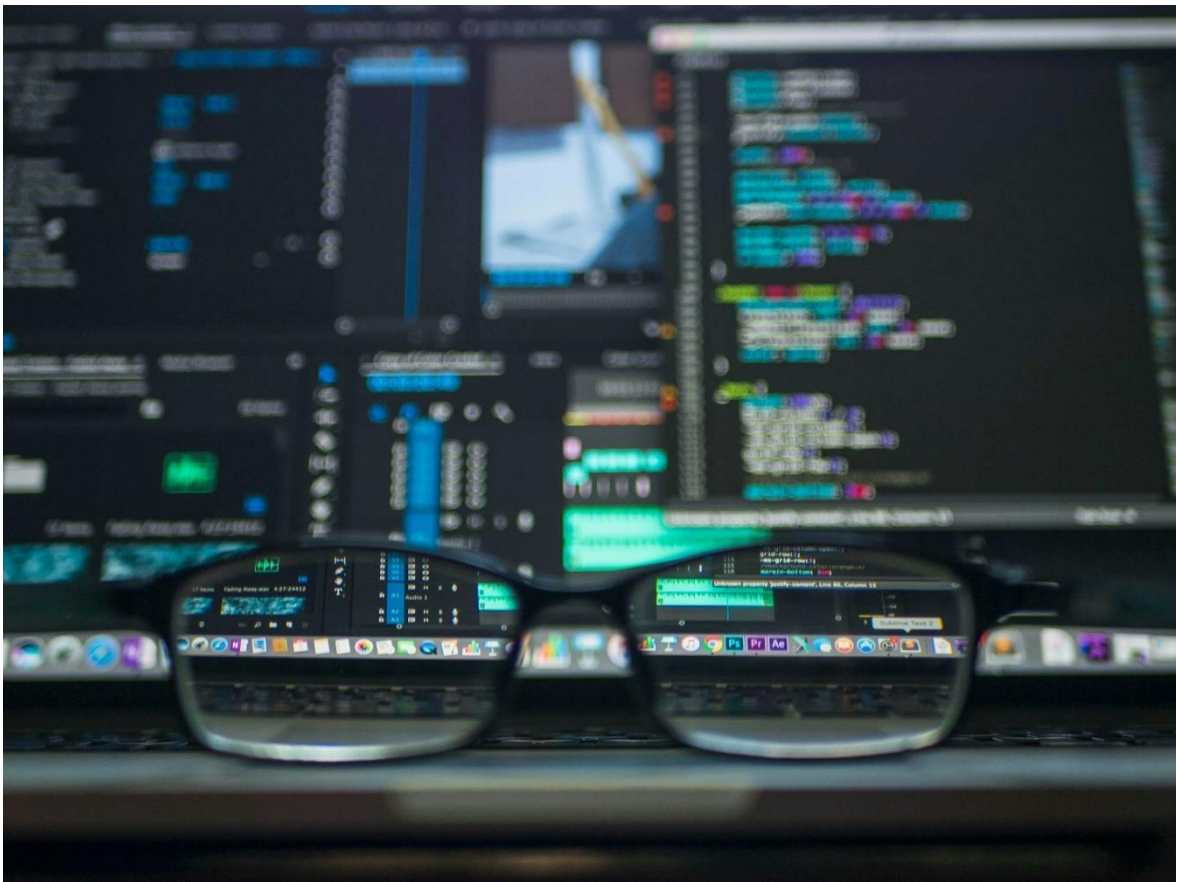


# Automatisierte Bereitstellung einer Virtuellen Umgebung

**Virtuelle Infrastruktur per Ansible vollständig konfigurier- und reproduzierbar aufbauen**



# Inhaltsverzeichnis

<b>1</b>	<b>Verwendete Abkürzungen .....</b>	<b>3</b>
<b>2</b>	<b>Einleitung .....</b>	<b>4</b>
2.1	Was ist das Projekt? .....	4
2.2	Warum wurde es umgesetzt? .....	4
<b>3</b>	<b>Zielsetzung .....</b>	<b>5</b>
3.1	Automatisierte Infrastruktur auf Proxmox .....	5
3.2	Einsatz von Infrastructure as Code (IaC) .....	5
<b>4</b>	<b>Techstack .....</b>	<b>6</b>
4.1	Proxmox VE .....	6
4.2	Ansible .....	6
4.3	Pfsense.....	6
4.4	Open vSwitch (OVS) .....	6
4.5	Network Attached Storage (NAS) / lösche ich evtl. ....	6
4.6	YAML, Bash .....	6
<b>5</b>	<b>Systemarchitektur .....</b>	<b>7</b>
5.1	Umgebung (Proxmox Host, VMs, VLANs, etc.) .....	7
5.2	Netzwerkaufbau, Firewall-Zonen, Storage-Konzept .....	7
<b>6</b>	<b>Umsetzung.....</b>	<b>8</b>
6.1	Rollen und Playbooks im Überblick .....	8
6.2	Konfiguration (Snippets) .....	8
6.3	Erstellte Ressourcen (VMs, VLANs, Netzwerk-Interfaces, etc.) .....	8
<b>7</b>	<b>Automatisierung &amp; Wiederverwendbarkeit .....</b>	<b>9</b>
7.1	Variablen in Ansible.....	9
7.2	Templates und Wiederverwendung.....	9
<b>8</b>	<b>Herausforderungen &amp; Learnings .....</b>	<b>10</b>
8.1	Technische Hürden & Lösungswege .....	10
8.2	Erkenntnisse im Umgang mit Proxmox & Ansible .....	10
<b>9</b>	<b>Fazit &amp; Ausblick .....</b>	<b>11</b>
9.1	Ziel erreicht? .....	11
9.2	Möglichkeit zur Erweiterung .....	11

# 1      **Verwendete Abkürzungen**

Unten in der Tabelle aufgelistet sind die verwendeten Abkürzungen in dieser Projekt Dokumentation

<b>Abkürzung</b>	<b>Bedeutung</b>
VM	Virtuelle Maschine
IaC	Infrastructure as Code

## **2 Einleitung**

In diesem Abschnitt werde ich kurz darauf eingehen um was sich dieses Projekt handelt und wieso ich es umgesetzt habe.

