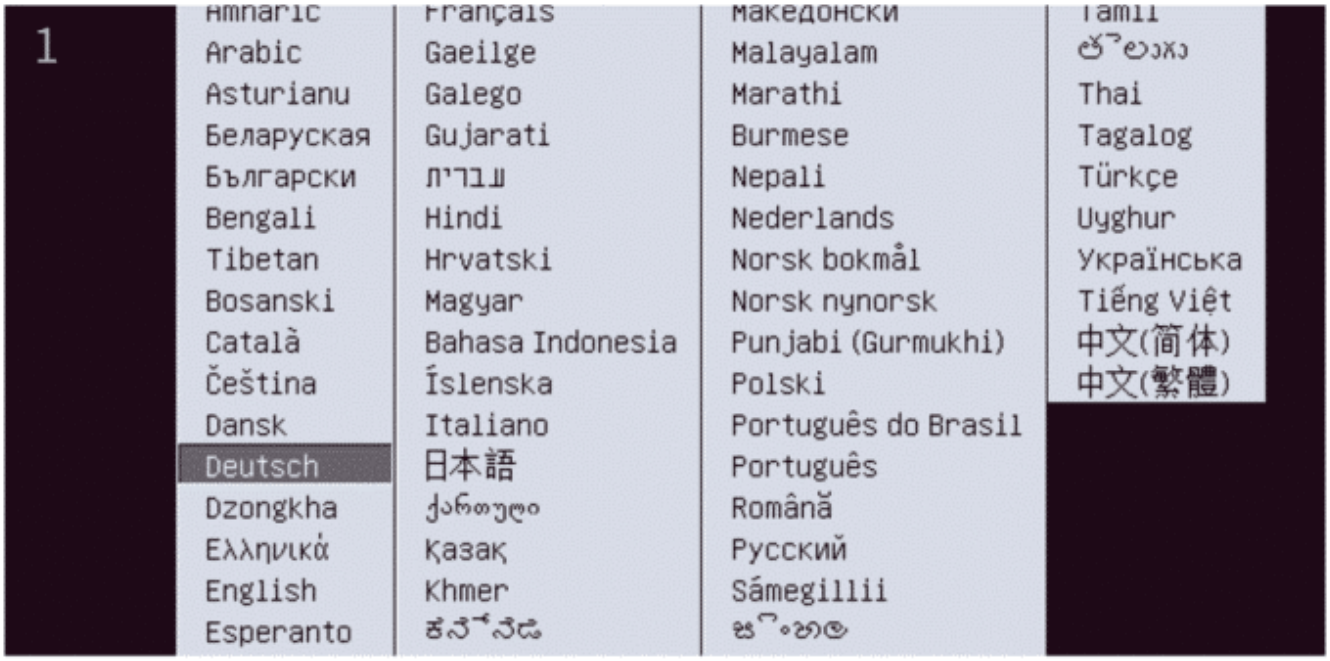
Die ISO-Datei kann leider nicht einfach auf den USB-Stick kopiert werden. Ein einfaches und plattformübergreifendes Programm zum Erstellen eines bootfähigen USB-Sticks und zum Übertragen der ISO-Datei ist Etcher. Etcher steht kostenlos zum Download bereit und ist sowohl für Linux als auch für Windows und MacOS verfügbar. Auf etcher.io kann man sich die passende Version für sein Betriebssystem herunterladen.



Installation von Ubuntu 18.04 starten

Nachdem der USB-Stick erfolgreich erstellt wurde kann dieser am Homeserver eingesteckt werden. Nun muss der Homeserver vom USB-Stick gestartet werden. Entweder geschieht dies beim Starten automatisch, oder es muss der USB-Stick als Bootmedium ausgewählt werden. Wie dies funktioniert ist von Computer zu Computer unterschiedlich. Normalerweise wird beim Starten aber angezeigt mit welcher Taste das Bootmedium gewählt werden kann, bzw. mit welcher Taste man das Bios/UEFI erreicht um die Einstellung vorzunehmen.

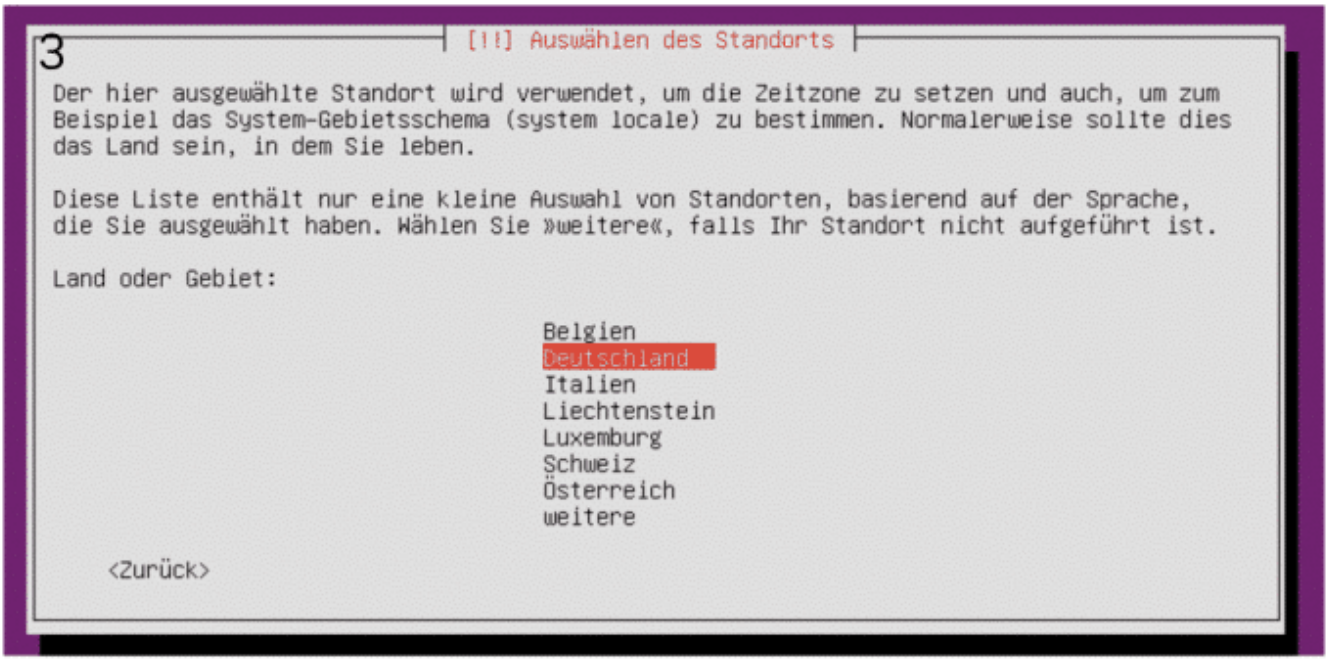
Zuerst begrüßt einen der Installer mit einem Bildschirm zur Sprachauswahl, für diese Reihe wurde Deutsch gewählt. Nun kommt eine Auswahl mit mehreren Aktionen die mit dem USB-Stick ausgeführt werden können. Wir wählen Ubuntu Server installieren.

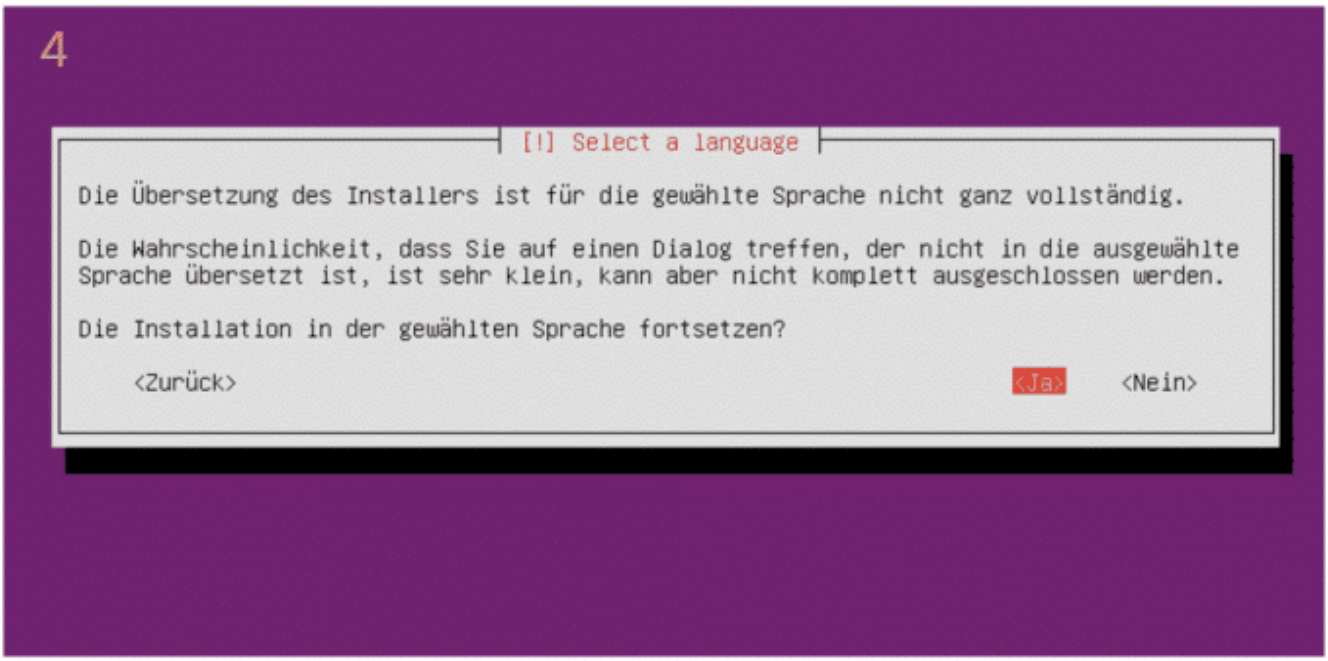




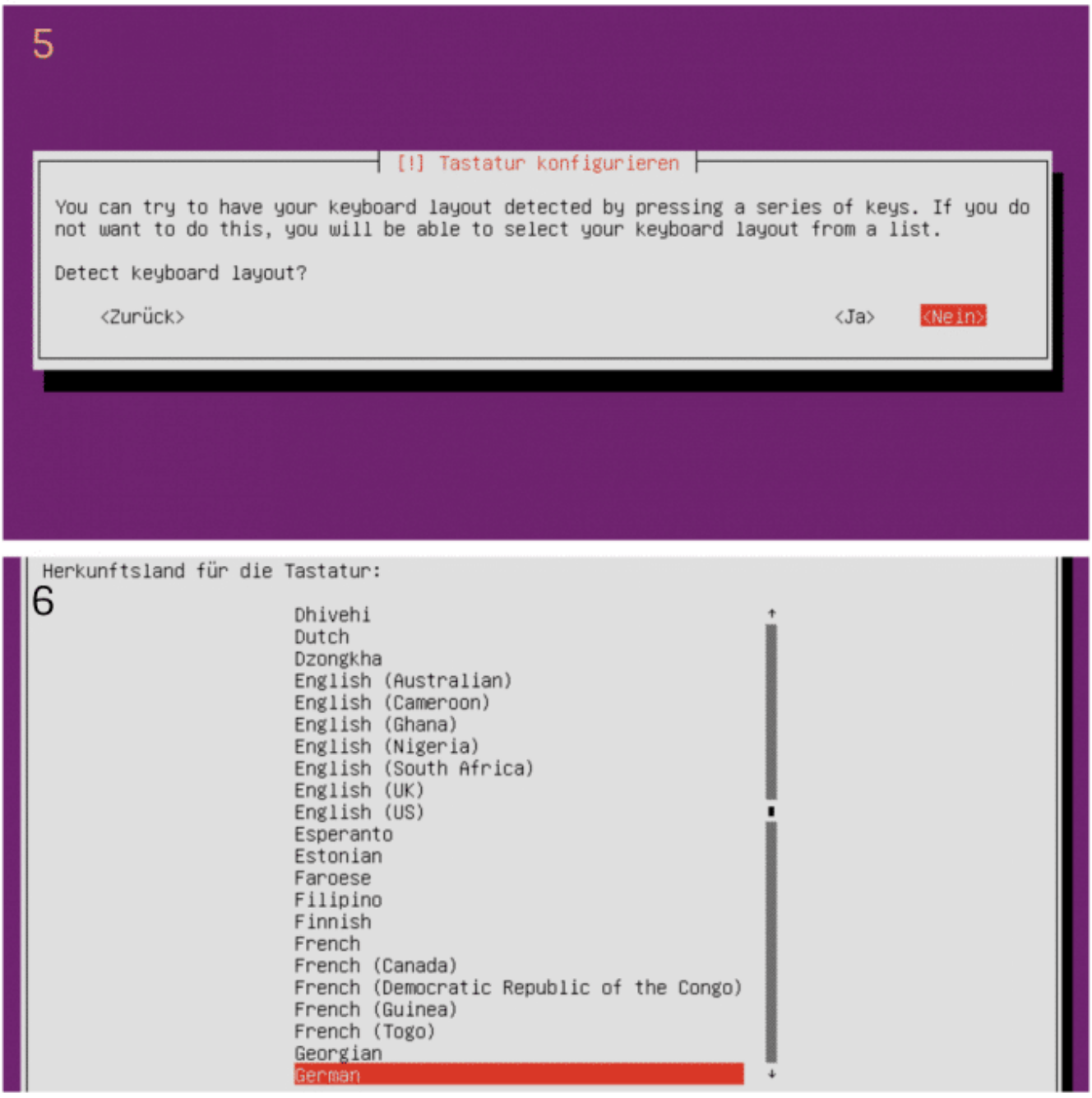
Es folgt ein Hinweis, dass die Übersetzung ins deutsche möglicherweise unvollständig ist. Der Hinweis

kann mit Ja übersprungen werden. Zur Bestimmung der korrekten Zeitzone muss nun eine Region ausgewählt werden. Es wird wieder Deutschland ausgewählt.

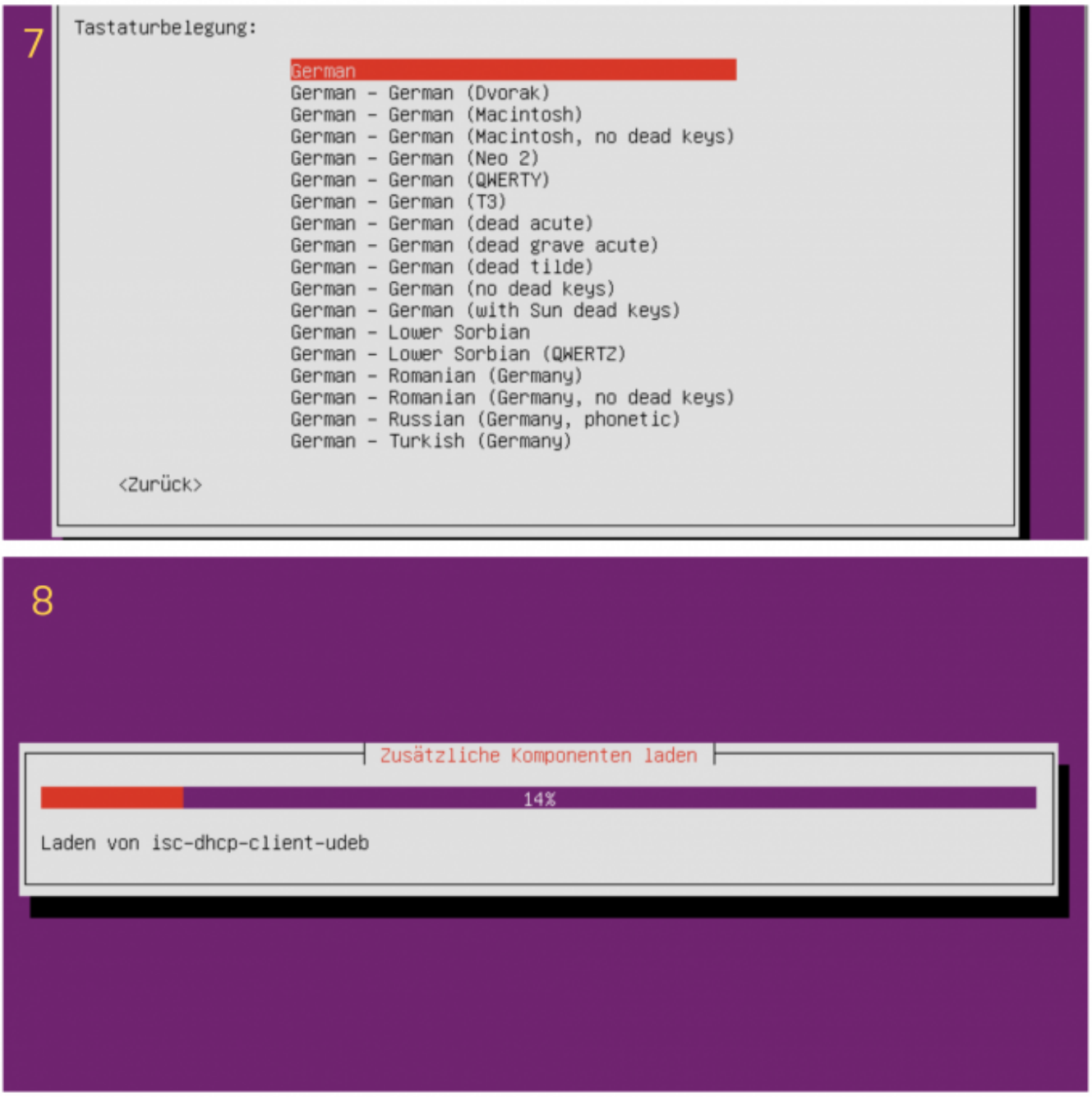




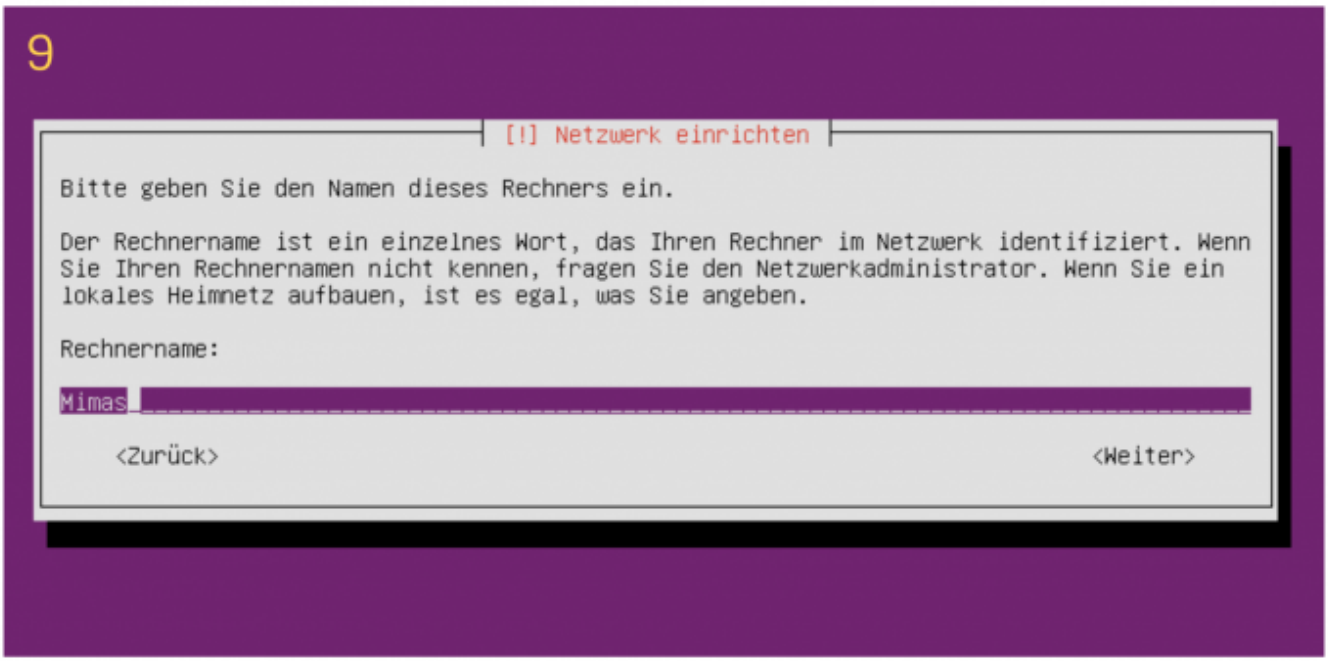
Nun folgt die Konfiguration der Tastatur. Das System fragt zuerst ob das Tastaturlayout automatisch bestimmt werden soll, indem verschiedene spezielle Tasten gedrückt werden, z.B. Umlaute und Sonderzeichen. Daran erkennt das System welches Layout die Tastatur hat. Wer weiß dass er eine normale Tastatur mit deutschem Layout angeschlossen hat kann diesen Punkt mit Nein überspringen. Im nächsten Schritt gibt man als Herkunftsland der Tastatur German an.

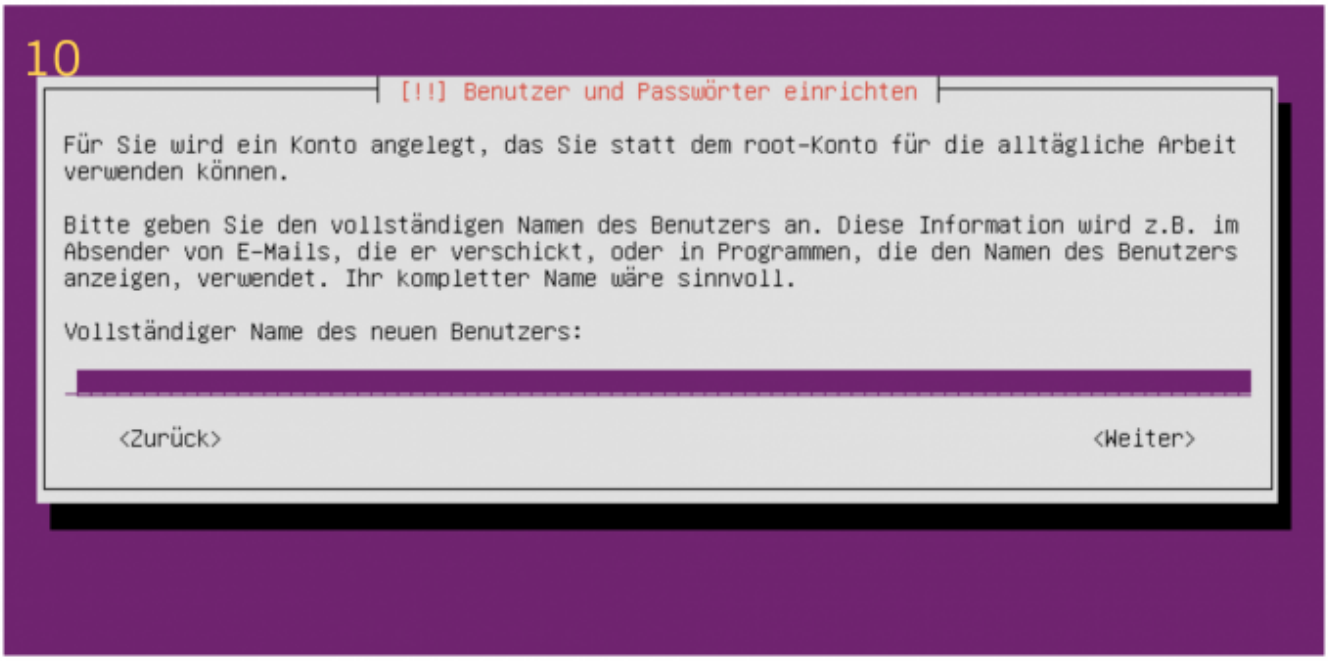


Im folgenden Schritt kann das genaue Tastaturlayout angegeben werden. Es sollte ausreichen hier wieder den Eintrag German zu wählen, ohne irgendwelche Zusätze. Nun kann man sich einen Moment zurücklehnen, da der Installer eine Hardwareerkennung ausführt und zusätzliche Komponenten nachlädt.

