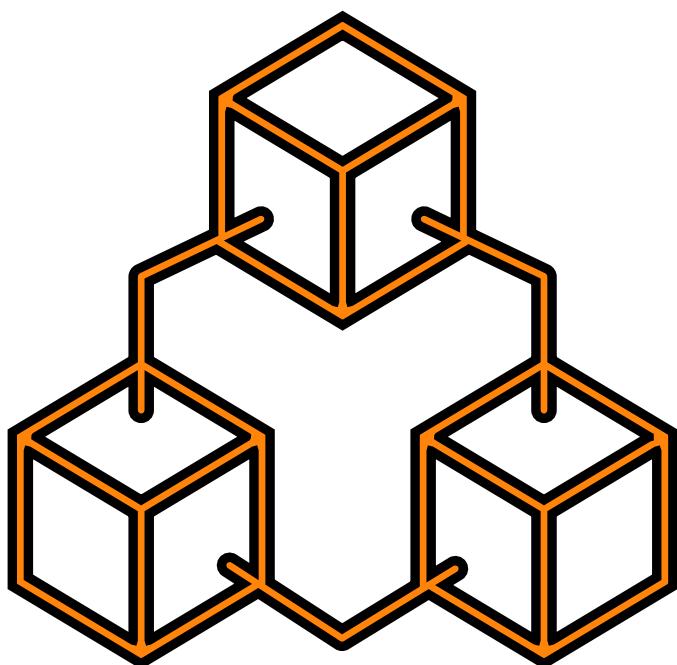


# AUTOMATISIERUNG HOSTING

Services mit Containern bereitstellen



Milan Milivojevic

# Projekt Dokumentation

## Services mit Containern bereitstellen

Klassifizierung	[REDACTED]
Status	genehmigt zur Nutzung
Programmname	[REDACTED]
Projektnummer	2443323
Projektleiter	Milan Milivojevic
Version	1.0
Datum	13. August 2025
Auftraggeber	[REDACTED]
Autor/Autoren	Milan Milivojevic (MiM)
Verteiler	Milan Milivojevic

## Änderungsverzeichnis

Version	Datum	Änderung	Autor
0.5	13.08.2025	Informationen sammeln und aufschreiben	MiM
0.6	15.08.2025	Planungsphase abgeschlossen vollständig	MiM
0.7	16.08.2025	Entscheidung getroffen und aufgeschrieben	MiM
0.8	17.08.2025	Realisierung wurde fertiggestellt	MiM
0.9	19.08.2025	Testing durchgeführt und dokumentiert	MiM
1.0	22.08.2025	Dokumentation vollständig.	MiM

Tabelle 1 Änderungsverzeichnis

## Beschreibung

Das Betriebshandbuch liefert alle Informationen, die der Betreiber benötigt, um das System ordnungsgemäss betreiben und im Fall von Problemen richtig reagieren zu können. Alle für den Betreiber betriebsrelevanten Informationen sind im Betriebshandbuch dokumentiert.

Dieses Betriebshandbuch liefert alle relevanten Informationen für den laufenden Betrieb. Es dient dem Betriebspersonal als Referenz, um das System ordnungsgemäss zu betreiben und bei Problemen korrekt zu reagieren.

## Inhaltsverzeichnis

1.	Informationen .....	4
1.1.	Auftragsbeschreibung.....	4
1.2.	Ausgangslage und Rahmenbedingungen .....	4
1.3.	Grundlagenrecherche .....	5
1.3.1.	Virtualisierung.....	5
1.3.2.	Container und Micro Services.....	5
2.	Planen .....	6
2.1.	Zeitplan .....	6
2.2.	Namenskonzept .....	6
2.3.	IP-Konzept .....	6
2.4.	Systemanforderungen.....	7
2.5.	Architektur-Schema.....	7
3.	Entscheiden.....	8
3.1.	Entscheidung für Virtualisierungslösung.....	8
3.2.	Entscheidung für Containerlösung.....	8
3.3.	Entscheidung für Automatisierung .....	8
3.4.	Alternative Optionen .....	8
3.5.	Installationsreihenfolge .....	8
3.6.	Theorie zum Ablauf der Umgebung .....	9
4.	Realisieren.....	10
4.1.	ISO-Datei bereitstellen.....	10
4.2.	Hypervisor einrichten.....	10
4.3.	Virtuelle Maschine erstellen .....	12
4.4.	Betriebssystem installieren .....	16
4.4.1.	Proxmox GUI über Webseite .....	22
4.5.	Netzwerk Konfigurieren .....	26
4.6.	Container installieren .....	32
4.7.	Skripte einbinden .....	37
4.7.1.	Automatisierte Services und Container .....	37
4.7.2.	Automatisierung Virtuelle-Switch und Hyper-V.....	38
5.	Kontrollieren .....	39
5.1.	T01 Ausführen und Prüfen .....	40

5.2.	T02 Ausführen und Prüfen .....	41
5.3.	T03 Ausführen und Prüfen .....	41
5.4.	T04 Ausführen und Prüfen .....	42
5.5.	T05 Ausführen und Prüfen .....	43
5.6.	T06 Ausführen und Prüfen .....	44
6.	Auswerten.....	45
6.1.1.	Herausforderungen.....	45
7.	Abbildungsverzeichnis .....	46
8.	Tabellenverzeichnis .....	48
9.	Quellenverzeichniss .....	48

## 1. Informationen

Alles erstes werden wir alle Informationen, die wir gesammelt haben, aufschreiben damit wir ein besseres Verständnis über das Projekt bekommen.

### 1.1. Auftragsbeschreibung

Das Ziel dieses Projektes ist die bis her Verlangsamte Methode der manuellen und Individuellen System Erstellung für jeden Kunden mit Automatisierungen zu ersetzen.

Das Problem der Manuellen Verwaltung kann mit Automatisierungen gelöst werden. Dies bringt viele Vorteile. Der Wichtigste Vorteil wäre das Einsparen von Arbeitszeit und Kosten.

Es gelten folgende Anforderungen:

- Die Erstellung von VMs, Container und Services sollen automatisiert werden.
- Es sollen folgende Umgebungen erstellt werden: Entwicklung, Test, Produktion.
- Es sollen Skripte für die Automatisierungen erstellt werden.
- Die neuen Skripte sollen getestet werden.
- Zugriff von anderen Geräten auf Server(Container und VM)

### 1.2. Ausgangslage und Rahmenbedingungen

#### Vorgaben:

Das Projekt soll bis zum 25.08.2025 abgeschlossen und demonstriert werden.

Es gibt keine Einschränkungen in der Nutzung von Software oder Hardware.

#### Endprodukt:

- Automatisierte Installation und Konfiguration von VMs durch Skripte.
- Automatisierte Bereitstellung von Containern mit Services durch Skripte.
- Abgeschlossene und Geprüfte Dokumentation mit Anleitung.
- Virtuelle Umgebungen: Entwicklung, Test, Produktion.

Damit Sie ein besseres Verständnis über das Projekt erhalten, werden wir als nächstes ein paar Grundlagen zu Containern, Micro Services und Virtualisierung erläutern.

## 1.3. Grundlagenrecherche

### 1.3.1. Virtualisierung

Eine Virtualisierungssoftware wie zum Beispiel Proxmox dient dazu, mehrere virtuelle Computer (VMs) oder Container auf einer einzigen physischen Maschine zu betreiben. Man kann sich das wie ein Hotel vorstellen. Der physische Server oder Client ist das Gebäude. Die Virtualisierungssoftware ist der Hotelmanager, der die Zimmer (Rechenressourcen) zuteilt. Die virtuellen Maschinen oder Container sind die einzelnen Hotelzimmer, in denen jeweils ein eigenes <Gast> Betriebssystem läuft.

### 1.3.2. Container und Micro Services

Ein **Container** ist wie ein abgeschotteter Raum innerhalb des bestehenden Betriebssystems. Er teilt sich den Kernel mit dem Hostsystem und benötigt daher kein eigenes vollständiges Betriebssystem, nur die Dateien und Bibliotheken, die er tatsächlich braucht, um seine Anwendung laufen zu lassen.

Das heißtt. Container starten extrem schnell, verbrauchen viel weniger Ressourcen und sind ideal, wenn man viele kleine, spezialisierte Dienste gleichzeitig betreiben will, wie zum Beispiel eine Datenbank, einen Webserver, einen Mailserver).

Allerdings sind sie weniger stark isoliert als VMs und sie müssen in der Regel mit demselben Grundbetriebssystem wie der Host kompatibel sein (Linux-Container auf Linux, Windows-Container auf Windows).

**Container** eignen sich perfekt für schnelle, ressourcensparende Anwendungen, die oft skaliert oder aktualisiert werden müssen. Sie sind leicht zu kopieren und zu verschieben, aber nicht so stark von der Hostumgebung getrennt.

**VMs** eignen sich perfekt, wenn man maximale Trennung, volle Flexibilität bei der Wahl des Betriebssystems oder sicherheitskritische Anwendungen braucht. Sie sind stabil und universell einsetzbar, aber dafür ressourcenintensiver.

Eine **Microservice-Architektur** ist ein Ansatz in der Softwareentwicklung, bei dem eine grosse Anwendung in viele kleine, eigenständige Dienste (Services) aufgeteilt wird. Container sind dafür ideal:

- **Jeder Microservice** kommt in seinen **eigenen Container**, mit genau den Abhängigkeiten, die er braucht.
- Startet in Sekunden, verbraucht nur minimal Ressourcen.
- Falls ein Service abstürzt, betrifft das die anderen nicht.
- Man kann beliebig viele Instanzen eines Services starten, wenn mehr Leistung gebraucht wird.

## 2. Planen

Dieses Projekt erfordert eine genaue Planung da sonst die Fehlersuche durch die Komplexität des Auftrags zu viel Zeit in Anspruch nehmen würde.

### 2.1. Zeitplan

Datum	Aufgabe	Rechtzeitig	Verspätet	Soll	Ist
13. Aug	Projektantrag und Zielbeschreibung fertigstellen. Architektur-Schema zeichnen. Projektplan grob erstellen.			4.5	4.5
14. Aug	Grundlagen-Dokument (A,B,C Kompetenzraster) schreiben, Dokumentation Einleitung und Anforderungen erfassen, Systemarchitektur Beschreiben und Erfassen.			5	2.0
15. Aug	Zielarchitektur fertig ausarbeiten (Daten, Sicherheit, Automatisierung, Tests)			5	3.0
16. Aug	Systemanforderungen dokumentieren. Installationsreihenfolge festlegen. ISO-Dateien vorbereiten			5	3.5
17. Aug	VM-Umgebungen aufsetzen und testen.			5	5.0
18. Aug	Docker-Umgebung mit DHCP/DNS einrichten.			5	3.0
19. Aug	Automatisierungsskripte entwickeln und testen.			5	11.0
20. Aug	Funktionstests und Sicherheitsprüfungen.			5	3.0
21. Aug	Technische Anleitung schreiben.			5	5.0
22. Aug	Skripte dokumentieren. Vorstellung vorbereiten.			5	4.0
23. Aug	Projektbericht finalisieren. Dokumentation finalisieren			5	5.0
24. Aug	Letzte Überprüfung & Formatierung aller Dokumente (5 Std.)			5	5.0
25. Aug	<b>Abgabe</b>			59.5 Std.	54.0 Std

Abbildung 1 Zeitplan

### 2.2. Namenskonzept

Als Namenskonzept werden wir immer die ersten 3 Buchstaben der Systeme nehmen und nach den Buchstaben 2 Zahlen für eine Aufzählung bei mehreren Systemen benutzen. Für die Skripts werden wir die Abkürzung des System Namens mit der Abkürzung SC verlängern. Siehe Beispiel in der Tabelle.

VMs	Container	Skripte
ENT01, TES01, PRO01	CON-DNS, CON-WS, CON-DB, CON-WB	DB-SC, WS-SC, WB-SC, CON-DB-SC, CON-WS-SC, CON-WB-SC, ENT01-SC

Tabelle 2 Namenskonzept

### 2.3. IP-Konzept

DNS	Virtuelle Maschine	PROXMOX	Container Web-Server	Container MariaDB	Container Workbench
1.1.1.1 9.9.9.9 Oder 172.16.100.1	172.16.100.1 Gateway + NAT	172.16.100.10	172.16.100.20 Falls mehrere Bis .29	172.16.100.30 Falls mehrere Bis .39	172.16.100.40 Falls mehrere Bis .49

Tabelle 3 IP-Konzept

## 2.4. Systemanforderungen

Damit wir das System einrichten können brauchen wir folgende Hardware und Software:

**Hardware:** CPU Dynamisch / RAM 32 GB(Dynamisch) / Speicher 160 GB

**Virtualisierung Software:** Hyper-V Feature Windows 11 / Proxmox VE 9.0

	Virtuelle Maschine	Container Web-Server	Container Datenbank	Container Workbench-CLI
<b>CPU</b>	dynamisch	4	4	4
<b>RAM</b>	32 GB	4 GB	4 GB	4 GB
<b>Speicher</b>	160 GB	40 GB	40 GB	40 GB
<b>Betriebssystem</b>	Proxmox VE 9.0	Ubuntu 24.04	Ubuntu 24.04	Ubuntu 24.04
<b>Software</b>		Apache	MariaDB-Server	MySQL-CLI

Tabelle 4 Systemanforderungen

Verwenden Sie die erwähnten Versionen! Ansonsten kein Support möglich!

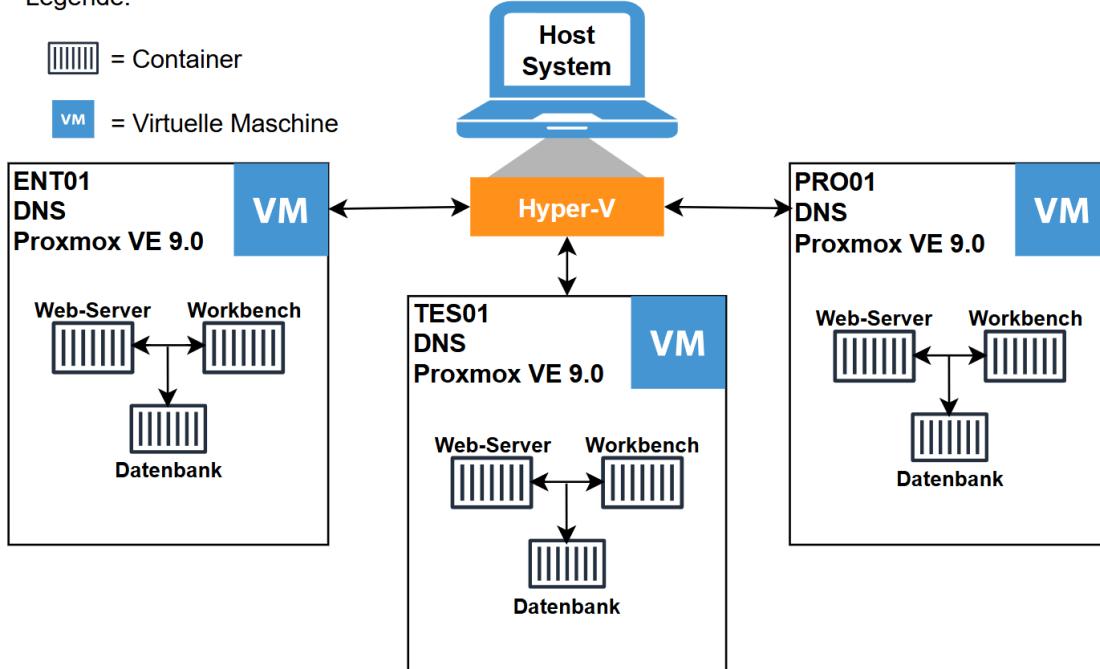
## 2.5. Architektur-Schema

Damit wir ein leichteres Verständnis über die Architektur des Projekts erlangen haben wir ein Schema gezeichnet und vertieft erklärt. Wie Sie auf dem Bild sehen können, verwenden wir als Host-System unser Laptop mit 32GB RAM und Windows 11 Pro.

Windows bietet als Virtualisierungssoftware den Hyper-V-Manager. Auf diesem Manager werden wir die VMs erstellen und konfigurieren. Die VMs wiederum werden 4 Container enthalten, die alle ein Service enthalten. Die Services, die wir brauchen sind Web-Server, Datenbank, DNS und Workbench. Die verschiedenen Services werden alle miteinander kommunizieren können durch verschiedene Protokolle.

Legende:

- [Container] = Container
- [VM] = Virtuelle Maschine



### 3. Entscheiden

Wir haben uns für die geplante Architektur entschieden da es durch die Aufteilung der Services auf verschiedene Container die Wartbarkeit erleichtert und Fehleranfälligkeit minimiert. Wenn ein Container ausfällt, werden nicht alle Services gestoppt.

#### 3.1. Entscheidung für Virtualisierungslösung

Wir haben uns für die **Hyper-V** Virtualisierungslösung entschieden. Diese ist auf unseren Host-Systemen integriert und ist kostenlos. Wir haben schon mehrmals Automatisierungen mit Hyper-V durchgeführt und haben daher vorhandenes Wissen. Die Kompatibilität mit Proxmox ist gewährleistet. Hyper-V verbraucht auch weniger Ressourcen auf unserem System da wir keine Software installieren müssen.

#### 3.2. Entscheidung für Containerlösung

Als Container Lösung haben wir **LXD** ausgewählt. LXD entspricht allen Anforderungen, die wir hatten. Es ist mit Proxmox eine integrierte Containerlösung, alles in einem. Die Community ist gross und es ist Open Source. Es ermöglicht das Auswählen von verschiedenen Templates (Betriebssystemen) beim Container erstellen.

#### 3.3. Entscheidung für Automatisierung

Wir werden **PowerShell und Bash** verwenden. Da diese über mehrere Systeme kompatibel sind und keine Entwicklungsumgebung wie VSC benötigen da sie Integrierte Syntax Unterstützung enthalten.

#### 3.4. Alternative Optionen

**Virtualisierung:** VMware, VirtualBox

**Containerlösung:** Dokker, Podman,

Nach Recherche haben wir gemerkt das andere Virtualisierungssoftwares wie VMware mehrere zusätzlichen Softwares und Extension gebraucht hätte, um die Automatisierung zu ermöglichen. Proxmox kann VMs und Container mit LXD erstellen und gleichzeitig noch als Betriebssystem dienen, weshalb wir uns für diese Lösung anstatt Dokker entschieden haben.

#### 3.5. Installationsreihenfolge

Die Installationsreihenfolge muss unbedingt beachtet werden, um Fehler bei der Ausführung zu vermeiden. Der Ablauf sieht wie folgt aus.

1. ISO-Datei bereitstellen
2. Hypervisor einrichten
3. Virtuelle Maschine erstellen
4. Betriebssystem installieren
5. Netzwerk konfigurieren
6. Container installieren
7. Skripte einbinden
8. Kontrollieren
9. Auswerten

## 3.6. Theorie zum Ablauf der Umgebung

### Microservices + Container + Proxmox – wie das zusammenhängt:

Eine **Microservice-Architektur** lebt davon, dass viele kleine, unabhängige Dienste laufen.

Damit das gut funktioniert, braucht man eine Umgebung, in der man:

- schnell neue Services starten kann
- sie sauber voneinander trennt
- flexibel skaliert
- im Idealfall auch leicht sichern oder verschieben kann

**Container** (LXD von Proxmox) sind dafür ideal:

- Jeder Microservice kommt in **seinen eigenen Container**, mit genau den Abhängigkeiten, die er braucht.
- Startet in Sekunden, verbraucht nur minimal Ressourcen.
- Falls ein Service abstürzt, betrifft das die anderen nicht.
- Man kann beliebig viele Instanzen eines Services starten, wenn mehr Leistung gebraucht wird.

**Virtualisierung** Proxmox ist die **Management-Ebene**.

- Wir installieren Proxmox direkt auf dem Host (als Betriebssystem).
- Darin können wir **Container für die Microservices anlegen**, Beispiel. einen für Datenbank, einen für Web-Server, einen für Authentifizierung.
- Falls nötig, können wir zusätzlich **VMs** für spezielle Dienste starten (z. B. Windows-Software).
- Alles steuern wir bequem über die Proxmox-Weboberfläche, inklusive Backups, Snapshots und Ressourcen-Zuweisung.

**Kurz gesagt:**

Microservices = die **Bausteine** der Anwendung

Container = die **Boxen**, in denen diese Bausteine laufen

Proxmox = der **Werkzeugkasten**, mit dem wir die Boxen organisieren, starten und verwalten

Jetzt haben wir die Funktionen Oberflächlich angeschaut und verstehen somit die Grundlagen. So können wir uns leichter ein Bild davon machen, was genau wir aufbauen werden. Als nächstes fangen wir mit dem ersten Schritt der Realisierungsphase an:

### Hypervisor einrichten (Hyper-V)

## 4. Realisieren

In der Realisierungsphase haben wir eine Anleitung mit Erklärungen wie das System aufgebaut wird und wo es Probleme haben kann und wie man diese löst.

