

**Table Of Contents**

Homelab? 2

Why Homelab? 2

My Purpose 2

Goals 2

Hardware 3

Software 3

Network stack 3

Proxmox 4

Virtual machine inside proxmox 5

Remote access 8

Future plans and Upgrades 9

Conclusion 10

Resources 10

**Homelab?**

A homelab is a personal setup where people create a small-scale IT environment at home. It’s like having a mini data center or computer network for experimenting, learning, or testing things. This mini data center consists of all hardware, software and networking equipment that resembles the original data center. An environment set-up to test, experience and learn new IT technologies and environments in their home. For setting up a homelab you don’t need to have advanced hardware and equipment, a old computer is enough!

**Why Homelab?**

A homelab is basically a personal tech setup at home where you can learn, experiment, and build IT skills. It’s perfect for practicing things like networking, programming, cybersecurity, or managing servers in a safe and controlled environment. You can try out new software, test configurations, or mess around with tools without worrying about breaking anything important. It’s also great for building cool projects like hosting a website, running a game server, or setting up your own private cloud. If you’re looking to grow your career in IT or cybersecurity, a homelab is a fantastic way to get hands-on experience and show off your skills. “Think of it as your own little tech playground at home”.

**My Purpose**

With an enthusiastic and eager mind for learning about servers and networking, I found the concept of a homelab incredibly fascinating, it felt like an evening snack for my brain. Initially, I started this homelab with no specific plans, but now I have exciting projects in mind. I’ve set up my own movie streaming platform, an attack box using Kali Linux, and a Docker environment running on an Ubuntu virtual machine. Having a personal lab offers numerous advantages, and the satisfaction of making everything work through my own efforts is truly rewarding.

**Goals**

I want to understand the importance of a homelab and take my learning to the next level. Throughout my journey, I’ve enjoyed the process of making things work. With this homelab, I can create projects that not only help me learn but also enhance my home with innovative tech ideas, improved security features, and smarter access controls.

**Hardware**

I strongly believe in the concept of “Stop thinking, just do” which pushes me to take action. This mindset is complemented by “Do the best with what you currently have” encouraging me to make the most of available resources. My homelab journey began with a single computer featuring the following specs:

CPU: Intel Pentium dual core

Ram: 4gb

Storage: 260gb HDD

That’s all I know about this machine, but it’s enough to get started.

