# Ultimate Docker Server: Getting Started with OS Preparation [Part 1] | SHB

Many people have emailed me after reading <u>Docker</u> and <u>Traefik</u> guides, asking how I prep my *Docker Server* OS before starting to build my stack. Or, what to do after setting up a Docker stack.

That got me thinking. It's time to revise my Docker and Traefik guides anyways. So why not do a full series of posts from beginning to end on how I build my Docker Stack for Home Server, Media Server, Web Server, DNS/Adblock Server, and Synology NAS.

Note that what I will cover in this series is my actual setup that I have continued to improve over 6 years, not some random tutorial based on theory.

So, I decided to put together this series:

#### Ultimate Docker Server Series:

This post is part of the Docker Server Tutorial Series, which includes the following individual chapters/parts:

- 1. <u>Ultimate Docker Server: Getting Started with OS Preparation [VIDEO]</u> [2024]
- 2. Docker Media Server Ubuntu/Debian with 60+ Awesome Apps [VIDEO] [2024]
- 3. ZeroTier <u>VPN</u> Ubuntu, Docker, Synology, Windows: Secure on-the-go access [coming soon]
- 4. Nginx Proxy Manager Docker Compose Guide: Simplest Reverse Proxy [coming soon]
- 5. Traefik Reverse Proxy
  - v3: <u>Ultimate Traefik v3 Docker Compose Guide: Best Reverse Proxy</u> [VIDEO] [2024]
  - v2: <u>Ultimate Traefik Docker Compose Guide: LE, SSL, Reverse Proxy</u> [2024]
- 6. <u>Authelia Docker Compose Guide: Secure 2-Factor Authentication [VIDEO]</u> [2024]
- 7. <u>Google OAuth Docker Compose Guide: Multi-Factor Authentication [VIDEO]</u> [2024]
- 8. Docker Security Practices for Homelab: Secrets, Firewall, and more
- 9. <u>Cloudflare Settings for Docker Traefik Stacks</u>

10.Implementing a Backup System for Docker Traefik Stack [coming soon]

11.<u>Automate Homelab Setup Deployarr: 60+ Apps in Minutes</u>

#### It is best to

follow the series from the beginning so you not only understand WHAT to do but also WHY we do it

This post covers all the basic and good-to-know information before starting the **Docker Server setup**.

# Automating the Docker Reverse Proxy Stack Setup

If you have not heard of <u>Auto-Traefik</u>, then you should consider checking out this post or watching the video series below:

Playlist: Auto Traefik 2 - Docker, Traefik, SSL, Authelia, and more in minutes

Auto-Traefik was launched as a perk to my supporters and to find a way to financially support what I do with this site.

With the launch of this video series, I am revising all my guides to align with what I have implemented with Auto-Traefik.

Everything that the Auto-Traefik Script does should be possible by following this series without paying for Auto-Traefik. But my hope is that you continue to support my work by becoming a member.

#### Be the <u>1 in 200,000</u>. Help us sustain what we do.

114 / 150 by Dec 31, 2024

You will gain benefits such as discord roles, exclusive content, ad-free browsing, raffles, and more.

## **My Environment**

Wait why do I need to read about some random guy's environment?

You do not have to. But I publish all my "real" setup on my <u>Github Repo</u>. Since I sync all Docker related configuration files between all m Docker hosts (more on this below), it is important to differentiate them with prefixes, suffixes, folders, etc.

Therefore, to follow my guides and GitHub repo, it can be beneficial to understand my environment.

Mini Homelab Tour - I do a LOT with this Little Proxmox Server and 100+ Docker Apps

## **My Docker Server Hosts**

I have 5 docker hosts. I sync all my Docker stacks using Syncthing and push the files to GitHub so I can share with the community.

Syncing also allows me to have a backup of one system's configuration file in all the other hosts.

For this reason, where applicable, I segregate or name files/folders with their hostname (for example: **<u>hs</u>** for Home Server):

- **docker-compose-hs.yml:** Docker Compose for <u>Home Server</u> on Ubuntu Server Proxmox LXC Container.
- **docker-compose-mds.yml:** Docker Compose for <u>Media/Database Server</u> on Ubuntu Server Proxmox LXC Container.
- docker-compose-dns.yml: Docker Compose for <u>AdBlock/ DNS Server</u> on Raspberry Pi 4B
- **docker-compose-ws.yml:** Docker Compose for <u>Web Server</u> on <u>Digital Ocean</u> <u>VPS</u>, which powers this website.
- **docker-compose-ds918.yml:** Docker Compose for <u>Synology DS918+</u> NAS.

This is the reason why even the <u>Auto-Traefik</u> script creates files and folders with hostname as suffix/prefix where applicable.

You may not have to Sync or push the files out to GitHub. Therefore, you may omit the hostname (adjust wherever applicable). But eventually if you plan to go towards a multi-host setup without getting into Docker Swarm etc., then my way could help.

## **My GitHub Repository**

In my GitHub Repo, you see the current working version of my Docker server stacks.