### 4.1. ISO-Datei bereitstellen

Damit wir das System aufbauen können brauchen wir das Betriebssystem Proxmox, dieses können wir unter folgendem Link als ISO-File installieren:

<https://www.proxmox.com/en/downloads/proxmox-virtual-environment/iso/proxmox-ve-9-0-iso-installer>

### 4.2. Hypervisor einrichten

Als erstes richten wir unseren Hypervisor (Hyper-V) auf unserem Host-System ein. Folgen Sie denn nächstes schritten. [Hyper-V automatisch einrichten](#)

**Suchen Sie** in der Suchleiste auf Ihrem Gerät die **Windows-Features** und **öffnen Sie diese**:

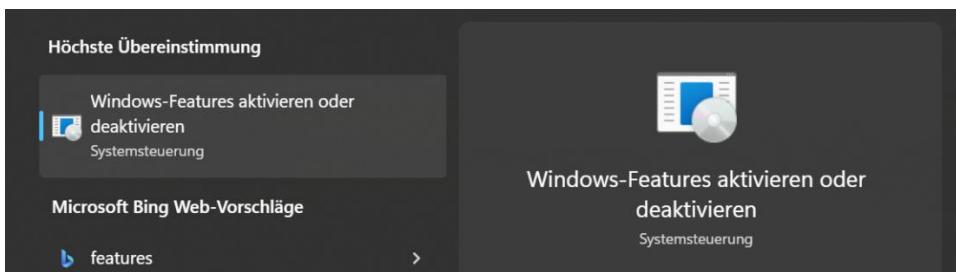


Abbildung 2 Windows-Features

**Suchen Sie** in den Features den Teil **Hyper-V** und **Aktivieren** Sie wie im Bild zu sehen, die Blauen Häkchen:

Bestätigen Sie mit **OK**

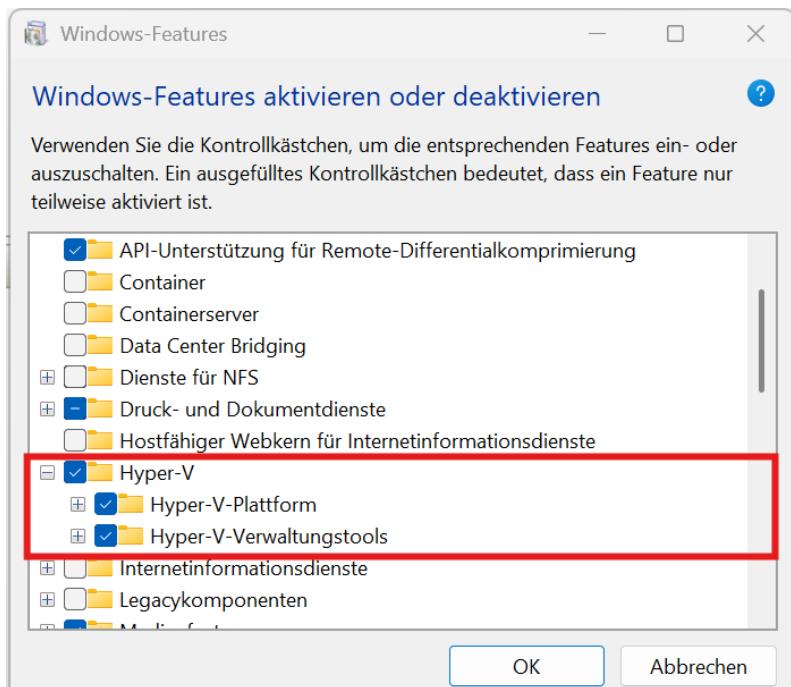


Abbildung 3 Hyper-V Aktivieren

Als nächstes öffnen wir den Hyper-V-Manager:

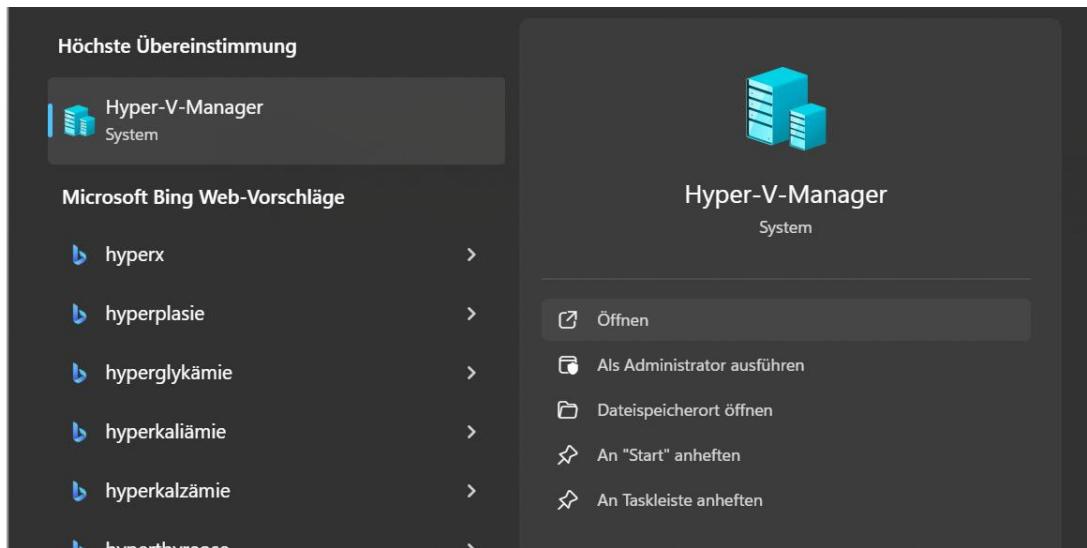


Abbildung 4 Hyper-V-Manager

Wie Sie sehen können, haben wir die Virtualisierung Umgebung Hyper-V aktiviert und können jetzt mit der Erstellung der Virtuellen Maschine fortfahren.

Um eine **neue Maschine** zu erstellen, wählen Sie Ihr Gerät und gehen auf **Neu**:

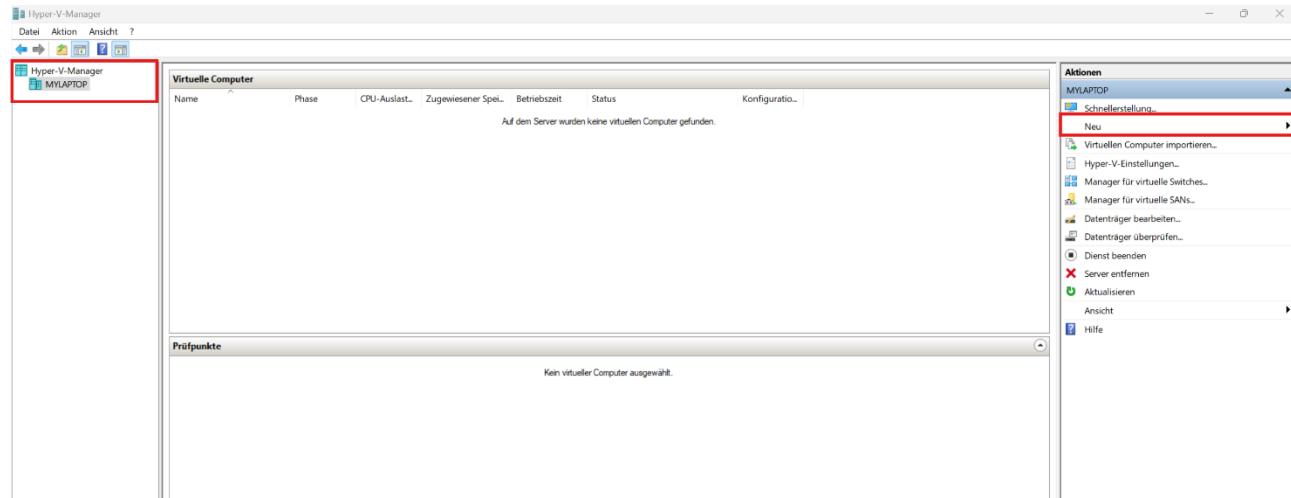


Abbildung 5 Hyper-V-Menu

### 4.3. Virtuelle Maschine erstellen

Jetzt erstellen wir eine Virtuelle Maschine mit dem Betriebssystem Proxmox wie in der Planung erwähnt.

Wenn Sie auf **Neu** gehen sehen Sie die Option **Virtueller Computer** wählen Sie diese aus:

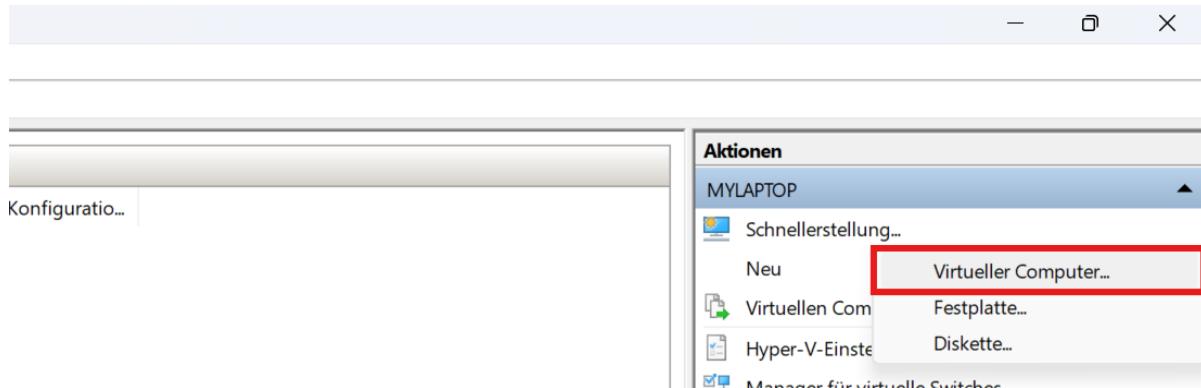


Abbildung 6 Virtueller Computer

Ihnen werden **Vorbemerkungen** aufgezeigt. Lesen Sie diese sorgfältig durch und drücken dann auf **Weiter >**

Als nächstes geben wir der **VM** einen **Namen** (Namenskonzept). Dann richten wir den **Speicherort** der VM ein und gehen auf **Weiter>**

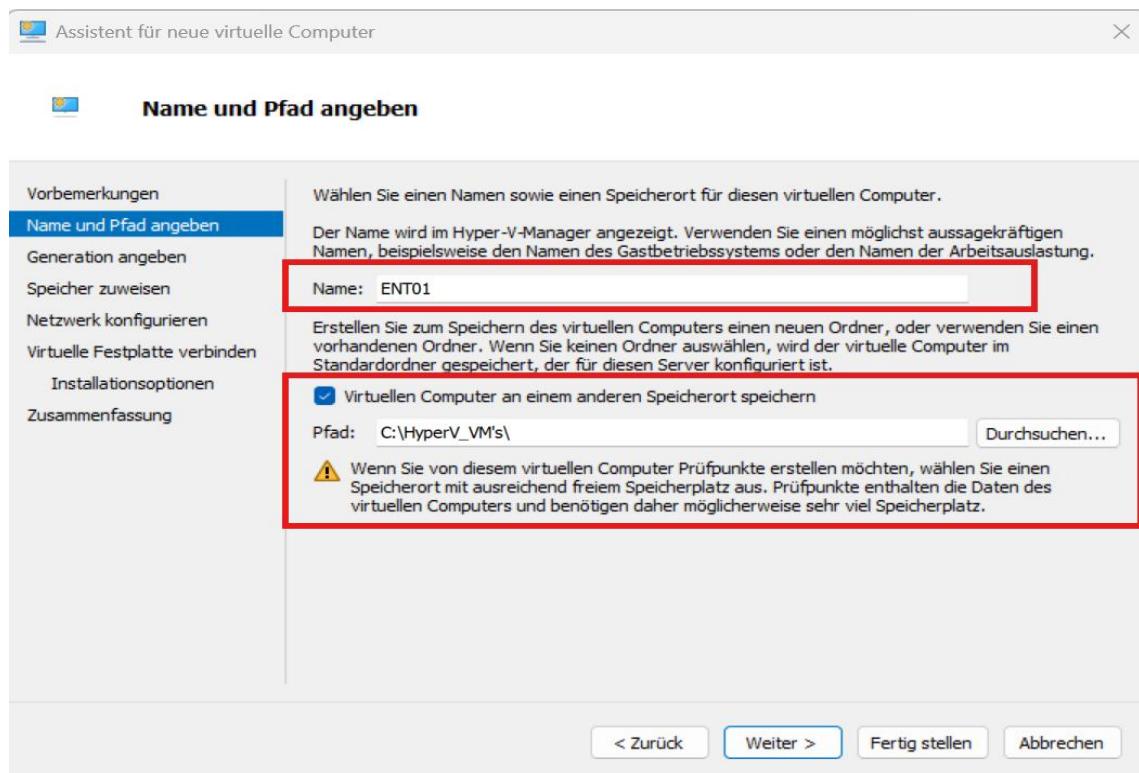


Abbildung 7 VM-Speicherort

Wählen Sie als nächstes die **Generation 2 und Weiter>**:



Abbildung 8 VM-Gastbetriebssystem 64-bit

Wie schon in den Systemanforderungen erwähnt brauchen Sie **mindestens 32 GB RAM** um dieses System zu betreiben.

Also **schreiben Sie wie im Bild markiert** die Grösse des zu Verfügung stehenden RAMs und **aktivieren das Blaue Häkchen**, um die RAM-Nutzung zu optimieren.

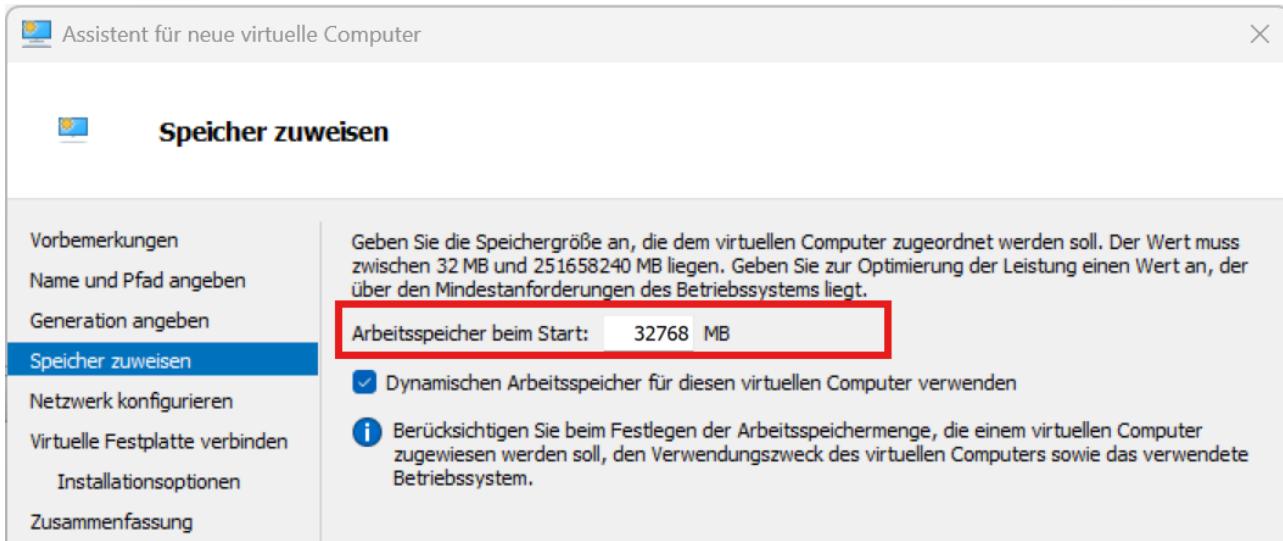


Abbildung 9 VM-RAM

Gehen Sie auf **Weiter>**

Für jetzt lassen wir die Netzwerk Konfiguration so wie sie ist und gehen **Weiter>**

Damit wir unsere Festplatten auf dem Host nicht zufällig bereinigen, erstellen wir eine **Virtuelle Festplatte** und geben den **Pfad** an wo diese gespeichert werden soll.

Schauen Sie das Sie die **Festplatte dort speichern, wo Sie die Virtuelle Maschine gespeichert haben.**

**Geben Sie** der Maschine **160GB** und gehen Sie **Weiter>**

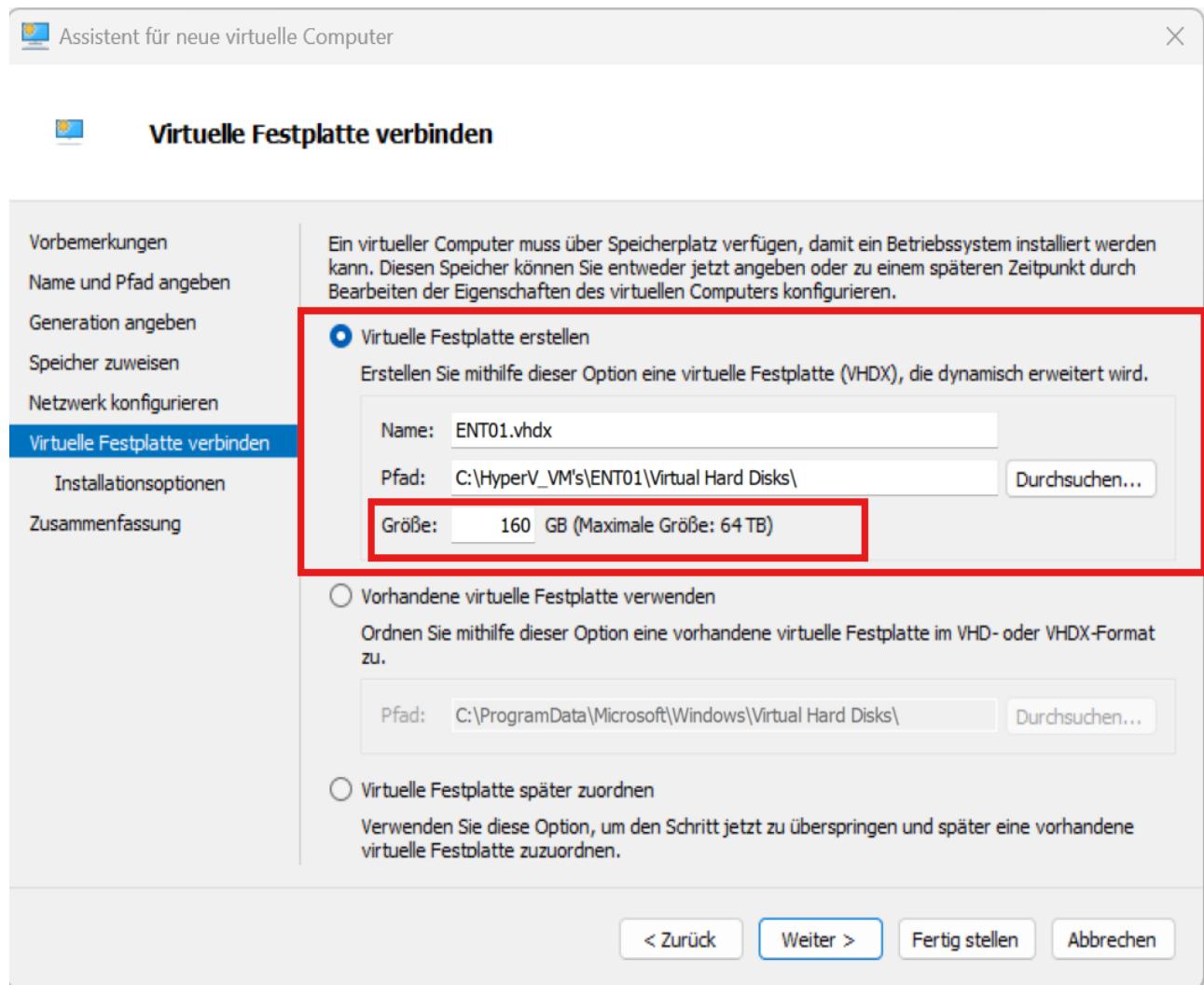


Abbildung 10 VM-Speicher

Jetzt können wir das vorhin installierte ISO-File für Proxmox einbinden.

Wählen Sie, **Betriebssystem von CD/DVD-ROM installieren** und gehen Sie dann auf **Durchsuchen...**

Das ISO-File ist im **Download Ordner**. Falls Sie die Datei **nicht bewegt** haben!

