

Übungen zur Prüfungsvorbereitung_Teil1

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

Inhalt

1. Zielsetzung	2
2. Geräteübersicht	2
2.1 Netzwerkeinstellungen (Hardware).....	2
2.2 User & Zugangsdaten (Software).....	2
3. Grundinstallation	3
4. Erstinbetriebnahme & Konfiguration OS	4
5. Samba.....	4
6. Samba Benutzer	5
7. Gruppen erstellen	5
8. Apache.....	6
9. Webmin installieren.....	7
10. Cockpit	8
11. MySQL-Server/MariaDB.....	9
12. WordPress.....	10
12.1 Einrichtung.....	11
13. Docker	12
13.1 Installation	12
13.2 Portainer (Docker-UI) installieren	13

1. Zielsetzung

PI:
4 Benutzer
2 Gruppen
Samba mit 3 Shares..
Apache mit neuer Startseite (Name und Datum)
Webmin
Cockpit
mysql-Server
Wordpress
Docker mit Portainer

2. Geräteübersicht

	Device (# & Name)	Rolle	Hypervisor (VM)	Betriebssystem (OS)	IP-Adresse (LAN)
1	Raspberry PI	Server	nein	Ubuntu 24.10	192.168.1.15
2					
3					

2.1 Netzwerkeinstellungen (Hardware)

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

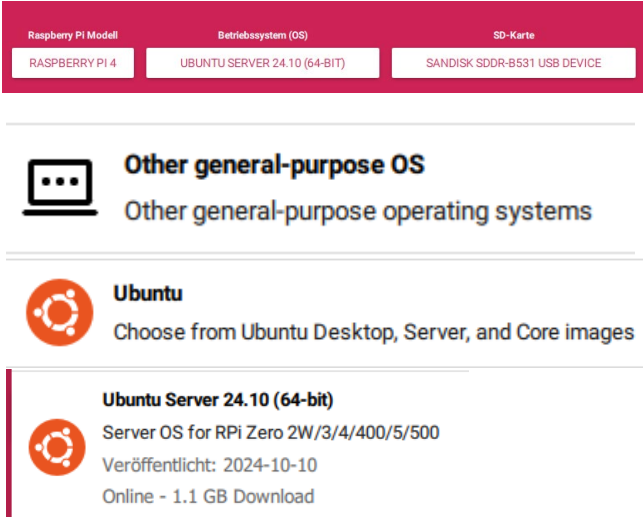
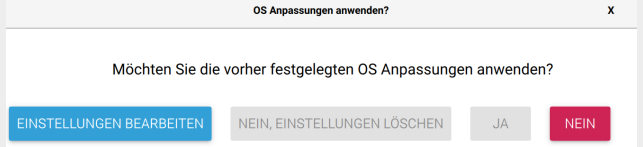
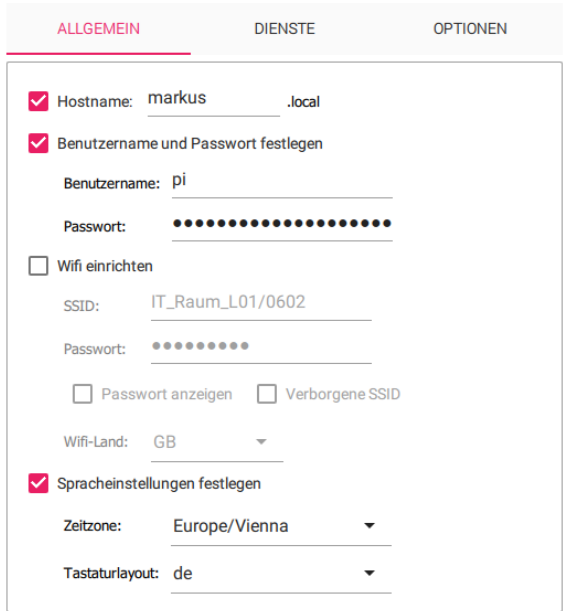
2.2 User & Zugangsdaten (Software)

Client	Geräte #	Anwendung	Benutzername	Passwort	Gerätename	domain (AD)
	PI	OS	pi	Aa_123456		
	Webmin		pi	Aa_123456		
Server	Geräte #	Anwendung	Benutzername	Passwort	Hostname	

3. Grundinstallation

Installation des OS „**Ubuntu Server 24.10 (64-BIT)**“ per Raspberry Pi-Imager.

