

Christopher Gorski

Data Engineer | AI Experts | AWS & GCP Specialists

<https://www.linkedin.com/in/christopher-gorski-7a762038a> | chris.dev1472@outlook.com

Work Experience

Senior Software Engineer – Allegro

April.2024 – Present

- *Standardized progressive delivery: org-wide Argo Rollouts with health-checks, analysis templates (PromQL & Web Vitals), and automatic rollback; deploy failure 3.1% → 0.6%, MTTR 27 → 9 min.*
- *Observability platform: rolled out OTel collectors (traces/logs/metrics) and golden dashboards with exemplars; MTTD -60%, cross-service debug 45 → 12 min.*
- *Autoscaling & capacity: introduced Karpenter + HPA/VPA; kept CPU at 55–65% target while cutting node hours -28%; monthly infra spend -18% (~\$5.7k/mo).*
- *Reliability patterns: PDBs, retry/backoff, idempotency keys, rate-limiters; eliminated two cascading failures and reduced incident frequency -30%.*
- *Security hardening: SBOM (Syft), image signing (Cosign), OPA/Gatekeeper policies, CVE gates in CI; critical vulns -92% over two quarters.*
- *Network & TLS: mutual TLS between services via mesh, short-lived cert rotation, strict mTLS; zero man-in-the-middle regressions.*
- *Disaster recovery: Velero backups with scheduled restores; quarterly game-days proved RTO < 20 min, RPO < 5 min for tier-1 services.*
- *Cost hygiene: right-sized JVM/Node/Python containers via profiling; CPU -22%, k8s requests -18% with no SLO regression.*
- *Frontend ops console: React dashboard for SLOs/deloys/quotas; on-call triage time -35%.*
- *Compliance: CIS benchmarks automated, encrypted secrets in Vault, audit trails via OTel + Loki; passed external audit with 0 critical findings.*

Senior Software Engineer – Artisan AI

Mar.2024 – Mar.2025

- *GPU platform at scale: Triton + KServe with auto-batching and per-model SLOs; throughput +41%, unit cost -24%; configured PodDisruptionBudgets to keep capacity during node drains.*
- *Traffic management: Envoy filters + rate-limit service (Redis) for per-tenant quotas and shadow traffic; sustained 99.9% availability during launch spikes.*
- *Eventing & exactly-once: Kafka with idempotent outbox/inbox; zero data-loss incidents; reprocessing cost -37% across pipelines.*
- *Secure supply chain: SBOM, signed images, SLSA level 2 provenance; blocked 3 misconfigured images pre-prod.*
- *Mesh & policy: Istio peer authentication, mTLS, and OPA data-plane policies; no lateral-movement findings in penetration testing.*
- *Cluster lifecycle: blue/green cluster upgrades with surge nodes and drain budgets; zero downtime for gateway pods.*
- *Observability maturity: OTel → Tempo/Loki with tail-based sampling, synthetic probes, rum ↔ backend trace stitching; MTTR -40% over 2 quarters.*
- *Runbooks & chaos: documented operator playbooks, monthly chaos drills (node kill, network delay); recovery objectives consistently met.*
- *Ops UI: Vue console for quotas/SLOs/toggles; operator effort -35% and onboarding time -50%.*

DevOps Engineer – AioCare

Oct.2021 – Feb.2024

- Rolled out a Kubernetes platform (3 envs, 8 node pools) with Terraform + Helm + Argo CD, cutting deploy time 45 min → 7 min (-84%) and enabling ~20 prod releases/week; verified in Argo CD history and Jira releases.
- Built progressive delivery (blue/green + canary 5%→25%→100%) on GitHub Actions → Argo Rollouts, reducing failed deploys by ~72% and MTTR 38 min → 11 min; confirmed via PagerDuty + Grafana incident timelines.
- Introduced SLOs (p95 API < 250 ms, error budget 1%/30d) with burn-rate alerts; lifted API uptime to 99.95% over 12 months and cut MTTD 12 min → 3 min; Prometheus/Grafana dashboards as evidence.
- Optimized cloud spend -28% YoY using HPA/VPA, spot nodes, storage lifecycle policies, and right-sizing; environment spin-up shrank 2 days → 2 hours with reusable Terraform modules; validated by Cost Explorer/Billing + CI timestamps.
- Shipped end-to-end observability (OpenTelemetry traces, Prometheus metrics, Loki/ELK logs) and golden dashboards per service, lowering “unknown root cause” postmortems by ~60% and after-hours pages by ~50%; PagerDuty analytics + RCA tags.
- Hardened SDLC for medical data: SBOM + image signing (Syft/Grype, Cosign), OPA/Gatekeeper policies, Vault/KMS secrets; reduced critical CVEs in images by >90% within 2 quarters; security scan trends used for proof.
- Delivered DR runbooks and quarterly game-days achieving RPO ≤ 5 min **and** RTO ≤ 30 min via cross-region backups and infra-as-code restores; results documented in DR drill reports.

Software Engineer – Spyrosoft

Jul.2016 – Sep.2020

- Designed, developed, and maintained enterprise and consumer-facing applications for multiple international clients, such as a fintech dashboard for BNP Paribas, an e-commerce platform for a retail client, and a healthcare management tool for a telemedicine startup; delivered scalable solutions using React, Angular, and Vue on the front end, with FastAPI, Django, Java, and Golang powering high-performance back ends, ensuring seamless integration, security, and excellent user experience.

Education

Master's Degree in Computer Science

Lublin University of Technology

10/2014 - 03/2016

Bachelor's Degree in Computer Science

Lublin University of Technology

05/2010 - 09/2014

Skills

- Kubernetes/EKS/GKE, Argo CD & Rollouts, Helm, Karpenter/Cluster Autoscaler, Istio/Envoy
- CI/CD & Release: GitHub Actions, GitLab CI, Canary/Blue-Green, Progressive delivery, SBOM (Syft/Grype), Cosign signing, SLSA provenance
- Reliability: SLOs & burn-rate policies, multi-AZ HA, chaos drills (Litmus), PDBs, retry/backoff, rate-limits, circuit breakers
- Observability: OpenTelemetry (traces/logs/metrics), Prometheus/Loki/Tempo, Grafana dashboards, Datadog, Sentry, synthetic probes
- Security: IAM least privilege, OPA/Gatekeeper, Kyverno, Vault, secrets rotation, mTLS, WAF, image allow-lists
- Cost & Capacity: Karpenter, Vertical/Horizontal Pod Autoscaler, right-sizing, Spot capacity, AWS Cost Explorer/Granite, Datadog CSM
- Data/State: PostgreSQL, Redis, Elastic, S3, Velero, EBS/EFS snapshots, Backups/DR

- *Frontend for Ops: React/Angular/Vue dashboards (SLOs, deploy health, quotas)*