Dieses Dokument zeigt eine Mitschrift zur Ausarbeitung eines Arbeitsauftrage/Laborauftrages im Unterrichtsfach **Betriebssysteme & Labor**.

# Firmennetzwerk (Test-Prüfungsaufgaben)

## Inhalt

# 1. Aufgabenstellung

- **Aufsetzen eines Raspberry zum Server**
- **Einrichtung von**
  - **Datei und Web-Dienst (Samba & Apache bzw. LAMP)**
  - **Benutzerverwaltung (Cockpit**
  - **CMS**
  - **Druckdienst**
  - **Überwachung**
  - **Cloud-Lösung**
  - **Automatische Backups (cron jobs)**
  - **Firewall (UFW)**

# 2. Geräteübersicht

| Device (# & Name) | | Rolle | Hypervisor (VM) | Betriebssystem (OS) | IP-Adresse (LAN) |
|---|---|---|---|---|---|
| 1 | WIN11_pro | Client | nein | WIN11 | 192.168.1.222* |
| 2 | Raspberry 400 | Server | nein | Raspberry PI OS Lite (64-bit) | 192.168.1.15* |
| 3 | | | | | |

## 2.1 Netzwerkeinstellungen (Hardware)

| | Device # | # 1 | # 2 | # 3 | # 4 | # 5 |
|---|---|---|---|---|---|---|
| Devices | IP Adr. | 192.168.1.222* | 192.168.1.15* | | | |
| | DHCP/static | DHCP | DHCP | | | |
| | SUB | 255.255.255.0 | 255.255.255.0 | | | |
| | Gateway | 192.168.1.1 | 192.168.1.1 | | | |
| | DNS | | | | | |
| | Mac | | | | | |

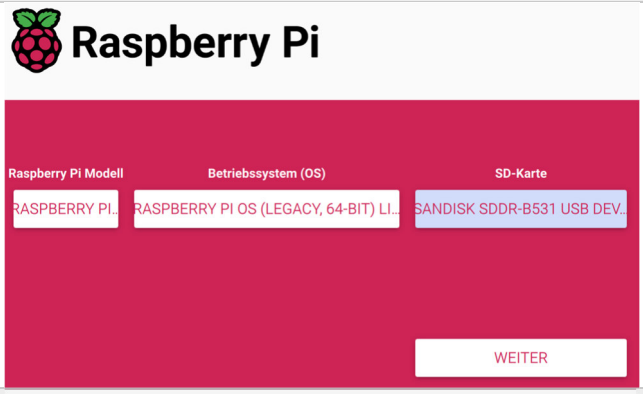
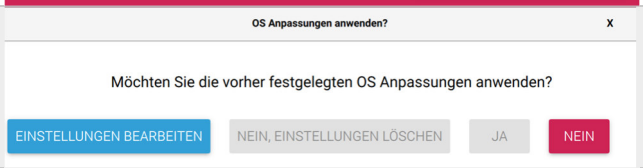
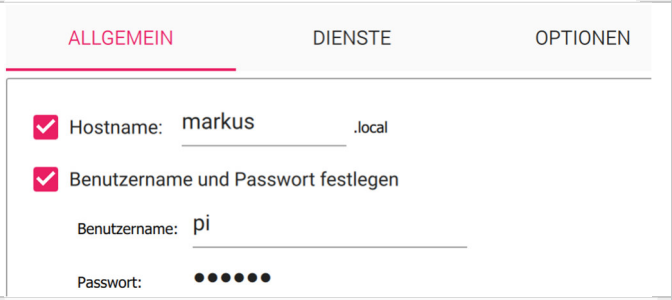
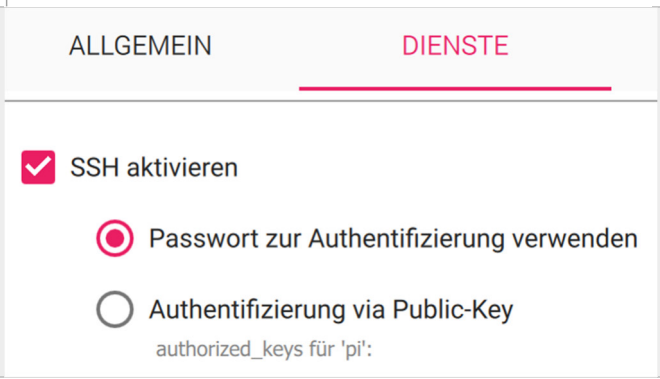
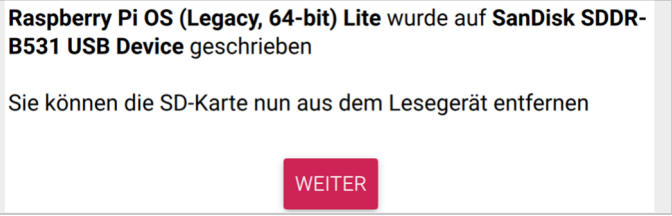
*IP-Adressierung kann durch unterschiedliche Netzwerkstrukturen wechseln (Home/BBRZ_Lan)

## 2.2 User & Zugangsdaten (Software)

| | Geräte # | Anwendung | Benutzername/Rolle | Passwort | Gerätename | Gruppe/domain |
|---|---|---|---|---|---|---|
| Client | Raspberry | OS/Lite 64bit | Pi | Aa_123456 | *pi* | admin |
| | Raspberry | OS/Lite 64bit | anna | useranna | *pi* | vertrieb |
| | Raspberry | OS/Lite 64bit | max | usermax | *pi* | entwicklung |
| | Raspberry | OS/Lite 64bit | lisa | usrlisa | *pi* | support |

| | Geräte # | Anwendung | Benutzername/Rolle | Passwort | Hostname |
|---|---|---|---|---|---|
| Server | Raspberry | OS/Lite 64bit | **Pi**/Admin | Aa_123456 | *markus.local* |
| | | | | | |
| | | | | | |

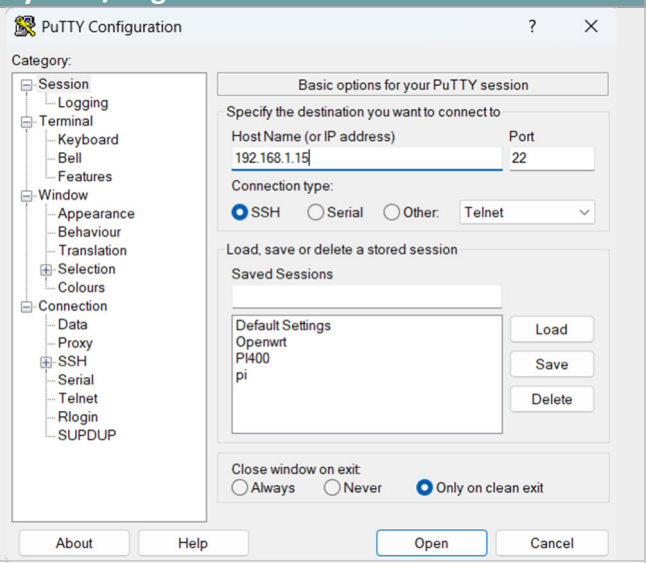
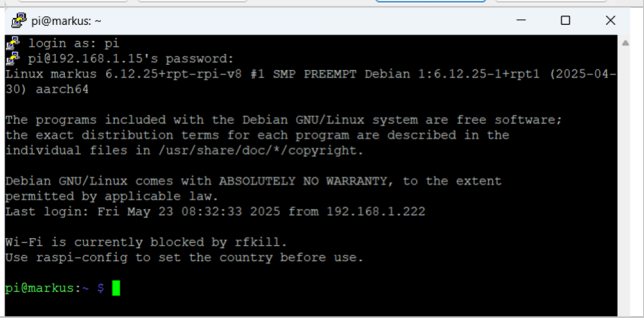
# 3. Grundinstallation Server

| Erklärung | Beispiel |
|---|---|
| Auswahl eines geeigneten Betriebssystems | Raspberry PI OS Lite (64-bit) |
| **Auswahl**<br>- PI-Modell<br>- Betriebssystem<br>- Datenträger |  |
| Einstellungen bearbeiten |  |
| Unter **„Allgemein“**<br><br>Eingabe des Hostnames<br>Festlegen von Benutzernamen & Passwort |  |
| Unter **„Dienste“**<br><br>Aktivierung von SSH |  |
| Bestätigung des Schreibvorganges mit<br><br>JA |  |
| **Wurde der Schreibvorgang beendet, kann die SD-Karte in den Raspi gesteckt und das System gestartet werden.** |  |

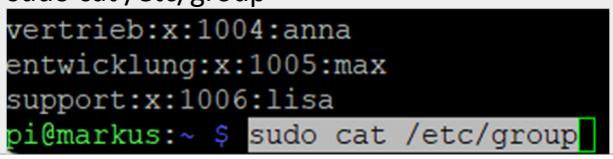
Zugriff auf host-System

Für den Erstzugriff verwende ich Putty.