Alternative: „Raspberry Pi OS Lite (64-bit)“

Erklärung	Beispiel
Auswahl eines geeigneten Betriebssystems	
Einstellungen bearbeiten	
Unter „Allgemein“ Eingabe des Hostnames Festlegen von Benutzernamen & Passwort	

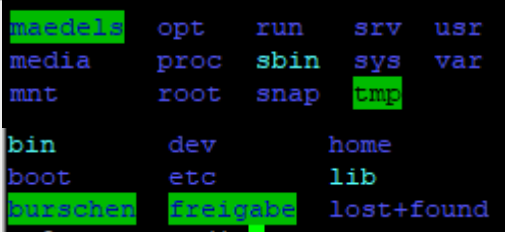
Unter „Dienste“ Aktivierung von SSH	ALLGEMEIN DIENTE ☒ SSH aktivieren ☒ Passwort zur Authentifizierung verwenden ☐ Authentifizierung via Public-Key authorized_keys für 'pi':
Bestätigung des Schreibvorganges mit JA	Alle vorhandenen Daten auf 'SanDisk SDDR-B531 USB Device' werden gelöscht. Möchten Sie wirklich fortfahren? NEIN JA
Wurde der Schreibvorgang beendet, kann die SD-Karte in den Raspi gesteckt und das System gestartet werden.	Raspberry Pi OS (Legacy, 64-bit) Lite wurde auf SanDisk SDDR-B531 USB Device geschrieben Sie können die SD-Karte nun aus dem Lesegerät entfernen WEITER

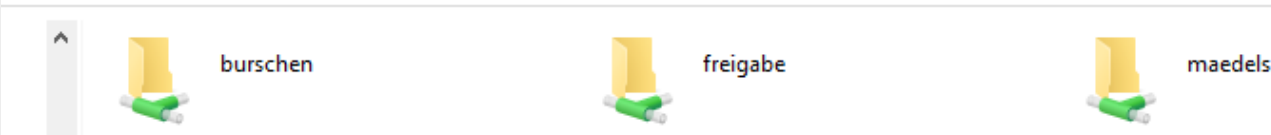
4. Erstinbetriebnahme & Konfiguration OS

Suchen des Gerätes im Lan per IP Scanner oder per Screen im Terminal → 192.168.1.15

Login per SSH (PuTTY) unter der IP-Adresse mit Usernamen: pi | Passwort: Aa_123456

5. Samba

Syntax	Erklärung
sudo apt update	System update & upgrade durchführen
sudo apt upgrade	
sudo apt-get install samba	samba installieren
sudo reboot	Neustart
sudo service smbd restart	Alternative zum Neustart → Serviceneustart!
cd .. cd ..	Wechsel ins root-Verzeichnis
sudo mkdir freigabe sudo mkdir burschen sudo mkdir maedels sudo chmod 777 freigabe sudo chmod 777 burschen sudo chmod 777 maedels	 Ordner erstellt und freigegeben!
sudo nano /etc/samba/smb.conf	samba.conf Datei mit nano öffnen
[freigabe] path = /freigabe	Eintrag in die samba-config- Datei

read only = no browseable = yes create mask = 0777	
[maedels] path = /maedels browseable = yes read only = no guest ok = no valid users = @maedels	Eintrag in die samba-config- Datei für „maedels“
[burschen] path = /burschen browseable = yes read only = no guest ok = no valid users = @burschen	Eintrag in die samba-config- Datei für „burschen“
sudo service smbda restart sudo service nmbd restart	beide Services werden neu gestartet!
192.168.0.179 >	
	

6. Samba Benutzer

Erstellen von 4 Benutzern und für jeden User ein Passwort vergeben.

Syntax	Erklärung
sudo useradd Karl sudo useradd Sepp sudo useradd Mona sudo useradd Maria	Hinzufügen von Benutzern!
sudo smbpasswd -a Karl sudo smbpasswd -a Sepp sudo smbpasswd -a Mona sudo smbpasswd -a Maria	Passwortvergabe für jeden Benutzer
password: Aa_123456 (für jeden User) 4 User mit den Namen Karl, Mona, Sepp, Maria erstellt und Passwort vergeben	

7. Gruppen erstellen

Damit die 4 User je einer Gruppe zugeordnet werden können, werden 2 Gruppen erstellt.