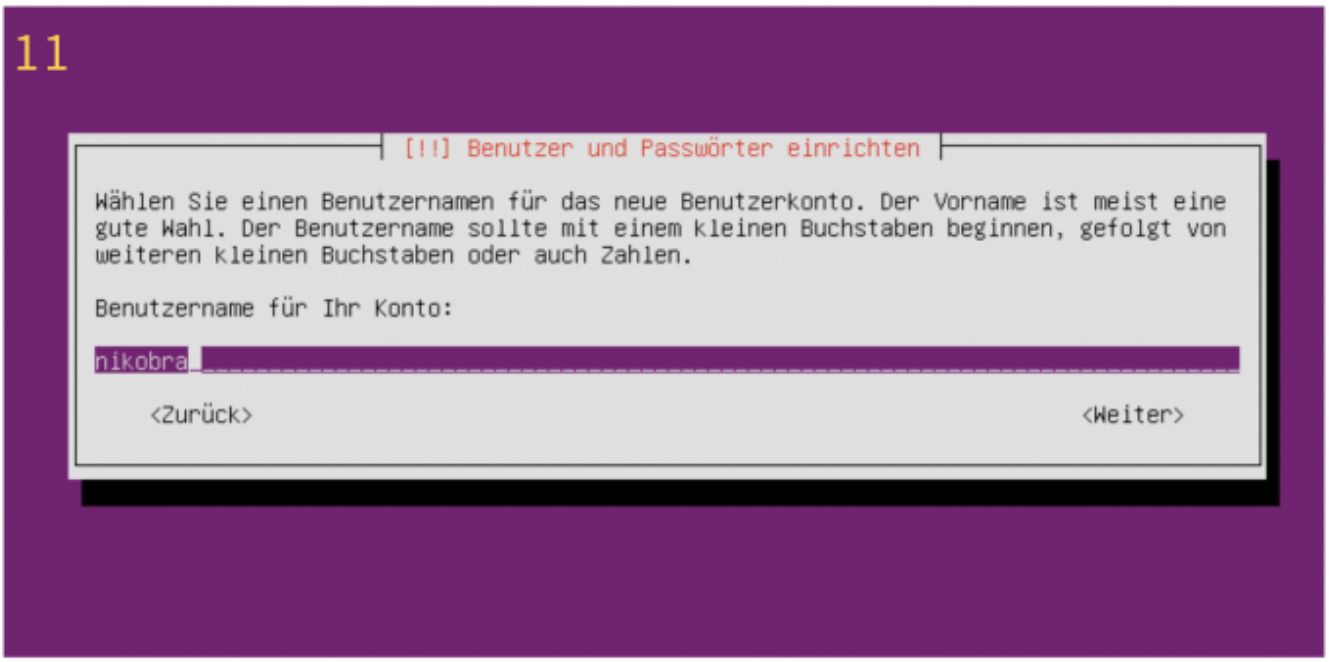


Nach einer kurzen Pause fragt das System welcher Rechnername verwendet werden soll. Hier kann ein beliebiger Name verwendet werden. Da ich meine Rechner nach den Monden des Saturn benenne, wähle ich in diesem Fall den Namen Mimas. Anschließend wird der Administratoraccount des Systems angelegt. Mit diesem Account loggt man sich zukünftig auf das System ein, um Installations- und Wartungsaufgaben vorzunehmen. Der Installer fragt zuerst nach dem vollständigen Namen. Allerdings ist es unerheblich ob man hier seinen vollen Namen oder irgendeinen Nickname oder etwas anderes einträgt. Dies ist nicht der Name der zum einloggen verwendet wird.





Die nächste Abfrage ist wichtiger. Hier wird der Benutzername abgefragt. Dies ist der Name, der zum einloggen benötigt wird. Außerdem muss ein Passwort für diesen Benutzer vergeben werden.





Nun fragt der Installer noch ob die Zeitzone korrekt ist, die aus der Angabe des Landes abgeleitet wurde. Diese sollte korrekt sein und somit kann der Dialog mit Ja beantwortet werden.

Partitionieren der Festplatten

Nachdem die allgemeinen Fragen beantwortet wurden, geht es jetzt an das Einrichten der Festplatten. Problematisch an der Partitionierung ist, dass der Datenträger je nach verwendetem Mainboard unterschiedlich partitioniert werden muss. Je nachdem ob das Mainboard noch über ein BIOS verfügt, oder das modernere UEFI verwendet. Aber auch wenn ein UEFI genutzt wird kann dieses noch in einem Bios-Kompatibilitätsmodus betrieben werden. Ein Problem mit der alten, auf Ubuntu 14.04 basierenden Anleitung war genau dieses Problem. Bei vielen Lesern hat meine Partitionierung nicht funktioniert, da das BIOS/UEFI-Setup ein anderes war. Zusätzlich verkompliziert wurde das Setup durch die Installation des Betriebssystems auf einem Raid.

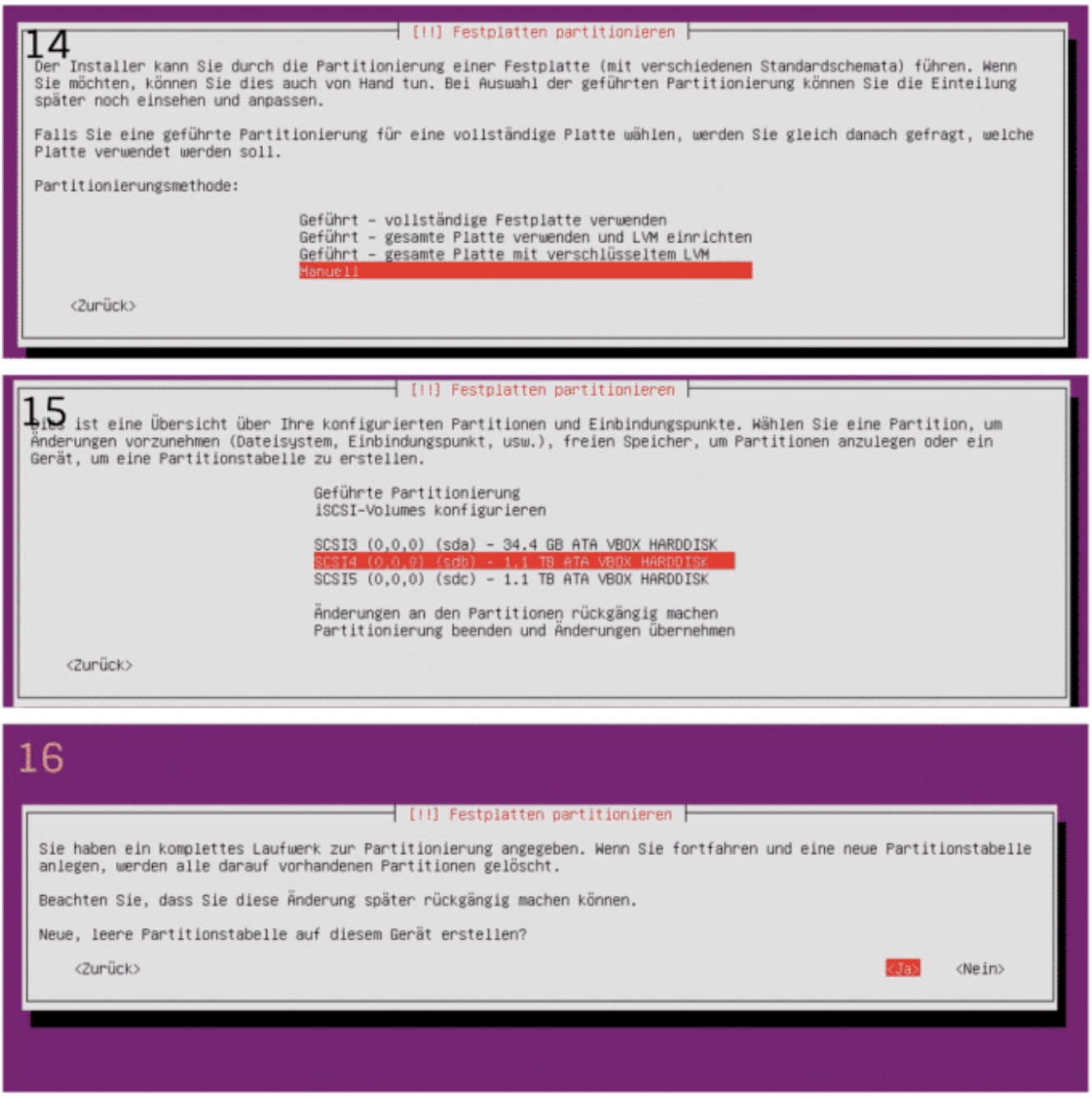
Ich habe das Setup in diesem Punkt in dieser Artikelreihe drastisch vereinfacht. So wird das System nur auf einer einzigen Platte installiert. Dadurch kann für das Betriebssystem die automatische Partitionierung des Installers genutzt werden. Dieser erkennt selbständig welche Art der Partitionierung für die Systemplatte passend ist.

Die geringere Ausfallsicherheit dürfte vertretbar sein, da auf der Systemplatte keine wichtigen Benutzerdaten gespeichert werden. Im Falle eines Ausfalls der Systemplatte muss einfach die Installation wiederholt werden. Konfigurationsdateien können aus dem Backup wiederhergestellt werden. Insbesondere für einen unerfahrenen Benutzer dürfte auch der Zeitaufwand geringer sein als das Wiederherstellen eines defekten Raid auf der Kommandozeile.

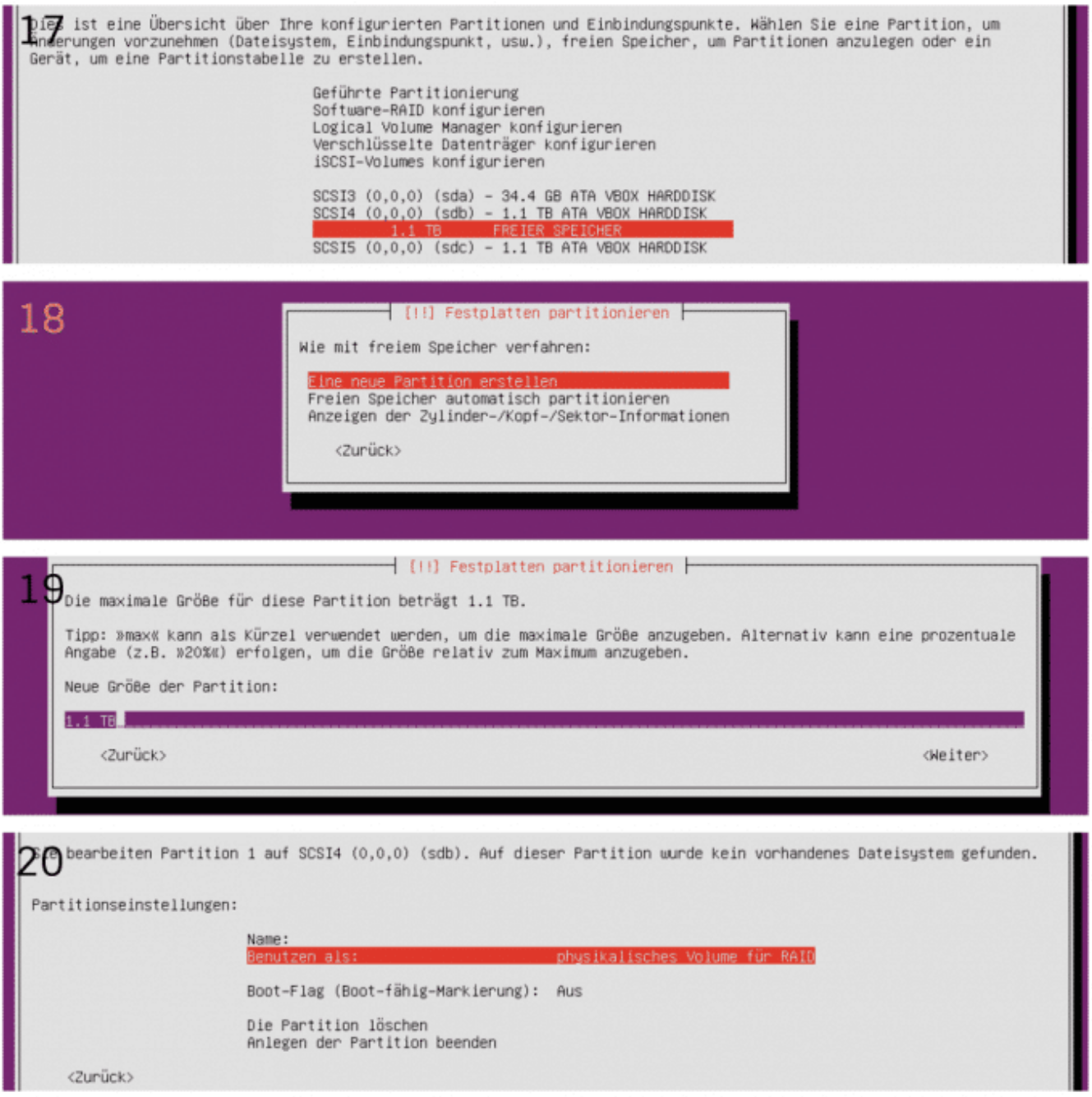
Das Raid aus den zwei großen Festplatten wird dann im nächsten Schritt manuell erstellt. Die Problematik der unterschiedlichen Partitionierung besteht hier nicht, da von diesen Datenträgern nicht gebootet werden muss. Raid-Verbund erstellen

Zuerst muss der Speicherpool aus den beiden großen Festplatten manuell erstellt werden. Wird die Systemplatte zuerst automatisch partitioniert, läuft die Installation anschließend vollständig durch und es gibt während des installationsprozesses keine Möglichkeit mehr Speicherpool zu konfigurieren.

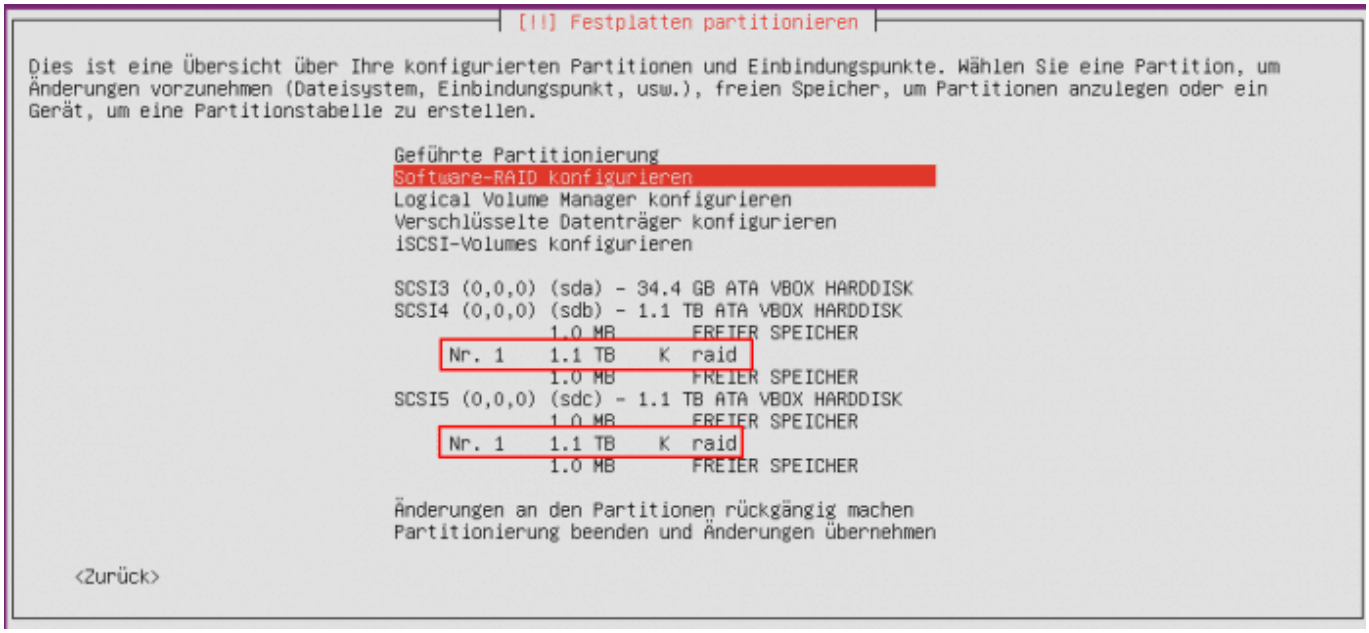
Daher wählen wir zuerst den Punkt Manuell aus. Im nächsten Bildschirm werden alle gefundenen Datenträger aufgelistet. Hier wählen wir zuerst eine der beiden großen Festplatten. Da die Festplatte neu ist muss zuerst eine leere Partitionstabelle auf die Festplatte geschrieben werden. Diese Anfrage des Systems wird daher mit Ja beantwortet.



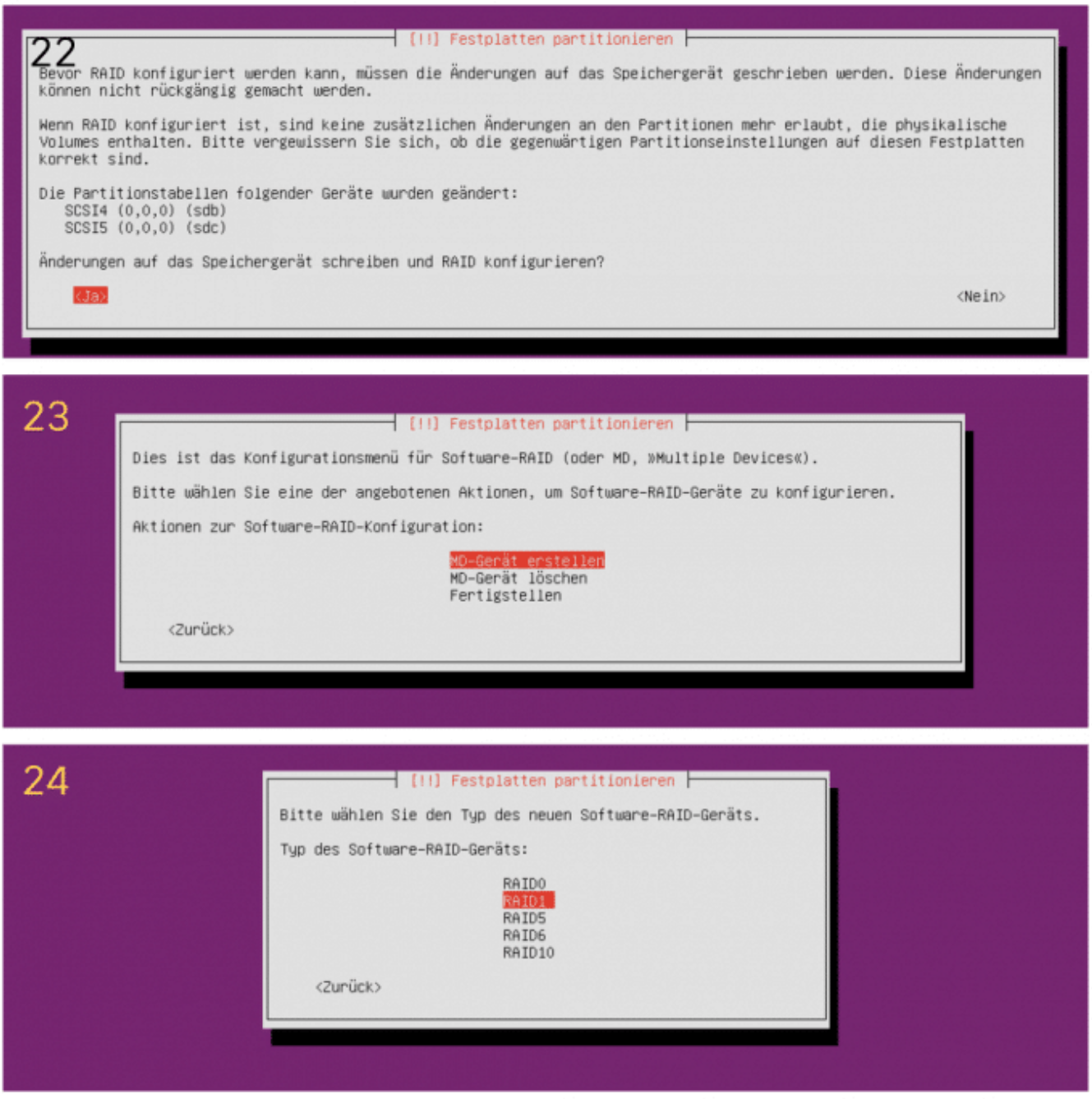
Zurück in der Übersicht sehen wir dass auf einer der Festplatten ein Eintrag “freier Speicher” hinzugekommen ist. Um diesen für die Nutzung mit einem Raid bereitzustellen muss sie entsprechend markiert werden. Hierfür wählen wir den Eintrag FREIER SPEICHER aus und im nächsten Schritt Eine neue Partition erstellen. Die Abfrage nach der größe der Partition überspringen wir einfach mit weiter, da der gesamte Speicherplatz verwendet werden soll. Im nächsten Dialogfeld muss gewählt werden wie die Partition verwendet werden soll. Entscheidend ist hier der Punkt Benutzen als. Diesen wählen wir aus und wählen die Option physikalisches Volume für RAID und gehen anschließend auf Anlegen der Partition beenden. Es folgt wieder die Übersicht mit den installierten Datenträgern. Hier ist nun ein neuer mit raid bezeichneter Eintrag entstanden.



Diese Schritte müssen nun mit der zweiten Festplatte genau gleich wiederholt werden, so dass am in der Übersicht zwei gleiche mit raid bezeichnete Einträge vorhanden sind.

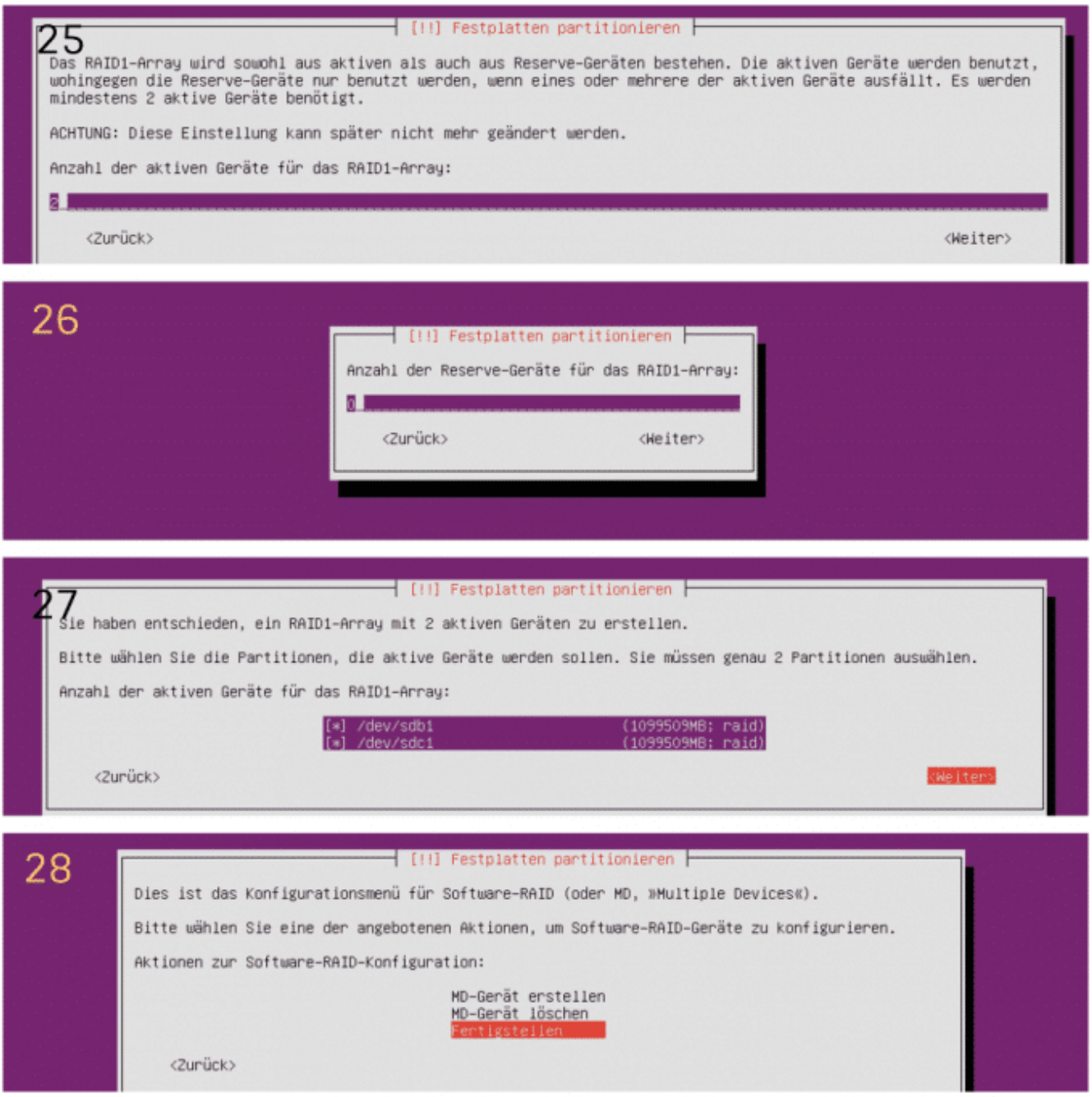


Nun müssen die beiden Datenträger noch zu einem RAID1-Speicherverbund zusammengefügt werden. Dazu gehen wir auf den Menüpunkt Software-RAID konfigurieren. Die Abfrage ob die beiden gerade erstellten Partitionstabellen auf die Festplatten geschrieben werden sollen beantworten wir mit Ja. Zum Erstellen eines Speicherpools nutzen wir die nächste Option MD-Gerät erstellen. Als nächstes kommt die Abfrage welchen Raid-Typ wir verwenden möchten. Das Spiegeln zweier Festplatten übernimmt der Typ RAID1, welchen wir hier auswählen.

