AWS for Industries

# Blazing a Trail: How Peloton Rebuilt the SDLC for the Agentic Era with Amazon Bedrock

by Jay Ramachandran, Abhay Kulkarni, Alex Niderberg, Esther Agyekum, Michael Luzzi, Taq Karim, and Tomasz Mysliwiec | on 08 JUL 2026 | in Amazon Bedrock, Amazon DynamoDB, AWS CloudTrail, AWS Identity and Access Management (IAM), Industries, Retail | Permalink | Comments | Share

Just like a Peloton HIIT ride, the highest-performing engineering teams in 2026 aren't pedaling harder: they're sprinting on the problems that demand human judgment (intent, architecture, quality) and letting autonomous agents handle repetition (generating code, running tests, triaging failures, opening pull requests).

Peloton took this philosophy and rebuilt the entire AI-driven development life cycle (AI-DLC) around it. The company didn't bolt an AI assistant onto an existing workflow. Instead, it built a foundation on Amazon Web Services (AWS) that shifted engineers from writing code to defining intent, constraints, and quality bars and let autonomous agents handle the rest.

Peloton uses Amazon Bedrock for access to frontier models, including Anthropic's Claude Sonnet 4.6, Opus 4.7 and Opus 4.8. Amazon Bedrock also provides global cross-region inference, integration into AWS CloudTrail and Amazon CloudWatch, and model access logging for the observability and audit controls its engineering organization requires.

## The problem: Credential sprawl kills AI adoption at scale

When Peloton committed to accelerating AI adoption across its engineering and product organization, the first blocker wasn't the models; it was the credentials. Static API keys are a security risk at scale. Individual engineers juggling credentials creates toil, exposure, and an uneven adoption curve. Agentic AI workflows require a different architecture entirely: one where the agent itself can act on behalf of the organization, not just a single user. The solution was a pair of complementary platforms built on AWS.

## Quarry: Amazon Bedrock access for every engineer

Quarry is Peloton's internal credential portal and AI coding gateway, backed by Amazon Bedrock. Before Quarry, AI adoption was limited by the friction of obtaining and managing credentials. Engineers manually requested static API keys, stored them in dotfiles and environment variables, and had no centralized visibility into usage or cost. This created a security risk: once static keys are copied into local environments, they become difficult to monitor, rotate, revoke, or constrain to approved workflows.

Quarry replaced all that friction with a single flow: engineers authenticate with Peloton SSO and receive automatically refreshing, short-lived AWS Security Token Service (STS) credentials scoped to Amazon Bedrock. One command on a Peloton-issued Mac delivers a token to Claude Code, Windsurf, VS Code, or Xcode. The same credential flow works through a browser-based web portal. Tokens expire and rotate automatically with no long-lived access keys, eliminating manual juggling and making the environment more secure. The native AWS security stack—AWS Identity and Access Management (IAM), AWS (STS), and AWS CloudTrail—and its deep integration with Amazon Bedrock made this design possible.

All AI model inference (Claude Sonnet, Haiku, Opus, and Claude Code) routes through Amazon Bedrock, giving Peloton a single secure plane for model access, usage tracking, and cost attribution. Amazon DynamoDB stores all session state, mission directives, outputs, context, and user history while powering low-latency dashboards and analytics. Amazon CloudWatch and Logs Insights capture every Amazon Bedrock invocation, feeding a real-time leaderboard that shows per-engineer usage, model calls, and token consumption. The leaderboard is visible to all engineers, and it fosters community, not compliance.

Quarry operates with zero static API keys, and contractor access runs on the same secure path with per-user token tracking, cost attribution, and model selection built in. Today, Quarry serves 600+ users, including 534 who were active in the last month, along with 10+ cross-organization contributors building on the platform.

As Quarry creator Tom Mysliwiec put it: "I've personally handwritten exactly 0 lines of code on this project."

## Platform 2 Bureau: Autonomous AI agents on Amazon EKS

Where Quarry puts AI in engineers' hands, Bureau acts on behalf of the organization. Bureau is Peloton's autonomous AI agent platform, executing end-to-end software missions from a single natural-language directive. It runs inside Peloton's infrastructure with access to GitHub, AWS, Slack, databases, Jira, Datadog, and Google Workspace.