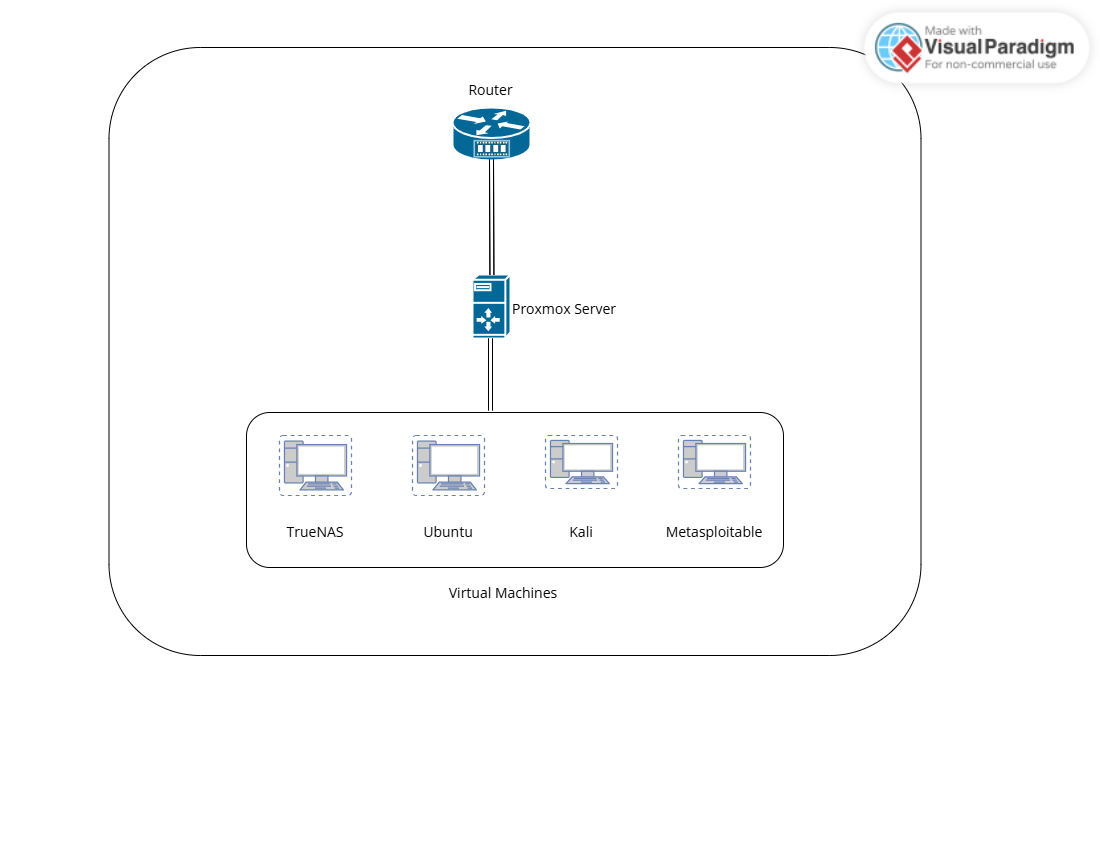
**Software**

For my purpose, I needed a hypervisor that allows me to run multiple virtual machines, so I chose Proxmox, a popular Linux-based hypervisor. Using Proxmox, I’ve currently set up three virtual machines:

1. TrueNAS Scale: Used as a Network Attached Storage (NAS) for personal streaming and data management.
2. Kali Linux: Configured as an attack box for pentesting and cybersecurity experiments.
3. Ubuntu with Docker: A virtual machine running a Docker environment for containerized applications and projects.
4. Metasploitable: Used for performing attacks.

**Network Stack**

Currently, I don’t have additional networking devices like switches or access points. I only have the router provided by my internet service provider. For the network setup, I’ve connected the server directly to the router. Providing the diagram as follow:

