Edward Vartanessian

Platform Engineer — Infrastructure · Observability · Automation

Greater Boston | U.S. citizen | e_vartanessian@live.com | linkedin.com/in/evartanessian | github.com/eddiev2 | evartanessian.dev

SUMMARY

Software engineer building cloud platforms on AWS with Terraform and Python. At Cisco, own the data layer for a **greenfield internal procurement + ITSM platform** built to replace two enterprise SaaS products — Aurora PostgreSQL, Lambda, and private VPC networking defined in Terraform across three environments, delivered through GitHub Actions. Previously engineered reliability and observability for a global 10,000+ device fleet at Splunk: shipped a **custom OpenTelemetry Collector distribution**, authored the upstream proposal that added macOS host metrics to the Collector (contrib #33393), and raised macOS patch compliance to 97–99% fleet-wide. Building Copywarden (copywarden.com), an encrypted macOS clipboard manager in Swift 6 — applied cryptography, signed and notarized.

SKILLS

Languages & Frameworks: Python, SQL, Swift (SwiftUI), TypeScript (React, Next.js), JavaScript, Bash, PowerShell, HCL (Terraform)

Cloud & Infrastructure: AWS (Lambda, Aurora PostgreSQL, RDS Proxy, S3, CloudFront, Route 53, KMS, Secrets Manager, IAM, VPC, EC2), Terraform, Docker, Kubernetes, Puppet

CI/CD: GitLab CI, GitHub Actions, Docker-based pipelines, GitOps, code signing & notarization

Observability & SRE: OpenTelemetry, Splunk, Prometheus, Grafana, SLOs/SLIs, incident management

Platforms & Security: ServiceNow development (scripting, workflows, REST integrations), Okta/IAM-as-code, CrowdStrike, Jamf, Installomator, Tenable, SonarQube, SOC 2 / ISO 27001

EXPERIENCE

Cisco — Software Engineer, Platform

Sept 2025 – Present · joined via Cisco's acquisition of Splunk

- **Own the data-layer workstream** for a greenfield procurement and ITSM platform being built to replace two enterprise SaaS products — Aurora PostgreSQL, schema design, database connectivity, and migration safety across dev, staging, and production — and **one of two required approvers for infrastructure and security changes platform-wide**
- **Built reusable Terraform infrastructure for Aurora Serverless v2** — environment-specific scaling, private subnet placement, KMS encryption, enforced TLS, Multi-AZ production configuration, and no public database access
- **Designed the platform's passwordless database access model** — no stored password anywhere in the path: an in-VPC Lambda migrator authenticating through IAM, with schema-migration and application roles separated so the application cannot alter schema. Version-controlled, peer-reviewed Drizzle migrations, validated end to end against dev Aurora including a tested forward-only rollback
- **Designed and built the self-service machine-ordering site** — the platform's primary user-facing workflow — end to end, across the React frontend and the API and data paths behind it
- **Automated ServiceNow asset reassignment for new-hire fulfillment**, replacing a manual Asset Management task with controlled decision logic, identity validation, record updates, and an auditable work-note trail
- **Own incident management for device automation** across the fleet: automated triage, runbooks, and post-incident fixes that prevent repeats

Splunk — Client Platform Engineer (DevOps/SRE)

Apr 2022 – Sept 2025

- Engineered automation and observability for a **global 10,000+ device fleet** (macOS, ~1,200 Windows, ~1,000 Linux) with zero-downtime updates, sustaining SOC 2, ISO 27001, and UK Cyber Essentials compliance — defining health/compliance SLIs on OpenTelemetry + Splunk (patch currency, security-agent liveness, configuration drift, enforcement latency) with SLO-backed alerting
- **Built and shipped a custom OpenTelemetry Collector distribution** — authored the OCB manifest to compile a minimal purpose-built binary instead of deploying vendor agents, then packaged, Developer ID-signed, and Apple-notarized it for endpoint distribution at Splunk and Cisco; identified its missing macOS CPU/disk host metrics (official builds ship CGO_ENABLED=0) and drove the fix upstream — authored proposal opentelemetry-collector-contrib #33393, since implemented

