Abhiman Kolte

Staff Software Engineer | Reliability Infrastructure and Distributed Systems
SF Bay Area, CA | (424) 234-8080 | abhimanbhau@gmail.com | linkedin.com/in/abhimank

SUMMARY

Staff Software Engineer in Production Engineering with 6+ years building reliability and developer infrastructure at LinkedIn and Amazon. Created LinkedIn's change safety platform from scratch, a Go and gRPC system now gating nine change agent systems including the central deployment pipeline, which cut change failure rate by over 60%. Also own SLI/SLO practice and capacity planning for the org.

SKILLS

Distributed Systems and Backend: Go, gRPC/Protobuf, Temporal workflow engine, microservices, API design, concurrency, distributed locking and coordination, idempotent APIs, deterministic replay, multi-region and multi-colo design, high throughput low latency services
Reliability and Production Engineering: Deployment and change safety, SLI/SLO management, error budgets, capacity planning, incident detection and response, on-call rotation, postmortems, alerting, runbooks, failure domains, blast radius containment, progressive delivery, fail-open design
Infrastructure and Observability: Linux, Kubernetes, Docker, Terraform, deployment pipelines, infrastructure as code, performance benchmarking, Prometheus, Grafana, OpenTelemetry, distributed tracing, centralized alerting, KQL/Kusto
AI for Reliability: Agentic triage, evidence grounded RAG, LLM tool integration, GenAI guardrails
Languages: Go, Python, Java, JavaScript, Bash

EXPERIENCE

LinkedIn *Sunnyvale, CA | June 2022 - Present*

Staff Software Engineer, Production Engineering *(promoted from Senior Software Engineer)*

***Seatbelt**: change safety platform for production deployments (Go, gRPC, Temporal)*

- Created LinkedIn's change safety platform from scratch. It blocks production changes during system instability to prevent cascading failures, and cut change failure rate by over 60% org wide.
- Engineered the policy engine as a gRPC service sustaining thousands of requests per second under a strict sub-300ms budget, inline in the critical path of every gated change.
- Designed a pluggable validator architecture so distributed teams author and own custom safety checks, moving safety policy to the engineers closest to each risk.
- Integrated the platform across nine change agent systems, including the central deployment pipeline, then led the migration off the legacy Deployment Policy Engine to unify safety standards org wide.
- Built the rollout orchestrator on Temporal for durable, replay safe execution across thousands of concurrent workflows, advancing one colo at a time with every transition re-checking the policy engine first.
- Fed centralized alerting and OpenTelemetry signals into that decision, so a critical alert on the deploying service pauses its rollout mid-flight even when the broader environment is healthy, catching regressions the deployment itself introduced.
- Eliminated race conditions during concurrent lock acquisition across independent deployment pipelines with a two-phase coordination protocol: a sync pre-check at workflow start plus async conflict detection at acquisition, inside a 5 second budget.
- Shipped a fail-open architecture with independent kill switches and a shadow to enforce ramp, so platform outages degrade gracefully rather than blocking deployments.

***Functional test automation platform**: Playwright web E2E and API probing*

- Built the platform behind automated functional verification at LinkedIn, covering Playwright web E2E flows and an API prober. Onboarded 900+ services, over half the microservice fleet, to automated API testing and 50+ business critical services to frontend testing.
- Made the suites callable at every stage of the change lifecycle: ad hoc before a pull request, as a pre-deploy gate, on a schedule for continuous monitoring, and inside canary, where a service's own tests run against the single host carrying the new version and trigger automatic rollback on failure, catching broken builds instead of waiting for metrics to degrade.

***Incident triage automation**: agentic evidence collection and root cause analysis*

- Delivered a production agentic triage service that cut on-call investigation from 15 to 20 minutes down to under 30 seconds per failure.
- Built a multi-source evidence fusion layer correlating synthetic monitoring, L1 proxy logs (KQL), trace spans, and raw job logs into a grounded root cause payload, keeping the endpoint deterministic and AI disabled for an orchestrating agent to call, and recording the retrieval strategy behind each decision so every conclusion stays auditable.

SRE practice and tooling

- Own SLI/SLO management, capacity planning, and service availability, running infrastructure at scale on Kubernetes and Terraform. Wrote Confetti, a benchmarking library that profiles the performance impact of configuration changes inside CI/CD pipelines.

Amazon Lab126 *Sunnyvale, CA | Nov 2020 - June 2022*

Software Development Engineer

- Owned architecture and delivery of a monolith to microservices migration for a core device platform, decomposing coupled subsystems into independently deployable services with defined API contracts, distributed tracing, and automated observability, holding 99.99% availability post-launch.
- Cut new pipeline onboarding from 8 weeks to 4 through self-service infrastructure automation and standardized deployment patterns.

Amazon Web Services *Dallas, TX | Jan 2020 - Nov 2020*

Software Development Engineer

- Shipped Python SDK abstractions for internal AWS developer tooling used by thousands of engineers, and built client-side permission validation for the object storage SDK that eliminated a class of runtime authorization failures.

EDUCATION

M.S. Computer Science, Santa Clara University, 2019
B.S. Computer Science, Pune University, 2016