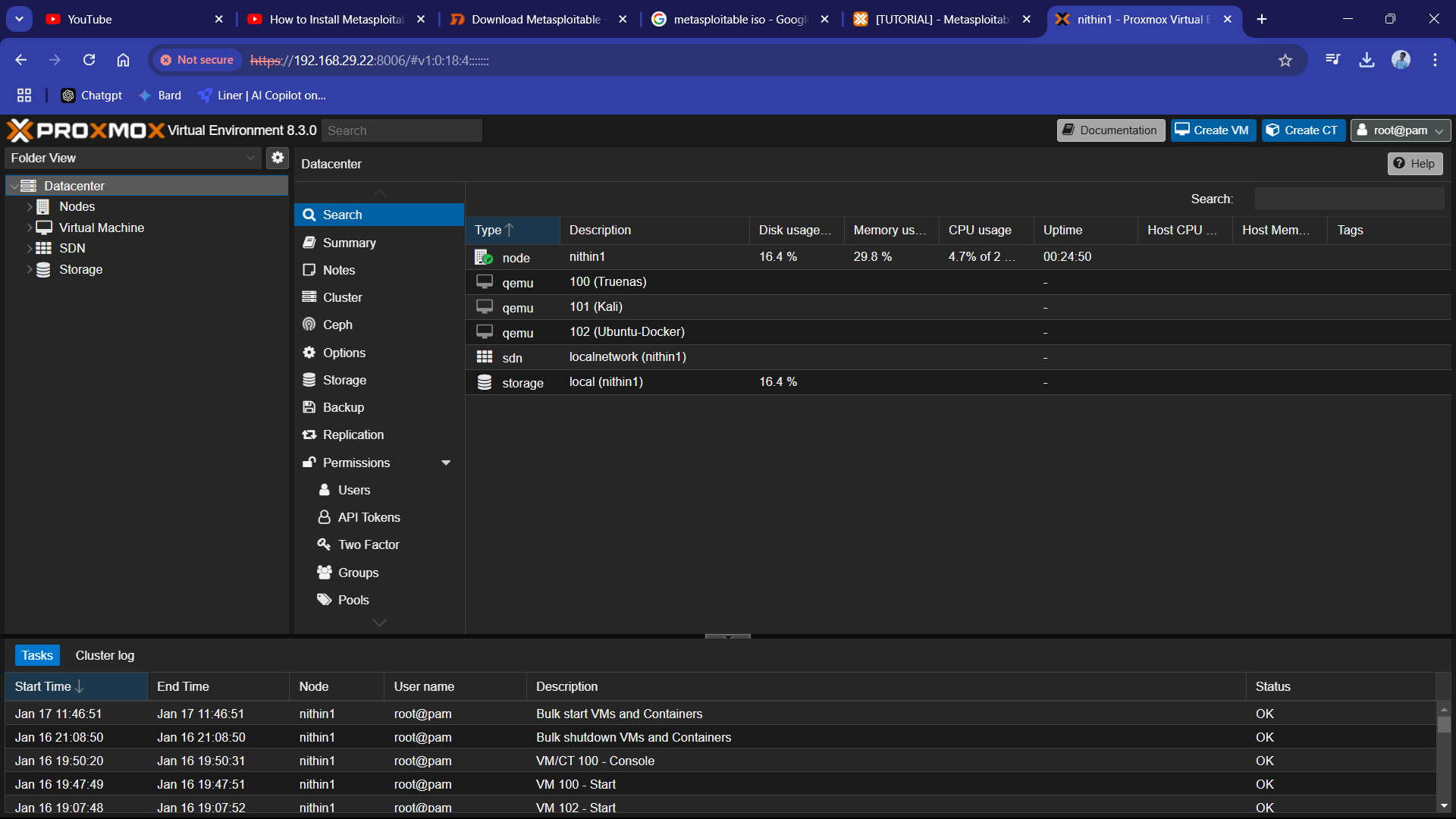


**Proxmox**

Proxmox is an open-source virtualization platform that allows you to manage and run virtual machines (VMs) and containers. It’s based on Debian Linux and is popular for its simplicity, ease of use, and powerful features. Proxmox provides a web-based interface to manage your virtual environments, which can include virtual machines, containers (like Docker), and even storage. It is free and open source contributing to many communities, Centralized management.

INSTALLATION

1. Download the Proxmox VE ISO from the official proxmox website and flash the ISO to a pendrive.
2. Switch on the server and choose the bootable pendrive on the boot loader, follow the on-screen installation for installing proxmox.
3. After installing the server will reboot and provide an IP which is the Web UI IP.
4. Enter the IP address along with the port number to access the Web UI.



