

JAY CHITALIA

DEVOPS ENGINEER

Austin, TX · jaychitalia2@gmail.com · jchitalia.dev · github.com/jaychitalia95

SUMMARY

DevOps engineer, and genuinely passionate about it. I'm relentlessly curious, I learn by building, and I'm at my best making the platform the rest of engineering ships on faster and calmer. I lean on AI to execute and move quickly without outsourcing the thinking, and I care as much about lifting the team as I do about shipping the work.

SKILLS

AWS	Terraform [†]	Terragrunt	GitHub Actions	Docker	ECS	Step Functions
Lambda	Python	Bash	Snowflake	PagerDuty	Kubernetes	GCP
Agentic workflows		Multi-agent orchestration		AI harness optimization		Org-wide AI enablement

[†] HashiCorp Certified · Terraform Associate (004)

EXPERIENCE

Resonate Networks

Remote, USA

DevOps Engineer II (Mar 2025 – Present) · DevOps Engineer (Jun 2021 – Mar 2025)

Core engineer on a four-person DevOps team that owns our cloud infrastructure, CI/CD, and platform tooling. I treat the rest of engineering as my customers.

- Designed the reusable workflow-template pattern that standardized our CI/CD on GitHub Actions and brought a GitOps model to how we ship: engineers build and deploy in a single workflow, straight from the repo where they commit. The pattern stuck and now backs everything from ECS services to Lambda, Step Function, JAR, and Glue deploys; it's still how the org ships.
- Re-architected our ECS deployments so each service versions its per-environment task definitions alongside its own code, instead of in a separate Terraform repo, keeping application code and deploy config in sync through a single pipeline. That ended a class of task-definition drift outages and let engineers deploy with confidence.
- Replaced a single static CI runner with an autoscaling fleet of ephemeral spot runners that scales to zero when idle and up to ten under load. CI stopped being a bottleneck and runner cost dropped by roughly 60%.
- Shaped our Terragrunt monorepo so any engineer, increasingly with an AI agent alongside them, can land an infrastructure change without deep Terraform expertise. Also codified an entire disaster-recovery region (VPC, IAM, ECS, databases, dozens of services) so the business has a tested path back online.
- Built the alerting Lambda that, the moment a Step Function or Lambda failure pages PagerDuty, uses AI to investigate and attach a rich root-cause writeup with one-click links to the failing execution, function, and alarm. The hard part, piecing together what went wrong from the logs, is already done, so the on-call engineer can focus on the fix instead of the hunt.
- Layered quality gates across our repos, from pre-commit checks on the developer's machine to Dependabot and SonarCloud gates that block any failing merge, so defects in our mission-critical, customer-facing services are caught and fixed long before they can reach production.
- Wrote the Lambdas that keep our cloud bill honest: right-sizing oversized resources, enforcing S3 lifecycle/TTL rules, and shutting down idle dev environments. Cloud spend is operating expense, and the less we waste the more shows up in profits.
- Own our recurring security program: quarterly Qualys vulnerability scans, Lacework cloud-posture review, and SOC 2 findings, keeping our controls audit-ready.
- Built the CI/CD and test harness for our Snowflake Native App, learning Snowflake's publishing model from scratch (release channels, version caps, mandatory scans, automatic rollback) and writing the UAT suite. Nothing reaches a customer without clearing unit, integration, and UAT gates.
- Built the DaaS customer-access portal: a React front end on a serverless Python back end (API, queue, and worker Lambdas with DynamoDB), where a form submission becomes the Terraform change that provisions a customer's SFTP/S3 access, taking DevOps out of the critical path for onboarding.
- Champion AI inside the company: shipped AI-powered CI workflows (PR review, build review, autofix) into the shared templates so every repo gets AI by default, demo what I learn in standups and all-hands, and onboard and ramp new teammates by example rather than gatekeeping.

Searce Inc.

Austin, TX · Feb 2021 – Jun 2021

Cloud Engineer

- Migrated client workloads from AWS, Heroku, DigitalOcean, and on-prem to GCP; ran cost-optimization sessions and evangelized containers, Kubernetes, and 12-factor methodology.

Validity Inc.

Boston, MA · Jan 2020 – Aug 2020

Jr. DevOps Engineer

- Containerized a legacy DNS system with Docker + Helm on Kubernetes (~\$10K/yr saved); built Jenkins CI/CD, provisioned AWS with Terraform, and wrote Go Prometheus exporters with Grafana SLI/SLO dashboards.

EDUCATION

Northeastern University

Boston, MA · Dec 2020

M.S. Information Systems