Koh Guan Chin James

Platform & DevOps Engineer | Internal Developer Platforms · Kubernetes · DevSecOps · AI-Assisted Reliability

Singapore | linkedin.com/in/kohguanchinjames | github.com/gcjk768 | james.kohgc@gmail.com

PROFESSIONAL SUMMARY

Platform and DevOps engineer who owns the path to production for a 300-repository, 50+ pipeline enterprise estate in a heavily regulated environment. I build the golden paths other teams ship on — reusable GitLab CI templates, Terraform-managed AWS infrastructure, zero-downtime EKS/ECS releases, and security gates that enforce compliance in the pipeline instead of in review. Instrumented all four DORA metrics across the estate and used the data to drive a ~35% reduction in mean time to deploy. I also put AI inside the platform itself: a production agentic self-healing pipeline that diagnoses CI failures with an LLM and opens its own draft remediation merge requests, running air-gapped on CPU-only runners with no external API calls. AWS Solutions Architect – Associate; Claude Certified Architect – Professional (CCAR-P) in progress; MTech (NUS-ISS) in flight.

CORE SKILLS

Cloud & Kubernetes: AWS (EKS, ECS, EC2, IAM, VPC, S3, CloudWatch, SSM), Kubernetes, zero-downtime cluster upgrades, kubeadm, CoreDNS, NetworkPolicy, Helm, Oracle Cloud

CI/CD & Platform: GitLab CI/CD, reusable pipeline templates & golden paths, Docker, Terraform, Terragrunt, CloudFormation, Backstage, Nexus, Maven/Gradle, merge-request workflows

Reliability & Observability: DORA metrics (deployment frequency, lead time, MTTR, change failure rate), SLI/SLO-driven monitoring, Elasticsearch/ELK, Grafana, Prometheus, incident triage, root-cause analysis, performance tuning, runbooks

DevSecOps: SonarQube quality gates, Fortify SAST, OWASP dependency checks, Nexus IQ (SBOM & licence policy), Trivy image scanning, OWASP ZAP DAST, vulnerability management, least-privilege IAM

AI Engineering: LLM-integrated pipelines (failure analysis, draft-fix MRs, anomaly detection), agentic and multi-agent systems, RAG, Model Context Protocol (MCP), Ollama/Qwen self-hosting, Claude API, LLM red-teaming, evaluation & fairness gates in CI, SHAP explainability

Languages & Runtimes: Python, Node.js/JavaScript, React, Java, Bash, SQL, Go (learning); Spring Boot, Tomcat, Kafka, SQS

PROFESSIONAL EXPERIENCE

Platform & DevOps Engineer — NCS Group

Apr 2024 – Present

Enterprise platform engineering for regulated, security-cleared public-sector accounts

- Built and maintained the reusable GitLab CI template library behind 50+ production pipelines across SIT, UAT, Non-Prod and Prod — delivery teams consume a golden path instead of hand-rolling YAML, eliminating pipeline duplication and cutting authoring time from days to hours.
- Led DORA instrumentation across a 300-repository estate, wiring up deployment frequency, lead time, change failure rate and MTTR, then using the data to drive a ~35% reduction in mean time to deploy.
- Executed zero-downtime EKS cluster upgrades and ECS rolling deployments through pipeline-driven orchestration, managing CloudFormation changesets and SSM-based deployment paths across environments.
- Embedded security gates at every pipeline stage — SonarQube quality gates, Fortify SAST, OWASP dependency checks, Nexus IQ SBOM and licence policy, Trivy image scanning — so compliance is enforced by the platform rather than by manual review.
- Migrated 9 applications from on-premises to cloud-hosted GitLab with standardised governance and change control; shipped reference pipelines for Java, Node.js and React covering static analysis, dependency scanning, WAR/JAR/image packaging and Nexus artifact management.
- Triage and root-cause production build and deployment incidents across the estate alongside application, network, security and DBA teams, closing each one out with a runbook or a pipeline control so the same failure cannot recur.
- Modernised legacy Ant builds to Maven/Gradle, containerised test suites and tuned runner caching for reproducible, fast developer feedback.

