

COMMAND CENTER - APPLIANCE MANAGEMENT AUTOMATION

THE PROBLEM

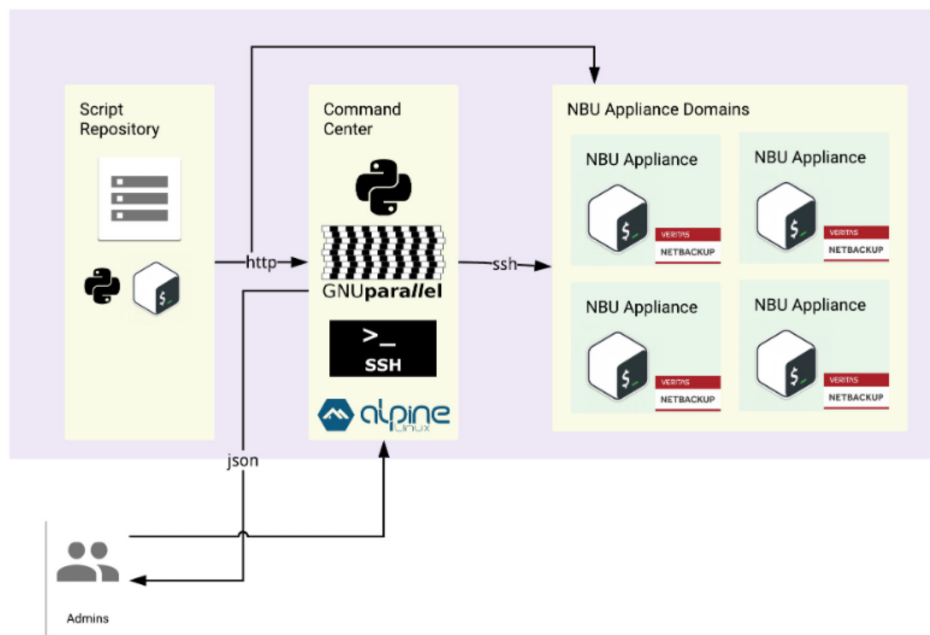
Our client is a global technology company who provides solutions such as multi-cloud data management, data protection, storage optimization, compliance readiness and workload portability. Their client, a Fortune 500 global retailer, had a challenge administering Backup Appliances, which was a manual and cumbersome operation due to the lack of an admin/management portal.

This client needed to query the level of patch applied to each of its Appliances (over 3000 nodes). It was a time-consuming, error-prone manual task, requiring constant attention and input from the administrator.

Our team was tasked with automating the appliance management process. Due to the global retailer's restrictive IT policies, we were confronted with several key limitations:

- Could not use key authentication (ssh-keys,)
- There was no CMDB with the list of nodes
- It took between 2 and 5 minutes for each appliance to be checked manually, since different menus had to be navigated to obtain the information
- Policies dictated that configuration management solutions such as Ansible were not authorized
- The solution had to be agentless, and processes were not to be left running on the appliances
- SNMP was not enabled to perform data extraction from the nodes

HIGH-LEVEL ARCHITECTURE OF THE SOLUTION



In order to optimize the manual and repetitive tasks that do not scale well in large infrastructure deployments, Tangonet Solutions proposed the development of a solution called Command Center, an management automation tool to manage these Appliances, and easily adaptable to manage any physical infrastructure component that supports ssh protocol. Architectural overview

The solution allows IT Infrastructure Administrators to provide a list of nodes on which to interact, credentials and scripts to use on each Appliance through a web layer or via CLI (Command Line Interface). Command Center parallels the execution of the selected script by applying it to the selected component domain, generating a report in json format with the expected outputs, or reporting problems during execution.

It also uses templates that facilitate automation in customized menus or non-standard CLIs. We also contemplated the creation of simple templates prepared to interact with the complexity of NBU.

The solution also allows the client a simple process of permissions and schemes, or git-flow, that allows maintaining control of the published scripts, after the creation of a Script Repository component with Docker.

To validate its operation and test the scalability of the solution, Tangonet built an NBU Appliance emulator, which allowed us to test the interaction of the tool with hundreds of nodes.

BENEFITS OBTAINED

After several brief meetings with the Tangonet's Architecture and Technical group, we were able to capture the requirements and design a solution according to the client's needs.

The customized solution benefited the client in the following ways:

- Client can interact with more than 3000 nodes through an automated engine, substantially reducing administration and management time
- Appliance admin tasks that previously

took hours now take seconds to manage hundreds of nodes

- The existing infrastructure required no modifications, since Command Center was built respecting IT policy restrictions
- Operators can now use custom inventories without using a CMDB
- The scripts created by the administrators are now stored in and managed from a centralized Script Repository - to be reused and rescheduled as needed
- Administrators can now choose to manage appliances via CLI or WebUI, both friendly and intuitive
- Script Repository can reside in the delivered Docker solution or using any GIT repository available on the market providing maximum flexibility
- Being a Docker-based solution, this is the only dependency for deployment
- The solution requires minimum resources to operate (and executes in a matter of seconds for thousands of nodes)
- The solution can be easily and inexpensively customized and upgraded, having been built with Open Source technologies

TECHNOLOGY STACK

- Git
- Docker
- GNU/parallel
- Python
- Ssh
- Script Server
- Alpine Linux