### **2.1 Was ist das Projekt?**

Ziel des Projektes ist die automatisierte Konfiguration und Bereitstellung einer virtuellen Infrastruktur mittels Ansible, wobei nicht nur die VMs automatisiert werden, sondern auch Netzwerkkonfigurationen, Firewallzonen etc. bereitgestellt werden. Dadurch soll es auch möglich sein, die komplette Umgebung in kurzer Zeit zu reproduzieren.

### **2.2 Warum wurde es umgesetzt?**

Die Umsetzung dieses Projekts entstand aus meinem persönlichen Interesse an automatisierten Infrastrukturen und dem Wunsch, meine Kenntnisse im Bereich Infrastructure as Code (IaC) und Virtualisierungstechnologien praxisnah zu vertiefen. Als angehender Informatiker im Bereich der Plattformentwicklung ist es mir wichtig, nicht nur manuell gesetzte Konfigurationen zu verstehen, sondern auch skalierbare und reproduzierbare Lösungen zu entwickeln.

Das Projekt diene mir somit nicht nur als fachliche Übung, sondern auch als persönliche Weiterentwicklung mit dem Ziel, komplexe Systeme effizient, strukturiert und dokumentiert aufzubauen - Fähigkeiten, die im Berufsalltag eines System Engineers essentiell sind.

### 3 Zielsetzung

In diesem Abschnitt werde ich mir Ziele setzen, die ich an dieses Projekt habe.

#### 3.1 Automatisierte Infrastruktur auf Proxmox

Als Ziel setze ich mir die Erstellung und Verwaltung einer kompletten virtuellen Infrastruktur auf einem Proxmox-Server zu automatisieren mithilfe von Ansible. Ziel ist es wiederholbare und konsistente Deployments von VMs, Netzwerkkonfigurationen und VLANs zu ermöglichen ohne manuelles Eingreifen über die Proxmox-Oberfläche.

Die Automatisierung umfasst folgende Punkte:

- **Bereitstellung von VMs:** Anlegen neuer virtueller Maschinen basierend auf Templates, inklusive die Zuweisung von CPU, RAM, Storage und Netzwerkkonfigurationen.
- **Netzwerkkonfiguration:** Erstellung und Zuweisung von VLANs, Schnittstellen und IP-Konfigurationen.
- **Integration in Proxmox:** Ansible nutzt die Proxmox-API, um die Konfigurationen serverseitig auszuführen.

