We Are the Ancestors of Tomorrow's Systems

Designing Cloud Infrastructure as Legacy, Not Just Code

Nigel Dsouza

In tech, we worship the new. The newest frameworks, the latest releases, the cutting edge of edge computing. But here's a radical thought: **What if we built cloud systems like we were building cathedrals?**

Not for today. Not for this quarter's KPIs. But for the architects and engineers who will inherit our code long after we're gone.

As someone who's led the architecture of enterprise systems handling billions in transactions at Fidelity Investments, I've come to realize that the most overlooked stakeholder in modern cloud design is the one who doesn't exist yet: the engineer who comes after you.

What Will They Think of You?

When a developer opens your Terraform files five years from now — will they curse your name, or will they quietly thank you?

Will your IAM policies explain themselves? Will your deployment pipeline make sense without tribal knowledge? Will your system degrade with grace, or collapse with mystery?

Most of us code like we're writing temporary scripts. But the truth is, cloud infrastructure is now institutional memory. It's the new archaeology.

The Philosophy of Infrastructure

Let's stop pretending our systems are disposable. They're not. They get cloned, forked, dockerized, scaled, passed on.

And so, we must write with **legacy in mind**. Not in the "technical debt" sense — but in the moral sense. The way we speak of elders, ancestors, founding architects.

Here's a terrifying but beautiful question: *What values are embedded in your system design?*

Code as Culture

At Fidelity, my team builds cloud-native systems for alternative investments. Our pipelines span continents. Our DR plans cross failover zones. And yet the hardest part is not the tech — it's embedding **culture into code**.

Do we comment with care? Do we document decisions, not just endpoints? Do we leave tools behind — or wisdom?

A cloud function might run in 15 milliseconds, but its consequences can echo for years.

Your Future Teammate is Watching

Imagine writing every line of code like a message in a bottle.

To the next engineer. The next lead. The future you.

Not just: "Does it work?"
But: "*Will it last?*"
And more importantly: "*Does it teach?*"

Infrastructure as a Time Capsule

We talk a lot about scale, about automation, about optimization. But what about **intention**?

The best systems I've ever encountered weren't just fast or elegant. They were humane. Understandable. Designed by someone who cared who would inherit them.

You're not just an engineer. You're an ancestor.

About the Author

Nigel Dsouza is a Principal Software Engineer and Technical Lead at Fidelity Investments, where he builds global-scale cloud systems that outlive trends and toolchains. He believes that the most powerful metric in engineering is not performance — it's *legacy*.