Aufbau einer

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Aufbau einer SSH-Verbindung per PuTTy**<br><br>login as: **pi**<br>password: **Aa_123456** |  |
| **Login succesfull!**<br><br>**update und upgrade ausführen** |  |

# 4. Benutzer- und Gruppenverwaltung

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Erstellung von User** | sudo useradd anna (max & lisa) |
| **Passwort vergeben (z.B. passanna; passmax; passlisa–)** | sudo passwd anna (max & lisa) |
| **Gruppen erstellen** | sudo groupadd vertrieb (entwicklung & support) |
| **User zu Gruppen hinzufügen** | sudo usermod -aG vertrieb anna<br>sudo usermod -aG entwicklung max<br>sudo usermod -aG support lisa |
| **Gruppenzugehörigkeit ansehen** | sudo cat /etc/group<br>  |

| | |
|---|---|
| **Testen der Login-Funktion** | sudo login anna ( max & lisa) → logout (um wieder zu pi zurückzukommen!)<br>`anna@markus:~ $` |
| | cat /etc/passwd<br>`anna:x:1001:1001::/home/anna:/bin/bash`<br>`max:x:1002:1002::/home/max:/bin/bash`<br>`lisa:x:1003:1003::/home/lisa:/bin/bash`<br>`pi@pilumex5:~ $` |

# 5. Samba-Dateifreigaben

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Installation und Konfiguration Samba** | sudo apt-get install samba samba-common smbclientshare |
| **Benutzerkonten zur Samba-Datenbank hinzufügen** | sudo smbpasswd -a anna<br>`pi@markus:~ $ sudo smbpasswd -a anna`<br>`New SMB password:`<br>`Retype new SMB password:`<br>`Added user anna.`<br>`pi@markus:~ $ sudo smbpasswd -a max`<br>`New SMB password:`<br>`Retype new SMB password:`<br>`Added user max.`<br>`pi@markus:~ $ sudo smbpasswd -a lisa`<br>`New SMB password:`<br>`Retype new SMB password:`<br>`Added user lisa.`<br>`pi@markus:~ $` |
| **Konfig neu laden, Änderungen übernehmen** | sudo service smbd reload |
| **Freigabeordner erstellen für jede Abteilung** | sudo mkdir vertrieb (entwicklung & support) |
| **Rechte für Freigabeordner vergeben:**<br>`bin entwicklung lib mnt root srv tmp vertrieb`<br>`boot etc lost+found opt run support usr`<br>`dev home media proc sbin sys var` | `pi@markus:/ $ sudo mkdir entwicklung`<br>`pi@markus:/ $ sudo mkdir support`<br>`pi@markus:/ $ sudo chmod 777 vertrieb`<br>`pi@markus:/ $ sudo chmod 777 entwicklung`<br>`pi@markus:/ $ sudo chmod 777 support` |
| | |
| **zusätzlichen Ordner für gruppenübergreifenden Zugriff erstellen** | sudo mkdir share2<br>`pi@markus:/ $ sudo chmod 777 share2`<br>`pi@markus:/ $ ls`<br>`bin entwicklung lib mnt root share2 sys var`<br>`boot etc lost+found opt run srv tmp vertrieb`<br>`dev home media proc sbin support usr`<br>`pi@markus:/ $` |
| **Freigabe der Dateien konfigurieren** | sudo nano /etc/samba/smb.conf |
| **Konfigurationseintrag in smb.config für gemeinsamen Ordner:**<br><br>**[share2]**<br>**path = /share2**<br>**read only = no**<br>**browsable = yes**<br>**create mask = 0777** | `[share2]`<br>`path = /share2`<br>`read only = no`<br>`browsable = yes`<br>`create mask = 0777` |

**Konfigurationseintrag in smb.config für Ordner vertrieb**

**[vertrieb]**
**path = /vertrieb**
**browsable = yes**
**read only = no**
**guest ok = no**
**valid users = @vertrieb**

```
[vertrieb]
   path = /vertrieb
   browsable = yes
   read only = no
   guest ok = no
   valid users = @vertrieb
```

**[entwicklung]**
**path = /entwicklung**
**browsable = yes**
**read only = no**
**guest ok = no**
**valid users = @entwicklung**

```
[entwicklung]
   path = /entwicklung
   browsable = yes
   read only = no
   guest ok = no
   valid users = @entwicklung
```

**[support]**
**path = /support**
**browsable = yes**
**writable = yes**
**guest ok = no**
**valid users = @support**

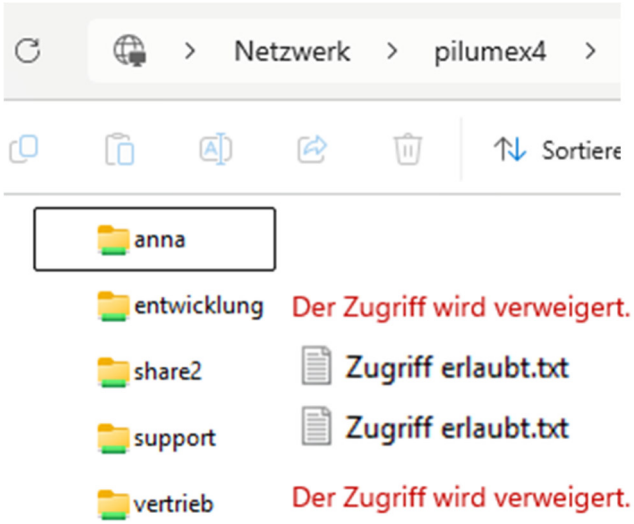
```
[support]
   path = /support
   browsable = yes
   writable = yes
   guest ok = no
   valid users = @support
```

**Exemplarischer Test nur für User „anna“ durchgeführt und dokumentiert, auch für andere User funktioniert diese Vorgehensweise!**

**Zugriff per Dateimanager „Explorer“ von User „anna“:**

**\\pilumex4 (od. \\192.168.0.167)**

gleichzeitig erscheint auch ein home-Ordner des angemeldeten Users?



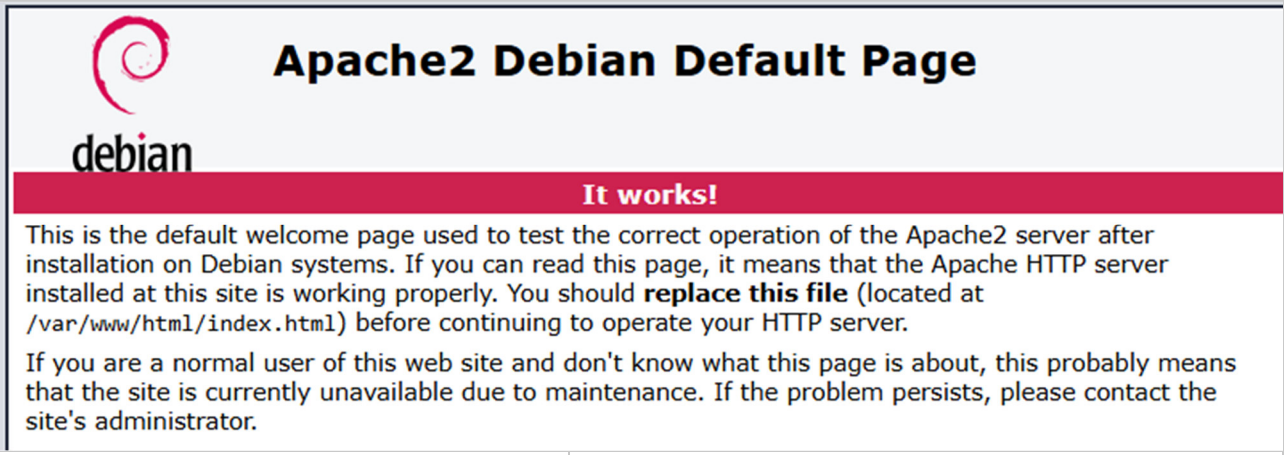
**Wenn Zugriff auf Ordner nicht möglich:**

**net use * /delete**


```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\tryit>net use * /delete
Sie verfügen über folgende Remoteverbindungen:

    Z:              \\NAS6CE19A\Bunker
                    \\pilumex4\IPC$
Fortsetzen beendet die Verbindungen.

Möchten Sie diesen Vorgang fortsetzen? (J/N) [N]: J
Der Befehl wurde erfolgreich ausgeführt.


C:\Users\tryit>
```

# 6. Apache mit PHP

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Update & upgrade durchführen** | sudo apt update<br>sudo apt upgrade |
| **Installation von „apache2“** | sudo apt install apache2 |




| | |
|---|---|
| **Update & upgrade durchführen** | sudo apt update<br>sudo apt upgrade |
| **Installation von „php“** | sudo apt-get install php |
| **Installation von „libapache2-mod-php“**<br>(ist ein Paket, das das PHP-Modul für den Apache 2 Webserver bereitstellt. Es ermöglicht Apache, PHP-Skripte auszuführen, die in HTML-Dokumente eingebettet oder als eigenständige Web-Anwendungen verwendet werden.) | sudo apt-get install libapache2-mod-php |
| **Verzeichnis wechseln nach html** | cd /var/www/html |
| **Anzeigen des Verzeichnisinhaltes** | ls |
| **Dort befindet sich die Datei „index.html“, von dieser eine Sicherungskopie erstellen mit der Endung .bak** | sudo mv index.html index.bak |

| | |
|---|---|
| **kopiere die index.html auf den Freigabeordner „share2“ um sie von dort aus auf dem Windows-Rechner mittels Editor (Online) zu bearbeiten und wieder zurück in den Ausgangsordner!!** | sudo cp /share2/index.html /var/www/html/ |


**Testen mit einer phpinfo()-Datei**

| | |
|---|---|
| **erstellen einer Datei „phpinfo.php“ im Ordner /var/www/html** | sudo touch phpinfo.php |
| **Öffnen der Datei mit nano und folgenden Eintrag hinzufügen und speichern.** | <?php phpinfo(); ?> |
| **Im Browser folgende Adresse aufrufen und dann sollte so was in der Richtung erscheinen:** | http://pilumex4/phpinfo.php |

