

Proxmox auf dem Raspberry Pi 5 installieren

Dieses Dokument zeigt eine Mitschrift zur Ausarbeitung eines Arbeitsauftrage/Laborauftrages im Unterrichtsfach **Betriebssysteme & Labor** (Hr. Klaus Pesendorfer).

Diese Aufgabenstellung war zusätzlich zu erledigen.

Inhalt

1. Zielsetzung	2
2. Geräteübersicht	2
2.1 Netzwerkeinstellungen (Hardware)	2
2.2 User & Zugangsdaten (Software)	2
3.-Erkenntnis vorweg	3
4. Grundinstallation des Hostsystems	4
5. Vorbereitung & Grundinstallation für Proxmox	5
6. Installation von Proxmox (PXVIRT)	6
7. USB-Festplatte anschließen und Partitionen einrichten	7
8. PXVIRT-Speicher konfigurieren	8
9. Debian 13 (Trixie) Container erstellen	9
10. GUI (Desktop) & Remote Desktop (RDP)	10
11. Fehlerbehebung & bekannte Probleme	11

1. Zielsetzung

Ziel dieser Dokumentation ist es, auf einem Raspberry Pi 5 eine funktionierende Proxmox (PXVIRT)-Umgebung zu installieren, eine externe USB-Festplatte als Speicher einzubinden und darauf einen Debian 13 (Trixie) Container mit Desktop-Umgebung bereitzustellen.

2. Geräteübersicht

	Device (# & Name)	Rolle	Hyper visor	Betriebssystem (OS)	IP-Adresse (LAN)
1	Raspberry Pi 5 (pilumex5)	Server / Proxmox Host	nein	Raspberry Pi OS Lite (64-bit) mit PXVirt (Proxmox VE 8.x)	10.2.205.160
2	Debian Trixie (LXC 102)	Virtueller Container (Client)	Ja (LXC)	Debian 13 "Trixie" (ARM64)	10.2.205.180
3	Lenovo E16	Administrations- PC / Client	Nein	Windows 11 Pro	10.2.205.21

2.1 Netzwerkeinstellungen (Hardware)

Devices	Device #	# 1	# 2	# 3	# 4	# 5
	IP Adr.	10.2.205.160	10.2.205.180	10.2.205.21		
	DHCP/static	static	static	DHCP		
	SUB	255.255.255.0	255.255.255.0	255.255.255.0		
	Gateway	10.2.205.254	10.2.205.254	10.2.205.254		
	DNS	10.2.205.254	10.2.205.254	10.2.205.254		
	Mac					

2.2 User & Zugangsdaten (Software)

Client	Gerät	Anwendung	Benutzername	Passwort	Gerätename	domain (AD)
	PI (Host)	OS / PXVirt / SSH	pi	Aa_123456	<i>pilumex5</i>	---
	Webmin / GUI (Proxmox)	Web Interface	root	Aa_123456	<i>pilumex5</i>	---
	Debian Trixie (LXC 102)	Debian-Container	root	As_123456	<i>dietpi-trixie</i>	
Server	Gerät	Anwendung	Benutzername	Passwort	Hostname	
	(Proxmox Host)	PXVirt Console / SSH	root	Aa_123456	Pilumex5	
	(LXC 102)	Debian 13 „Trixie“	root	As_123456	Dietpi-trixie	

3. Erkenntnis vorweg

Architekturproblem!

Während der Recherchen zu dieser Aufgabe kam ich zu folgender Schlussfolgerung:

Die Aufgabenstellung erschien mir so, dass Proxmox auf einem Raspberry Pi5 (ARM64) installiert werden soll um anschließend DietPi-Images unter Proxmox zu virtualisieren.



- ➔ DietPi-Images sind aber x86_64 (Intel/AMD).
- ➔ Eine ARM-CPU kann kein x86-System direkt starten.

Proxmox auf dem Pi hat keine Hardware-Virtualisierung (KVM) für x86_64, und QEMU-Emulation findet kein bootfähiges System (weil CPU-Instruktionen inkompatibel sind).