Syntax	Erklärung
--------	-----------

<code>sudo groupadd maedels</code>	Gruppe „maedels“ erstellt.
<code>sudo groupadd burschen</code>	Gruppe „burschen“ erstellt.
<code>sudo usermod -aG burschen Karl</code> <code>sudo usermod -aG burschen Sepp</code>	Beide männlichen User zur Gruppe „burschen“ hinzugefügt.
<code>sudo usermod -aG maedels Mona</code> <code>sudo usermod -aG maedels Maria</code>	Beide weiblichen User zur Gruppe „maedels“ hinzugefügt.
<code>sudo nano /etc/samba/smb.conf</code>	Erweitern der Einträge in der config-Datei
<code>[maedels]</code> <code>path = /maedels</code> <code>browseable = yes</code> <code>read only = no</code> <code>guest ok = no</code> <code>valid users = @maedels</code>	Eintrag für Gruppe „maedels“
<code>[burschen]</code> <code>path = /burschen</code> <code>browseable = yes</code> <code>read only = no</code> <code>guest ok = no</code> <code>valid users = @burschen</code>	Eintrag für Gruppe „burschen“

8. Apache

Installation und Modifikation eines Web-Servers

Syntax	Erklärung
<code>sudo apt update</code>	Systemupdate durchführen
<code>sudo apt install apache2</code>	Installieren von Apache2
<code>sudo systemctl start apache2</code>	
<code>cd /var/www/html</code>	Wechselt in das Verzeichnis „html“
<code>sudo mv index.html index.bak</code>	umbenennen der Datei
Datei „index.html“ bearbeiten und unter „freigabe“ abspeichern	
<code>sudo cp /freigabe/index.html /var/www/html/</code>	kopiert die Datei index.html nach html

< > ↻ ⚠ Nicht sicher 192.168.0.179

📁 @home 📁 BBRZ 📁 Download 📁 Energie 📁 Finanzen 📁 Foren 📁

Mein Name ist Markus Lugstein

Heute ist Samstag der 12. April 2025

9. Webmin installieren

Syntax	Erklärung
<code>sudo apt-get update</code> <code>sudo apt upgrade</code>	Systemupdates durchführen
<code>sudo apt install -y wget apt-transport-https</code> <code>software-properties-common</code>	Abhängigkeiten sicherstellen
<code>sudo wget</code> <code>https://www.webmin.com/download/deb/webmin-current.deb</code>	Lade das aktuelle Webmin .deb-Paket direkt herunter
<code>sudo dpkg -i webmin-current.deb</code>	Paket installieren
<code>sudo apt --fix-broken install</code>	Falls es Paket-Abhängigkeitsfehler gibt, behebe sie

https://192.168.1.15:10000

Erstanmeldung

Username: pi
Password: Aa_123456



Webmin

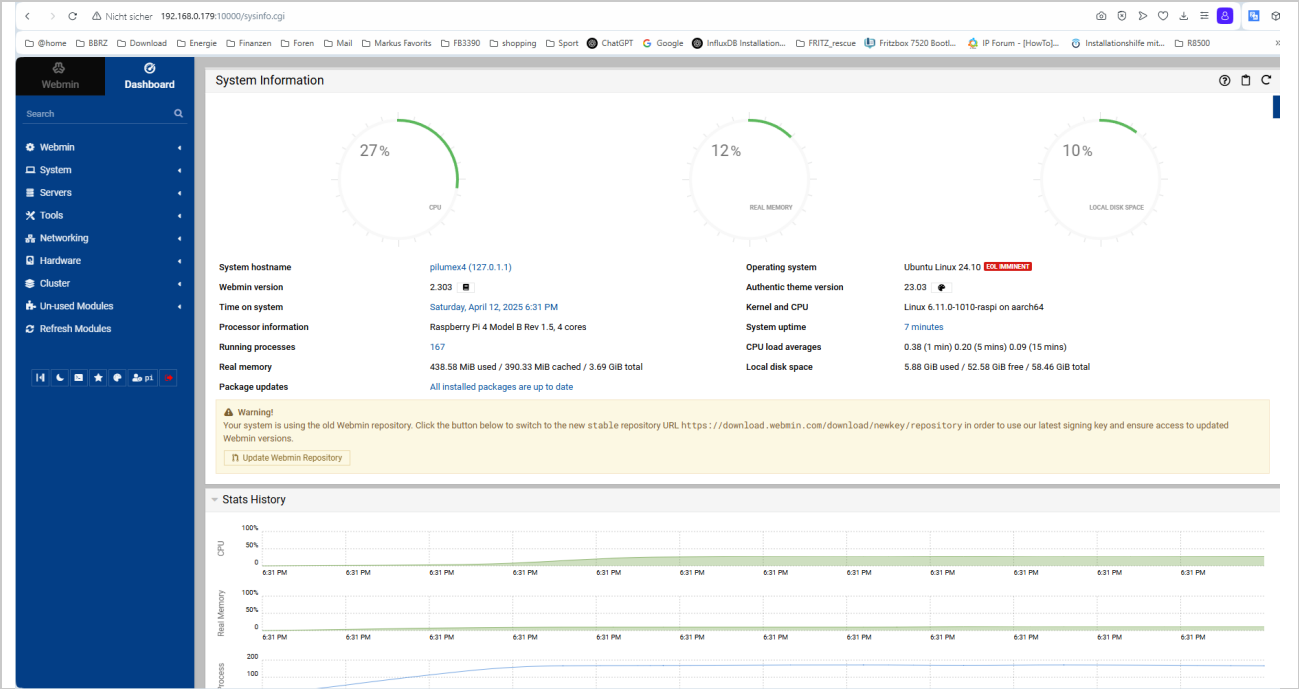
You must enter a username and password to login to the server on 192.168.0.155