Durch diese automatisierte Infrastruktur können wir den Aufwand reduzieren und damit auch mögliche Konfigurationsfehler minimieren. Ausserdem folgen wir damit den IaC-Prinzipien, was die Wartbarkeit & Nachvollziehbarkeit der Umgebung langfristig verbessert.

#### 3.2 Einsatz von Infrastructure as Code (IaC)

Im Mittelpunkt dieses Projektes steht die konsequente Nutzung von IaC zur Beschreibung und Automatisierung der Infrastruktur. Ziel ist die Ablösung der aufwändigen manuellen Erstellung von virtuellen Maschinen und Netzwerken über die Proxmox-Schnittstelle.

## **4 Techstack**

In diesem Abschnitt werde ich auf meinen verwendeten Techstack eingehen diese aufgelisteten Technologien habe ich für die Umsetzung meines Projektes verwendet.

### **4.1 Proxmox VE Virtualisierungsplattform (Hypervisor)**

Als Hypervisor für das Virtualisieren unserer Virtuellen Maschinen verwenden wir Proxmox. Proxmox ist eine Open-Source Lösung die perfekt für dieses Projekt passt. Proxmox bietet eine leistungsfähige Weboberfläche, Snapshot-Funktionen sowie Unterstützung für VLANs, Bridges und SDN (Software Defined Networking).

### **4.2 Ansible Automatisierung & Infrastructure as Code (IaC)**

Ansible wird zur Automatisierung der Infrastruktur verwendet. Über Playbooks in YAML-Syntax wird die Erstellung, Konfiguration und Verwaltung der virtuellen Maschinen sowie der Netzwerkumgebung gesteuert. Durch die Nutzung von Ansible kann die Virtuelle Infrastruktur wiederholbar ausgerollt werden.

### **4.3 Pfsense Firewall & Routing**

Pfsense fungiert als virtuelle Firewall und Routing-Komponente innerhalb des virtuellen Netzwerks. Sie wird als VM auf Proxmox bereitgestellt und ermöglicht die Segmentierung des Netzwerks, den Aufbau von VLANs sowie grundlegende Sicherheitsmechanismen wie NAT, Portweiterleitungen und Paketfilterung.

### **4.4 Open vSwitch (OVS) Virtuelles Switching / SDN**

Open vSwitch wird in Kombination mit Proxmox verwendet, um flexible virtuelle Netzwerke aufzubauen. Es ermöglicht das Anlegen von VLANs, das Routing zwischen Netzwerksegmenten sowie die Definition komplexer Switching-Regeln.

### **4.5 Network Attached Storage (NAS) / lösche ich evtl. Zentraler Speicher für Images und Backups**

Unser NAS-System dient als zentraler Speicherort für VM-Templates, ISO...

### **4.6 YAML, Bash Skripting und Konfigurationssprache**

YAML wird als deklarative Sprache zur Erstellung von Ansible-Playbooks genutzt. Bash dient für einfache Hilfsskripte.

## **5 Systemarchitektur**

In diesem Abschnitt wird die Systemarchitektur dargestellt anhand verschiedener Grafischen Abbildungen.

### **5.1 Umgebung (Proxmox Host, VMs, VLANs, etc.)**

### **5.2 Netzwerkaufbau, Firewall-Zonen, Storage-Konzept**

## **6      Umsetzung**

### **6.1    Rollen und Playbooks im Überblick**

### **6.2    Konfiguration (Snippets)**

### **6.3    Erstellte Ressourcen (VMs, VLANs, Netzwerk-Interfaces, etc.)**



## **7      Automatisierung & Wiederverwendbarkeit**

### **7.1    Variablen in Ansible**

### **7.2    Templates und Wiederverwendung**

## **8 Herausforderungen & Learnings**

### **8.1 Technische Hürden & Lösungswege**

### **8.2 Erkenntnisse im Umgang mit Proxmox & Ansible**

## **9      Fazit & Ausblick**

### **9.1      Ziel erreicht?**

### **9.2      Möglichkeit zur Erweiterung**