

Joel Freeman

Platform & Infrastructure Lead, defence tech - first infrastructure hire, built the platform, then hired and now lead the team that owns it

Wellington, NZ | Relocating to Auckland, Sydney or Melbourne | Full Australian work rights | contact@jxel.dev | jxel.dev
linkedin.com/in/joelfreemannz | github.com/jxeldotdev

Profile

Platform engineer who takes an estate nobody owns, writes the architecture, builds the core as an IC and migrates production onto it. Did that in 14 months at Starboard, from landing zone to API gateway to a passed defence customer audit, then hired and now lead the four-person team that owns it. Seven years across GCP, AWS and Azure.

Experience

Senior Engineer - Platform & Infrastructure Lead, Starboard – Wellington, NZ July 2025 – present

First dedicated infrastructure hire for a maritime intelligence product used by defence, government and law-enforcement agencies in 30+ countries (~25 engineers). Inherited a fragmented, hand-built GCP and Kubernetes estate that had accumulated over five years without dedicated infrastructure ownership, with no safe deployment path. Wrote the platform roadmap and architecture for each layer, built most of the core myself, migrated production onto it, then hired the team that now owns the rest.

- **Took production deploys from hours or days, with one to three senior engineers and rollbacks that often failed, to five-minute self-service deploys with one-click rollback for any engineer**, by architecting and building a Kubernetes platform on GKE: five clusters, ~600 vCPUs. HA Argo CD reconciles every cluster from Git, so what is running is always what was reviewed. Kargo orchestrates signed promotions through dev, staging and production. Kyverno blocks anything reaching production that is not a signed promotion. 20 OPA Rego policies gate workload placement and every GitOps change before merge. CI posts a visual diff of what will change on each cluster to the pull request. A three-chart Helm family means a 10-line service definition drives deployment, secrets, network policy, gateway routes and alerts without a platform ticket, with External Secrets Operator syncing every secret from GCP Secret Manager so nothing is applied by hand or stored in Git. 15+ engineers ship through it; a teammate onboarded 12 services unaided.
- **Grew the Platform & Infrastructure function from one person to a team of four**: wrote the job descriptions and interview process, mentored three interns, two of whom became associate engineers and now own per-PR preview environments, synthetic monitoring and the compute restructure. Built standalone UAT and staging environments; their Envoy Gateway ingress became the pattern every environment now uses.
- **Migrated all 38 production services onto the new platform with zero downtime, no code freeze and nine days early** by turning 22 stalled pull requests into a staged rollout. Merged dormant configuration, enabled services one at a time and snapshotted each stage for rollback. Wrote a render-versus-live diff tool and a two-phase secrets CLI so engineers migrated in parallel, and a CI pipeline that renders every GitOps change once and runs nine validation stages (Kubernetes schema, OPA policy, diff against Argo CD live state, PromQL on shipped alert rules) into one verdict on the pull request. Retired 50 bespoke CI workflows. Cut over the ingestion service, which processes ~1 billion AIS messages a day and had been undeployable for four years, in ten minutes; it can now be rebuilt from Git in an hour.
- **Moved infrastructure changes from two engineers running Terraform locally with administrator credentials and heavy drift, to every engineer in the company shipping about 36 reviewed changes a week, applied minutes after merge, by building keyless Terraform delivery on GitHub Actions**. Each stack applies through Workload Identity Federation to a least-privilege identity scoped to its project. No long-lived credentials remain in the delivery path. Built the approval tooling: detects changed stacks on each PR, posts the plan, and gates production behind a named approver via a Slack ping and a GitHub issue that forms the audit trail; if the approved commit is no longer current, the tooling blocks the apply.
- **Removed all standing production access and passed a contractual NIST 800-171 and NZISM customer audit by designing just-in-time access on GCP Privileged Access Manager**. Implemented 87 tiered entitlements, a CLI for requesting and approving access in seconds, and automation that keeps on-call permissions aligned with the rotation. Fixed the audit findings and wrote up the evidence the auditors accepted.
- **Replaced a flat GCP organisation, where team cost and production access required manual investigation, with a Terraform landing zone in one month, importing 15 unmanaged projects with zero destructive changes**. It holds 37 projects across 26 team and environment folders, four hub-and-spoke Shared VPCs with a deny-all default firewall, 35 organisation policies and 22 custom least-privilege roles. It runs on Google Fabric FAST and follows the Google Cloud Well-Architected Framework. Every new project gets the same guardrails: no public VM IPs, no service-account keys, org-level audit logs to a locked-down project, and cost attribution derived directly from the folder hierarchy rather than labels. A new project is a YAML file reviewed by pull request; six engineers and the data team now provision projects independently.
- **Designed and shipped Starboard's first production API gateway, cutting the time to expose a new service publicly from a multi-day load-balancer change to under an hour and giving defence and government customers one public entry point**. Replaced three public entry points, each with its own load balancer, with Envoy Gateway on the Kubernetes Gateway API. Proved Cloud Armor Enterprise could protect the network load balancer directly and removed an entire load-balancer tier from the architecture, so WAF, DDoS protection and rate limiting apply once to every route. Moved live traffic