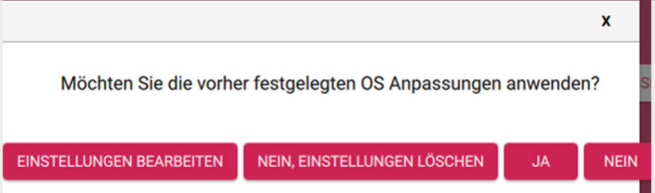
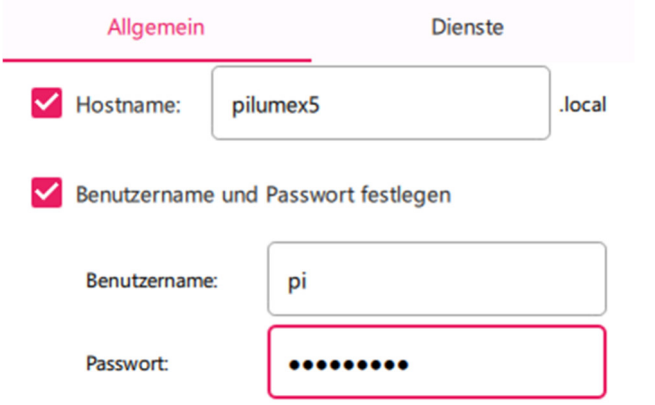
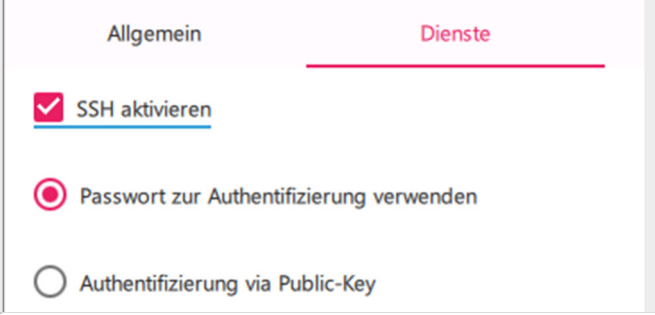
Selbst wenn es starten würde, wäre es extrem langsam (TCG-Emulation)

– aber es scheiterte schon beim Bootloader.

Lösung: DietPi als LXC-Container (ARM64) installieren

Somit hat man ein echtes DietPi-System, nativ auf ARM, mit voller Performance.

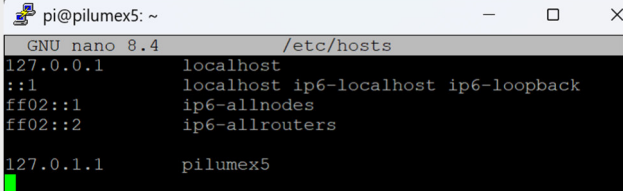
4. Grundinstallation des Hostsystems

Erklärung	Beispiel
Auswahl eines geeigneten Betriebssystems	 Raspberry Pi OS Lite (64-bit) A port of Debian Trixie with no desktop environment (Compatible with Raspberry Pi 3/4/400/5) Veröffentlicht: 2025-10-01 Online - 0.5 GB Download
Auswahl - PI-Modell - Betriebssystem - Datenträger	
Einstellungen bearbeiten und mit JA fortfahren	
Unter „Allgemein“ Eingabe des Hostnames Festlegen von Benutzernamen & Passwort Hostname: pilumex5 User: pi PW: Aa_123456	
Unter „Dienste“ Aktivierung von SSH	
Bestätigung des Schreibvorganges mit	Alle vorhandenen Daten auf 'SanDisk SDDR-B531 USB Device' werden gelöscht. Möchten Sie wirklich fortfahren? JA 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

Zugriff „headless“ auf den Raspi per PuTTY mit dem hostname „pilumex5“ ➔ IP-Adresse: <u>10.2.205.21</u> per DHCP ➔ WLAN ist down!	<pre> oup default qlen 1000 link/ether 2c:cf:67:33:32:a3 brd ff:ff:ff:ff:ff:ff inet 10.2.205.21/24 brd 10.2.205.255 scope global dynamic noprefixroute eth0 valid lft 863975sec preferred lft 863975sec inet6 fe80::5f25:63b:6a08:b746/64 scope link noprefixroute valid lft forever preferred lft forever 3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000 </pre>
--	--