 Username

 Password

☐ Remember me

 Sign in



10. Cockpit

Cockpit ist ein webbasiertes Linux-Administrationstool, mit dem du mehrere Linux-Server über ein einziges Dashboard verwalten kannst.

sudo apt update	
sudo apt install cockpit -y	Installiert cockpit
sudo systemctl start --now cockpit	startet und aktiviert den Cockpit-Dienst
sudo systemctl is-enabled cockpit	
sudo systemctl status cockpit	Überprüft die Installation von Cockpit

```
pi@pilumex4:~$ sudo systemctl start --now cockpit
pi@pilumex4:~$ sudo systemctl is-enabled cockpit
static
pi@pilumex4:~$ sudo systemctl status cockpit
● cockpit.service - Cockpit Web Service
   Loaded: loaded (/lib/systemd/system/cockpit.service; static)
   Active: active (running) since Fri 2025-04-11 17:56:00 BST; 1min 11s ago
   TriggeredBy: ● cockpit.socket
     Docs: man:cockpit-ws(8)
    Process: 23093 ExecStartPre=/usr/lib/cockpit/cockpit-certificate-ensure --f
   Main PID: 23109 (cockpit-tls)
      Tasks: 1 (limit: 3916)
         CPU: 1.371s
    CGroup: /system.slice/cockpit.service
            └─23109 /usr/lib/cockpit/cockpit-tls
```