**PHP Version 8.2.28**

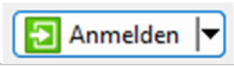
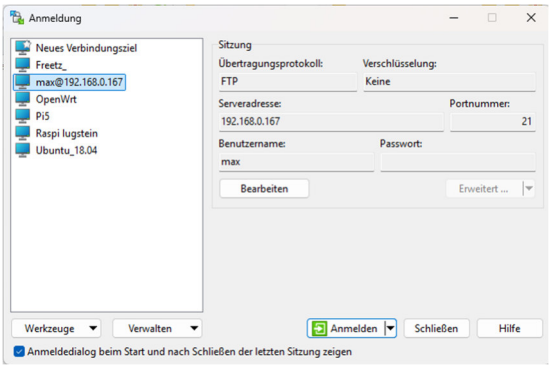
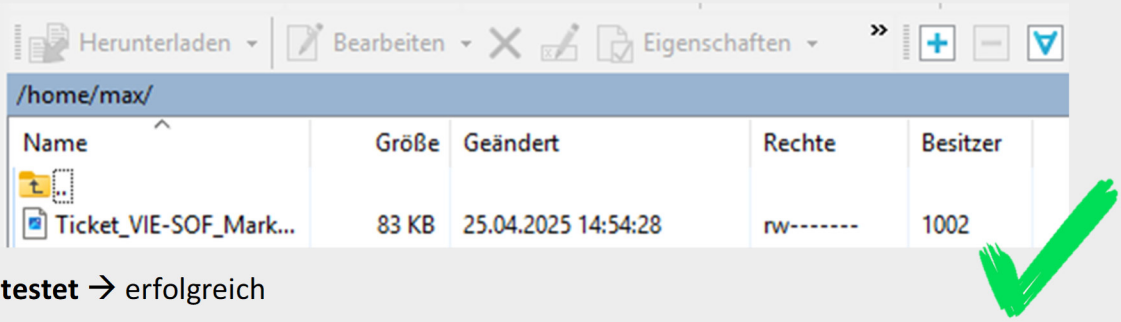
| | |
|---|---|
| System | Linux pilumex4 6.12.25+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.12.25-1+rpt1 (2025-04-30) aarch64 |
| Build Date | Mar 13 2025 18:21:38 |
| Build System | Linux |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/8.2/apache2 |
| Loaded Configuration File | /etc/php/8.2/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/8.2/apache2/conf.d |
| Additional .ini files parsed | /etc/php/8.2/apache2/conf.d/10-opcache.ini, /etc/php/8.2/apache2/conf.d/10-pdo.ini, /etc/php/8.2/apache2/conf.d/20-calendar.ini, /etc/php/8.2/apache2/conf.d/20-ctype.ini, /etc/php/8.2/apache2/conf.d/20-exif.ini, /etc/php/8.2/apache2/conf.d/20-ffi.ini, /etc/php/8.2/apache2/conf.d/20-fileinfo.ini, /etc/php/8.2/apache2/conf.d/20-ftp.ini, /etc/php/8.2/apache2/conf.d/20-gettext.ini, /etc/php/8.2/apache2/conf.d/20-iconv.ini, /etc/php/8.2/apache2/conf.d/20-phar.ini, /etc/php/8.2/apache2/conf.d/20-posix.ini, /etc/php/8.2/apache2/conf.d/20-readline.ini, /etc/php/8.2/apache2/conf.d/20-shmop.ini, /etc/php/8.2/apache2/conf.d/20-sockets.ini, /etc/php/8.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.2/apache2/conf.d/20-sysvsem.ini, /etc/php/8.2/apache2/conf.d/20-sysvshm.ini, /etc/php/8.2/apache2/conf.d/20-tokenizer.ini |
| PHP API | 20220829 |
| PHP Extension | 20220829 |
| Zend Extension | 420220829 |
| Zend Extension Build | API420220829,NTS |
| PHP Extension Build | API20220829,NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | enabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | disabled |
| Zend Max Execution Timers | disabled |
| IPv6 Support | enabled |
| DTrace Support | disabled |
| Registered PHP Streams | https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3 |
| Registered Stream Filters | zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.* |

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.2.28, Copyright (c) Zend Technologies
with Zend OPcache v8.2.28, Copyright (c), by Zend Technologies

# 7. FTP-Server (vsftpd)

Ein FTP-Server ist ein Computer, der für die Übertragung von Dateien über ein Netzwerk über das File Transfer Protocol (FTP) bereitgestellt wird. Er fungiert als ein zentraler Ort, an dem Dateien gespeichert, ausgetauscht und verwaltet werden können, oft im Rahmen von Webhosting oder Dateifreigabe.

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Update & Upgrade durchführen** | sudo apt update \| sudo apt upgrade |
| **Installiert „vsftpd“ (FTP-Server) für UNIX** | sudo apt-get install vsftpd |
| **Verzeichnis wechseln und öffnen der Datei „vsftpd.conf“ mit nano** | sudo nano /etc/vsftpd.conf |
| **folgende Einträge kontrollieren bzw. ändern** | local_enable=YES<br>write_enable=YES |
| | sudo mkdir /home/anna |

| | |
|---|---|
| **Erstellung von home - Verzeichnissen für jeden lokalen Benutzer (waren nicht erstellt!)**<br><br>Besitzer (owner) und die Gruppe (group) des Verzeichnisses ändern.<br><br>Setze die Zugriffsrechte des Verzeichnisses /home/anna auf 755. | sudo chown anna:anna /home/anna<br>sudo chmod 755 /home/anna<br><br>sudo mkdir /home/max<br>sudo chown max:max /home/max<br>sudo chmod 755 /home/max<br><br>sudo mkdir /home/lisa/<br>sudo chown lisa:lisa /home/lisa<br>sudo chmod 755 /home/lisa |
| **Server Neustart durchführen** | sudo systemctl restart vsftpd |
| **Test mit WINSCP (FTP-Client)** |  |
| **Eingabe folgender Daten:**<br><br>Auswahl Übertragungsprotokoll: FTP<br>**Serveradresse:**<br>**Benutzername*:**<br>**Passwort*:**<br><br>***Login-Daten von lokalen Benutzern**<br>  |  |
| **Zugriff getestet** → erfolgreich |  |

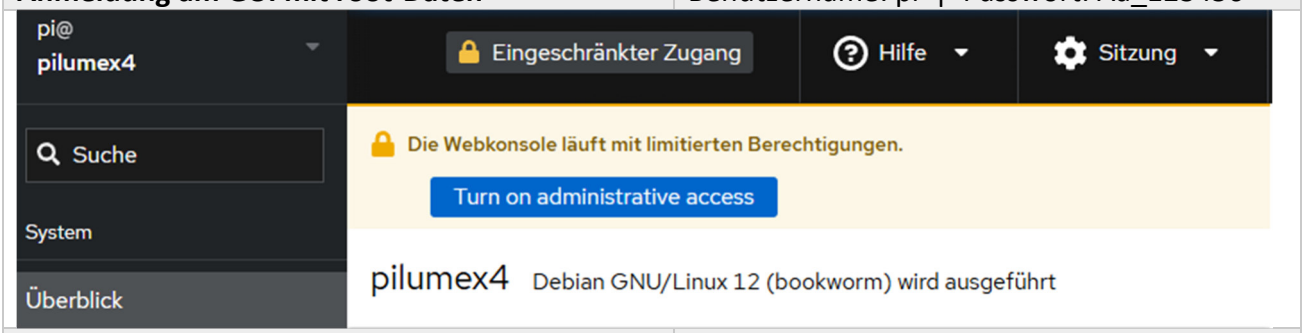
# 8. Cockpit

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Update & Upgrade durchführen** | sudo apt update<br>sudo apt upgrade -y |
| **sudo apt install cockpit -y** | Installiert cockpit |
| **sudo systemctl is-enabled cockpit**<br>**sudo systemctl status cockpit** | Überprüft die Installation von Cockpit |

```
pi@pilumex4:/ $ sudo systemctl status cockpit
o cockpit.service - Cockpit Web Service
     Loaded: loaded (/lib/systemd/system/cockpit.service; static)
     Active: inactive (dead)
TriggeredBy: • cockpit.socket
       Docs: man:cockpit-ws(8)
pi@pilumex4:/ $
```

