Application Transformation onto AWS

Executive Summary

Citigroup Inc. (Citi) sought to migrate their legacy application from on-prem to AWS and then leverage end-to-end automation to deploy the application into end-customers' AWS environments.

Key Challenge/Problem Statement

The Client had a legacy application that played a critical role in their consumer business. Due to the inefficient infrastructure and the nature of running an application on-premise, the application had issues around scalability, setup, and making changes in a reasonable period of time. These issues were causing impact to the business.

- Inability of the application to scale properly during peak periods of load due to lack of resource accessibility.
- Delayed deployment speeds and alignment with customer requests.

Furthermore, there were technical constraints that added additional complexity. The application was a 3-tier application and required an entirely isolated environment to be provisioned for each end-customer. This additional complexity caused problems with:

- Infrastructure setup - Manual processes would increase the time and difficulty of provisioning a new application environment.
- Deployment – Difficulty in managing deployments for each target environment.

The Client sought to deploy the application onto AWS and then leverage end-to-end automation to deploy the application into end-customers. Additionally, the AWS environments to take advantage of AWS cost efficiencies and deployment automation.

Proposed Solution and Architecture

The proposed solution for migrating the application was broken down into review and implementation:

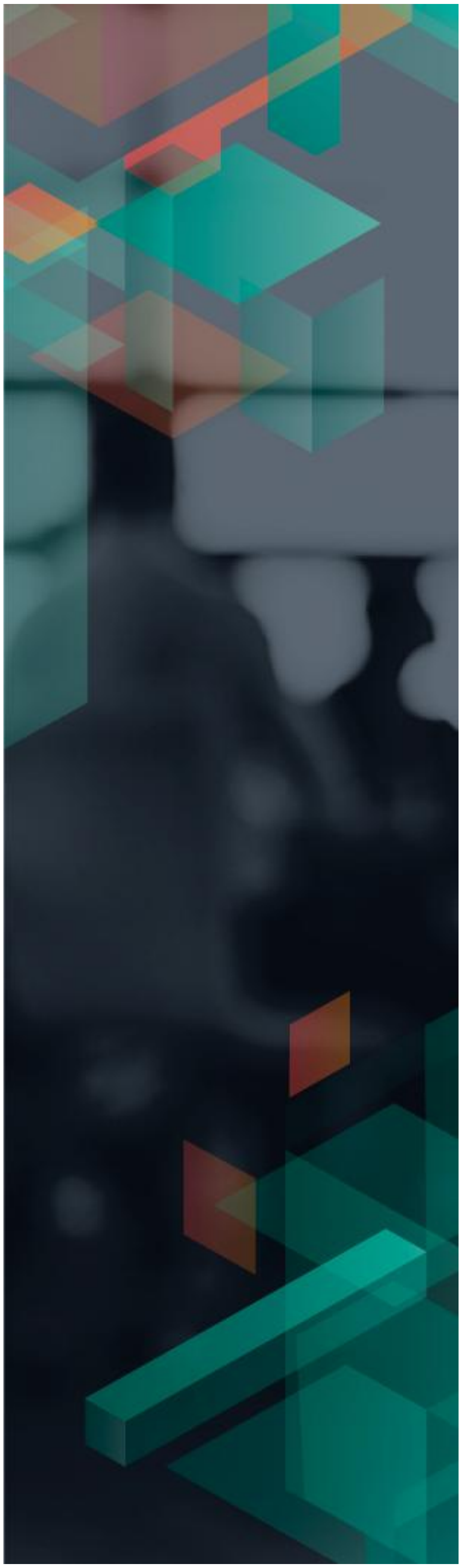
- Application Architecture – A review of current application and infrastructure architecture.
- Application Cost Governance – A review of projected AWS spend.
- Application Migration and Automation – Application migration, pipelines, infrastructure, and Security & Compliance automation.

Upon completion of the Review and Recommendations phase, we implemented solutions that aligned to the recommendations.

About the Client

Citigroup Inc. (Citi) is an American multinational investment bank and financial services corporation. Citi offers checking and savings accounts, for SMB and commercial banking along with personal wealth management. Citi has over 200 million customer accounts and does business in more than 160 countries.

Citigroup is the third largest banking institution and is one of the Big Four banking institutions of the United States. It is considered a systemically important bank by the Financial Stability Board. It is one of the nine global investment banks in the Bulge Bracket.



Infrastructure Automation – Fully scripted the 3-tier application, and its associated resources using CloudFormation.

Pipelines – Fully scripted and automated pipelines were built to quickly deploy the 3-tier application for each of their customers when needed. Using a parameterized CloudFormation based approach, CodePipeline pipelines were defined that would go through a series of stages which consisted of testing, validating, and deploying the application to production.

Security & Compliance Pipelines – SecOps automation pipelines were implemented that would deploy security-based AWS services to help ensure compliance, detect/prevent security violations, and alert personnel on security events when they occurred.

Security & Compliance Detective Controls – AWS Custom Config Rules were deployed that would monitor for account resource violations including non-compliant SG/NACLs, instance/database sizes/types, and missing resource tags.

Security & Compliance Reactive Controls – When a violation was identified by one of the custom Config rules, the associated Lambda function would trigger and perform auto-remediation against the violating resource and send out a notification to the appropriate personnel alerting them of the event.

Cost allocation – Defined tags and categories were automatically configured for the resources that get provisioned in each application environment.

- "Application_Workload" - used for defining the differentiating the type of workload or application the resource was allocated to.
- "Business_Unit" - used for identifying the business unit within the organization that the resource was allocated to.
- "Client_ID" - used for displaying the unique identification for each of the Client's customer specific application environment resources.

Results

- 1000%+ increase in speed of application environment setup
- 25+ custom preventative & detective controls
- 10+ custom reactive controls

Summary

By engaging with Vertical Relevance, the Client has now gone from multi-week datacenter-based implementation to sub-hour AWS environment deployments, creating substantial efficiencies and meeting end-customer requirements.

About Vertical Relevance

Vertical Relevance is a Financial Services focused (Wealth Management, Asset Management, Banking, Insurance) consulting firm helping with the design & delivery of effective transformation programs across people, process, & systems. With 10+ years of AWS & 20+ years of Financial Services experience, we understand the business needs & build solutions to meet sales, marketing, & compliance goals.