https://192.168.1.15:9090

Ubuntu 24.10

Benutzername

Passwort

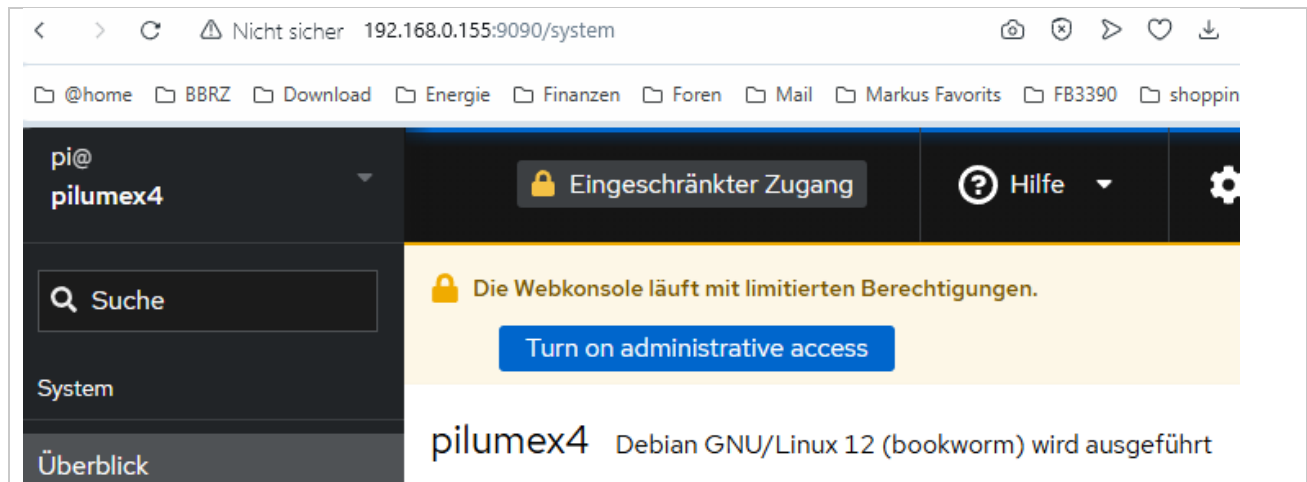
► Andere Einstellungen

Anmelden

Server: pilumex4

Melden Sie sich mit dem Server-Benutzerkonto an.





11. MySQL-Server/MariaDB

Syntax	Beschreibung
<code>sudo apt update</code>	
<code>sudo apt upgrade</code>	
<code>sudo apt install mariadb-server</code>	Installiert den MariaDB-Datenbankserver.
<code>sudo apt install php-mysql</code>	Installiert das PHP-MySQL-Modul, das die Verbindung zwischen PHP und MariaDB ermöglicht.
<code>sudo apt install php php-mysql php-curl php-gd php-mbstring php-xml php-xmlrpc php-zip php-soap php-intl unzip -y</code>	Installiert PHP & benötigte Module
<code>php -v</code>	Prüft, ob PHP funktioniert.
<code>sudo mysql -u root -p</code>	Meldet sich als root-Benutzer am MariaDB-Server an. Es wird ein Passwort abgefragt → Aa_123456
<pre> pi@pilumex4:~ \$ sudo mysql -u root -p Enter password: Welcome to the MariaDB monitor. Commands end with ; or \g. Your MariaDB connection id is 31 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)]> </pre>	
<code>CREATE USER 'lumex'@'localhost' IDENTIFIED BY '123456';</code>	Neuen Benutzer „lumex“ am localhost anlegen
<code>PASSWORD [FOR lumex] = 123456;</code>	Passwort für User „lumex“ gesetzt = 123456
<code>GRANT ALL PRIVILEGES ON *.* TO 'lumex'@'localhost';</code>	Gewährt dem Benutzer alle verfügbaren Rechte.
<code>FLUSH PRIVILEGES;</code>	Änderungen mit dem Befehl übernehmen!

12. WordPress

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

Syntax	Beschreibung
<code>sudo apt update</code> <code>sudo apt upgrade</code>	Systemupdate
<code>cd /var/www/html</code>	Wechselt in das Webverzeichnis, in dem Apache Webseiten bereitstellt
<code>sudo rm index.html</code>	Löscht die Standard-Apache-Startseite (die leere index.html)
<code>sudo wget https://wordpress.org/latest.zip</code>	Lädt die aktuellste WordPress-Version als ZIP-Datei herunter
<code>sudo unzip latest.zip</code>	Entpackt die WordPress-ZIP-Datei
<code>sudo mv wordpress/* .</code>	Verschiebt alle entpackten WordPress-Dateien in das aktuelle Verzeichnis (<code>/var/www/html</code>)
<code>sudo rm -r wordpress latest.zip</code>	Entfernt das leere „wordpress“-Verzeichnis und die ZIP-Datei
Berechtigungen setzen	
<code>sudo chown -R www-data:www-data /var/www/html</code>	Setzt den Eigentümer aller Dateien auf <code>www-data</code> (Apache-Nutzer), damit Apache Zugriff hat
<code>sudo find /var/www/html -type d -exec chmod 755 {} \;</code>	Gibt allen Verzeichnissen die Berechtigung <code>755</code> (lesen/ausführen für alle, schreiben nur für Eigentümer)
<code>sudo find /var/www/html -type f -exec chmod 644 {} \;</code>	Gibt allen Dateien die Berechtigung <code>644</code> (lesen für alle, schreiben nur für Eigentümer)
wp-config.php einrichten	
<code>cd /var/www/html</code>	Wechselt wieder ins Webverzeichnis
<code>sudo cp wp-config-sample.php wp-config.php</code>	Kopiert die Musterkonfigurationsdatei und benennt sie in <code>wp-config.php</code> um
<code>sudo nano wp-config.php</code>	Öffnet die Konfigurationsdatei im Terminal-Texteditor Nano , um die Datenbankverbindung einzutragen