5. Vorbereitung & Grundinstallation für Proxmox

Erklärung	Beispiel
System aktualisieren	<code>sudo apt update && sudo apt upgrade -y</code>
Installiert „CURL“ *	<code>sudo apt install curl bridge-utils -y</code>
*CURL ist eine Programmbibliothek und ein Kommandozeilen-Programm zum Übertragen von Daten in Rechnernetzen.	
Root PW-festlegen new root - PW: As_123456	<pre> sudo passwd root pi@pilumex5:~ \$ sudo passwd root New password: Retype new password: passwd: password updated successfully </pre>
GPG-Key hinzufügen & PXVIRT zu den Quellen hinzufügen	
Du wirst danach als Benutzer root angemeldet und brauchst kein sudo mehr vor Kommandos zu schreiben.	<code>sudo su</code>
Lädt den GPG-Signaturschlüssel von dem angegebenen Server herunter	<code>curl -L https://mirrors.lierfang.com/pxcloud/lierrfang.gpg -o /etc/apt/trusted.gpg.d/lierrfang.gpg</code>
Lädt die Systemvariablen aus der Datei /etc/os-release in die aktuelle Shell.	<code>source /etc/os-release</code>
Damit kann das System nun Pakete aus dieser Quelle installieren (nach apt update).	<code>echo "deb [arch=arm64] https://mirrors.lierfang.com/pxcloud/pxvirt trixie main" sudo tee /etc/apt/sources.list.d/pxvirt-sources.list</code>
Verlässt die Root-Shell und kehrt zum normalen Benutzer zurück	<code>exit</code>
System aktualisieren	<code>sudo apt update</code>
Systemaktualisierung & notwendige Paketinstallation	
System aktualisieren	<code>sudo apt update && sudo apt upgrade -y</code>
Installiert das Paket ifupdown2, ein modernes Netzwerkverwaltungstool	<code>sudo apt install curl bridge-utils ifupdown2 -y</code>
Netzwerkdienst vorbereiten	
deaktiviert den automatischen Start des Dienstes <i>NetworkManager</i> beim Systemstart.	<code>systemctl disable NetworkManager</code>
stoppt den aktuell laufenden NetworkManager-Dienst sofort.	<code>systemctl stop NetworkManager</code>
# interfaces(5) file used by ifup(8) and ifdown(8)	<code>sudo nano /etc/network/interfaces</code>

<pre> auto lo iface lo inet loopback allow-hotplug eth0 auto eth0 iface eth0 inet manual auto vmbr0 iface vmbr0 inet static address 10.2.205.160/24 gateway 10.2.205.1 bridge-ports eth0 bridge-stp off bridge-fd 0 </pre>	
Verhindert Bootverzögerungen durch Netzwerk-Wartezeit	<code>sudo systemctl mask systemd-networkd-wait-online.service</code>
Hosts-Datei anpassen	
öffnet die hosts-Datei → 	<code>sudo nano /etc/hosts</code> <pre> 127.0.0.1 localhost ::1 localhost ip6-localhost ip6-loopback ff02::1 ip6-allnodes ff02::2 ip6-allrouters </pre>
ersetze die letzte Zeile durch diese →	10.2.205.160 pilumex5
System neu starten	<code>sudo reboot</code>

6. Installation von Proxmox (PXVIRT)

Erklärung	Beispiel
System auf neuesten Stand bringen	<code>sudo apt update</code>
Die Installation dauert nun etwas.... Folgende Abfrage beantworten pxvirt-sources.list: N und Enter	<code>sudo apt install proxmox-ve pve-manager qemu-server pve-cluster -y</code>
System neu starten	<code>sudo reboot</code>
Aufruf der grafischen Benutzeroberfläche von PROXMOX https://pilumex5:8006 oder https://10.2.205.160:8006	

Login

root
As_123456

PXVIRT Login

User name:

Password:

Realm: Linux PAM standard authentication

Language: English - English

Save User name: ☐ **Login**

Login erfolgreich, System läuft!

