

# NYAKUNDI EMMANUEL

Systems & Infrastructure Engineer

nyariboemmanuel8@gmail.com | [GitHub: Emmanuel326](#) | [emmanuel326.github.io](#)

---

## PROFESSIONAL SUMMARY

Systems engineer specializing in **Go, Kubernetes, and Linux kernel programming**. Active contributor to **GitLab Runner** (Page 5 globally, 479 points in 2 weeks) and **Kubernetes Gateway API (CNCF)**. Deep expertise in container runtimes, concurrency primitives, and performance analysis. Proven ability to identify security vulnerabilities, optimize critical infrastructure code, and provide maintainer-level technical reviews. Strong focus on **cloud-native systems, networking data planes, and production reliability**.

---

## OPEN SOURCE CONTRIBUTIONS

### GitLab Runner Contributor | GitLab (Go, Kubernetes, Docker)

*January 2025 - Present*

- **Ranked Page 5 on global contributor leaderboard** within 2 weeks (479 points, 465 from merged code)
- **Merged 3 production MRs** fixing critical issues in retry infrastructure, context cancellation, and timer lifecycle
  - Fixed backoff retry to respect context cancellation, improving shutdown responsiveness across Runner codebase
  - Eliminated timer leaks in network layer by replacing `time.After` with properly managed timers
  - Corrected context error handling in runner wrapper, preventing incorrect error propagation
- **Provided maintainer-level technical reviews** on security-critical Kubernetes features
  - Identified 7 critical issues in Kubernetes Secrets implementation: feature flag bypass, RBAC permission gaps, secret lifecycle bugs, resource leaks, and missing test coverage
  - Proposed architectural improvements and concrete code fixes adopted by maintainers
- **Approved as Community Contributor** by GitLab engineering leadership after demonstrating deep K8s executor knowledge
- **Reduced maintainer review burden** by triaging complex Docker/Kubernetes MRs with detailed technical analysis

**Technical Impact:** Reviews prevent security vulnerabilities from reaching production; merged fixes improve reliability for millions of CI/CD pipelines.

Links: [MR #6061](#) | [MR #6063](#) | [MR #6064](#) | [Security Review](#)

### Kubernetes Gateway API Contributor | CNCF (Go, Kubernetes)

*February 2025*

- **Security Enhancement:** Added length validation (253 chars) to controller name validation,

- preventing ReDoS attacks
- **Test Coverage:** Achieved 100% coverage with 20 comprehensive test cases per API version (v1, v1beta1)
- **Standards Alignment:** Ensured validation adheres to Kubernetes DNS subdomain limits and best practices
- **Applied to both API versions:** Consistent validation across v1 and v1beta1 with benchmark tests

Link: [PR #4514](#)

---

## TECHNICAL SKILLS

**Languages:** Go (production-level), C, Rust, Python, Shell Scripting

**Cloud Native:** Kubernetes (RBAC, Secrets, Pod lifecycle, Operators), Docker, GitLab CI/CD, Container Runtimes

**Networking & Kernel:** eBPF/XDP, Linux Kernel Programming, Packet Processing, Network Data Planes

**Systems:** PID Namespaces, Signal Handling, Page Cache, Syscall Analysis, Memory Management, Process Lifecycle

**Concurrency:** Goroutines/Channels, Pthread Primitives, Lock-Free Algorithms, Backpressure Design, Context Propagation

**Tools:** Git, GDB, pprof, strace, perf, Observability Tooling (Prometheus patterns)

**Distributed Systems:** Graceful shutdown, Error handling patterns, Feature flags, Production debugging

---

## TECHNICAL PROJECTS

### Procjail | Minimal Process Isolation in Go

*Go, Linux Kernel, Namespaces*

- Built process jail implementing **PID namespace semantics** and kernel contracts without high-level runtime abstractions
- Managed **PID 1 responsibilities:** signal forwarding, process group management, orphan reaping, zombie prevention
- Engineered production-grade graceful shutdown with proper signal delivery and context cancellation
- **Applied learnings directly to GitLab Runner contributions:** Deep understanding of container internals enabled high-quality reviews and bug identification in production CI/CD infrastructure

Link: [github.com/Emmanuel326/procjail](https://github.com/Emmanuel326/procjail)

### Syslat | System Latency Analysis Suite

*C, Go, Linux Kernel*

- Designed microbenchmarking suite measuring **syscall latency and page cache behavior** for `pread()`
- Conducted experiment-driven analysis exposing why naive benchmarks provide misleading performance data
- Quantified kernel-level latency vs measurement noise in production Linux environments
- Applied findings to optimize I/O patterns in distributed systems

[Link: `github.com/Emmanuel326/syslat`](https://github.com/Emmanuel326/syslat)

## Mutex & Condition Variable Work Queue | Concurrency Primitives

*C, Pthreads*

- Implemented bounded work queue from scratch using **pthread mutexes and condition variables**
- Optimized for correctness under high contention: spurious wakeups, shutdown semantics, invariant maintenance
- Achieved thread-safe backpressure management validated through stress testing
- **Applied patterns to Go concurrency:** Context-aware cancellation and graceful shutdown in GitLab contributions

[Link: `github.com/Emmanuel326/mutex\_condvar`](https://github.com/Emmanuel326/mutex_condvar)

## False Sharing Analysis | CPU Cache Behavior

*C, Performance Analysis*

- Designed 2×2 microbenchmark matrix demonstrating **cache-line sharing impact** on execution time
- Quantified performance degradation (up to 10x slowdown) from cache coherence latency
- Controlled experiments isolating when false sharing is hidden vs. when it dominates performance
- Applied learnings to optimize multi-threaded Go code

[Link: `github.com/Emmanuel326/false-sharing`](https://github.com/Emmanuel326/false-sharing)

---

## TECHNICAL WRITING

Published Research & Analysis ([emmanuel326.github.io](https://emmanuel326.github.io))

- “**What I Learned by Removing All Container Abstractions**” - Deep dive into PID 1 semantics, signal delivery, and kernel contracts. Documents production-level container failure modes.
  - “**Measuring pread() Latency and Page Cache Effects**” - Methodology for syscall benchmarking and understanding kernel behavior vs measurement artifacts.
  - “**False Sharing, Made Obvious**” - Controlled experiments demonstrating when cache coherence latency impacts performance.
  - “**Bounded Work Queue using Mutex + Condition Variables**” - Implementation guide covering backpressure, spurious wakeups, and shutdown correctness.
- 

## EDUCATION

**Kenyatta University** | Bachelor of Science, Mathematics & Computer Science  
2023 - 2027 (*Expected*)

**Relevant Coursework:** Algorithmic Complexity, Systems Architecture, Operating Systems, Computational Logic

---

# WHAT I'M LOOKING FOR

Remote **Backend/Infrastructure Engineer** role (intern, junior, or mid-level) working on:

- **Kubernetes-native infrastructure** (operators, controllers, cluster tooling)
- **CI/CD & Developer Tooling** (build systems, containerization, pipeline optimization)
- **Observability platforms** (eBPF-based monitoring, distributed tracing, metrics systems)
- **High-performance systems** (data planes, packet processing, service mesh, networking)

Strong preference for **remote-first teams** building cloud-native infrastructure with mentorship opportunities. Open to companies in observability (Grafana, Datadog), infrastructure (HashiCorp, Cloudflare), CI/CD (GitLab, CircleCI, Buildkite), or CNCF projects.

---

**Available immediately for remote work. Timezone: EAT (UTC+3) - flexible for EU/US overlap.**