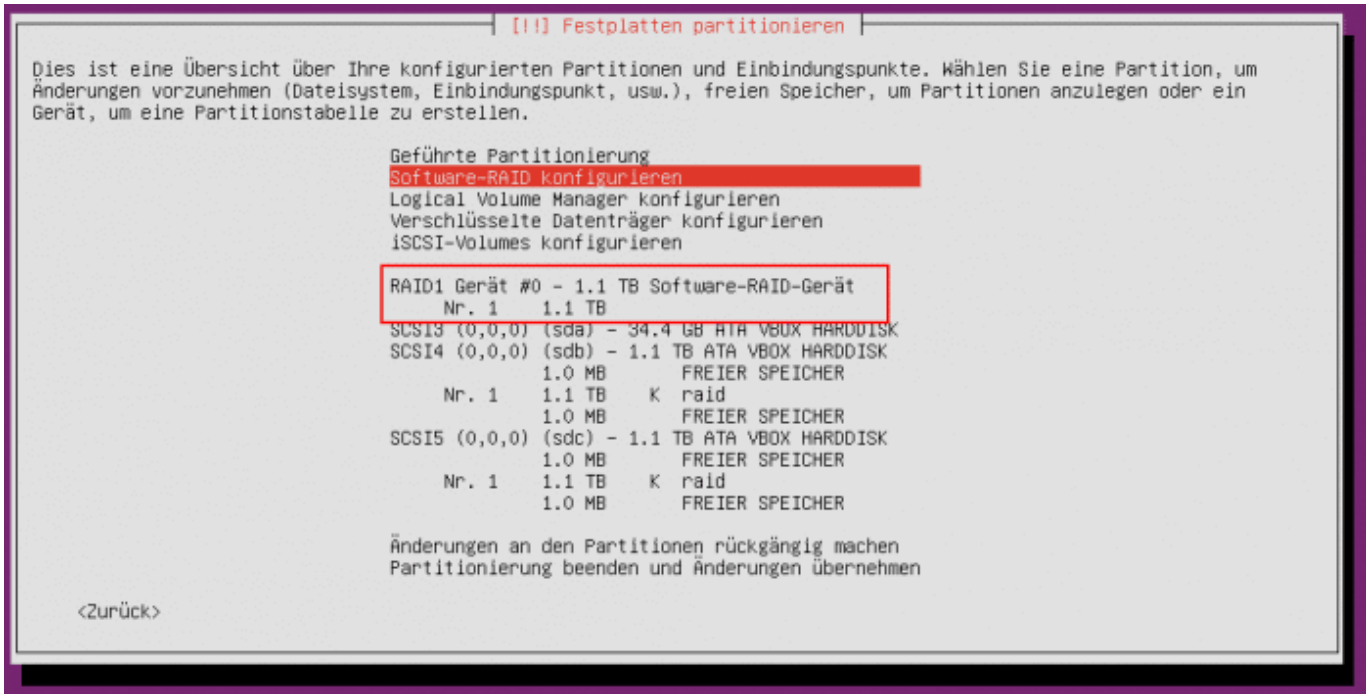


MD-Raid ermöglicht es Reserve-Festplatten vorzuhalten. Diese springen automatisch ein, wenn eine der verwendeten Festplatten ausfällt. Dies wäre für unseren selbstgebauten Homeserver/NAS etwas übertrieben. Deshalb wählen wir bei der Anzahl der aktiven Geräte zwei und bei der Anzahl der Reservegeräte null aus.

Im nächsten Schritt wird abgefragt welche Geräte wir zu unserem Speicherverbund zusammenfassen wollen. Hier sollten die beiden vorhin konfigurierten Laufwerke zu sehen sein. Es werden beide Laufwerke markiert und mit weiter bestätigt. Der Verbund ist nun erstellt. Im nächsten Schritt können wir daher den Eintrag Fertigstellen auswählen.

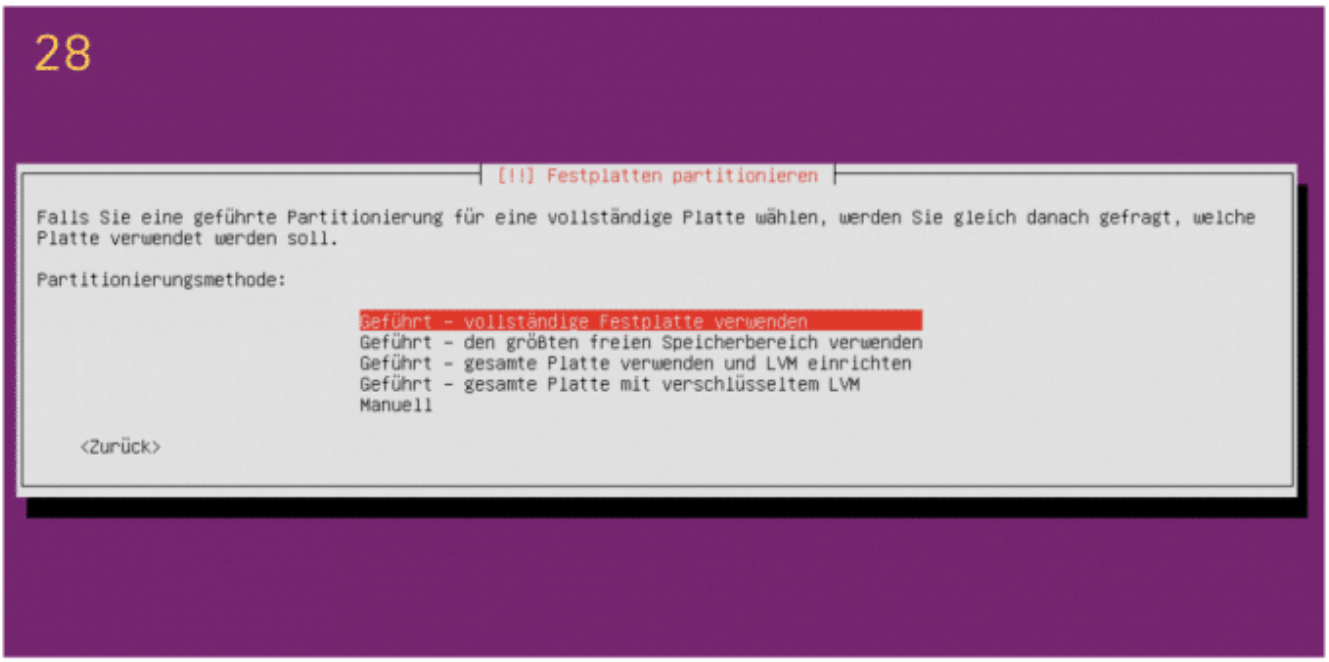


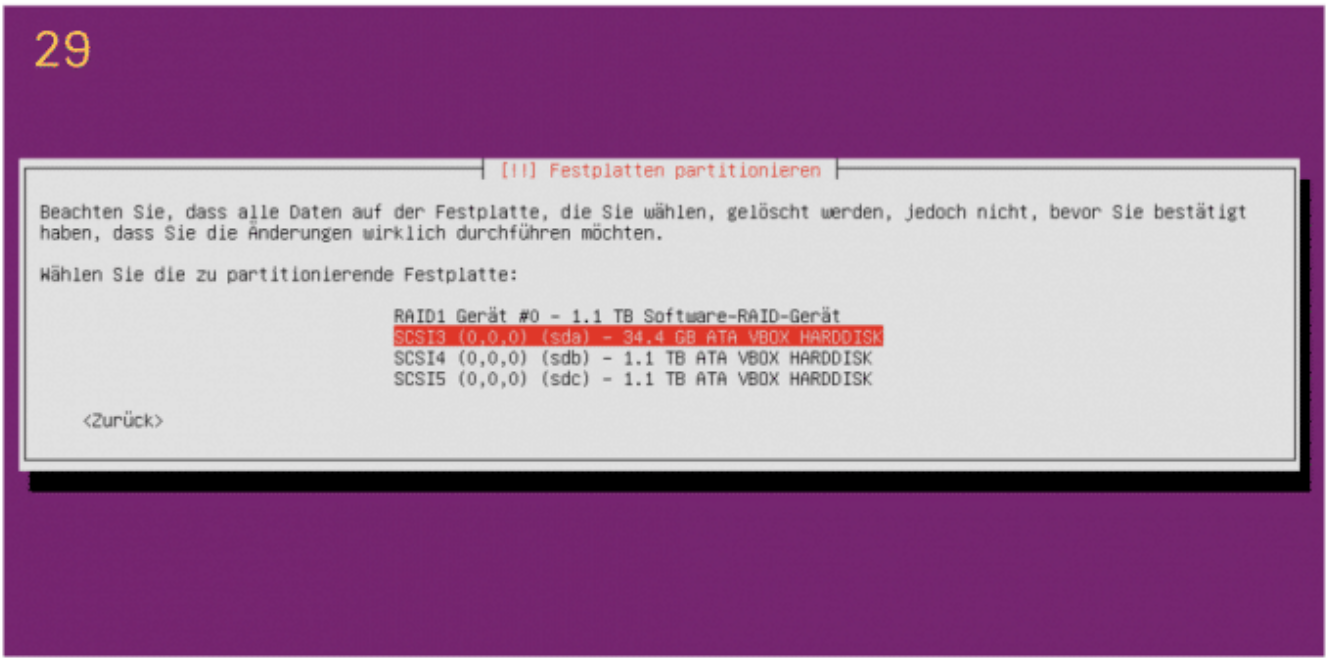
Wir kommen nun wieder zurück zur bereits bekannten Übersicht. Hier ist wieder ein neuer Eintrag entstanden, unser Raid1 Speicherpool.



Automatische Partitionierung der Systemplatte

Als nächstes kann die automatische Partitionierung für die Systemfestplatte verwendet werden. Dafür wählen wir in der Übersicht den Eintrag geführte Partitionierung und wählen in der folgenden Abfrage Geführt – vollständige Festplatte verwenden. Es folgt wieder eine Übersicht mit allen gefundenen Geräten. Hier wählen wir nun die kleine SSD, zu erkennen an der Speicherplatzangabe. Nun wird automatisch die Partitionierung vorgenommen.



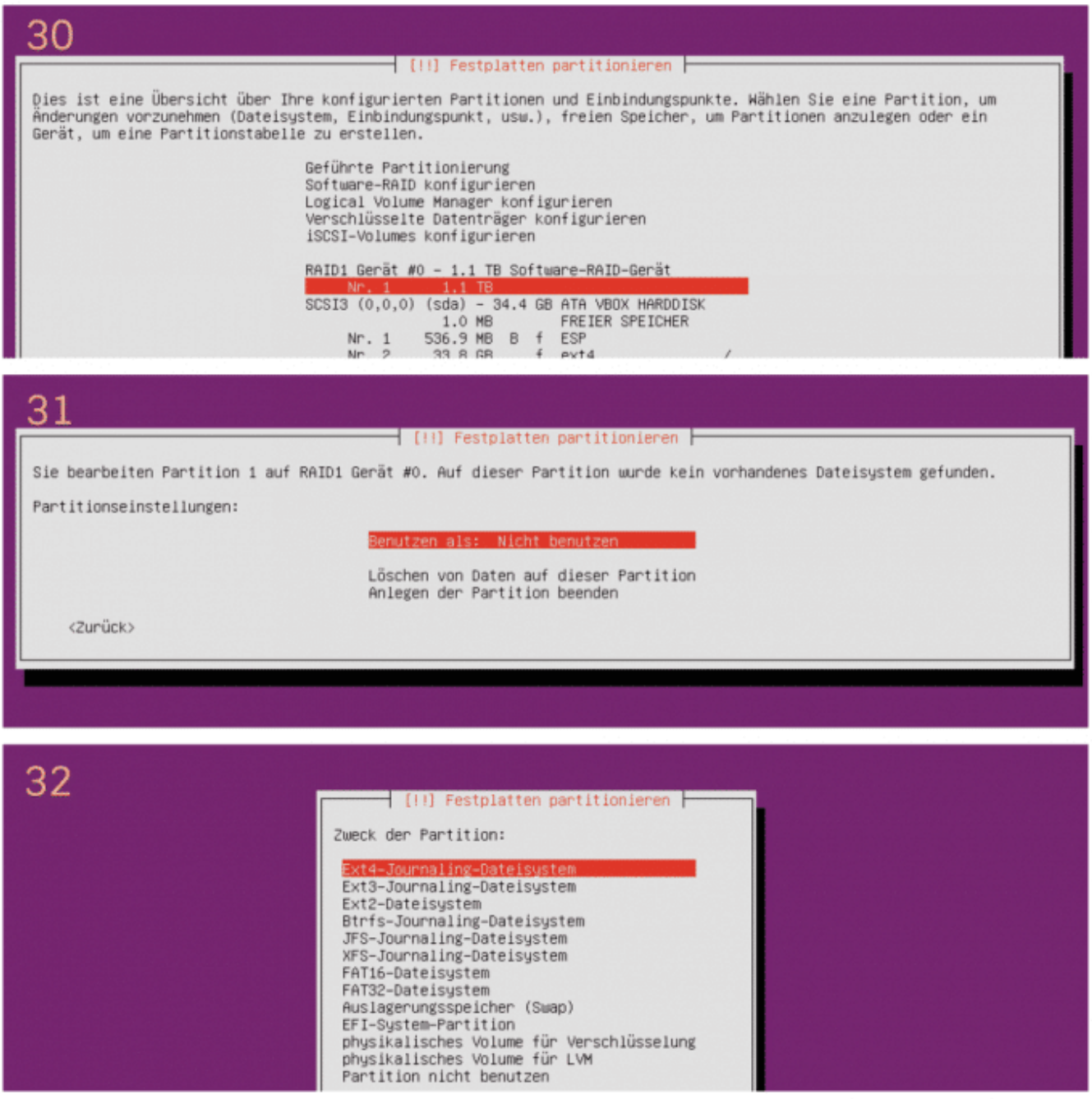


Nach einem kurzen Moment finden wir uns direkt in der bekannten Übersicht wieder. In diesem Fall wurde eine ESP-Bootpartition erstellt und eine weitere Partition mit dem restlichen Speicherplatz der SSD mit ext4-Dateisystem für die Installation des Systems.

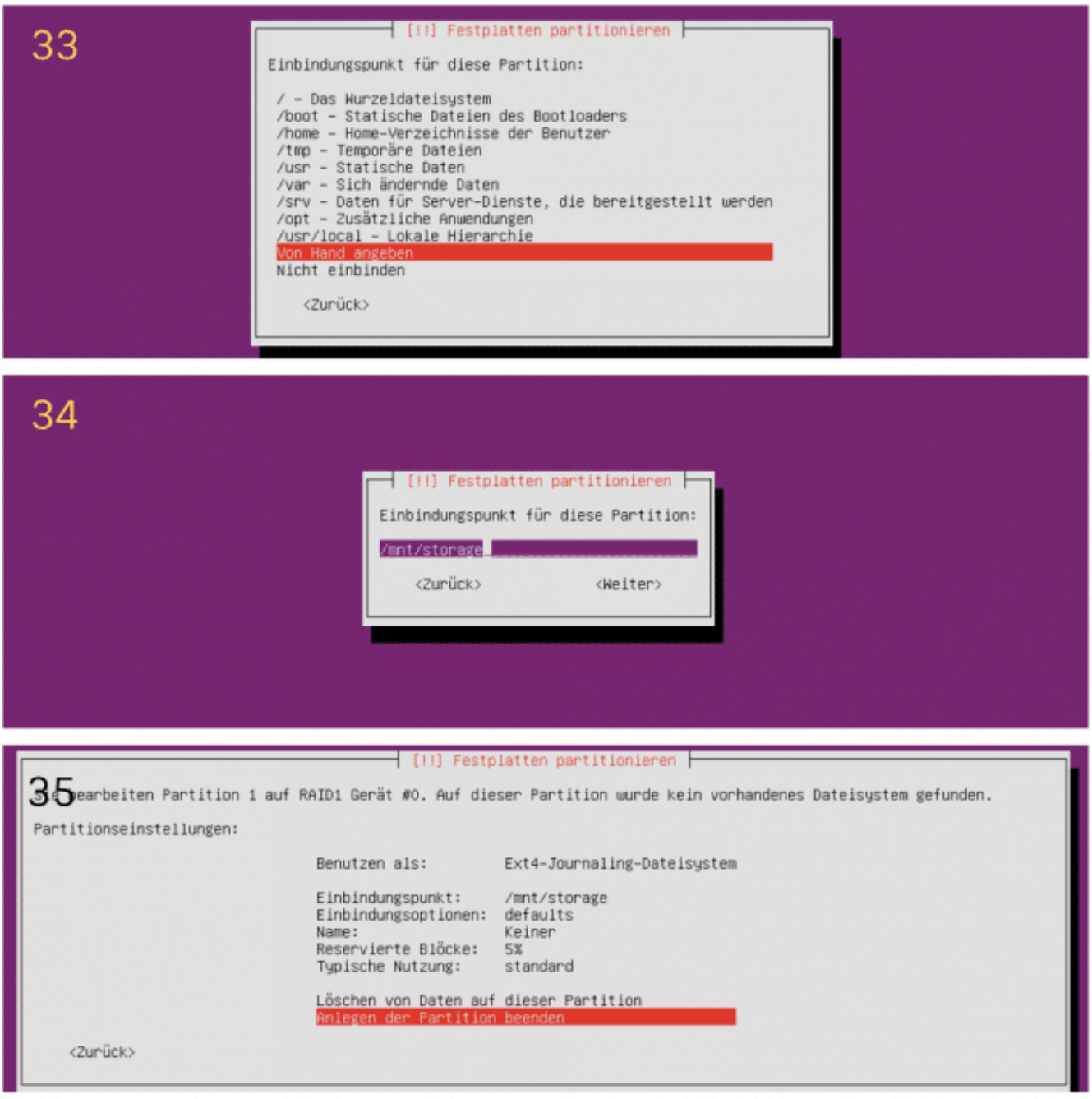
Als letzten Punkt müssen wir noch den Ort angeben, an dem unser Speicherpool eingebunden wird. Linux verwendet im Gegensatz zu Windows keine Buchstaben um Laufwerke zu benennen. Stattdessen werden diese wie ein Ordner an einem bestimmten Punkt im Dateisystem eingebunden, der sogenannte mountpoint.

Hätten wir diesen Schritt bereits vorhin vorgenommen, wäre die Einstellung mit der automatischen Partitionierung der Systemfestplatte wieder entfernt worden.

Um den Mountpoint festzulegen wählen wir den leeren Eintrag unter dem sogenannten Software-RAID-Gerät aus und gehen in der nächsten Abfrage auf Benutzen als. Hier wählen wir jetzt ext4-Journaling-Dateisystem, das Standarddateisystem von Ubuntu.

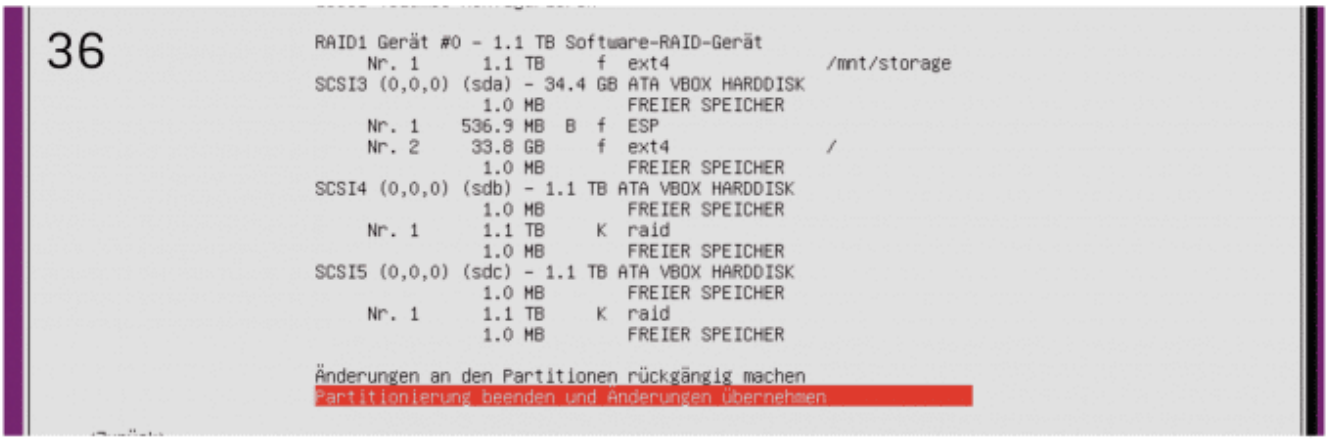


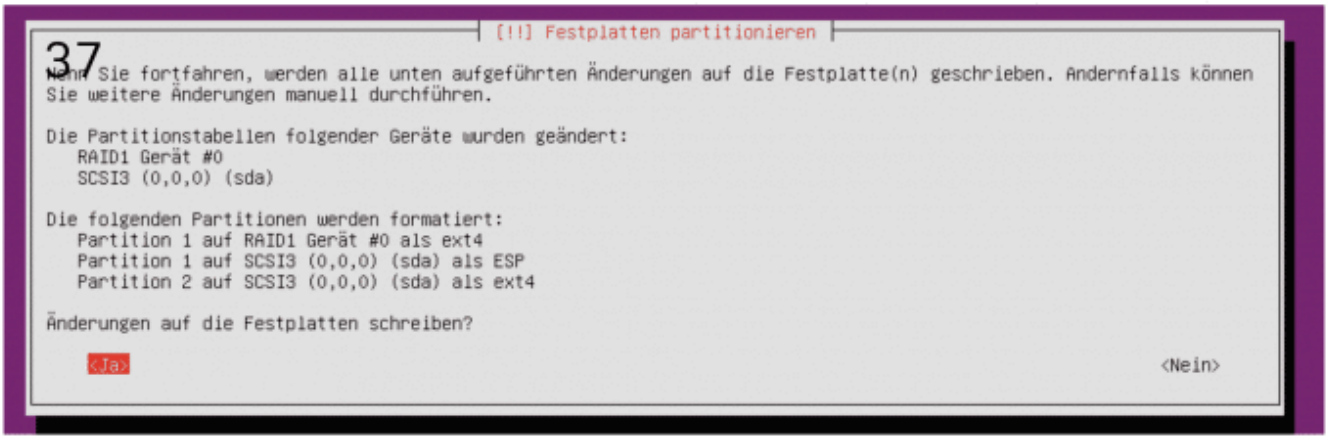
Den Mountpoint legen wir unter dem Eintrag Einbindungspunkt fest. Diesen wählen wir aus und wählen anschließend die Option von Hand eingeben und legen als Einbindungspunkt /mnt/storage fest.



Die restlichen Optionen können unverändert bleiben. Die Konfiguration wird mit Anlegen der Partition abgeschlossen.

Damit ist die Partitionierung abgeschlossen. Die Installation wird fortgesetzt mit der Auswahl von Partitionierung beenden und Änderungen Übernehmen. Es erfolgt nochmals eine Abfrage ob die erstellten Partitionen auf die Platten geschrieben werden sollen. Diese wird mit Ja beantwortet. Anschließend werden die Partitionen erstellt, was je nach System etwas dauern kann. Danach wird die Installation des Betriebssystems automatisch fortgesetzt.