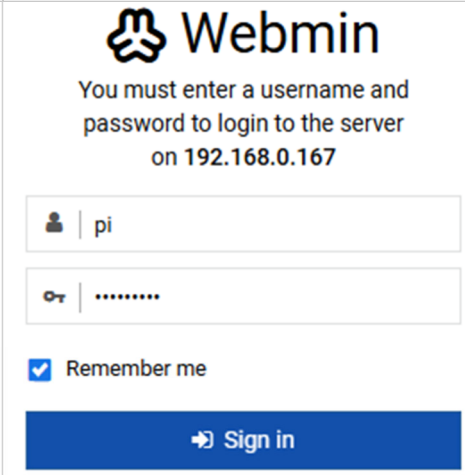
**https://192.168.0.167:9090**

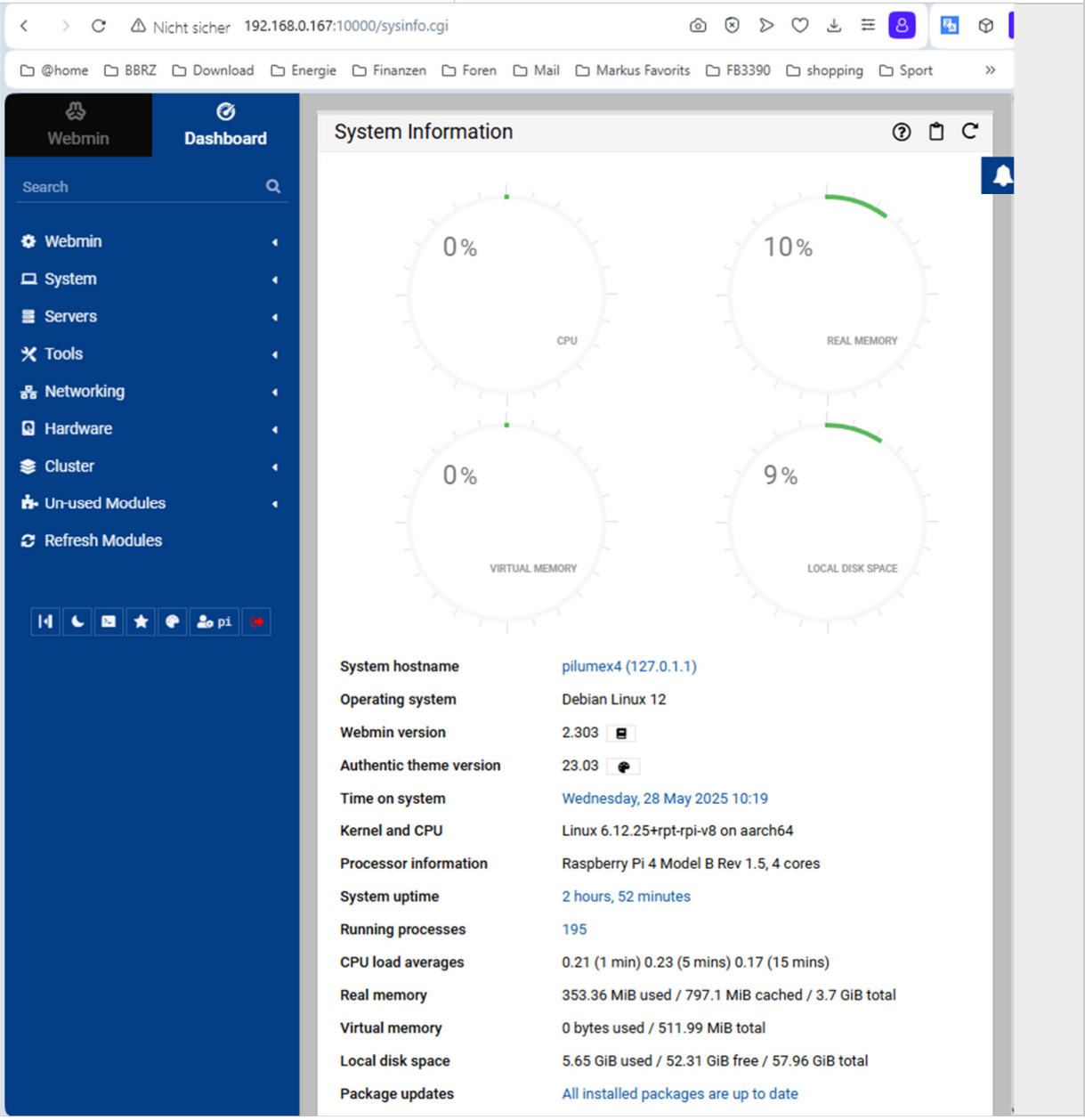

**Anmeldung am GUI mit root-Daten** | Benutzername: pi | Passwort: Aa_123456



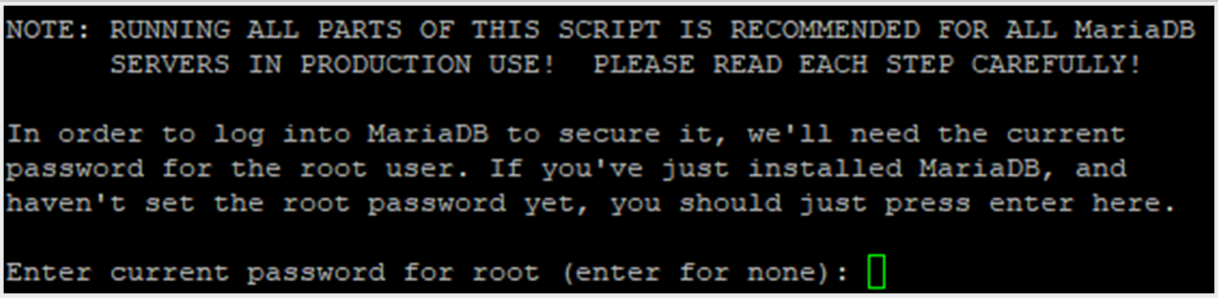
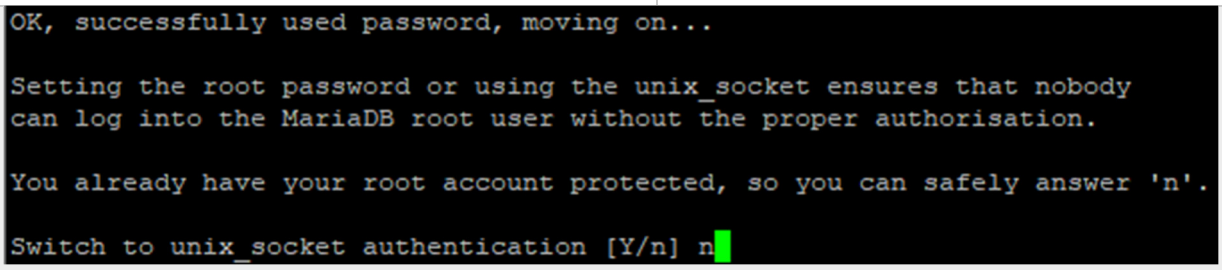

# 9. Webmin

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Systemupdate & Upgrade** | sudo apt-get update \| sudo apt upgrade -y |
| **installiert mehrere wichtige, notwendige Pakete.** | sudo apt install -y wget apt-transport-https software-properties-common |
| **versucht, die Datei webmin-current.deb von der offiziellen Webmin-Website herunterzuladen.** | sudo wget https://www.webmin.com/download/deb/webmin-current.deb |
| **installiert die zuvor heruntergeladene .deb-Datei von Webmin auf dem System.** | sudo dpkg -i webmin-current.deb |
| **Repariert eine fehlerhafte Paketinstallation, z. B. wenn ein** | sudo apt --fix-broken install |

| | |
|---|---|
| **vorheriger dpkg-Vorgang wegen fehlender Abhängigkeiten nicht vollständig abgeschlossen wurde.** | |
| **https://192.168.0.167:10000**<br><br>Benutzer: pi<br>Passwort: Aa_123456 |  |



# 10. MySQL/MariaDB

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Update & Upgrade durchführen** | sudo apt update \| sudo apt upgrade -y |
| **Installation mariaDB-Server** | sudo apt install mariadb-server |
| **Installiert das PHP-Modul php-mysql, das es PHP erlaubt, mit einer MySQL/MariaDB-Datenbank zu kommunizieren.** | sudo apt install php-mysql |
| **Installiert PHP selbst und eine ganze Reihe nützlicher Erweiterungen** | sudo apt install php php-mysql php-curl php-gd php-mbstring php-xml php-xmlrpc php-zip php-soap php-intl unzip -y |
| **Zeigt die aktuell installierte PHP-Version und einige Infos über die PHP-Umgebung.** | php -v |
| **~~Startet die MySQL/MariaDB-Shell als Root-Benutzer.~~** | ~~sudo mysql -u root -p~~<br>~~Passwort: Aa_123456~~ |
| **Führt ein Sicherheits-Setup für MySQL/MariaDB aus.**<br>• **Root-Passwort zu setzen**<br>• **Anonyme Benutzer zu löschen**<br>• **Root-Remote-Login zu deaktivieren**<br>• **Testdatenbank zu entfernen**<br>• **Rechte neu zu laden** | sudo mysql_secure_installation |
| **interaktives Setup startet** | |
|  <br>**Passwort: Aa_123456** | |
| | |
|  <br>**Drücke N, wenn du mit einem Passwort statt per UNIX-User-Login arbeiten möchtest (empfohlen für Fernzugriffe oder Anwendungen wie phpMyAdmin).** | |
| | |

```
Switch to unix_socket authentication [Y/n] n
 ... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
```

**Nein, wir wollen das Passwort beibehalten.**

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n]
```

**Drücke Y, um anonyme Benutzer zu entfernen. Wichtig für Sicherheit.**

```
Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]
```

**Drücke Y, um Root-Zugriff nur lokal zu erlauben (sehr empfohlen).**

```
By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
```

**Drücke Y, um die unsichere Testdatenbank zu löschen.**

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
```

**Drücke Y, damit die Änderungen sofort wirksam werden.**

```
Thanks for using MariaDB!
pi@pilumex4:~ $
```

| | |
|---|---|
| **Prüfen ob alles funktioniert hat mit dem Login** | sudo mysql -u root -p |

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

| | |
|---|---|
| **Erstellen der Datenbank „wordpress_db“** | CREATE DATABASE wordpress_db DEFAULT CHARACTER SET utf8mb4; |
| **Benutzer „wp_user“ erstellen** | CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'Aa_123456'; |
| **Passwort „passuser“ für user „wp_user“ gesetzt -- > mit exit beendet man.** | SET PASSWORD FOR 'wp_user'@'localhost' = PASSWORD('passuser'); |
| **Gewährt dem Benutzer alle verfügbaren Rechte.** | GRANT ALL PRIVILEGES ON *.* TO 'wp_user'@'localhost'; |

| | |
|---|---|
| **Änderungen mit dem Befehl übernehmen!** | FLUSH PRIVILEGES; |

# 11. WordPress-Installation

WordPress ist ein kostenloses Content-Management-System (CMS), mit dem du ganz einfach Webseiten und Blogs erstellen und verwalten kannst.