**Wählen Sie das ISO-File** und gehen Sie auf **Weiter>**

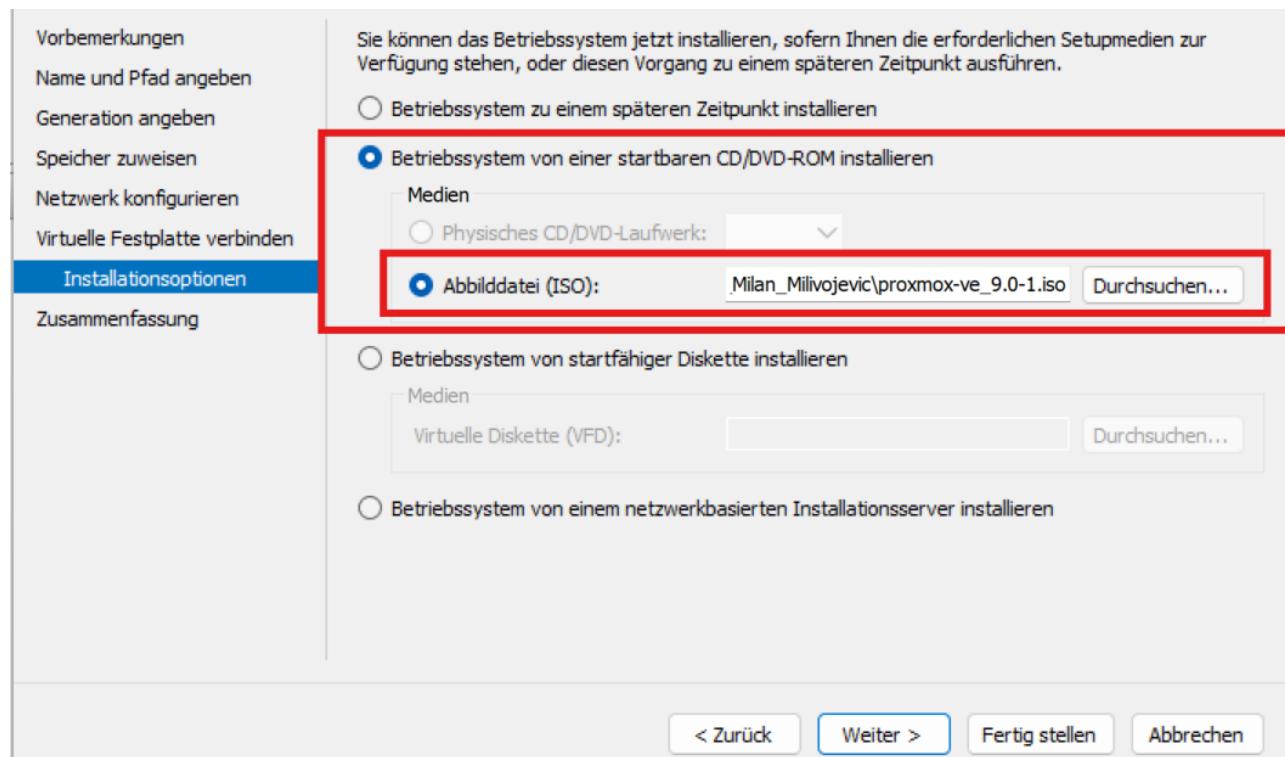


Abbildung 11 VM-ISO-File

Sie sehen eine **kurze Zusammenfassung**, was alles gemacht wurde. **Kontrollieren Sie**, ob alles so ist wie in der **Anleitung** beschrieben!

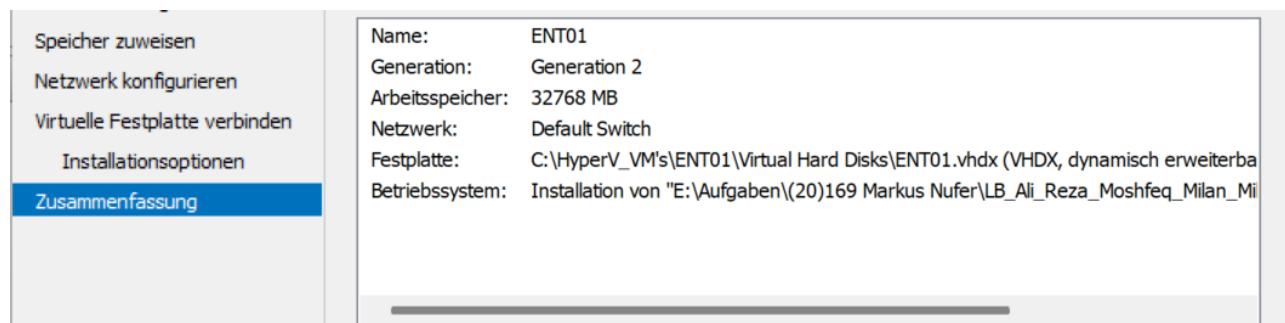


Abbildung 12 VM-Zusammenfassung

Wenn Sie alles kontrolliert haben, **drücken Sie auf Fertigstellen** und die **VM ist erstellt**.

Somit kommen wir zum nächsten Schritt:

**Betriebssystem installieren**

## 4.4. Betriebssystem installieren

Damit wir unser Betriebssystem Fehlerfrei installieren können müssen wir den Modus

**Sicherer Start deaktivieren.** Gehen Sie in die **Einstellungen der VM**.



Abbildung 13 VM-Einstellungen

Sie finden unter **Sicherheit** die Option den **Sichereren Start zu deaktivieren**, siehe Bild.

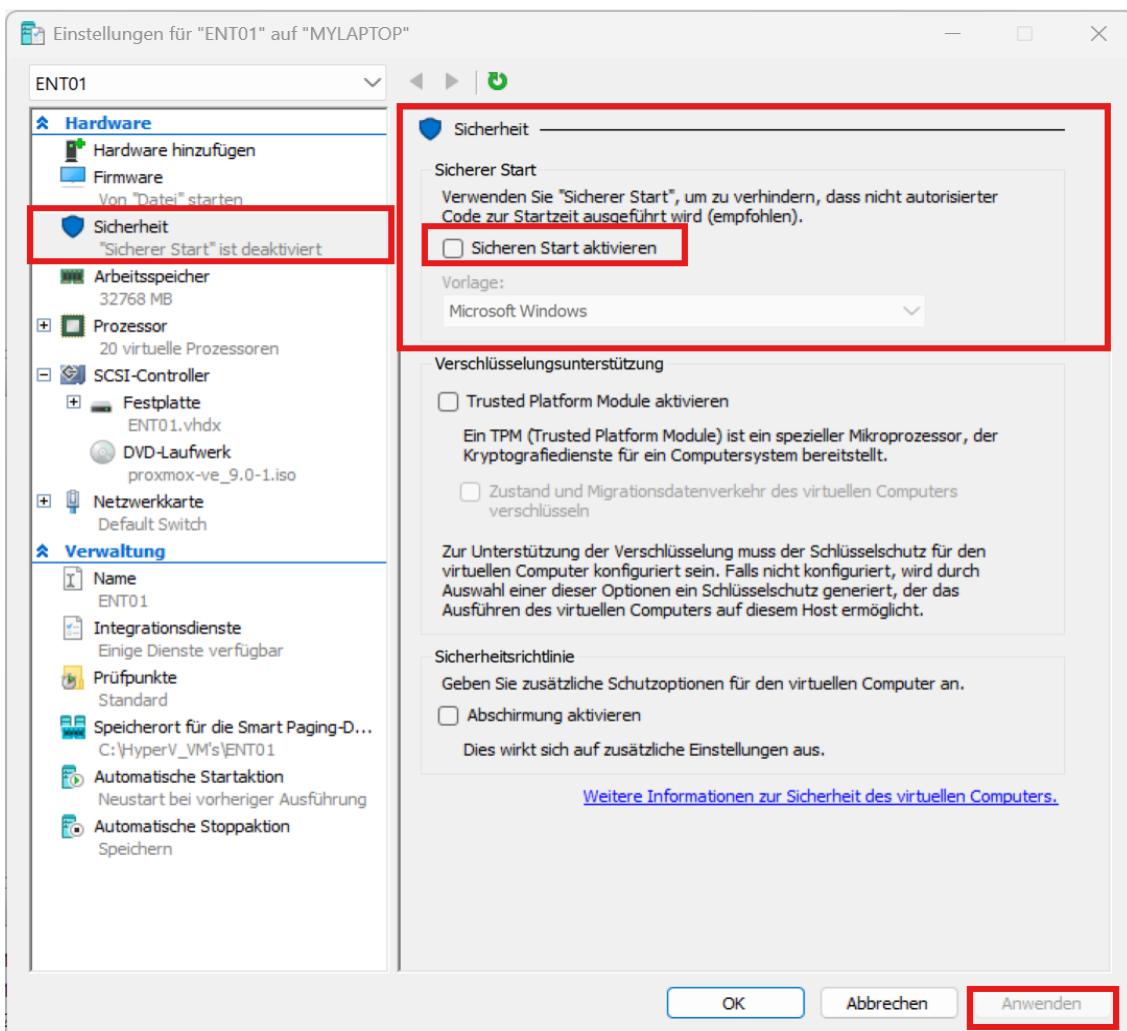
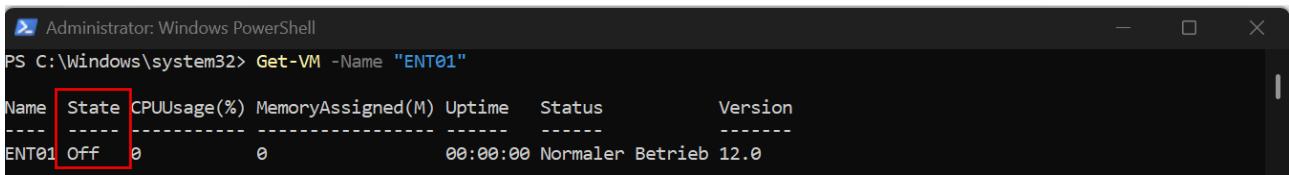


Abbildung 14 VM-Sicherheit

**Drücken Sie als nächstes auf Anwenden**

Öffnen Sie auf Ihrem Host-Gerät das Programm PowerShell als Administrator, führen Sie dort den Folgenden Befehl aus:



```
Administrator: Windows PowerShell
PS C:\Windows\system32> Get-VM -Name "ENT01"
Name      State    CPUUsage(%) MemoryAssigned(M) Uptime          Status      Version
----      ----    (%)           (M)            (days)        (hours)   (minutes)
ENT01    off          0             0       00:00:00 Normaler Betrieb 12.0
```

Abbildung 15 VM-Status Check

**Falls im rot markierten Bereich (On) steht, dann müssen Sie den folgenden Befehl ausführen:**

```
PS C:\Windows\system32> Stop-VM -Name "ENT01"
```

Ansonsten können Sie mit den nächsten Befehlen fortfahren.

Aktivieren wir ExposeVirtualizationExtensions auf unserer VM mit folgenden Befehlen:

```
PS C:\Windows\system32> Get-VMProcessor -VMName "ENT01" | Select-Object ExposeVirtualizationExtensions
ExposeVirtualizationExtensions
-----
False

PS C:\Windows\system32> Set-VMProcessor -VMName "ENT01" -ExposeVirtualizationExtensions $true
PS C:\Windows\system32> Get-VMProcessor -VMName "ENT01" | Select-Object ExposeVirtualizationExtensions
ExposeVirtualizationExtensions
-----
True

PS C:\Windows\system32>
```

Abbildung 16 ExposeVirtualizationExtensions

Wenn Sie alle Befehle ausgeführt haben können Sie die **PowerShell schliessen** und zurück in den **Hyper-V-Manager**. Verbinden Sie sich mit der **VM**, um sie zu starten:

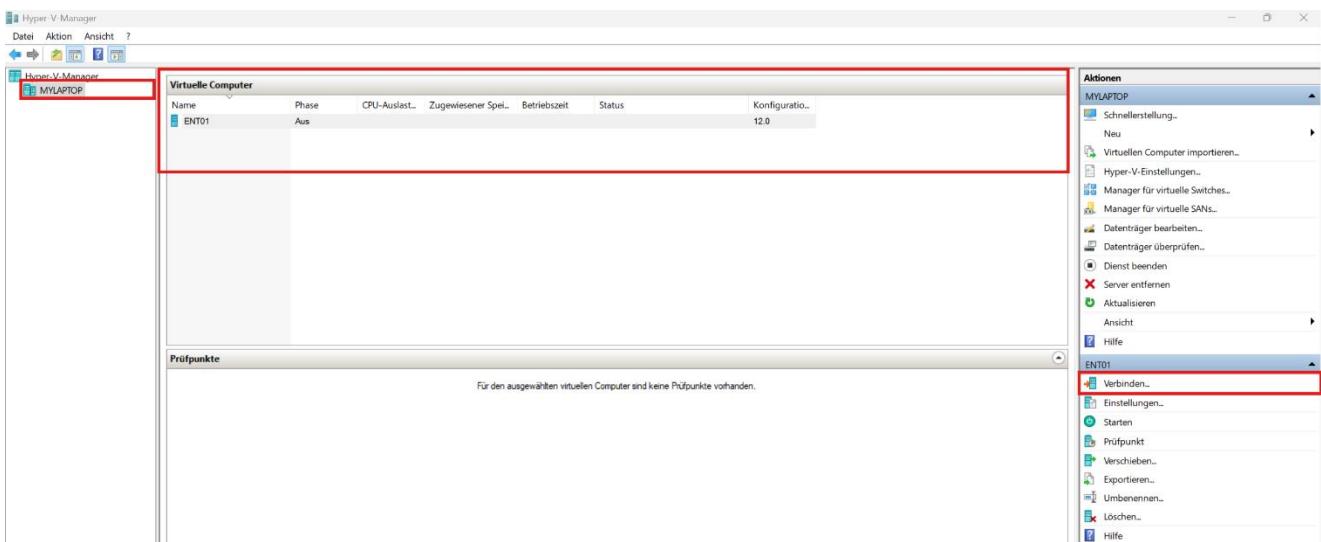


Abbildung 17 mit VM Verbinden

Wenn Sie auf Verbinden gedrückt haben wird folgendes **Fenster geöffnet. Drücken Sie auf Starten**, um die Installation von Proxmox zu starten.

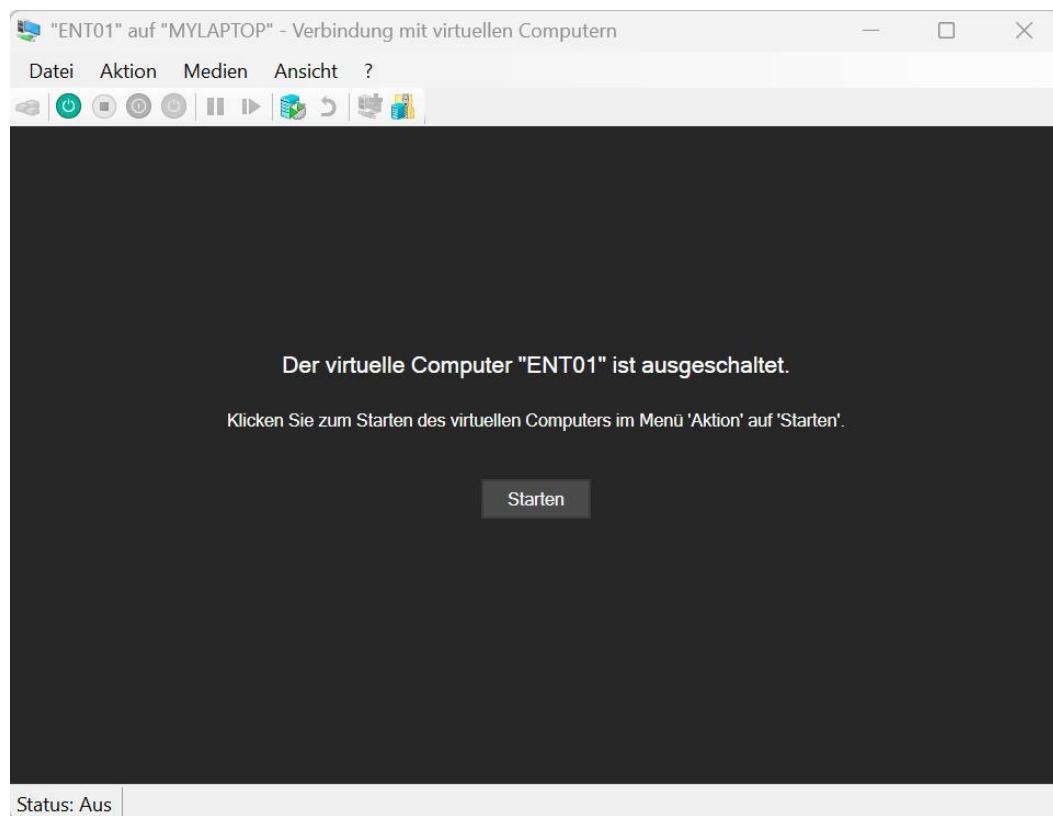


Abbildung 18 VM-Starten

Proxmox ist ein Betriebssystem, das auf Linux Debian basiert. **Gehen Sie** mit den Pfeiltasten wie **auf** dem Bild markiert auf **(Graphical)** und drücken **Enter**.

Warten Sie auf die **Erfolgreiche Installation**. Wenn diese beendet ist, **Akzeptieren** Sie die **Lizenz** und gehen **Weiter>**



Welcome to Proxmox Virtual Environment



Abbildung 19 Proxmox Installation

Als nächstes müssen wir Proxmox einen **Speicher zuweisen**, diesen haben wir vorhin beim Erstellen der VM eingerichtet. Jetzt müssen Sie diesen **Speicher auf der dropdown liste wählen** und auf **Next** gehen

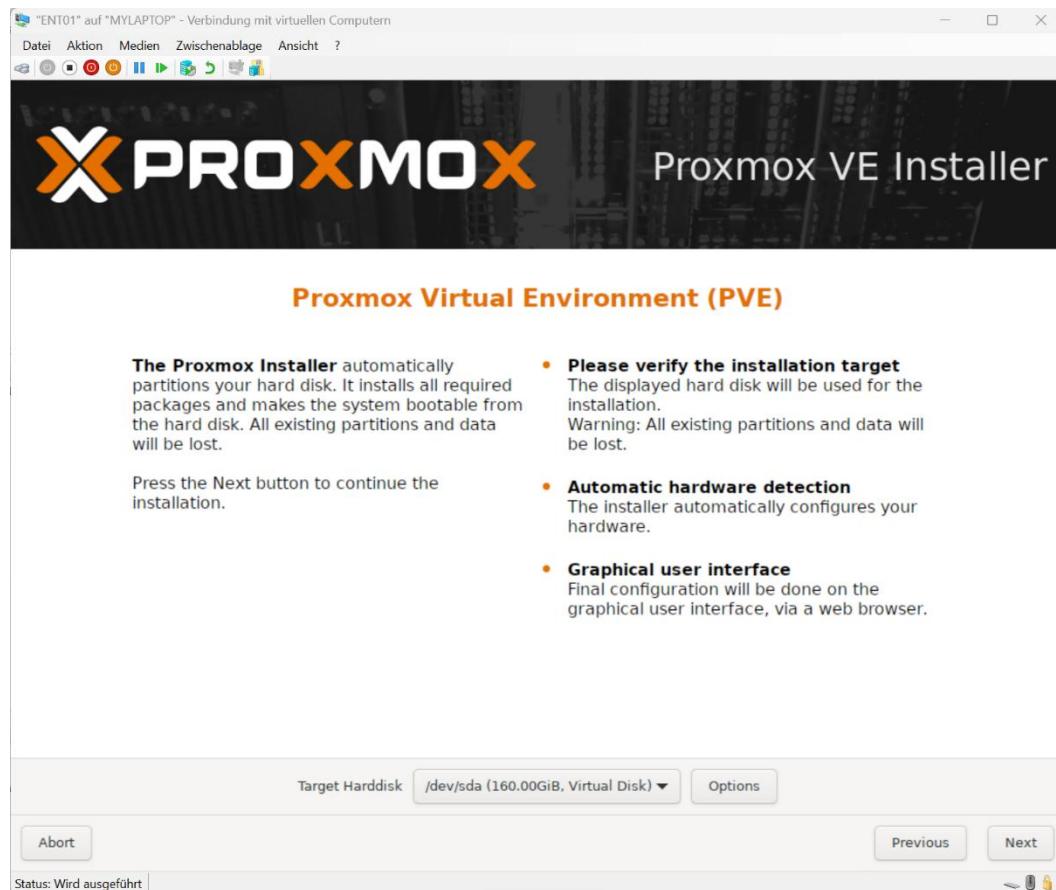


Abbildung 20 Proxmox-Speicher

Geben Sie auf der **nächsten Seite das Land, Zeitzone und Ihr Keyboard-Layout** an und gehen auf **Next**.

Richten Sie ein **Passwort und E-Mail** ein, Das Passwort wird für den Root User benutzt.

Merken Sie sich Ihr Passwort, dieses wird später verwendet, um sich in Proxmox einzuloggen!

