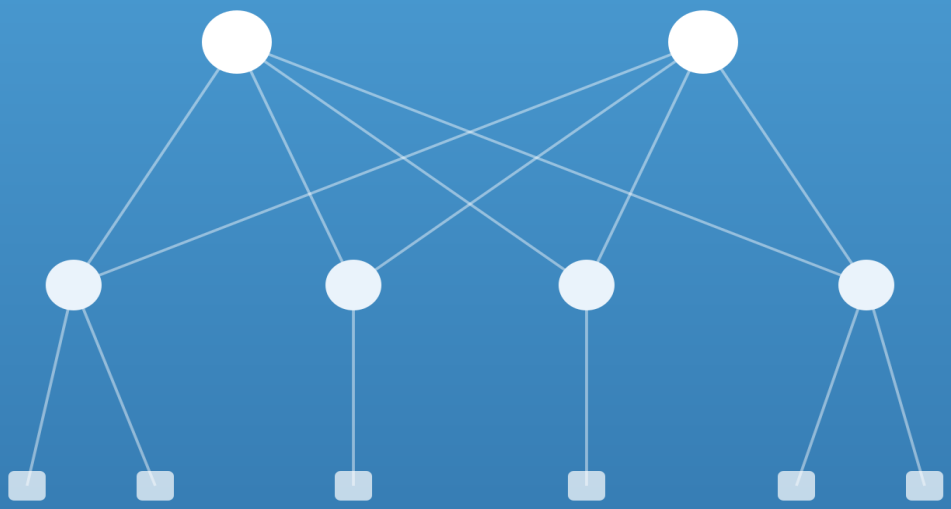
SECOND EDITION

Open Source Network Management

A hands-on guide to a modern, open source management stack.



Josh VanDeraa

Network · Source of Truth · Secrets Management
Automation · Metrics · Logs · Visualization & Alerting

Open Source Network Management

2nd Edition

Josh VanDeraa

2026

Open Source Network Management

A Free Sample — 2nd Edition

This is a free preview of *Open Source Network Management, 2nd Edition* by Josh VanDeraa. It includes the complete Chapter 1 along with the full table of contents for the book.

The complete book walks you through building an entire open source network management platform — source of truth, secrets, automation, metrics, logs, dashboards, and alerting — all running together on your own infrastructure, on your own terms.

Get the full book at **osnm.josh-v.com**.

What's in the full book

Plus appendices and companion lab files.

Chapter 1

Introduction

1.1 Mission

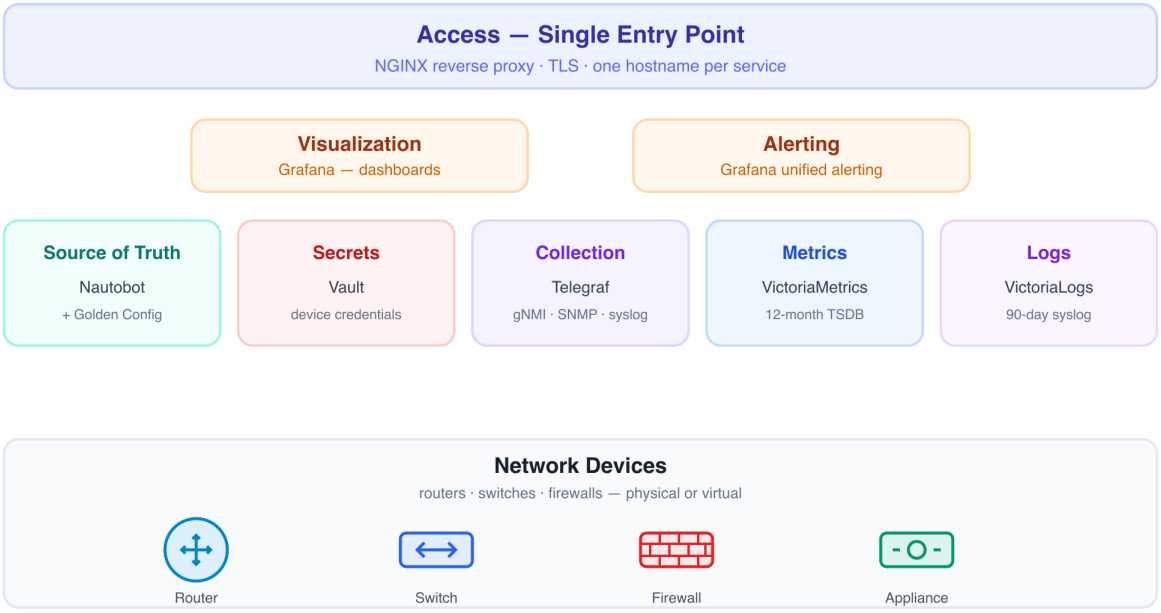
This book exists to show that open source is a practical proof of concept and proof of value path for modern network management, without starting with expensive commercial licenses. By combining best-in-class open source projects, any network team can stand up inventory, configuration management, metrics, logging, and visualization on their own terms. That gives you a low-friction way to validate the workflow, test the operational fit, and demonstrate business value before making a larger platform decision.

