

Christopher Gorski

Senior Software Engineer / AWS & GCP Specialists
chris.dev1472@outlook.com | +48732126312 | [Linkedin](#)

Work Experience

Senior Software Engineer – Allegro

April.2025 – Present

- Migrated checkout APIs to FastAPI + async SQLAlchemy with Kafka Outbox/Saga; p99 420 → 210 ms at 6.5k rps; tuned Unicorn workers + uvloop.
- Built cache-aside with Redis and request-scoped async caches; DB reads -38%, ≈\$14k/qtr IOPS saved.
- Progressive delivery via Argo Rollouts; failures 3.1% → 0.6%, MTTR 27 → 9 min.
- End-to-end OTel (asgi middleware), structlog correlation; debug 45 → 12 min across 12 services.
- Eliminated long GC pauses by isolating CPU-bound work in Cython + background workers; CPU -22%, k8s req -18%.
- Frontend: Angular 17 MFE + React promo editor; LCP 3.2 → 1.9 s, CLS 0.14 → 0.04.
- Security: OAuth2/OIDC flows (Auth Code + PKCE) with PyJWT rotation; zero auth incidents in 2 quarters.
- Hardening: pydantic validations, rate limits via Redis, idempotency keys; prevented double-charge edge cases.

Senior Software Engineer – Artisan AI

Mar.2024 – Mar.2025

- Built Python inference gateway (FastAPI + gRPC + httpx) in front of KServe; p95 ~180 ms@2.1k rps sustained.
- Implemented RAG with pgvector + Redis Bloom + re-ranking; F1 +13%, p99 520 → 260 ms.
- Reliable Celery pipelines with Kafka backpressure + exactly-once outbox; -37% reprocessing costs.
- Safety: prompt templates, sanitization, toxicity classifier; hallucinations -28%, policy -62%.
- Frontend: React admin + Vue playground with eval diffs; PR cycle 3.4 → 2.1 days.
- Packaging: manylinux wheels + slim base images; cold starts -35%.
- Implemented feature-flagged experiments (FastAPI dependencies) to rollback risky models in <5 min.

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

- Programming Languages: Python, Go, Java, TypeScript/JavaScript, SQL
- Frameworks & Libraries: **FastAPI**, Django REST, Celery/Redis, SQLAlchemy (async), Pydantic v2, Uvicorn/Gunicorn, gRPC, Airflow, Node.js (NestJS/Fusion.js), React, Angular, Vue
- Cloud & Infrastructure: AWS (Lambda, S3, EC2, EKS etc), GCP, Docker, Kubernetes
- Database Technologies: PostgreSQL, MongoDB, MySQL, Firebase, Redis
- Message Queues & Real-Time Systems: RabbitMQ, Kafka, WebSocket, Firebase, SNS
- Security & Compliance: OAuth 2.0, JWT, SSL/TLS, GDPR
- DevOps & CI/CD: GitLab CI/CD, Jenkins, Travis CI, Docker, Kubernetes, AWS Lambda
- Testing Frameworks: JUnit, Mockito, Jest, Cucumber, Postman, Selenium
- Version Control & Collaboration: Git, GitHub, GitLab, Bitbucket, Jira, Confluence
- Monitoring & Logging: Prometheus, Grafana, AWS CloudWatch, Datadog