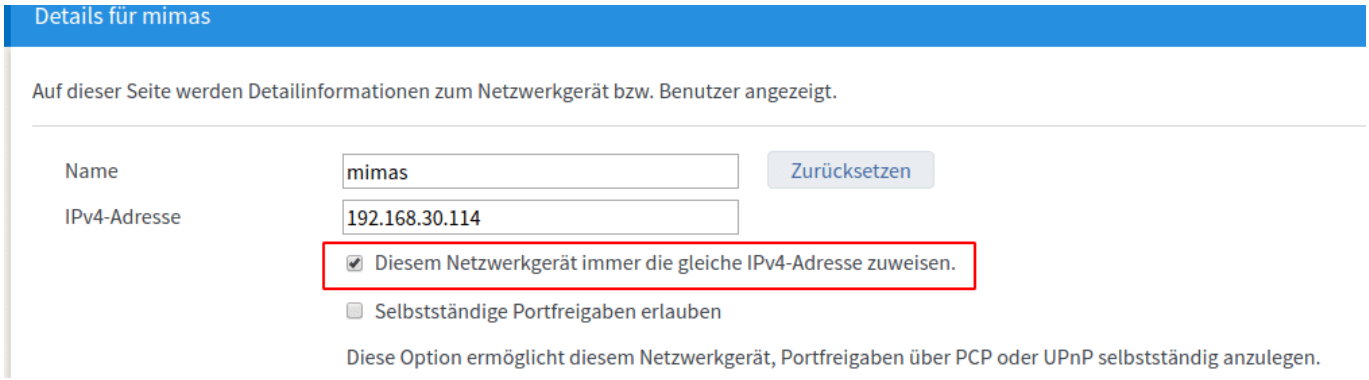
Abschluss der Installation von Ubuntu Server 18.04

Die restliche Installation geschieht weitestgehend automatisch. Bevor das Installationsprogramm benötigte Dateien aus dem Internet herunterlädt, fragt es ob ein Proxyserver verwendet wird. Die Frage kann einfach mit weiter übersprungen werden.

Wichtiger ist die Frage nach der automatischen Installation von Sicherheitsupdates. Um ein wartungsarmes und trotzdem sicheres System zu bekommen empfiehlt es sich hier die Auswahl Sicherheitsaktualisierungen automatisch installieren zu wählen.

Als letzte Abfrage kommt noch eine Auswahl mit Software, welche automatisch vorinstalliert werden soll. Hier wählen wir nur den OpenSSH-Server aus. Damit ist es möglich sich via Kommandozeile von einem anderen Rechner auf dem Homeserver einzuloggen. D.h. wir können den Homeserver nachdem

die Installation abgeschlossen ist, ohne angeschlossenen Monitor oder Tastatur in die Ecke stellen. Die weitere Einrichtung erfolgt dann via SSH vom Laptop oder Desktop aus.



Exkurs: Remote Login via SSH

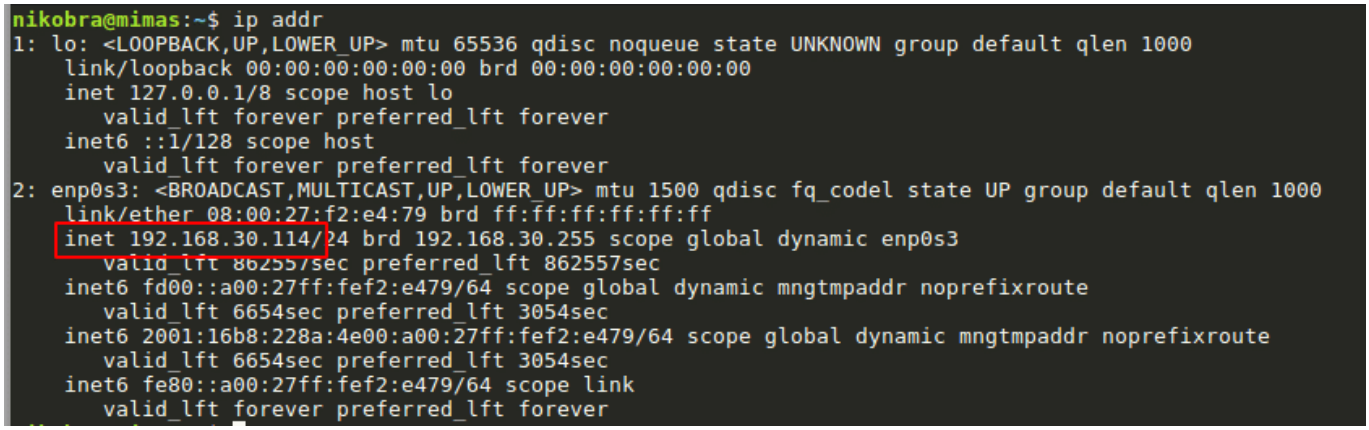
Das NAS bzw unser selbstgebauter Homeserver soll später ohne Bildschirm oder Tastatur im Keller oder Flur stehen. Die Administration erfolgt von einem anderen Computer aus, mit welchem man sich auf dem Homeserver einloggt. Da der Server nicht über eine grafische Benutzeroberfläche verfügt, sondern komplett über die Kommandozeile administriert wird, ist OpenSSH hier das Mittel der Wahl. OpenSSH besteht aus einer Serverkomponente, die wir bereits bei der Installation auf dem Server installiert haben.

Außerdem wird auf dem Client, also dem Computer mit welchem man sich auf dem Server einloggt, ein SSH-Client benötigt. Zum Login auf dem Server benötigt man dessen IP-Adresse, sowie Benutzername und Passwort.

Eventuell ist die IP-Adresse des Homeservers bereits bekannt, weil man diese wie oben beschrieben aus der Konfigurationsoberfläche des Routers kennt. Falls nicht, können wir uns diese auf dem Homeserver selbst ausgeben lassen. Dazu loggen wir uns zuerst über die noch angeschlossenen Bildschirm und Tastatur am Server ein und führen dort den Befehl

```
1. id addr
```

aus. Damit bekommen wir alle verfügbaren IPv4, IPv6 und MAC-Adressen des Systems angezeigt. Uns interessiert die IPv4 Adresse die unter dem Punkt inet beim Interface enp0s3 oder eth0 o.ä. angegeben ist, in diesem Fall die Adresse 192.168.30.114.



Login auf dem Homeserver mit Linux oder MacOS

Bei allen mir bekannten Linux Distributionen sowie bei MacOS ist ein SSH-Client standardmäßig installiert. Man kann sich somit direkt auf dem Homeserver einloggen, indem man das Terminal öffnet und den Befehl

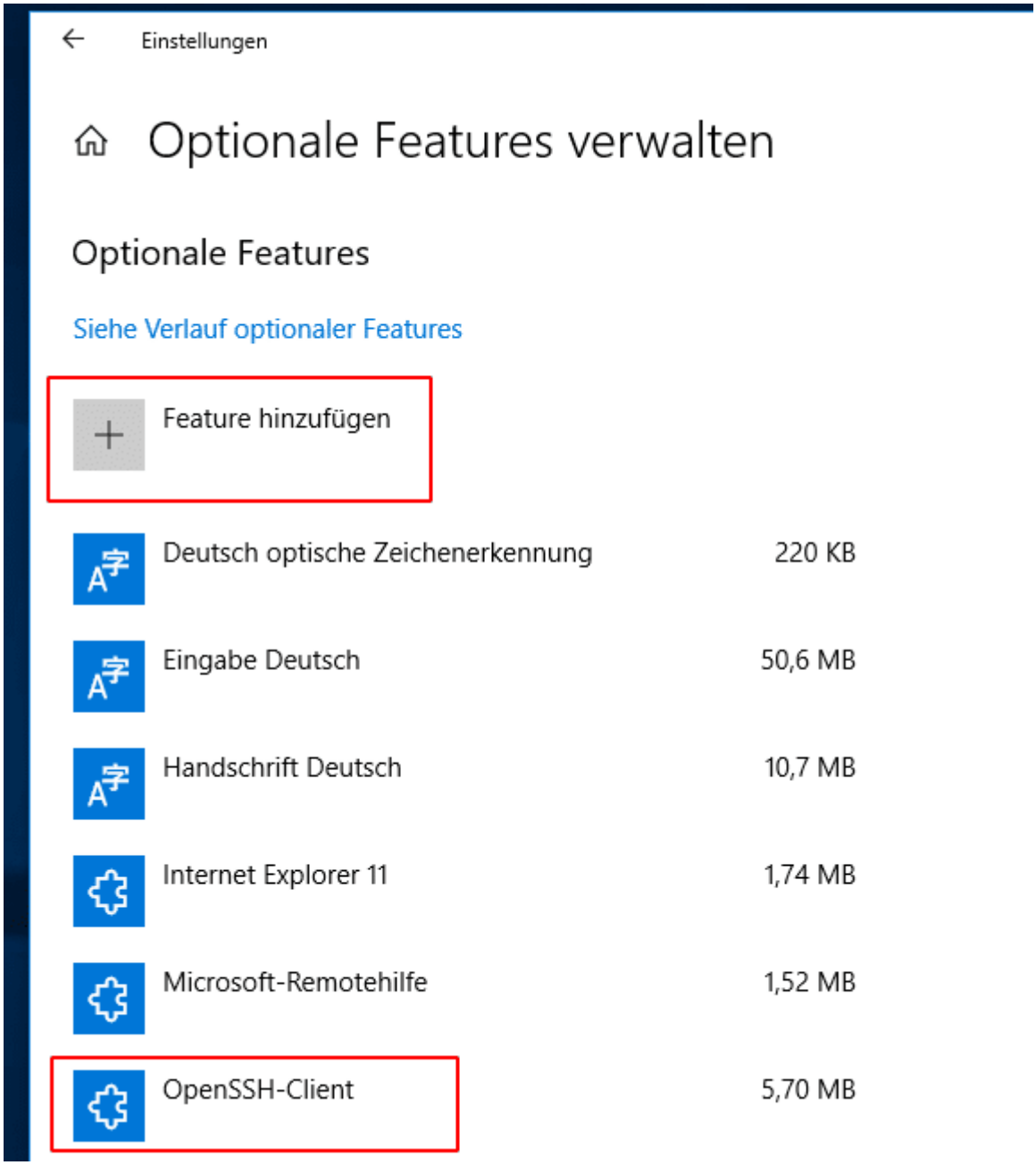
```
1. ssh USERNAME@192.168.30.114
```

ausführt.

Login auf dem Homeserver via SSH mit Windows 10

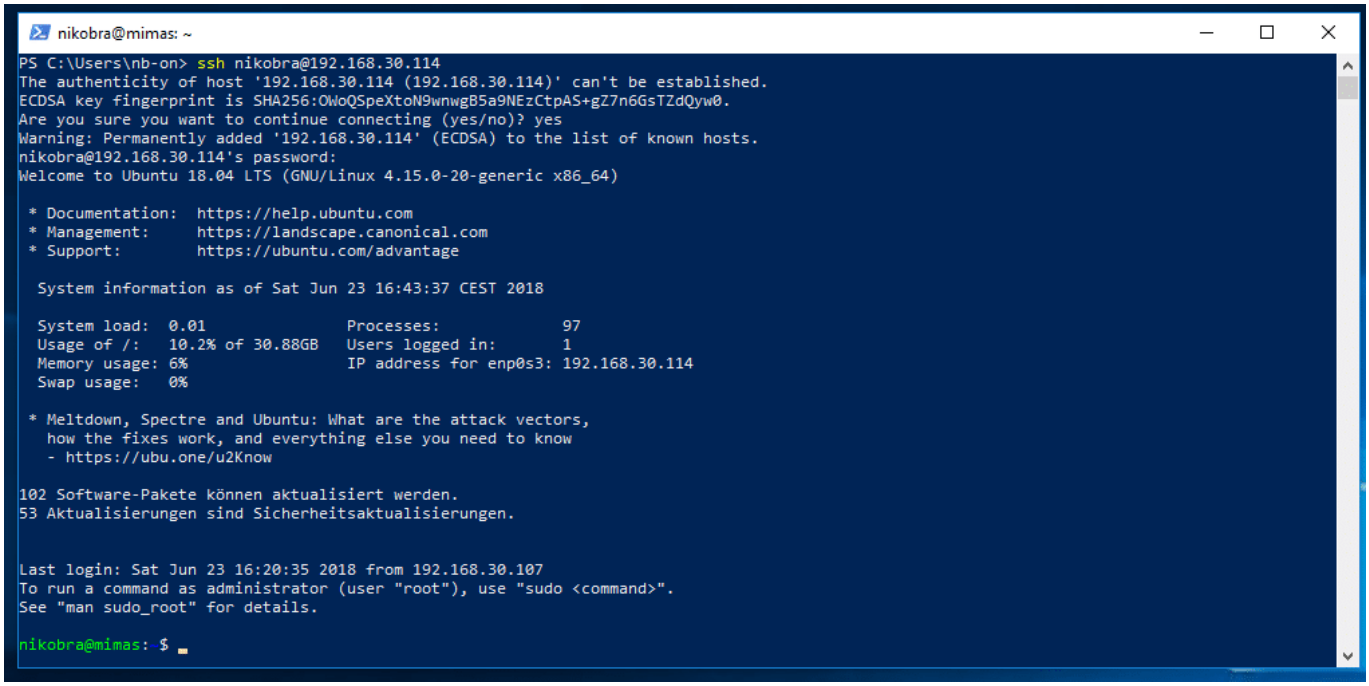
Bis einschließlich Windows 8 wollte Microsoft nichts mit SSH zu tun haben. Wie so vieles hat sich das mit Windows 10 geändert. Mittlerweile bringt auch Windows 10 einen eigenen SSH-Client mit, der allerdings noch als Beta gekennzeichnet ist. Trotzdem funktioniert er und ist durch die volle Integration in das System deutlich angenehmer zu nutzen als ein Programm eines Drittherstellers. Wenn man den integrierten OpenSSH-Client von Windows verwendet, kann man sich mit der Eingabeaufforderung (cmd.exe) oder über die Powershell via SSH auf dem Homeserver einloggen.

Mittlerweile scheint Microsoft den SSH-Client unter Windows 10 standardmäßig zu aktivieren. Ob dies der Fall ist, sieht man in den Optionalen Features von Windows 10. Diese findet man wenn man im Startmenü nach “optionale features” sucht. Wenn hier der OpenSSH-Client aufgeführt ist, ist er bereits installiert. Ansonsten kann dieser über die Auswahl “feature hinzufügen” installiert werden.



Nun kann man sich auf dem Homeserver einloggen indem man die PowerShell aufruft und ebenfalls folgenden Befehl ausführt:

```
1. ssh USERNAME@192.168.30.114
```

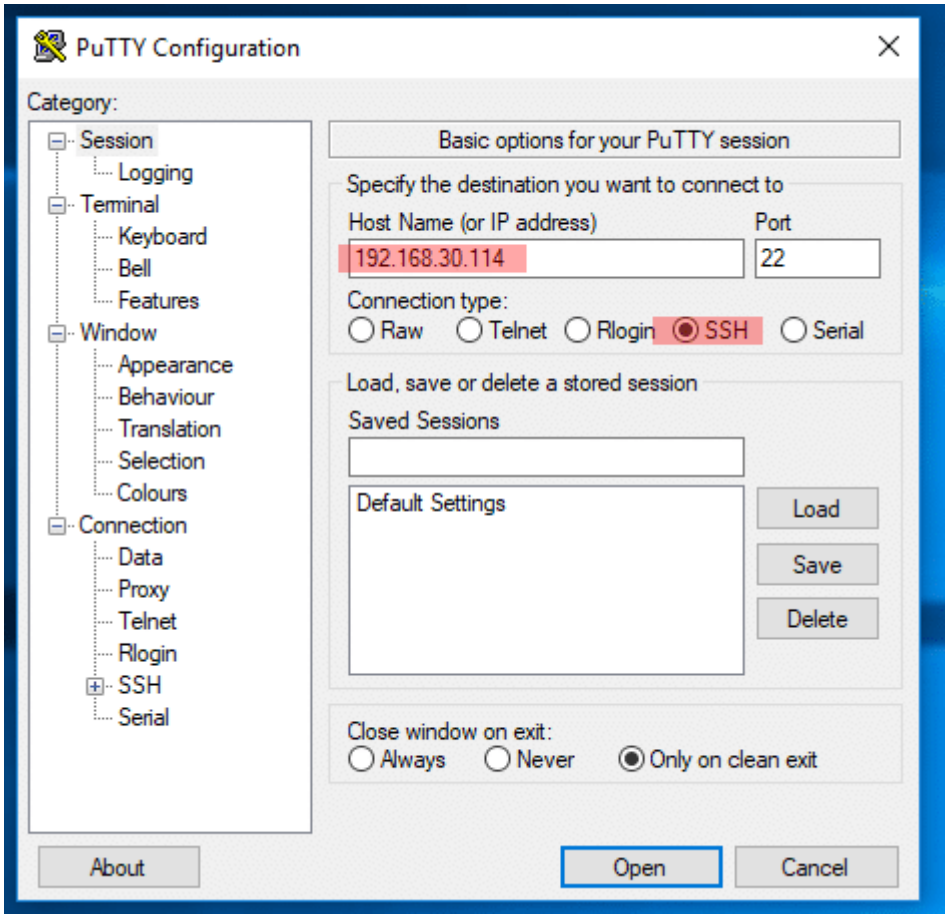


Nun ist man mit dem Homeserver verbunden und kann alle weiteren administrativen Aufgaben von seinem Windowscomputer aus vornehmen.

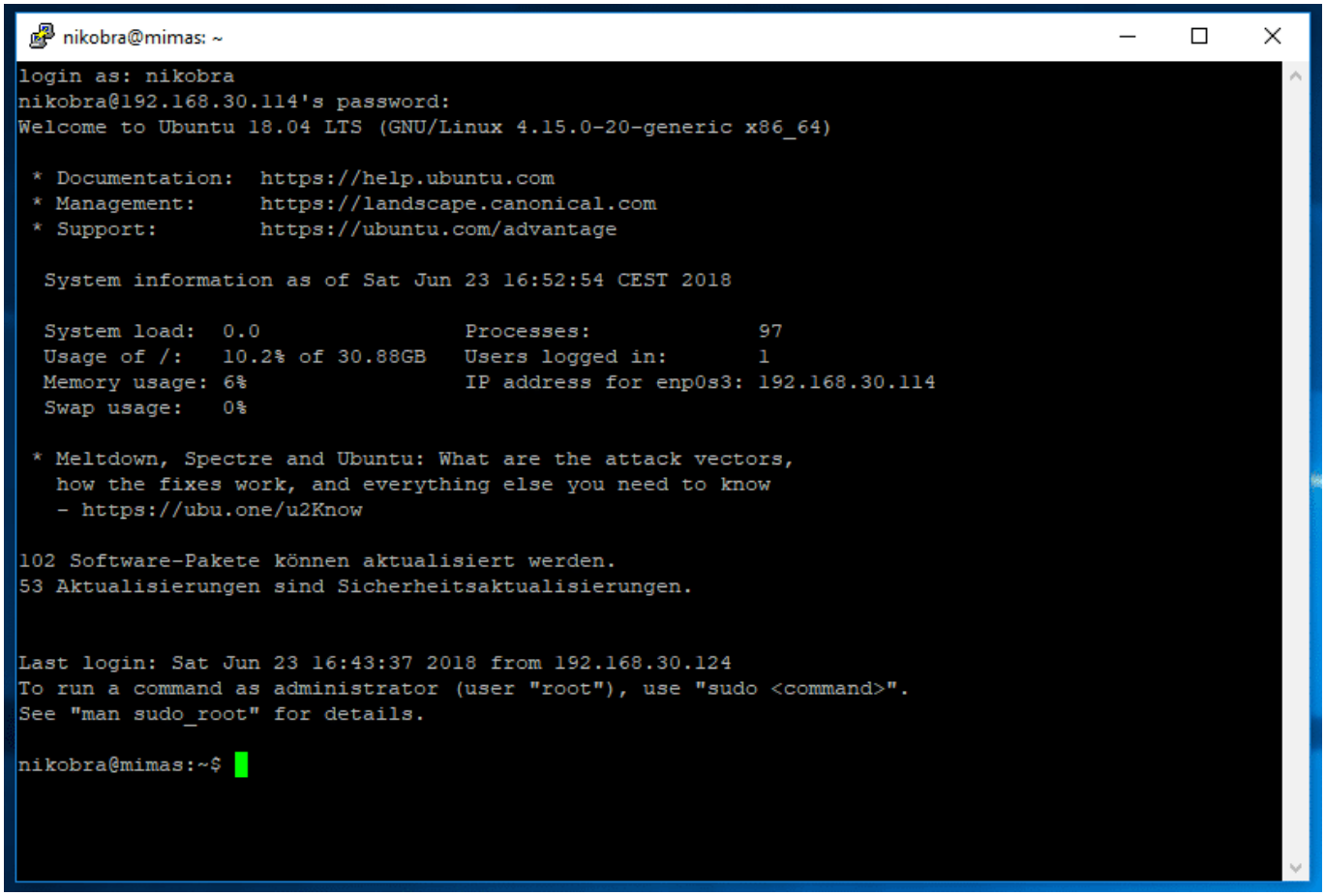
Login auf dem Homeserver via SSH mit Windows 7

Unter Windows 7 hat Microsoft SSH noch ignoriert, da es für die Administration von Windows Servern nicht benötigt wird. Wenn man noch Windows 7 einsetzt muss man auf eine Lösung von einem Drittanbieter setzen. Am gebräuchlichsten ist hier wahrscheinlich die Software Putty. Putty kann kostenlos auf der Homepage des Projekts unter putty.org heruntergeladen werden.

Im Konfigurationsfenster gibt man die IP-Adresse des Homeservers ein und wählt den Connection type “SSH”.



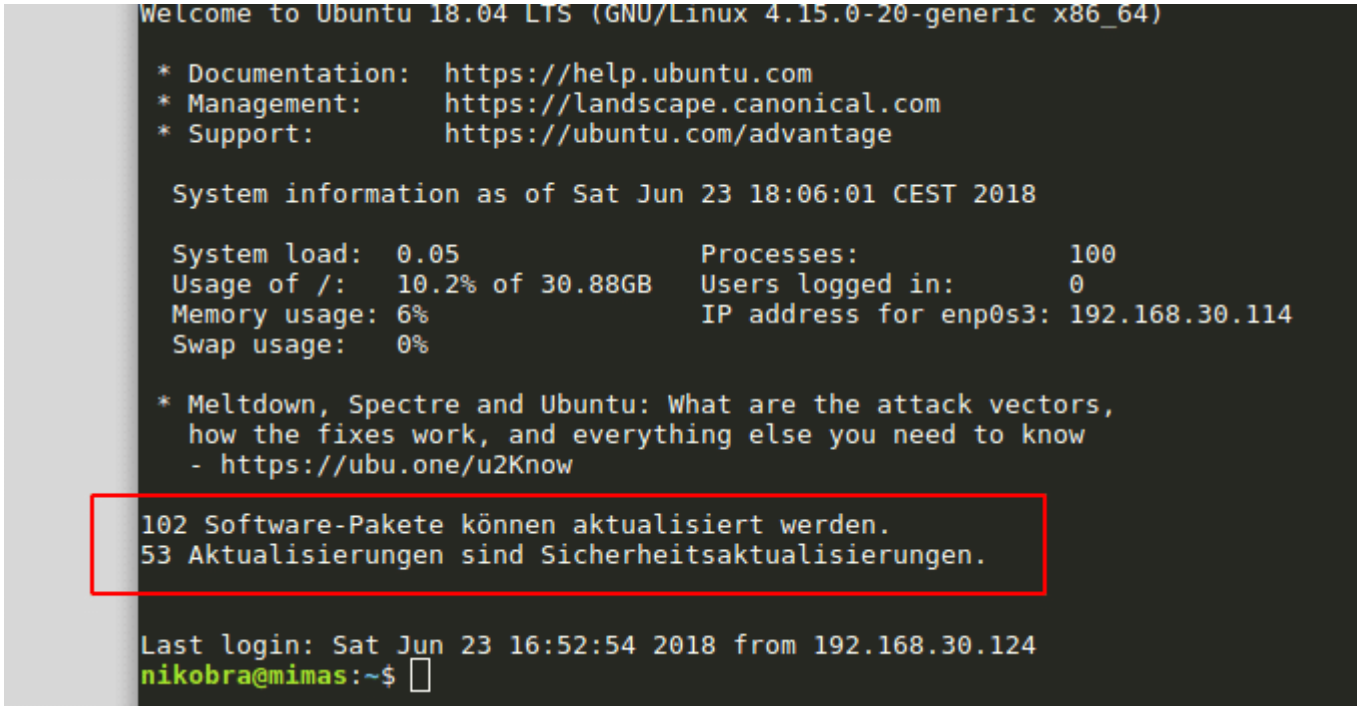
Mit einem Klick auf Open öffnet sich ein neues Fenster und die Verbindung wird aufgebaut. Damit lässt sich auch unter Windows 7 eine Verbindung mit dem selbstgebauten Homeserver/NAS herstellen.



Das System aktuell halten

Nur ein aktuelles System ist auch ein sicheres System. Das System wurde im letzten Teil zwar so konfiguriert dass Sicherheitsupdates in regelmäßigen Abständen automatisch eingespielt werden, alle anderen Updates müssen aber manuell vorgenommen werden. Außerdem ist es von Zeit zu Zeit nötig das System neu zu starten, damit alle Updates angewendet werden.

