FOX

» CUSTOMER CASE STUDY

Automated Infrastructure as Code Deployments for Splunk

Solving the challenges of deploying an non-cloud native application in the cloud using IaC

NEBULAWORKS

Executive Summary

Nebulaworks Engineering was tasked with developing automated cloud Infrastructure as Code (IaC) deployments of a complex distributed system called Splunk, a Security Information and Event Management (SIEM) tool. 20th Century Fox has infrastructure on-premise and in the cloud, and needed the features of this SIEM tool to collect data from network devices, servers, and much more. This Splunk deployment would span 12 business units and roll up into a single pane of glass for federated security monitoring of these business units. Manual deployments at this scale were not tenable, but they were initially thought to be impossible to build.

Why Nebulaworks?

This deployment was not a simple one-off build. It required several iterations and continued evolution of the Fox team's codebase to achieve the desired state. Nebulaworks specializes in leveraging software engineering techniques in the cloud, leveraging standard development processes, and cloud subject matter experts. Prioritizing standardization for the IaC development allowed the teams to work together efficiently and provided the right foot forward for the Fox team to own the deployments as a product at the end of the engagement.

The Challenge

The architecture proposed by Splunk to Fox had never been attempted in the cloud. With a dozen Splunk deployments and a federated single pane of glass to view all the deployments, this was a complex architecture with many components. The vendor recommended that the large majority of this infrastructure was going to require to be deployed manually. Following this recommendation would result in long lead times to deploy environments, and it would not provide the ability to redeploy the architecture quickly and consistently. This would also result in high ongoing costs and human capital for continuously maintaining many services and infrastructure. Instead of solving the problem with large teams manually deploying the infrastructure, Fox decided to reach out to Nebulaworks to see if it would be possible to automate the majority of resources and integrations needed for not only the initial deployment but continuously iterating on the distributed cloud environment so that the team can service the ongoing needs to an organization of the size of Fox. Nebulaworks was up to the challenge.

The Solution

Team collaboration was critical to the project's success with engineers from Fox, Nebulaworks, and Splunk, working together to create this automated deployment. Initially, there were several highly collaborative working sessions between the three teams, planning out how to take a once manual process, and build high levels of automation to avoid the lead time of deploying and configuring by hand. The team focused on reviewing and documenting the Splunk deployment's management activities and evaluating them against cloud-native application deployment methodologies. Since this tool was ordinarily not deployed with IaC and was traditionally managed by Splunk SMEs, it was important to understand all the considerations of a fully automated deployment. Working closely with engineers provided by Splunk, Nebulaworks documented and understood the various components of a production-grade Splunk deployment and began leveraging working configuration files that would be injected during EC2 instance boot time. The team was leveraging Chef to provide last mile configuration management for all EC2 instances in the Splunk deployment. Using user data and cloud-init, the engineering team used bootstrap scripts to run Chef cookbooks, such as provisioning and configuration. Additionally, given that Splunk is a distributed system, race conditions needed to be solved when automating the configuration and bootstrapping of Splunk components. All of the components necessary for a production deployment could not stand on their own.

They had a complex Directed Acyclic Graph (DAG) that defined dependencies and precedence for component provisioning. The interesting use of Consul made its way into this deployment to solve this problem. Using a distributed lock feature of Consul, the engineering team could account for the order of precedence in provisioning specific Splunk components by using Consul distributed locks during EC2 bootstrapping. In addition to the tooling that helped provision resources in the cloud, One key practice that made this project successful was establishing a development workflow that would continuously support this deployment throughout its entire life cycle. This process defined a release process that is stable, reliable, and, most importantly, minimized production downtime. Equipped with a release engineering process, the Fox team maintained velocity and iterated on the development of Splunk environments without affecting production.

Outcome

By using Nebulaworks to leverage automation techniques in the cloud to deploy the dozen Splunk environments, Fox benefited in several ways. They did not need to hire dozens of Splunk SMEs at a premium. All infrastructure was defined as code that allowed Fox to leverage its internal Cloud and IaC Engineers to manage the business unit environments' deployment and management. By using IaC tools and techniques, the engineering teams achieved consistent and predictable results, even more so than a Splunk SME manually configuring the machines. The development could happen without impacting production thanks to the established software and release engineering processes pioneered by Nebulaworks. With these processes in place, the teams created and destroyed business unit environments in minutes vs. days. With the right approach and engineering talent, this engagement proved that it was possible to automate one of the largest deployments of Splunk on AWS.