7. USB-Festplatte anschließen und Partitionen einrichten

Erklärung	Beispiel
Prüft angeschlossene Laufwerke	lsblk
Startet Partitionierung der externen HDD	sudo fdisk /dev/sda
Erstellt zwei Partitionen – 100 GB und Rest	Eingaben in fdisk:
	n → Partition 1 (+100G)
	n → Partition 2 (Rest)
	w → speichern
Formatiert die Partitionen	bash sudo mkfs.ext4 /dev/sda1 sudo mkfs.ext4 /dev/sda2
Erzeugt Einhängepunkte	bash sudo mkdir -p /mnt/pve/proxmox_hdd sudo mkdir -p /mnt/pve/proxmox_iso
Hängt die Partitionen ein	bash sudo mount /dev/sda1 /mnt/pve/proxmox_hdd sudo mount /dev/sda2 /mnt/pve/proxmox_iso
UUIDs der Laufwerke anzeigen	sudo blkid

```

pi@pilumex5:~$ sudo blkid
/dev/mmcblk0p1: LABEL_FATBOOT="bootfs" LABEL="bootfs" UUID="1C94-4EC3" BLOCK_SIZE="512" TYPE="vfat" PARTUUID="blb0bff8-01"
/dev/mmcblk0p2: LABEL="rootfs" UUID="f0abac56-08be-42e2-8726-9baa083e8685" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="blb0bff8-02"
/dev/loop0: TYPE="swap"
/dev/zram0: LABEL="zram0" UUID="a4ee0aeb-f29a-498d-ac51-b1c4a70d8df0" TYPE="swap"
/dev/sda1: UUID="707941fd-2905-430e-8ce1-3b37f49d8843" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="f6fe3ca9-ladf-47d2-8fea-dfafe28bed84"
/dev/sda2: UUID="c50264a7-94d9-46a9-a97e-3f6e83780279" BLOCK_SIZE="4096" TYPE="ext4" PARTUUID="0c0c553b-345f-4205-b9ea-157878f24dd6"

```

Diese UUID per copy-paste im Anschluss einfügen

Dauerhafte Einbindung in /etc/fstab

bash sudo nano /etc/fstab # Beispielzeilen:

```

UUID=xxxx /mnt/pve/proxmox_hdd ext4 defaults 0 1
UUID=yyyy /mnt/pve/proxmox_iso ext4 defaults 0 1

```

pi@pilumex5: ~

```

GNU nano 8.4 /etc/fstab
proc /proc proc defaults 0 0
PARTUUID=blb0bff8-01 /boot/firmware vfat defaults 0 2
PARTUUID=blb0bff8-02 / ext4 defaults,noatime 0 1
UUID=707941fd-2905-430e-8ce1-3b37f49d8843 /mnt/pve/proxmox_hdd ext4 defaults 0 1
UUID=c50264a7-94d9-46a9-a97e-3f6e83780279 /mnt/pve/proxmox_iso ext4 defaults 0 2

```

Testet die Einbindung

bash sudo mount -a df -h

```

pi@pilumex5:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G     0  3.9G   0% /dev
tmpfs           1.6G   15M   1.6G   1% /run
/dev/mmcblk0p2  58G   6.4G   50G  12% /
tmpfs           4.0G   47M   3.9G   2% /dev/shm
tmpfs           5.0M   48K   5.0M   1% /run/lock
tmpfs           1.0M     0   1.0M   0% /run/credentials/systemd-journald.service
tmpfs           4.0G     0   4.0G   0% /tmp
/dev/mmcblk0p1  510M   72M  439M  14% /boot/firmware
/dev/fuse       128M   16K  128M   1% /etc/pve
tmpfs           1.0M     0   1.0M   0% /run/credentials/getty@tty1.service
tmpfs           1.0M     0   1.0M   0% /run/credentials/serial-getty@ttyAMA10.service
tmpfs           806M   48K   806M   1% /run/user/1000
/dev/sda1       98G   2.1M   93G   1% /mnt/pve/proxmox_hdd
/dev/sda2      122G   2.1M  115G   1% /mnt/pve/proxmox_iso

```

Beide Partitionen sind gemountet!!