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'lumex' );

/** Database password */
define( 'DB_PASSWORD', '123456' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

12.1 Einrichtung

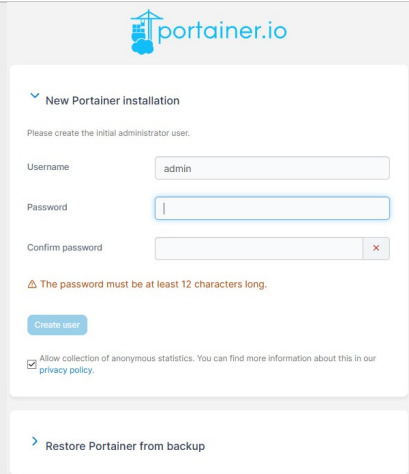
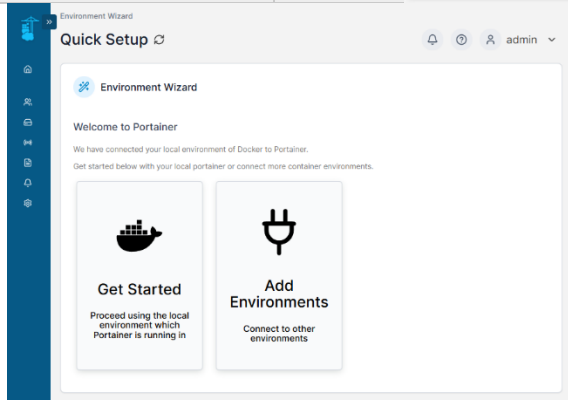
Auswahl der Sprache samt Region →		Beschreibung	
Div. Eingaben und abschließen der Installation.... Willkommen Willkommen bei der berühmten 5-Minuten-Installation von WordPress! Gib unten einfach die benötigten Informationen ein und schon kannst du starten mit der am besten erweiterbaren und leistungsstarken persönlichen Veröffentlichungsplattform der Welt. Benötigte Informationen Bitte trage die folgenden Informationen ein. Keine Sorge, du kannst all diese Einstellungen später auch wieder ändern. Titel der Website Meine Seite Benutzername lumex Benutzernamen dürfen nur alphanumerische Zeichen, Leerzeichen, Unterstriche, Bindestriche, Punkte und das @-Zeichen enthalten. Passwort 123456 Ganz schwach Verbergen Passwort bestätigen ☒ Bestätige die Verwendung eines schwachen Passworts Deine E-Mail-Adresse Bitte überprüfe nochmal deine E-Mail-Adresse auf Richtigkeit, bevor du weitermachst. Sichtbarkeit für Suchmaschinen ☐ Suchmaschinen davon abhalten, diese Website zu indizieren Es ist Sache der Suchmaschinen, dieser Bitte nachzukommen. WordPress installieren Deutsch (Schweiz) Deutsch Deutsch (Sie) Deutsch (Schweiz, Du) Deutsch (Österreich)		WordPress Installation erfolgreich! WordPress wurde installiert. Vielen Dank, und nun viel Spaß! Benutzername lumex Passwort Das von dir gewählte Passwort. Anmelden WordPress Benutzername oder E-Mail-Adresse lumex Passwort ☐ Angemeldet bleiben Anmelden Passwort vergessen? → Zurück zu Meine Seite Deutsch (Österreich) Wechseln Ansicht anpassen Hilfe	
Dashboard			
Willkommen bei WordPress! Erfahre mehr über die Version 6.7.2.			
Erfolgreiche Installation!!!!			

13. Docker

13.1 Installation

Syntax	Beschreibung
<code>sudo apt update</code>	System vorbereiten & alte Docker-Versionen entfernen (falls vorhanden)
<code>sudo apt remove docker docker-engine docker.io containerd runc</code>	
<code>sudo apt install apt-transport-https ca-certificates curl software-properties-common -y</code>	notwendige Pakete installieren
<code>curl -fsSL https://download.docker.com/linux/ubuntu/gpg sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/docker.gpg</code>	Docker GPG-Schlüssel hinzufügen
<code>echo "deb [arch=\$(dpkg --print-architecture) signed-by=/etc/apt/trusted.gpg.d/docker.gpg] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable" \nsudo tee /etc/apt/sources.list.d/docker.list > /dev/null</code>	Docker-Repository hinzufügen
<code>sudo apt update</code> <code>sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y</code>	Docker installieren
<code>sudo systemctl enable docker</code> <code>sudo systemctl start docker</code>	Docker aktivieren und starten
<code>sudo docker run hello-world</code>	Test: Läuft Docker?
<pre>Hello from Docker! This message shows that your installation appears to be working correctly.</pre>	
Optional: Docker ohne sudo nutzen	
<code>sudo usermod -aG docker \$USER</code>	Danach: Ab- und wieder anmelden, damit die Gruppenänderung aktiv wird.

