

Case Study

Industry: Healthcare

From Legacy Bottlenecks to Scalable DevOps



Quick Stats

Verticals Served

Cybersecurity

AWS

DevOps

Technology Stack

AWS CDK Typescript, AWS CodeBuild, YAML, Docker

Team Size

8

Fixed or Ongoing Project

Ongoing Project

Project Build

Started from Scratch

Background

In the past, development teams at BCI relied on Jenkins for CI/CD. However, as time went on, Jenkins required excessive permissions and became unmanageable. Developers installed many plugins that were never maintained or validated. There was no oversight of the master Jenkins instance to ensure it remained current. Sometimes, a developer would create new Jenkins pipelines by pushing branches.

Over time, the number of excess pipelines grew to over 300. All had direct admin access across each of the current 45 AWS accounts.

Key Results

300+ Pipelines Migrated

300+ Jenkins pipelines were migrated across 45 AWS accounts, lowering security risks and complexity.

\$1M Investment in Infrastructure

A \$1M ground-up project modernized BCI's infrastructure and apps, enabling lasting digital transformation.

Streamlined DevOps Workflow

GitLab automation replaced EC2 runners, reducing DevOps overhead and simplifying daily workflows.

The Challenge

The Jenkins pipeline setup within BCI created significant blockers for various teams. Many underlying packages and dependencies were not updated. Some were no longer supported by the current Jenkins versions. This caused operational inefficiencies and maintenance challenges, and harmed security benchmarks and scores.

BCI's adherence to NIST and other security guidelines was at critical risk.

The Process

The project consists of several key milestones:

1. Evaluate and categorize existing Jenkins pipelines by complexity and priority. Also, identify their dependencies and design templates for the AWS migration.

We developed a POC around the use of AWS CodeBuild with GitLab's native CI/CD tools. The POC deploys new CI/CD pipelines using GitLab. This eliminates runner dependencies and maintenance on EC2.

1. Pilot a subset migration of low-complexity pipelines to AWS CodePipeline. This includes improving the migration process based on insights and feedback.
2. Focus on the full migration of the remaining pipelines. We aim to move 4-6 pipelines each day. We also keep an eye on important performance metrics and validate them. The metrics include build times and reliability.
3. The last milestone will complete the migration process by testing pipeline triggers - verifying artifact storage and archiving Jenkins pipelines per Blue Cross of Idaho policies.

This migration process includes:

- Evaluating the current Jenkins pipelines.
- Mapping and managing pipeline dependencies.
- Implementing security and compliance measures.

Collaborating with project teams is key. It helps ensure a smooth migration and redeployment of current applications and infrastructure. After the migration, we validate the performance, reliability, and functionality of all pipelines. This initiative aims to enhance scalability, streamline, and standardize pipeline management within BCI.

Results & Impact

✓ Strengthened Security & Compliance

We retired old Jenkins instances and switched to GitLab's CI/CD with AWS CodeBuild. This new system meets NIST security standards. It also improves vulnerability detection with built-in SAST and DAST tools.

✓ From Legacy to Modern DevOps

Moving from legacy Jenkins and GOCD to a cloud-native solution improved build reliability. It also cut down technical debt and offered a unified way to deploy.



Get in Touch:

(208) 258-2424
info@intimetec.com