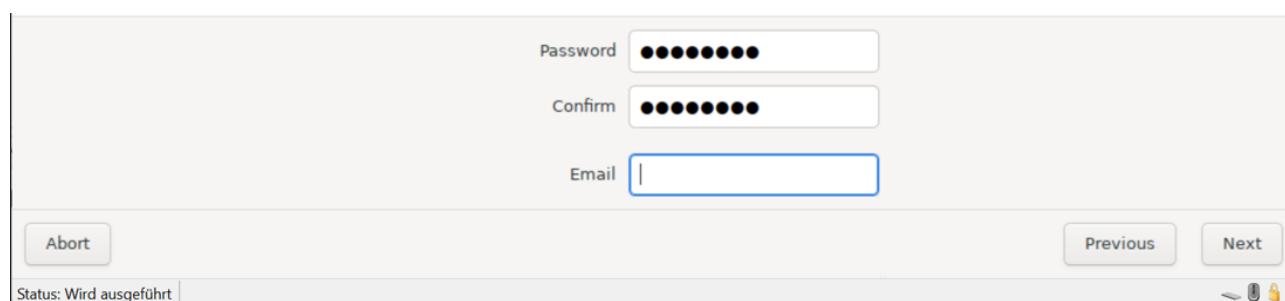


Abbildung 21 Root-Passwort

Gehen Sie auf **Next**

Da wir später effektiv die Netzwerk Einstellungen verändern werden. Können wir hier alles so wie im Bild lassen.

Nehmen Sie die **Einstellungen wie im Bild** vor und gehen Sie auf **Next**

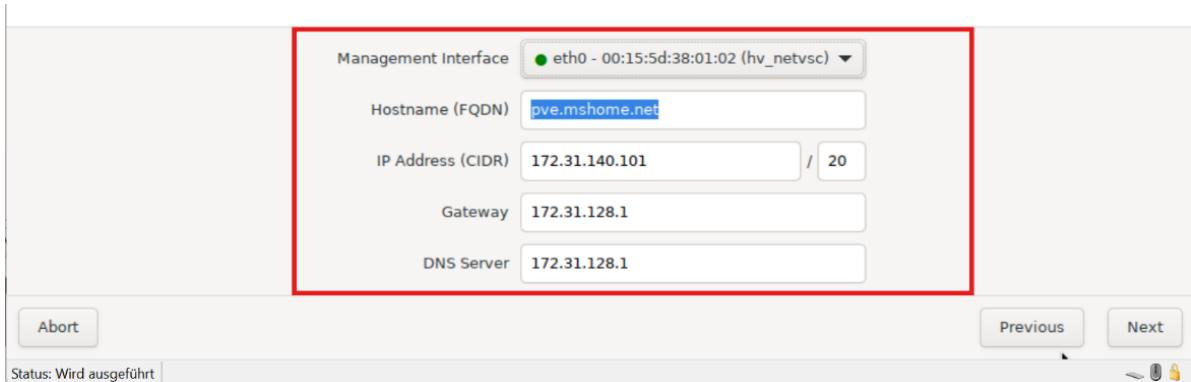


Abbildung 22 Proxmox-Network

Dann sehen Sie die Zusammenfassung der Ausgewählten Optionen. Somit haben wir alles bereitgestellt und können das **VM-Installieren**. Drücken Sie auf **Install**

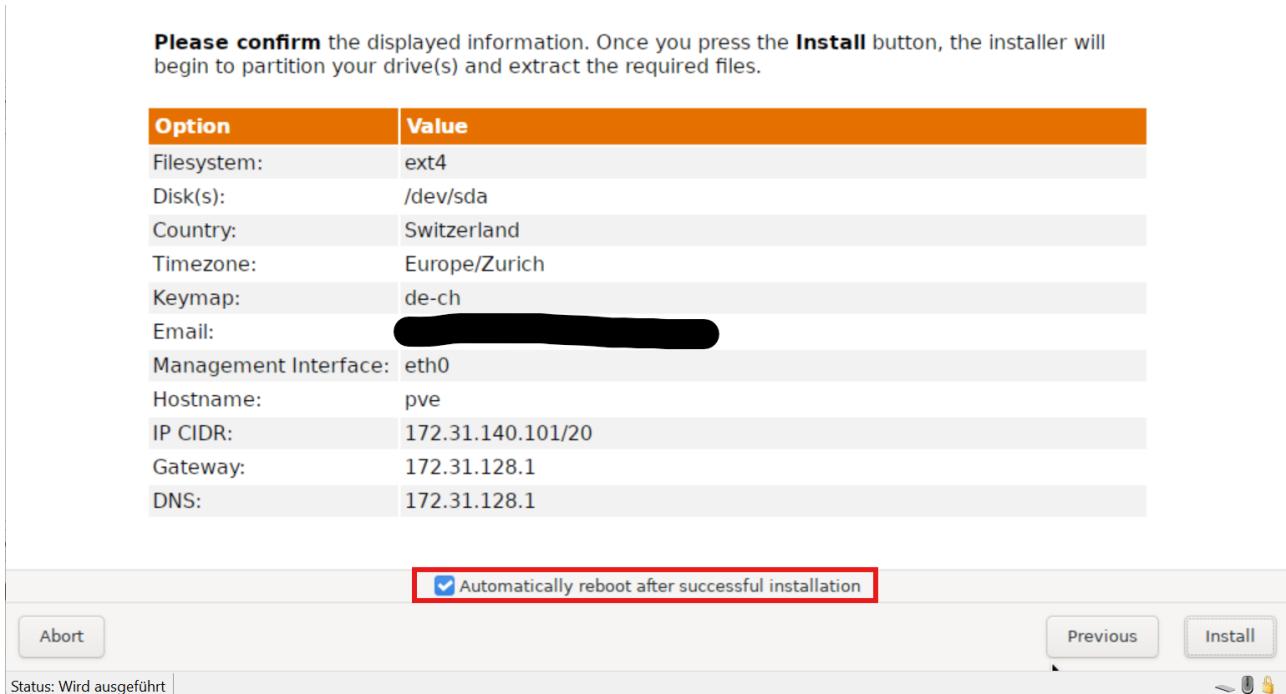


Abbildung 23 Zusammenfassung-Proxmox

Nachdem Die VM installiert wurde, wird das System automatisch ein Reboot ausführen. Also die VM-Neustarten. Achtung das geschieht schnell!

Hier sehen Sie das Bild nach der Installation dieses hat einen Timer wegen dem Automatic Reboot das wir eingestellt haben. Schauen Sie das Sie ein Screenshot von diesem Fenster machen damit Sie sich die IP merken können.

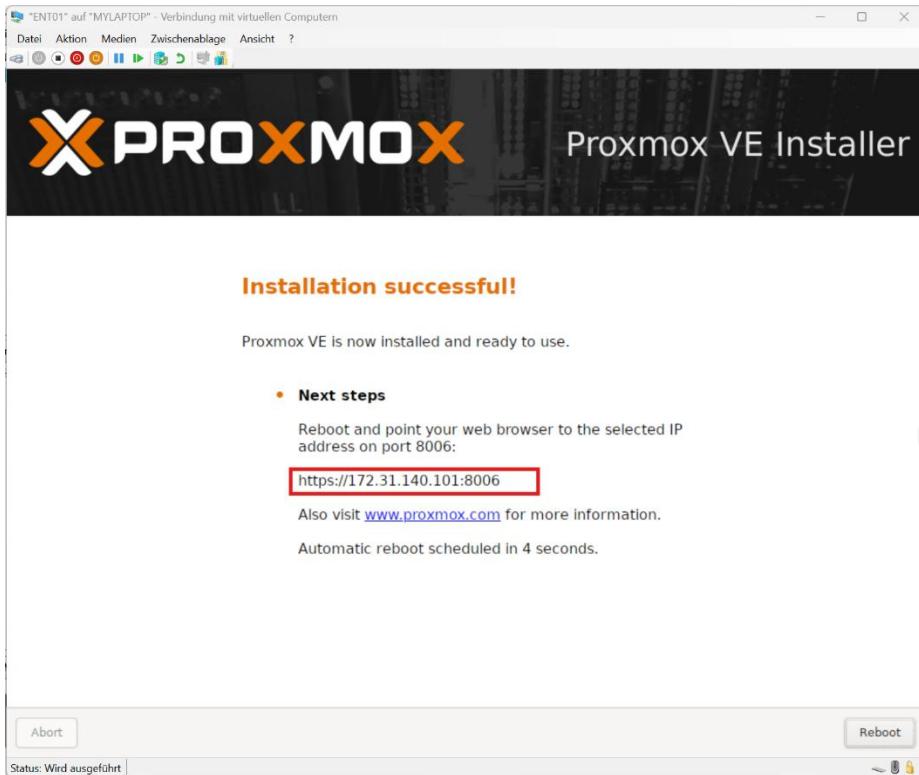


Abbildung 24 Proxmox-Reboot

Nach dem Reboot werden Sie folgendes Fenster sehen:

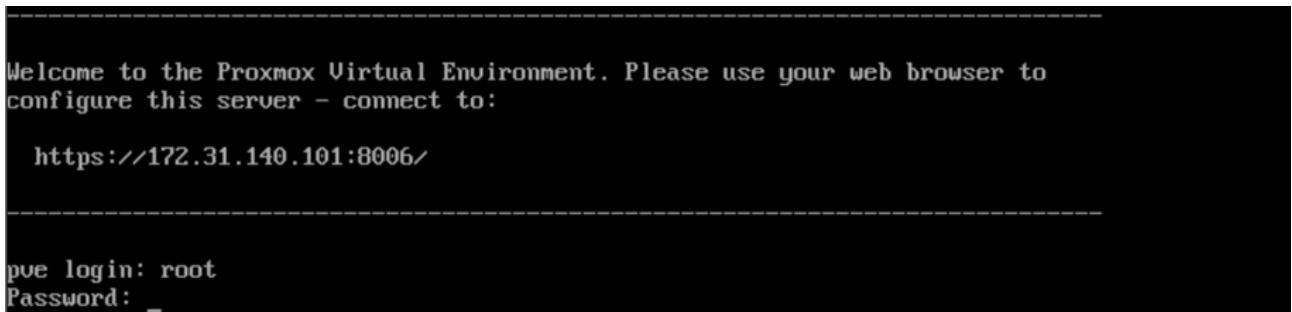


Abbildung 25 Proxmox-Terminal-Login

Das ist das Terminal der Proxmox VM, hier können Sie beliebige Befehle ausführen.

Als erstes müssen Sie sich einloggen. Wie vorhin erwähnt das System erstellt automatisch den Benutzer <root>. Diesen geben Sie bei **pve login** ein und drücken **Enter**.

Dann werden Sie aufgefordert das vorhin definierte **Passwort einzugeben**. Dann Enter.

**! HINWEIS !** bei Linux Systemen wird **nicht angezeigt** was oder wie viel Sie getippt haben!

#### 4.4.1. Proxmox GUI über Webseite

Damit wir einen besseren Überblick erhalten, über die Optionen die Proxmox bietet, können wir uns mit der IP, die wir vorhin im Terminal gesehen haben, über einen Browser verbinden.

Geben Sie diese **IP**, die Sie im Terminal oder beim Setup gesehen haben, in die **Suchleiste** Ihres **Browsers** ein.

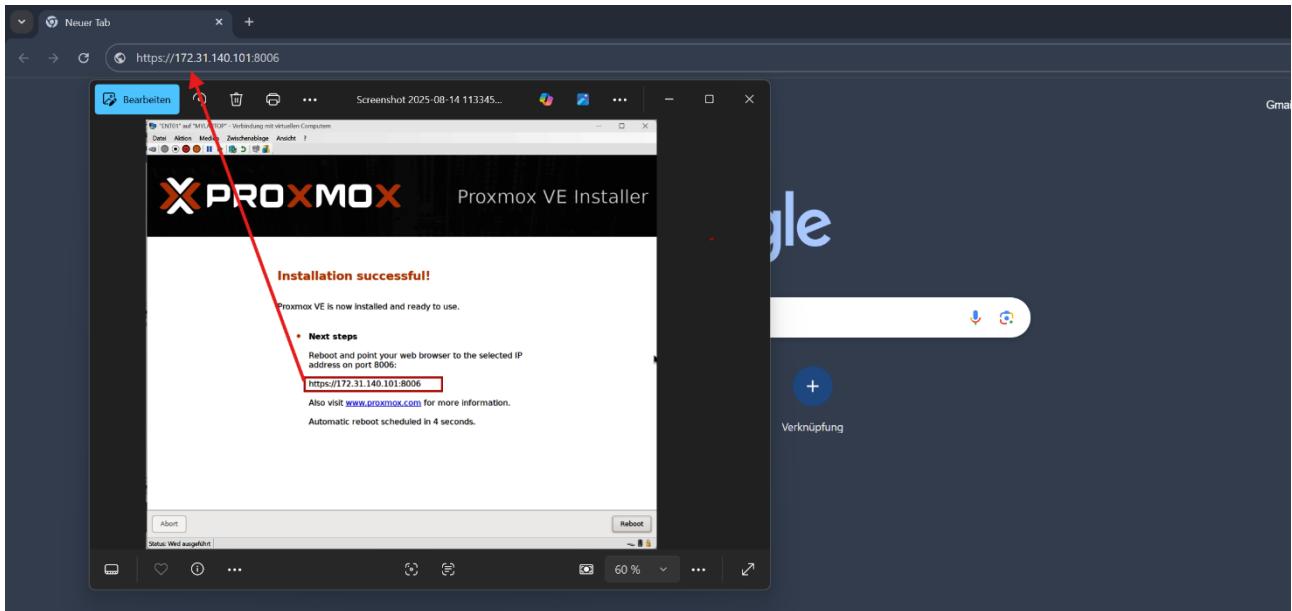


Abbildung 26 Proxmox auf Browser

Sie werden folgende Warnung erhalten, dies ist normal da Sie über **kein Sicherheitszertifikat** für diese Proxmox VM verfügen.

Gehen Sie auf **Eweitert** und dann auf **Weiter zu <IHRE IP> (unsicher)**, wie im Bild zu sehen.

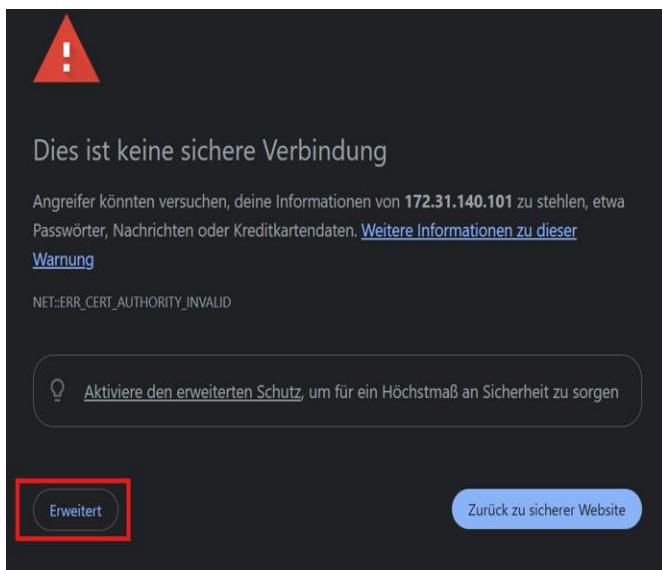


Abbildung 28 Browser-Sicherheitswarnung

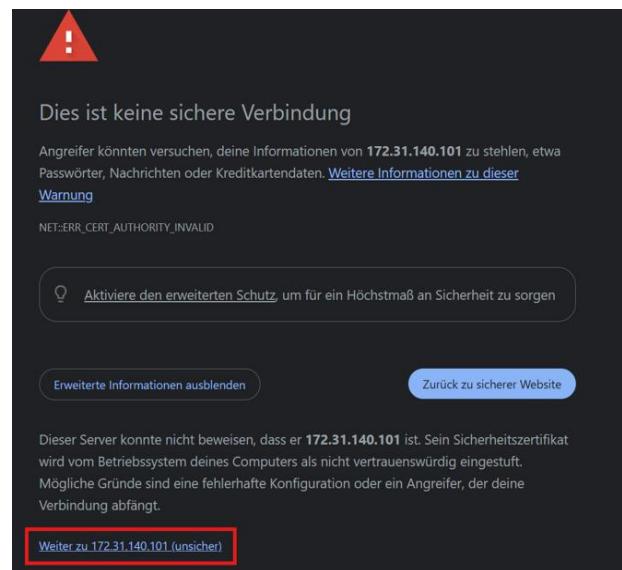


Abbildung 27 Sicherheitswarnung-Umgehen

Sie haben die Webseite mit dem GUI erreicht. Sie werden aufgefordert sich Anzumelden, es gelten die gleichen Bedingungen. **Root** als **User** und **das gleiche Passwort** wie im **Terminal**. Wenn Sie alles eingegeben haben, drücken Sie auf **Login**

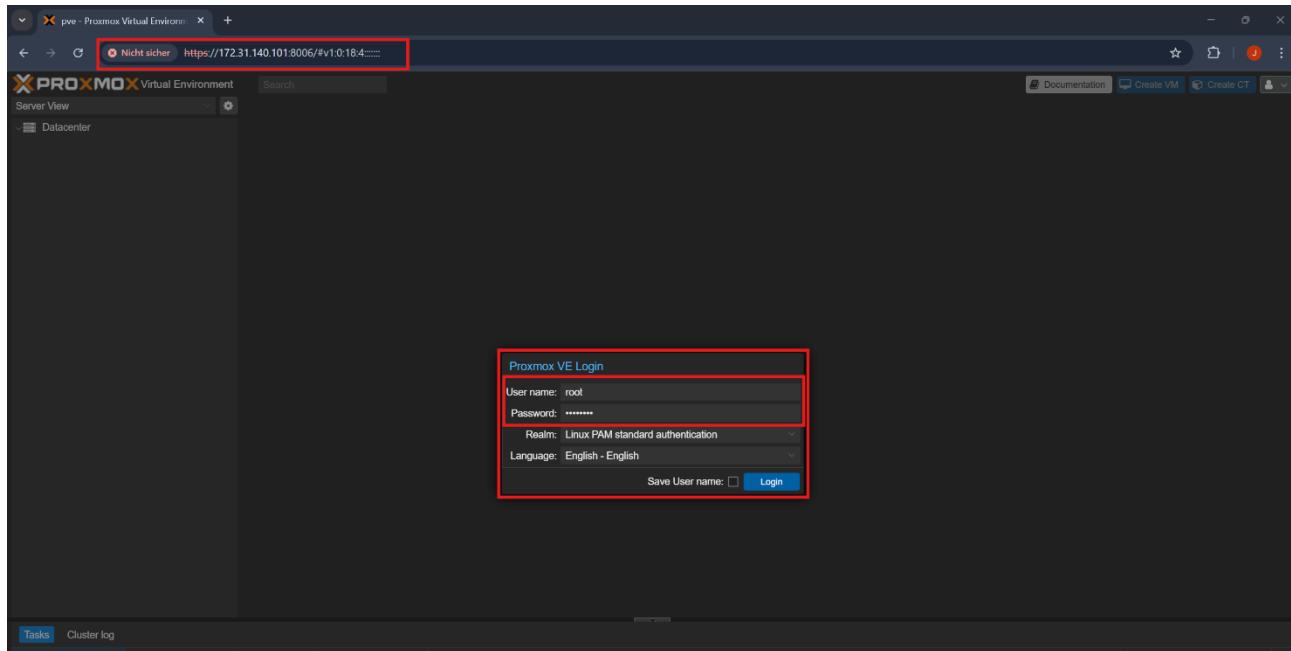


Abbildung 29 ProxmoxWeb-Login

Jetzt können Sie alle Optionen von Proxmox sehen. Sie werden **mehrmals hingewiesen** das Sie **keine Subscription** haben, bestätigen Sie jedes Mal mit **OK**

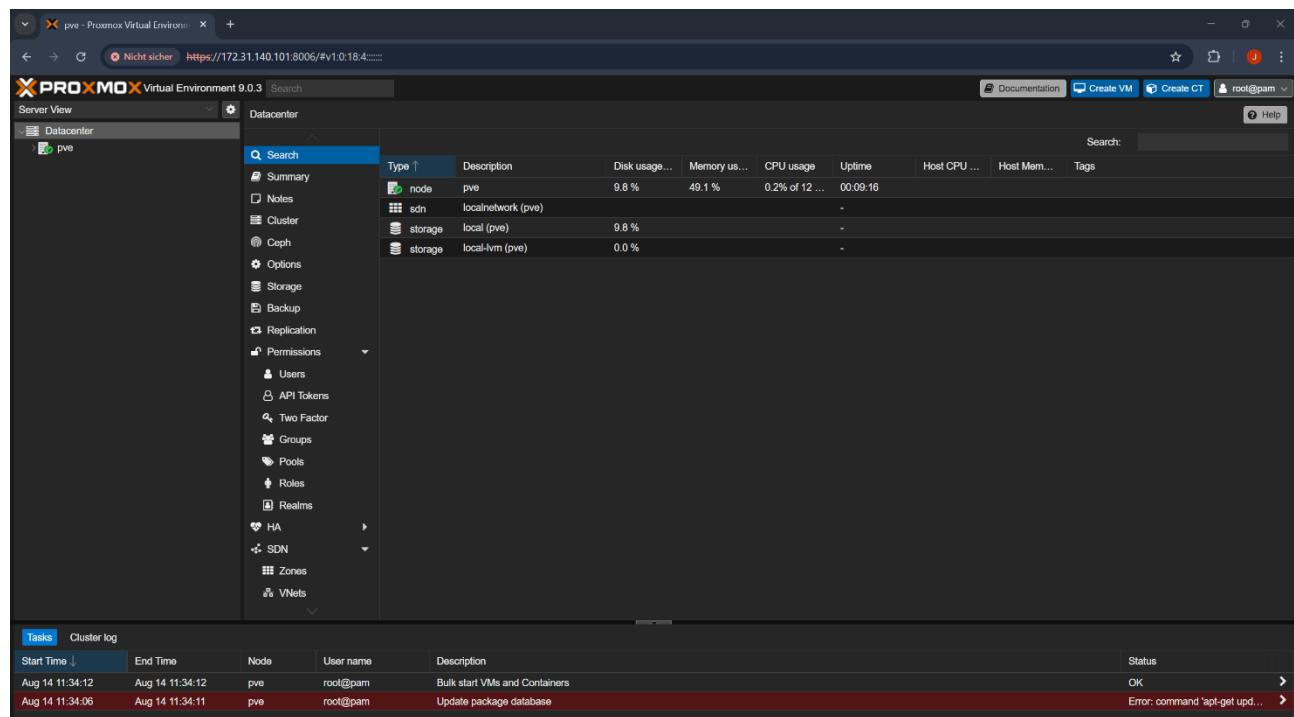


Abbildung 30 Proxmox-GUI

Folgen Sie den nächsten schritten auf dem Bild, um Ihre Proxmox Repository einzurichten.

