

## Curriculum Vitae

# Marty Sluijtmán

Location: Tilburg, Noord Brabant, Netherlands

Date of Birth: 2001-06-13

[voidcruiser.nl](https://voidcruiser.nl) | [docs.antimattercloud.nl](https://docs.antimattercloud.nl)

[forge.antimattercloud.nl](https://forge.antimattercloud.nl)

[GitLab.com/EternalWanderer](https://GitLab.com/EternalWanderer)

Email: [marty.wanderer@disroot.org](mailto:marty.wanderer@disroot.org)

## :: Introduction

---

I've been aware of Linux from quite a young age. In my household there has always been at least one computer running Ubuntu. When I was around 18 years old, I decided to properly check it out. I installed it on a USB thumb drive to try it out without having to commit. It took me a week of running Ubuntu like this before I shrank down my Windows installation to make room for a Windows 10/Ubuntu dual boot configuration. I ended up using Windows only to run a few games that I couldn't get to work on Linux no matter what I tried.

When I started to work with Linux, it was only a matter of time before I would start looking into the Free and Open Source movement and from there developing my own software more seriously.

## :: Education

---

2018-2023      **SiNTLUCAS - Game development** Here I learned the basics of C# and the Unity Game engine. During this time I also discovered my passion for Linux; teaching myself the basics of Linux systems administration in my free time.

## :: Work Experience

---

May 2022 - April 2023      **Internship @ Tilburg University under LIS UNIX Infra.** Here I worked on the management of 300 Debian GNU/Linux servers; In particular I did the following:

- wrote a hand full of Nagios monitors in POSIX Shell
- wrote a set of tools to ensure both backup solutions stay in sync with the desired state in POSIX Shell and GNU Make.

Here I also had the opportunity to participate in the 4 day Kubernetes for administrators training from the good folks over at Fullstaq.

May 2024 - November 2024      **Sysadmin @ TriNed/Fiberoperator** Here I worked on the following projects:

- set up the basis of a Kubernetes cluster using Talos Linux in VMWare VX Rail
- wrote several puppet classes facilitating docker, docker-compose; utalizing these to set up Vaultwarden and GitLab instances.
- migrated said GitLab instance from the docker container to directly on the VM when having a separate SSH binary in the container caused issues.
- helped with the transition to puppet r10k.

## :: Skills

---

### :: Technologies

**Operating Systems** : Debian & derivatives, Alpine Linux, NixOS  
**Container Technologies** : Docker, Kubernetes, Nix Containers  
**Provisioning Tools** : Nix, Caspar, Puppet  
**Webservers** : Apache2, Nginx  
**VCS** : Git, Subversion

## :: Languages

---

### :: Spoken Languages

**Dutch** : Native  
**English** : Fluent

### :: Computational Languages

**Programming** : Haskell, Go, C#  
**Scripting** : POSIX Shell, Nix, Nushell  
**Markup** : Graphviz/Dot,  $\text{\LaTeX}$ , HTML, CSS  
**Misc** : GNU Make

### :: Less proficient

**Scripting** : JavaScript, Python, Lua, Elm

In my free time during and since SiNTLUCAS I picked up Go and started to learn Haskell because both languages seemed interesting to me. I wrote a few projects in Go. Since starting to learn Haskell, I've decided to slowly rewrite those Go projects in it alongside writing every new personal project in Haskell as well.

## :: Hobby Projects

---

### :: SSH

Nearly all interactions with my personal infrastructure happen over SSH. In this regard, I'm a big proponent of only using SSH keys to log in to server.

### :: Personal Websites

I host two personal websites generated using the Hugo static site generator. One of them is a wiki of sorts written with the goal of explaining several concepts relating to digital privacy and security. The topics on this site are topics I found myself frequently explaining to friends. The goal for this website being to refer people to this website, rather than to spend time verbally explaining it to them, thus saving time and serving as a reference.

<https://docs.antimattercloud.nl>

The other serves as my personal blog.

<https://voidcruiser.nl>

The source code for my blog can be found [here](#) and the Hugo theme can be found [here](#).

## :: Nix Flakes

Since I switched to NixOS, I decided it would be a good idea to port (some of) my packages over to Flakes. From there it kind of grew out of proportion and nearly everything I do is inside of a Flake these days, including this curriculum vitae. The Flake and repository can be found [here](#).

In fact, this document can be compiled on any system running Nix with Flakes enabled using the following command:

```
nix build git+https://forge.antimattercloud.nl/Egg/curriculum-vitae
```

## :: Lambda Calculus, Combinatory Logic & adjacent mathematical topics

When I started my Haskell journey, it was only matter of time before I decided to look at Lambda calculus. From here, I quickly found out about the [APL](#) programming language. Fast forward a few weeks and from APL, I've discovered combinatory logic and the [BQN](#) language; and I'm very slowly learning to apply combinators to my day to day problems in Haskell and otherwise.