The architecture is straightforward but powerful: Bureau receives a directive through its web interface, provisions an EKS agent pod, executes the mission using Claude on Amazon Bedrock through the same short-lived token model as Quarry, and terminates cleanly. Every action is logged. Every session produces a structured report.

**Code and engineering automation.** Nearly half of all missions—48.7 percent—involve engineers submitting directives like "review this PR for correctness and security" or "implement this feature and open a PR." Bureau handles cloning, coding, testing, and the PR lifecycle autonomously.

**Non-engineers shipping production software.** Product Managers (PMs) use Bureau to build fully functional web apps—deployed with shareable URLs—from PRD descriptions alone. In March 2026, six PMs independently built and deployed working prototypes without a single engineering sprint, recovering an estimated 18 engineer-days in one month.

**Security without tickets.** Sr. Application Security Engineer Paul Bryant described the experience: "I let this run yesterday and went to pick up my kids. I came back with a passing fresh build and a PR, and no more high or critical CVEs in the dependencies." A full dependency audit, PR, and clean build—without hands-on engineering effort.

**CI/CD self-improvement loops.** When an instrumented GitHub Actions workflow fails, Bureau is automatically dispatched. It reads the failure logs, identifies root cause, opens a fix PR, and posts findings to Slack—engineers only engage on judgment-call issues.

**Automated weekly status reports.** A Sr. Program Manager built three fully automated Bureau protocols that generate and post program status reports to Slack, pulling from GitHub, Jira, and Confluence. These reports post weekly, without any human interaction.

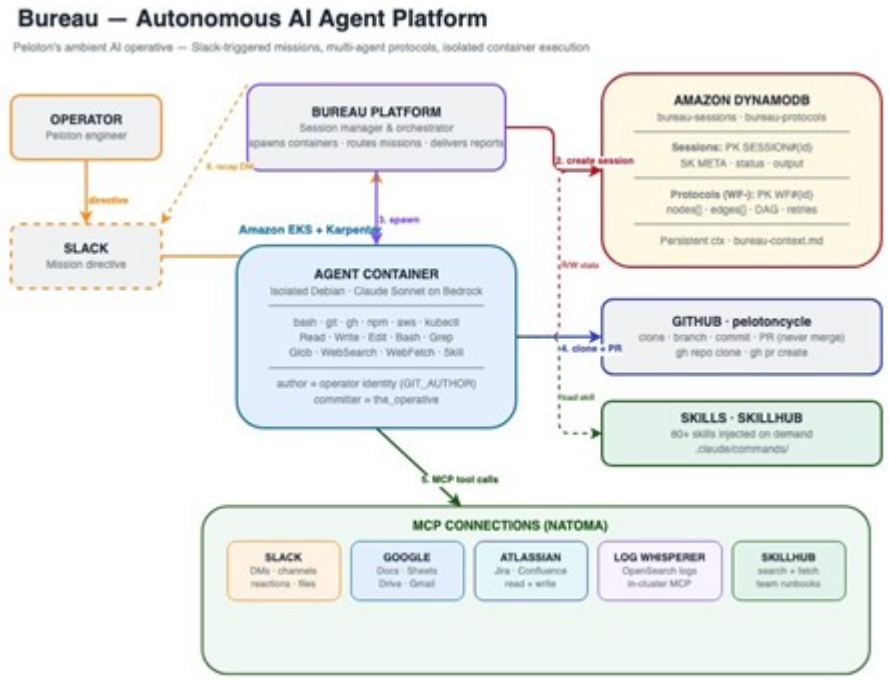


*Figure 1. How Bureau executes autonomous software missions*

## The numbers

In Bureau's first 20 days of operation, the results speak for themselves:

- 517% user growth from February to March 2026
- 184 unique users running Bureau missions in 18 days
- 1,151 human-initiated AI missions in 18 days
- 347 scheduled/automated missions running daily with zero human input
- 400+ pull requests merged, averaging 33+ per day
- $370K+ in annual savings identified
- $120K in annual cloud savings (Cloudflare LB reduced from $11K/month to $1K/month in 2.5 days—a task that previously took four weeks)
- $50K in vendor spend eliminated (hackathon portal replaced with AI-built alternative)
- ~2 hours from product idea to flag-gated production feature (Club Peloton Points Optimizer)
- Zero static API keys in the system
- Zero CVEs remaining after autonomous security audit

