AWS DevOps Competency Case Study IV

Containerized DevOps for Content Management Systems

Introduction to Customer:

The General Services Administration (GSA), an independent agency of the United States government, was established to help manage and support the basic functioning of federal agencies. GSA required CTAC to design and build a new cloud based content management platform in which to host several of its public and private applications. Sites and applications to be hosted on the CTAC Platform included flagship sites including GSA.gov and USA.gov.

Overview of Challenges:

The challenge in building a new platform against legacy applications is that the underlying tools, services, and technologies being leveraged by those applications are not consistent or compatible. The sites and applications were hosted on on-premise legacy servers. CTAC needed to provide a standard and automated way for independent, dispersed development teams to continue developing and maintaining the legacy applications on the new CTAC built and managed AWS platform. A solution to this problem was the use of Docker, a containerization technology. A Docker container image is a standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. In order to meet unique program security requirements, CTAC designed an automated and unified DevOps build pipeline with Jenkins leveraging Docker in local environments, and Packer built Amazon Machine Images (AMIs) for production runtimes. Jenkins was used to facilitate automated, repeatable, and predictable deploys for GSA regardless of individual application needs.

Details of Solution:

CTAC worked closely with GSA stakeholders from the technical, business and security teams and designed a FedRAMP/NIST 800-53 compliant AWS platform. CTAC's solution provided the dispersed development teams with a containerized development life-cycle, but a dockerless runtime. CTAC leveraged both AWS Services and third party tools in order to accomplish this.

CTAC built the ubuntu base as the official base image provided to development teams and hardened it according to the CIS Docker Benchmarks. Each deployable service is required to be derived from the CTAC Base Image. Any missing features from a base image can be added by the developer, or if the need is common enough, baked into the base image by request. printBase image version tags are updated as security patches come through the individual application stacks, and it is the responsibility of the developer to choose when to move their application to the newer version, the default version provided is the latest.

A Jenkins-based continuous integrations process pulls code for each customer application, to build components and system versions and deploy them into the appropriate tiered environment in the appropriately secured VPC for customer application. Each customer application instance utilizes the GSA GitHub SSH private key to gain read access to it's appropriate repository so that

the Jenkins based continuous integration process and deploy the latest built code for that specific application.

Within the build pipeline, CTAC leverages Packer to create the non docker EC2 Amazon Machine Images (AMIs). Packer is an open source tool for creating identical machine images for multiple platforms from a single source configuration. Packer is lightweight, runs on every major operating system, and is highly performant, creating machine images for multiple platforms in parallel. The CTAC built platform uses the Packer Amazon EC2 Builders, Puppet (Configuration Management Tool), file and shell provisioners to create base application AMIs for use in the build/deployment process. AMIs are built within the Platform's Core Services VPC and stored private to the AWS Account.

Summary:
CTAC designed and built a modern DevOps focused platform built on AWS and migrated two of GSA's flagship sites from legacy on-premise servers to it. Understanding the risk adverse nature of the government, CTAC was able to provide a containerized development life-cycle supporting multiple development teams while creating an automated CI/CD build pipeline and a security compliant dockerless runtime environment.