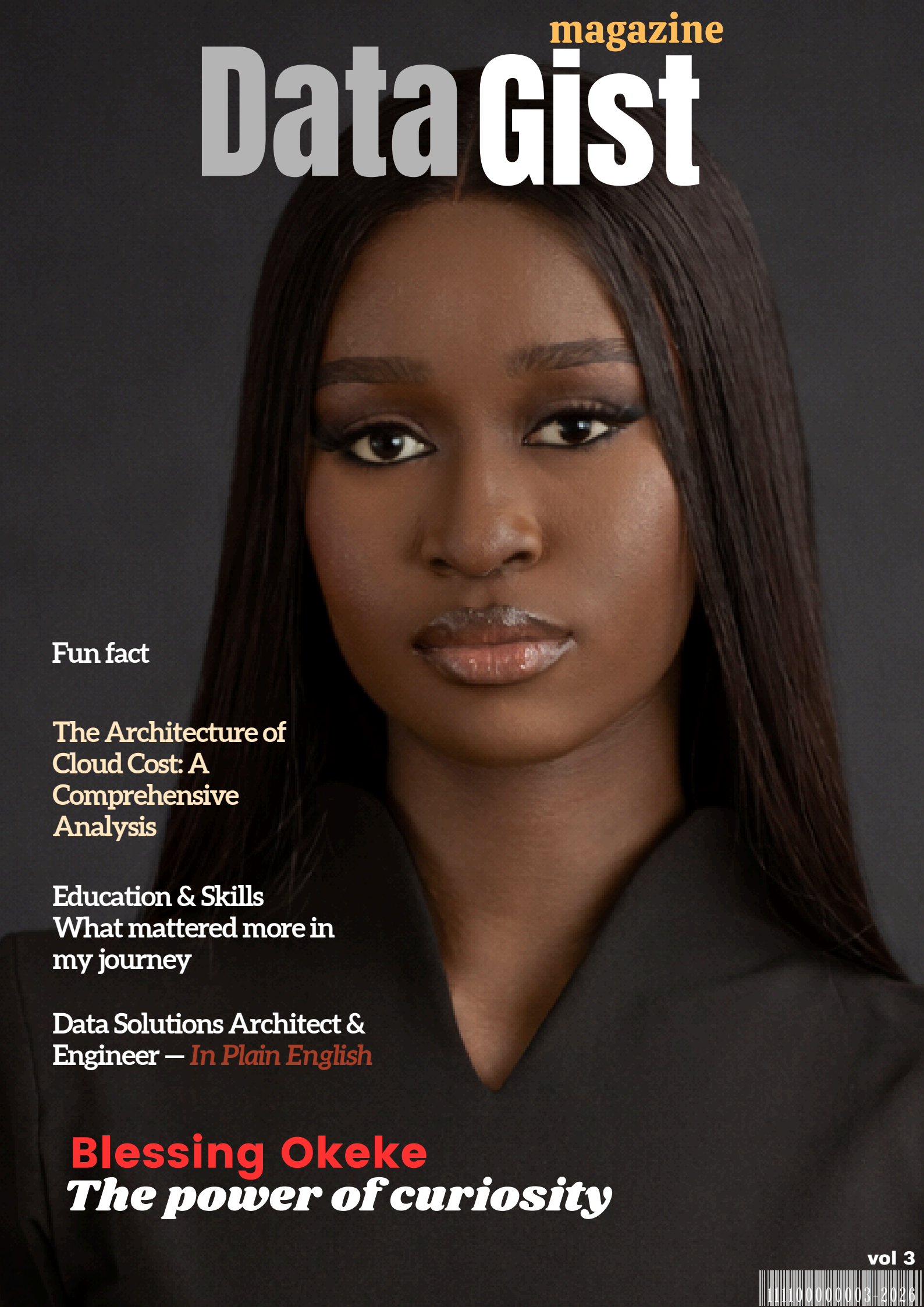
magazine

Data Gist



Fun fact

The Architecture of Cloud Cost: A Comprehensive Analysis

Education & Skills
What mattered more in my journey

Data Solutions Architect & Engineer — *In Plain English*

Blessing Okeke

The power of curiosity

vol 3

The power of curiosity

A data solution architect and engineer, *Blessing Okeke* with a background in Industrial Mathematics, where she developed a strong foundation in structured thinking and problem-solving, which she now applies to building reliable data systems and translating complex data challenges into practical solutions.

