AVIATRIX

CASE STUDY

# Federal Software Provider Replaces Chokepoint Security with Kubernetes-Native Containment Architecture

A FedRAMP-authorized federal software provider collapses hub-and-spoke firewall inspection into CRD-driven workload containment across GKE, EKS, and Azure GCC High – without adding agents, replacing CNIs, or breaking Cluster-as-a-Service automation.

**14 → 0**

**Chokepoint firewalls replaced by Kubernetes-native Distributed Enforcement at every cluster**

25 K8s clusters • 20+ VPCs • 3 GovClouds • 5 regions • FedRAMP High

### CONTAINMENT, DEFINED

When prevention fails and detection is too slow, containment decides whether the incident becomes a catastrophic breach.

## Hub-and-Spoke Firewalls Were Choking Kubernetes-Native Infrastructure Built for Federal Speed

The company builds software for federal agencies, operating across **AWS GovCloud, Google Cloud (FedRAMP High), and Azure GCC High with 25 Kubernetes clusters spanning GKE and EKS in a Cluster-as-a-Service model.** Every customer tenant runs in a dedicated VPC, with FortiGate High Availability pairs sitting between each tenant and the network – the architecture Aviatrix names **Chokepoint Security**. At 14 firewalls, 20+ VPCs, and 25 clusters, that chokepoint was throttling the very Kubernetes-native velocity the platform was built to deliver.

As the VP of Infrastructure put it:

***"We need a single, standardized firewall solution across all three clouds. The Fortigates can't keep up."***

### COMPANY

Federal Software Provider - Government IT Services

FEDRAMP-authorized federal software provider

SBA 8(a) • INC 5000 • CMMI L3

### CHALLENGES

- 14 FortiGate firewalls in hub-and-spoke model creating chokepoint bottleneck across 20+ VPCs
- 25 Kubernetes clusters (GKE + EKS) in CLaaS model with no granular pod-level security policy
- Manual firewall rule management via FortiManager unable to keep pace with Terraform-driven infrastructure

### RESULTS

- Consistent enforcement and unified governance across AWS GovCloud, GCP, and Azure GCC High through Aviatrix Cloud Native Security Fabric (CNSF)
- Blast Radius contained at the pod and namespace across 25 K8s clusters without agents or CNI replacement with Aviatrix Kubernetes Firewall
- Full Fortinet chokepoint firewall displacement through Aviatrix Distributed Cloud Firewall (DCF)
- Network-layer observability and automation through CRD-driven policy-as-code integrated with CI/CD pipeline

### 01 Every packet had to hair-pin through a firewall pair.

14 FortiGate firewalls (7 High Availability pairs) sat between customer tenant VPCs and the network in a hub-and-spoke model via AWS Transit Gateway. Every east-west flow – between Kubernetes clusters, between tenants, between clouds – traveled through a centralized chokepoint, creating both a performance tax and a single Blast Radius expander: compromise of one firewall pair exposed the entire east-west plane.

### 02 Kubernetes policy stopped at the subnet, not the pod.

FortiGate rules operated on CIDRs generated by Terraform and manually added to FortiManager. The firewalls had no awareness of Kubernetes namespaces, service accounts, or pod identity. In a CLaaS model spinning up new clusters for each customer, the gap between workload granularity and enforcement granularity widened with every deployment.

### 03 Three GovClouds, three policy dialects, one compliance regime.

FortiGate pairs in AWS GovCloud, Palo Alto inspection points at the region level, and Azure GCC High with its own controls each had their own enforcement dialect. FedRAMP audits required reconciling all three – a Vulnerability Deficit that widened every time the cloud estate grew. FIPS 140-2 encryption requirements added another layer of complexity the fragmented model could not address uniformly.

### 04 Manual rule management could not keep pace with CLaaS velocity.

The company had achieved Terraform-native automation for infrastructure provisioning. But security policy still lived in FortiManager templates – customer CIDRs generated in Terraform, then manually added to configure L4 rules on FortiGates. The gap between the speed of cluster creation and the speed of enforcement change was widening at a rate no manual process could close.

