



Case Study

# How a large financial services firm standardizes local AWS development with LocalStack

# The challenge: Slow iteration cycles and high AWS test spend

With thousands of engineers supporting a broad portfolio of AWS-based applications, cloud efficiency and developer productivity were strategic priorities. But as teams scaled modern workloads—serverless functions, Kubernetes microservices, database-backed systems, and AI integrations—recurring bottlenecks often slowed delivery and increased costs.

Engineering teams ran into two persistent blockers that made day-to-day development harder to standardize and more expensive to operate:

- **Provisioning delays:** Developers often had to wait on cloud provisioning to iterate, validate changes, and test service interactions—slowing feedback loops for common cloud workflows.
- **High test-environment spend:** Several teams saw elevated AWS costs in test environments, especially for heavy serverless/Kubernetes usage and complex database workloads.

Over time, these constraints created friction across the developer workflow and made it harder to scale consistent cloud development practices. At enterprise scale, that translates into slower time-to-market and higher total cost of ownership for cloud development and testing.

## Risks if not addressed

As AWS usage expanded across teams, the organization's cloud development workflow became increasingly dependent on shared cloud environments for everyday iteration and testing. What started as "some teams waiting on provisioning" quickly became a systemic drag on delivery: delays multiplied across teams, costs scaled with every test cycle, and engineering leaders had less leverage to enforce consistent patterns. In a large organization, these issues don't stay local—they compound across roadmaps, release trains, and budgets. Without a standardized approach to local AWS development and testing, the company risked:

- **Slower iteration cycles** that negatively impact internal developer experience metrics
- **Sustained (or rising) AWS test spend** driven by iterative dev/test in the cloud
- **Inconsistent practices across teams**, making it harder to scale standardized cloud development workflows

### Industry

Financial services

### Engineering scale

1K–5K engineers across the organization

### Teams involved

Platform Engineering / DevEx + application teams (serverless, Kubernetes, data, AI-integrated workloads)

### Cloud footprint

AWS-first (AWS-based development across serverless, Kubernetes, databases, and AI-integrated workloads)

### Strategic goal / initiative

Evaluate → prove → standardize local AWS development as an approved engineering practice (phased rollout)

### Org priority

Reduce provisioning bottlenecks and AWS test spend while improving iteration speed and standardization across teams

## What mattered in the decision

To eliminate the dependency on slow, cloud-based feedback loops, the organization needed more than “a faster test tool.” They needed a solution that could become an approved standard—one that engineering leaders could roll out with confidence across teams, without disrupting existing workflows or creating new governance concerns.

In practice, that meant balancing developer productivity with operational realities: AWS fidelity, service coverage, integration into CI/CD, and a clear adoption path from a few high-impact teams to broader standardization.

With those constraints in mind, the organization prioritized a solution that could:

- Provide a **controlled local environment** while maintaining AWS parity and service coverage expectations
- **Reduce cloud test spend** by shifting development and testing off the cloud
- Enable earlier validation of **infrastructure-as-code (IaC)** and cloud workflows, without waiting on cloud provisioning
- Scale through a **phased rollout**—starting with high-impact teams and expanding as outcomes are validated

## The solution: Enterprise-scale local AWS emulation platform

To reduce dependence on slow, cloud-based feedback loops, the organization evaluated LocalStack as a controlled local AWS development environment. By emulating AWS services locally and integrating into existing development and CI workflows, teams could validate AWS workflows earlier in the lifecycle—before changes reached shared AWS environments—reducing wait time and improving consistency across teams.

### Primary workloads included in the evaluation

- **Serverless workflows** (e.g., Lambda)
- **Kubernetes-based microservices** (e.g., EKS patterns)
- **Database-backed workflows** (e.g., PostgreSQL/RDS patterns)
- **AI integrations** (e.g., Bedrock patterns)
- **Infrastructure-as-code validation** (e.g., CloudFormation)
- Additional AWS services used by specialized teams

## Results and benefits

- **Faster developer feedback loops:** Accelerate dev and test cycles by eliminating cloud provisioning wait times when pushing code changes
- **Lower AWS test spend:** Move more CI and developer-in-loop testing off the cloud bill, reducing waste and avoiding unexpected cost spikes
- **Better standardization:** Establish an approved, repeatable approach that can scale across teams if validated
- **Earlier validation:** Catch issues sooner by validating service interactions and infrastructure definitions before changes reach cloud environments

# Request a Demo Session

If your teams need to validate cloud infrastructure patterns at high volume—without waiting on live AWS feedback loops—LocalStack can help you standardize fast, repeatable validation that scales with your delivery pipeline.

[Get in Touch](#)