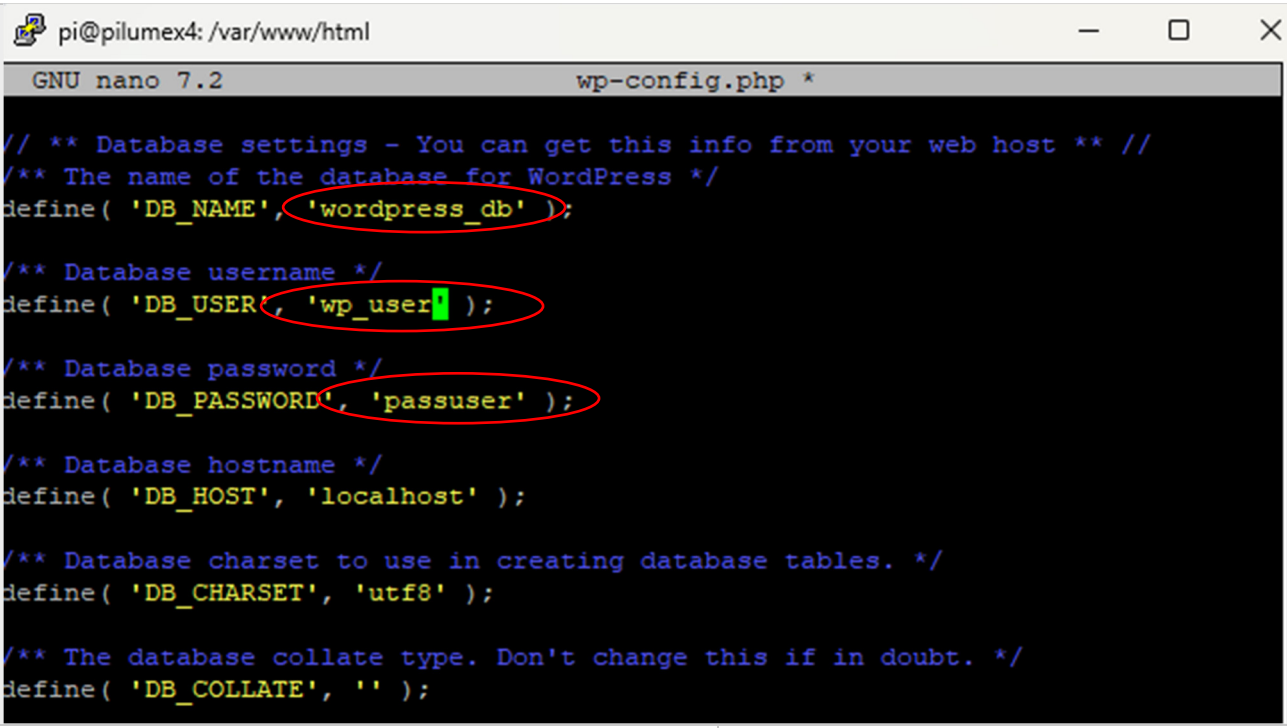
| Erklärung | Syntax / Ergebnis |
|---|---|
| **Systemupdate** | sudo apt update \| sudo apt upgrade |
| **Wechsel in das Webverzeichnis**<br>**Inhalt sollten diese 3 Dateien sein** | cd ..<br>`pi@pilumex4:/var/www/html $ ls`<br>`index.bak  index.html  phpinfo.php`<br>`pi@pilumex4:/var/www/html $` |
| **Neuen Ordner unter /var/www/html erstellen zur Installation von „wordpress“ aber nicht in den Ordner wechseln!** | `pi@markus:/var/www/html $ ls`<br>`index.bak  index.html  phpinfo.php  wordpress`<br>`pi@markus:/var/www/html $` |
| **In diesen Ordner lade ich dann wordpress!** | sudo wget https://wordpress.org/latest.zip |
| **Lädt die aktuellste WordPress-Version als ZIP-Datei herunter** | |
| **Entpackt die WordPress-ZIP-Datei** | sudo unzip latest.zip |
| **Entfernt die leere ZIP-Datei** | sudo rm -r latest.zip |

## 11.1 Berechtigungen für Wordpress setzen

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Setzt den Eigentümer aller Dateien auf www-data (Apache-Nutzer), damit Apache Zugriff hat** | sudo chown -R www-data:www-data /var/www/html |
| **Gibt allen Verzeichnissen die Berechtigung 755 (lesen/ausführen für alle, schreiben nur für Eigentümer)** | sudo find /var/www/html -type d -exec chmod 755 {} \; |
| **Gibt allen Dateien die Berechtigung 644 (lesen für alle, schreiben nur für Eigentümer)** | sudo find /var/www/html -type f -exec chmod 644 {} \; |

## 11.2 wp-config.php einrichten

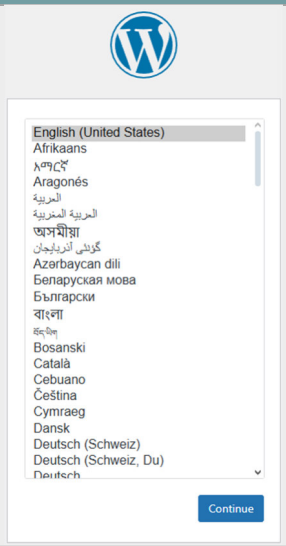
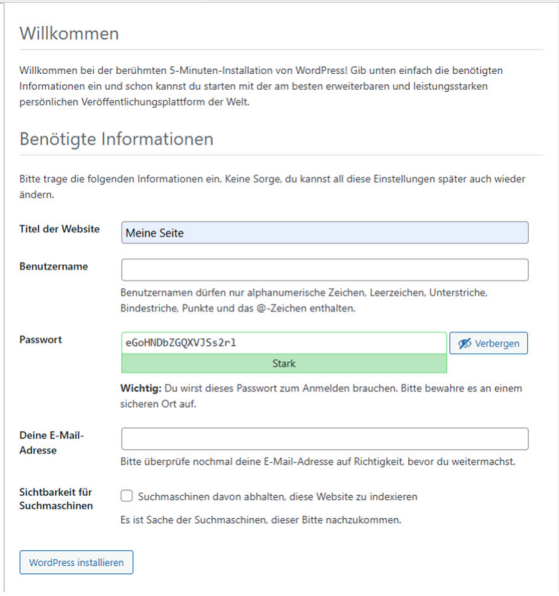
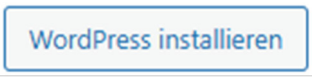
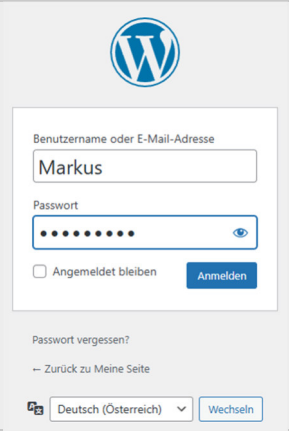
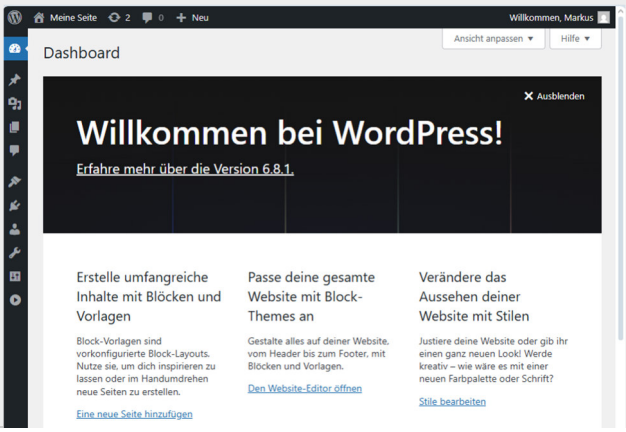
| Erklärung | Syntax / Ergebnis |
|---|---|
| **In den Ordner „/var/www/html/wordpress“ wechseln.** | cd /var/www/html/wordpress |
| **Kopiert die Musterkonfigurationsdatei und benennt sie in wp-config.php um** | sudo cp wp-config-sample.php wp-config.php |
| **Öffnet die Konfigurationsdatei im Terminal-Texteditor Nano, um die Datenbankverbindung einzutragen** | sudo nano wp-config.php |
| **Hier werden die unter Daten die wir unter der „Erstellung der Datenbank“ in mysql/MariaDB benötigt und entsprechend ersetzt!** | |