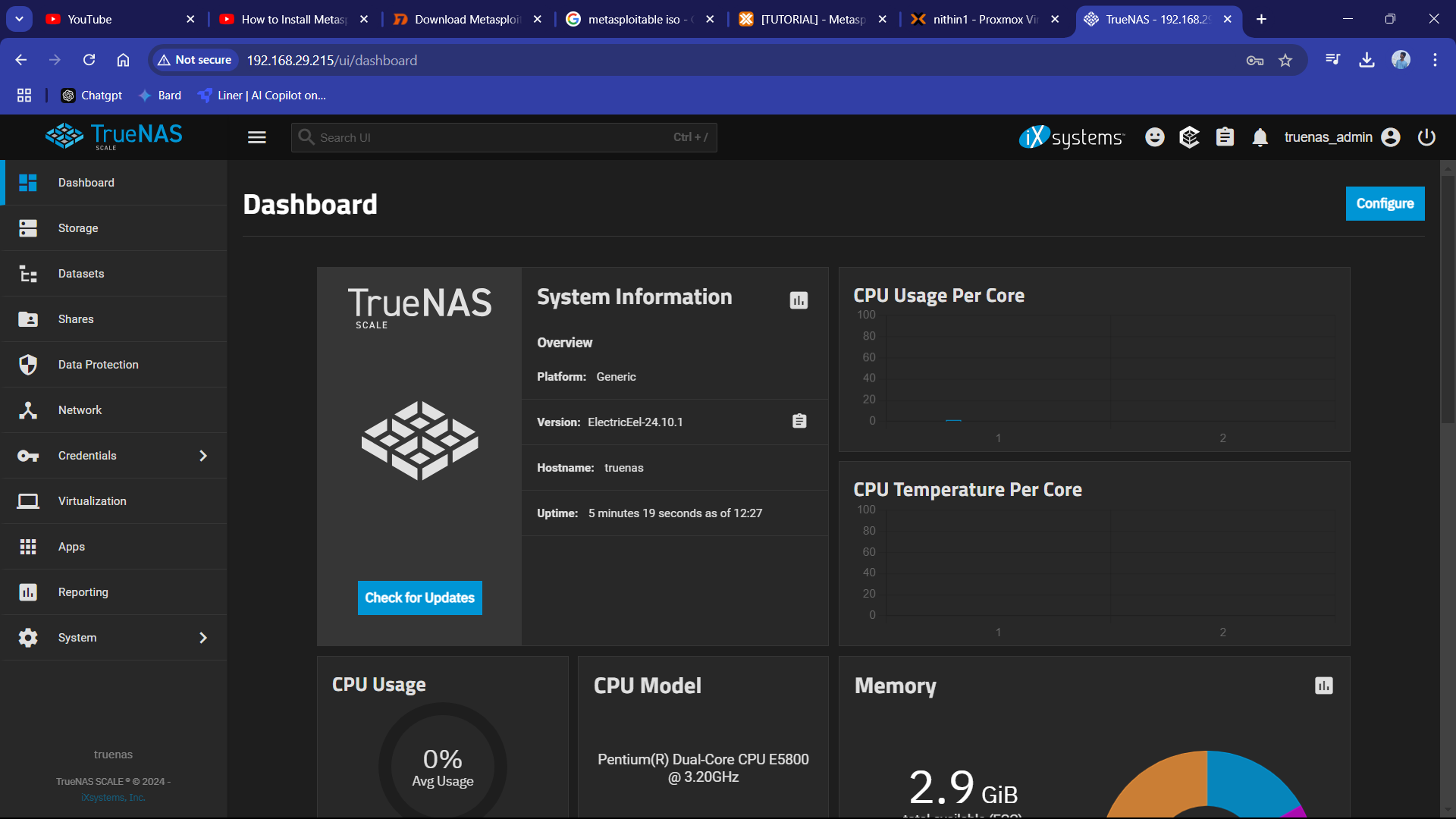
**Virtual Machine inside Proxmox**

TrueNAS:

TrueNAS is an open-source operating system designed specifically for building and managing Network Attached Storage (NAS) systems. It allows you to create a centralized storage solution for storing, sharing, and managing data across your network. TrueNAS is built on FreeBSD, a highly stable and secure operating system, and it uses the ZFS file system, known for its reliability, scalability, and advanced features.

I installed TrueNAS to set up Network Attached Storage (NAS), which provides my home with a centralized storage solution. The shared folders I create on TrueNAS can be accessed from any device connected to my network. TrueNAS also gives me the ability to set up user-specific access controls, allowing me to assign relevant permissions for accessing certain data.

Additionally, TrueNAS supports online streaming through Plex, enabling me to host my own movie library and stream content seamlessly across my network. This feature enhances my home entertainment experience by turning my NAS into a personal media server.