over with no customer impact. Teams can now enable canary releases and JWT/OIDC authentication through the standard delivery path. Teammates shipped their first routes within 48 hours.

- **Shipped an air-gapped build of a cloud-only product, passing Google Marketplace validation, for a NZ Defence bid.** Replaced managed cloud services with in-cluster PostgreSQL, MinIO and Valkey packaged behind a single Helm chart deployable unchanged to Google Distributed Cloud, GKE or a laptop.
- **Made monitoring and paging service-owned across all five clusters, and found four monitors reporting healthy while broken.** Built it on Grafana Cloud, Alloy and Alertmanager in Terraform, added eBPF tracing with no code changes, put SEV1-4 alert rules beside each service's deployment config and paged through Rootly. Wrote a CI check that renders every alert rule against live Grafana Cloud metrics and blocks the pull request when an alert is broken, does not meet the standard, or could never fire. Set the observability standard teams now work to: Bronze, Silver and Gold tiers, SLO burn-rate alerts over raw thresholds, and a definition of done for readiness reviews. Designed synthetic user-journey monitoring; an engineer I mentored rolled it out company-wide.
- **Designed and evaluated a workflow that lets coding agents plan and prove infrastructure changes while a human does every merge, apply and promotion.** Benchmarked on nine real merged infrastructure pull requests: unaided agents produced a usable validation plan for one. The workflow produced one for all nine. Most agent-written pass criteria could still appear to pass even when the underlying change was deliberately broken, so I added an adversary agent that tests each plan against constructed failure scenarios. This shipped as seven skills and four evaluation suites in the team's shared Claude Code toolkit, now used in my Platform workflow.
- **Cut NZ\$420k in annualised cloud spend.** Separately, traced a sharp bill increase to NZ\$45k of expiring credits rather than usage. Delivered every reduction through Terraform, validated each against actual billing, and rejected two proposed savings that would have removed production database storage or under-provisioned a live cache.

DevOps Platform Engineer, AWS Platform Team, MetService – Wellington, NZ

Sept 2024 – July 2025

- **Gave every environment one deploy path and one image** by building the multi-account ECS delivery pipeline on Terraform, GitHub Actions and CodeDeploy, with end-to-end tests and vulnerability scans posting back to the PR, and containerising the MuleSoft-hosted Java APIs onto it.
- **Replaced legacy DataWeave APIs with the Aviation Resiliency Project's IWXXM-to-GeoJSON pipeline,** designed and built in typed TypeScript on Lambda, S3, SQS and SNS, with missing-data alerting so it could not look healthy while files never arrived.
- **Cut Azure spend by more than NZ\$5,000 a month** during the migration back to AWS, and automated cost-centre exports to SharePoint so Finance could see spend without a hand-built report.

DevOps Platform Engineer, Azure Data Platform, MetService – Wellington, NZ

Sept 2023 – Sept 2024

- **Proved the replacement weather-data platform matched the legacy system before cutover** by building the comparison tooling, as a core engineer on the real-time medallion architecture on Databricks, Spark and Delta Lake.
- **Built the delivery and module layer the data platform shipped through:** GitHub Actions for Terraform, Azure Functions and Container Apps, a shared library of tested Terraform modules for Databricks, Functions and Synapse, and the REST APIs that exposed Delta Lake data to applications.