8. PXVIRT-Speicher konfigurieren

Dieser Schritt wurde dokumentiert jedoch bin ich nicht sicher ob er notwendig war.

Erklärung	Beispiel
Öffne die Webinterface	https://10.2.205.160:8006
Gehe zu: Datacenter → Storage → Add → Directory	
ID:	proxmox_hdd
Directory:	/mnt/pve/proxmox_hdd
Content:	Disk image, Container
Nodes:	pilumex5
Enabled aktivieren → Add	

Add: Directory

General Backup Retention

ID: proxmox_hdd Nodes: pilumex5

Directory: /mnt/pve/proxmox_hdd Enable: ☒

Content: Disk image, Container Shared: ☐

Help Advanced ☐ Add

ID: proxmox_iso
Directory: /mnt/pve/proxmox_iso
Content: ISO image, VZDump backup file,
Container template
Nodes: dein Hostname

Add: Directory

General Backup Retention

ID: proxmox_iso Nodes: pilumex5

Directory: /mnt/pve/proxmox_iso Enable: ☒

Content: Disk image, ISO image, Shared: ☐

Help Advanced ☐ Add

9. Debian 13 (Trixie) Container erstellen

Wir benötigen eine Datei „RootFS“, die wir downloaden.

Erklärung	Beispiel
SSH-Verbindung auf den Host und ins Verzeichnis wechseln	cd /mnt/pve/proxmox_iso
sudo wget https://fra1lxdmirror01.do.letsbuildthe.cloud/images/debian/trixie/arm64/cloud/20251025_05%3A24/rootfs.tar.xz	
Wenn download fertig, erstellen wir einen container mit diesem Befehl →	pct create 102 /mnt/pve/proxmox_iso/rootfs.tar.xz --arch arm64 --hostname dietpi-trixie --rootfs proxmox_hdd:8 --memory 512 --swap 512 --features nesting=1,keyctl=1 --net0 name=eth0,bridge=vbr0,ip=dhcp --unprivileged 1
	