As a Data & Analytics Engineer at a Big Four consulting firm, she designs and builds data solutions that help organizations move from scattered information to confident, insight-driven decisions.

Beyond her consulting work, she is a committed data educator who has trained numerous professionals in data engineering and analytics. Whether building pipelines or teaching others how to think with data, her focus remains the same: making data understandable, useful, and impactful.

What sparked my interest in data engineering?

Funny enough, data wasn't the original plan. I studied Industrial Mathematics and had a clear path in mind, I was going to become an actuary. I enjoyed calculations; numbers felt calm, predictable, and I had already mapped out the journey: take two actuarial exams before graduation and continue from there.

"However, curiosity led me in a different direction."

During my fourth-year internship at a data and analytics company, what started as simple exposure quickly turned into exploration. It was my first real encounter with the world of data, and I found myself drawn to it, the blend of logic, systems, and real-world impact. Before then, I didn't fully understand what the field involved, but once I stepped into it, something clicked. I remember thinking, "This is it." And from that moment, I never looked back.

What was my "almost gave up" moment in data?

Interestingly, I never experienced a dramatic "this field isn't for me" moment. However, there was a phase that truly tested my resolve, the period right after all the learning and preparation, when it was time to break into the job market.

"It felt like everyone was ahead, and it was easy to question whether breaking into the field was even possible.

What kept me disciplined when things got tough?

What kept me motivated was a combination of curiosity and community. The dynamic nature of the field meant there was always something new to learn, new ideas, tools, and better ways to solve problems, which made the journey continually exciting.
Equally important was the data community. From peers to mentors and individuals sharing knowledge online, the openness and support within the ecosystem played a significant role in sustaining my progress and growth.

"There's a saying that it takes a community to raise a child, and in many ways, I consider myself one of the many professionals shaped and supported by the data community."

Tea or coffee? and what does my choice say about how I approach debugging?

Tea.

My debugging method is very tea-aligned: when the bug gets stubborn, I close everything, step away, maybe squeeze in a quick nap, then come back with fresh eyes. Nine times out of ten this strategy works like magic… I suddenly see the problem that was hiding in plain sight.

Education & Skills What mattered more in my journey?

A combination of both theory and practice has been essential in my development, with a stronger emphasis on hands-on experience approximately 70% practical and 30% theoretical. While formal education provided a solid foundation in logical and analytical thinking, true understanding came from building solutions and solving real-world problems. Ultimately, problem-solving remains the most critical skill, as tools continue to evolve, but a strong grasp of fundamentals is indispensable at every stage.



Blessing Okeke

How I stay relevant without burning out?

The data and technology space moves incredibly fast. There's always a new tool, platform, or approach emerging, and if you try to keep up with everything, burnout is almost inevitable.
The fast pace of the data field can easily lead to burnout, so I rely on curiosity rather than pressure to guide my learning. I focus on understanding new tools through fundamental concepts, learn at a steady pace, and prioritize what's relevant. I also take breaks to enable me reset, for example, during one of my breaks, I explored cinematography an entirely different environment with new people, perspectives, and a creative process. That experience helped me recharge mentally and return with renewed clarity and focus.