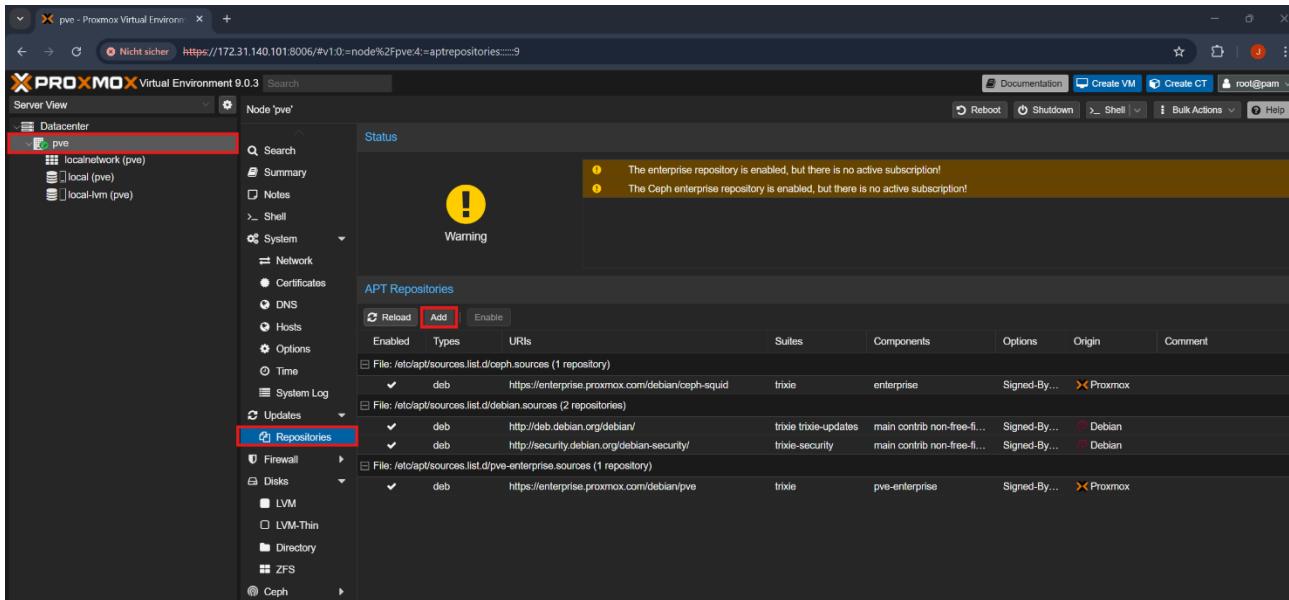


Abbildung 31 Proxmox-Repositories

**Wählen** wir die Repository **No-Subscription** damit wir keine Zahlungen tätigen müssen.  
Integrieren wir die Repository mit **Add**.

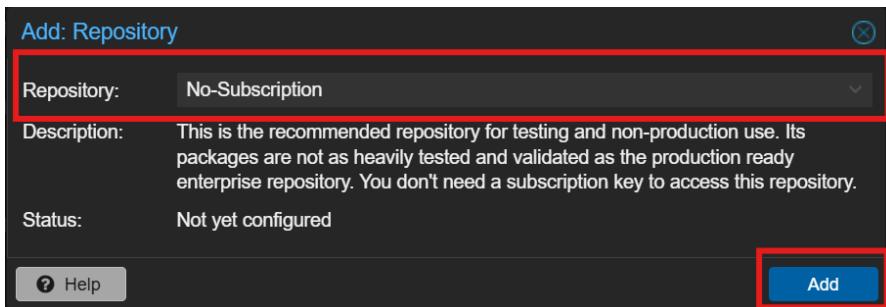


Abbildung 32 Repository-No-Subscription

Gehen Sie auf **pve** dann **Updates** und **Refresh**, somit schaut das System welche Updates verfügbar sind. Es wird ein **Task Viewer Fenster** mit Informationen **geöffnet** dieses können Sie **Schliessen**.

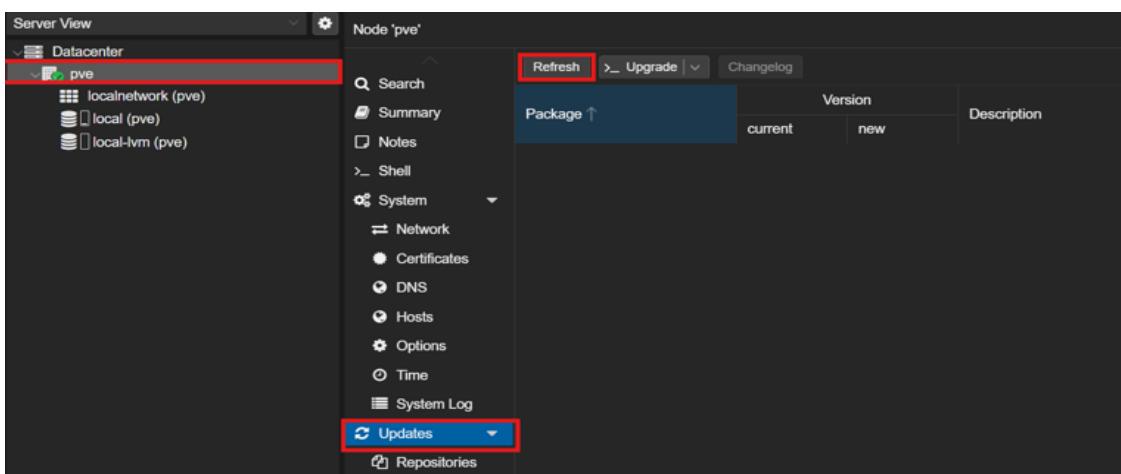


Abbildung 33 Proxmox-Updates

Wenn Sie das Fenster geschlossen haben werden Sie sehen das es mehrere Pakete gefunden hat, die ein Update brauchen.

Gehen Sie **wie im Bild** markiert auf **Upgrade**.

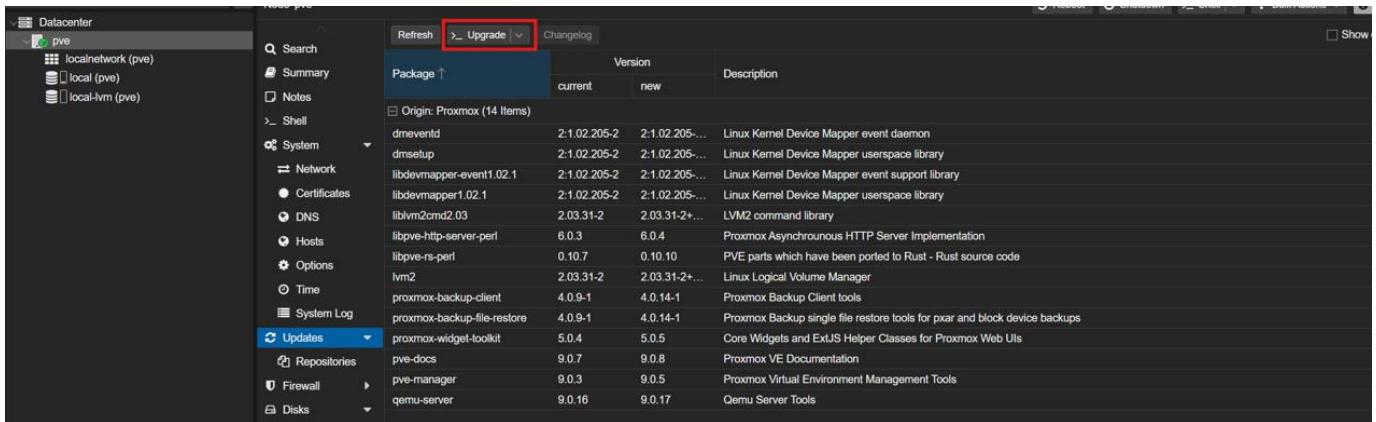


Abbildung 34 Proxmox-Upgrade

Das **Terminal** wird **automatisch geöffnet** und Sie werden gefragt, ob Sie sicher sind das Sie die Updates ausführen möchten, schreiben Sie **y**

und drücken danach **Enter**

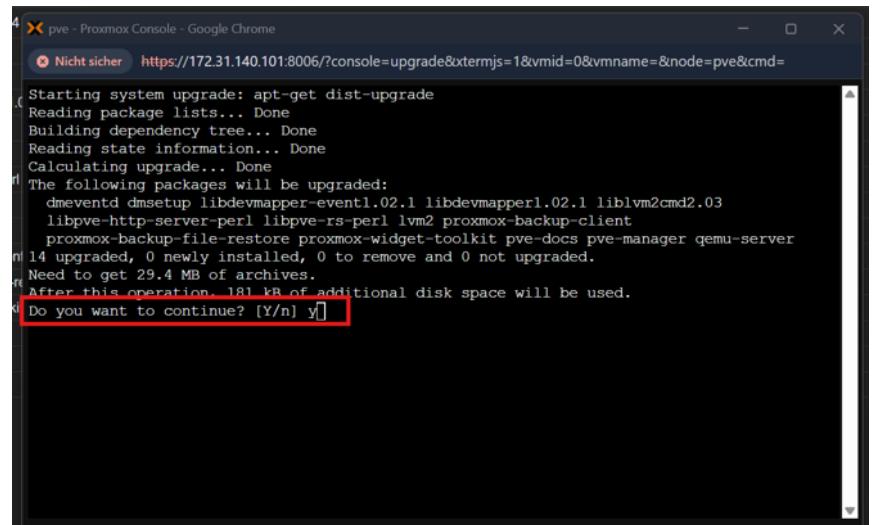


Abbildung 35 Upgrade-Bestätigen

Wenn die **Updates fertig** sind, können Sie das **Terminal schliessen**. Wenn Sie das Terminal geschlossen haben, navigieren Sie zurück zu **Repositories** und drücken Sie auf **Reboot**.

**Pve → Repositories → Reboot**

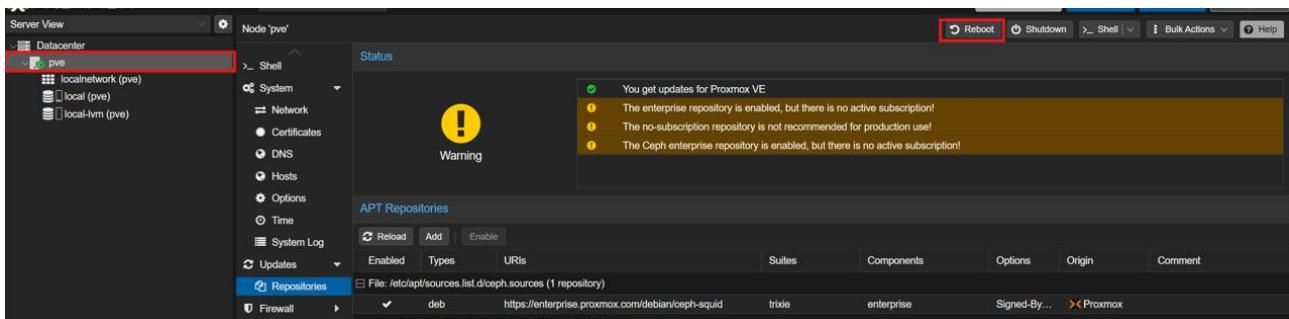


Abbildung 36 Update-Reboot

Somit ist unsere Proxmox VM installiert und bereit für die Netzwerk Konfigurationen.

## 4.5. Netzwerk Konfigurieren

Damit wir immer eine statische IP-Adresse für unsere Proxmox GUI Website haben, aber trotzdem Internet auf den Containern und der VM, muss als nächstes unser Netzwerk konfiguriert werden. Wir werden die IPs aus dem IP-Konzept verwenden.

### Automatisch Virtuelle Switch Adapter erstellen

Öffnen Sie auf Ihrem Host-System **Windows PowerShell** als **Administrator**.

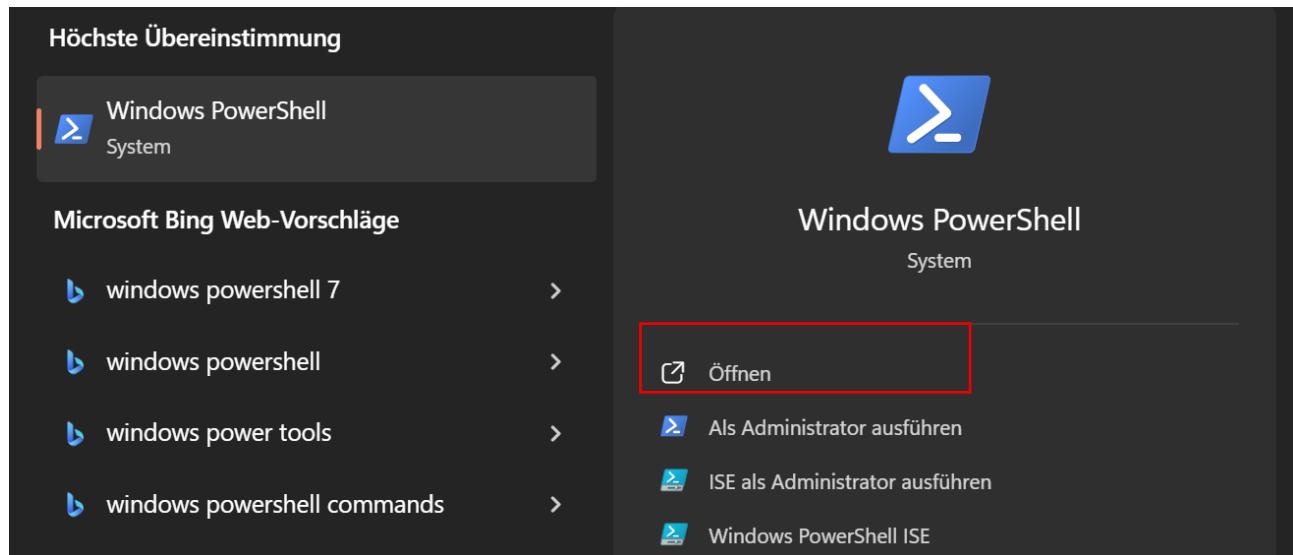


Abbildung 37 Powershell-Administrator

Führen Sie folgende Codes in dieser Reihenfolge aus um einen neuen Switch sowie Netzwerkadapter zu erstellen:

```
PS C:\Windows\system32> New-VMSwitch -SwitchName "LabInternal" -SwitchType Internal
Name      SwitchType NetAdapterInterfaceDescription
----      ----- -----
LabInternal Internal
```

Abbildung 38 Neue-Switch

Schauen, ob Switch erstellt.

```
PS C:\Windows\system32> Get-NetAdapter -Name "vEthernet (LabInternal)"
Name           InterfaceDescription           ifIndex Status     MacAddress      LinkSpeed
----           -----                   -----   -----   -----          -----
vEthernet (LabInternal) Hyper-V Virtual Ethernet Adapter #2       66 Up      00-15-5D-38-01-06    10 Gbps
```

Abbildung 39 Neue-Adapter

Zugeordnete IP-Adressen für neue Switch nachschauen

```
PS C:\Windows\system32> Get-NetIPAddress -InterfaceAlias "vEthernet (LabInternal)" | ft IPAddress,PrefixLength,AddressFamily,Type
IPAddress      PrefixLength AddressFamily     Type
-----          -----        -----          -----
fe80::b06f:6f45:54c6:6a1e%66          64        IPv6 Unicast
169.254.63.107                         16        IPv4 Unicast
```

Abbildung 40 IP-von-Adapter

## Zugeordnete IP-Adresse löschen

```
PS C:\Windows\system32> Remove-NetIPAddress -InterfaceAlias "vEthernet (LabInternal)" -IPAddress 169.254.63.107 -Confirm:$false
PS C:\Windows\system32> Get-NetIPAddress -InterfaceAlias "vEthernet (LabInternal)" | ft IPAddress,PrefixLength,AddressFamily,Type

```

IPAddress	PrefixLength	AddressFamily	Type
fe80::b06f:6f45:54c6:6a1e%66	64	IPv6	Unicast

Abbildung 41 Zugeordnete-IP-Löschen

## Neue IP-Adresse definieren und zuweisen

```
PS C:\Windows\system32> New-NetIPAddress -InterfaceAlias "vEthernet (LabInternal)" -IPAddress 172.16.100.1 -PrefixLength 24

```

IPAddress : 172.16.100.1	InterfaceIndex : 66	InterfaceAlias : vEthernet (LabInternal)	AddressFamily : IPv4	Type : Unicast
PrefixLength : 24	PrefixOrigin : Manual	SuffixOrigin : Manual	AddressState : Tentative	ValidLifetime :
PreferredLifetime :	SkipAsSource : False	PolicyStore : ActiveStore		
IPAddress : 172.16.100.1	InterfaceIndex : 66	InterfaceAlias : vEthernet (LabInternal)	AddressFamily : IPv4	Type : Unicast
PrefixLength : 24	PrefixOrigin : Manual	SuffixOrigin : Manual	AddressState : Invalid	ValidLifetime :
PreferredLifetime :	SkipAsSource : False	PolicyStore : PersistentStore		

Abbildung 42 Neue-IP-Zuweisen

## Nachschauen ob neue IP zugewiesen

```
PS C:\Windows\system32> Get-NetIPAddress -InterfaceAlias "vEthernet (LabInternal)" | ft IPAddress,PrefixLength,AddressFamily,Type

```

IPAddress	PrefixLength	AddressFamily	Type
fe80::b06f:6f45:54c6:6a1e%66	64	IPv6	Unicast
172.16.100.1	24	IPv4	Unicast

Abbildung 43 Adapter-IP-Kontrollieren

## Neues NAT für Adapter

```
PS C:\Windows\system32> New-NetNat -Name "LabNAT" -InternalIPInterfaceAddressPrefix 172.16.100.0/24

```

Name : LabNAT	ExternalIPInterfaceAddressPrefix :
InternalIPInterfaceAddressPrefix : 172.16.100.0/24	IcmpQueryTimeout : 30
TcpEstablishedConnectionTimeout : 1800	TcpTransientConnectionTimeout : 120
TcpFilteringBehavior : AddressDependentFiltering	UdpFilteringBehavior : AddressDependentFiltering
UdpIdleSessionTimeout : 120	UdpInboundRefresh : False
Store : Local	Active : True

Abbildung 44 Neue-NAT

Wenn Sie alle Befehle der Reihenfolge nach ausgeführt haben. Dann haben Sie auf Ihrem System einen Virtuellen Netzwerkadapter für den Hyper-V-Manager erstellt und konfiguriert.

Als nächstes öffnen wir den **Hyper-V-Manager** und gehen in die **Einstellungen** unserer VM.

