Omar Abdulaziz

+201007141064 | omarabdul3ziz@gmail.com | LinkedIn | GitHub

Experience

Software Engineer — Codescalers Egypt *Cairo, Egypt | 2021 — Present*

Zero-OS — lightweight operating system for distributed cloud nodes *Go, Linux, Networking, Virtualization, NFTables, RPC*

- Performed root-cause analysis and shipped fixes for production incidents across storage, scheduling, and networking layers, working with cross-team owners on rollout and rollback
- Migrated the node networking stack from macvlan to veth + netns across the fleet with a staged rollback strategy, keeping tenant workloads online during cutover
- Hardened multi-tenant isolation with NFTables rule sets scoping guest-VM egress, closing a class of cross-tenant exposure without breaking public-service reachability
- Eliminated recurring namespace-leak incidents with an automated cleanup worker that reclaims orphaned netns and veth pairs on long-running nodes
- Shipped a periodic node health-check service exposing CPU / memory / network / disk SLIs over RPC, used by ops for fleet-wide incident triage
- Built a Lens-style RPC inspector for tenant workloads that lists deployments and runs on-demand VM, network, and storage probes, used by ops for live triage
- Built a local image-validation harness that boots rootfs images under cloud-hypervisor with virtiofs + cloud-init wiring, used to test cloud images before fleet rollout
- Added an S3 storage backend to the internal content-addressable filesystem used to distribute Docker images and rootfs across the fleet

TFGrid SDK & GridProxy — public API over a 3,000+ edge-node grid *Go, PostgreSQL, Concurrency, REST, Microservices*

- Owned the GridProxy API across 60+ tagged releases — release engineering and a layered test suite spanning unit, integration, and end-to-end runs against a docker-compose grid
- Cut p95 API latency by ~3x through schema normalization, targeted indexes, and materialized-view caching for hot resource-aggregation endpoints
- Built concurrent indexers (Go generics, goroutine pools) that aggregate state from 3,000+ edge nodes into the query layer in near real time
- Extended the public API with filtering, sorting, and pagination over heterogeneous resources (nodes, farms, public IPs, GPUs), powering complex predicate queries from dashboards and developer tools
- Built a data-dump generator that seeds production-scale datasets for benchmarking and integration tests, surfacing regressions before deploy

Mycelium Cloud — managed Kubernetes service on a distributed compute platform *Go, Kubernetes, k3s, CRDs, PostgreSQL*

- Built the cluster deployer engine for multi-master k3s clusters, with persistent workflow state and exponential-backoff retries that resume deployments cleanly through partial failures
- Designed a rollback workflow that unwinds failed deployments step-by-step in reverse, recovering tenant clusters from partial-failure states without operator intervention
- Extended Kubernetes with custom resources and controllers that auto-reconcile gateway and ingress routing for tenant workloads, eliminating manual configuration
- Shipped a real-time notification system over SSE streaming workflow progress and deployment events to the dashboard, with pagination, validation, and read/unread state

Dashboard & Weblets — self-serve deployment UI for the grid *TypeScript, Vue.js, Node.js, Prometheus, Grafana*

- Migrated the user-facing dashboard from a legacy Svelte stack to a Vue 3 architecture, unifying two frontends and cutting maintenance surface
- Maintained 20+ production Docker images of community workloads (Discourse, PeerTube, etc.) so they install on the grid in one click
- Integrated Prometheus + Grafana for grid-wide metrics and dashboards used by operators for capacity and incident review

Grid-compose *Go, Cloud Deployment*

- Mentored the design of a Docker-Compose–style deployment service for the grid, letting users define full environments declaratively in one file

Software Engineer — Diatom *Cairo, Egypt | 2025 — Present*

Diatom Seller Platform *Go, PostgreSQL, Message Queues*

- Built the seller backend in Go + PostgreSQL — product catalog, real-time inventory, automated pricing — for the launch seller cohort
- Designed an async notification pipeline over a message queue delivering seller-facing alerts off the request path with at-least-once semantics

Skills

Languages: *Go (primary), Bash, SQL, TypeScript, Rust*

Systems & Networking: *Linux (namespaces, cgroups, netns); virtualization (KVM, cloud-hypervisor, virtio); networking (NFTables, veth, macvlan)*

Distributed Systems: *REST, RPC, message queues, async workflows, microservices, Go concurrency*

Cloud & Infrastructure: *Kubernetes (CRDs, controllers), Docker, Prometheus, Grafana, object storage (S3)*

Databases: *PostgreSQL (schema design, indexing, query tuning, materialized views), Redis*

Personal Projects

Gocker *Go, Linux namespaces, cgroups*

- Container runtime built from scratch in Go — implements PID / UTS / NET / MNT namespaces, cgroup-based CPU and memory limits, and overlay-based image layers

BusyNux *BusyBox, Linux, Bash*

- Minimal bootable Linux distribution: custom rootfs assembled around a BusyBox userland, exploring the smallest viable userland for containers

Diwan *Go, Claude API, Tavily, Telegram*

- RAG-based equities research assistant: screens stocks via Claude + Tavily web search and delivers sourced reports to Telegram

Education

B.Sc. Communications & Electronics Engineering — Nile Academy *2016 — 2021*