Data Solutions Architect & Engineer — *In Plain English?*

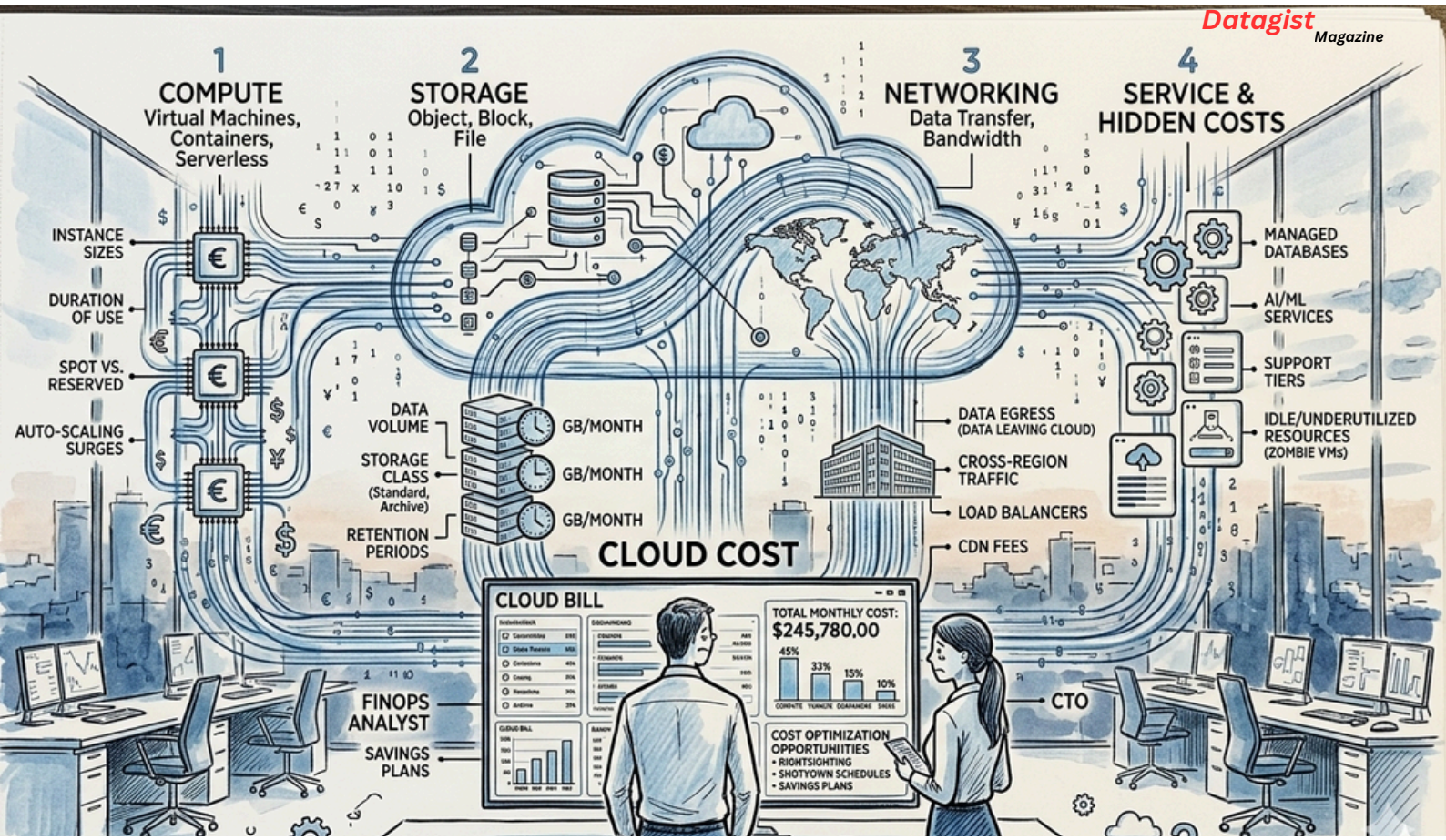
"I architect the solution and then engineer it."

In simple terms, solving problems using data. My role is to analyze problems, design efficient data flows and structures, and build the systems that enable reliable and scalable data solutions.

Standing Out?

