

The Hidden Carbon Cost of Your Cloud App

Why DevOps Needs to Go Green

Nigel Dsouza

Every time you spin up an EC2 instance or auto-scale a Kubernetes cluster, you're not just deploying code — you're burning energy. Yet, as engineers, we rarely stop to ask: What's the carbon footprint of my deployment?

Having spent the last several years designing high-availability systems for financial platforms at scale, I've seen firsthand how performance, uptime, and velocity dominate architectural decisions. But as climate change accelerates, our infrastructure choices must be evaluated not just by SLA — but by CO2.

Cloud Isn't Green by Default

It's easy to assume that cloud is “greener” than on-premise infrastructure. Providers like AWS, GCP, and Azure do invest in renewable energy and data center efficiency. But here's the catch: **optimization is your responsibility, not theirs.**

Provisioning oversized instances, underutilizing resources, or leaving staging environments running overnight all contribute to waste — and emissions. According to a 2023 report by the International Energy Agency, data centers accounted for nearly 1.5% of global electricity consumption. That's not abstract — that's us.

DevOps Practices Need a Sustainability Layer

Infrastructure-as-code has made deployment easy. But it's also made overprovisioning frictionless. We need a new mindset: **Sustainability-as-Code.**

Here are practices I've introduced in my cloud engineering work:

- **Tag everything with purpose and TTL:** Auto-delete resources after testing windows expire.
- **Use serverless architectures wisely:** Lambdas can be more efficient, but concurrency settings and payload bloat can negate gains.

- **Leverage carbon-intelligent regions:** AWS and GCP both offer cleaner-energy regions — deploy where it's green.
- **Measure cloud emissions:** Tools like Cloud Carbon Footprint or AWS Customer Carbon Reports should be as standard as cost dashboards.

Sustainability Is an Engineering Problem

The best engineers don't just write clean code — they leave a clean footprint. We obsess over CPU utilization and cost per request. Why not grams of CO2 per deployment?

Imagine if green KPIs were part of your pipeline metrics. Imagine if we bragged about cutting carbon like we do latency.

At Fidelity, our work on disaster recovery and failover infrastructure gave us the chance to rethink cross-region strategies. By selectively routing low-priority traffic to greener regions during off-peak hours, we reduced our carbon load without sacrificing SLAs.

Cloud Leaders Must Set the Standard

Sustainable architecture isn't just a nice-to-have — it's a leadership responsibility. Just as we invest in uptime and automation, we must design for energy efficiency.

If you're a cloud lead, ask yourself:

- Is your infrastructure elastic — or just oversized by default?
- Are your developers trained to see carbon as a cost?
- Are you reporting on sustainability as part of your engineering KPIs?

The green cloud isn't someone else's problem. It's ours. And the sooner we start treating carbon like technical debt, the faster we'll build a future we can actually sustain.

About the Author

Nigel Dsouza is a Principal Software Engineer and Tech Lead at Fidelity Investments. He specializes in cloud-native infrastructure, DevOps automation, and building scalable, efficient systems in AWS. He is passionate about sustainable architecture and embedding climate-conscious practices into cloud development.