# Modernizing DevOps for a Bank's Digital Transformation

By Arish Ballana

## Problem Statement

The main issues at hand are ensuring rapid deployment while at the same time remaining compatible with legacy systems, adhering to stringent financial industry regulations through strong quality assurance procedures that will result in the guarantee of high system reliability with minimal disruption, protecting sensitive customer data from cybersecurity risks that are constantly evolving, and efficiently scaling infrastructure to meet increasing demand.

## Proposed Solution

To tackle these obstacles, my suggested method is to incorporate **Continuous Integration and Continuous Deployment (CI/CD)** pipelines like Jenkins/MS Azure Pipelines, which include in addition, of course, the aspect of test automation. Together with the use of feature flags in order to ensure phased releases are supported without accidents, the procedure becomes more efficient. While executing **legacy system integration**, APIs and microservices could be employed to communicate with existing systems that will not create disruptions and performance testing of legacy services must be carried out using testing tools like JMeter in an automated way.

To make sure that **regulatory compliance** is met, the company should run security checks; for that, I suggest integrating tools like Cisco VM, IBM's Blackduck Scan, and Rapid 7's Insight VM. For proactive monitoring, you can use Site Reliability Engineering (SRE) methodologies; otherwise, **Application Performance Monitoring** such as Dynatrace and Appdynamics can be implemented; this can help us find the issue in the system even before customers realize it. The presence of load balancers might be the clue to **scalable infrastructure** that will utilize Kubernetes for the auto-scaling solutions, and container orchestration to dynamically distribute resources. This will quickly pinpoint the mistake in some of the services deployed in some regions and, therefore, allow us to revert the changes in a short period of time.

The options that are being envisioned are expected to be successful in terms of achieving the **objectives** they have, among which are the following: a decrease of 40% when it comes to the time it takes for a new feature to get to the market, a system that is up 99.9% of the time, and not a single compliance offence made in the audits. The **deployment plan** is an effectively tested format where every deployment will have to be tested on the dev and stage stack to finally reach the prod stack. The Dev stack will serve as a representation of feature testing and integration, while the stage stack will comprise load and performance testing. Subsequently, the feature will be gradually deployed to production regions, beginning with the region with the least amount of load.

The implementation roadmap comprises the steps of the following three stages: **Phase 1**, Identify and map the existing ecosystem by the pain points. **Phase 2**, Creating and running CI/CD pipelines with security and compliance embedded in them for piloting purposes. **Phase 3**, Deploying the plan step-by-step from dev through prod, is included, also using feature flag as a backup plan in case of specific region-wise failure where revert can be performed if necessary.