I'll walk you through setting up a modern network management solution using only open source software. This is a **getting started guide**. There are considerations that come along with scaling beyond a single node. Survivability, HA, and deployment variations based on your environment are all real concerns, but they're problems you'll be well-equipped to tackle once the foundation is solid. Once these capabilities become part of day-to-day operations, that's the time to evaluate whether a SaaS platform, commercial support, or a mix of both is the right long-term operating model for your team.

Much has changed since the first edition, but some things remain the same. Nautobot is still the foundation of the Open Source Network Management framework. I'm continuing to use Docker for all tool deployments. For network device simulation, I've moved from CML/GNS3 to Containerlab and Clabernetes.

1.2 What We're Building

This book builds a complete network management platform, one layer at a time. Before diving into any individual tool, it helps to see the full picture.



ONSM Architecture

The architecture has seven capability layers:

Source of Truth: The foundation. Every other layer pulls context from here: device inventory, IP addresses, site data, circuit records. If the Source of Truth doesn't know about a device, the rest of the stack can't manage it.

Secrets Management: Holds credentials, API keys, and certificates. Every service in the stack that needs a password gets it from here instead of a config file.

Automation: Backs up device configs, compares them against a known-good baseline, and flags or remediates drift.

Metrics: Collects time-series data from network devices and infrastructure, stores it efficiently, and makes it queryable.

Logs: Collects syslog, device events, and application logs. Keeps them searchable and correlated with your metrics.

Visualization and Alerting: Dashboards and alerts built on top of your Metrics and Logs layers. When something goes wrong, this is where you find out.

Network: The devices being managed: routers, switches, firewalls, and access control systems. Everything above this layer exists to manage what's down here.

This book takes an opinionated path. Each chapter covers one capability layer and the specific tool I've chosen to implement it. I'll tell you why I chose it and what tradeoffs I made. If you want to swap out a tool (use a different metrics storage backend, a different

secrets manager, a different SOT) you'll find alternative chapters at osnm.josh-v.com. The architecture stays the same. The tools are interchangeable.

The chapters follow the stack from the foundation up: Docker first, then Source of Truth, then each service layer, finishing with Visualization and how it all comes together.

1.3 Building vs Buying

Within this getting started guide you'll see how to use the open source tools available. Open source gives you room to run a proof of concept, turn it into proof of value, and decide what belongs in your operating model before you lock yourself into a long-term platform choice. But here's what I've learned: you get more value building new capabilities than you do doing day-to-day support of the platform. Once these tools become part of your ongoing operations, I encourage you to evaluate a SaaS platform, commercial support contracts, or both, so your team is not solely responsible for running and supporting everything themselves.

1.4 Network Automation and Network Management

This book is focused on network management tools. I'll also include some automation tooling. Modern network management tools must interact with today's network automation tooling: Python, Ansible, and the rest. That integration is what cuts down the time and effort to get up and running.

New in this edition: I'm using AI agentic workflows to review, update, and verify the automations in this book. The agents aren't writing it. I am. You can read more about the setup at osnm.josh-v.com, where I'll also post additional resources as the book evolves.