C htpcBeginner / docker-traefi	k Febre quarte 7 🖓 Décumente 🕥 Actions 🔘 Security	l≝ insgits			by ⊗ Sponur Q Pentheriese V Feet 343. ☆ Star 2.48 +		
	14 maater → 14 titrunch ⊙ 0 Tage		Q, Go to Sile	O Code +	About		
	🔯 NapoBegierner * Several repolation 2003_05_04. Sive full connent logi, 🗯		alfiddig i diwardwaigo	() 281 Carrows	Docker media and home server stack with Docker Compose, Traefik, Swarm Mode, Coppin Olumb 200 stacks, and sectorport		
	🖿 gittabi 🕴 t	and PUNDINGymt		2 years ago	Courge Constructional and Ecconcipal     www.struarditormbeginner.com/     darker interchoner     casest pice meansation robuse.		
	🖿 appdata -	Seiveral updates 2023_09_24	L. See full commit log.	4 months age			
	archives .	Several updates 2023,09,3	L See full coronit log.	4 months ago			
	auton .	Several updates 2021_02_M	L See full commit log.	4, months ago	sprotogy Booth, index Sadrastyr partiales jefyth (1) Doubtine		
	acripti *	Several updative 2023, 09, 24	4. See full commit log.	4 months ago			
	💼 socrets_example	Several updates 2022_07_14	L See full commit log.	it log Zylars ago 🤁 MiT license			
	🖬 shared/config 👘	Several updates 2023,09,34	. See full committing	d months ago	Ap Activity		
	🗅 envasample	Several updates 2022_06_20	. See full commit log.	2 участ заро	<ul> <li>(b) success</li> </ul>		
	D atorns .	Several updation 2025,00,24	L See Full commit log.	4 months age	Y ses toks		
	D CHANGELOG and	Several updates 2023 05 24	See full committion	8 months ago	sebou istratio.		

This Docker series is fully based on my GitHub repository.

My setup constantly evolves. It is nearly impossible to keep updating my guides to keep up with all the updates to my setup. But this series should give you a good foundation to implement any future Github repor changes or delayed guide-updates.

## Using the Docker Compose from My Github Repo

Even though all Docker files from my hosts are all combined and pushed to the repo, the compose files are all located in the folder called, well, **compose**.



The **archives** folder has all my old/unused Docker Compose files, the **hs** folder has all my home server Docker Compose YMLs, the **mds** folder has all my media/database server Docker Compose YMLs, and so on.

The docker compose examples can be copy-pasted as-is from one file to another, in most cases. In fact, you might even find the same the same service on multiple hosts (e.g. Traefik, Socket Proxy, Portainer, etc.).

## **Typical Target Users**

If you are reading this, there is a high chance that you are trying to setup your Home Media Server.

A Home Media Server is a server located in your home network that acts as a central data storage and serving device. It is a key part of most "homelabs".

Typically, a home server is always on, has tons of storage capacity and ready to serve files (including media) when the need arises. We have covered several home server topics in great detail in the past. If you do not yet have a home server or are considering building one, then read this summary on the <u>most</u> <u>common NAS or Home Server uses</u>.



Topton V700 Minipc From Aliexpress - \$481 - Proxmox Server

If you are ordering from AliExpress, remember 3 things: 1) Check with the seller if the item is in stock first, 2) Pick only sellers with a very high rating (>95% at least), and 3) Be prepared for delays/extended delivery times (in my case the seller asked for extra shipping time due to availability).

In my case, my Proxmox Server runs my Home Server, Media/Database Server, and AdBlock/DNS Server as Ubuntu 22.04 LXC Containers. I recently upgraded to the <u>TopTon V700 Mini PC</u> as Proxmox Host (for just \$481, tax included) and it's been killing it:

- TopTon V700 Mini PC with Intel 13th Gen "Raptor Lake" i7-13800H \$481
- 2x32GB Crucial DDR5 4800MHz SO-DIMM RAM \$163
- 2x2TB <u>Crucial T500 PCIe Gen4 M.2 NVME Drives</u> in ZFS RAID \$230
- 4TB<u>Crucial MX500 SATA III SSD</u> for non-critical data, cache, etc. \$200

The principles discussed in this Docker Server series also apply to the setup of any Docker Stack, including web servers. For simplicity, I will generically refer to them as "**Docker Server**". [**Read:** <u>5</u> <u>Best Mini PC for Proxmox Home Server [2024]</u>]

# **Choosing Right OS for Docker Server**

Let's start with choosing the best operating system for your Docker stack. We have a separate article that discusses all the <u>best home server OS option</u>s available for you.

In my opinion, Ubuntu Server (or Debian Server) is the best one to choose if you like building things from scratch and learning along the way. Maybe I am biased due to my 14 years of experience with Ubuntu, but it won't be a wrong choice.

An even more ideal setup is to run a Type 1 hypervisor and virtualize your Docker server in a container or virtual machine. [**Read:** <u>Proxmox vs ESXi: 9 Compelling reasons why my choice was clear</u>]



With VMware free version and perpetual licenses going away, this is the best time to make the switch Proxmox VE.