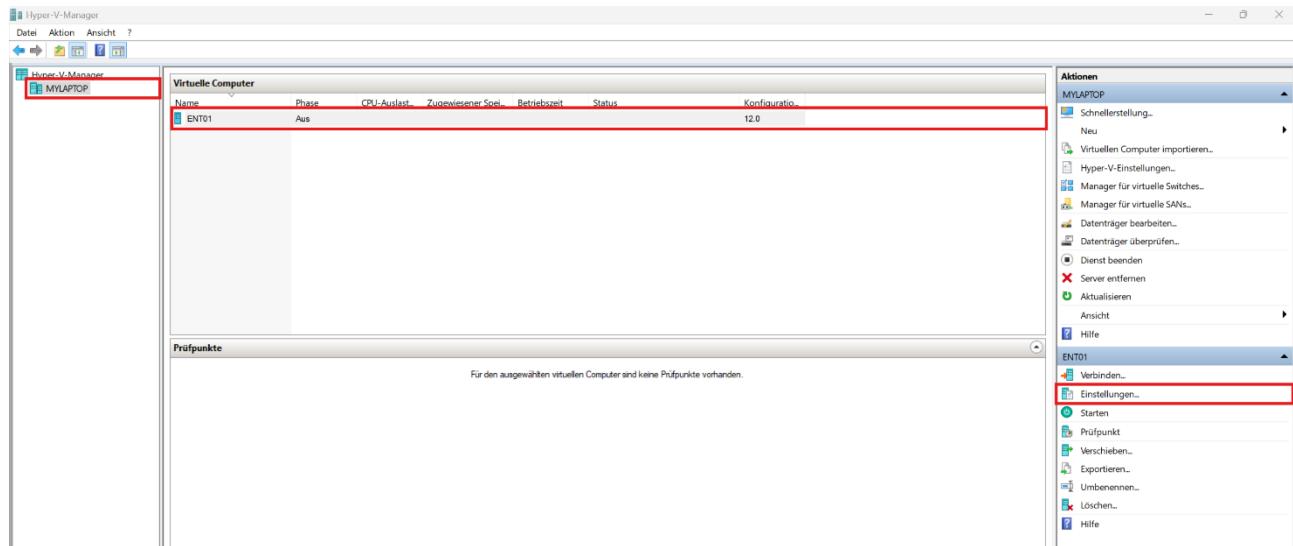


Abbildung 45 Hyper-V-Netzwerkkarte-Einstellungen

In den **Einstellungen** gehen wir unter **Netzwerkkarte** auf **Virtueller Switch** und wählen dort die von uns erstellte **Switch** und gehen auf **Anwenden**.

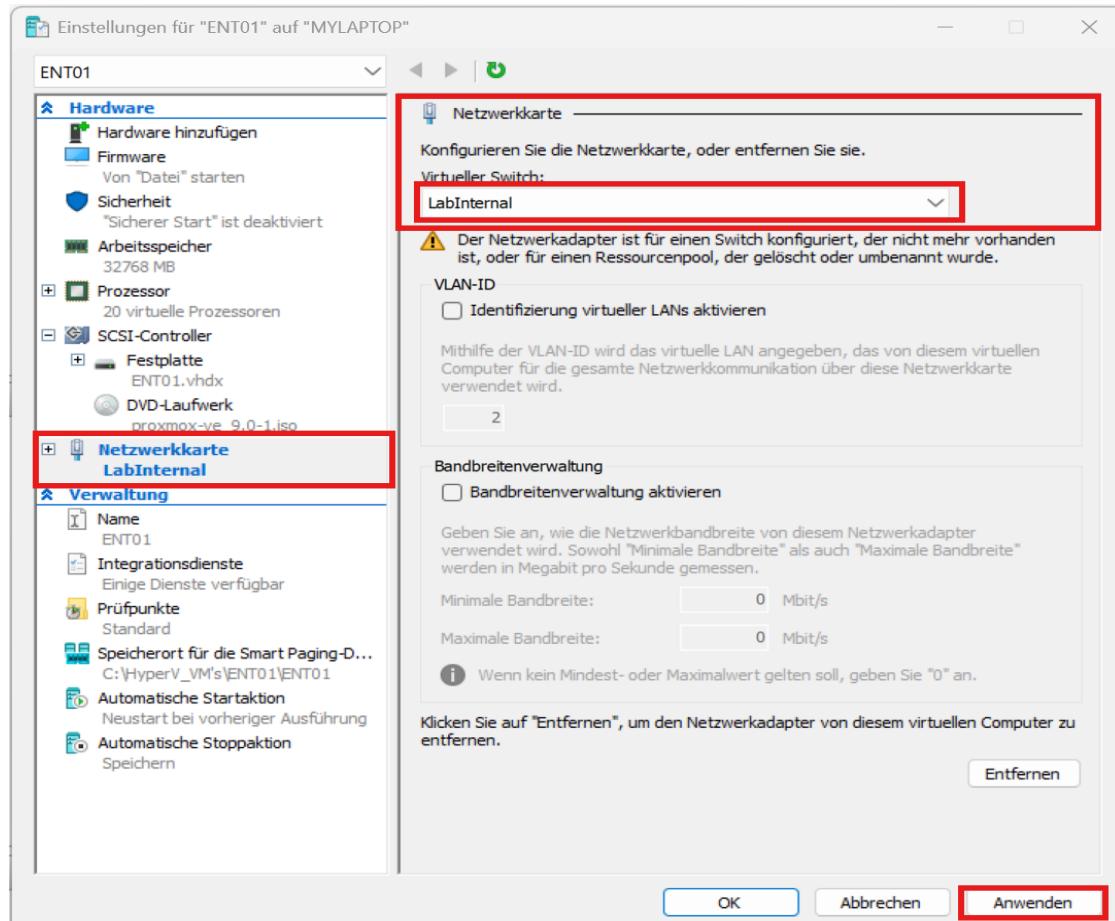


Abbildung 46 Switch-Umstellen

Starten wir unsere Proxmox VM und vollenden dort noch die Konfiguration des Netzwerks.

Wenn Sie die **VM gestartet** haben können Sie sich wieder als **root anmelden** und folgenden **Befehl ausführen**, um die Konfigurationsdatei für Netzwerkadapter zu öffnen.

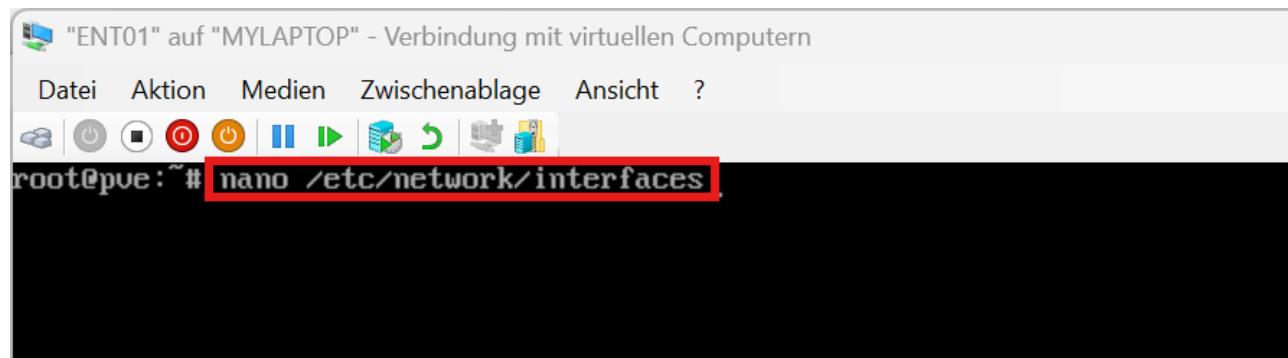


Abbildung 47 Konfigurationsdatei-Proxmox

In dieser Konfigurationsdatei müssen Sie alles **so ändern** das es **aussieht** wie auf dem **Bild**. Wenn Sie alles **geändert haben**, können Sie **folgende Befehle ausführen**, um die **Datei zu speichern** und zu **schliessen**.

**Ctrl + o** → **Enter** → **CTRL + x**

```
GNU nano 8.4                               /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto umbr0
iface umbr0 inet static
    address 172.16.100.10/24
    gateway 172.16.100.1
    bridge-ports eth0
    bridge-stp off
    bridge-fd 0
    dns-nameservers 1.1.1.1 9.9.9.9 172.16.100.1
    dns-search lab.local

source /etc/network/interfaces.d/*
```

Abbildung 48 Konfigurationsdatei-Setup

Machen Sie ein **Neustart** der **Netzwerk Einstellungen** und **reboot** der VM.

```
root@pve: ~# systemctl restart networking
root@pve: ~# reboot
```

Abbildung 49 Netzwerk-Neustart

Nachdem die **VM neugestartet** hat, fahren wir weiter mit der nächsten Datei. Öffnen Sie mit folgendem Befehl die resolv.conf Datei, um die Name Server einzurichten. Füllen Sie die Datei wie auf dem Bild zu sehen aus und Speichern Sie diese wie vorhin erwähnt mit:

**Ctrl + o → Enter → CTRL + x**

```
GNU nano 8.4
search lab.local mshome.net
nameserver 1.1.1.1
nameserver 9.9.9.9
nameserver 172.16.100.1
```

Abbildung 50 Nameserver-Datei

Führen Sie jetzt **folgende Befehle** aus, um zu testen, ob Sie wirklich die definierte IP bekommen und verwenden können.

**ip a**  
**ip r**  
**ping -c2 1.1.1.1**

Wenn Sie die definierte IP (**172.16.100.10**) auf dem **vmbr0 Adapter** bekommen haben Sie alles **richtig eingestellt**.

```
root@pve:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master vmbr0 state UP group default qlen 1000
    link/ether 00:15:5d:38:01:03 brd ff:ff:ff:ff:ff:ff
    altname enx00155d380103
3: vmbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:15:5d:38:01:03 brd ff:ff:ff:ff:ff:ff
        inet 172.16.100.10/24 brd 172.16.100.255 scope global vmbr0
            valid_lft forever preferred_lft forever
        inet6 fe80::215:5dff:fe38:103/64 scope link proto kernel ll
            valid_lft forever preferred_lft forever
root@pve:~# ip r
default via 172.16.100.1 dev vmbr0 proto kernel onlink
172.16.100.0/24 dev vmbr0 proto kernel scope link src 172.16.100.10
root@pve:~# ping -c2 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=54 time=184 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=54 time=93.4 ms

--- 1.1.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 93.430/138.483/183.536/45.053 ms
root@pve:~# ping -c2 deb.debian.org
PING debian.map.fastlydns.net (151.101.194.132) 56(84) bytes of data.
64 bytes from 151.101.194.132: icmp_seq=1 ttl=53 time=102 ms
64 bytes from 151.101.194.132: icmp_seq=2 ttl=53 time=124 ms

--- debian.map.fastlydns.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 101.609/112.955/124.301/11.346 ms
root@pve:~#
```

Abbildung 51 IP-Check

Gehen Sie jetzt auf Ihr **Host-System** und Suchen Sie dort unter dem im Bild zu sehenden Pfad nach der **hosts datei**. Öffnen Sie die Datei in **Notepad**.

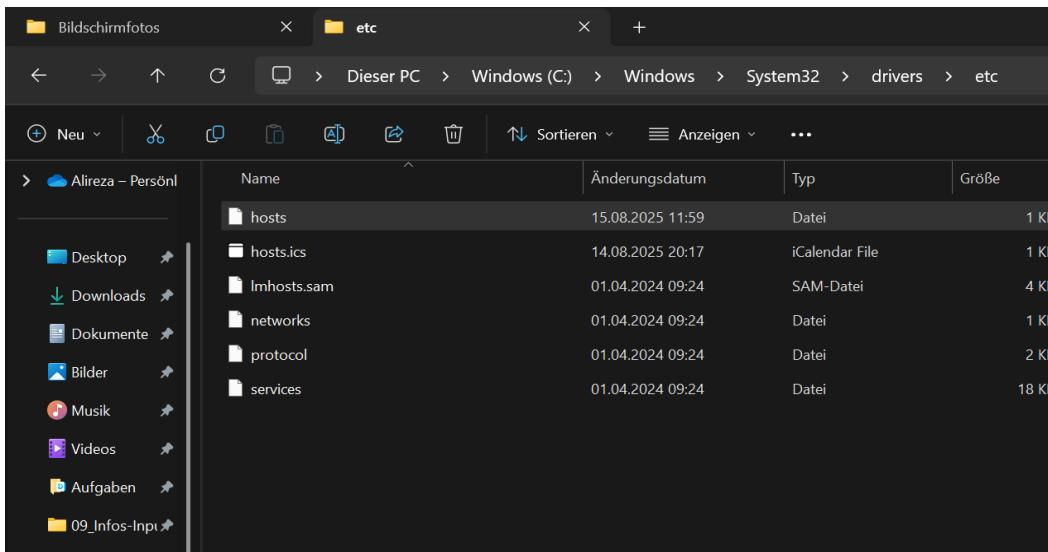


Abbildung 52 hosts-definieren

Fügen Sie nur das blau markierte in die hosts datei ein und speichern Sie diese.

```

# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#      38.25.63.10      x.acme.com            # x client host
#
# localhost name resolution is handled within DNS itself.
#      127.0.0.1        localhost
#      ::1              localhost
172.16.100.10 proxmox proxmox

```

Abbildung 53 host-einfügen

Somit haben wir die Netzwerkeinstellungen erfolgreich eingerichtet. Jetzt können wir durch diese Einrichtung auch mit <https://proxmox:8006> auf die Proxmox Webseite zugreifen sowie mit der neuen IP 172.16.100.10 die wir definiert haben.

Es gelingt uns auch die Virtuelle Maschine damit von einem anderen System aus anzusprechen und den Containern Internet zur Verfügung zu stellen.

Sie können auch durch dieses Setup unabhängig von Ihrem Standort und der Zuteilung der IP an das Hostsystem durch verschiedene Router, ohne Probleme mit der VM ins Internet kommen.

## 4.6. Container installieren

Da wir unsere Proxmox VM vollständig in Betrieb genommen haben können wir jetzt problemlos die Erstellung der Container Anfangen. [Container automatisch erstellen](#)

Gehen Sie auf die Proxmox Webseite mit einer der neuen Adressen:

**https://proxmox:8006**      oder    **https://172.16.100.10:8006**

Gehen Sie bei **pve** unter **local(pve)** auf **CT Templates** und dann **Templates**, um ein Betriebssystem für die Container zu installieren.

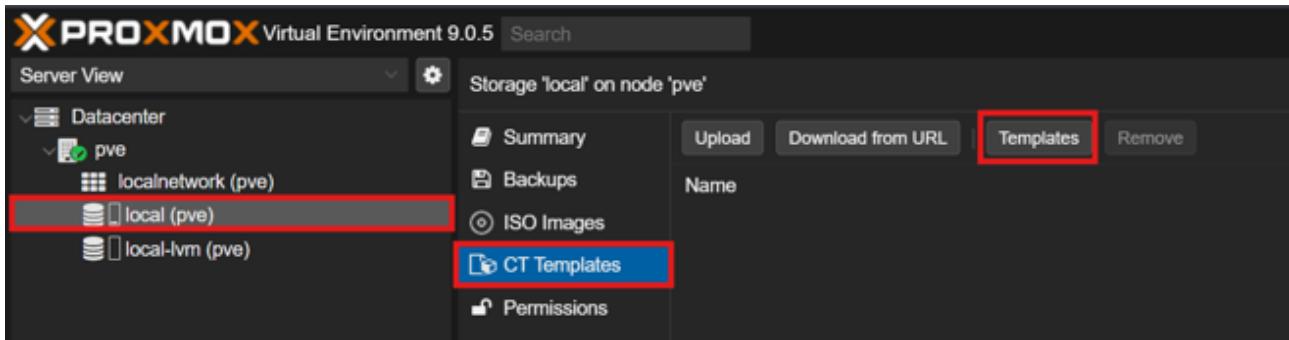


Abbildung 54 Container-Templates

Laden Sie das markierte Ubuntu Template herunter.

Type	Package	Version	Description
<b>Section: mail (1 Item)</b>			
lxc	proxmox-mail-gateway-8.2-standard	8.2-1	Proxmox Mail Gateway 8.2
<b>Section: system (14 Items)</b>			
lxc	devuan-5.0-standard	5.0	Devuan 5 (standard)
lxc	ubuntu-24.04-standard	24.04-2	Ubuntu 24.04 Noble (standard)
lxc	ubuntu-22.04-standard	22.04-1	Ubuntu 22.04 Jammy (standard)
lxc	openeuler-24.03-default	20250507	LXC default image for openeuler 24.03 (20250507)
lxc	opensuse-15.6-default	20240910	LXC default image for opensuse 15.6 (20240910)
lxc	debian-12-standard	12.7-1	Debian 12 Bookworm (standard)
lxc	fedora-42-default	20250428	LXC default image for fedora 42 (20250428)
lxc	centos-9-stream-default	20240828	LXC default image for centos 9-stream (20240828)
lxc	almalinux-9-default	20240911	LXC default image for almalinux 9 (20240911)
lxc	alpine-3.22-default	20250617	LXC default image for alpine 3.22 (20250617)
lxc	gentoo-current-openrc	20250508	LXC openrc image for gentoo current (20250508)
lxc	openeuler-25.03-default	20250507	LXC default image for openeuler 25.03 (20250507)
lxc	ubuntu-25.04-standard	25.04-1	Ubuntu 25.04 Plucky (standard)
lxc	rockylinux-9-default	20240912	LXC default image for rockylinux 9 (20240912)
<b>Section: turnkeylinux (111 Items)</b>			
			<b>Download</b>

Abbildung 55 Ubuntu-Template

Schliessen Sie das geöffnete Fenster nach der Installation und gehen Sie zum nächsten Schritt.

Jetzt sollten Sie bei **CT Templates** das **Ubuntu Image** sehen und somit auch den ersten **Container erstellen** können, indem Sie auf **Create CT** drücken.

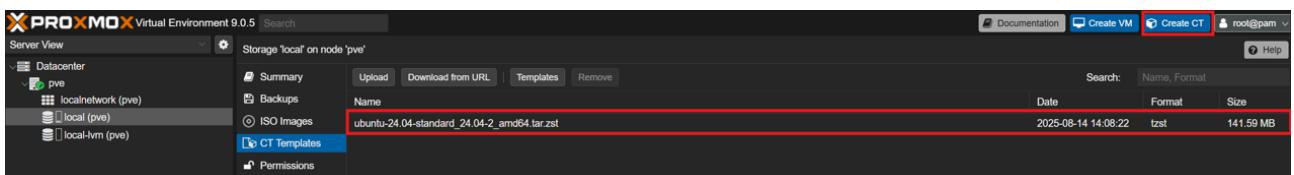


Abbildung 56 Container erstellen

Konfigurieren wir den Container mit den Informationen aus dem Plan, denn wir erstellt haben.

Den Namen können wir aus dem Namenskonzept nehmen. Das Passwort für den Container können Sie selbst definieren, **CT ID** ist die **Identifikationsnummer** für den Container, können Sie so belassen. Gehen Sie auf **Next**

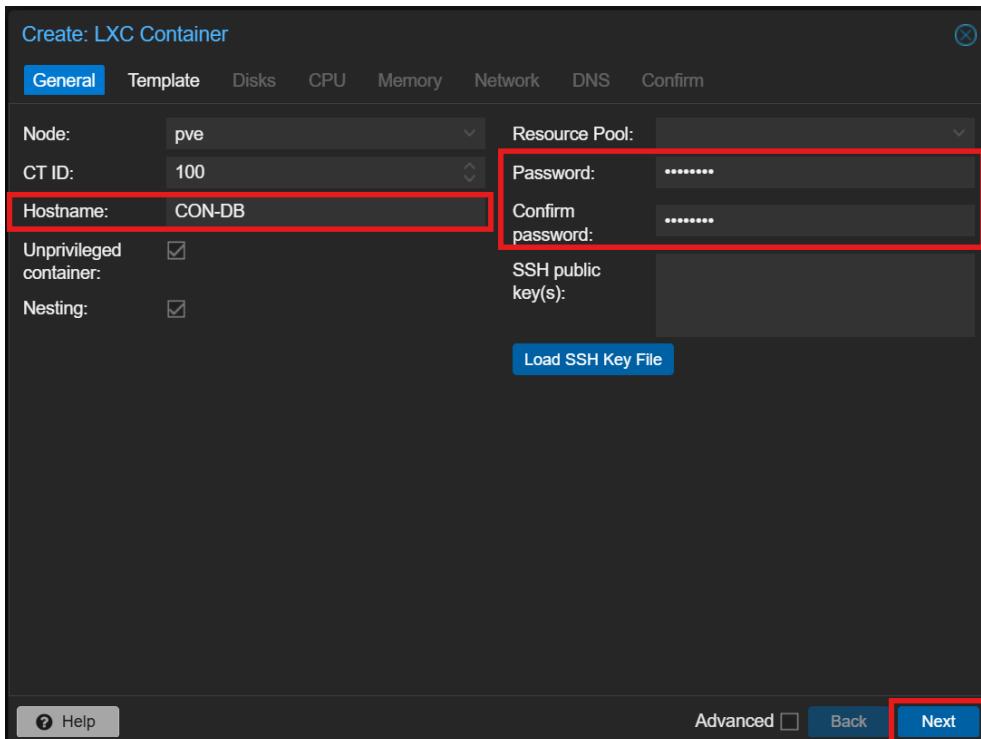


Abbildung 57 CT-General

Wählen Sie hier das **Ubuntu-Image**, das wir vorhin installiert haben und belassen Sie Storage auf **local**. Gehen Sie danach auf **Next**

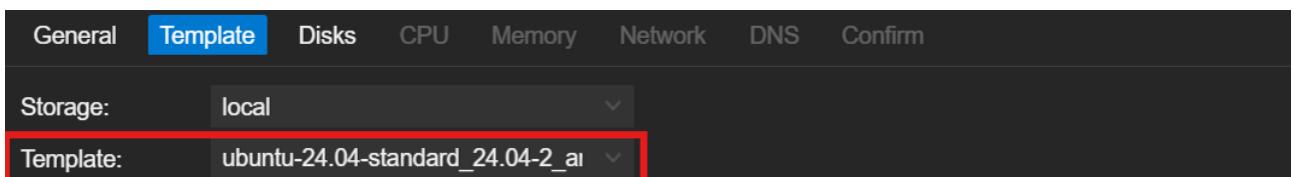


Abbildung 58 CT-Template

Hier definieren wir **wie viel Speicher** der Container für sich **reservieren** darf. Wir werden nach Plan **40GB pro Container** verwenden. Gehen Sie auf **Next**