This has started me on a journey of wanting to understand the rest of the mathematics Haskell and APL are built upon.

## :: Headscale as VPN

I run a Headscale instance to connect to my private infrastructure because I

1. am too much of a cheapscale to pay for tailscale
2. don't have- and don't want to have a personal Google account

I use it on basically all of my devices. Authentication is handled either through the CLI or my Keycloak instance.

## :: Step CA as Certificate Authority

There are some servers that refuse to work without an SSL certificate that I would like to be able to run inside of my VPN, so I set up my own certificate authority using Step CA. Thanks to it, I can point my ACME daemon to the corresponding URL to automatically request a certificate when needed.

## :: Dungeons & Dragons tools

I play Dungeons & Dragons every now and then with a group of friends. Traditionally, Dungeons & Dragons is in quite a literal sense a 'pen and paper RPG'. Personally, I got tired of having to deal with an ever-growing pile of paper and decided to see if there were any existing digital tools. While there are plenty, none of them *quite* suited my needs, so I wrote my own in Go.

- [dice-roller](#)
- [sheet-parser](#)

I am in the process of rewriting both of these in Haskell for the intellectual exercise. The repository can be found over on [my Forgejo instance](#).

## :: Radio

I have an internet radio playing music 24/7 with a publicly available frontend. It makes use of MPD to play the music; Icecast2 to manage the stream and buffering; and Nginx manages the reverse proxies to <https://antimattercloud.nl/radio> and <https://radio.antimattercloud.nl/>

## :: Keycloak

I host a Keycloak instance for SSO and MFA authentication for the various services I run privately. Next step for me is to try to host an LDAP server to see how integration with that might work.

## :: Git

I have a collection of privately hosted Git repositories on my home server. The goal here being to both get a better understanding of how Git works, as well as not having to depend on large, pre-existing infrastructure to host my personal repositories.

## :: Forgejo

I host a Forgejo instance for when I want to have some extra features around my git repositories and/or when I want to share a project with someone without having to give them either ssh access to a server or a tar file containing the entire project.

Authentication is handled using Keycloak

## :: ownCloud

I used host a private ownCloud instance for family and friends.

## :: ownCloud Infinite Scale

Some time later, I found out about ownCloud Infinite Scale, which I now host an instance of over at [scale.antimattercloud.nl](https://scale.antimattercloud.nl).

Authentication and authorization is handled using Keycloak.

## :: Matrix server (synapse and conduit)

Currently I host a Matrix server using synapse. It is connected to my Keycloak instance. I don't use it much. I mainly set it up because I wanted to have something else to hook up to my Keycloak. As a result, authentication and authorization are handled using Keycloak.

## :: XMPP server (prosody)

I used to host an XMPP server for a few years but it ended up underutilised.

## :: SearXNG

I run a lightly customised SearXNG instance over on <https://voidcruiser.nl/searx> in a docker container. I use this as my daily search engine.

## :: Phone Customisation

None of my phones still run their stock operating system. Most of the run LineageOS, an Android ROM that – by default – doesn't have any Google applications or services. Those that support it run GrapheneOS, a hardened Android fork which goes above and beyond.

## :: Owncast

I host an Owncast instance on a NixOS server to be able to stream gameplay every now and then for friends.

## :: Alpine Linux packages

Before moving everything over to NixOS, I maintained a few Alpine Linux packages in a self-hosted repository.

## :: Home Automation

A friend mentioned he uses Home Assistant to manage a collection of Zigbee compatible devices. This got me started on my home automation journey. So far I have a few lamps connected to a home-assistant instance running on my home server inside a podman container.

## :: (Creative) Writing in $\text{\LaTeX}$

Every now and then a story find itself in my consciouness. Usually I write the first draft of these stories in CommonMark, quickly transitioning to  $\text{\LaTeX}$  when I want to add more visual flair than CommonMark permits.

In a similar vein, every document that I write with the intention of having it be read by other people not within my bubble of the tech space is written in  $\text{\LaTeX}$ .

## :: Desktop Customization

I've been running Linux exclusively as my desktop OS since 2020. As a result of this I have produced quite a lot of configuration. Of which the crown jewel is my (at the time of writing) 11000 line NixOS mono repo spanning 7 machines – 5 of which are still in use. Within this repository, I am the most proud of the 1349 lines dedicated to my XMonad configuration – roughly 250 of which are Nix; the other 1100 are Haskell. As a result of this, it does as close to exactly what I want as I think is possible for a window manager.