Standing out to me simply means being truly good at what you do,

It's a 10/10 ecosystem



The Architecture of Cloud Cost: A Comprehensive Analysis

The Invisible Utility: Decoding the Modern Cloud Bill

For decades, building a digital business meant buying steel. IT departments were anchored by humming server rooms, heavy depreciating hardware, and massive upfront investments the classic Capital Expenditure (CapEx) model.

Then came the cloud, and everything shifted.

At its simplest, cloud cost is the price of renting the future. It is the total amount of money a business spends to harness remote computing resources the servers, databases, and global networks owned by tech giants like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), etc.

By migrating to the cloud, enterprises traded heavy iron for pure agility, pivoting to an Operational Expenditure (OpEx) model. Instead of buying physical hardware upfront, businesses now pay a variable, recurring "utility bill" based purely on what they consume.

It is the ultimate double-edged sword of modern business. The cloud offers infinite, frictionless scalability at the push of a button. But as many modern enterprises are finding out, when you leave the faucet running on a utility this powerful, the monthly bill can drain a budget before you even realize it's on. To master the digital economy, leaders must first master the architecture of cloud cost.

Compute -*The Engine of Innovation*

The Compute pillar is the primary powerhouse of the cloud, typically accounting for 45% of total expenditures. While it drives application performance, its flexibility often leads to significant waste if not strictly governed.

Core Drivers of Compute Cost

- Virtual Machines (VMs) & Instances: Costs are dictated by the "family" selected (e.g., memory-optimized vs. compute-optimized). Choosing a powerhouse instance for a simple task is a leading cause of budget overruns.
- Containers & Kubernetes: While offering high efficiency, the management layers and underlying infrastructure clusters introduce their own set of overhead costs.
- Serverless (e.g., AWS Lambda): Billed per execution and duration. It is highly cost-effective for "bursty" workloads but can become a liability if used for continuous, high-volume processing.

Optimization Strategies

- Rightsizing: The practice of matching instance performance to actual workload demand. Scaling down from a large to a medium instance can yield immediate savings.
- Autoscaling Caps: While autoscaling handles traffic spikes, without "caps," an unexpected surge can lead to a massive, unchecked bill.
- Purchasing Models: * Reserved Instances (RIs): Committing to 1–3 years for discounts up to 70%.
- Spot Instances: Using spare capacity for discounts up to 90%, ideal for fault-tolerant tasks.

Storage -*The Data Dilemma*

The Storage highlights the cost of holding the vast amounts of data generated and managed by modern organizations. Storage costs are deceptively complex, involving multiple tiers and hidden access fees.
It isn't just about how much data you have, but how often you touch it and how long you keep it.

The Lifecycle of Data Costs

- Data Volume: The absolute amount of data stored (measured in GB or TB). This is the most visible cost driver.
- Storage Class (Standard vs. Archive): Cloud providers offer different storage tiers. Standard storage is expensive but provides instant access. Archive storage (e.g., S3 Glacier) is very cheap but has significant costs for data retrieval and longer retrieval times.
- Retention Periods: The length of time data is kept. Organizations must define clear policies; holding data "forever" because storage seems cheap is a costly mistake.

Optimization Specifics:

- Lifecycle Management Policies: Automating the transition of data between storage classes based on access frequency. For example, moving data from standard to archive after 90 days.
- De-duplication & Compression: Reducing the physical footprint of the data itself.

Networking – *The Cost of Connectivity*

Networking is the virtual architecture that connects an enterprise's compute power, databases, and applications to each other and to the global internet. Networking is perhaps the most critical and least understood source of hidden costs. In modern operational expenditure (OpEx) model, data movement is metered. Understanding how traffic flows through this virtual grid is critical, because in the cloud, architecture is finance.