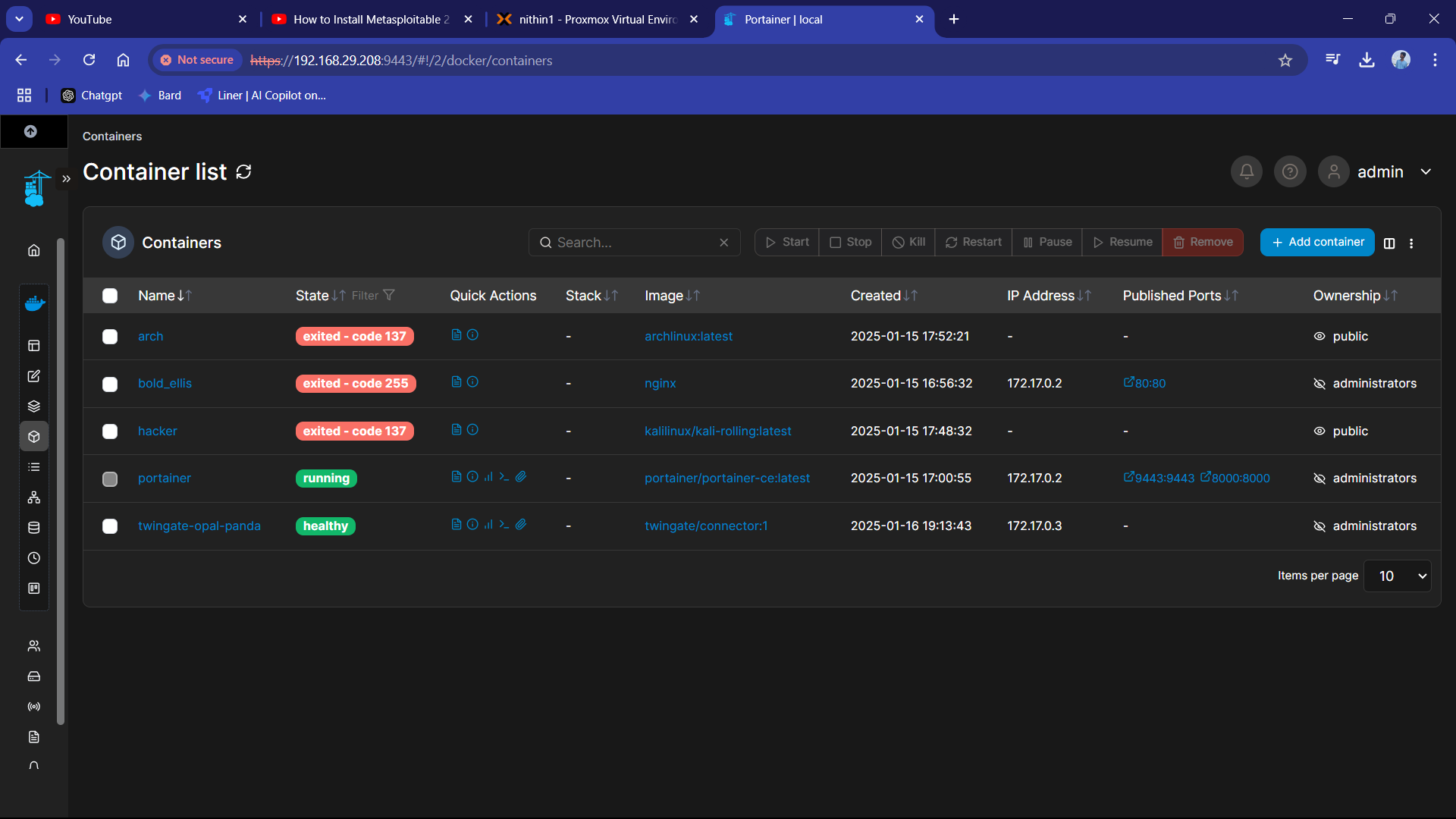


Docker On Ubuntu:

Docker is an open-source platform designed to simplify the creation, deployment, and management of applications using containers. Containers are lightweight, portable, and self-contained environments that include everything an application needs to run, such as code, libraries, and dependencies. Unlike traditional virtual machines, containers share the host system's kernel, making them faster and more resource-efficient.

To maximize Docker's efficiency and flexibility, I set up an Ubuntu server as a virtual machine (VM) in Proxmox and installed Docker on it. The Ubuntu server acts as a dedicated host for managing Docker, providing a stable and secure environment. Docker, in turn, handles the containers, allowing me to run and manage multiple isolated applications or services seamlessly.

This setup ensures that I can efficiently deploy, scale, and manage containers while keeping the host system optimized. I also deployed a Portainer container, which provides a user-friendly web UI for managing Docker. With Portainer, I can easily deploy, test, and configure various Docker containers through an intuitive interface, eliminating the need to rely heavily on the command-line interface (CLI). Additionally, Portainer offers a browser-based console for each container, allowing me to interact with and manage containers directly from the web UI. This greatly simplifies container management and enhances my workflow by making Docker more accessible and efficient.



Kali Linux:

Kali Linux is a free, open-source operating system designed for cybersecurity professionals and enthusiasts. It is based on Debian Linux and is widely recognized as one of the most powerful tools for penetration testing, ethical hacking, and digital forensics. Comes with hundreds of preinstalled tools for tasks such as vulnerability analysis, network scanning, password cracking, wireless network security, and reverse engineering. Examples Nmap, Metasploit, Wireshark, Burp suite, etc. Users can customize Kali Linux to fit their specific needs.

I installed Kali Linux as a virtual machine in my Proxmox environment, using it as an attack box for penetration testing. This setup allows me to simulate attacks on other virtual machines like Metasploitable, helping me learn about vulnerabilities and test security measures. Running Kali in a virtualized environment ensures safety and isolation, keeping my primary systems secure while experimenting.

