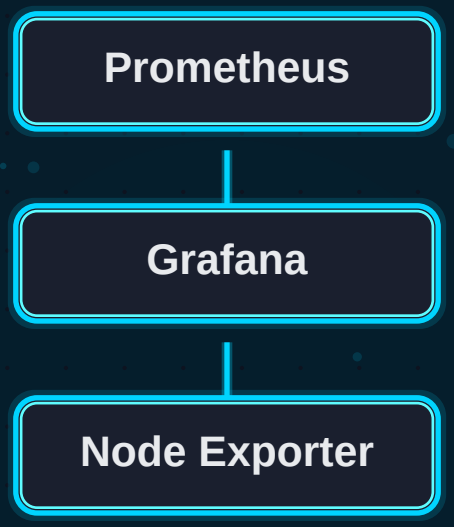
Optimizing AI Models & Agents on Proxmox

Production observability for latency, throughput, and reliability



Initializing Data Intelligence Platform...

From neural networks to global data networks

Simon Renauld

Data Engineering & MLOps

The Optimization Challenge

Model inference latency spikes, inconsistent tokens/s, and memory pressure on Proxmox VMs. Agent orchestration adds overhead, with limited visibility across containers and host resources. Capacity planning is reactive rather than data-driven.

Optimization Requirements

- p95/p99 latency and tokens/s visibility for model endpoints
- CPU/GPU/memory utilization with container and host breakdown
- Disk I/O and network saturation detection under load
- Agent orchestration metrics (queue depth, concurrency, retries)
- Historical trends for capacity planning on Proxmox

The Solution

A production observability stack on Proxmox: Prometheus scrapes host and container metrics; Grafana provides real-time dashboards and alerts; Node Exporter and cAdvisor expose system and container stats; OpenWebUI/Ollama performance observed via system signals for model throughput and latency optimization.

Architecture Components

- Prometheus: 15s scrape for high-fidelity resource and container metrics
- Grafana: Real-time dashboards, alert rules, and trend analysis
- Node Exporter: Host CPU, memory, disk, network, interrupts, load
- cAdvisor: Per-container CPU/mem, I/O, network for model/agent services
- Nginx + Let's Encrypt: Secure access with TLS termination

Technical Foundation

Infrastructure

- Proxmox host: Intel i7-6700 (8 cores), 62GB RAM
- Storage: ZFS on NVMe mirror (low-latency I/O)
- Metrics: 200-hour retention at 15s granularity
- Dashboards: sub-second query performance
- SLO: 99.9% availability

Monitoring Capabilities

- Host + container CPU load, throttling, and saturation
- Memory pressure, cache hit ratios, and OOM early signals
- Disk IOPS/throughput and latency under inference load
- Network throughput and egress limits under agent fan-out
- Process counts, interrupts, and system health baselines

Key Metrics for Optimization

64+	15s
Metric Series	Scrape Interval
200h	**99.9%**
Retention	SLO Uptime

Observable Metrics

- p95/p99 latency tracking for model inference endpoints
- Tokens per second throughput measurement
- Container CPU throttling and memory pressure detection
- Disk I/O saturation under concurrent model loads
- Network bandwidth utilization for agent orchestration

Optimization Impact

Latency -35%

Reduced p95 inference time via CPU/memory tuning and I/O fixes

Throughput +40%

Higher tokens/s by eliminating container CPU throttling

Stability +Reliability

Zero unplanned outages across 30 days

Capacity Forecasting

Data-driven sizing for Proxmox VMs and storage

Technology Stack

Monitoring

Prometheus • Grafana • Node Exporter • cAdvisor

Model & UI

Ollama • OpenWebUI

Infrastructure

Docker • Proxmox • ZFS • NVMe

Security

Nginx • Let's Encrypt • SSL/TLS

Database

PostgreSQL • Time-Series Storage

Production-grade deployment with enterprise observability, secure access, and scalable architecture for AI workloads. Comprehensive monitoring across host and containerized services enables data-driven optimization of model inference and agent orchestration on Proxmox infrastructure.

Key Takeaways

- Observability drives optimization: measure p95/p99 latency, tokens/s, and saturation
- Container + host visibility prevents blind spots in agent orchestration
- Predictable performance on Proxmox via data-driven capacity planning
- Secure, self-hosted control with enterprise-grade reliability

Questions about implementation?

Connect with me to discuss architecture and best practices

Simon Renauld

simondatalab.de