



JORDAN SIMONOVSKI

Staff Software Engineer

Blue Mountains, Australia

CONTACT

+61 451 309 913

jordan.simonovski@gmail.com

[jordansimonovski](https://github.com/jordansimonovski)

[linkedin.com/in/jsimonovski/](https://www.linkedin.com/in/jsimonovski/)

SKILLS & OTHER

Kubernetes platform engineering (EKS, Helm, GitOps, ArgoCD, custom operators)

AWS (10+ years, full platform scale up, compute, serverless)

Docker platforms (Kubernetes, EKS, GKE)

OpenTelemetry ecosystem (Weaver, Collector, SDK)

Observability (Prometheus/Mimir, Grafana, Tempo/ClickHouse)

Site Reliability Engineering (SLOs, War games, chaos testing/tooling)

EXTRACURRICULARS

Technical reviewer for [Observability Engineering, 2nd Edition](#).

DevOpsDays Wollongong organiser, 2025.

SRECon22 Asia/Pacific organiser.

PROFILE

Staff engineer owning global-scale observability infrastructure across Mimir, Tempo, and ClickHouse, operating largely independently across the full stack. A vocal advocate for moving beyond the three-pillar observability model toward wide events, with hands-on experience designing and driving that migration at scale. Builds Grafana Scenes applications that measurably reduce MTTR and unify visibility across heterogeneous technology stacks and data sources. Technical reviewer for Observability Engineering, 2nd Edition and an active contributor to the broader SRE and observability community.

EXPERIENCE

STAFF SOFTWARE ENGINEER - ATLISSIAN - OBSERVABILITY TEAM

May 2020 - Present

Led observability platform strategy and delivery for traces, metrics, and logs at Atlassian scale.

- Scaled telemetry platforms to 100TB/day+ on ClickHouse and 180TB/day on Tempo with strong reliability guardrails.
- Improved Tempo query performance by 30%+, reducing time-to-diagnosis and supporting faster incident response.
- Built and shipped operator UX for observability workflows that improved MTTR and made diagnostics easier for product teams.
- Evangelised the shift to wide events. Implemented ClickHouse, scaling to 100TB/day ingestion while focusing on sub-second query times and AI Observability.
- Built and singlehandedly ran observability tracing infrastructure with Grafana Tempo, scaling to 180TB/day ingestion.
- Introduced and implemented Grafana Scenes internally, implementing a centralised app for debugging core Atlassian experiences using various data sources, promoting good UX, explorability, leading to MTTR reductions for both novel and well understood incidents.
- Scaled Grafana Mimir to match existing metric usage with other products internally; Ran complex usage-accurate load tests to validate 200k alert evaluations.
- Standardised Kubernetes deployments by implementing helm charts built with my own OSS tools to better vendor, test, and control common requirements and constraints.
- Implemented a Grafana Scenes application to replace our existing explore workflows integrating with our internal AI gateway, using an AI assistant to generate accurate queries for: PromQL, TraceQL, SQL (ClickHouse), SPL (Splunk), explain data panels, and generate on-the-fly dashboards for debugging core Atlassian Experiences.
- Designed and implemented Atlassian's first accurate service map by covered experience, critical for reducing MTTR during high stress incident scenarios.
- Improved Tempo query performance 30%+ with minimal impacts on cost.
- Built out a local development observability solution for Atlassian engineers in my free time, with a strong focus on DX and flexibility.
- Designed and built multi-region (regional failover) infrastructure for Grafana and Grafana Mimir with a strong focus on low RTO/RPO, while keeping costs under control.

CLOUD ENGINEERING LEAD - LENDI

Jan 2018 - May 2020

Owned architecture and delivery for cloud-native migration, SRE adoption, and internal platform capabilities.

- Scaled platform maturity through SRE adoption, cloud migration, and internal DX-focused tooling.
- Implemented AWS multi-account and container platform patterns for safer delivery.
- Drove direction, architecture, implementation, and strategy for cloud-native migration.
- Implemented SRE principles such as SLOs and error budgets across the organisation.
- Built developer-centric tooling and internal PaaS capabilities to reduce deployment friction.
- Implemented Docker infrastructure on ECS and EKS.
- Implemented Envoy sidecar patterns for Auth0-based authn/authz flows.

JORDAN SIMONOVSKI

EXPERIENCE CONTD.

- Led from-scratch AWS setup, including multi-account foundations.
- Introduced production resilience practices including controlled chaos testing.

SENIOR BACKEND ENGINEER - HEALTHENGINE Jan 2017 - Jan 2018

Implemented resilient serverless messaging and observability foundations for backend systems.

- Delivered event-driven serverless workflows with stronger reliability characteristics.
- Improved service visibility through Prometheus/Grafana metrics and Elastic/CloudWatch logging.
- Built resilient serverless messaging workflows using SQS, DynamoDB, and scaling Lambda functions.
- Implemented Docker infrastructure in Rancher.
- Implemented monitoring infrastructure with Prometheus and Grafana.
- Implemented logging infrastructure with Elasticsearch and CloudWatch.
- Worked on Node.js backend applications.

DEVOPS ENGINEER - DOMAIN Jan 2017 - Dec 2017

Improved AWS/Docker platform reliability and delivery workflows with scheduler and CI/CD improvements.

- Implemented Nomad and Jenkins deployment workflows to improve release consistency.
- Enabled ChatOps and scaled containerized frontend rendering workloads.
- Maintained and improved ECS infrastructure.
- Implemented HashiCorp Nomad as a new Docker scheduler.
- Maintained and improved Elasticsearch cluster infrastructure.
- Implemented deployment tooling in Jenkins with Groovy.
- Set up a serverless deployment bot to enable ChatOps.
- Worked with frontend platform teams to scale React rendering workloads.
- Worked closely with developers on deployment tooling.

SIDE QUESTS

Incident Triage Copilot for Observability Workflows

- Built an assistant-driven query workflow that generated PromQL, TraceQL, and SQL from incident prompts.
- Reduced time-to-first-useful-query during incident debugging by improving discovery and query authoring ergonomics.
- I've kicked off an open source build of this in my free time.

Helm Coverage Tooling

- [HelmCov](#) is a dynamic coverage checker which aims to surface coverage issues in helm charts. This is useful for charts which have a lot of branched templating logic. Testing all paths is ideal.

Helm Semantic Versioning

- [Helmver](#) is a tool I wrote because I was really missing some of the nice semantic versioning tooling from the JS ecosystem. It borrows heavily from [changesets](#).

Heatmap Panel

- I should really rename this. Heatmap panel started off as a "could I do something like Bubble up but for ClickHouse?" and is turning into a lot more as I continue to test out various bits and pieces. [See repo here](#).

Local Developer Observability Stack

- Built a local-first observability stack for faster debugging in day-to-day engineering loops.
- Improved developer feedback loops by making traces, logs, and metrics available without production dependencies.

Extracurriculars

- Technical reviewer for [Observability Engineering, 2nd Edition](#).
- DevOpsDays Wollongong organiser, 2025.
- SRECon22 Asia/Pacific organiser.
- [Monitoring.Sydney](#) organiser.
- DevOpsDays Sydney 2019 organiser.

JORDAN SIMONOVSKI

EXPERIENCE CONTD.

- Currently building out an [open source Grafana Scenes app](#) on a ClickHouse data source to vastly improve developer experience when debugging applications