13.2 Portainer (Docker-UI) installieren

Syntax	Beschreibung																				
<code>docker volume create portainer_data</code>	Portainer-Datenvolumen erstellen																				
Portainer-Container starten																					
<pre>docker run -d \ --name portainer \ --restart=always \ -p 9000:9000 \ -p 9443:9443 \ -v /var/run/docker.sock:/var/run/docker.sock \ -v portainer_data:/data \ portainer/portainer-ce:latest</pre>	Erklärung in Tabellenform <table> <tr> <th>Teil</th><th>Funktion / Bedeutung</th></tr> <tr> <td><code>docker run</code></td><td>Startet einen neuen Container</td></tr> <tr> <td><code>-d</code></td><td>„Detached Mode“: Container läuft im Hintergrund</td></tr> <tr> <td><code>--name portainer</code></td><td>Gibt dem Container den Namen <code>portainer</code> (praktisch für Verwaltung)</td></tr> <tr> <td><code>--restart=always</code></td><td>Startet den Container automatisch neu, z. B. nach Reboot</td></tr> <tr> <td><code>-p 9000:9000</code></td><td>Verbindet Port 9000 im Container mit Port 9000 auf dem Host (HTTP-UI)</td></tr> <tr> <td><code>-p 9443:9443</code></td><td>Verbindet Port 9443 im Container mit Port 9443 auf dem Host (HTTPS-UI)</td></tr> <tr> <td><code>-v /var/run/docker.sock:/var/run/docker.sock</code></td><td>Mountet den Docker-Socket → Portainer kann den Docker-Dienst auf dem Host steuern</td></tr> <tr> <td><code>-v portainer_data:/data</code></td><td>Bindet ein benanntes Volume ein → Speichert Konfigurationen dauerhaft</td></tr> <tr> <td><code>portainer/portainer-ce:latest</code></td><td>Nutzt das neueste Image von Portainer Community Edition (CE) von Docker Hub</td></tr> </table>	Teil	Funktion / Bedeutung	<code>docker run</code>	Startet einen neuen Container	<code>-d</code>	„Detached Mode“: Container läuft im Hintergrund	<code>--name portainer</code>	Gibt dem Container den Namen <code>portainer</code> (praktisch für Verwaltung)	<code>--restart=always</code>	Startet den Container automatisch neu , z. B. nach Reboot	<code>-p 9000:9000</code>	Verbindet Port 9000 im Container mit Port 9000 auf dem Host (HTTP-UI)	<code>-p 9443:9443</code>	Verbindet Port 9443 im Container mit Port 9443 auf dem Host (HTTPS-UI)	<code>-v /var/run/docker.sock:/var/run/docker.sock</code>	Mountet den Docker-Socket → Portainer kann den Docker-Dienst auf dem Host steuern	<code>-v portainer_data:/data</code>	Bindet ein benanntes Volume ein → Speichert Konfigurationen dauerhaft	<code>portainer/portainer-ce:latest</code>	Nutzt das neueste Image von Portainer Community Edition (CE) von Docker Hub
Teil	Funktion / Bedeutung																				
<code>docker run</code>	Startet einen neuen Container																				
<code>-d</code>	„Detached Mode“: Container läuft im Hintergrund																				
<code>--name portainer</code>	Gibt dem Container den Namen <code>portainer</code> (praktisch für Verwaltung)																				
<code>--restart=always</code>	Startet den Container automatisch neu , z. B. nach Reboot																				
<code>-p 9000:9000</code>	Verbindet Port 9000 im Container mit Port 9000 auf dem Host (HTTP-UI)																				
<code>-p 9443:9443</code>	Verbindet Port 9443 im Container mit Port 9443 auf dem Host (HTTPS-UI)																				
<code>-v /var/run/docker.sock:/var/run/docker.sock</code>	Mountet den Docker-Socket → Portainer kann den Docker-Dienst auf dem Host steuern																				
<code>-v portainer_data:/data</code>	Bindet ein benanntes Volume ein → Speichert Konfigurationen dauerhaft																				
<code>portainer/portainer-ce:latest</code>	Nutzt das neueste Image von Portainer Community Edition (CE) von Docker Hub																				
http://192.168.0.179:9000																					
Username: admin Password: Aa_123456789																					
 Installation erfolgreich!																					

