

CVS Cloud Migration: Transitioning from Pivotal Cloud Foundry (PCF) to Microsoft Azure

Business Problem:

As a small business unit within CVS, the continued use of the PCF (Pivotal Cloud Foundry) to host an array of internal and customer-facing software applications would cost an additional \$5 million per annum. To enable cost savings and operational efficiencies, a decision was made to migrate from the private cloud to a more cost-effective solution: Microsoft Azure public cloud service.

Solution:

Creospan adopted a strategic approach that was laid on the foundation of holistic migration, proactive risk mitigation, adherence to security & compliance standards, infrastructure evolution, and collaboration.

- **Deployment Strategy:** Overall, 8 core systems were migrated, each with 7-8 core software services. Some of these systems were internal and some were customers facing. Utilized Azure Kubernetes Service (AKS) to orchestrate containerized applications, ensuring scalability, consistency, and easier management.
- **Technology Stack Overhaul:** Migrated to a more modern stack: Java 7 to 17, Spring Boot 3.x, and Angular versions ranging from 10 to 14, which met both security and operational requirements. This necessitated code changes and dependency adjustments.
- **Integration & Configuration:** Adopted Azure-native tools, including Key Vaults for secure credential management and Azure Service Bus for middleware communication. This ensured secure and seamless connectivity with third-party services like Oracle and Salesforce.
- **Infrastructure Evolution:** Transitioned the image management strategy from the Nexus repository in PCF to RedHat's Quay in AKS. This was complemented by comprehensive networking and security configurations, ensuring robust application deployments.
- **Monitoring & Insights:** Implemented Azure's Application Insights for granular application monitoring, alongside enterprise log management systems such as Splunk and Dynatrace for a complete view of the application landscape.
- **Compliance & Decommissioning:** Strategically planned the decommissioning of older systems while ensuring data retention met regulatory requirements.

Team & Timeline:

- The solution was built on a foundation of collaboration, leveraging expertise across development, networking, security, and compliance domains to ensure a smooth and efficient migration.

- A collective effort by a team of senior consultants from Creospan, alongside the CVS product, security, and networking teams, successfully migrated nine core systems, each with 7-8 services. These applications spanned both external and internal services.
- Each core system took around 4-5 months to complete the migration process.

Outcomes:

- **Cost-Efficiency:** The migration to Microsoft Azure culminated in substantial cost savings, significantly reducing the projected expenditure associated with the continued use of PCF. This financial prudence empowered CVS to allocate resources more effectively across other pivotal business avenues.
- **Robust & Scalable System:** With the integration of Azure Kubernetes Service (AKS), the migrated systems demonstrated enhanced robustness. They efficiently adapted to varying operational loads, ensuring consistent performance irrespective of demand surges, and paved the way for potential future expansions.
- **Security & Compliance:** The shift to Azure, complemented by the adoption of modern technology stacks and Azure-native tools, fortified the systems' security. Additionally, strategic decommissioning plans and data retention strategies ensured rigorous adherence to regulatory compliance standards.

Challenges:

- **Talent Pool Limitations:** Collaborative effort across various teams - including but not limited to Product, Networking & DNS, Logging, Security - was essential, emphasizing the need for seamless communication and coordination.
- **Quirks in Middleware Migration:** The team's familiarity with the RabbitMQ message queue service in PCF posed an initial challenge. Azure, in its default offerings, lacked a pre-configured RabbitMQ environment. The alternative, integrating RabbitMQ into Azure, would have necessitated custom configurations, securing additional approvals for security and compliance, and potential timeline delays. Given the critical nature and time sensitivity of the migration, the team prudently opted for Azure Service Bus, streamlining the middleware transition and ensuring project continuity.