SELECTED ENGINEERING PROJECTS

Agentic Self-Healing CI Pipeline *(production, NCS)*

Two-phase agentic system running inside GitLab CI. Phase 1 performs LLM failure analysis behind an 80% confidence gate; Phase 2 opens Draft merge requests carrying YAML fixes on isolated branches, and never auto-merges — a human always approves. Runs ai/qwen3-coder on CPU-only runners, fully air-gapped with zero external API dependencies.
Stack: GitLab CI, Docker, Qwen3-Coder (Ollama), Node.js, Python

Pipeline Health & DORA Dashboard *(production, NCS)*
Node.js and React 18 dashboard surfacing live DORA metrics across 50+ pipelines for the teams that own them, with AI-assisted failure analysis and a Library & Threat Radar that scores dependency risk. The insights it exposed drove the ~35% mean-time-to-deploy reduction.
Stack: Node.js, React 18, GitLab API, Elasticsearch

Backstage IaC Self-Service Portal *(proposal, NCS)*
Designed a self-service infrastructure provisioning platform: Backstage front door, Terraform/Terragrunt modules, GitLab CI execution, a rule-based selection engine, and a Claude template-selection layer scoped strictly to the approved module library. 24-slide proposal presented to NCS leadership.
Stack: Backstage, Terraform, Terragrunt, GitLab CI, Claude API

CareRoute AI — Multi-Agent Platform with an MLSecOps Assurance Pipeline *(NUS-ISS practice module — Architecting AI Systems)*
Supervisor multi-agent backend and React console over a real severity model. Shipped the full assurance layer as CI stages: LLM red-team suite, SHAP explainability, a Fairlearn fairness gate that fails the build, drift audit, model scanning, dependency scanning and OWASP ZAP DAST, plus an LLM provider failover chain.
Stack: Python, FastAPI, React, Claude/OpenAI, Fairlearn, SHAP, GitLab CI

Ollama Gateway *(personal, self-hosted)*
Central inference gateway for a fleet of local AI services: request queueing, health and reliability instrumentation, and a shared knowledge-vault backend behind a single endpoint. 31 automated tests green.
Stack: Python, Ollama, REST, pytest

Algorithmic Trading AI System *(personal)*
End-to-end paper-trading engine across crypto and US/SGX equities. Local Qwen (Ollama) as the reasoning engine, an Obsidian vault as persistent memory over REST, and Telegram as a two-way control interface. Backtested MA-crossover with a 200-day trend filter; exposes an MCP server.
Stack: Python, Coinbase API, IBKR, Ollama/Qwen, Telegram, MCP

CERTIFICATIONS

- AWS Certified Solutions Architect – Associate — Amazon Web Services (Mar 2026 – Mar 2029)
- Claude Certified Architect – Professional (CCAR-P) — Anthropic; in progress

EDUCATION

Master of Technology in Software Engineering (in progress, expected 2028) — NUS-ISS, National University of Singapore
Graduate Certificates awarded: Designing Modern Software Systems (2025); Architecting Scalable Systems (2026). In progress: Architecting AI Systems (2026). Selected coursework: DevSecOps Engineering & Automation, Platform Engineering, Cloud-Native Solution Design, Architecting Agentic AI Solutions, AI & Cybersecurity, Deploying & Operating AI Solutions.

B.Comp.Sci (Game & Mobile Development) — University of Wollongong (SIM), Singapore
High Distinction: Database Systems. Distinctions: Algorithms & Data Structures, Mathematics for Computer Science, Python Programming. Capstone — Fresh Beer Near Me: led a 4-member team to ship a React Native + Node.js/MongoDB app with 500+ pilot downloads.

Anthropic Academy: AI Fluency Framework & Foundations · Introduction to Model Context Protocol · MCP Advanced Topics · Introduction to Agent Skills · Building with the Claude API.