Renovating Systems of Differentiation and Innovation Without Customer Disruption

Point and Click Launch of Kubernetes Clusters on AWS Speeds Time to Market

An S&P 500 research and advisory company aimed to maintain its market leadership position and gain the advantages of a highly available, high uptime service running in Amazon Web Services. With over 400 applications to be re-platformed or re-factored, 200+ apps were identified as systems of Innovation and Differentiation. From this subset, 40 real-time transactional apps, part of the company's customer portal and mostly in Java/Tomcat, were identified as critical for customer service and innovation.

Being able to leverage services like Amazon CloudFront would reduce the latency of their application, known to be slow in some parts of the world, and allow them to improve geodiversity. And, while the company pushes releases every two weeks, their aim was to increase the speed and frequency of software releases to support a model of continuous improvement toward delivering high-value solutions for customers.

Renovating Systems of Differentiation and Innovation Without Customer Disruption

The company's customer portal provides access to content and delivers customer experience features including content curation, search and recommendations engine and other elements. The portal drives the company's primary source of revenue and is a strategic investment and source of innovation and differentiation. Instead of a lift and shift strategy that could migrate problems and wouldn't necessarily gain the company the benefits of infrastructure as code, the organization decided they wanted it to be agile in nature and adopt container-based micro-services.

While the portal includes elements that would be typical of most deployments, considerable additional complexity was created by the need to retrofit an old application that required sticky sessions and shared storage, and by a need to modernize both the application and the infrastructure at the same time. This innovative enterprise had also made the decision to prioritize open source software over enterprise versions. An enterprise-grade DevOps platform was already in place, precluding work that might normally be required in advance of starting roof-of-concept tests.

The POCs needed to answer questions including:

- How to get secrets from Hashicorp Vault into Kubernetes containers, a process that would require a Java/TomCat plug-in
- How to address shared storage needs
- How to work with apps with in-memory session state, exploring how ingress controllers could be applied and configured inside of Kubernetes

- How to address shared storage communication challenges using IAM roles in AWS to provide authorization and authentication to the services to talk to Amazon EFS
- How to ensure zero downtime and the ability to conduct blue/green deployments as well as address ingress because a requirement was zero downtime during migration
- How to manage role-based access controls (RBAC) using open source Kubernetes
- How operational data-gathering requirements could be met by using deploying applications like Splunk and DataDog
- How to incrementally move an Oracle database into AWS using strategies such as moving database tables relevant to each service one by one and using an Amazon RDS Oracle database

Point and Click Kubernetes Factory to Speed Time to Market

Building a point and click factory that developers could use to launch Kubernetes clusters on-demand started with a Kubernetes configuration pipeline. This pipeline automatically configures a Kubernetes cluster by installing plug-ins and configuring things like ingress controllers and DNS settings. A Git repo consisting of YAML files deploys the YAML files as necessary.

The actual deployment of services is achieved with container pipelines.

- First, an artifact-build builds the Java artifacts. It compiles the Java code and runs it through security tests using SonarQube which then creates the binaries and the JAR and WAR files which are pushed into a JFrog artifactory repository.

- Then, the pipeline checks the binaries and uses a Docker file to create a Docker Container Image. These container images are again pushed back into the JFrog artifactory.

- Finally, the actual deployment is accomplished by using the Kubernetes API to start a new service with Kubernetes.

Accelerating Testing For Better Outcomes

Developers can now efficiently test their ideas to address POC questions in production-ready, highly available, multi-AZ Kubernetes clusters. As the POCs started to finish, confidence on the teams went up and the first half dozen web services running inside the cluster were able to be demonstrated. The first demos started to appear about eight weeks after the project kicked off. After the initial challenges were worked out and a final workflow established, the pace has rapidly accelerated and two-three new services each week can be demonstrated on the new system.

By using Flux7's factory model built on a platform of regularly refreshed best practices, the organization can now accelerate the speed of innovation by removing barriers to progress such as third party management or cumbersome traditional processes. Extreme automation available in the platform means that developers or resources with domain-specific knowledge can be assured they are using best practices by creating the clusters from operations and security approved templates. And, it means that IT ops can be assured compliance, security and other best practices are being met without slowing the pace of innovation.

About Flux7

Flux7, an NTT DATA Company, is an IT services firm that helps enterprises reduce the complexities of a new or evolving cloud automation strategy. Agile and DevOps-native, Flux7's robust services portfolio prioritizes a fast path to ROI that meets the immediate needs of technical and innovation teams focused on transformation while forging a secure and stable pathway for security and operational excellence. Learn how Flux7 helps businesses bring solutions to market faster at https://www.flux7.com