## Key takeaways for AWS customers

1. **Build a credential portal, not a policy.** A centralized portal wrapping Amazon Bedrock with SSO authentication and auto-refreshing tokens takes friction to near zero. By using short-lived STS credentials, Peloton completely eliminated static credentials. Engineers can now adopt tools they don't have to configure, and Peloton's leaderboard turned adoption into a community sport.
2. **Separate access for humans vs. agents.** Quarry serves interactive human developers. Bureau serves autonomous agent workflows. The same underlying Amazon Bedrock and IAM primitives, but different token

lifetimes, scopes, and usage patterns, designed as separate surfaces from day one.

3. **Make the multiplier the platform, not the feature.** The highest-ROI work Peloton did was making AI access frictionless for every role—not just engineers. When PMs, security engineers, and program managers can trigger autonomous workflows with the same ease as opening a browser, the benefits compound across the entire organization.

## The cool down

The highest-performing engineers at Peloton trained like the best riders on the leaderboard: they stopped grinding at a constant pace and learned when to sprint and when to let the machine carry the cadence. They didn't write code they could automate, manage credentials they could eliminate, or review PRs that a machine could evaluate.

They didn't add AI to their workflow. They rebuilt the workflow around AI. The future of software development isn't AI-assisted. It's AI-native systems, running on AWS.

## Start building

Peloton's AI-DLC platform is powered by AWS services you have access to: Bedrock, STS, EKS, DynamoDB, and CloudWatch. Start building your own version of Quarry and Bureau today, with built-in AWS security, governance and observability.


### Jay Ramachandran

Jay Ramachandran is a Sr Solutions Architect in the retail and ecommerce space with deep expertise in software development, Infrastructure as Code, and security. He helps organizations design and build secure, scalable systems on AWS. He is passionate about applying Generative AI to solve real business problems

### Abhay Kulkarni

Abhay Kulkarni is a Senior Technical Account Manager at AWS, designing and supporting large-scale cloud-native architectures for retail customers. He focuses on best practices, ensuring optimal performance and reliability of cloud infrastructure.

### Alex Niderberg

Alex is an engineering leader at Peloton focused on high-availability systems, reusable product tooling, and AI-driven workflow optimization. By bridging the gap between complex infrastructure and practical AI enablement, he empowers teams to deliver faster, safer, and more scalable member experiences. Alex's career is rooted in a culture of rapid prototyping; as a founding member of Capital One Labs, he scaled numerous concepts from napkin sketches to in-market tests with thousands of

users. He is a passionate advocate for the developer community, having co-led five Peloton Global Hack Weeks and serving as a judge and mentor for national hackathons.

## Esther Agyekum

Esther is a Technical Account Manager at AWS supporting Enterprise customers in the retail industry. She helps customers implement secured and resilient cloud environments and navigate the adoption of AI-driven solutions that transform how businesses operate and serve their end customers.

## Michael Luzzi

Michael Luzzi is an Area Principal Customer Solutions Manager (CSM) with AWS since 2018. In his role, Michael utilizes his 30+ year's CPG & Retail experience to assist customers in solving business challenges in these industries utilizing AWS services and solutions. He works closely with customers to apply Generative AI to help teams build faster and collaborate more effectively.

## Taq Karim

Taq is a veteran technologist and engineering leader with a deep background in building the infrastructure that powers global fitness experiences. As the Senior Director of Engineering at Peloton, he oversees the core backend services and platform architecture that support millions of users. With previous experience as an adjunct professor at Baruch College and an instructor at General Assembly, Taq specializes in making complex technical concepts—such as platform scaling and AI integration—accessible and actionable for non-technical professionals. He holds a degree from The Cooper Union and is a native New Yorker dedicated to high-level education.

## Tomasz Mysliwiec

Tomasz Mysliwiec is a Senior Engineering Manager at Peloton Interactive, known for his builder's mindset and pragmatic approach to problem solving. He leads platform and traffic engineering efforts across Kubernetes, cloud infrastructure, and large-scale service reliability, focusing on systems that are both resilient and easy for engineers to operate. Tomasz excels at turning complex, distributed challenges into clear, actionable solutions—whether driving infrastructure modernization, improving cost efficiency, or building internal tools that connect and streamline workflows.

## Comments

Log in to comment