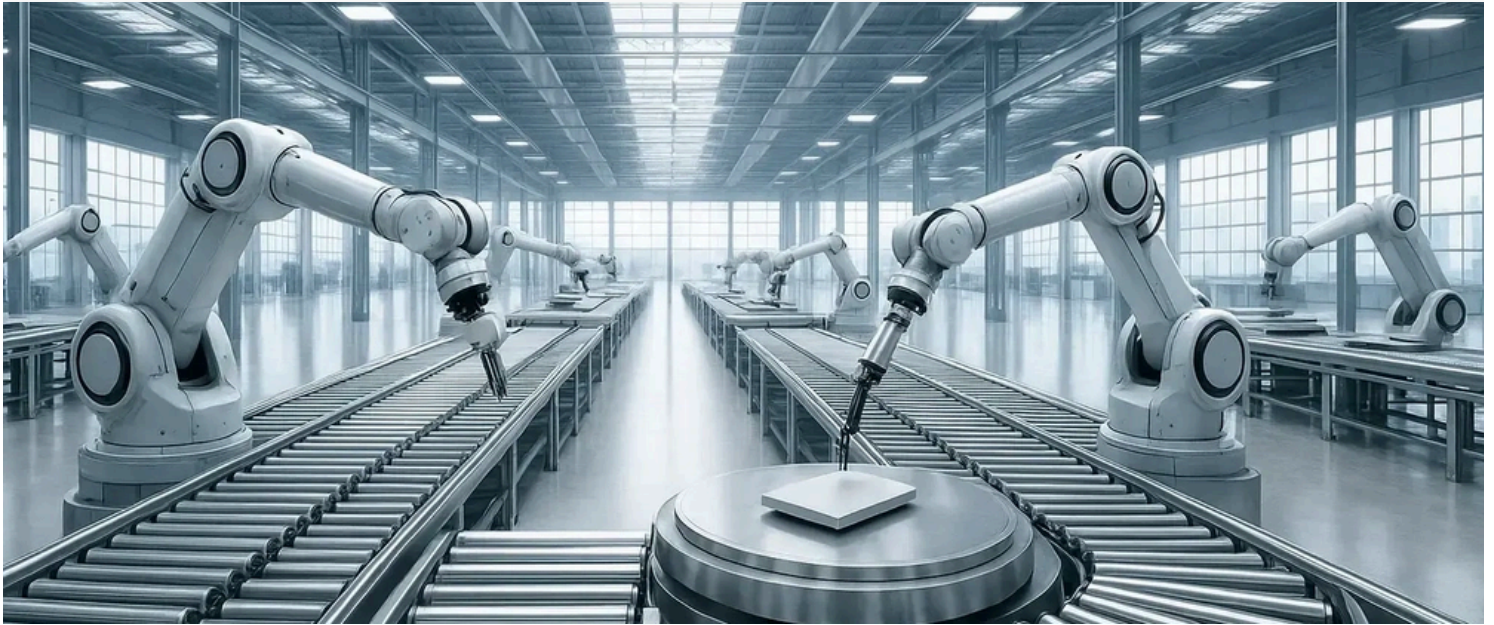


Unpublished Post. This URL is public but secret, so share at your own discretion. [Click to edit](#)



Nigel Dsouza

[Edit](#) [Stats](#)

The Art of Automation in Modern Engineering

[#devops](#) [#infrastructureascode](#) [#engineeringculture](#) [#cloudengineering](#)

The Art of Automation in Modern Engineering

Designing Systems That Think Without Forgetting to Feel

By Nigel Dsouza

We used to automate because we were lazy.

Now we automate because we must.

In a world of hyperscale systems, incident fatigue, and the unrelenting pressure of uptime, automation is no longer a luxury — it's the **nervous system** of modern

engineering.

But here's the catch:

Most automation isn't beautiful.

It's brittle. It's soulless. It gets the job done, but no one knows how. Or why. Or what to do when it breaks.

Which begs the question:

What if automation could be art?



Beyond Scripts and Cron Jobs

Real automation isn't about writing bash scripts or wiring up webhooks.

It's about **designing intent**.

It's about:

- Embedding wisdom into pipelines
- Anticipating failure
- Choreographing recovery
- Orchestrating workflows that feel alive

At **Fidelity**, I've built automation frameworks that touch everything from CI/CD to disaster recovery — and I've learned this:

Bad automation saves time.

Good automation saves teams.



When Systems Become Symphony

Think of your infrastructure like an orchestra:

- 🎵 **Jenkins** is your conductor
- 🎻 **Terraform** is the string section — precise, declarative
- 🥁 **Lambda functions** are percussion — fast, ephemeral, rhythmic
- 📢 **Alerts and monitors** are the brass — loud, essential, sometimes alarming

You don't just want them to play.

You want them to play **well together**.

That's the art.

⚠️ The Cost of Frictionless Systems

The danger with perfect automation is that it can remove too much friction.

- No one reads the logs.
- No one asks questions.
- No one remembers how things work — until they don't.

Which is why the best automation doesn't just remove humans.

It **supports them**. It **teaches**. It **leaves trails**.

It **respects context**.

✨ Designing for Delight, Not Just Delivery

Your pipeline can be a joy to use.

Your self-healing scripts can come with ASCII art.

Your alerts can be tuned like instruments — not alarms.

Because automation isn't just execution — it's **experience**.

And if we want engineers to trust the system, to enjoy using it, to build upon it, then we have to make it more than functional.

We have to make it **feelable**.



Conclusion: Automate Like an Artist

The next wave of engineering leaders won't be those who can write the fastest script — but those who can build systems that run themselves **without losing their soul**.

So the next time you write a pipeline, ask:

- Does this automate a task, or teach a mindset?
- Will the next person understand it, or fear it?
- Did I just save time — or create trust?

Because automation isn't just a tool.

It's a **medium**.

And some of us were born to make it **art**.



About the Author

Nigel Dsouza is a Principal Software Engineer and Technical Lead at Fidelity Investments,

where he designs automated cloud systems that are as resilient as they are human.

He believes every `Jenkinsfile` is a brushstroke —

and every pipeline, a performance.

Top comments (0) ↕



Nigel Dsouza

Tech Lead @ Fidelity | Full Stack Dev (Java, Node.js, AWS, Terraform) | CS + Eng Mgmt Grad | Cloud, CI/CD & System Design nerd | Rusty's human 🐾 | Always building, always learning.

JOINED

May 10, 2025

Trending on DEV Community 🔥



Game developers who are not artists, how do you manage?

#gamedev #design #discuss



Build a Terminal Task Manager in Python with pyfiglet and rich!!!

#python #programming #productivity #cli



Stop Copy-Pasting Dependency Injection Setups , Understand It First

#webdev #appconfig #app #devops