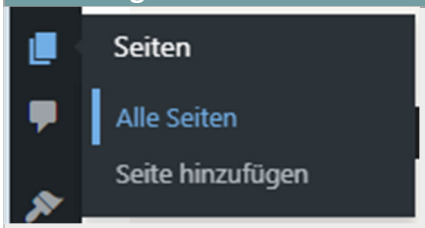
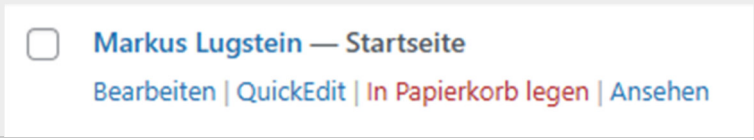
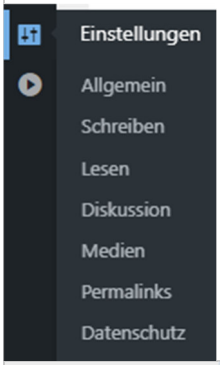
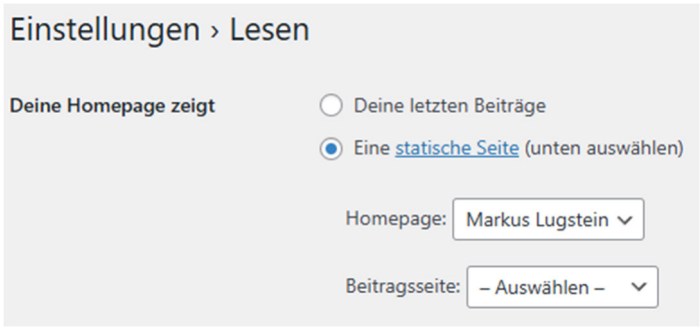
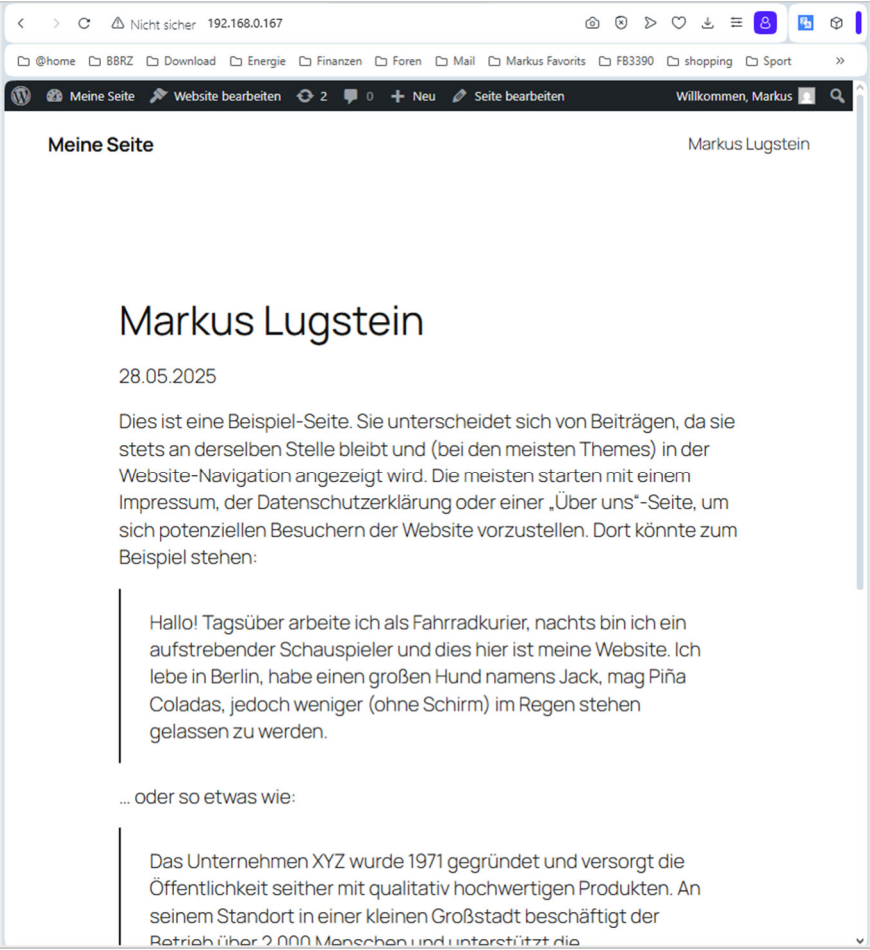

| | |
|---|---|
| **Apache neu starten** | sudo systemctl restart apache2 |

**http://192.168.0.167/wordpress**

# 11.3 WordPress einrichten

| Erklärung | Syntax / Ergebnis |
|---|---|
|  | Nachdem Aufruf der Adresse erscheint der Startbildschirm wo wir unsere Sprache einstellen.<br><br>mit Weiter bestätigen. |
|  | Hier sollst du neue Zugangsdaten erstellen.<br><br>❗ NICHT die MySQL-Zugangsdaten eintragen!<br><br>Meine Seite<br>Benutzer: Markus<br>Passwort: Aa_123456<br>mail: tryitto@gmx.at<br>Haken zur Bestätigung eines schwachen PW<br>kein Haken bei Suchmaschinen<br><br>  |
|   | **Installation erfolgreich durchgeführt!!!** |

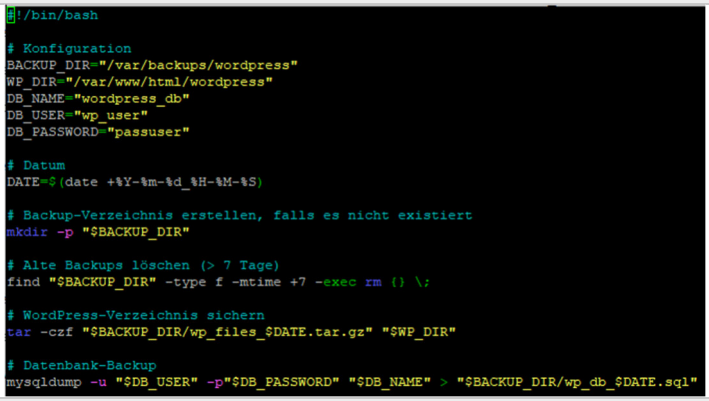
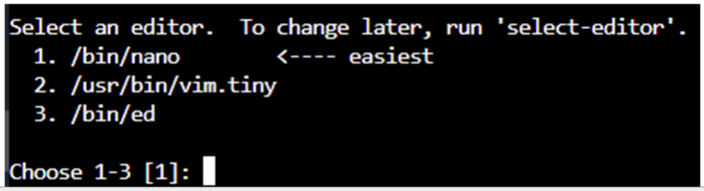
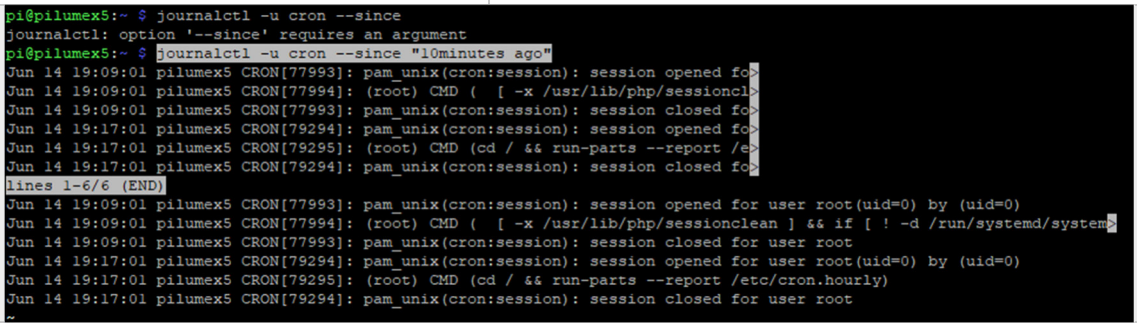
# 11.4 Startseite ändern/einrichten

| Erklärung | Syntax / Ergebnis |
|---|---|
| **unter Seiten „Alle Seiten“ auswählen und anschließend die „Startseite“ bearbeiten und mit eignen Daten ergänzen!!** |  <br>  |
| **anschließend wieder zurück zu den Einstellungen > Lesen und Auswahl wie am Bild treffen und speichern!** → |  <br>  |
| |  |

# 12. Automatisches Backup mit Cronjob

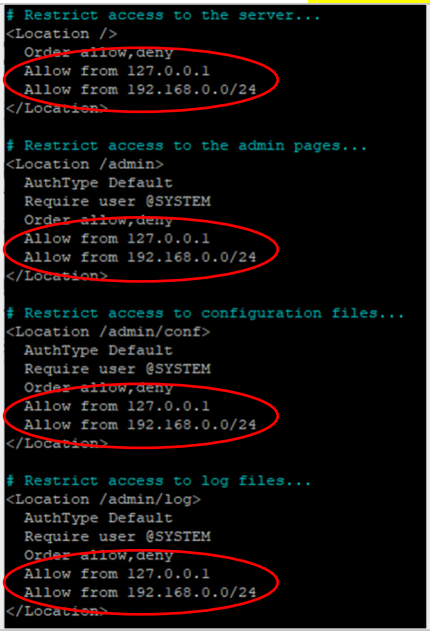
Ein automatisches Backup mit Cronjobs ermöglicht die regelmäßige Sicherung von Daten, ohne dass der Benutzer manuell eingreifen muss. Mit Cronjobs können Sie ein Backup-Skript in bestimmten Zeitintervallen ausführen lassen, z.B. täglich, wöchentlich oder monatlich.