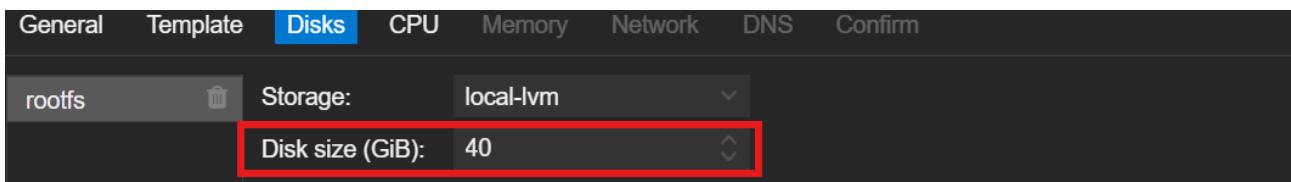


Abbildung 59 CT-Disks

Die CPUs(**cores**) werden wir auch wie nach Plan auf **3** setzen und dann auf **Next**

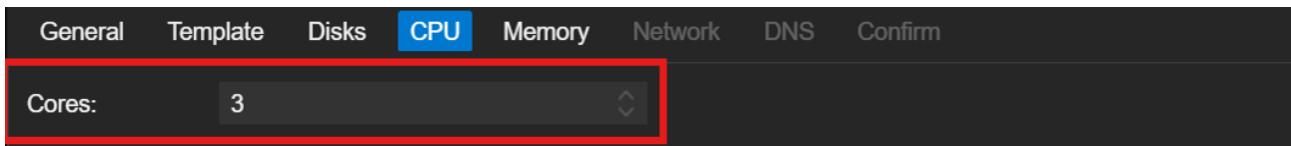


Abbildung 60 CT-cores

Der RAM soll nicht den RAM der VM übersteigen, weshalb wir pro Container nur **4GB** brauchen um auch Reserve zu schaffen für grössere Projekte. Dann auf **Next**

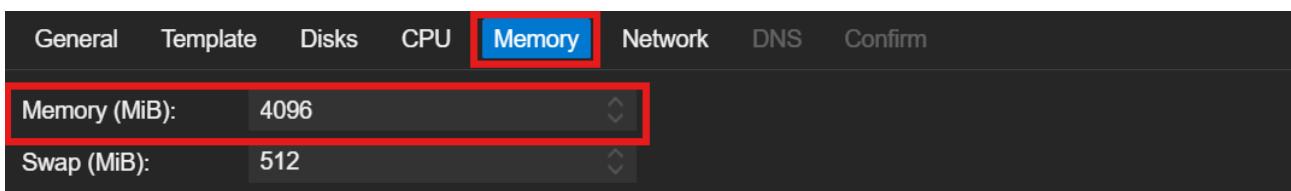


Abbildung 61 CT-RAM

Da wir im IP-Konzept jedem Container eine **Statische IP** gegeben haben werden wir das hier genauso tun. Da wir, denn Datenbank (**DB**) Container erstellen, werden wir **172.16.100.30**, wie im IP-Konzept definiert **verwenden**.

Das **Gateway** wird auf **172.16.100.1** gesetzt damit er auf die erstellte **Virtuelle Switch** zeigt.

Bridge wird so eingestellt das **vmbr0**, unser **Virtueller Switch**, verwendet wird.

Wenn alles eingestellt dann **Next**

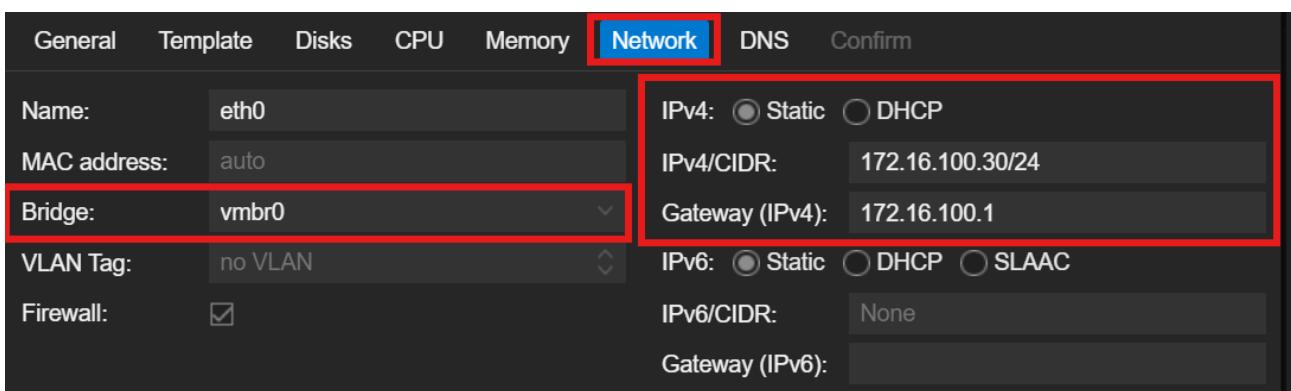


Abbildung 62 CT-Network

Als nächstes kommt die Option den **DNS** einzustellen, jedoch brauchen wir diesen **nicht** da wir Ihn vorhin auf dem **Host** schon eingestellt haben, weshalb wir die Option **use host settings** belassen und auf **Next** gehen.

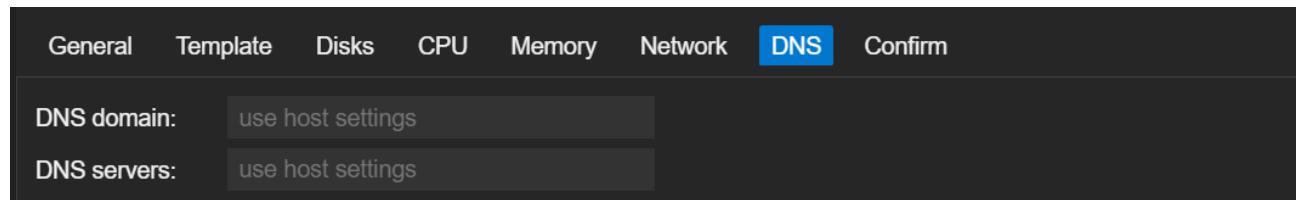


Abbildung 63 CT-DNS

Am Schluss unter **Confirm** haben Sie eine Zusammenfassung zu den getätigten Einstellungen. Diese können Sie auf Fehler überprüfen und wenn nötig mit **Back** zurück gehen und ausbessern.

Ansonsten können Sie **Start after created** aktivieren und auf **Finish** drücken.

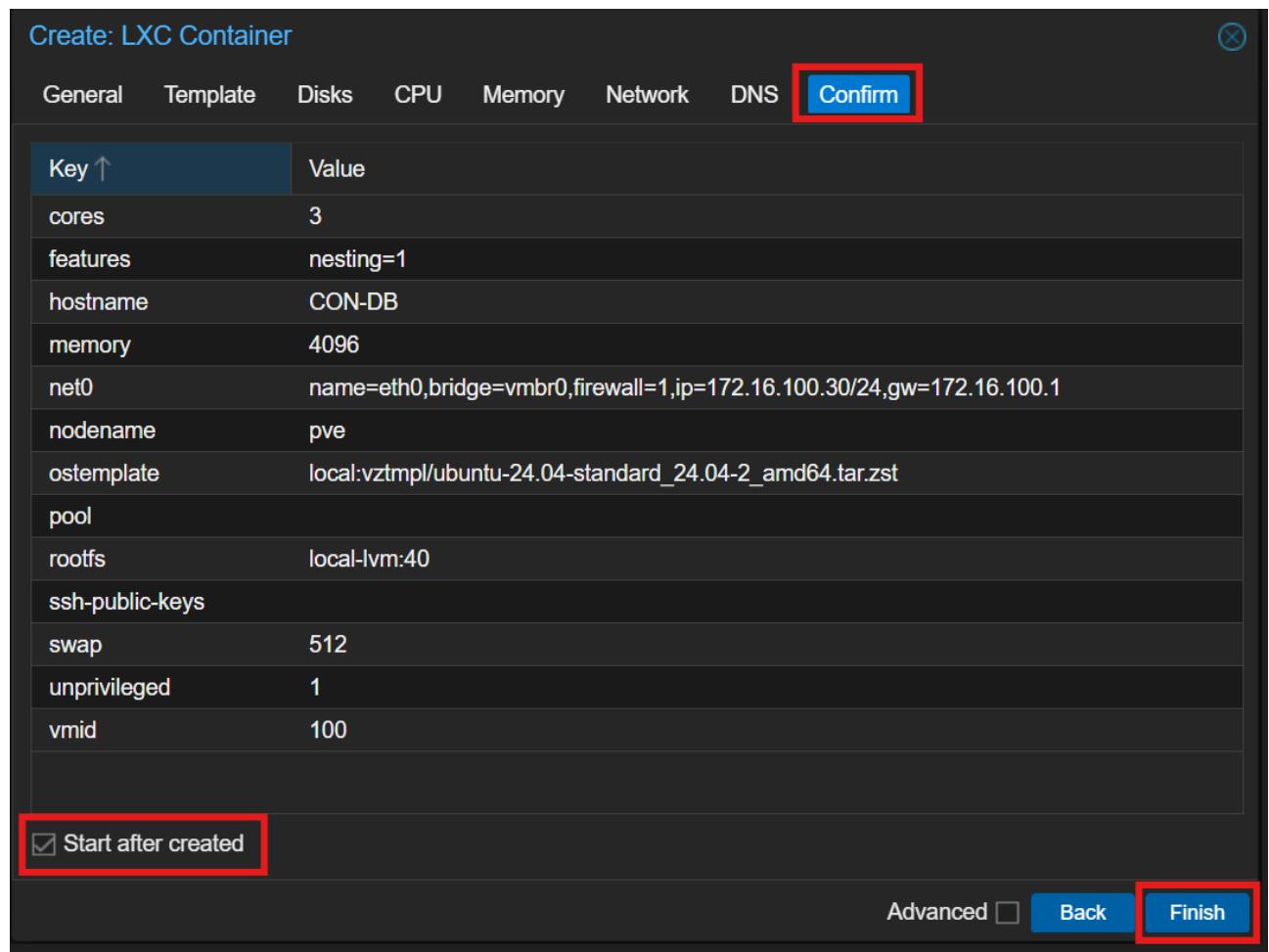
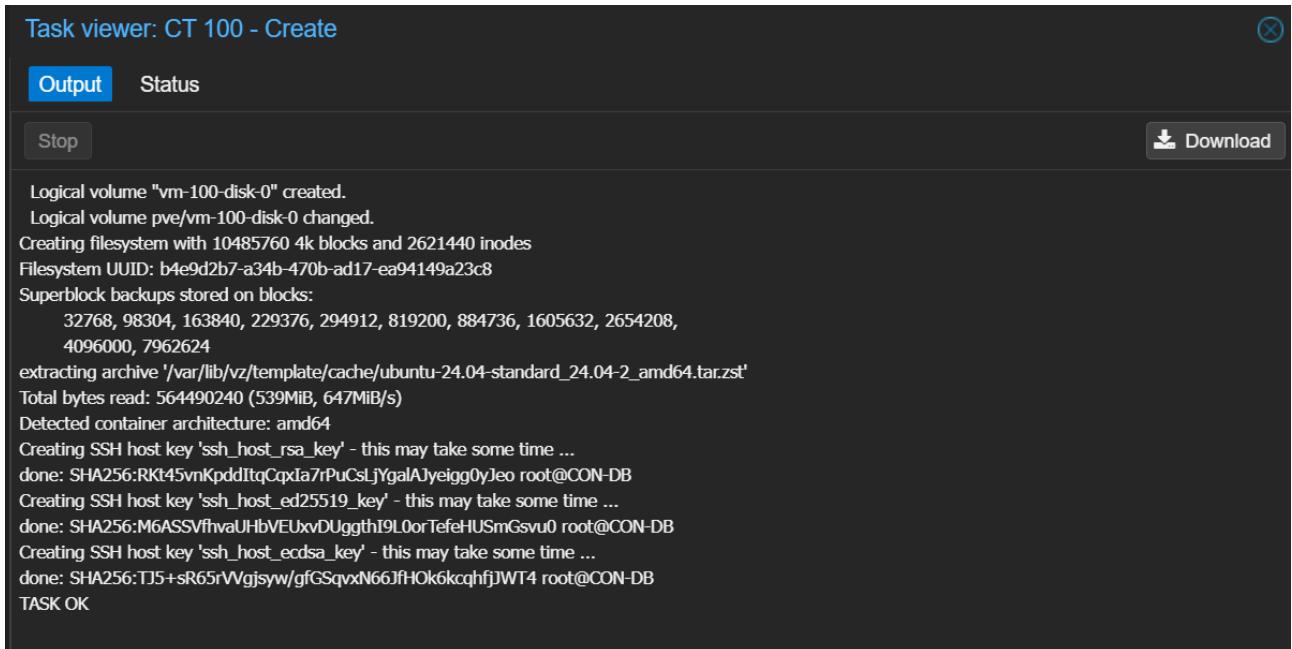


Abbildung 64 CT-Confirm

Somit **öffnet** sich ein kleines **Fenster** indem alle Befehle zur **Installation des Containers** automatisch ausgeführt werden und Sie nur auf **TASK OK** warten müssen, wenn **TASK OK** erscheint, können Sie den **Container Schliessen**.



```

Task viewer: CT 100 - Create
Output Status
Stop Download

Logical volume "vm-100-disk-0" created.
Logical volume pve/vm-100-disk-0 changed.
Creating filesystem with 10485760 4k blocks and 2621440 inodes
Filesystem UUID: b4e9d2b7-a34b-470b-ad17-ea94149a23c8
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624
extracting archive '/var/lib/vz/template/cache/ubuntu-24.04-standard_24.04-2_amd64.tar.zst'
Total bytes read: 564490240 (539MiB, 647MiB/s)
Detected container architecture: amd64
Creating SSH host key 'ssh_host_rsa_key' - this may take some time ...
done: SHA256:Rkt45vnKpddItqCqxIa7rPuCsLjYgalAJyeigg0yJeo root@CON-DB
Creating SSH host key 'ssh_host_ed25519_key' - this may take some time ...
done: SHA256:M6ASSVfhvaUhbVEUxDUgghI9L0rTefelHUSmGsvu0 root@CON-DB
Creating SSH host key 'ssh_host_ecdsa_key' - this may take some time ...
done: SHA256:TJ5+sR65rWgjsyw/gfGSqvxF66JfHOk6kcqhjJWT4 root@CON-DB
TASK OK

```

Abbildung 65 CT-Installieren

Hier Sehen Sie jetzt das Ihr Container **Erfolgreich installiert** und **eingerichtet** wurde. Sie können jetzt **Services** auf diesem Container herunterladen.

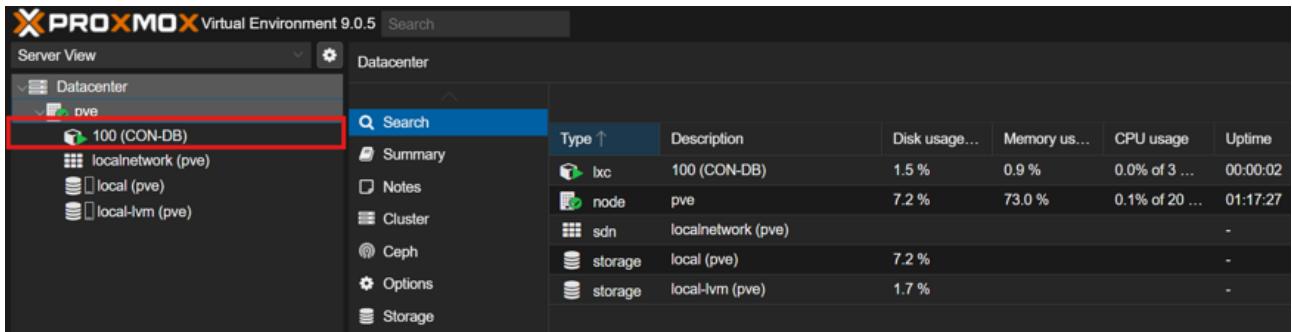


Abbildung 66 CT-Installiert

Somit haben wir Erfolgreich einen Container erstellt. Für die **Anderen Container** wie zum Beispiel für **Web-Server** und **MySQL-CLI**, können Sie **fast alle Schritte gleich** machen mit **Ausnahme von Netzwerk**, also die **IP des Containers** und der **Name des Containers**.

## 4.7. Skripte einbinden

Als nächstes schauen wir uns die Automatisierungen an. Diese können einzeln verwendet werden, um die Aufsetzung der Umgebung zu beschleunigen.

### 4.7.1. Automatisierte Services und Container

Hiermit kommen wir zum nächsten Schritt. Die Services Automatisch installieren. Wir haben mehrere Skripts im ZIP-Folder, den Sie erhalten haben, erstellt. Sie finden die Skripts im Ordner Alle\_Skripts. Die Skripts für die Services müssen immer in folgender Reihe ausgeführt werden.

1. Datenbank
2. Workbench (User)
3. Webserver

Beachten Sie auch das die Skripts so erstellt wurden dass Sie die Variablen nach Ihrem Wunsch anpassen können. Variablen wie zum Beispiel Passwort, Name oder Benutzer können beliebig angepasst werden. Es wird bei jedem Skript ein Logfile erstellt, das bei der Fehlerbehebung behilflich sein kann.

Damit Sie das Skript ausführen können müssen Sie in Ihrem Container Terminal folgenden Code eingeben und als Root angemeldet sein, um eine Datei mit Berechtigungen zu erstellen. (Dateiname kann hier selbst bestimmt werden)

**nano <dateiname> && chmod 700 <dateiname>**

Wenn Sie diesen Code ausgeführt haben werden Sie eine leere Datei haben, in diese Datei kopieren Sie den Inhalt des Skripts, denn wir Ihnen zur Verfügung gestellt haben.

Wenn Sie den Inhalt aus der Skript Datei in die nano-Datei im Terminal kopiert haben, speichern und schliessen Sie sie mit **ctrl + o → Enter → ctrl + x**

Dann können Sie das Skript mit folgendem code ausführen:

**./<dateiname>**

Wenn die Installation durchgeführt wurde, sehen Sie im Terminal ein paar Beispiele wie sie Testen können, ob alles richtig installiert wurde. Für die Fehlerbehebung ist dies sehr hilfreich da es im Skript Kommentare gibt, die den Code beschreiben. Das gleiche können Sie mit allen anderen Service Skripts machen. Das Einzige, was zu beachten ist, ist die Reihenfolge und der Dateiname, den Sie vergeben.

Um die **Container Skripts** auszuführen, müssen Sie das gleiche im Proxmox Terminal machen. Beachten Sie das Sie als erstes die Container installieren müssen und dann erst die Services im Container.

## 4.7.2. Automatisierung Virtuelle-Switch und Hyper-V

Das Netzwerk macht immer Probleme, weshalb das Automatisieren dieser Einrichtung vieles beschleunigt. Da wir ein Windows Betriebssystem als Host verwenden ist der Skript nur für Windows Umgebungen die PowerShell haben geeignet.

Wir haben im Skript Ordner mehrere Versionen hinterlegt, Ohne Admin Zustimmung und mit Admin Zustimmung. Sowie Skripts zum Löschen des erstellten Switches.

Der Skript ohne Admin fordert Sie nach dem Start auf, das Ausführen des Skripts zu bestätigen. Während sie, denn anderen Skript als Administrator ausführen müssen.

Öffnen Sie **PowerShell ISE als Administrator**:

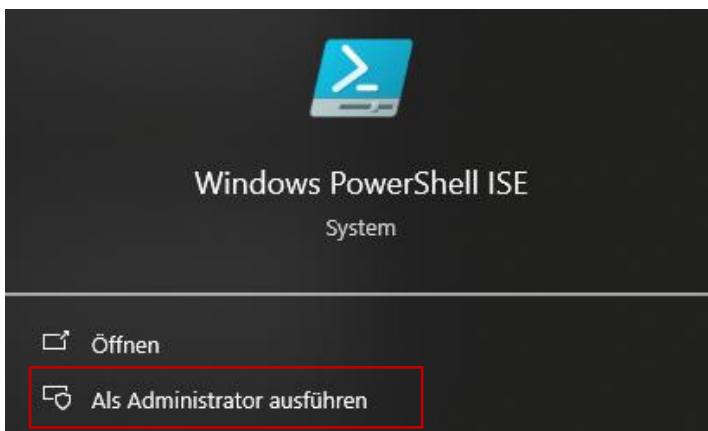


Abbildung 67 PowerShell ISE

Öffnen Sie mit (**ctrl + o**) denn Dateipfad und wählen Sie die **virtuelle\_switch\_adapter\_erdellen.ps1** Datei. Dann können Sie auf den folgenden **Knopf drücken**, um denn Skript auszuführen:



Abbildung 68 Skript-Start

Jetzt wird automatisch der Virtuelle Switch Adapter eingerichtet. Wie vorhin erwähnt haben Sie auch hier Kommentare im Code und die Möglichkeit variablen zu verändern wie Name, IP oder Typ.