DevOps Engineer, Hyperscale Cloud Practice, AC3 – Wellington, NZ

Feb 2022 – Sept 2023

- **Led customer-facing modernisation across 12+ AWS accounts and hundreds of instances** with CDK, Terraform and CloudFormation; wrote custom AWS Config compliance tooling and ran upgrades for customer Kubernetes clusters.
- **Cut a customer's WordPress hosting cost 30% and improved performance 25%** by moving it to an ARM autoscaling group; hardened customers' VM images to CIS benchmarks with Azure Image Builder.

Technical Support Engineer (contract), Red Hat – Wellington, NZ

Dec 2021 – Mar 2022

Infrastructure Automation Engineer, Synerty – Wellington, NZ (Remote)

Nov 2020 – Sept 2021

- **Automated a critical GE ADMS (grid management system) upgrade on a grid serving 200,000+ homes and businesses** with vSphere and Attune deployment automation; shipped a Python LDAP(S) proxy that gave legacy SQL Server Reporting Services Active Directory auth without replacing it.

Cloud Administrator, Seekom – Wellington, NZ

Jan 2020 – Nov 2020

- **Built the multi-stage CircleCI pipelines for a distributed web app's move from fixed EC2 to Kubernetes** while running the existing EC2, RDS and IAM estate.

Service Management Analyst, Tribe – Hastings, NZ

Aug 2019 – Jan 2020

- L1/L2 Windows and Microsoft 365 support; wrote the PowerShell that automated the team's routine Active Directory administration.

Skills

Cloud & Architecture: GCP, GKE, Kubernetes, AWS, Azure, Terraform, landing zones, IAM, networking, Google Distributed Cloud

Delivery & Policy: GitOps, Argo CD, Kargo, Helm, GitHub Actions, OPA, Kyverno, Envoy Gateway, Cloud Armor

Observability & Reliability: Grafana, Prometheus, OpenTelemetry, eBPF, SLOs, incident response, FinOps

Languages & Agentic Engineering: Python, TypeScript; Claude Code, Codex, MCP, agent eval suites and permission boundaries

Projects

Atlas - Co-founder & CTO, 313 Management

2026, evenings and weekends

- **Co-founded 313 Management and built Atlas, an operations and audience-intelligence product plus attendee app, from first commit to both app stores for Rolling Meadows, a ~10,000-attendee music festival and our first paying customer.** The operator sells ~33,000 tickets a year across Rolling Meadows, Mayhem and a gig calendar. Own product strategy, UX, architecture, implementation, deployment and production operations, working directly with the festival operators to turn their problems into shipped features.
- **Built the whole stack so a two-person company can run it:** the operations platform where the event team manages artists, schedules, content, maps, notifications and publishing; the backend, APIs and PostgreSQL data layer; and an offline-first iOS and Android app that keeps schedules, artists, favourites and maps working when thousands of people are on site and connectivity fails. Content ships as immutable versioned bundles on a CDN that event staff preview, publish and roll back themselves. Serverless and low-operations on Next.js, React Native, Expo, Prisma and TypeScript on Vercel.
- **Turned every meaningful interaction into a signal and joined it with Flicket ticketing and attendance history into one profile per attendee:** artists viewed, favourites, schedule use, content engagement, notification response and how that changes over time. Where ticketing only knew that someone bought a ticket and the app only knew an anonymous user liked certain artists, Atlas connects them, so organisers target repeat, high-value or lapsed attendees instead of blasting everyone, and plan bookings and programming from what people actually did.
- **Built it as one person with an agent-native workflow:** decomposed the product into a canonical architecture, milestone build plans with exit criteria and a decision ledger, then ran concurrent coding agents in isolated Git worktrees for mobile, backend and admin, on scoped credentials and non-production defaults, with human approval for deploys, secrets and destructive database changes.

Education

Eastern Institute of Technology, Certificate in IT Essentials – Napier, NZ

Feb 2019 – Aug 2019