Ob Updates zur Verfügung stehen, bzw. ob ein Neustart nötig ist erfährt man wenn man sich wie im letzten Teil beschrieben via SSH auf dem Server einloggt. In diesem Fall stehen nach der Installation Updates für über 100 Pakete bereit.



Mit folgendem Befehl kann man das System updaten.

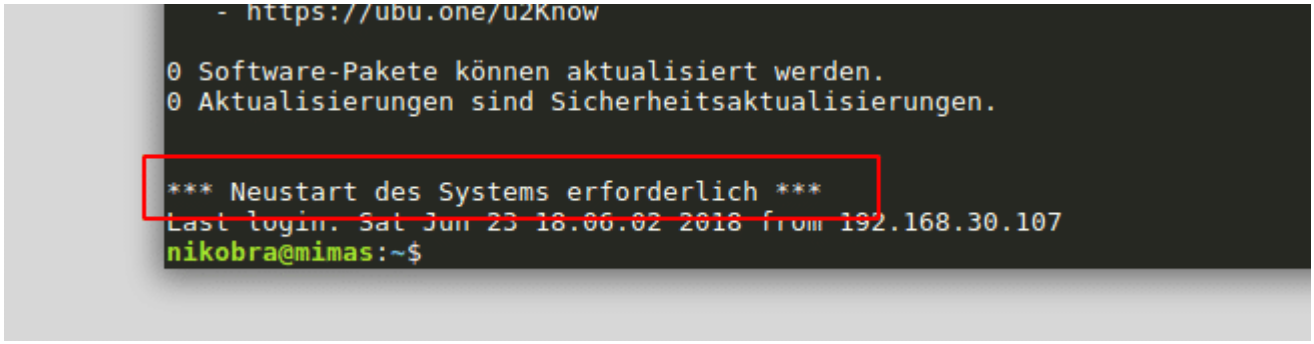
```
1. sudo apt update && sudo apt upgrade
```

Das vorangestellte sudo führt dazu dass der Befehl mit erhöhten Rechten (Rootrechte) ausgeführt wird, da ansonsten keine Systemdateien verändert werden dürfen. Apt ist das Programm welches für die Paketverwaltung und damit für die Aktualisierung zuständig ist. Der erste Teil des Befehls (vor dem &&) sorgt dafür dass die Paketquellen auf Aktualisierungen überprüft werden. Der zweite Teil spielt diese ein.

Nach dem Einspielen einer Vielzahl von Updates ist es wahrscheinlich dass das System einen Neustart benötigt. Auch hierüber wird man nach dem Login informiert. Also melden wir uns zuerst mit dem Befehl

```
1. exit
```

vom System ab, um uns anschließend direkt neu zu verbinden. Tatsächlich weist uns das System darauf hin dass ein Neustart nötig ist.



Einen Neustart führen wir mit dem Befehl

```
1. sudo reboot
```

aus. Wenn das System nur heruntergefahren werden soll, ohne neu zu starten wird folgender Befehl verwendet:

```
1. sudo shutdown -h now
```

Homeserver mit DynDNS aus dem Internet erreichbar machen

Damit wir auch unterwegs über Nextcloud Zugriff auf unsere Daten bekommen, bzw unsere Kontakte und Termine mit Nextcloud synchronisieren können oder Medien via Plex streamen, müssen wir das System über das Internet erreichbar machen. Das Problem sind hier einerseits die wechselnden IP-Adressen am DSL-Anschluss, andererseits will sich auch niemand eine IP-Adresse merken.

Damit der hier beschrieben Zugriff aus dem Internet funktioniert, muss der Internetanschluss – wie in der Einleitung beschrieben – über eine öffentliche IP-Adresse verfügen. Bei einem DSL-Anschluss sollte das normalerweise der Fall sein. Bei einem Kabelanschluss mit DS-Lite funktioniert der Zugang über Internet auf das Heimnetz leider nicht.

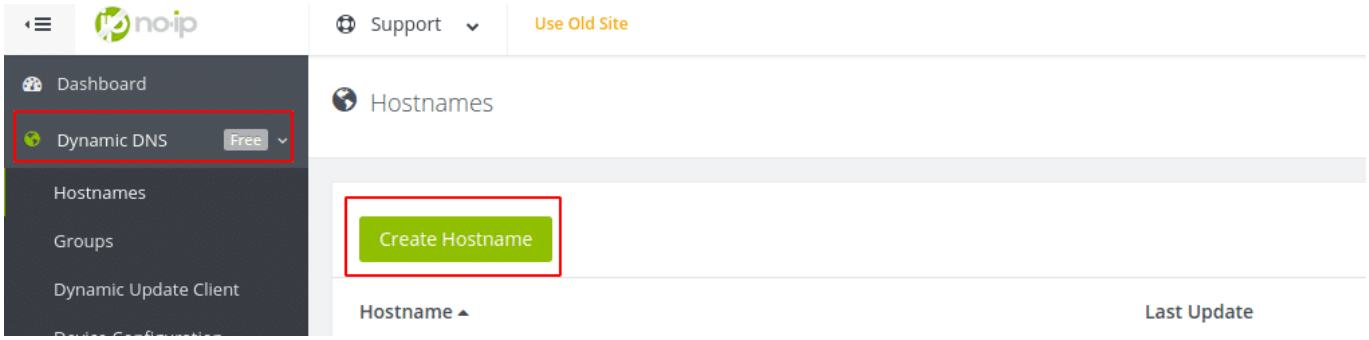
Die wohl bekannte Lösung um den DSL-Anschluss aus dem Internet erreichbar zu machen ist DynDNS. Ein System dass die hinter einem Domainnamen stehende IP-Adresse bei jeder Änderung aktualisiert. Somit ist der Homeserver von unterwegs immer über eine leicht zu merkende Internetadresse erreichbar.

Es gibt eine Vielzahl von DynDNS Anbietern, die z.B. auch direkt von der Fritzbox unterstützt werden, so dass nur Benutzername, Passwort und Domainname angegeben werden muss. Ich bin schon seit längerem zufriedener Nutzer des Dienstes Feste-IP.net, welchen ich aus diesem Grunde gerne hier erwähnen möchte. Allerdings ist der Dienst kostenpflichtig (was ich zwar als Vorteil sehe [der Kunde bin ich, nicht ein Werbepartner], allerdings die Einstiegshürde für die Umsetzung dieser Anleitung hochsetzten würde.). Auch der Webhoster Strato unterstützt DynDNS. Wer dort Kunde ist, kann mit einer “richtigen” Internetadresse auf seinen Homeserver zugreifen. Also z.B. meinname.de anstatt meinname.ddns.net

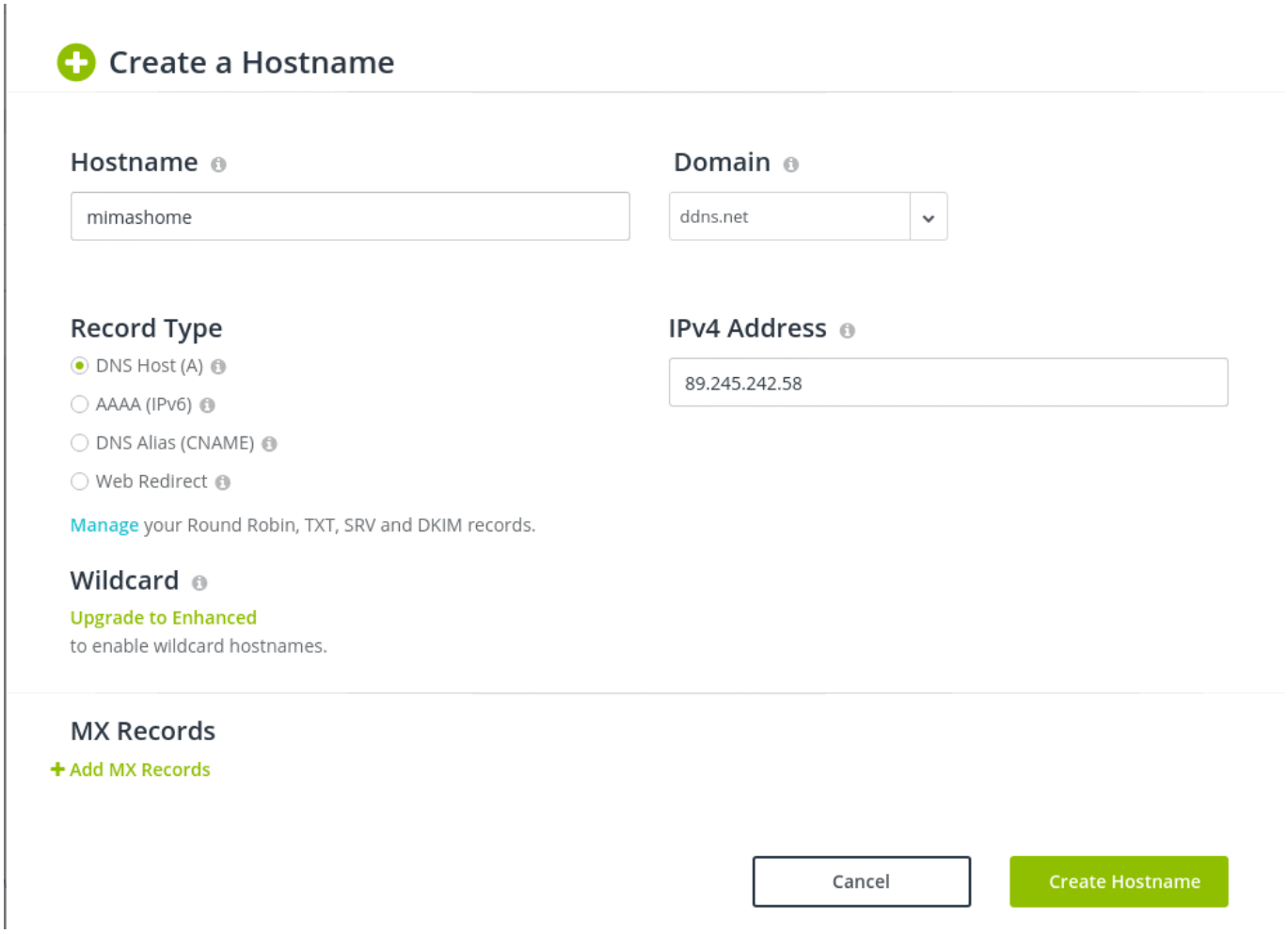
Um die Einstiegshürde niedrig zu halten zeige ich die Einrichtung in diesem Artikel anhand des kostenlosen Anbieters No-IP.com, welcher außerdem direkt von der Fritzbox unterstützt wird.

DynDNS mit No-IP.com

Zuerst muss man sich einen Account bei No-IP.com anlegen. Nach dem Login kann der Domainname angelegt werden. Hierzu geht man links im Menü auf Dynamic DNS und anschließend auf Create Hostname.



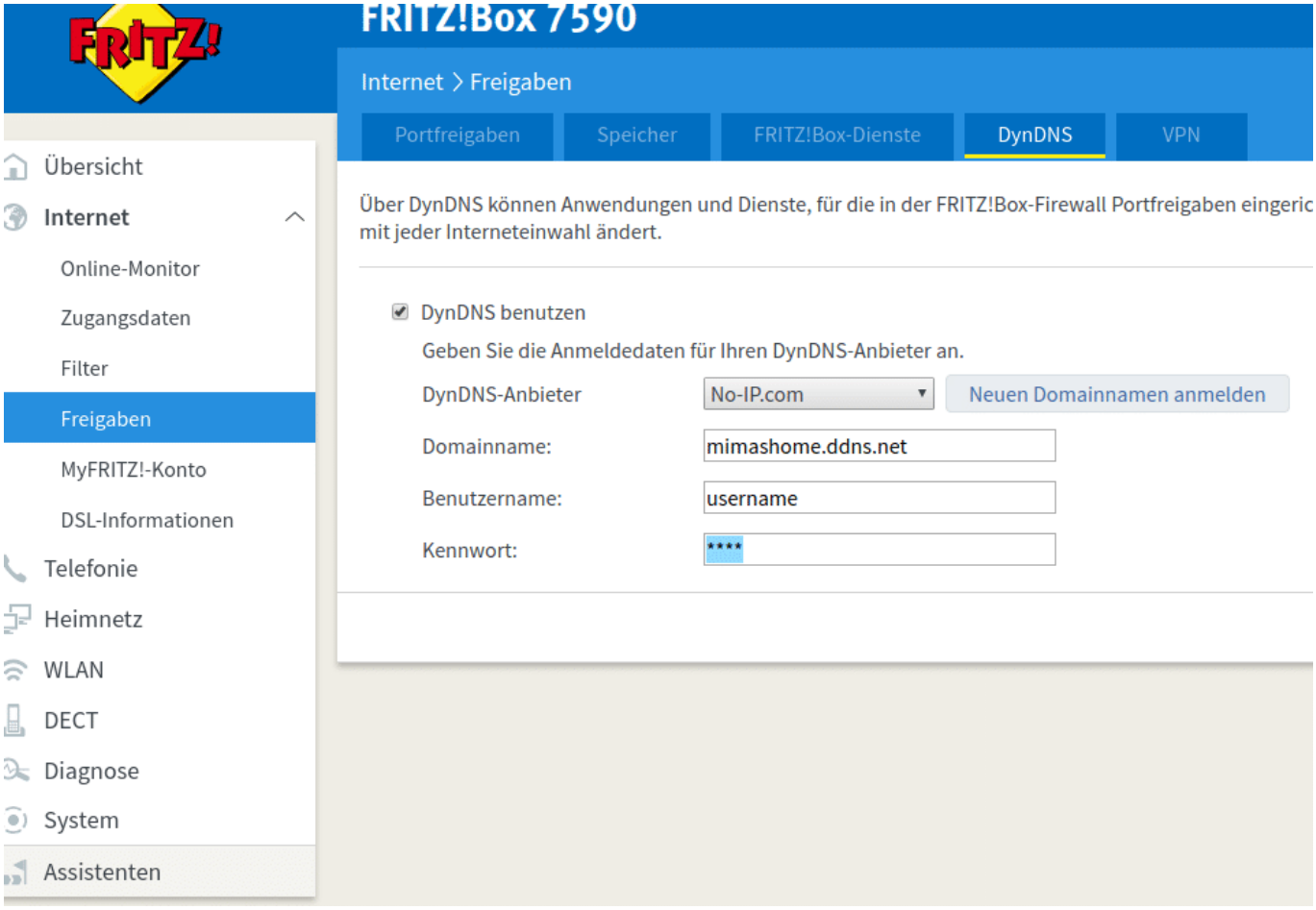
Dann kann man sich einen Hostnamen und eine Domain wählen. Daraus setzt sich dann die Adresse zusammen unter der der Homeserver später erreichbar sein wird. im Beispiel ist das https://mimashome.ddns.net. Alle anderen Felder können ignoriert werden bzw. müssen nicht verändert werden. Angelegt wird die Adresse mit einem Klick auf “Create Host”



DynDNS Dienst in der FritzBox aktivieren

Da No-IP in der Fritzbox standardmäßig bereits vorkonfiguriert ist, ist die Konfiguration schnell erledigt. Die Einstellungen hierzu findet man unter Internet → Freigaben → DynDNS

Hier gibt man seine gerade gewählte Internetadresse sowie seinen Benutzernamen und sein Passwort von No-IP an.



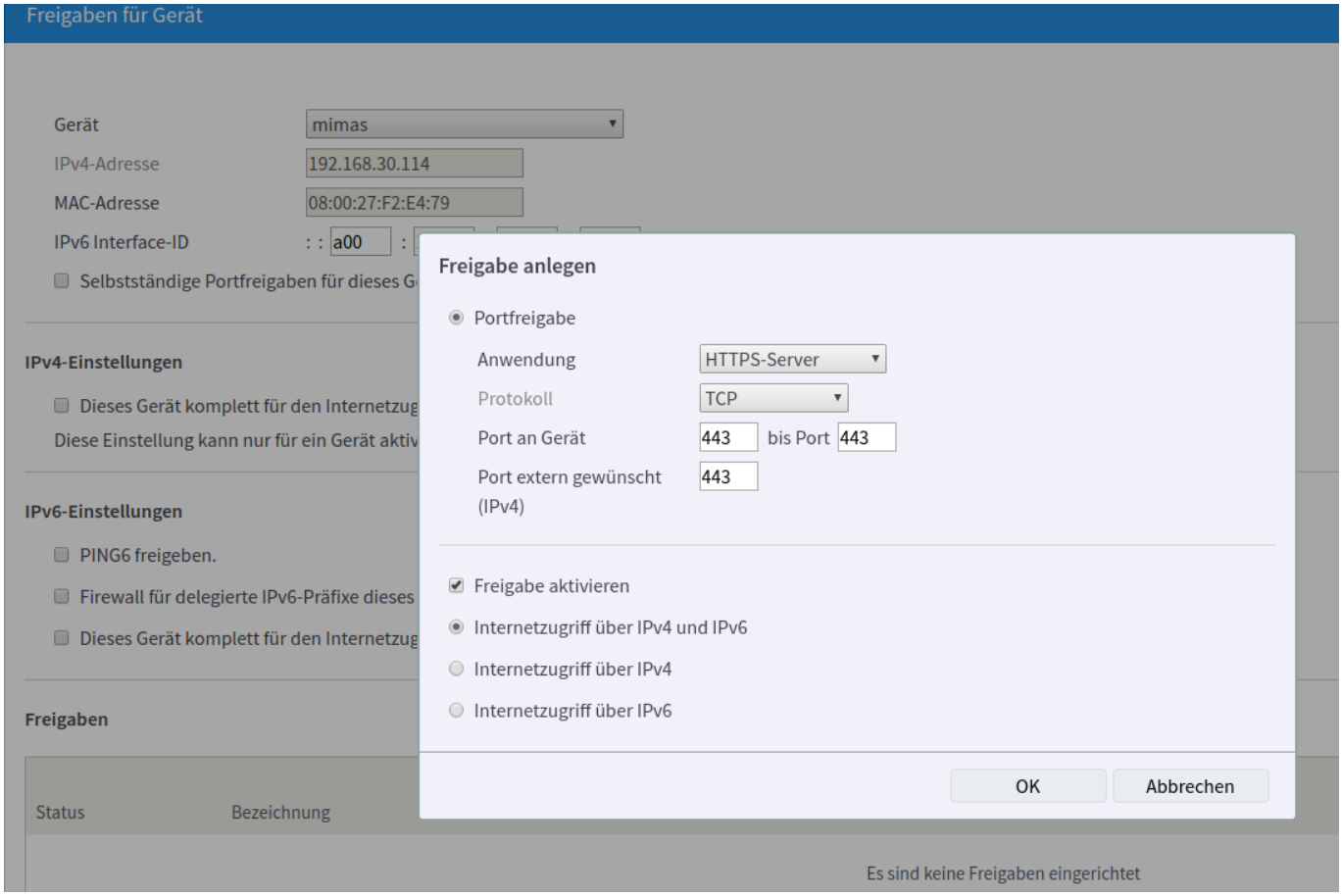
Nach einem Klick auf “Übernehmen” meldet die Fritzbox bei No-IP an und Informiert den Anbieter jedes Mal wenn sich die IP Adresse ändert. Somit funktioniert die Adresse auch dann wenn sich die IP-Adresse am Internetanschluss ändert.

Portfreigabe für den Zugriff auf den Homeserver einrichten

Grundsätzlich ist das Heimnetzwerk damit aus dem Internet erreichbar. Der Versuch über die oben vergebene Adresse auf das Heimnetz zuzugreifen scheitert jedoch an der Fritzbox, die Standardmäßig alle Zugriffe auf das Heimnetz blockt. Damit Zugriffe auf den Homeserver möglich werden muss eine Portfreigabe eingerichtet werden. Auf der Fritzbox findet man die Einstellungen hierzu unter Internet → Freigaben → Portfreigaben. Dort wählt man dann den Button “Gerät für Freigabe hinzufügen”

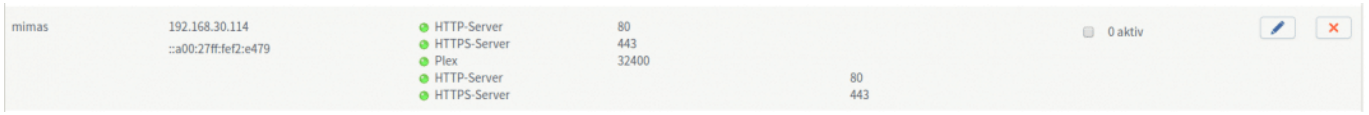
Im Dropdownmenü sollte unser Homeserver mit dem Namen Mimas bereits vorhanden sein und kann ausgewählt werden. Fall das Gerät fehlt, den Homeserver einmal neu starten. Nachdem der Rechner gestartet ist muss die Konfigurationsoberfläche im Browser neu geladen werden. Dann sollte das Gerät erscheinen.

Im unteren Bereich muss zum Erstellen einer Portfreigabe noch der Button “Neue Freigabe” gedrückt werden. Im neuen Fenster wird als Anwendung “HTTPS-Server” ausgewählt, wodurch alle weiteren Felder automatisch ausgefüllt werden. Fall der eigene Router das automatische Ausfüllen nicht anbietet, muss als Portnummer sowohl für interne als auch externe Ports die Nummer 443 angegeben werden. Mit einem Klick auf “OK” wird der Homeserver wirklich über das Internet erreichbar.



Diesen Schritt muss für die Portnummern 80 sowie 32400 wiederholt werden. Port 80 für unverschlüsselte http-Verbindungen wird benötigt um das Let’s Encrypt Zertifikat für die verschlüsselte https-Verbindung auszustellen. Port 32400 wird für das Streaming von Musik und Videos mit Plex verwendet. Für die Freigabe des Ports 32400 wird im Menü bei Anwendung “Andere Anwendung” ausgewählt.

Nach dem Speichern der angelegten Freigaben sieht die Übersicht in der Fritzbox folgendermaßen aus:



Wie man sieht sind folgende Ports freigegeben

- 80
- 443
- 32400

Dateifreigabe im Heimnetz

In den vorangegangenen Teilen wurde das System soweit eingerichtet dass es nun bereit für die Konfiguration der eigentlichen Dienste ist.

In diesem Teil richten wir die Dateifreigaben für das Heimnetz ein. Hierzu wird der Dienst SAMBA

genutzt, da dieser schnelle Datenübertragungen im Netzwerk ermöglicht und von allen Systemen unterstützt wird. Auf die von Samba erstellten SMB/CIFS Freigaben kann mit Windows Computern genau so zugegriffen werden wie mit Linux, MacOS oder mobilen Betriebssystemen. Auch Clients wie das Mediacenter Kodi unterstützen diese Freigaben. Aus Sicherheitsgründen ist der Zugang über Samba aber auf das Heimnetz beschränkt. Für den Dateizugriff über das Internet wird Nextcloud verwendet, was sicherer konfigurierbar ist (SMB ist unter Umständen unverschlüsselt), aber auch langsamer. Beim Zugriff über das Internet spielt das aufgrund der langsameren Internetzugänge jedoch keine Rolle.

In diesem Beispiel werden zwei Benutzer angelegt: heimnetz: Dies ist unser Benutzer. Mit diesem darf auf alle Freigaben lesend und schreibend zugegriffen werden. heimgast: Das Passwort kann an Gäste herausgegeben werden, oder in einem Mediaplayer wie Kodi gespeichert werden. Der Benutzer hat nur Lesezugriff und kann damit keinen Schaden anrichten.

Selbstverständlich können hier auch andere Namen gewählt werden.

Außerdem werden drei Arten von Freigaben eingerichtet. Dokumente, Video, Audio: Auf diese Freigaben hat der Benutzer heimnetz Schreib- und Lesezugriff, der Benutzer heimgast nur Lesezugriff. Private: Eine private Freigabe. Hier hat nur der Benutzer heimnetz Zugriff. Heimgast kann nicht zugreifen. Public: Die öffentliche Freigabe. Hier dürfen alle die Zugang zum Heimnetz haben lesen und schreiben. Es wird kein Passwort benötigt. Diese Freigabe dient dem Austausch von Dateien mit anderen Personen.

