Théo BORI

Contact Information

theobori@disroot.org
theobori.cafe
github.com/theobori
linkedin.com/in/theo-bori
repology.org/maintainer/theobori@disroot.org

Profile

Passionate about the UNIX ecosystem and the open-source world since my teenage years, I regularly contribute to community projects. I value knowledge sharing and rigorous technical documentation.

Education

Epitech, Montpellier, Paris, France — *September 2020–October 2025*

- *Expert Degree in Information Technology*

Technological University Dublin, Dublin, Ireland — *October 2023–August 2024*

- *Degree in Data Science*

Professional Experience

DevOps & Cloud Engineer Intern, Thales AVS, Mérignac, France — *March 2025–August 2025*

- Deployment of GitLab Workspaces in a Kubernetes cluster managed by Azure (AKS).
- Creation of Python projects that automate security verification of IT projects in the software forge and help them comply with company best practices. These projects are launched from GitLab CI/CD pipelines and produce detailed security reports with scores and intuitive graphics.
- Development of a syntax analysis algorithm to process data produced by functional test execution and optimizations through multi-threading.
- Use of AI programming agents, including CodeCompanion, to analyze and secure the source code of test execution software.

Site Reliability Engineer Intern, VINCI Autoroutes, Vedène, France — *September 2024–March 2025*

- Creation of Azure resources followed by deployment of Backstage.io with AKS.
- Integration of existing users and groups in Microsoft Entra ID, implementation of authentication with Azure, securing access with an Application Gateway followed by Azure Firewall, then integration of software factory services into Backstage.io.
- Creation of a Python project that produces a CLI to simplify the addition of existing projects in Backstage.io.

Site Reliability Engineer Intern, VINCI Autoroutes, Vedène, France — *April 2023–July 2023*

- Automated deployment of resources in Azure, NeuVector (AKS) and its configuration with Terraform. Monitoring was set up with Grafana's Prometheus, and the entire solution implementation was fully documented.
- Contributions to NeuVector's public source code and to NeuVector's Prometheus exporter.
- Creation of a library in Go to interact with the NeuVector Controller as well as a Terraform provider in Go that manages the lifecycle of NeuVector objects.

Notable Personal Projects

Website and Services — *September 2022–Present*

- Website where I share things I find interesting.
- Provision of free services open to everyone. They are all containerized with Docker and accessible behind NGINX, deployment is automated with an Ansible Playbook and DNS records are managed with a Terraform project. Prometheus and Grafana are used to monitor

service and system status and automatic backups are performed weekly. Service authentication is managed by an OpenLDAP server and blocking mechanisms have been implemented for malicious authentication attempts. All technical documentation and user statistics are publicly available.

- Maintenance for over a year of an organized online knowledge base that provides all my notes.

Application Packaging *March 2024–Present*

- Regular contributions to the official Nix(OS) expression collection (over a hundred in 2025) and service modularization.
- Porting of X11-compatible applications and games for OpenBSD.
- Portages of Plan 9 programs for UNIX.

Algorithms and Data Structures *December 2023–Present*

- Strong theoretical and practical foundations in algorithms and data structures.
- Solving algorithmic problems on various dedicated platforms and completing the Advent Of Code.

Packet Filtering with Linux *January 2024–November 2024*

- Development of an XDP project in C to filter packets in kernel space. The goal is to have a high-performance packet filter loaded in kernel space, with the ability to manage custom filtering rules for different network layers from user space.

Lox Interpreters *March 2024–November 2024*

- Implementation of interpreters for the Lox language from scratch with different parsing and code evaluation methods, including a Tree-Walk interpreter in Python and a bytecode virtual machine in C.

Languages

French, native language

English, spoken, read and written