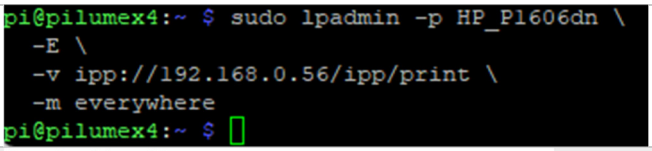
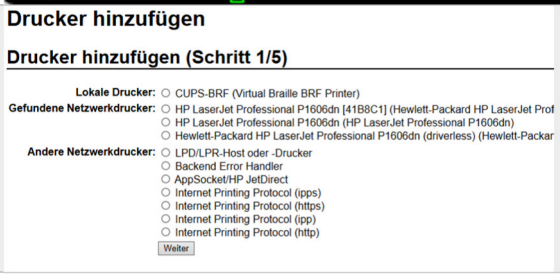
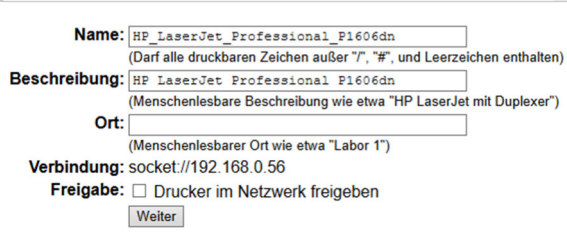
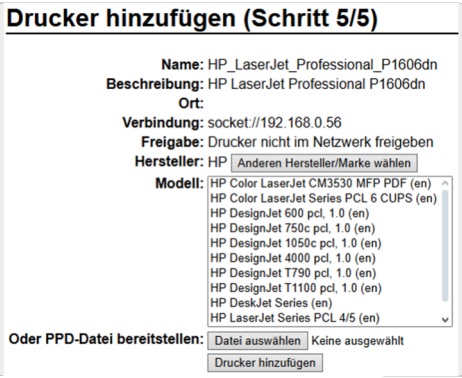
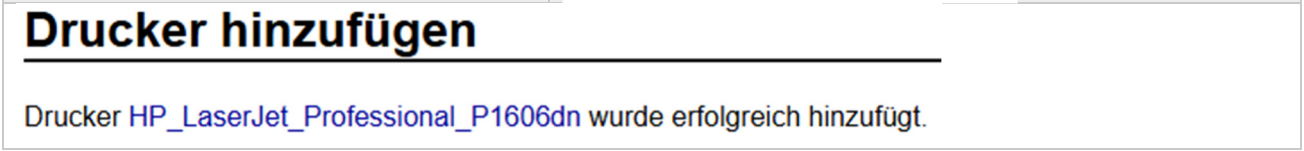
1. Erstellung eines skriptes (

| Erklärung | Syntax / Ergebnis |
|---|---|
| **erstellen der skript Datei „wp_backup.sh“ unter /usr/local/bin** | sudo touch /usr/local/bin/wp_backup.sh |
| **öffnen der Datei mit nano und erstelle diese lt. Bild.** |  |
| **Skript ausführbar machen** | sudo chmod +x /usr/local/bin/wp_backup.sh |
| **Cronjob einrichten**<br><br>**Nr. 1 auswählen** | crontab -e<br>  |
| **ganz unten im Dokument folgendes einfügen** | 0 2 * * * /usr/local/bin/wp_backup.sh |
| **um später zu prüfen ob der Dienst ausgeführt wurde…** | journalctl -u cron --since "10minutes ago" |
|  | |
| **Der Cron-Dienst läuft korrekt und führt geplante Jobs als root aus.**<br><br>**Es wurden keine Fehler gemeldet.**<br><br>**Der Befehl journalctl -u cron --since "10 minutes ago" hat wie gewünscht die letzten Cron-Aktivitäten gezeigt.** | |

# 13. Printserver mit CUPS

CUPS ist eine Druckerverwaltungssoftware, die Druckdienste über das Netzwerk oder lokal bereitstellt.

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Systemupdate durchführen** | sudo apt update \| sudo apt upgrade |
| **Installation** | sudo apt install cups |
| | sudo cp /etc/cups/cupsd.conf /etc/cups/cupsd.conf.original |
| | sudo chmod a-w /etc/cups/cupsd.conf.original |
| | sudo nano /etc/cups/cupsd.conf |
| **CUPS-Dienst starten und aktivieren** | sudo systemctl start cups<br>sudo systemctl enable cups |
| **öffnen der Konfigurationsdatei um Zugriff zu aktivieren und von anderen Rechnern zu erlauben!** | sudo nano /etc/cups/cupsd.conf |
| **Externen Netzwerkzugriff aktivieren** | Listen localhost:631 → ersetzen durch Port 631 |
| **Hier die Einträge entsprechend des verwendeten Netzwerkes eintragen.**<br><br>**In meinem Fall hinzugefügt:**<br>**Allow from 127.0.0.1**<br>**Allow from 192.168.0.0/24**<br><br>**Strg + O Speichern > ENTER > Strg + X schließen!** |  |
| **CUPS-Dienst neu starten** | sudo systemctl restart cups |
| |  |

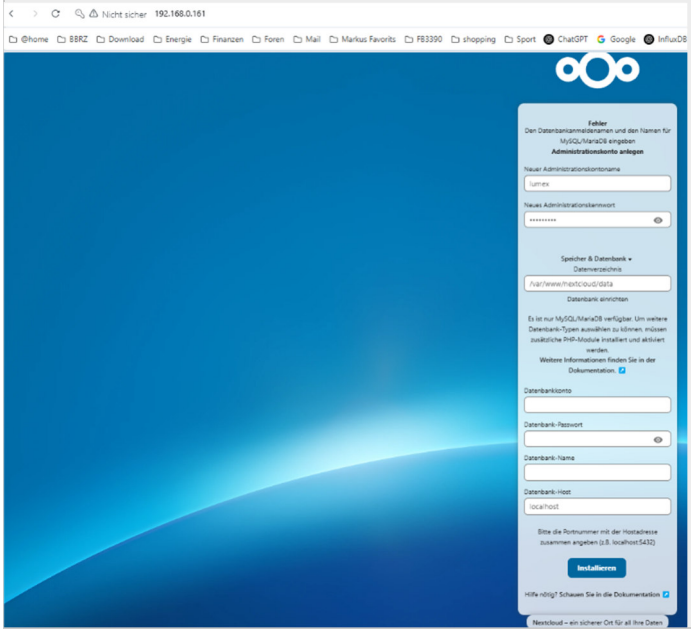
| | |
|---|---|
| **Netzwerkdrucker per Terminal in CUPS einrichten** |  |
| **Netzwerkdrucker per GUI in CUPS einrichten**<br>**Verwaltung > Drucker hinzufügen > sämtliche gefundenen Geräte werden gelistet!**<br>**Auswahl eines Druckers** |  |
| **Diverse Informationen**<br><br>**Drucker im Netzwerk freigeben?** |  |
| **Keine Ahnung!!!** |  |
|  | |

# 14. Eigene Cloud mit Nextcloud (Webinstaller)

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Systemupdate durchführen** | sudo apt update \| sudo apt upgrade |
| **Nextcloud-Archiv herunterladen** | cd /var/www/ |
| | sudo wget https://download.nextcloud.com/server/releases/latest.zip |
| **Archiv entpacken** | sudo unzip latest.zip |
| **Rechte korrekt setzen** | sudo chown -R www-data:www-data nextcloud/<br>sudo chmod -R 755 nextcloud/ |
| **Apache-Konfiguration für Nextcloud erstellen** | sudo nano /etc/apache2/sites-available/nextcloud.conf |
| **Füge diesen Inhalt ein (ersetze ggf. dein-domainname.local oder lass es einfach so für den Start)** | <VirtualHost *:80><br>DocumentRoot /var/www/nextcloud<br>ServerName localhost |

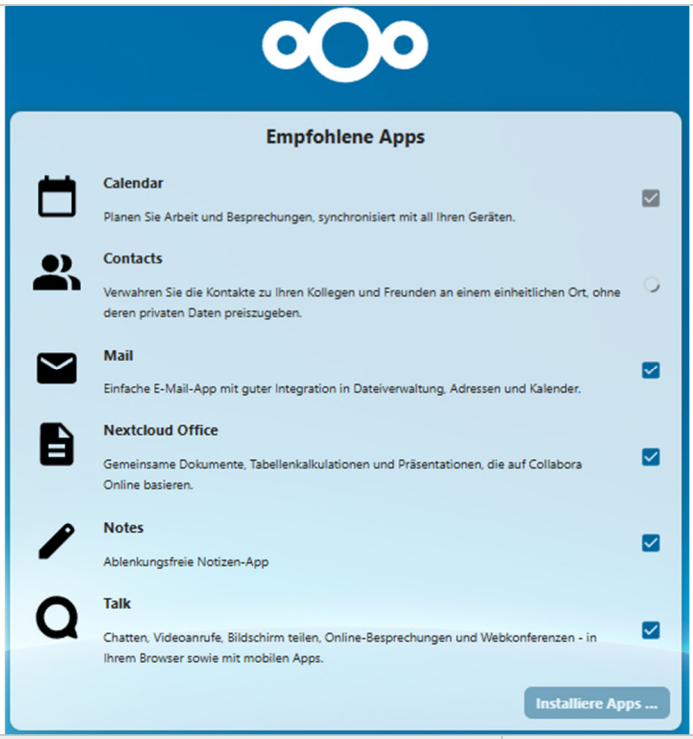
| | |
|---|---|
| | <Directory /var/www/nextcloud><br>Require all granted<br>AllowOverride All<br>Options FollowSymLinks MultiViews<br></Directory><br><br>ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log<br>CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log combined<br></VirtualHost> |
| **Nextcloud-Site aktivieren** | sudo a2ensite nextcloud.conf |
| **Standardseite deaktivieren** | sudo a2dissite 000-default.conf |
| **Apache-Module aktivieren** | sudo a2enmod rewrite headers env dir mime php8.2 |
| **Apache neu starten** | sudo systemctl restart apache2 |
| **Test: PHP-Funktion prüfen** | `echo "<?php phpinfo(); ?>" |
| **PHP-Funktionsseite im Browser prüfen** | http://<Raspberry-IP>/test.php |
| **Testdatei wieder löschen** | sudo rm /var/www/nextcloud/test.php |
| **Notwendige PHP-Module für Nextcloud installieren** | sudo apt install php8.2-{cli,gd,xml,mbstring,zip,curl,bcmath,intl,imagick,mysql} -y |
| **Apache erneut neustarten nach Modulinstallation** | sudo systemctl restart apache2 |
| **Browser: Nextcloud aufrufen** | http://192.168.0.161 |
| | |