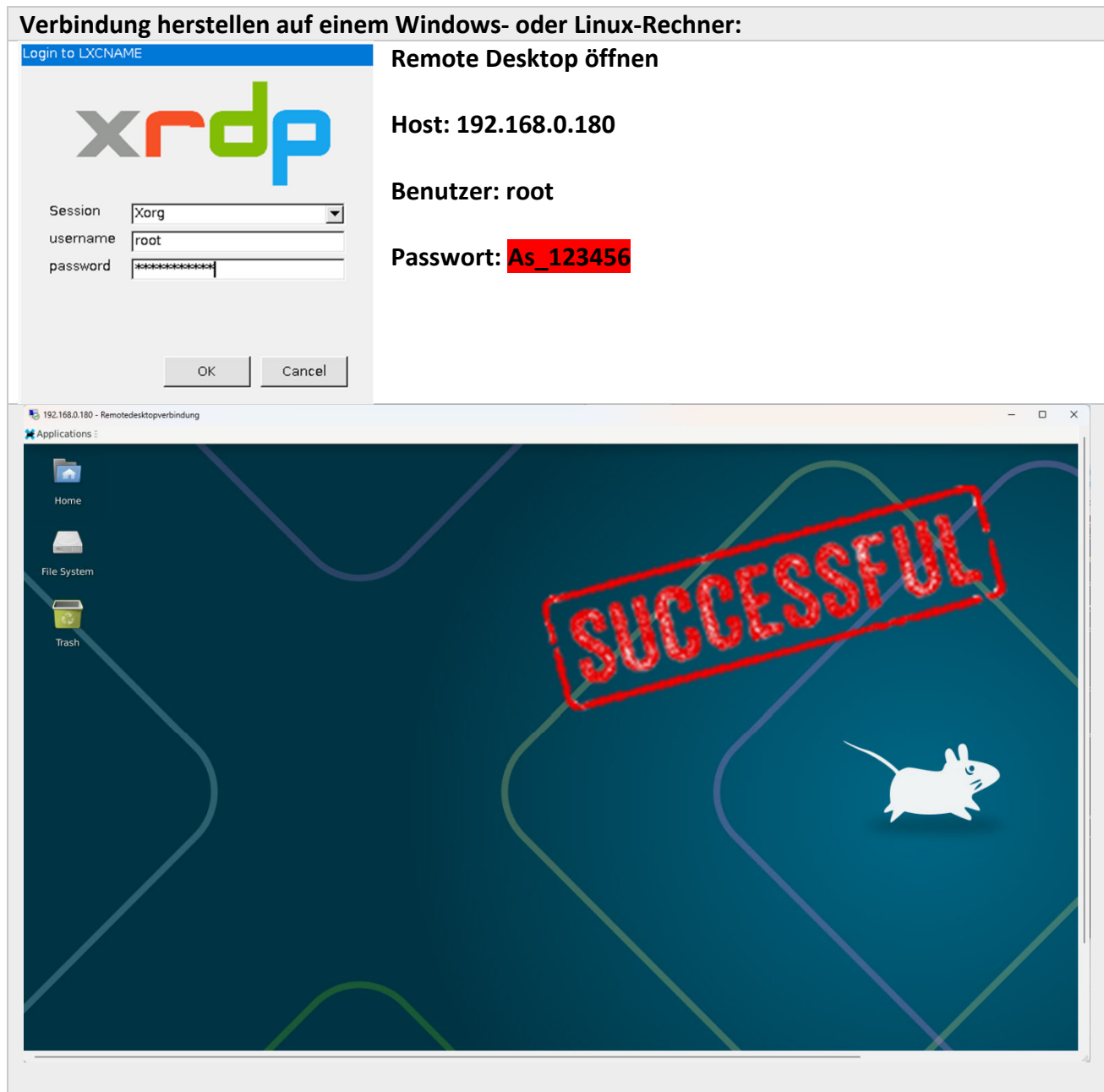
NUN die shell des nodes „pilumex5“ öffnen und nachfolgende Eingaben machen

Netzwerk hinzufügen	pct set 102 --net0 name=eth0,bridge=vibr0,ip=dhcp
Container starten	pct start 102
In Container wechseln	pct enter 102 <pre>root@pilumex5:~# pct enter 102 root@LXCNAME:~#</pre>
Netzwerk und DNS im Container aktivieren	
DHCP aktivieren	<pre>mkdir -p /etc/systemd/network cat >/etc/systemd/network/eth0.network <<'EOF' [Match] Name=eth0 [Network] DHCP=yes EOF systemctl enable --now systemd-networkd</pre>
DNS konfigurieren	<pre>apt install -y systemd-resolved ln -sf /run/systemd/resolve/stub-resolv.conf /etc/resolv.conf systemctl enable --now systemd-resolved</pre>
Verbindung testen	<pre>ping -c2 1.1.1.1 ping -c2 deb.debian.org</pre>

10. GUI (Desktop) & Remote Desktop (RDP)

Um „grafischen“ Zugriff auf das headless-System zu erlangen habe ich einen Desktop sowie RDP installiert.

Erklärung	Beispiel
Desktop und RDP installieren	<pre>apt update && apt install -y xfce4 xfce4-goodies xrdp echo xfce4-session > ~/.xsession systemctl enable --now xrdp</pre>
Root-Passwort setzen PW=As_123456	Passwd



11. Fehlerbehebung & bekannte Probleme

Problem	Ursache	Lösung
Temporary failure resolving deb.debian.org	Kein DNS im Container	DNS mit systemd-resolved aktivieren
„storage local-lvm does not exist“	Standard-Storage fehlt	eigenen Speicher proxmox_hdd verwenden
Kein Boot der DietPi-VM	x86-Image inkompatibel	ARM-RootFS (Trixie arm64) verwenden