

Severance Protocol

When Code Remembers What You're Paid to Forget

Nigel Dsouza

In a world where infrastructure deploys itself, AI models write our pipelines, and systems monitor our every move, a strange question emerges:

What if your code knew more about you than you did?

Welcome to the age of Severance Engineering — where our professional systems remember what our personal selves are told to forget.

The Two Selves of the Modern Engineer

If you've seen the show *Severance*, you know the premise: a chip in your brain separates your work memories from your personal life. The worker self lives entirely in the office. The personal self has no clue what happens inside.

Now think about your cloud infrastructure.

You've built pipelines that deploy without you. Lambda functions that run long after you've gone. Security policies written at 2am under pressure. IAM roles that grant trust you no longer feel. Terraform plans that haven't changed in three years — but still run daily.

Your systems have a memory.

Code is the Consciousness You Leave Behind

Every line of code is a decision. Every architecture diagram is a frozen belief. Every CI/CD pipeline is a philosophy.

And yet — we forget.

We move jobs. We rotate out. We burn out. We "sever."

But the system does not.

AI is Your Permanent Shadow

With generative AI entering our DevOps lifecycle, we are no longer just writing the infrastructure — we are training the next layer of it.

Every prompt, every pattern, every resolution becomes a behavioral artifact. Your tools remember. Your assistants autocomplete your legacy.

You are not writing infrastructure. You are writing memory.

Terraform, Jenkins, and the Archive of You

Think of it:

- Your Terraform modules are maps of your past risk tolerance.
- Your GitHub repo is a forensic mirror.
- Your observability dashboards reflect not just usage — but belief.

In *Severance*, the outie doesn't know what the innie does. In our world, it's reversed: the systems remember what the humans suppress.

The Ethics of Infrastructure Amnesia

We must ask:

- Who audits our intent when we're no longer around to explain it?
- Who maintains the mental model when the modeler is gone?
- Should code age out like memories do?

Maybe infrastructure needs a half-life. Maybe policies should come with expiry dates. Maybe systems should forget — gracefully.

Conclusion: Remember With Purpose

In this era, the line between developer and infrastructure is vanishing.

You don't just write code. You inscribe identity.

You deploy versions of yourself — into AWS, into Jenkins, into memory.

So write wisely. Comment generously. And leave breadcrumbs.

Because the system will remember. Even when you're severed from it.

About the Author

Nigel Dsouza is a Principal Software Engineer at Fidelity Investments. He writes code with the full knowledge that someday, someone else will live in it — and maybe, just maybe, try to remember who he was.