DietPi Server Setup mit Docker, WordPress & Portainer

Version: 2025

Plattform: DietPi OS (anstelle von Ubuntu 24.10)

Ziel: LAMP + Docker + WordPress + Portainer auf einem Raspberry Pi

1. Zielsetzung

- Einrichten eines leichtgewichtigen Servers mit DietPi OS
- Installation von:
 - MariaDB
 - Apache2
 - PHP
 - WordPress
 - Docker
 - Portainer
 - Webmin (optional)

2. Voraussetzungen

- Raspberry Pi mit DietPi OS (aktuelles Image von [dietpi.com])
- Netzwerkzugang (per DHCP oder statischer IP)
- SSH-Zugriff aktiviert (Standard bei DietPi)
- Benutzer: root oder dietpi

3. Erste Schritte nach der Installation

Login per SSH:

```
ssh root@<IP-ADRESSE>
```

System aktualisieren:

```
dietpi-update
```

```
apt update && apt upgrade -y
```

DietPi Software Center starten:

```
dietpi-software
```

Im Software-Center folgende Pakete installieren:

- Apache2
- MariaDB Server
- PHP
- PHP-Module:
 - php-mysql
 - php-gd
 - php-curl
 - php-xml
 - php-mbstring
 - php-zip
 - php-intl

4. WordPress installieren

```
cd /var/www
```

```
rm index.html
```

```
wget https://wordpress.org/latest.zip
```

```
apt install unzip -y
```

```
unzip latest.zip
```

```
mv wordpress/* html/
```

```
rm -r wordpress latest.zip
```

```
chown -R www-data:www-data /var/www/html
```

```
find /var/www/html -type d -exec chmod 755 {} \;
```

```
find /var/www/html -type f -exec chmod 644 {} \;
```

Datenbank für WordPress anlegen:

```
mariadb -u root -p
```

SQL:

```
CREATE DATABASE wordpress;
```

```
CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'sicherespasswort';
```

```
GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

WordPress konfigurieren:

```
cd /var/www/html
```

```
cp wp-config-sample.php wp-config.php
```

```
nano wp-config.php
```

Wichtige Einträge ändern:

```
define( 'DB_NAME', 'wordpress' );
```

```
define( 'DB_USER', 'wpuser' );
```

```
define( 'DB_PASSWORD', 'sicherespassword' );
```

```
define( 'DB_HOST', 'localhost' );
```

Apache neu starten:

```
systemctl restart apache2
```

5. Docker auf DietPi installieren

```
dietpi-software install 162 # Docker
```

Docker Test:

```
docker run hello-world
```

Optional: Docker ohne sudo nutzen:

```
usermod -aG docker dietpi
```

Dann abmelden und wieder anmelden, damit die Gruppenänderung wirksam wird.

6. Portainer installieren

Daten-Volume erstellen:

```
docker volume create portainer_data
```

Container starten:

```
docker run -d --name portainer --restart=always -p 9000:9000 -p 9443:9443 -v  
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
```

Portainer WebUI aufrufen:

http://<RASPBERRY-IP>:9000

7. Optional: Webmin installieren

```
apt install software-properties-common apt-transport-https wget -y
```

```
wget -q http://www.webmin.com/jcameron-key.asc -O- | apt-key add -
```

```
echo "deb http://download.webmin.com/download/repository sarge contrib" >> /etc/apt/sources.list
```

```
apt update
```

```
apt install webmin -y
```

Webmin im Browser aufrufen:

https://<RASPBERRY-IP>:10000

8. Abschluss

- WordPress aufrufen: http://<RASPBERRY-IP>

- Portainer aufrufen: http://<RASPBERRY-IP>:9000

- Webmin nutzen: https://<RASPBERRY-IP>:10000

Server ist jetzt komplett eingerichtet und einsatzbereit.