**Setup im Browser durchführen**

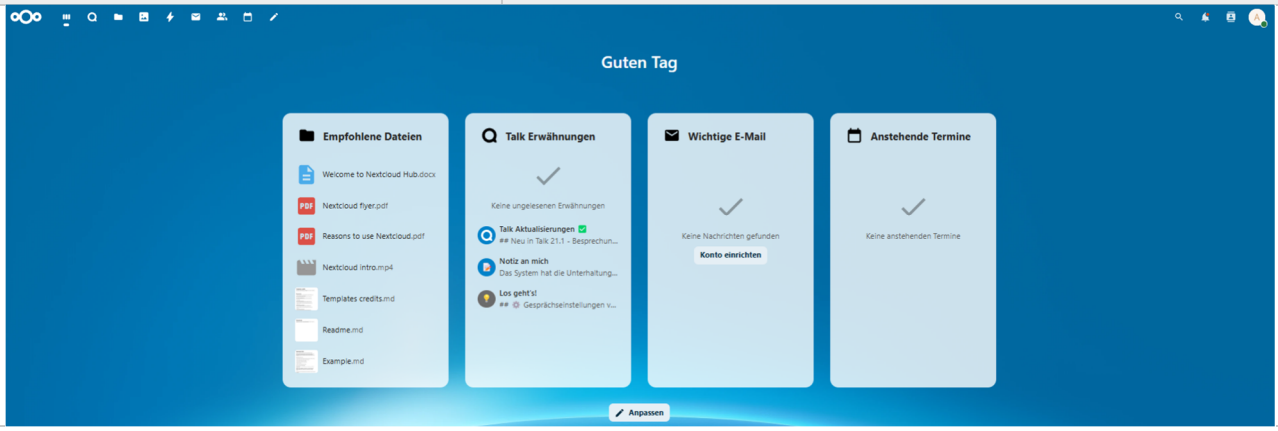



**Administrationskontoname: admin**
**Administrationskennwort: Nextcloud2025!**

**Datenbankkonto: ncuser**
**Datenbank-Passwort: Aa_123456**
**Datenbank-Name: nextcloud**
**Datenbank-Host: localhost**




**Empfohlene Apps alle installiert!**




# 15. Netdata installieren

| Erklärung | Syntax / Ergebnis |
|---|---|
| **System aktualisieren** | sudo apt update \| sudo apt upgrade |
| | bash <(curl -sSL https://my-netdata.io/kickstart.sh) |
| **Anmeldung per Google-Account** | |
| **Sofort nach http://192.168.0.161:19999 aufrufbar.** | |


pilumex5
Total Disk Reads
0
KiB/s
Total Disk Writes
0,17
MiB/s
Avg CPU per Node
2,87
Avg Used RAM per Node
13
Total Network Inbound
115
kbit/s
Total Network Outbound
247,6
kbit/s
Top Nodes by CPU
Top Nodes by Used RAM
Avg System Load (1 min)
0,06
Compute
CPU
Load
Metrics
Metric Correlations
Anomaly Rate %
System
Compute
Memory
Storage
Network
Hardware
Processes
Overview
Apps
CPU
Memory
Swap
Disk
Tasks
FDs
Uptime
Users
Groups
IPC
O/S Services
Synthetic
CPU Idle Jitter
Metadata
Apache
Exim
Web Log

# 16. Firewall mit UFW

| Erklärung | Syntax / Ergebnis |
|---|---|
| **Systemupdate durchführen** | sudo apt update \| sudo apt upgrade |
| **install ufw** | sudo apt install ufw |
| ⚠ **Wichtig bei SSH-Zugriff** | sudo ufw allow ssh |
| **prüfen, was das System für Ports verwendet** | sudo ss -tulnp |
| **je nachdem die entsprechenden Ports erlauben** | |
| **SSH** | sudo ufw allow 22 |
| **HTTP/HTTPS** | sudo ufw allow 80<br>sudo ufw allow 443 |
| **Netdata Dashboard** | sudo ufw allow 19999 |
| **Webmin** | sudo ufw allow 10000 |
| **FTP** | sudo ufw allow 21 |
| **SMB/CIFS** | sudo ufw allow 139<br>sudo ufw allow 445<br>sudo ufw allow proto udp to any port 137<br>sudo ufw allow proto udp to any port 138 |
| **Web-Dienst auf Port 9090** | sudo ufw allow 9090 |
| **Optional: Avahi/mDNS** | sudo ufw allow 5353/udp |
| **UFW aktivieren** | sudo ufw enable |
| **prüfen** | sudo ufw status numbered |

```
Firewall is active and enabled on system startup
pi@pilumex5:~ $
```

CMS = Content Management System (wordpress
DMS - Dokument Management System (paperless

## Ziel der Übung

Sie richten am Raspberry PI einen vollwertigen Server ein, der zentrale Dienste für ein mittelständisches Unternehmen bereitstellt. Dazu gehören Datei- und Webdienste, Benutzerverwaltung, CMS, Druckdienste, Überwachung, eine eigene Cloud-Lösung, Sicherheitsmaßnahmen sowie automatische Backups.

Samba Apache
Cockpit + Webmin
Firewall UFW zum Schluss
cron jobs

### Ausgangssituation

Das Unternehmen besteht aus mehreren Abteilungen (Vertrieb, Entwicklung, Support) und benötigt einen zentralen Server, der folgende Funktionen übernimmt:

- Benutzer- und Gruppenverwaltung
- Dateiablage über das Netzwerk
- Webserver mit PHP und WordPress
- FTP-Zugang für Dateiübertragungen
- Webbasierte Verwaltungstools
- Interner Druckserver
- Datenbankdienste
- Eigene Cloud zur Dateisynchronisation
- Monitoring der Systemressourcen
- Firewall-Schutz und automatisierte Sicherung

### Teil 1: Benutzer- und Gruppenverwaltung

1. Erstellen Sie Gruppen: `vertrieb`, `entwicklung`, `support`
2. Legen Sie Benutzer an:
   a. `anna` (vertrieb) useranna
   b. `max` (entwicklung) usermax
   c. `lisa` (support) userlisa
3. Weisen Sie Benutzer ihren Gruppen zu. ✓
4. Testen Sie Gruppenzugehörigkeit und Login-Funktion.
   ↳ anmelden (nur GUI)

### Teil 2: Samba-Dateifreigaben

1. Installieren und konfigurieren Sie Samba. ✓
2. Erstellen Sie Freigabeordner für jede Abteilung. ✓
3. Erstellen Sie einen zusätzlichen Freigabeordner für den gruppenübergreifenden Zugriff share2

4. Konfigurieren Sie in `/etc/samba/smb.conf` jeweils eine Freigabe mit Gruppenzugriff.
5. Richten Sie Samba-Passwörter für die Benutzer ein.

**Teil 3: Apache mit PHP**

1. Installieren Sie `apache2`, `php`, `libapache2-mod-php`.
2. Testen Sie mit einer `phpinfo()`-Datei.
3. Dokumentieren Sie die obige Seite mit einem Screenshot.

**Teil 4: FTP-Server (vsftpd)**

1. Installieren Sie `vsftpd`.
2. Erlauben Sie lokalen Benutzern Dateiübertragungen.
3. Testen Sie mit einem FTP-Client.
4. Dokumentieren Sie den Zugriff mit einem Screenshot `

**Teil 5: Cockpit**

1. Installieren Sie `cockpit`, aktivieren Sie den Dienst.
2. Dokumentieren Sie den Zugriff mit einem Screenshot `

**Teil 6: Webmin**

1. Fügen Sie das Repository hinzu und installieren Sie `webmin`.
2. Zugriff über: `https://<IP-Adresse>:10000`
3. Dokumentieren Sie den Zugriff mit einem Screenshot `

**Teil 7: MySQL/MariaDB**

1. Installieren Sie den Datenbankserver.
2. Führen Sie `mysql_secure_installation` durch.
3. Erstellen Sie Datenbank `wordpress_db` und Benutzer `wp_user`.

**Teil 8: WordPress-Installation**

1. Laden Sie WordPress herunter und installieren Sie es
2. Ändern Sie die Startseite (Name und Datum) und dokumentieren Sie dies mit einem Screenshot

**Teil 9: Automatisches Backup mit Cronjob**

1. Erstellen Sie ein Skript `/usr/local/bin/wp_backup.sh` für: -nano
    a. Alte Backups (>7 Tage) löschen
    b. WordPress-Verzeichnis sichern
    c. Datenbank-Backup
2. Cronjob:

**Teil 10: Printserver mit CUPS**

1. Installieren Sie `cups`.
2. Zugriff über: `https://<IP-Adresse>:631`
3. Fügen Sie einen Netzwerkdrucker hinzu.

**Teil 11: Firewall mit UFW** Zum

1. Installieren Sie `ufw`, aktivieren Sie nur notwendige Ports:
    a. SSH, HTTP/S, CUPS
3. Aktivieren Sie die Firewall und testen Sie die Regeln.

**Teil 12: Eigene Cloud mit Nextcloud**

3. Installieren Sie Nextcloud (via Webinstaller oder Snap):
4. Testen Sie den Zugriff und dokumentieren Sie diesen mit einem Screenshot

**Teil 13: Monitoring mit Netdata oder Nagios** - Realtime

Installieren Sie eines der beiden Systeme...

Für OBS Studio