[simple] Technology Solutions

CASE STUDY

Enhancing Enterprise DevOps Adoption

BACKGROUND

A large U.S. Government Agency had to migrate over 80 applications from their data center into Amazon Web Services (AWS) and Microsoft Azure (Azure) Cloud Hosting Environments (CHE). The customer had a very tight deadline to migrate all of its applications before the hosting service shut down its legacy data center. There were a number of expectations for this migration effort: a) development teams should be empowered to provision and manage their own cloud infrastructure, b) teams should use and embrace modern automation and DevOps practices, c) teams should utilize open source tools, and d) infrastructure should be managed in a way that was transparent to the Office of the CIO thus eliminating shadow IT office requirements.

ANALYSIS

Simple Technology Solutions (STS) identified a number of traditional infrastructure engineering tasks that could be managed by development teams via automation. Using tools such as Chef, Terraform, and Jenkins, STS engineers provided these solutions to application teams. However, many application teams continued relied on legacy infrastructure engineering staff to complete these tasks in lieu of automation. After discussing legacy dependency and barriers to cloud migration with government IT project managers and team leads, STS identified a large skill and cloud technology knowledge gap for many team members.

SOLUTION

To address the identified cloud technology skills gap, STS team designed a template for a series of three 'onboarding' meetings. These meetings were designed to provide a personalized, user centric introduction to the cloud hosting environment. The training was customized to the cloud service provider and the application's specific technology stack.

First meeting: Focused on ensuring all team members had access to all necessary cloud automation and devops tools such as a) AWS/Azure console access, along with the associated Software Development Kit (SDK), b) Terraform installed in the development environment, c) Chef Server access, and Chef's SDK, d) Enterprise Github access, and e) access to the Jenkins CI/CD server.
By assembling the engineering teams and government approvers in the same location, approval issues and troubleshooting times were reduced from weeks of emails to a single morning meeting.

Second meeting: Provided initial code for Terraform and Chef so the team could create their own infrastructure to be automatically hardened, secure, and compliant with agency requirements. At the end of this meeting, the team had a repository in Github Enterprise so they could continue developing Infrastructure as Code (IaC).

Third and final meeting: Discussed cloud native application design with the Cloud Architect. They identified a goal to develop Jenkins pipelines for automated deployments and explored opportunities to use cloud technology to reduce operational requirements and focus on development.

BENEFIT

The STS team meetings led to an open discussion of pain points and a steep increase in cloud automation adoption. STS engineers were able to emphasize how these changes would empower developers, shorten wait time for infrastructure provisioning, have higher capability to scale, and reduce operational overhead. Putting development teams in the driver's seat encouraged the teams embrace the cloud rather than focus on the agency mandate as a difficult challenge. It was also a valuable opportunity for the cloud team to hear feedback from cloud hosting users and effectively address technical priorities.

1775 I St. NW #1150
Washington, DC 20006

Phone: 202-446-2134

info@keepitsts.com
www.keepitsts.com