

# Driving Consistent and Efficient Development Cycles Through Infrastructure Automation on AWS



*Learn how a global retail bank worked with Stelligent to take advantage of automation on AWS and drive new products to market rapidly for the benefit of its customers.*

## Don't Just Keep Up: Stay Ahead of the Curve Through Automation

Today, consumers can address most banking tasks in a streamlined and frictionless manner. With just the click of a button, consumers can, for example, transfer money between accounts, deposit checks, and freeze a lost or stolen debit card. To keep up with customer expectations and interaction preferences, retail banks need to quickly launch reliable environments in which to develop consistently, iterate quickly, and shorten the feedback loop for customers and developers.

A large global retail bank recognized how manual development and deployment processes caused inefficiencies for its developers. The company turned to the experts at Stelligent, who excel at helping customers achieve complete automation of their environments, to learn how it could automate its infrastructure and application deployment processes on Amazon Web Services (AWS).

## Manual Processes: Less Time Spent Innovating, and More Time Spent Frustrated

Having already deployed business-critical applications on AWS, the bank understood AWS was integral to the company's ability to digitally transform and drive customer success. A key motivation for using AWS was to enable developers to deploy testing and production environments rapidly. But the company's intentions didn't line up with its approach, as AWS resources were provisioned manually and inconsistently.

Developers were often kept waiting for weeks to retain a fully provisioned AWS environment ready for testing and development. Rather than submit a new environment request, teams often resorted to retaining environments from previous testing and development for net-new development work. Further, the company's development and production environments were both manually configured and were thus inconsistent in look and design. Developers would test on development environments and when they'd move code to production environments, they would encounter errors and environment inconsistencies such as different configurations and different packages being used. Developers lost time, cycles, and patience simply working to correct environment issues and couldn't devote as much effort to application testing, development, and deployment.

### AWS Services Used

- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS CloudFormation
- Amazon Simple Storage Service (Amazon S3)
- AWS Identity and Access Management (IAM)
- Amazon Virtual Private Cloud (VPC)

The result? Higher costs, longer development cycles, slower feedback loops, and lost opportunity to drive additional value for customers.

## How Stelligent Created Company-Wide Automation Using Chef on AWS

By embedding with the company's development team and working side-by-side, Stelligent became intimately familiar with the challenges the developers faced and made many recommendations for how the company could achieve a fully managed, fully automated environment.

Stelligent designed an entire networking infrastructure for the customer and codified development environments to launch with a click of a button. The team also created a massive enterprise-grade Chef infrastructure deployment platform capable of managing upwards of 10,000 nodes. The Chef platform manages the customer's entire AWS infrastructure, including development, testing, staging, and production environments; everything is managed by the Chef server. All the configuration on the servers are configured by Chef. Many AWS services are used in the automation process, including:

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) provides the company compute power and instances on which they run their environments.
- [AWS CloudFormation](#) manages and provisions AWS resources in a consistent and predictable fashion. A developer uses AWS CloudFormation to specify the size and type of Amazon EC2 instance they'd like to use and then Chef defines the EC2 instance internal configuration and executes scripts to install components onto the Amazon EC2 instance.
- [Amazon Simple Storage Service \(Amazon S3\)](#) provides large-scale file storage.
- [AWS Identity and Access Management \(IAM\)](#) securely controls user access to various AWS resources.
- [Amazon Virtual Private Cloud \(VPC\)](#) provisions a logically isolated section of AWS enabling the company to have complete control over its virtual networking environment. All VPCs, associated subnets, and route tables are defined in code that is checked in and committed to a central version control repository. Any changes to be made have to go through code, committed to the repository, and then executed through an automated process.

Ruby is used for general purpose scripting and orchestration-layer implementation, and Hudson, an open source continuous integration tool written in Java, runs self-service components implemented by Stelligent. Hudson provides users a single pain of glass to execute automated tasks, such as pushing new Chef cookbook changes to Chef Server, launching new development environments, and running Continuous Integration builds for the various Chef cookbooks and libraries.

## Delighting Internal Teams and Customers Through Rapid Deployment Cycles

Thanks to Stelligent's automation of its entire environment, the bank began to fully realize the benefits of using AWS and drove significant improvements for the business, including:

- Reducing the lead time to launch an environment from 720 hours to two hours
- Significantly shortening coding, testing, developing, and deployment cycles and bring products to market faster
- Eliminating manual developer tasks to allow more focus on developing value-added features to drive customer satisfaction and engagement
- Improving infrastructure governance and limiting risk by designing automation while adhering to AWS best practices and prohibiting developers from making manual changes to network components
- Providing instant ability to scale up environments to meet market demands and scale down environments to eliminate idle resources and reduce costs
- Designing for high availability and eliminating any single point of failure within the architecture
- Shortening customer feedback loops to be able to rapidly improve the company's products and services
- Improving end customers' experiences by ensuring applications are thoroughly tested in the right environments prior to deployment and eliminating application errors

The bank can now confidently focus on being at the forefront of digital innovation on behalf of customers rather than struggling to keep up with industry trends.



#### About Stelligent

Stelligent, a professional services and consulting firm with deep expertise in DevOps automation services on Amazon Web Services (AWS), enables security-conscious enterprises to focus on developing software users love by leveraging automation on AWS. Our goal is to work closely with customers to develop fundamentally secure infrastructure automation code, deployment pipelines, and feedback mechanisms for faster, more consistent software and infrastructure deployments. By embedding with our customers engineering teams, we empower customers through education and knowledge transfer of our expertise while developing the automation to make them self-sufficient on AWS. As a Premier AWS Consulting Partner, AWS Public Sector Partner, and AWS DevOps and Financial Services Competency holder, we use our demonstrated expertise to help customers benefit from continuous AWS innovation.