augustschell

CASE STUDY

Defense Agency Gains Push-Button Process for Repetitive Splunk Tasks, Maximizes IT Effectiveness

Executive Summary

A defense agency of the U.S. government faced a significant challenge that required an innovative solution. With a Splunk environment consisting of multiple search head clusters and index clusters, over 10,000 endpoints, and a data ingest rate amounting to terabytes and growing quickly, the customer faced the need for the automation of Splunk administration and maintenance. Hundreds of custom apps were being created from the command line, where configuration files were maintained, as well. To simplify the process, they'd need a friendly interface to work with and a push-button process for the routine administration of Splunk.

Challenges

- A large scale Splunk infrastructure growing at an extremely fast pace
- Many Splunk commands were being performed manually, introducing significant risk and inconsistency into the environment

Solutions

ANSIBLE

GitHub

Jenkins

splunk>

Business Impacts

- ✓ Shortened the maintenance windows from hours to minutes, allowing for more frequent maintenance windows throughout the week.
- ✓ By automating complex manual tasks, the customer greatly reduced the risk of introducing human error into their environment.
- ✓ Internal staff with less expertise were empowered to use Splunk easily as a result of automation.
- ✓ ROI was increased, thanks to the reallocation of high level engineering to more impactful projects, while lower level admins could work on Splunk tasks.
- ✓ Daily regular Splunk routines went from complex and risky to the simple click of a button.

Defense Agency Hesitant at the Command Line, DevOps Rises to the Occasion

A U.S. defense agency was running a large Splunk environment with three separate search head clusters. Further, their Splunk license was sized for eight terabytes of daily data ingests, which is a significantly large amount of data to contend with on a daily basis.

While they didn't struggle to implement Splunk initially, problems began to arise as their environment grew more complex, and they lacked the internal expertise to carry out the proper system maintenance. "The biggest challenge for this customer was the fact that they were without a push-button environment," Eric Nicholson, Certified Splunk Consultant at August Schell explained.

Deployed apps and the management of configuration files within Splunk also became a growing problem. "Some of their Splunk users had to log into the operating system command line to perform Splunk commands by hand, and it wasn't comfortable for them," he went on.

Nicholson determined that the best way to alleviate their challenges was to automate the management of Splunk, which is where DevOps was key. August Schell's Splunk engineering experts teamed up with the agency personnel to automate manual Splunk processes and create a push-button workflow.

Dreaming Up an Innovative, All-Encompassing DevOps Workflow

Given the nature and amount of processes the customer needed to automate, a multilayered, inventive solution would be necessary. No run-of-the-mill automation tool would create the environment their challenges called for.

"I was determined to find a way to create an environment where you can log in, click a button, and routine tasks are run on each different search head separately: the general search cluster, the security investigation server, and the Splunk deployment server for managing endpoints," Nicholson recalled.

The customer essentially needed to be able to click a button that would send out application configurations to each different search head cluster automatically and run the command in the background, whether to initiate a restart or a deployment of apps to a cluster.

"Knowing what they needed to accomplish, I selected three separate tools, including Ansible, Jenkins, and GitHub to get the job done," Nicholson explained.

Building an Unmatched Solution for Splunk Automation

GitHub was first on the list. Nicholson worked tirelessly to ensure repositories were in place for all five search centers, as well as the deployment center. He also included the Splunk apps related to each, allowing the customer's admins to log into GitHub within a user interface and easily edit configuration files to add new apps or add-ons, rather than editing files directly.

"They can even submit their edits for peer review before it's committed to the GitHub production repository," Nicholson said.

Next up was Jenkins and Ansible, brought together to generate the "push-button" feature the customer was looking for. Jenkins is an open source tool used to automate software development builds, and run Ansible Playbooks, a popular scripting language used to create and run automation playbooks. You can also create a schedule to run projects in Jenkins, which then trigger a specific an Ansible playbook to do the actual work.

"I designed an Ansible Playbook for each search and deployment center to pull the apps in the background, from the appropriate GitHub repository, and sync them to the correct deployment center. Once it does that, it sets all executable permissions and runs a particular command to deploy the apps. This allowed the customer to deploy bundles into core Splunk production servers," Nicholson explained.

Customers would be set up to make edits in GitHub. During change windows, they could easily log into Jenkins, click a button, and watch the percentage complete until it reaches the end.

"Creating these workflows and the controls to execute them has eliminated the need for command line access for Splunk administration, and it provides the ability for peer review before committing to production," he said.

State-of-the-Art Solution Brings Relief to Splunk Users, Reduces Risk of Human Error

Within a month and a half, the solution was working in production, with all manual processes automated for DevOps through Splunk.

"Originally, I went into this engagement thinking I'd be working on app configuration and implementing Splunk Enterprise Security. But users being uncomfortable with modifying files and working with Splunk from the command line was a major pain point, and to be honest, it was introducing risk into their agency," Nicholson said. "The gears changed, and while I'm not usually brought into development, I was up to the task of making Splunk easy for them."

Post-implementation, using an alternative solution to taking the manual processes out of Splunk management made a great difference to the customer. Internal staff were relieved to be released from tasks they didn't feel confident in executing, while the agency gained peace of mind and stronger security.