1.5 Environment

I'm running the tooling on two Ubuntu 24.04 LTS VMs on a Proxmox 8.4 hypervisor:

- **osnm**: the management stack. 4 vCPUs, 8 GB of RAM, and 100 GB of storage. Every tool in this book runs here as a Docker container.
- **clab01**: the network lab. 8 vCPUs, 32 GB of RAM, and 100 GB of storage. This VM runs the Containerlab topology of Arista cEOS devices that the management stack will manage. Simulated network operating systems are memory-hungry, which is why this VM gets the larger allocation.

You could combine everything onto a single larger VM, but separating the management stack from the simulated network keeps the resource profiles independent and mirrors a real deployment: the management tools live somewhere other than the network they manage. If you'd rather run on dedicated hardware, a small form-factor mini PC with similar combined specs works just as well.

1.6 About Me

Not long ago, I wasn't using any open source software in my day-to-day. Everything had a service contract I could call for help. I wanted to focus on the networking aspect. That all changed once I started to get into network automation and programmability around 2015. I had my eyes opened up to what the world of automation could bring and how easy it was to get started.

Several times I attempted to run Linux and install open source tools. Back then, it meant compiling software, chasing dependencies, and jumping through hoops. I wasn't successful, and it put me off open source for a while.

That's changed completely. Installing these tools is straightforward now, and I'll show you the quickest path: Docker Engine. The availability of container images for every tool in this stack has made the open source network management story dramatically simpler.

1.7 Disclaimer

The configurations, scripts, and recommendations in this book are provided for educational purposes only. The author is not liable for outages, data loss, security incidents, or any other damages resulting from following the instructions in this book. Readers are responsible for testing all configurations in non-production environments before deploying to production. Use the contents of this book at your own risk.

For production deployments, consult with your organization's security and operations teams and consider obtaining support contracts from the respective open source vendors.

By the time you finish this book, you'll have a working network management stack on a single VM. You'll have full management and strong monitoring for your environment.

1.8 Licensing of Products

All license information unless specifically stated was gathered on 2025-03-22.

1.8.1 Ansible

License: GNU GPLv3[1]

1.8.2 Nautobot

License: Apache License 2.0[2]

[1] https://github.com/ansible/ansible/blob/devel/COPYING
[2] https://github.com/nautobot/nautobot/blob/develop/LICENSE.txt

1.8.3 HashiCorp Vault

License: Business Source License 1.1 (BSL 1.1)[3]

> HashiCorp changed Vault's license from Mozilla Public License 2.0 to BSL 1.1 in August 2023. Under BSL 1.1, Vault remains free to use but cannot be offered as a competing hosted service. This change applies to Vault 1.14+ and is discussed further in the Secrets Management chapter.

1.8.4 InfluxData Telegraf

License: MIT License[4]

1.8.5 VictoriaMetrics

License: Apache License 2.0[5]

1.8.6 Grafana

License: GNU Affero GPLv3[6]

> Grafana has been licensed under AGPL v3 since version 7.0. The AGPL requires that if you modify Grafana and provide it as a network service, you must make your modifications available under the same license.

1.8.7 Containerlab

License: Apache License 2.0[7]

1.8.8 Docker Engine

License: Apache License 2.0[8]

1.8.9 NGINX

License: BSD 2-Clause License[9]

[3] https://github.com/hashicorp/vault/blob/main/LICENSE
[4] https://github.com/influxdata/telegraf/blob/master/LICENSE
[5] https://github.com/VictoriaMetrics/VictoriaMetrics/blob/master/LICENSE
[6] https://github.com/grafana/grafana/blob/main/LICENSE
[7] https://github.com/srl-labs/containerlab/blob/main/LICENSE
[8] https://github.com/moby/moby/blob/master/LICENSE
[9] https://github.com/nginx/nginx/blob/master/docs/text/LICENSE

1.8.10 Victoria Logs

License: Apache License 2.0[10]

> Victoria Logs is part of the VictoriaMetrics project and shares the same license.

[10] https://github.com/VictoriaMetrics/VictoriaMetrics/blob/master/LICENSE