Zuerst muss der Sambaserver installiert werden. Die geschieht mit dem Befehl

```
1. sudo apt install samba samba-common
```

Dann werden die beiden Benutzer auf Systemebe angelegt. Da diese Benutzer nur dazu dienen auf Samba-Freigaben zuzugreifen, wird kein Homeverzeichnis angelegt und der Login direkt am System deaktiviert. Die Abfragen von Name, Zimmer usw. können einfach mit Enter übersprungen werden.

```
1. sudo adduser --no-create-home --disabled-login --shell /bin/false
   heimnetz
2. sudo adduser --no-create-home --disabled-login --shell /bin/false
   heimgast
```

Als nächstes werden die Passwörter für beide Benutzer vergeben, die beim Zugriff auf die Freigaben eingegeben werden müssen.

```
1. sudo smbpasswd -a heimnetz
2. sudo smbpasswd -a heimgast
```

Jetzt können wir die Ordner anlegen, welche anschließend im Netzwerk freigegeben werden. Hier muss man darauf achten, dass diese auch auf unserem Storagepool liegen und nicht auf der Systemplatte. Der Storagepool wurde bei der Systeminstallation unter **/mnt/storage** angelegt. Also müssen alle Dateien unterhalb von /mnt/storage gespeichert werden. In diesem Schritt wird der übergeordnete Ordner erstellt, in welchem sich wiederum die Freigaben befinden. Mit dem zweiten

Befehl wechseln wir in diesen Ordner. Innerhalb des Ordners shares werden dann die eigentlichen Ordner für die Freigaben audio, video, dokumente, public und private erstellt. Der letzte Befehl macht den Benutzer heimnetz auf Systemebene zum Besitzer der Ordner.

```
1. sudo mkdir /mnt/storage/shares
2. cd /mnt/storage/shares
3. sudo mkdir audio video dokumente public private
4. sudo chown -R heimnetz: /mnt/storage/shares
```

Das Erstellen der Freigaben erfolgt über die Datei /etc/samba/smb.conf welche mit dem Texteditor geöffnet und bearbeitet wird. Wir öffnen die Konfigurationsdate mit dem Editor Nano, da dieser auf der Kommandozeile einfach zu bedienen ist.

```
1. sudo nano /etc/samba/smb.conf
```

In dieser Datei scrollen wir mit der Pfeil-nach-unten Taste ganz ans Ende und fügen folgende Zeilen ein:

```
 1. [Video]
 2. comment = Videos
 3. path = /mnt/storage/shares/video
 4. write list = heimnetz
 5. valid users = heimnetz,heimgast
 6. force user = heimnetz
 7.
 8. [Audio]
 9. comment = Audio
10. path = /mnt/storage/shares/audio
11. write list = heimnetz
12. valid users = heimnetz,heimgast
13. force user = heimnetz
14.
15. [Dokumente]
16. comment = Dokumente
17. path = /mnt/storage/shares/dokumente
18. write list = heimnetz
19. valid users = heimnetz,heimgast
20. force user = heimnetz
21.
22. [Public]
23. comment = Public
24. path = /mnt/storage/shares/public
25. writeable = yes
26. guest ok = yes
27. force user = heimnetz
28.
29. [Private]
```

```
30. comment = Private
31. path = /mnt/storage/shares/private
32. write list = heimnetz
33. valid users = heimnetz
34. force user = heimnetz
```

Mit den Tastenkombinationen **Strg+o** wird die Datei gespeichert. Mit **Strx+x** wird der Editor wieder beendet.

Die erste Angabe in eckigen Klammern ist der Name, mit dem die Freigabe im Netzwerk sichtbar ist. Der Kommentar ist ein beliebiges Wort oder Satz, der die Freigabe beschreibt. Hinter path wird angegeben wo sich die Freigabe im System befindet. Write list ist die Liste mit Benutzern die auf die Freigabe schreiben dürfen. Valid users sind alle Benutzer die auf die Freigabe zugreifen dürfen. Mit force user wird der Systemuser angegeben mit welchem Dateien geschrieben oder verändert werden. Da heimgast auf Systemebene keinerlei Rechte in den freigegebenen Verzeichnissen hat, wird hier immer die Benutzung des Users heimnetz erzwungen. Auf die Rechte beim Zugriff über das Netzwerk hat diese Angabe keinen Einfluss.

Damit die Änderungen wirksam werden muss der Samba-Dienst einmal neu gestartet werden. Dies geschieht mit dem Befehl

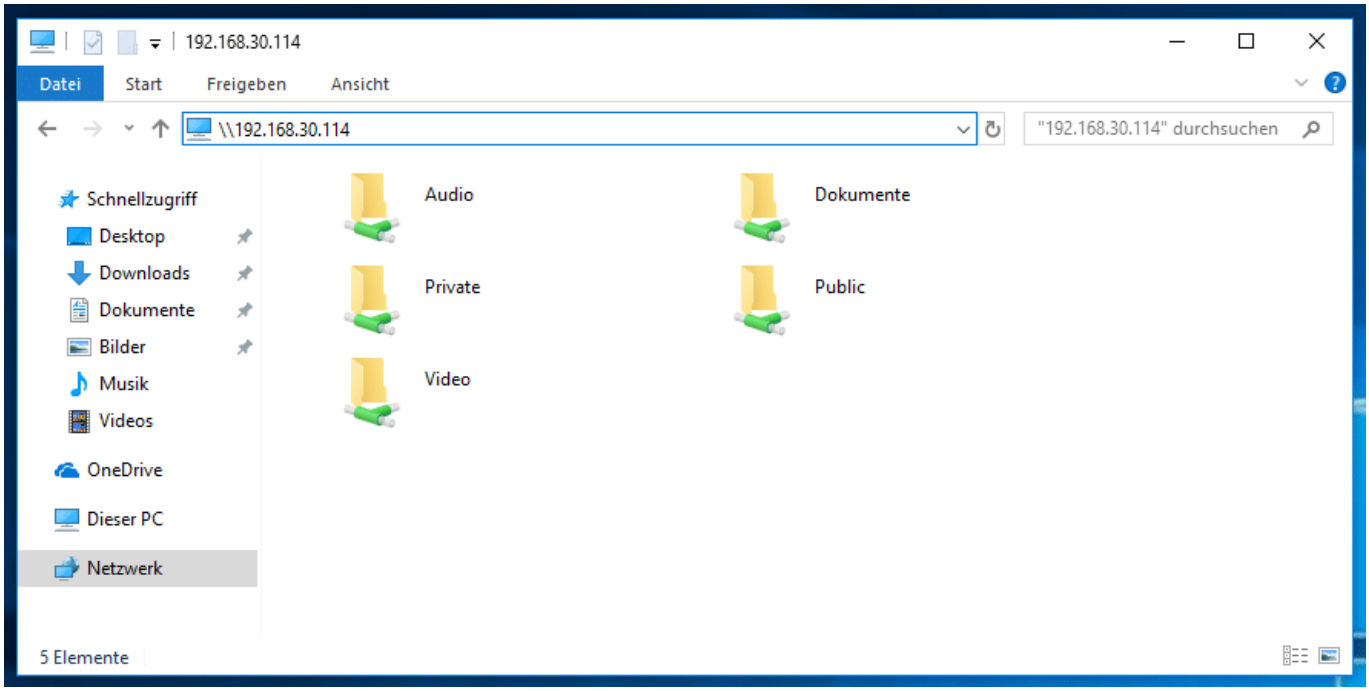
```
1. sudo systemctl restart smbd.service
```

Anschließend sind die Freigaben im Netzwerk verfügbar.

Schnellzugriff auf Freigaben mit Windows 10

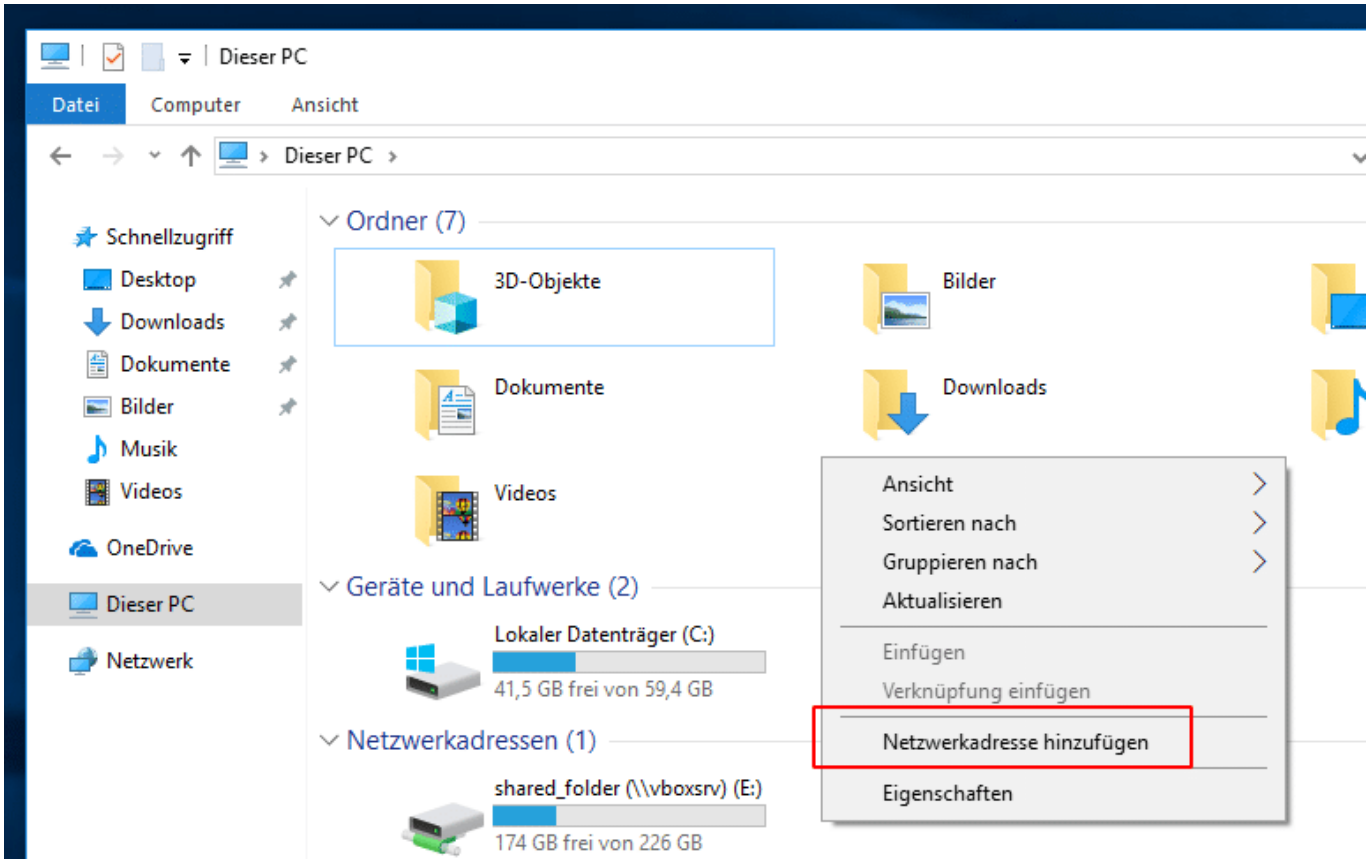
Wenn man nur gelegentlich, oder mit einem fremden Computer auf die Freigaben zugreifen will lohnt es sich nicht diese fest einzubinden. Die schnellste und einfachste Möglichkeit mit Windows 10 auf die Freigaben unseres Homeservers zuzugreifen ist es, einfach in die Adresszeile des Exlorers die Adresse des Homeservers einzugeben. In der Form **\\192.168.30.114**

Damit werden alle Freigaben die unter dieser Adresse zu finden sind angezeigt.

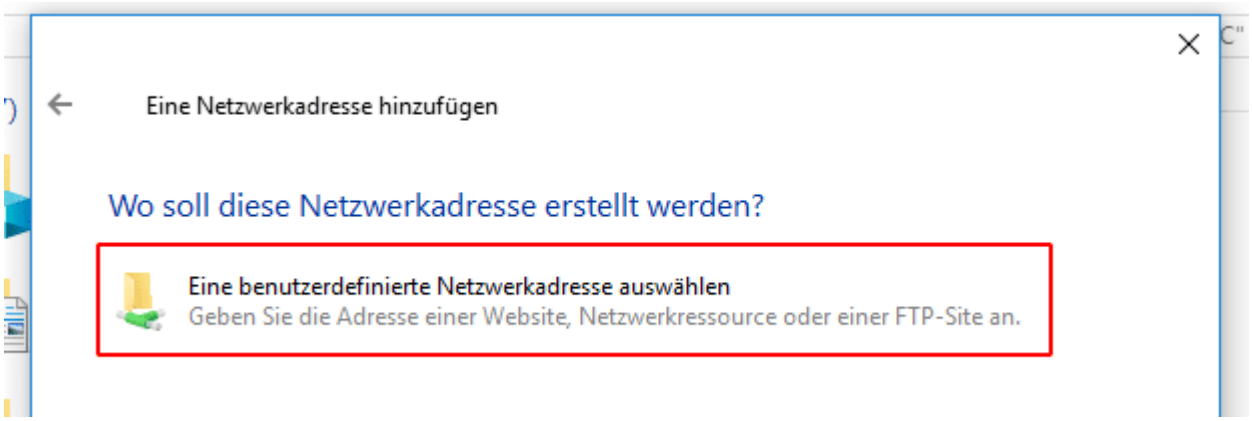


Freigaben unter Windows 10 dauerhaft als Netzlaufwerk einbinden

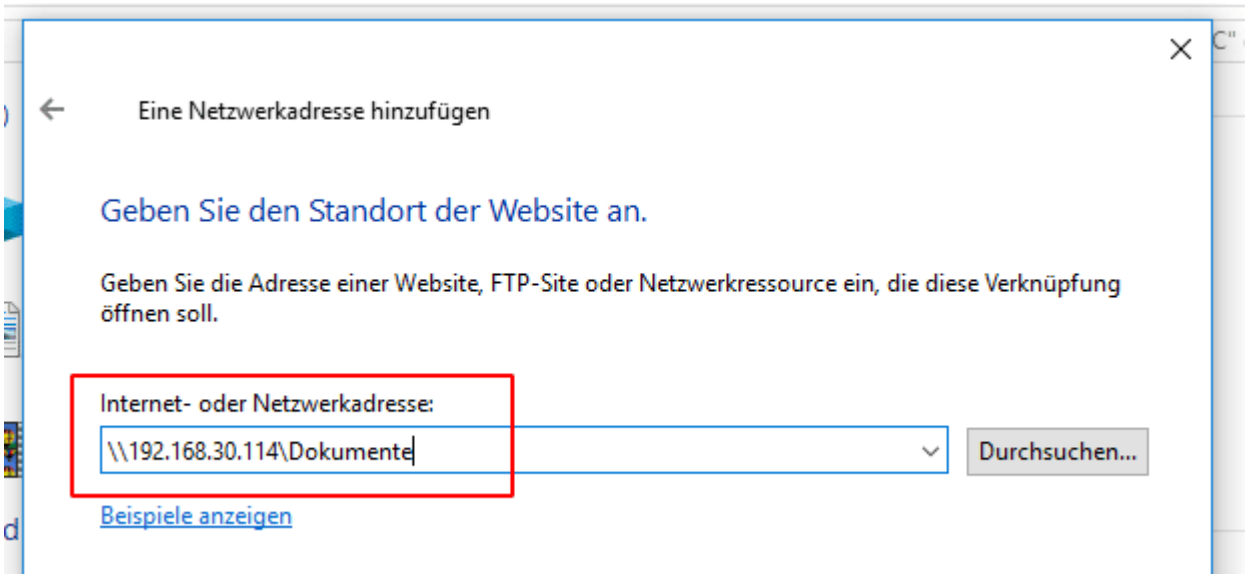
Für eine regelmäßige Nutzung ist der Schnellzugriff zu unbequem. Für diesen Fall ist es sinnvoller die Freigaben dauerhaft als Netzlaufwerk im Windows Explorer einzubinden. Zur Einrichtung öffnet man den Dieser PC und macht einen Rechtsklick auf eine freie Stelle. Dann klickt man auf Netzwerkadresse hinzufügen.



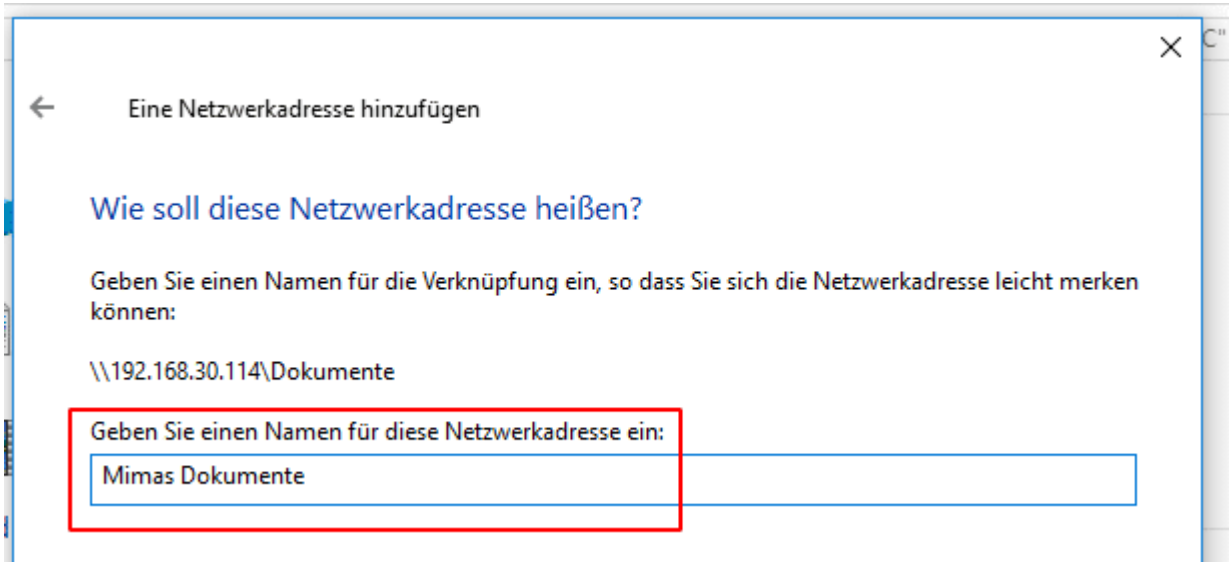
Im nächsten Schritt wählt man “Eine Benutzerdefinierte Netzwerkadresse auswählen“ und klickt auf weiter. Je nach Windowsversion ist dies unter Umständen auch die einzig auswählbare Option.



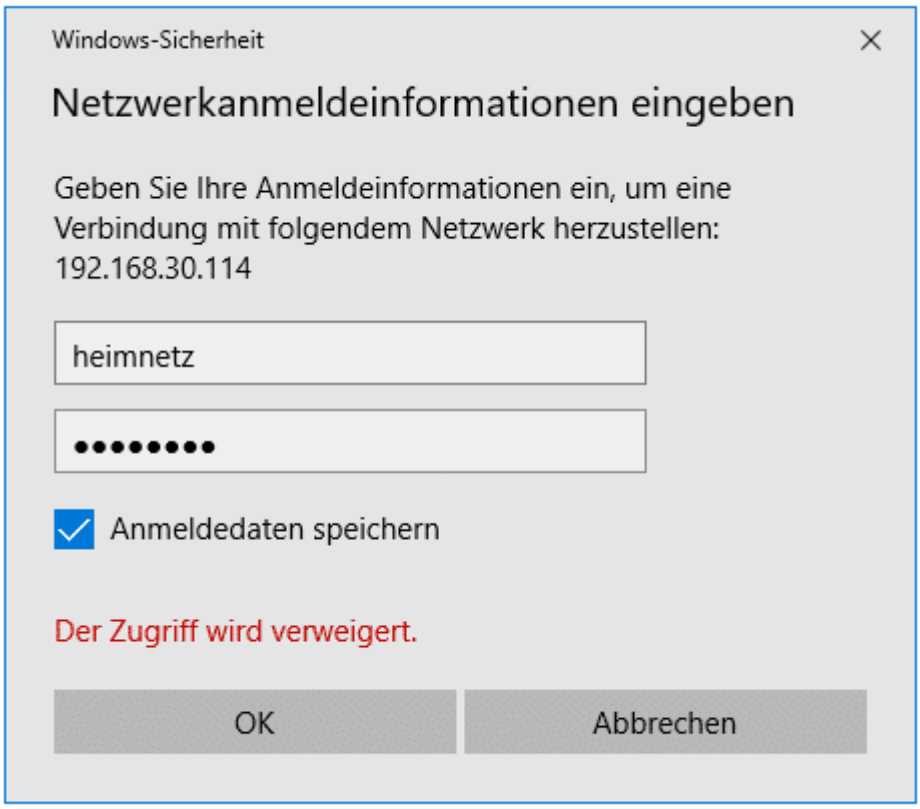
In das Feld zur Eingabe der Netzwerkadresse geben wir die IP Adresse unseres Homeservers gefolgt von der Freigabe in Form von “\\192.168.178.114\Dokumente” ein.



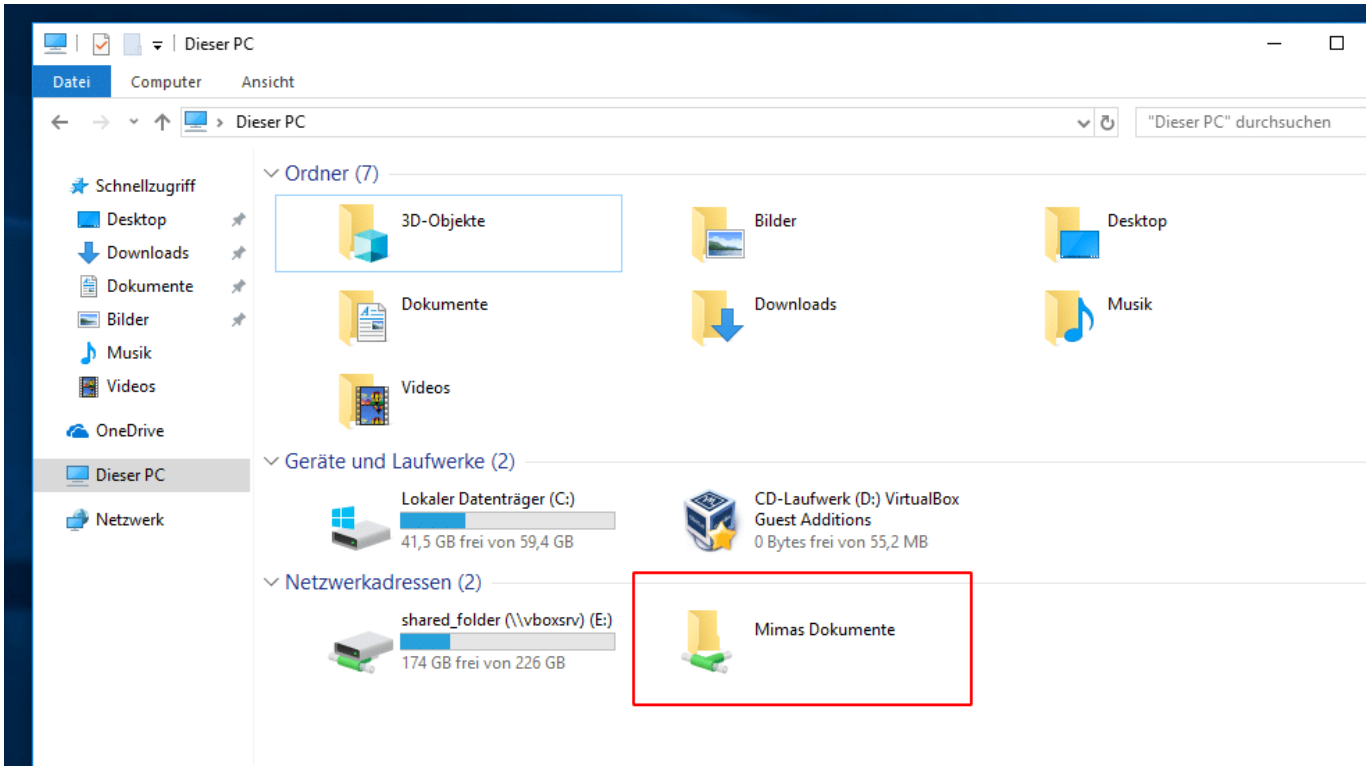
Im nächsten Schritt können wir einen beliebigen Namen für die Freigabe eingeben, z.B. “Mimas Dokumente”



Nach einem Klick auf weiter und fertig stellen, kommt die Passwortabfrage. Hier geben wir die Zugangsdaten des Benutzers heimnetz ein und setzen einen Haken bei “Anmeldedaten speichern“.

