

Are There Values in Cloud Architects? In the Era of Infrastructure Automation, Ethics in Engineering

Nigel Dsouza

In a time when global systems can be spun up in minutes using infrastructure-as-code and AI-driven deployments manage more logic than human hands, cloud architects are becoming more than just builders; they are custodians of trust.

Leading the design of vital financial infrastructure at Fidelity Investments has allowed me to witness this directly, since even little errors in architectural choices may have a significant impact on billions of dollars in transactions, investor confidence, and regulatory compliance.

The unpleasant reality is that, despite speeding up delivery pipelines, we frequently neglect to consider the underlying reasons behind the systems we create and the people they affect.

Cloud Computing Isn't Neutral

It's easy to see CI/CD pipelines, disaster recovery frameworks, and AWS Lambda functions as tools that are morally neutral. They are merely scripts, lines of code, or Terraform modules, after all.

However, automating how, when, and where systems recover—or fail to—requires making decisions on resilience, continuity, and access.

The technological issue was obvious when I assisted in leading the construction of a real-time failover solution for our Alternative Investments platform: guarantee zero downtime without the need for manual intervention.

The moral conundrum, however, was more nuanced: Should we give latency or geographic variety priority? Do we design for speed or for fail-safes that could delay releases but better safeguard users?

These are not technical questions. They are founded on values.

Decisions That Are Invisible But Have Visible Effects

Consider CI/CD pipelines. Creating reusable, Terraform-based deployment flows was one of my major contributions to several squads. Over 75% less time was spent on deployment thanks to these pipelines.

However, they also centralized control over the software's production process while facilitating speed and standardization.

It made others wonder: Who can see what is deployed? How can we stop the sacrifice of quality for speed? What kind of governance is present, and where is it absent?

Tools that maximize the developer experience may unintentionally obfuscate accountability if careful consideration is not given.

The Need for a Values Framework in Engineering

Every stand-up meeting doesn't have to become a philosophical argument. However, ethical review loops must be incorporated into system architecture in the same way that unit tests, logging, and monitoring are.

I've used this as a beginning point with my teams:

- **Openness over intricacy:** Record not only a system's operation but also the rationale behind its design.
- **Don't just fail quickly, fail safely:** In edge scenarios, give gentle deterioration precedence over flawless uptime.
- **Build for the human at the edge:** This includes the investor checking their portfolio at midnight and the junior developer troubleshooting a production issue.

The True Infrastructure is Engineering Culture

A culture that disregards ethical reasoning cannot be made up for by any amount of cloud tooling.

I've worked with engineers that can set up whole microservices ecosystems on EKS clusters, but they find it difficult to explain why a rollback approach is important for reasons other than uptime.

In addition to teaching scalable system development, we also need to teach responsible system development. Because we're only automating apathy at scale if the infrastructure we're building doesn't align with the principles we uphold.

Concerning the Writer

Nigel Dsouza works for Fidelity Investments as a Technical Lead and Principal Software Engineer. His area of expertise is designing cloud-native, scalable systems on AWS, with an emphasis on automation, disaster recovery, and finance technology.

With two master's degrees in Computer Science and Engineering Management, Nigel is a well-known innovator in cloud infrastructure at the enterprise level.