Having said that, the docker compose examples in this guide will also work on Network Attached Storage devices such as Synology or QNAP:

- <u>Ultimate Synology NAS Docker Compose Media Server</u>
- <u>Ultimate QNAP Docker Compose and Container Station Guide</u>

# **Preparing Ubuntu/Debian for Docker Server**

At this point, I will assume that you have a freshly installed Ubuntu/Debian operating system with nothing done after.



## 1. Create a New User

Some operating systems allow the creation of a non-root user during installation and some do not. If you did not create a primary user during setup, use the following commands to create it:

1 sudo adduser anand

2 sudo adduser anand sudo

With the first command, we are adding a new user called "anand". The second command I am adding "anand" to the "sudo" group, so I have the privilege to use **sudo** command.

## 2. Update the OS

Most fresh OS installations lack all the security and package updates. So, let's take care of that next.

Run the following commands to refresh the packages list and install any updates.

1 sudo apt update

2 sudo apt upgrade

## 3. Secure SSH Access

Until this point, you are probably physically in front of the server or a web-based console offered by the host/provider (e.g. Proxmox or <u>VPS</u> provider such as <u>Digital Ocean</u>).

Let's enable SSH access. Edit /etc/ssh/sshd\_config using the following command:

```
1 nano /etc/ssh/sshd_config
```

Find the port line, uncomment it by removing the # in front (if it has one) and change the port number from default 22 to something random (e.g. 2053).

Restart SSH server using the following command:

```
1 sudo systemctl restart sshd
```

Note that changing port number is the absolute minimum if you are going to open the SSH port to the internet. Using key-based authentication instead of passwords, specifying the allowed users list, disabling password authentication, etc. are strongly urged - not discussed in this series.

In addition, an intrusion prevention system such as CrowdSec + Host Firewall Bouncer is also recommended (you may do this using our CrowdSec Docker guides, after completing this series).

#### How-To Series: Crowd Security Intrusion Prevention System

- 1. <u>Crowdsec Docker Compose Guide Part 1: Powerful IPS with Firewall Bouncer</u>
- 2. <u>CrowdSec Docker Part 2</u>: Improved IPS with Cloudflare Bouncer
- 3. <u>CrowdSec Docker Part 3: Traefik Bouncer for Additional Security</u>
- 4. CrowdSec Multiserver Docker (Part 4): For Ultimate Protection

## 4. Install Basic/Required Packages

At this point, you may SSH into your server and continue remotely if you prefer. Let's install some basic/required packages (based on my experience).

<sup>1</sup> sudo apt install ca-certificates curl gnupg lsb-release ntp htop zip unzip apt-transport-https net-tools ncdu apache2-utils

**zip** and **unzip** are for compression, **net-tools** is to check port usage and availability, **htop** provides a nice UI to see running processes, and **ncdu** helps visualize disk space usage.

## 5. Perform Server Tweaks

A few final system configuration tweaks to enhance the performance and handling of large list of files (e.g. Plex/Jellyfin metadata). Edit /**etc/sysctl.conf** using the following command:

1 sudo nano /etc/sysctl.conf

Add the following 3 lines at the end of the file:

```
1 vm.swappiness=10
```

```
2 vm.vfs_cache_pressure = 50
```

3 fs.inotify.max\_user\_watches=262144

Save and exit by pressing **Ctrl X**, **Y**, and **Enter**. That is all the prep work I do before I start building my **Docker server stack**.

## 6. Enable Firewall

Ubuntu/Debian systems come with a built-in firewall called UFW (Universal Firewall). It is disabled by default.

Let's start by adding some default policies (deny all incoming and allow all outgoing) and then allow incoming connections only from your local network (e.g. 192.168.1.0/24). Customize the network subnet based on your situation.

```
    1 sudo ufw default deny incoming
    2 sudo ufw default allow outgoing
    3 sudo ufw allow from 192.168.1.0/24
```

This will allow all connections from devices in the IP range 192.168.1.1 to 192.168.1.254.

This is a good starting point. But as we go through this series, we will keep modifying these firewall rules.

Then let's activate UFW using the command:

Finally check the status of UFW using **sudo ufw status**, you should see that it is active with the rule we added above:

[AnandSHB@mds: Status: active	~]\$ su	do ufw status	
То		Action	From
Anywhere		ALLOW	192.168.1.0/24

If you have decided to follow this series and do things the way I do it, great. **Keep in mind that my way is not the only way or even the best way.** 

But in the last 5 years or so of hosting my Docker Traefik repository (<u>2.4k stars</u> until Jan 2024) and the guides, I have yet to receive feedback saying my way sucks. So, I feel quite confident that it will meet most typical user's needs.

In addition, this website itself runs on the Docker web server stack in a production server on Digital Ocean. It has been performing well in handling the load and keeping everything secure.

If you do other things on your OS before getting started, <u>be sure to share with the rest of us</u> in the comments below.

That said, there may be other better ways to accomplish the same thing. Feel free to share them in the comments. With this series, I hope to teach my readers how they can replicate my Docker server stacks. So, let's begin the ride.