BAILEY FITCHETT
Senior DevOps / Platform Engineer

Six applications. Modernized for Azure.

PROFESSIONAL CASE STUDY / AZURE APPLICATION MODERNIZATION

6	18-21	100%
applications	supporting services	target release dates

The environment and my ownership

A hybrid estate combined on-premises infrastructure, cloud resources and legacy hosting. The IIS-hosted ASP.NET applications used web, application-services and SQL database tiers. I owned architecture, pipelines, monitoring, disaster recovery, coordination, application migration and audit responsibilities.

Key decisions

Treat the migration as one system. Coordinate application, platform, data, delivery and operational work instead of separating them into disconnected handoffs.

Diagnose before adding capacity. Distinguish application-level causes from hosting symptoms, then apply focused improvements rather than using compute as the automatic answer.

Include operations in the architecture. Build monitoring, audit and RTO/RPO-aligned disaster recovery into migration readiness.

Modernization and delivery

Targeted application work included using **StringBuilder** where repeated string operations created avoidable allocations, tuning cache behavior and implementing Redis. The wider program used repeatable CI/CD, infrastructure automation and application monitoring across approximately 18-21 services and 4-6 environments.

Outcome, scope and retrospective

All six application migrations met their target release dates. Performance and stability improved, but no benchmark percentage is claimed. With more modernization time, I would extract suitable capabilities into Azure Functions and deliberately remove or upgrade NuGet dependencies with the surrounding code and tests.

Read the full story and sources

Full case study: baileyfitchett.com/azure-modernization/