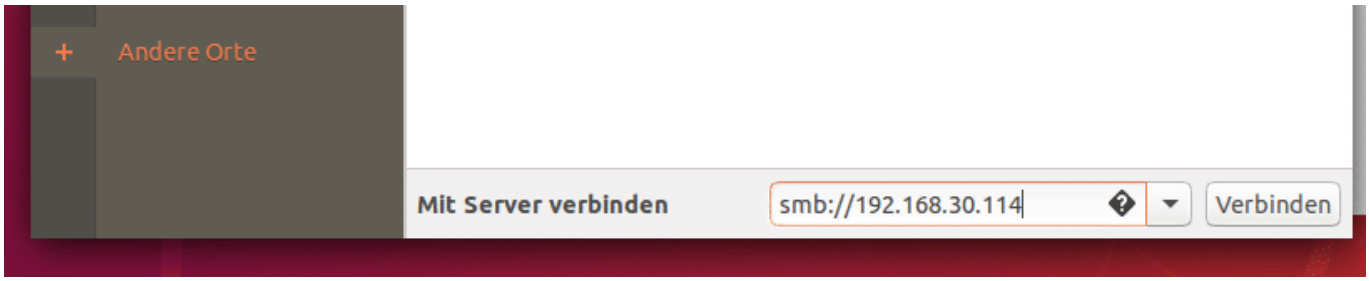


Damit ist die Freigabe im Explorer unter Dieser PC fest eingebunden und verfügbar.



Schnellzugriff auf die Freigaben mit Ubuntu 18.04

Auch unter Ubuntu ist ein einfacher Schnellzugriff auf die Freigaben möglich. Dies ist besonders geeignet wenn die Freigaben nicht dauerhaft verfügbar sein müssen, z.B. für einen Gast. Für den schnellen Zugriff öffnet man Dateien und wählt in der linken Menüleiste Andere Orte. Im unteren Bereich gibt es das Feld mit Server verbinden. Hier geben wir die Adresse unseres Homeservers in der Form smb://192.168.30.114



Mit einem Klick auf Verbinden werden alle unter dieser Adresse gefundenen Freigaben angezeigt.

Dauerhaftes Einbinden der Freigaben auf dem Ubuntu Desktop-System

Bei allen selbst genutzten Computern ist es praktischer die Freigaben dauerhaft zu mounten, damit man immer ein bequemer Zugriff auf den Homeserver möglich ist. Dies ist ein bisschen aufwändiger als unter Windows, dafür sind die Freigaben anschließend nahtlos in das Dateisystem eingebunden und können genau wie lokale Ordner auf der Festplatte genutzt werden.

Damit Ubuntu Sambafreigaben beim Start mounten kann muss das Paket **cifs-utils** installiert werden.

```
1. sudo apt install cifs-utils
```

Anschließend legen wir die Ordner an, in denen die Freigaben unseres Homeservers später zu finden sein werden. In diesem Fall wird ein Ordner namens Homeserver erstellt und unterhalb je ein Ordner für die Freigaben und wechseln anschließend wieder in unser Homeverzeichnis zurück.

```
1. mkdir Homeserver
2. cd Homeserver
3. mkdir Dokumente Audio Video Public Private
4. cd
```

Damit die Zugangsdaten nicht nach jedem Neustart neu eingegeben werden müssen, werden diese in einer Datei abgelegt, auf welche nur der Rootuser zugriff hat. Zuerst wird die Datei erstellt und anschließend mit dem Texteditor geöffnet

```
1. mkdir .credentials
```

```
2. touch .credentials/smbcredentials
3. nano .credentials/smbcredentials
```

in diese Datei kommen Benutzername und Passwort in der Form

```
1. username=heimnetz
2. password=MEINPASSWORT
```

Die Datei wird mit Strg+o gespeichert und mit **Strg+x** wird der Texteditor wieder geschlossen. Aus Sicherheitsgründen werden nun die Lese und Schreibrechte zu dieser Date so angepasst, dass nur der Rootuser Zugriff darauf hat.

```
1. sudo chown root: .credentials/smbcredentials
2. sudo chmod 600 .credentials/smbcredentials
```

Damit Ubuntu die Freigaben kennt müssen diese in die Datei /etc/fstab eingetragen werden. Die Datei wird wieder mit dem Texteditor geöffnet und folgender Text am Ende der Datei eingefügt

```
1. sudo nano /etc/fstab
```

```
1. #Mount Homeserver
2. //192.168.30.114/Dokumente /home/MEINUSERNAME/Homeserver/Dokumente cifs
   credentials=/home/MEINUSERNAME/.credentials/smbcredentials,users,uid=10
   00,gid=1000 0 0
3. //192.168.30.114/Audio /home/MEINUSERNAME/Homeserver/Audio cifs
   credentials=/home/MEINUSERNAME/.credentials/smbcredentials,users,uid=10
   00,gid=1000 0 0
4. //192.168.30.114/Video /home/MEINUSERNAME/Homeserver/Video cifs
   credentials=/home/MEINUSERNAME/.credentials/smbcredentials,users,uid=10
   00,gid=1000 0 0
5. //192.168.30.114/Public /home/MEINUSERNAME/Homeserver/Public cifs
   credentials=/home/MEINUSERNAME/.credentials/smbcredentials,users,uid=10
   00,gid=1000 0 0
6. //192.168.30.114/Private /home/MEINUSERNAME/Homeserver/Private cifs
   credentials=/home/MEINUSERNAME/.credentials/smbcredentials,users,uid=10
   00,gid=1000 0 0
```

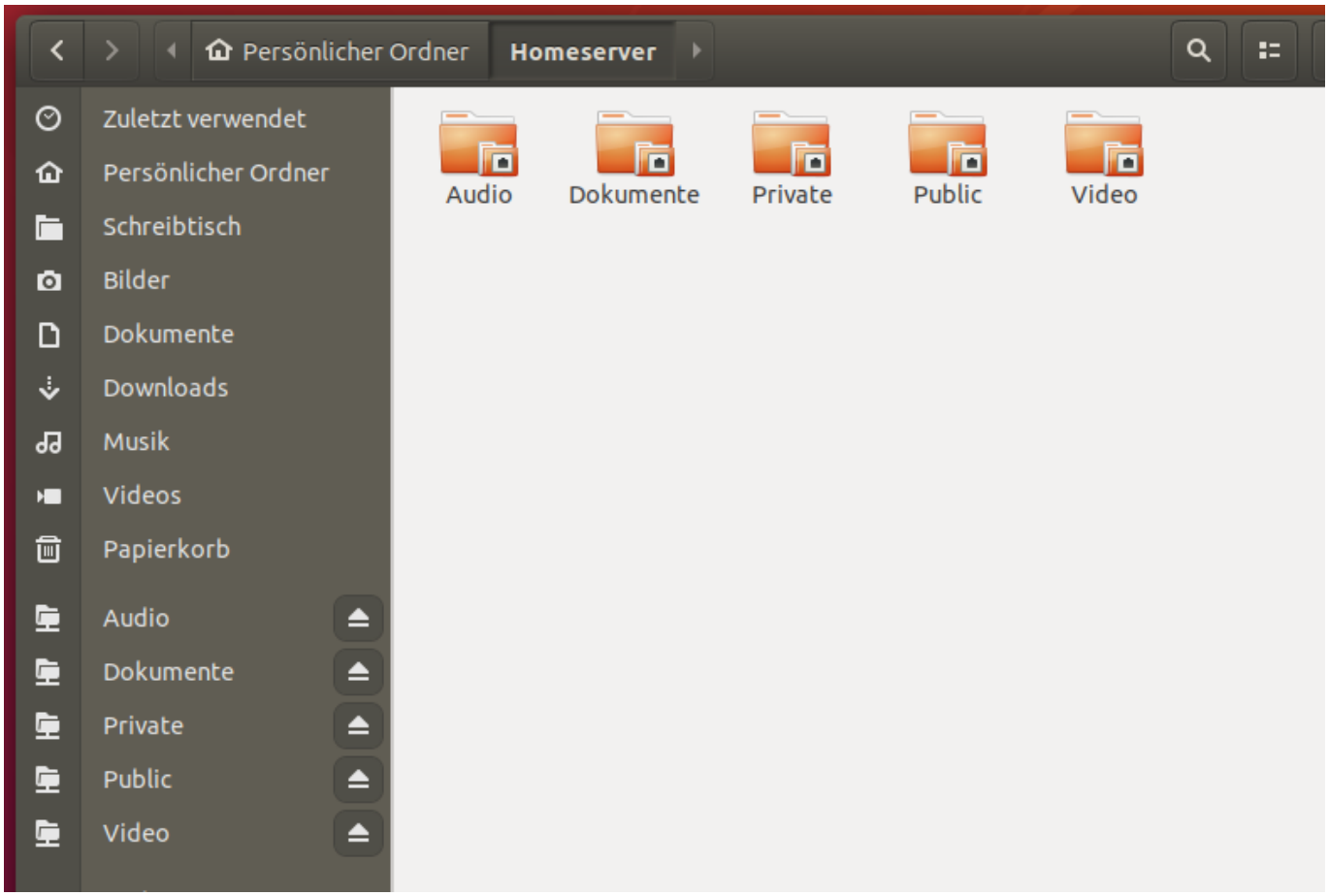
Es muss darauf geachtet werden dass jeder Eintrag in einer Zeile eingetragen wird. Also dass jede Zeile mit //192.168.30… beginnt.

MEINUSERNAME muss durch den eigenen Benutzername ersetzt werden, mit welchem man sich an seinem Laptop oder Desktop angemeldet hat. Bei UID und GID muss es sich um die IDs des eigenen Benutzers handeln. Bei einer Standardinstallation mit nur einem Benutzer ist dies 1000 und muss dementsprechend nicht angepasst werden. Wenn aber mehrere Benutzer einen Computer nutzen, haben diese unterschiedliche IDs. Welche das ist lässt sich im Terminal mit dem einfachen Befehl id

herausfinden.

Nach einem Neustart sollten die Freigaben unseres Homeservers nun im lokalen Dateisystem eingebunden sein, ganz so als würden sie auf der lokalen Festplatte liegen. Wer ungeduldig ist, oder den Rechner nicht neu starten möchte kann das Einbinden der Freigaben auch manuell erledigen mit dem Befehl

```
1. sudo mount -a
```



Installtion von NextCloud

Nextcloud ist die Eierlegende-Wollmilchsau für selbstgehostete Cloudanwendungen. Das Projekt findet sich unter nextcloud.com, ist OpenSource und kann kostenlos heruntergeladen und genutzt werden.

In Nextcloud können Kalender und Kontakte gespeichert werden und mit dem Smartphone synchronisiert werden. Nextcloud kann Dienste wie Google Calendar komplett ersetzten. Außerdem gibt es einen Desktopclient mit welchem Dateien über mehrere Computer synchronisiert werden können, genau wie das bei Diensten wie Dropbox der Fall ist. Eine App für das Smartphone ermöglicht den Zugriff auf die Dateien von unterwegs. Außerdem ermöglicht die App den sofortigen Upload von Fotos oder Videos die mit dem Smartphone erstellt werden. Nextcloud bietet ein übersichtliches Webinterface, über welches man von jedem Computer mit jedem Webbrowser auf die Dateien auf

dem Homeserver zugreifen kann.

Nextcloud ist in PHP geschrieben und benötigt einen Webserver und eine Datenbank.

Als Webserver wird Apache und als Datenbank MariaDB installiert.

Installation der MariaDB-Datenbank

MariaDB ist ein Datenbank-Server, welcher kompatibel zu Oracles MySQL-Server ist. Da Oracle den freien MySQL-Server sehr stiefmütterlich behandelt wird zunehmend auf MariaDB als Alternative gesetzt.

Der Datenbank-Server wird installiert mit

```
1. sudo apt install mariadb-server
```

Mit dem nächsten Befehl kann die MariaDB Installation abgesichert, bzw. für den Produktivbetrieb bereit gemacht werden indem z.B. der anonyme Benutzer gelöscht wird.

Im Gegensatz zu anderen Distributionen, oder wenn MariaDB aus anderen Quellen installiert wird, findet bei Ubuntu die Authentifizierung über einen Unix-Socket mit den Rechten des lokalen Benutzer statt. Wenn also ein MariaDB-Befehl als Root oder mit sudo aufgerufen wird, wird dieser auch als Rootuser im Datenbankserver ausgeführt. Ein lokaler Benutzer der sudo nicht nutzen darf (z.B. die bereits angelegten User heimnetz und heimgast) hat auch keinen Rootzugriff auf den Datenbankserver. Aus diesem Grund ist es nicht nötig und nicht sinnvoll in dieser Konfiguration ein Rootpasswort für den Datenbankserver zu vergeben. Wer sich für weitere Details interessiert findet hierzu einen interessanten Beitrag im Blog von Michael Kofler.

Die Absicherung erfolgt nun mit folgendem Befehl und folgenden Antworten:

```
1. sudo mysql_secure_installation
```

1. Enter current password for root (enter for none): **Enter**
2. Set root password? [Y/n]: **N**
3. Remove anonymous users? [Y/n]: **Y**
4. Disallow root login remotely? [Y/n]: **Y**
5. Remove test database and access to it? [Y/n]: **Y**
6. Reload privilege tables now? [Y/n]: **Y**

Anlegen der Datenbank für Nextcloud

Nach der Installation des Datenbabank-Servers kann die Datenbank für die Nextcloud angelegt werden. Als erstes muss man sich auf dem Datenbank-Server einloggen.

```
1. sudo mysql -u root
```

Dann wird eine neue Datenbank mit dem Namen nextcloud angelegt

```
1. CREATE DATABASE nextcloud;
```

Im nächsten Schritt wird ein neuer Datenbankbenutzer für die Nextcloud Installation vergeben. Für diesen Benutzer wird ein Passwort vergeben, da dieser im Gegensatz zum Rootuser nicht an ein lokales Benutzerkonto gekoppelt ist. Das Passwort muss natürlich individuell festgelegt werden.

```
1. CREATE USER 'nextcloud'@'localhost' IDENTIFIED BY 'password';
```

im letzten Schritt werden dem Benutzer nextcloud alle Rechte an der gleichnamigen Datenbank eingeräumt.

```
1. GRANT ALL ON nextcloud.* TO 'nextcloud'@'localhost' IDENTIFIED BY
   'password' WITH GRANT OPTION;
```

Damit ist die Datenbank bereit. Die Änderungen werden nur noch gespeichert und anschließend loggen wir uns aus dem Datenbankserver wieder aus.

```
1. FLUSH PRIVILEGES;
2. EXIT;
```

Installation des Apache Webserver

Nun wird der Apache Webserver installiert. Die Wahl fällt auf Apache, weil dieser weit verbreitet, gut dokumentiert ist und in der offiziellen Dokumentation von Nextcloud verwendet wird. Das Ausführen der PHP-Dateien übernimmt mod_php, eine Erweiterung für Apache.

Apache und mod_php werden mit diesem Befehl installiert:

```
1. sudo apt install apache2 libapache2-mod-php7.2
```

Außerdem benötigt Nextcloud weitere PHP-Module, welche mit den folgenden Befehl installiert werden:

```
1. sudo apt install php7.2-gd php7.2-json php7.2-mysql php7.2-curl php7.2-
   mbstring php7.2-intl php-imagick php7.2-xml php7.2-zip
```

Damit ist der Webserver grundsätzlich schon einsatzbereit. Ein Aufruf im Webbrowser über http://mimashome.ddns.org oder die IP-Adresse http://192.168.30.114 gibt bereits eine Webseite aus.

BILD LINK

Einrichten des Let’s Encrypt https-Zertifikats (Auch Eingenzertifizierungen funktionieren!)

Damit die Übertragung der Daten über das Internet verschlüsselt und ohne störende Fehlermeldungen des Browsers erfolgen kann, wird ein Schlüsselpaar und ein Verschlüsselungszertifikat benötigt. Seit der Gründung von Let’s Encrypt sind solche, von allen Browsern und Betriebssystemen anerkannte Zertifikate nicht nur kostenlos sondern auch automatisiert und sofort zu erhalten. Ein echter Meilenstein für das Internet.

Die Erstellung der Schlüssel und das Beantragen, sowie die anschließende Zertifikatsverwaltung übernimmt die Software Certbot, welche zuerst wieder installiert werden muss.

```
1. sudo add-apt-repository ppa:certbot/certbot
2. sudo apt update
3. sudo apt install certbot python-certbot-apache
```

Dann wird ein Schlüsselpaar erstellt und ein Zertifikat für unsere DynDNS Adresse beantragt.

```
1. sudo certbot --apache certonly
```

Hierzu müssen ein paar Fragen beantwortet werden

- Enter email address (used for urgent renewal and security notices) (Enter ‘c’ to cancel): deine Emailadresse für Benachrichtigungen
- Please read the Terms of Service. (A)gree/(C)ancel: **A**
- Would you be willing to share your email address with the Electronic Frontier Foundation. (Y)es/(N)o: **Y** oder **N**, je nachdem ob man Mitteilungen Lets Encrypt bekommen möchte
- No names were found in your configuration files. Please enter in your domainname(s) (comma and/or space separated) (Enter ‘c’ to cancel): **mimashome.ddns.net**

Jetzt werden die Daten überprüft und die Zertifikate erstellt und unter /etc/letsencrypt/live/mimashome.ddns.net gespeichert.

Wichtig: Dieser Punkt muss erfolgreich abgeschlossen werden, ansonsten werden die weiteren Schritte, sowie der Zugriff auf Nextcloud nicht funktionieren. Wenn die Erstellung der Zertifikate fehlschlägt, liegt wahrscheinlich ein Fehler bei der DynDNS-Konfiguration oder der Portweiterleitung im Router vor.

Wenn die Verifizierung der Domain und die Erstellung der Zertifikate erfolgreich war, beendet sich Certbot mit folgendem Hinweis.

Automatische Erneuerung der Zertifikate einrichten

Aus Sicherheitsgründen sind Let's Encrypt Zertifikate nur drei Monate gültig. Sofern einem das Zertifikat abhanden kommt, ist der potentielle Schaden geringer als bei einem Zertifikat mir langer Laufzeit. Um ein wartungsarmes System zu bekommen muss die Aktualisierung der Zertifikate automatisiert werden. Dies übernimmt ein Cronjob, der wiederum regelmäßig einen Certbot-Befehl zum erneuern des Zertifikats ausführt.

Hierzu öffnen wir die Datei **/etc/crontab**

```
1. sudo nano /etc/crontab
```

und fügen vor der abschließenden Raute den folgenden Befehl ein:

```
1. @weekly root certbot renew
```

die crontab sollte dann etwa so aussehen

Damit wird einmal wöchentlich die Gültigkeitsdauer aller vorhandenen Zertifikate überprüft und bei Bedarf eine Neuausstellung beantragt.

Installation von Nextcloud und Konfiguration des Apache Webserver

Als erstes besorgen wir uns die Nextcloud Dateien. Diese findet man auf der offiziellen Webseite unter Downloads. Die Dateien können aber auch direkt über die Kommandozeile heruntergeladen werden, wobei wir zuerst in unser Homeverzeichnis wechseln und dann den Download starten.

```
1. cd ~
2. wget https://download.nextcloud.com/server/releases/latest.tar.bz2
```

Damit wird automatisch die aktuelle Version von Nextcloud heruntergeladen. Nach dem Download werden die Dateien nach **/var/www/html** entpackt. Dies ist das Standardverzeichnis in welchem der Apache Webserver Webseiten erwartet. Beim entpacken wird automatisch das Unterverzeichnis Nextcloud erstellt. Anschließend werden die Dateirechte auf den Benutzer des Apache Webservers geändert, so dass Apache die Dateien lesen und für Updates auch beschreiben darf.

```
1. sudo tar -xjvf latest.tar.bz2 -C /var/www/html/
2. sudo chown -R www-data:www-data /var/www/html/nextcloud
```

Apache konfigurieren

Nun folgt noch die Konfiguration des Webservers. Für jeden Dienst der über Apache ausgeliefert wird, wird eine sogenannte Virtual-Host unter /etc/apache2/sites-available Konfiguration erstellt. Diese erstellen wir mit dem folgenden Befehl und öffnen die leere Datei anschließend mit dem Texteditor.

```
1. sudo touch /etc/apache2/sites-available/001-nextcloud.conf #touch als
   Texteditor oder
2. sudo nano /etc/apache2/sites-available/001-nextcloud.conf  #nano als
   Texteditor.
```

In diese Date wird folgender Inhalt kopiert:

```
 1. <VirtualHost *:80>
 2. ServerName mimashome.ddns.net
 3.
 4. ErrorLog ${APACHE_LOG_DIR}/error.log
 5. CustomLog ${APACHE_LOG_DIR}/access.log combined
 6.
 7. Redirect permanent / https://mimashome.ddns.net
 8. </VirtualHost>
 9.
10. <IfModule mod_ssl.c>
11. SSLStaplingCache shmcb:/tmp/stapling_cache(128000)
12. <VirtualHost *:443>
13.
14. ServerName mimashome.ddns.net
15.
16. DocumentRoot /var/www/html/nextcloud
17.
18. ErrorLog ${APACHE_LOG_DIR}/error.log
19. CustomLog ${APACHE_LOG_DIR}/access.log combined
20.
21. # SSL Konfiguration
22. SSLEngine on
23. SSLCertificateFile
    /etc/letsencrypt/live/mimashome.ddns.net/fullchain.pem
24. SSLCertificateKeyFile
    /etc/letsencrypt/live/mimashome.ddns.net/privkey.pem
25. SSLProtocol All -SSLv2 -SSLv3 -TLSv1.1
26.
27. <FilesMatch "\.(cgi|shtml|phtml|php)$">
28. SSLOptions +StdEnvVars
29. </FilesMatch>
30. <Directory /usr/lib/cgi-bin>
31. SSLOptions +StdEnvVars
32. </Directory>
33.
```

```
#Nextcloud Konfiguration
<Directory /var/www/html/nextcloud/>
Options +FollowSymlinks
AllowOverride All

<IfModule mod_dav.c>
Dav off
</IfModule>

SetEnv HOME /var/www/html/nextcloud
SetEnv HTTP_HOME /var/www/html/nextcloud
</Directory>

</VirtualHost>
</IfModule>
```

Nun wird die bereits vorhandene Standardseite deaktiviert und dafür unsere neue Nextcloudkonfiguration aktiviert.

```
sudo a2dissite 000-default.conf
sudo a2ensite 001-nextcloud.conf
```

Außerdem benötigt Nextcloud noch ein paar weitere Apache Module, die mit diesem Befehl aktiviert werden.

```
sudo a2enmod ssl rewrite headers env dir mime
```

Damit alle jetzt vorgenommenen Änderungen aktiviert werden muss der Apache Webserver einmal neu gestartet werden mit

```
sudo systemctl restart apache2
```

Nextcloud einrichten

Jetzt kann Nextcloud über den Webbrowser eingerichtet werden. Dazu wird die angelegte Dyndns Adresse über eine verschlüsselte https-Verbindung aufgerufen. **https://mimashome.ddns.net**. Hier werden wir bereits von der funktionierenden, aber noch nicht eingerichteten Nextcloud empfangen.

Zuerst müssen wir jedoch im Dateisystem noch ein Verzeichnis anlegen in welches Nextcloud unsere Dateien speichert. In der Standardeinstellung würden diese auf der SystemSSD liegen. Wir wollen aber natürlich dass die Dateien auf unserem sicheren Storage-Raid abgelegt werden. Dazu wird das Verzeichnis zuerst erstellt und anschließend die Rechte so angepasst dass Apache und damit Nextcloud Zugriff bekommen.

```
1. sudo mkdir -p /mnt/storage/nextcloud
2. sudo chown -R www-data:www-data /mnt/storage/nextcloud
```

Dann werden die Felder auf der Konfigurationsseite ausgefüllt.

- **Benutzername** und **Passwort** können hier frei vergeben werden. Hiermit loggt man sich später in die Nextcloud ein
- **Dateiverzeichnis** sollte geändert werden, so dass dieses nicht auf der SystemSSD sondern auf unseren Storage-Raid liegt. In diesem Fall /mnt/storage/nextcloud
- **Datenbank-Benutzer** ist **nextcloud**
- **Datenbank-Password** ist das beim Anlegen der Datenbank vergebene Passwort
- **Datenbank-Name** ist **nextcloud**
- **localhost** bleibt unverändert, da sich die Datenbank auf dem gleichen Computer befindet.

Mit einem Klick auf Installation abschließen wird Nextcloud eingerichtet und einsatzbereit gemacht.

BILD-LINK

Damit ist Nextcloud grundsätzlich Einsatzbereit. Allerdings haben wir derzeit nur Zugriff auf die Dateien unter /mnt/storage/nextcloud. Daten die wir über die im vorherigen Artikel angelegt haben sind noch nicht erreichbar.

Dies lösen wir dadurch, dass über die Nextcloud App “external storage support” die bereits vorhandenen Verzeichnisse eingebunden werden. Damit Nextcloud auch Schreibzugriff auf die Dateien hat, die ja mit dem Benutzer heimnetz erstellt werden und nicht mit dem Benutzer des Webservers, benötigen wir noch die Software smbclient, die wieder über das Terminal installiert werden muss. Damit Nextcloud die Änderung mitbekommt starten wir anschließend den Apache einmal neu.

```
1. sudo apt install smbclient
2. sudo systemctl restart apache2
```

Zum aktivieren der App in Nextcloud klickt man in der rechten oberen Ecke auf das **Symbol** mit dem Anfangsbuchstaben des Benutzernamens und anschließend auf **Apps**.

