



Department of Defense Joint Communications Unit (JCU) Leverages Nautobot® for Network Source of Truth and Automation Needs

The Customer: Department of Defense Joint Communications Unit (JCU)

As the premier technical unit within the United States Special Operations Command, the Joint Communications Unit (JCU) plays a crucial role in standardizing and ensuring interoperability of communication procedures and equipment for the Joint Special Operations Command.

The JCU, activated in 1980, has earned the reputation of "DoD's Finest Communicators" due to its commitment to excellence.

The Challenge: Network Configuration Compliance and Standardization in a Highly Dynamic Environment

Imagine picking up your entire network, flying it to a new location, setting it up for new users, integrating new equipment, and operating it to meet new mission objectives. This is the role the JCU faces every day.

JCU faced two major challenges that necessitated a shift toward network automation.

First, there was a lack of defined network configuration standards. The prevailing method of configuring devices relied on a relentless drive to get the job done. Unfortunately, this led to excessive configuration drift and hours-long troubleshooting processes.

The second challenge emerged with the introduction of the Modular Family of Systems (MFS), a new suite of expeditionary communication packages. The modular nature of MFS, the requirement to support any possible scenario, and manual configuration processes led to excessive documentation and time-consuming error-prone executions. As a result, the first set of equipment took over three weeks to configure and test.

The JCU team initially turned to Ansible, GitLab, and Jinja2 to standardize and configure deployments. This solution proved more successful; however, the static inventory and individual YAML files were not scalable. The team needed a smooth, quick, and streamlined process for adding devices and making changes.



The Solution: Nautobot Source of Truth & Automation

Recognizing the limitations of a static inventory and individual YAML files, JCU sought a Network Source of Truth (NSoT) with Nautobot.

Several features, including Config Context and GraphQL, made it clear that **Nautobot could solve these challenges**. Config Contexts allowed for dynamic and nuanced variable assignment, and GraphQL expedited inventory generation. **The early successes Nautobot introduced built trust and enhanced visibility** into the desired state of network devices for all users.

The Results: Increased Automation Speed & Flexibility

Nautobot has proven to be the solution the JCU team was looking for to increase the speed and flexibility of automation.

Inventory Runtime Reduction

By optimizing Nautobot's group variable structure and leveraging GraphQL to query the exact information needed from Nautobot, JCU improved the inventory runtime from 5 minutes to under 10 seconds.

Network Configuration Scalability

Config Context gave the JCU greater flexibility in generating configurations by empowering users to replace default group variables with their own validated variables. This enabled the users to quickly and safely change the default behavior of network equipment.

Dynamic Inventory for Nautobot

The team implemented a Python script to dynamically query Nautobot each time AWX generated configurations. This ensured that configurations would always contain the most recent changes represented in the NSoT.

Standardized Device Configuration

A custom Python job in Nautobot facilitates easy device-type changes for equipment, simplifying the lifecycle management of devices. Instead of manually deleting and re-creating these records, users run a job that accomplishes the necessary changes—resulting in new records with standardized device configurations.

Looking Ahead: Evolving with Network to Code Solutions

JCU's journey with automation didn't stop at Nautobot implementation. Over time, the team has incorporated more complex logic into Jinja2 templates, implemented template inheritance, and introduced CI/CD pipelines for testing network configurations. The team plans on incorporating several Nautobot Plugins to streamline the deployment of equipment. Additionally, the JCU has developed a four-month platform engineer onboarding course, to ensure that every team member is well-equipped to navigate the NetDevOps landscape.

For a comprehensive look into how Nautobot is utilized to deploy the world's most critical equipment, we invite you to read the report written by the team at JCU.

[Access the full report here](#)