Metasploitable:

Metasploitable is an intentionally vulnerable virtual machine (VM) designed for practicing penetration testing, ethical hacking, and cybersecurity skills in a controlled and safe environment. It is maintained by the creators of the Metasploit Framework, which is a powerful tool for developing and executing exploits.

I installed Metasploitable as a virtual machine in my Proxmox environment. It serves as a target for my Kali Linux attack box, allowing me to practice exploiting vulnerabilities and testing various security tools. By using Metasploitable in an isolated network, I can safely simulate real-world hacking scenarios and enhance my cybersecurity skills without affecting other systems.

**Remote Access**

Although having a homelab is fun and engaging, one limitation is that I cannot access it remotely from different locations. My lab is currently restricted to my local network. To make the lab accessible remotely, I would need to forward ports from my network and purchase a domain to access its contents. However, port forwarding requires a static IP address from my ISP, which isn't feasible on my current budget as my ISP uses CGNAT and does not offer static IPs affordably.

Considering these challenges, I explored alternative solutions and decided to use a VPN (Virtual Private Network). A VPN creates a secure virtual tunnel to my network, enabling me to connect to my lab environment remotely. After researching and watching numerous YouTube tutorials, I discovered Twingate, a modern alternative to traditional VPNs that simplifies secure remote access.

Twingate follows a Zero Trust security model, keeping private resources and internet traffic protected without the need for additional hardware. It’s built for the modern world of work and eliminates many of the challenges associated with traditional VPNs. I followed the official documentation to successfully set up remote access for my Ubuntu-Docker virtual machine.

Currently, I am working on figuring out the setup for remote access to my TrueNAS server. This step will further enhance the accessibility and utility of my homelab from anywhere

**Future Plans and Upgrades**

**Future plans**:

I aim to enhance my learning and hands-on experience by working on various homelab projects, including:

* Installing Pfsense on the server and configure my own
  + Firewall
  + DNS server
  + Content filtering for better control over internet traffic.
  + More and more.
* Setting up a Windows Server to delve into on-demand topics like Active Directory and centralized domain management.
* Installing Pi-hole to function as an ad-blocker for the entire network.
* Creating a malware analysis lab within Proxmox to safely study malicious software in an isolated environment.
* Setting up a DMZ (Demilitarized Zone) for added security and safe exposure of services to the internet.
* Implementing a CCTV camera system for home surveillance and further experimentation with IoT devices.

**Upgrades:**

Currently, my homelab consists of a single server connected directly to my router. To expand and improve its functionality, I plan to:

* Add networking equipment like access points and a network switch for better connectivity and scalability.
* Integrate additional computers into the setup to create a Proxmox cluster, enabling more efficient resource management and redundancy.

These future projects and upgrades will not only broaden my technical knowledge but also offer opportunities for research and development, helping me explore advanced topics in networking, security, and server management.

**Conclusion**

Building and managing a homelab has been an exciting and rewarding journey, offering endless opportunities for learning and experimentation. It has allowed me to explore real-world concepts in networking, virtualization, storage, and security while fostering creativity and problem-solving skills.

With future projects and planned upgrades, my homelab will evolve into a more advanced setup, enabling me to tackle complex challenges and gain hands-on experience with modern technologies. This endeavor is not just about creating a technical environment but also about developing a deeper understanding of how systems work together. Ultimately, the knowledge and skills gained from this journey will serve as a strong foundation for future growth in the field of IT and cybersecurity.

**Resources**

On building my homelab I crossed various of useful resources which I attach below:

1. <https://cybercademy.org/cybersecurity-homelab-project/>
2. <https://www.xda-developers.com/use-old-pc-dedicated-firewall-at-home/>
3. <https://www.joe0.com/2019/11/16/converting-single-nic-mini-pc-into-pfsense-router-firewall-by-using-virtual-lan-configuration-on-a-managed-switch/>
4. <https://www.linkedin.com/posts/activity-7231191585353293824-Zfst?utm_source=share&utm_medium=member_desktop>
5. <https://medium.com/@mouhamedjouhari007/active-directory-penetration-test-lab-e47853c5c0e1>
6. https://youtu.be/oOzihldLz7U?si=vH4o05KXFt4Pyi11