## From Chokepoint Inspection to Kubernetes-Native Containment

The company's leadership reached a shared conclusion: **perimeter-style inspection, even virtualized, had hit its architectural ceiling for a Kubernetes-native platform.** The enforcement model had to move from chokepoint to distributed – from "every packet through a firewall pair" to "every pod inside its own containment boundary." After a successful proof-of-concept that met all success criteria, **the VP of Infrastructure moved to full procurement,** reflecting three Containment Era principles:

- **Distributed enforcement beats centralized inspection at Kubernetes scale:** 25 clusters carrying their own CRD-driven policy outperforms 14 firewalls trying to inspect their traffic – and eliminates the single-point failure mode that put every tenant at risk.
- **Policy follows the pod, not the network path:** SmartGroups and Kubernetes-native CRDs replace CIDR-bound FortiManager rules that break every time a cluster scales or a namespace changes.
- **Containment shrinks Blast Radius at the workload:** Every pod sits inside its own enforcement boundary – bounding lateral movement to a single namespace if an attacker lands, not the entire east-west plane the chokepoint model exposed.

## The Math Behind the Move

Independent 2026 research formalized the gap that drove this decision. The **Vulnerability Deficit Equation** proves the median enterprise must patch **6.5× faster** than is physically achievable to keep pace with exposure. CISA KEV data shows median time-to-exploit has moved to **negative seven days**. And **82% of intrusions** now use valid credentials, not unpatched CVEs. Detection cannot close that gap. Only containment – architectural enforcement that holds independent of whether the compromise has been detected – can.

## CRD-Driven Automation, Agentless Architecture, and Cluster-as-a-Service Security

Every element of the company's Aviatrix estate – 26 spokes, 5 transits, DCF policies, Kubernetes firewall CRDs – integrates directly with the existing Terraform and CI/CD pipeline. With Aviatrix Professional Services engaged for the FortiGate-to-DCF migration, new SmartGroups and pod-level policies deploy in the same release cadence as application code. Kubernetes Custom Resource Definitions drive policy-as-code: a YAML manifest in the CI/CD pipeline creates, modifies, or removes firewall rules without touching a console – the CLaaS model this provider's federal customers require.

The architecture is fully agentless – **no sidecars, no CNI replacement, no agents on any workload.** Aviatrix integrates with Kubernetes via CRDs and webhooks, delivering pod-level granularity across GKE and EKS without disrupting the existing Istio service mesh or requiring changes to application code. FIPS-validated encryption secures all inter-cloud transit across the three GovCloud environments.

The outcome maps to every testable property of containment.

01 **Path-complete:** enforcement governs every communication path at every one of 25 clusters and 26 spokes.

02 **Identity-aware at L7:** CRD-driven policy operates at the granularity of Kubernetes namespace, service account, and pod identity – not IP addresses and subnets.

03 **Compute-model agnostic:** policy reaches pods, VMs, containers, and managed services without installing an agent on any workload – no sidecars, no CNI replacement.

04 **Universally propagated:** a single CRD or SmartGroup change enforces across 20+ VPCs, three GovCloud environments, and five regions in subseconds.

05 **Detection-independent:** the enforcement state holds before, during, and after any breach.

Together, the architecture closes the Vulnerability Deficit the prior Chokepoint Security model kept widening – without adding agents, replacing CNIs, or introducing a second control plane.

## Results and Business Value

| Area | Before CNSF | With CNSF |
|---|---|---|
| **Enforcement Model** | 14 FortiGate firewalls (7 High Availability pairs) hair-pinning traffic from 25 K8s clusters and 20+ VPCs | Distributed enforcement at every cluster, namespace, and pod |
| **Kubernetes Security** | CIDR-based L4 rules in FortiManager with no pod, namespace, or service account awareness | CRD-driven pod-level containment with namespace and service account granularity |

| | | |
|---|---|---|
| **Multicloud Governance** | Fragmented policy across AWS GovCloud, GCP, Azure GCC High, and FortiManager templates | Unified communication governance across 3 GovClouds from a single policy plane |
| **Deployment Velocity** | Manual CIDR-to-FortiManager rule pipeline lagging Terraform-driven cluster creation | CRD-driven policy-as-code deploying through CI/CD in lockstep with CLaaS provisioning |
| **Blast Radius** | Chokepoint model – single firewall pair compromise exposes entire east-west plane | Pod-level containment – lateral movement bounded to a single namespace |

## The Question for Every Federal Platform Executive

If a valid credential is used against a Kubernetes workload serving a federal agency, what – architecturally – prevents it from reaching the tenant data in an adjacent namespace? **In the Containment Era, that question has one answer: containment, enforced at the workload, on every path, independent of detection.**

## About Aviatrix

Aviatrix® is pioneering the Cloud Native Security Fabric (CNSF) – the architecture the Containment Era requires. CNSF governs every workload communication path across every cloud, every VPC, every Kubernetes cluster, and every serverless function, from a single policy plane. One rule. Universal propagation. Enforced at the workload, not at a chokepoint. Trusted by more than 500 of the world's leading enterprises. For more information, visit aviatrix.ai.