Der Skript zum Löschen des Adapters kann auf die genau gleiche Art ausgeführt werden. Sowie der Skript **hyperV\_aktivieren.ps1** für die Aktivierung der **Hyper-V Features** auf Ihrem Gerät.

Somit haben wir die Mehrheit der Inbetriebnahme automatisiert.

## 5. Kontrollieren

Als nächstes überprüfen wir, ob die Umgebung wie gewollt funktioniert. Dafür erstellen wir ein Testkonzept. T01 = Test 01

Testkonzept			
Testname	Erwartetes Ergebnis	Tatsächliche Ergebnis	Getroffene Massnahmen
T01	Beim Ausführen des Datenbank Skripts wird die Datenbank mit allen Tabellen und Einträgen erstellt.	Die Datenbank wurde vollständig mit allen Tabellen und Einträgen fehlerfrei installiert.	Keine, die Datenbank wurden problemlos installiert.
T02	Der Ausgeführte Datenbank Skript hat erfolgreich die bind-address auf 0.0.0.0 gewechselt.	Die bind-address wurde erfolgreich ohne manuellen eingriff auf 0.0.0.0 gewechselt.	Keine, es wurde alles richtig konfiguriert und installiert.
T03	Der Ausgeführte Datenbank Skript hat erfolgreich den Benutzer labuser mit dem richtigen Host vorgaben installiert.	Der Benutzer labuser wurde erfolgreich erstellt und mit dem Host richtig verbunden.	Keine, der Benutzer wurde wie vorgesehen erstellt.
T04	Der Workbench Skript wurde erfolgreich ausgeführt und kann die erstellte labdb Datenbank sehen.	Die Workbench wurde erfolgreich mit dem Datenbank-Server verbunden.	Keine, Workbench erfolgreich verbunden.
T05	Der Web-Server Skript installiert Apache und erstellt automatisch Verbindung zur Datenbank	Der Web-Server Skript hat Apache installiert und erfolgreich die Datenbank verbunden.	Keine, Apache und Datenbankverbindung richtig installiert und verbunden.
T06	Zugriff auf Datenbank labdb von erstellter Webseite aus, mit Login möglich.	Zugriff auf Datenbank durch Webseite erfolgreich.	Keine, alles wurde richtig konfiguriert.

Tabelle 5 Testkonzept

grün = erfolgreich

rot = Nicht erfolgreich

blau = verbessert

Der Testablauf wird beim Testen jeweils für jeden Test erläutert.

## 5.1. T01 Ausführen und Prüfen

**Erwartetes Ergebnis:** Beim Ausführen des Datenbank Skripts wird die Datenbank mit allen Tabellen und Einträgen erstellt.

Führen Sie wie vorhin im Kapitel 4.7.1 auf Seite 38 vorgezeigt, denn Datenbank Skript aus.

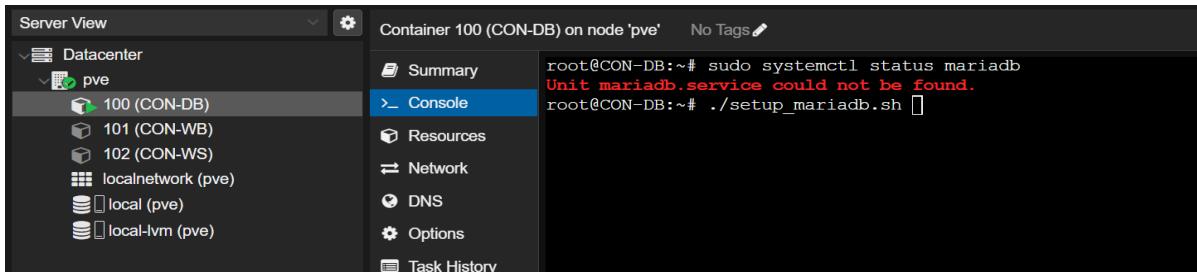


Abbildung 69 DB-Skript-Test

Nachdem Sie das Skript ausgeführt haben können Sie folgende Befehle, die Sie im Bild sehen ausführen und darauf achten, was als Antwort kommt.

```
root@CON-DB:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.13-MariaDB-Ubuntu0.24.04.1 Ubuntu 24.04

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)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| labdb          |
| mysql           |
| performance_schema |
| sys             |
+-----+
5 rows in set (0.001 sec)

MariaDB [(none)]> use labdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [labdb]> show tables;
+-----+
| Tables_in_labdb |
+-----+
| bestellungen    |
| kunden          |
| users           |
+-----+
3 rows in set (0.000 sec)

MariaDB [labdb]> select * from kunden;
+----+-----+-----+
| id | name   | email        | erstellt_am  |
+----+-----+-----+
| 1  | Max Mustermann | max@example.com | 2025-08-28 07:59:43 |
| 2  | Erika Muster   | erika@example.com | 2025-08-28 07:59:43 |
+----+-----+-----+
2 rows in set (0.001 sec)
```

Abbildung 70 DB-Nach-Installation-Test

Wenn die Datenbank **labdb** und die **Tabellen** und **Daten, wie im Bild** zu sehen, erscheinen, hat das Skript einwandfrei funktioniert und alle Vorgaben erfüllt.

**T01 bestanden.**

## 5.2. T02 Ausführen und Prüfen

**Erwartetes Ergebnis:** Der Ausgeführte Datenbank Skript hat erfolgreich die bind-address auf 0.0.0.0 gewechselt.

Öffnen Sie nach der Ausführung des Datenbank Skripts die config-Datei. In dieser Überprüfen Sie jetzt ob wie im Bild die bind-address auf 0.0.0.0 gesetzt wurde. **T02 bestanden.**

```

GNU nano 7.2
/etc/mysql/mariadb.conf.d/50-server.cnf

# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
[server]

# this is read by the standalone daemon and embedded servers

[mysqld]

#
# * Basic Settings
#


#user          = mysql
pid-file      = /run/mysqld/mysqld.pid
basedir       = /usr
#datadir       = /var/lib/mysql
#tmpdir        = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 0.0.0.0

```

Abbildung 71 DB-Config-Datei-Test

## 5.3. T03 Ausführen und Prüfen

**Erwartetes Ergebnis:** Der Ausgeführte Datenbank Skript hat erfolgreich den Benutzer labuser mit dem richtigen Host vorgaben installiert.

Loggen Sie sich mit dem Befehl mysql -u root -p in die Datenbank ein und geben Sie denn Befehl wie im Bild zu sehen ein.

```

MariaDB [(none)]> SELECT User, Host FROM mysql.user WHERE User = 'labuser';
+-----+----+
| User   | Host |
+-----+----+
| labuser | %    |
+-----+----+
1 row in set (0.001 sec)

MariaDB [(none)]>

```

Abbildung 72 DB-Benutzer-Test

Wenn Sie wie im Bild den **User labuser** haben und **Host auf %** gestellt ist, wurde alles bei der Automatisierung richtig eingestellt und der Benutzer erfolgreich eingerichtet.

**T03 bestanden.**

## 5.4. T04 Ausführen und Prüfen

**Erwartetes Ergebnis:** Der Workbench Skript wurde erfolgreich ausgeführt und kann die erstellte labdb Datenbank sehen.

Führen Sie wie in Kapitel 4.7.1, Seite 38 beschrieben, denn Workbench Skript aus. **Nachdem** Sie das **Skript ausgeführt** haben wird Ihnen ein **Hinweis angezeigt** wie Sie sich auf der Datenbank-Server als **labuser** anmelden.

```

root@CON-WB:~# ./setup_mysql_client.sh
[INFO] $
[INFO] $
[INFO] $
[INFO] $
[INFO] $
[INFO] $
server_version
10.11.13-MariaDB-0ubuntu0.24.04.1
Database
information_schema
labdb
Tables_in_labdb
bestellungen
kunden
users
id      name      email      erstellt_am
1      Max Mustermann  max@example.com  2025-08-28 07:59:43
2      Erika Muster   erika@example.com 2025-08-28 07:59:43
id      kunde_id    artikel_menge  preis      bestellt_am
1      1           USB-Stick       2         9.90     2025-08-28 07:59:43
2      1           Headset        1         49.00    2025-08-28 07:59:43
3      2           Tastatur       1         29.99    2025-08-28 07:59:43
id      name      artikel_menge  preis      bestellt_am
1      Max Mustermann  USB-Stick       2         9.90     2025-08-28 07:59:43
2      Max Mustermann  Headset        1         49.00    2025-08-28 07:59:43
3      Erika Muster   Tastatur       1         29.99    2025-08-28 07:59:43
[INFO] $
[INFO] $

[HINWEIS] Du kannst dich auch manuell mit folgendem Befehl einloggen:
mysql -h172.16.100.30 -P3306 -ulabuser -p
root@CON-WB:~# mysql -h172.16.100.30 -P3306 -ulabuser -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 45
Server version: 10.11.13-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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)]> show Databases;
+-----+
| Database |
+-----+
| information_schema |
| labdb          |
+-----+
2 rows in set (0.001 sec)

MariaDB [(none)]> 
```

Abbildung 73 Workbench-Test

Wenn Sie nach der Anmeldung (wie im Hinweis beschrieben) die Datenbank labdb wie im Bild sehen. Hat das Skript erfolgreich die Verbindung zum Datenbankserver erstellt.

Somit hat das Skript einwandfrei funktioniert.

T04 bestanden.

## 5.5. T05 Ausführen und Prüfen

**Erwartetes Ergebnis:** Der Web-Server Skript installiert Apache und erstellt automatisch Verbindung zur Datenbank.

Führen Sie den Web-Server Skript wie in Kapitel 4.7.1, Seite 38 beschrieben aus.

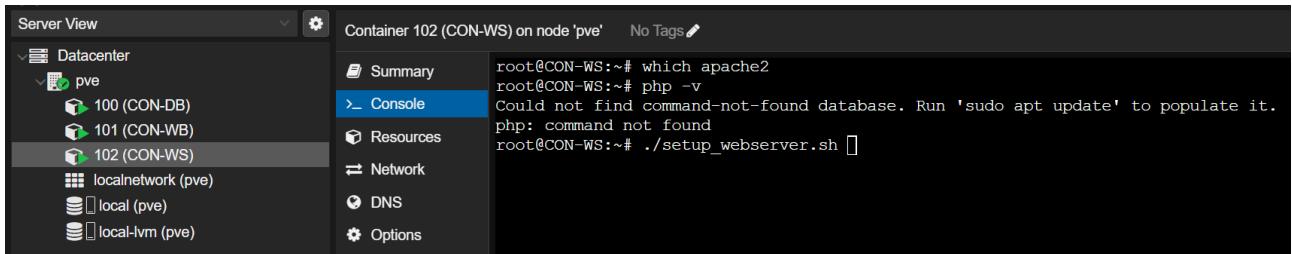


Abbildung 74 Web-Server-Skript-Test

Jetzt prüfen Sie mit den Folgenden Befehlen, die Sie im Bild sehen, ob PHP, sowie Apache2 installiert wurden.

```
[INFO] Pakete installiert.
[INFO] Erstelle Web-App: labdb.php
[INFO] Webseite labdb.php erstellt.
[INFO] Füge Benutzer in Datenbank ein...
mysql: [Warning] Using a password on the command line interface can be insecure.
[INFO] Benutzer 'admin' hinzugefügt.
mysql: [Warning] Using a password on the command line interface can be insecure.
[INFO] Benutzer 'labuser' hinzugefügt.
[INFO] Starte Apache neu...
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2

Zugriff auf deine Seite:
http://172.16.100.20/labdb.php

Verbindung zur Datenbank:
mysql -h172.16.100.30 -ulabuser -p
root@CON-WS:~# php -v
PHP 8.3.6 (cli) (built: Jul 14 2025 18:30:55) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
root@CON-WS:~# which apache2
/usr/sbin/apache2
root@CON-WS:~# ]
```

Abbildung 75 Apache-PHP-Test

Wenn Sie die Versionisierung von PHP sowie den Pfad von Apache2 wie im Bild sehen, hat der Skript einwandfrei die definierten Dienste installiert. **T05 bestanden.**

## 5.6. T06 Ausführen und Prüfen

Erwartetes Ergebnis: Zugriff auf Datenbank labdb von erstellter Webseite aus, mit Login möglich.

Geben Sie auf Ihrem **Browser** in der **Suchleiste** die **Adresse** die Sie nach der Ausführung des Web-Server Skripts als Kommentar erhalten haben ein.

Zugriff auf deine Seite:  
**http://172.16.100.20/labdb.php**

Abbildung 76 Webseite-DB-Adresse

Wenn Sie dies gemacht haben müssen Sie die **Logindaten** für den **labuser eingeben** und auf **Einloggen** drücken.



Abbildung 77 Webseite-DB-Login-Test

Wenn Sie eingeloggt wurden, sehen Sie eine PHP-Seite, auf der Sie die Datenbank labdb manipulieren können (Wie im Bild unten). Somit ist die Funktionalität des Skripts geprüft.

**T06 bestanden.**

ID	Name	Email
1	Max Mustermann	max@example.com
2	Erika Muster	erika@example.com

ID	Kunde	Artikel	Menge	Preis
1	Max Mustermann	USB-Stick	2	9.90

Abbildung 78 Labdb-PHP-Test

## 6. Auswerten

### Auswertung des Projekts

**Projektname:** [REDACTED]

**Autoren:** Milan Milivojevic

**Auftraggeber:** [REDACTED]

**Projektnummer:** 2443323

**Abschlussdatum:** 22.08.2025

### Projektziel

Das Ziel des Projekts war es, die bisher langsame und manuelle Erstellung von Systemen für einzelne Kunden durch eine automatisierte Lösung zu ersetzen.

Durch den Einsatz von **Proxmox**, **Hyper-V**, **LXD-Containern** und **Skripten** wurde eine Umgebung geschaffen, die die Bereitstellung von **VMs**, **Containern** und **Services** stark beschleunigt und gleichzeitig die Wartung vereinfacht.

#### 6.1.1. Herausforderungen

- Komplexe **Netzwerkkonfiguration** mit statischen IPs, NAT, Virtuelle Adapter
- Sicherstellung der **Kompatibilität** zwischen Hyper-V, Proxmox und LXD
- Strukturierung der Automatisierungen, damit sie unabhängig, flexibel und erweiterbar bleiben
- Fehlerbehebung bei Skript Ablauf Fehlern

## 7. Abbildungsverzeichnis

Abbildung 1 Zeitplan .....	6
Abbildung 2 Windows-Features .....	10
Abbildung 3 Hyper-V Aktivieren .....	10
Abbildung 4 Hyper-V-Manager .....	11
Abbildung 5 Hyper-V-Menu .....	11
Abbildung 6 Virtueller Computer .....	12
Abbildung 7 VM-Speicherort .....	12
Abbildung 8 VM-Gastbetriebssystem 64-bit .....	13
Abbildung 9 VM-RAM .....	13
Abbildung 10 VM-Speicher .....	14
Abbildung 11 VM-ISO-File .....	15
Abbildung 12 VM-Zusammenfassung .....	15
Abbildung 13 VM-Einstellungen .....	16
Abbildung 14 VM-Sicherheit .....	16
Abbildung 15 VM-Status Check .....	17
Abbildung 16 ExposeVirtualizationExtensions .....	17
Abbildung 17 mit VM Verbinden .....	17
Abbildung 18 VM-Starten .....	18
Abbildung 19 Proxmox Installation .....	18
Abbildung 20 Proxmox-Speicher .....	19
Abbildung 21 Root-Passwort .....	19
Abbildung 22 Proxmox-Network .....	20
Abbildung 23 Zusammenfassung-Proxmox .....	20
Abbildung 24 Proxmox-Reboot .....	21
Abbildung 25 Proxmox-Terminal-Login .....	21
Abbildung 26 Proxmox auf Browser .....	22
Abbildung 27 Sicherheitswarnung-Umgehen .....	22
Abbildung 28 Browser-Sicherheitswarnung .....	22
Abbildung 29 ProxmoxWeb-Login .....	23
Abbildung 30 Proxmox-GUI .....	23
Abbildung 31 Proxmox-Repositories .....	24
Abbildung 32 Repository-No-Subscription .....	24
Abbildung 33 Proxmox-Updates .....	24
Abbildung 34 Proxmox-Upgrade .....	25
Abbildung 35 Upgrade-Bestätigen .....	25
Abbildung 36 Update-Reboot .....	25
Abbildung 37 Powershell-Administrator .....	26
Abbildung 38 Neue-Switch .....	26
Abbildung 39 Neue-Adapter .....	26
Abbildung 40 IP-von-Adapter .....	26
Abbildung 41 Zugeteilte-IP-Löschen .....	27

Abbildung 42 Neue-IP-Zuweisen .....	27
Abbildung 43 Adapter-IP-Kontrollieren .....	27
Abbildung 44 Neue-NAT .....	27
Abbildung 45 Hyper-V-Netzwerkkarte-Einstellungen .....	28
Abbildung 46 Switch-Umstellen.....	28
Abbildung 47 Konfigurationsdatei-Proxmox .....	29
Abbildung 48 Konfigurationsdatei-Setup .....	29
Abbildung 49 Netzwerk-Neustart .....	29
Abbildung 50 Nameserver-Datei .....	30
Abbildung 51 IP-Check.....	30
Abbildung 52 hosts-definieren.....	31
Abbildung 53 host-einfügen .....	31
Abbildung 54 Container-Templates .....	32
Abbildung 55 Ubuntu-Template .....	32
Abbildung 56 Container erstellen.....	33
Abbildung 57 CT-General .....	33
Abbildung 58 CT-Template.....	33
Abbildung 59 CT-Disks .....	34
Abbildung 60 CT-cores .....	34
Abbildung 61 CT-RAM .....	34
Abbildung 62 CT-Network.....	34
Abbildung 63 CT-DNS .....	35
Abbildung 64 CT-Confirm .....	35
Abbildung 65 CT-Installieren.....	36
Abbildung 66 CT-Installiert .....	36
Abbildung 67 PowerShell ISE .....	38
Abbildung 68 Skript-Start .....	38
Abbildung 69 DB-Skript-Test .....	40
Abbildung 70 DB-Nach-Installation-Test.....	40
Abbildung 71 DB-Config-Datei-Test.....	41
Abbildung 72 DB-Benutzer-Test .....	41
Abbildung 73 Workbench-Test .....	42
Abbildung 74 Web-Server-Skript-Test.....	43
Abbildung 75 Apache-PHP-Test .....	43
Abbildung 76 Webseite-DB-Adresse .....	44
Abbildung 77 Webseite-DB-Login-Test .....	44
Abbildung 78 Labdb-PHP-Test .....	44

## 8. Tabellenverzeichnis

Tabelle 1 Änderungsverzeichnis .....	1
Tabelle 2 Namenskonzept .....	6
Tabelle 3 IP-Konzept .....	6
Tabelle 4 Systemanforderungen.....	7

## 9. Quellenverzeichniss

### Hyper-V:

- ❖ <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/get-started/install-hyper-v?tabs=powershell&pivots=windows>
- ❖ <https://learn.microsoft.com/en-us/windows-server/virtualization/>
- ❖ <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/overview>

### Proxmox:

- ❖ <https://pve.proxmox.com/pve-docs/chapter-sysadmin.html>
- ❖ <https://phoenixnap.com/kb/install-proxmox>
- ❖ <https://www.proxmox.com/en/downloads/proxmox-virtual-environment/iso/proxmox-ve-9-0-iso-installer>

### Netzwerk Adapter:

- ❖ <https://4sysops.com/archives/create-a-hyper-v-virtual-switch-with-powershell/>
- ❖ <https://msftwebcast.com/2025/05/how-to-create-a-hyper-v-virtual-switch-using-powershell.html>

### Container:

- ❖ <https://www.tecmint.com/proxmox-create-container/>
- ❖ <https://forum.proxmox.com/threads/internal-network-between-containers.25962/>

### Services:

- ❖ <https://gcore.com/learning/setting-up-basic-web-server-ubuntu>
- ❖ <https://ubuntu.com/tutorials/install-and-configure-apache#4-setting-up-the-virtualhost-configuration-file>
- ❖ <https://mariadb.com/get-started-with-mariadb/>
- ❖ <https://documentation.ubuntu.com/server/how-to/databases/install-mysql/>
- ❖ <https://www.bytebase.com/reference/mysql/how-to/how-to-install-mysql-client-on-mac-ubuntu-centos-windows/>

### Automatisierung:

- ❖ <https://learn.microsoft.com/en-us/training/modules/script-with-powershell/>
- ❖ <https://stackoverflow.com/questions/2035193/how-to-run-a-powershell-script>
- ❖ <https://blog.netwrix.com/powershell-scripting-tutorial/>