The Complexity of Data Transfer

- Data Egress (Data Leaving the Cloud): This is the critical cost driver highlighted in the diagram. Data Ingress (data coming in) is typically free. However, any data that leaves the cloud provider's network to the internet or to an on-premise location incurs costs. This includes serving content to users or backing up data to an external location.
- Cross-Region Traffic: Sending data between different geographical regions of the same cloud provider (e.g., from AWS US-East to AWS EU-West) often incurs separate, substantial transfer fees.
- Load Balancers: Distributing traffic among multiple targets (e.g., EC2 instances). The balancer itself has an hourly cost, plus costs for the data it processes.
- CDN Fees: Content Delivery Networks (like CloudFront) speed up delivery by caching content globally. While they reduce direct egress costs from the origin, they introduce their own tier-based transfer pricing.

Optimization Specifics

- Architecture Design: Designing applications to minimize data movement, such as processing data in the same region where it is stored.
- VPC Peering & Endpoints: Using direct, internal network paths to connect services (e.g., using an S3 Gateway Endpoint), which can bypass expensive data transfer fees.

Service & Hidden Costs - *The Long Tail*

It is a common misconception that cloud optimization is only about core infrastructure. While Compute, Storage, and Networking are the heavy hitters on a cloud bill, the Long Tail of specialized services is where budgets quietly derail. In the modern cloud, convenience comes at a premium. Managed Services like automated databases, serverless functions, and machine learning pipelines relieve the burden on IT labor, but they introduce steep, variable usage fees. Neglecting this pillar is a costly mistake; individually, these line items seem harmless, but left unchecked, they can hollow out an enterprise IT budget.

Key Elements

- Managed Databases (RDS, DynamoDB): Outsourcing database management to the cloud is a great time-saver, but it shifts labor costs into permanent infrastructure premiums.
- AI/ML Engines: From pre-built models to custom SageMaker deployments, machine learning is a massive resource drain. Costs can spiral overnight based on training durations, data volumes, and live endpoint queries.
- The Escalating Support Tax: Enterprise support plans are tied directly to your total usage. When your compute bill spikes, your support bill spikes right alongside it.
- Zombie VMs: Abandoned testing environments are the silent killers of enterprise margins. Left unmonitored, these idle virtual machines rack up thousands in phantom charges.

Optimization Specifics

- Tagging & Financial Attribution: You cannot optimize what you cannot measure. Comprehensive tagging bridges the gap between engineering and finance, allowing enterprises to trace every single dollar back to a specific project or department.
- Automated Off-Switches: Modern cloud discipline replaces human oversight with automation. By scripting the automatic shutdown of dormant dev environments and orphaned storage disks, companies can eliminate waste before the weekend even starts.

FinOps- The Integrated View

When you look at the "Cloud Bill" flanked by the CTO and a FinOps analyst, it underscores a vital truth: you cannot manage modern tech in a vacuum. It takes a unified, data-driven approach to keep digital infrastructure from derailing a budget.

FinOps (Financial Operations) is the cultural and operational discipline that brings financial accountability to the variable, real-time world of cloud computing.

The modern cloud is an ecosystem of hidden cause-and-effect. A global footprint creates massive, interconnected dependencies. Heavy compute scaling in one region can silently trigger a cascade of cross-border networking and egress fees. In the cloud, a single architectural shift ripples across the entire balance sheet.

Optimization

Optimization is not a static, one-time audit, it is a continuous operational loop. To win, teams must lean into three high-ROI actions:

- Precision Rightsizing: Trimming fat by matching provisioned capacity to actual traffic demand.
- Dynamic Shutdowns: Automating the termination of idle, non-production environments (Dev/Test) after hours.
- Savings Plans: Locking in massive volume discounts by mapping out predictable, steady-state usage.

By mastering this four-pillar architecture and embedding a rigorous FinOps culture into daily engineering workflows, organizations can flip the script. What was once a chaotic, unpredictable cost center becomes a lean, highly optimized engine of strategic enterprise growth

fun fact

DID YOU KNOW

Data scientists spend nearly 80% of their time cleaning and preparing data, not building models.