Do Cloud Architects Have Values?

Engineering Ethics in the Age of Infrastructure Automation

Nigel Dsouza

In an era when infrastructure-as-code can spin up global systems in minutes and AI-driven deployments handle more logic than human hands, cloud architects are no longer just builders — they're gatekeepers of trust. I've seen this firsthand while leading the design of critical financial infrastructure at Fidelity Investments, where small missteps in architectural decisions can ripple across billions in transactions, investor confidence, and regulatory compliance.

But here's the uncomfortable truth: while we've accelerated delivery pipelines, we've often left behind the deeper reflection on **why** we build the systems we do — and **who they impact**.

Cloud Infrastructure Isn't Neutral

It's tempting to view AWS Lambda functions, CI/CD pipelines, or disaster recovery frameworks as morally neutral tools. After all, they're just lines of code, scripts, or Terraform modules — right?

But when you automate how, when, and where systems recover (or don't), you're making **judgment calls** about continuity, access, and resilience. When I helped lead the architecture of a real-time failover system for our Alternative Investments platform, the technical challenge was clear — ensure zero downtime with no manual intervention. But the **ethical challenge** was subtler: Should we prioritize geographic diversity or latency? Do we build for speed, or for fail-safes that might slow down releases but protect users better?

These questions aren't technical. They're values-based.

Invisible Decisions, Visible Consequences

Take CI/CD pipelines — one of my key contributions across multiple squads was designing reusable, Terraform-based deployment flows. These pipelines reduced deployment time by over 75%. But in enabling speed and standardization, they also **centralized control** over how software moves to production.

It raised questions: Who has visibility into what gets deployed? How do we prevent the erosion of quality in favor of speed? What governance is embedded — and where is it missing?

Without thoughtful reflection, tools that optimize developer experience can inadvertently obscure accountability.

Why Engineers Need a Values Framework

We don't need to turn every stand-up meeting into a philosophy debate. But we do need to embed **ethical review loops** into system architecture — just as we embed logging, monitoring, and unit tests.

Here's a starting point I've adopted with my teams:

- **Transparency over complexity**: Document not just how a system works, but why it was designed that way.
- **Fail safely, not just fast**: Prioritize graceful degradation over perfect uptime in edge cases.
- **Build for the humans at the edge**: That includes the junior developer debugging a prod issue and the investor checking their portfolio at midnight.

Engineering Culture Is the Real Infrastructure

No amount of cloud tooling can compensate for a culture that ignores ethical reasoning. I've mentored engineers who can deploy entire microservices ecosystems on EKS clusters — but struggle to articulate why a rollback strategy matters beyond uptime.

We need to teach not just **how to build scalable systems**, but also how to build **responsible systems**. Because if the infrastructure we're creating doesn't reflect the values we stand for, then we're just automating indifference at scale.

About the Author

Nigel Dsouza is a Principal Software Engineer and Technical Lead at Fidelity Investments. He specializes in architecting scalable, cloud-native systems on AWS, with a focus on financial technology, automation, and disaster recovery. Nigel holds two Master's degrees — in Computer Science and Engineering Management — and is a recognized contributor to enterprise-scale innovation in cloud infrastructure.