Bei den Apps findet man unter dem Punkt **deaktivierte Apps** die App External Storage Support, welche mit einem Klick auf aktivieren verfügbar gemacht wird.

BILD-LINK

Zum Einbinden der vorhandenen Ordner gehen wir wieder auf das Symbol in der rechten oberen Ecke und anschließend auf Einstellungen. Im linken Bereich findet man verwirrender Weise zwei Punkte mit dem Namen “Externe Speicher“. Zum Einrichten benötigen wir den unteren.

- **Ordnername** wird entsprechend der Freigabe benannt. Z.B. lokale Videos
- **Externer Speicher** ist SMB/CIFS

- **Authentifizierung** ist Benutzername und Passwort
- Konfiguration:

1. **Host** ist localhost
2. **Share** ist der Name der Freigabe. Z.B. Video
3. **Entfernter Unterordner** bleibt leer
4. **Domain** bleibt ebenfalls leer
5. **Benutzername** ist heimnetz oder heimgast. Je nachdem ob die Dateien mit Schreibrechten (heimnetz) oder aus Sicherheitsgründen nur mit Leserechten (heimgast) eingebunden werden sollen
6. **Passwort** ist das zum Benutzername gehörende Passwort

Ein klick auf den kleinen Haken am Ende des Formularfeldes bindet den Ordner in Nextcloud ein.

BILD-LINK

Dies wird für alle Freigaben wiederholt, die in Nextcloud eingebunden werden sollen. Also z.B. Audio, Dokumente…

Anschließend sind die eingebundenen Freigaben über Nextcloud erreichbar.

Media Streaming mit Plex

Plex ist eine Kombination aus Serversoftware und Client. Als Client eignen sich eine Vielzahl von Geräten. So ist Plex als Abspielgerät für Smartphones vorhanden, für die Amazon FireTV Geräte gibt es eine App, außerdem lassen sich Medien über den Webbrowser und viele weitere Geräte abspielen.

Die Serversoftware ist ebenfalls für eine Vielzahl an Plattformen verfügbar, unter anderem als fertiges Installationspaket für Ubuntu.

Plex ist in einer kostenlosen und in einer kostenpflichtigen Version erhältlich. Vorteil der kostenpflichtigen Version ist vor allem dass Inhalte auch auf das Smartphone heruntergeladen werden können um diese offline zu konsumieren. Die Unterschiede zwischen kostenloser und kostenpflichtiger Version werden auf der Homepage von Plex erklärt.

Die Installation und Einrichtung von Plex ist dank der fertigen Installationspakete relativ schnell erledigt.

Um Plex zu nutzen muss man sich zuerst einen kostenlosen Account auf der plex.tv anlegen.

Die Installation von Plex Mediaserver unter Ubuntu 18.04

Zuerst laden wir die Installationsdatei für den kostenfreien Plex Mediaserver herunter. Diese findet man unter plex.tv auf der Downloadseite.

Choose your platform: Linux

Choose Distribution: Ubuntu 64-bit

Hier macht man einen **Rechtsklick** auf den Eintrag und kopiert den Downloadlink in die Zwischenablage mit **Adresse des Links kopieren** o.ä.

BILD-LINK

Jetzt müssen wir uns wieder über ein Terminal oder Putty auf dem Homeserver einloggen, anschließend laden wir den Plex Mediaserver herunter indem wir den Befehl wget gefolgt vom gerade kopierten Downloadlink eingeben. Anschließend wird die Datei installiert.

```
1. wget https://downloads.plex.tv/plex-media-server/...._amd64.deb
2. sudo dpkg -i plex*.deb
```

Standardmäßig legt Plex seine Datenbank unter /var/lib/plexmediaserver auf der Systemplatte an. Natürlich wollen wir diese wieder auf unseren Raidverbund verschieben. Dazu verschieben wir das Verzeichnis und setzen anschließend eine Verknüpfung an die alte Stelle. Ein Vorgehen dass sich bei mir bewährt hat, auch nach dem Restore eines Backups auf einer neuen Systeminstallation wird das Verzeichnis von Plex so akzeptiert.

Um sicherzustellen dass der laufende Plexserver keine Dateien blockiert, wird dieser vor dem Verschieben des Ordners gestoppt und anschließend wieder gestartet

```
1. sudo systemctl stop plexmediaserver
2. sudo mv /var/lib/plexmediaserver /mnt/storage
3. sudo ln -s /mnt/storage/plexmediaserver /var/lib/
4. sudo systemctl start plexmediaserver
```

Damit ist der Plex Mediaserver bereits einsatzbereit und kann konfiguriert werden.

SSL-Zertifikat für den Mediaserver erstellen

Natürlich soll auch die Übertragung unserer Musik und Videos über das Internet verschlüsselt erfolgen. Hierfür können wir wieder das Let’s Encrypt Zertifikat verwenden, das wir bereits für die Nextcloud eingerichtet haben. Allerdings liegt dieses bisher in einer Form vor, die der Plexserver nicht versteht. Das Zertifikat muss also umgewandelt werden. Außerdem muss auch dieser Umwandlungsprozess automatisiert werden, da das Zertifikat ansonsten nach drei Monaten abläuft und nicht mehr funktioniert.

Zuerst wird das bereits vorhandene Zertifikat umgewandelt. Hier muss jeweils der eigene DynDNS Name angepasst werden und ein Passwort vergeben werden. Der untenstehende Befehl muss in einer Zeile in das Terminal kopiert werden.

```
1. sudo openssl pkcs12 -export -nodes -out
   /var/lib/plexmediaserver/mimashome.ddns.net.pfx -inkey
```

```
    /etc/letsencrypt/live/mimashome.ddns.net/privkey.pem -in
    /etc/letsencrypt/live/mimashome.ddns.net/cert.pem -certfile
    /etc/letsencrypt/live/mimashome.ddns.net/chain.pem -passout
    pass:meinpasswort
```

Damit liegt nun unter /var/lib/plexmediaserver/mimashome.ddns.net.pfx und damit an der verlinkten Stelle ein Zertifikat, welches Plex verarbeiten kann.

Damit der Mediaserver das Zertifikat lesen kann, müssen die Dateirechte noch angepasst werden.

```
1. sudo chmod 640 /var/lib/plexmediaserver/mimashome.ddns.net.pfx
2. sudo chown root:plex /var/lib/plexmediaserver/mimashome.ddns.net.pfx
```

Dieser Schritt muss nun immer erfolgen, wenn Certbot ein neues Zertifikat von Let's Encrypt bekommt. Certbot sieht solche Fälle glücklicherweise vor und verarbeitet nach dem Erstellen eines Zertifikats alle skripte welche unter /etc/letsencrypt/renewal-hooks gespeichert sind. Also erstellen wir an dieser Stelle ein Skript und machen dieses ausführbar.

```
1. sudo touch /etc/letsencrypt/renewal-hooks/post/plexcert.sh
2. sudo chmod +x /etc/letsencrypt/renewal-hooks/post/plexcert.sh
```

Die Datei wird wieder mit einem Texteditor geöffnet

```
1. sudo nano /etc/letsencrypt/renewal-hooks/post/plexcert.sh
```

und mit folgendem Inhalt befüllt

```
1. #!/bin/bash
2. openssl pkcs12 -export -nodes -out
   /var/lib/plexmediaserver/mimashome.ddns.net.pfx -inkey
   /etc/letsencrypt/live/mimashome.ddns.net/privkey.pem -in
   /etc/letsencrypt/live/mimashome.ddns.net/cert.pem -certfile
   /etc/letsencrypt/live/mimashome.ddns.net/chain.pem -passout
   pass:meinpasswort && systemctl restart plexmediaserver.service
```

Auch hier muss bitte wieder beachtet werden, dass es sich um einen langen Befehl handelt, der nicht mit einem Zeilenumbruch unterbrochen werden darf. Der Inhalt der Datei muss also folgendermaßen aussehen.

BILD-LINK

Plex Mediaserver konfigurieren

Die Oberfläche des Mediaservers auf unserem selbstgebauten Homeserver/NAS auf Ubuntu 18.04 Basis lässt sich bereits jetzt über jeden Webbrowser aufrufen über die Adresse http:mimashome.ddns.net:32400/web oder http://192.168.30.114:32400/web.

Nach dem Login mit dem bei Plex bereits angelegten Account erfolgt die Servereinrichtung.

In **Schritt 1** muss ein beliebiger Name für den Server vergeben werden. Ich bleibe bei Mimas.

In **Schritt 2** werden Mediatheken hinzugefügt. Dies kann auch später noch geändert oder ergänzt werden. Wir legen zwei Mediatheken an. Eine für unsere Audio- und eine für die Videodateien. Entsprechend wird der Typ Filme/Serien oder Andere Videos gewählt.

- Für die Audiodateien wählen wir **Musik**.
- Als Ordner für die Mediathek wird **/mnt/storage/shares/audio** gewählt.
- Unter Optionen steht uns nur die Möglichkeit eine **einfache Musik-Mediathek** erstellen zur Verfügung. Für die Premium-Mediathek ist ein kostenpflichtiger Plexpass nötig.
- Die Einstellungen unter Erweitert können **unverändert** bleiben.

Die Mediethek wird nun erstellt, die Dateien durchsucht und in die Datenbank des Mediaservers eingetragen. Diese Schritte müssen nun wiederholt werden für alle Medien die über Plex eingebunden und gestreamt werden sollen. Also Audiodateien, Videodateien und Bilder.

In **Schritt 3** folgt ein Hinweis auf die zur Verfügung stehenden Apps, die zum Abspielen auf dem Smartphone oder dem Fernseher benötigt werden. Mit einem Klick auf **Fertig** ist der erste Teil der Konfiguration abgeschlossen.

Plex empfängt einen nun mit der normalen Benutzer- und Konfigurationsoberfkäche, auf der wahrscheinlich schon die gerade hinzugefügten Medien zu sehen sind. Als letzter Schritt muss nur noch die verschlüsselte Verbindung über das Internet konfiguriert werden.

Dazu geht man in die Servereinstellungen, welche man über das **Werkzeugsymbol** in der rechten oberen Ecke erreicht. Anschließend wählt man den Reiter **Server** aus und dann den Punkt **Netzwerk**. Außerdem muss man mit einem Klick auf **erweiterte Optionen** weitere Felder freischalten. Nun werden folgende Eintragungen vorgenommen.

- Sichere Verbindung: **Bevorzugt**, wenn alles funktioniert sollte dies auf Erforderlich umgestellt werden.
- Eigener Zertifikatsspeicherort: **/var/lib/plexmediaserver/mimashome.ddns.net.pfx**
- Eigener Zertifikatsschlüssel: Das Passwort welches für das Zertifikat vergeben wurde
- Eigene Zertifikatsdomain: Die DynDNS Adresse. Hier **mimashome.ddns.net**

Die weiteren Felder müssen nicht ausgefüllt, bzw. verändert werden.

BILD-LINK

Unter dem Punkt Fernzugriff muss nun noch der Zugriff aus dem Internet erlaubt werden und der entsprechende Port angegeben werden. Dazu einen Haken bei **“Öffentlichen Port manuell definieren”** setzen und den Port **32400** angeben. Anschließend auf **“erneut versuchen”** klicken.

Nach einem kurzen Moment sollte die rote Warnmeldung verschwinden und durch eine grüne Erfolgsmeldung ersetzt werden. Wichtig ist hierzu dass die entsprechende Portweiterleitung im Router korrekt gesetzt ist.

BILD-LINK

Damit ist der Plex Mediaserver auf unserem selbstgebauten NAS bzw. Homeserver auch über das Internet erreichbar.

In die Plexapp, oder den Webbrowser muss ab jetzt folgende Adresse eingegeben werden um den eigenen Streamingserver zu erreichen

https://mimashome.ddns.net:32400

Backups mit Duplicati und Rsnapshot

Über die Bedeutung von Backups will ich nicht viele Worte verlieren. Unser Homeserver ist dafür gedacht unsere wichtigsten Daten zu speichern, wie Fotos und Dokumente. Dementsprechend sollte auch für den Fall vorgesorgt werden dass das System ausfällt, oder Dateien versehentlich gelöscht werden. Denn gegen versehentliches löschen hilft auch das beste RAID-System nichts.

In diesem Artikel wird die Einrichtung von zwei verschiedenen Backuplösungen beschrieben. Die Backupsoftware Duplicati wird für ein Cloudbackup eingerichtet. Duplicati lässt sich einfach installieren und anschließend komfortabel über den Webbrowser bedienen.

Außerdem wird ein lokales Backup auf einer externen Festplatte mit Rsnapshot eingerichtet. Rsnapshot ist ein Backuptool für die Kommandozeile und basiert auf der Nutzung von Rsync und Hardlinks. Dies hat den Vorteil, dass die Daten auf der Backupfestplatte von jedem beliebigen LinuxPC wieder gelesen werden können, ohne dass ein spezielles Tool benötigt wird um auf die Daten zugreifen zu können.

Cloudbackup mit Duplicati

Duplicati ist seit vielen Jahren sehr aktiv in der Entwicklung, aber leider gibt es zum derzeitigen Zeitpunkt noch keine finale Version. Allerdings gibt es Beta-Versionen, welche ich seit längerer Zeit ohne Probleme einsetze. Die Entwicklung findet in sogenannten experimental-Versionen statt. In unregelmäßigen Abständen wird eine experimentelle Version, die sich als zuverlässig herausgestellt hat, als Betaversion herausgegeben.

Duplicati ist ein Open Source Projekt. Die Projekthomepage ist unter duplicati.com zu finden, der Code steht auf GitHub. Duplicati ist eine sehr vielseitige Backupsoftware, besonders was die Speicherziele angeht. Neben lokalen Datenträgern wie USB-Festplatten werden eine Vielzahl an Cloudspeichern unterstützt. Darunter z.B. Dropbox, Amazon Cloud Drive, Microsoft OneDrive und viele mehr. Außerdem werden Standardprotokolle wie FTP, SFTP, WebDAV uvm. unterstützt, so dass sich eine Vielzahl weiterer Cloudspeicher, oder ein eigener Server als Backend einsetzen lassen.

Die Bedienung von Duplicati erfolgt komfortabel über den Webbrowser

Die Installation von Duplicati

Zuerst wird der Downloadlink für die aktuelle (Beta-)Version für Ubuntu von der Duplicati Downloadseite benötigt. Diese wird via Rechtsklick und **“Adresse des Links kopieren”** (je nach Browser unterschiedlich) in die Zwischenablage kopiert. Über die Kommandozeile auf unserem Homeserver laden wir die Datei anschließend herunter.

BILD-LINK

```
1. cd
2. wget https://updates.duplicati.com/beta/duplicati_X.X.X.X-X_all.deb
```

Das heruntergeladene .deb Installationspaket kann jetzt installiert werden. Da Duplicati in C# geschrieben ist, wird zur Ausführung unter Linux das Mono-Framework benötigt, welches automatisch aus den Ubuntu Paketquellen mitinstalliert wird. Die Installation erfolgt mit

```
1. sudo apt install ./duplicati_*.deb
```

Standardmäßig erlaubt Duplicati den Zugriff auf das Webinterface nur von der selben Maschine aus. Wir wollen aber natürlich von einem beliebigen Computer aus auf das Webinterface zugreifen. Um dies zu ermöglichen muss die Konfigurationsdatei /etc/default/duplicati angepasst werden. Dazu wird diese wieder zuerst mit einem Texteditor geöffnet.

```
1. sudo nano /etc/default/duplicati
```

Dort erweitern wir die Zeile DAEMON_OPTS mit untenstehenden Optionen.

```
1. DAEMON_OPTS="--webservice-port=8200 --webservice-interface=any"
```

Die Konfigurationsdatei sollte danach folgendermaßen aussehen

BILD-LINK

Jetzt wird der Duplicati-Dienst noch aktiviert, so dass die Software nach einem Neustart automatisch wieder aktiv ist. Außerdem wird Duplicati gestartet, so dass es direkt einsatzbereit ist.

```
1. sudo systemctl enable duplicati.service
2. sudo systemctl start duplicati.service
```

Nun kann Duplicati über den Webbrowser über folgende Adresse aufgerufen werden.

http://192.168.30.114:8200

Die Konfiguration von Duplicati

Zuerst wird man von Duplicati mit einem Sicherheitshinweis konfrontiert. Man kann zuerst ein Passwort für die Benutzeroberfläche festlegen, um zu verhindern dass unbefugte Zugriff auf die Einstellungen und die Backups bekommen. Da die Konfigurationsoberfläche von Duplicati von jedem Computer im Netzwerk erreicht werden kann, ist es sinnvoll hier ein Passwort festzulegen. Dies kann aber auch später in den Einstellungen jederzeit nachgeholt werden.

Die Konfigurationsoberfläche und das Einrichten von Backups mit Duplicati ist selbsterklärend und kinderleicht. Aus diesem Grund verzichte ich auf eine Anleitung zum einrichten des Backups.

Man sollte bei der Einrichtung darauf achten die Ordner **/etc** sowie **/mnt/storage** zu sichern. Im ersten liegen die Konfigurationsdateien des Systems, im zweiten unsere Daten. Auch das Sichern des /home Verzeichnisses ist in der Regel sinnvoll.

Außerdem ist es sinnvoll die Konfiguration des Backups z.B. auf einem USB-Stick zu sichern, oder sich diese für den Fall der Fälle per Email zu senden. Damit ist es möglich die Duplicati Konfiguration auf einem neu aufgesetzten System einfach wieder zu importieren. Hierzu klickt man auf den Namen des Backups, in diesem Fall “Home”. Damit klappen weitere Optionen herunter. Hier findet man unter dem Punkt Konfiguration → Exportieren die Möglichkeit sich alle Einstellungen des Backupjobs “Home” als Datei herunterzuladen.

Lokales Backup mir Rsnapshot

Rsnapshot ist ein Backuptool für die Kommandozeile welches auf rsync basiert. Rsnapshot erstellt, wie der Name bereits andeutet, sogenannte Snapshot-Ordner auf der externen Festplatte. Dabei wird bei jedem Backup ein neuer Ordner erstellt welcher das Backup enthält, wobei es so aussieht als würde jeder Ordner eine vollständige Kopie der gesicherten Dateien enthalten. Durch die Verwendung von rsync wird eine unveränderte Datei aber nur einmal gesichert. Mit einem Hardlink wird die Datei im folgenden Backup verlinkt, wodurch Speicherplatz gespart wird.

Da es sich trotzdem um normale Dateikopien handelt, können die Sicherungen auf der externen Festplatte von jedem beliebigen PC wieder auslesen. Somit ist sichergestellt dass sich die Dateien auch einfach wiederherstellen lassen.

Rsnapshot lässt sich direkt aus den Ubuntu Paketquellen installieren. Um die USB-Festplatte für die Datensicherung automatisch beim einstecken ins System einzubinden, wird zusätzlich das Programm **usbmount** installiert.

```
1. sudo apt install rsnapshot usbmount
```

USBMount konfigurieren

Usbmount ist direkt nach der Installation einsatzbereit. Beim einstecken einer USB-Festplatte wird dieses automatisch in **/media/usb** eingebunden. Steckt man mehrere USB-Laufwerke an, so werden die Einbindungspunkte nummeriert, also **/media/usb0**, **/media/usb1** usw.

Standardmäßig unterstützt **usbmount** die Dateisysteme vfat ext2 ext3 ext4 und hfsplus. Wenn die Backupfestplatte nur für den Homeserver genutzt wird sollte sie mit dem Dateisystem ext4 formatiert werden, da dies das Standarddateisystem bei Ubuntu ist. Wenn man die Festplatte unbedingt auch auf Windows PCs nutzen möchte und die Festplatte somit NTFS formatiert sein muss, muss dieses Dateisystem in der Konfigurationsdatei von **usbmount** aktiviert werden.

Dazu wird die Datei **/etc/usbmount/usbmount.conf** mit dem Texteditor geöffnet und in die Zeile **FILESYSTEMS=** noch **ntfs** hinzugefügt, so dass die Zeile folgendermaßen aussieht:

```
1. FILESYSTEMS="vfat ext2 ext3 ext4 hfsplus ntfs"
```

BILD-LINK

Nun kann die USB-Festplatte eingesteckt und das Backupprogramm Rsnapshot konfiguriert werden.

Rsnapshot konfigurieren

Rsnapshot ist einfach zu Konfigurieren. Die Einstellungen erfolgen über die Konfigurationsdatei in **/etc/rsnapshot.conf**, welche wieder mit einem Texteditor geöffnet wird.

Wichtig: die Trennung von Schlüssel und Werten, also der Zwischenraum zwischen einer Einstellung und der dahinterstehenden Zahl muss mit der Tabulatortaste **[TAB ⇆]** erstellt werden. Wenn hier normale Leerzeichen stehen wird die Konfigurationsdatei von Programm nicht akzeptiert und Rsnapshot bricht mit einer Fehlermeldung ab.

```
1. sudo nano /etc/rsnapshot.conf
```

In dieser Datei müssen wenige Einstellungen vorgenommen werden. Im oberen Teil wird unter **snapshot_root** das Verzeichnis angegeben, in welchem die Sicherungen gespeichert werden. In unserem Fall also **/media/usb**. Außerdem wird die Einstellung **no_create_root** von **0** auf **1** gesetzt, damit vor dem Backup überprüft wird ob der angegebene Mountpunkt überhaupt existiert.

```
1. ############################
2. # SNAPSHOT ROOT DIRECTORY #
3. ############################
4.
5. # All snapshots will be stored under this root directory.
```

```
#
snapshot_root /media/usb

# If no_create_root is enabled, rsnapshot will not automatically create the
# snapshot_root directory. This is particularly useful if you are backing
# up to removable media, such as a FireWire or USB drive.
#
no_create_root 1
```

Als nächstes werden die Aufbewahrungszeiträume festgelegt. In diesem Beispiel wird jedes Backup der letzten sieben Tage aufbewahrt. Von den einmal pro Woche ausgeführten Backups werden vier aufgehoben und von den Monatlichen sechs. Das aktuellste Backup ist somit von heute, bzw von gestern und das älteste Backup ist ein halbes Jahr alt.

Zuerst suchen wir in der Konfigurationsdatei den Block **“Backup Levels“**. Die bereits vorhandenen Levels, die mit alpha, beta usw. benannt sind, werden mit einer Raute (#) auskommentiert oder gelöscht. Darunter fügen wir unsere Konfiguration ein, die wir der besseren Übersicht wegen daily, weekly und monthly nennen. Wichtig ist hier wieder die Trennung mittels Tabulatortaste. Der bearbeitete Block sieht dann folgendermaßen aus.

```
########################################
# BACKUP LEVELS / INTERVALS #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc. #
########################################
#retain alpha 6
#retain beta 7
#retain gamma 4
#retain delta 3
#retain hourly 6
retain daily 7
retain weekly 4
retain monthly 6
```

Im nächsten Schritt wird festgelegt welche Ordner überhaupt gesichert werden sollen. Dies geschieht im Block **“Backup Points / Scripts“**. In diesem Beispiel werden die Verzeichnisse **/home /etc** und **/mnt/storage** gesichert. Mimas gibt den Unterordner auf der der externen Festplatte an unter welchem die Sicherungen gespeichert werden. Das tägliche Backup wird in diesem Fall z.B. unter **/media/usb/daily.0/mimas** erstellt. Der Backupblock sieht aus wie folgt.

```
###############################
### BACKUP POINTS / SCRIPTS ###
###############################
# LOCALHOST
backup /home/ mimas/
backup /etc/ mimas/
```

```
7. backup /mnt/storage/ mimas/
```

Die Konfigurationsdatei kann nun gespeichert werden. Mit dem folgenden Befehl kann überprüft werden ob die die Datei von Rsnapshot verarbeitet werden kann. Wenn es hier zu einer Fehlermeldung kommt, dann wurde wahrscheinlich an irgendeiner Stelle ein Leerzeichen anstelle von Tab verwendet.

```
1. sudo rsnapshot configtest
```

Aktivieren der Rsnapshot Backups und erstellen des ersten Backups

Die Backups werden über einen Cronjob gestartet. Dazu wird die Datei /etc/crontab im Texteditor geöffnet und um folgende Zeilen vor der abschließenden Raute erweitert.

```
1.
```

```
1.
```

```
1.
```

```
1.
```

```
1.
```

```
1.
```

From:
https://jmz-elektronik.ch/dokuwiki/ - **Bücher & Dokumente**

Permanent link:
https://jmz-elektronik.ch/dokuwiki/doku.php?id=start:linux:ubuntu:samba&rev=1553680752

Last update: **2019/03/27 10:59**