- **Brought GitOps to fleet configuration**: put ~300 Jamf scripts and extension attributes under version control with peer-reviewed merge requests and a GitLab CI pipeline that deployed to production on merge, replacing hand-edits in a web console and **establishing peer review as the team's standard for fleet configuration changes**; wrote the API extraction tooling that seeded the repo and later migrated the configuration during the Cisco acquisition
- **Built a self-service CLI for the security team** that created Jamf software restrictions in seconds instead of minutes of console clicking — removing myself as the bottleneck for a recurring request
- **Replaced a legacy Perl patcher that force-quit users' applications mid-meeting with a Python patching service** (Installomator + custom packages): stages updates in the background, applies on next app relaunch, and defers while a user is in a meeting — reaching **97–99% macOS patch compliance fleet-wide** (effectively every device that came online) with no user interruption, and ending recurring audit findings for stale software
- Cut security state-change enforcement from hours to **<15 minutes** — the management check-in floor — against a 3-minute security target, using SLO-backed alerts and Python/Bash auto-remediation; prototyped local polling daemons to close the remaining gap. Staged **global** OS rollouts in geographic waves targeted by IP range after opt-in UAT sign-off, containing the blast radius of a bad change to one region
- Built **self-healing recovery for always-on security agents** (CrowdStrike, Code42): each management plane could repair the other, so a downed agent or broken management framework recovered without hands-on remediation
- **Owned Puppet as the configuration-as-code layer across a heterogeneous fleet** — ~1,200 Windows and ~1,000 Linux devices under version-controlled state enforcement and drift reduction
- Built Docker-based GitLab CI/CD pipelines; automated AWS (Lambda, EC2, S3, CloudFront, Secrets Manager) with Terraform; led fleet vulnerability remediation (Splunk, Tenable, SonarQube) against SLAs
- Carried the team's **weekly on-call rotation** (VictorOps, later Splunk On-Call) — including through the 2024 CrowdStrike outage that took Windows endpoints down fleet-wide
- **Selected by the Director to set the technical bar for the team's hiring** — technical interviewer for senior and junior engineers and for the team's incoming manager

Starburst Data — IT / Cloud Engineer

Dec 2020 – Jan 2022

- **Owned Okta as the company IDP and the Harbor container registry** — the company's only IT engineer, running multi-cloud infrastructure in Terraform as headcount grew from ~100 to 300
- Codified Okta IAM in Terraform (groups, app assignments, SCIM): least-privilege access by default and same-day onboarding/offboarding through the growth
- **Automated container-registry access end to end** — replaced hand-provisioning with a Jira → Harbor API workflow issuing scoped credentials with an audit trail, cutting turnaround from hours to minutes; scoped the design and brought in a software engineer for the Jira API integration

Earlier career — Systems & IT Engineering

2013 – 2020 · Arent Fox, McDermott Will & Emery, LiveCare

- Enterprise Windows/macOS environments, Azure/O365 administration, PowerShell automation, VoIP migration lead

SELECTED PROJECTS

- **Copywarden** — copywarden.com — encrypted, screen-share-aware clipboard manager for macOS, designed and built solo in **Swift 6** (strict concurrency) with AES-256-GCM, HKDF-derived per-item keys, Secure Enclave key wrapping, Argon2id account crypto, and a downgrade-attack defense binding the envelope version into the AEAD tag. Signed and notarized universal builds, published threat model and disclosure policy, Next.js site on S3 + CloudFront. Built AI-assisted behind an enforced pre-push gate — **artifact integrity and build-pipeline trust** via SHA-pinned actions, secret scanning, tiered test lanes, and a claims linter that fails the build on unsupported copy
- **iRacing telemetry platform** — OpenTelemetry-native racing telemetry, built solo: each lap a trace, each corner a span, shipped over OTLP. FastAPI backend (16 data models, 14 routers, versioned migrations), six analysis services, and a packaged Windows collector released through GitHub Actions
- **evartanessian.dev** — this resume's claims, demonstrated: a static site that publishes its own live SLOs and error budget from real-user monitoring and synthetic probes; keyless OIDC CI deploys; S3 + CloudFront + ACM in Terraform; source and pipeline public (github.com/EddieV2/my-website)