AVIATRIX

SOLUTION BRIEF

Contain AWS Bedrock AgentCore

Powered by Aviatrix + AWS

Aviatrix Validated Containment Architectures are lab-tested containment deployment blueprints for the AI platforms enterprises are actually running. Ship-ready, policy-included, validated before they arrive. This Validated Containment Architecture covers AWS Bedrock AgentCore.

The Threat

By default, an AWS Bedrock AgentCore Runtime can reach any destination on the internet the moment it starts. Its outbound tool calls, model invocations, and remote Model Context Protocol (MCP) connections are indistinguishable on the wire from legitimate work. Compromised or misbehaving agent traffic looks identical to sanctioned traffic – until it appears in flow logs the next day.

Three problems compound this exposure:

Threat	What It Means
Arbitrary agent egress	Any outbound flow to unsanctioned model providers, attacker-controlled domains, or public file-sharing services is permitted by default. A prompt-injected agent exfiltrating confidential data faces no network-layer barrier.
Supply-chain compromise of sanctioned hosts	Agents legitimately reach domains such as raw.githubusercontent.com for tooling. Attackers exploit this trust: a poisoned path on a sanctioned domain is indistinguishable by simple domain allowlist. The Shai-Hulud npm-worm Infrastructure as Code pattern demonstrates the risk.
Control-plane drift to PUBLIC mode	A developer can create an AgentCore Runtime outside Virtual Private Cloud (VPC) mode, running on AWS-managed infrastructure with no customer VPC endpoint. That Runtime is invisible to any customer-controlled enforcement point by construction.

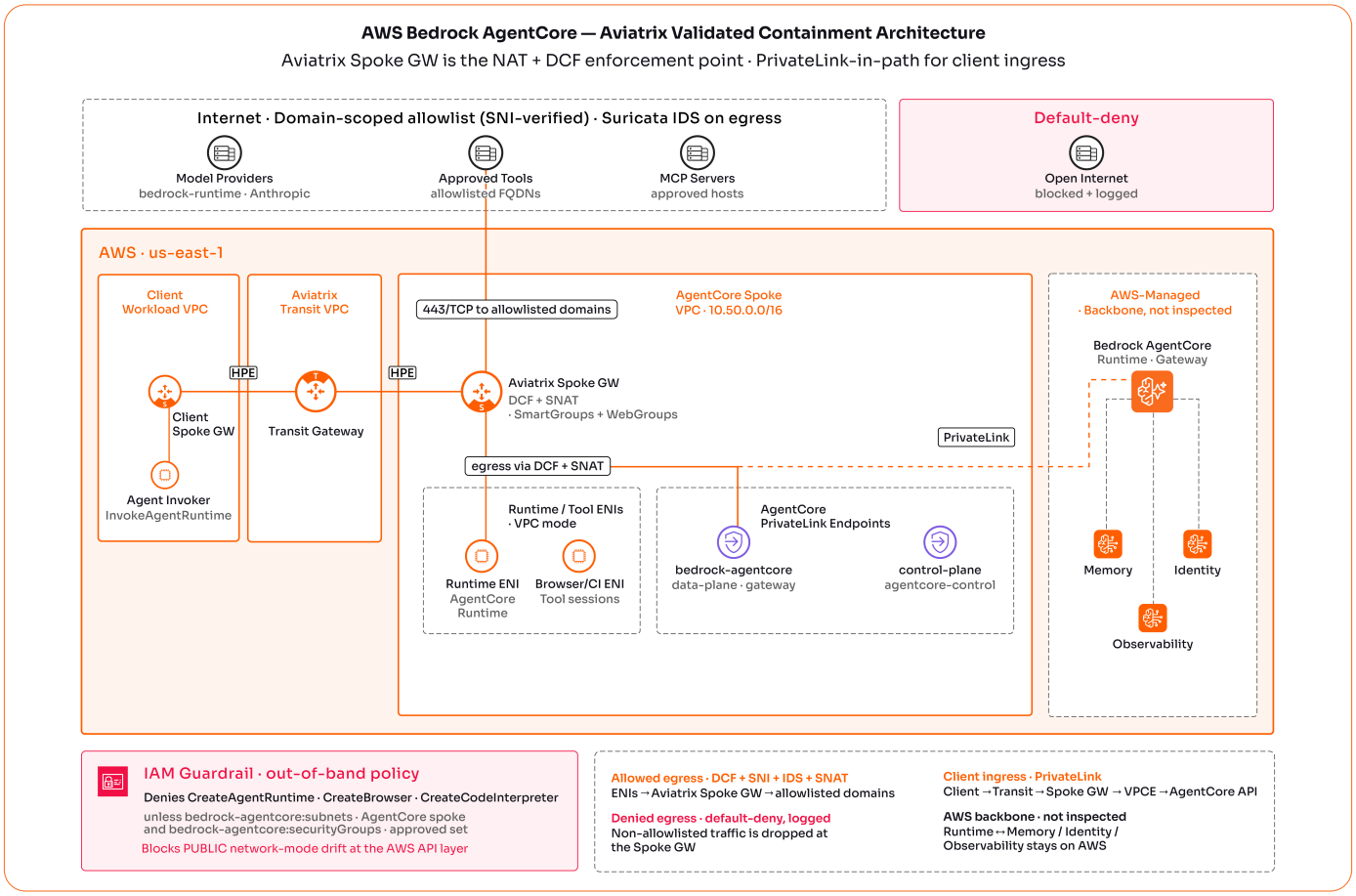
WHY THIS MATTERS

The March 2026 supply chain wave – compromising Axios (70 million weekly downloads), LiteLLM, Trivy, KICS, and Telnyx – proved that trusted code running in trusted pipelines is invisible to detection. AgentCore workloads are the same class of risk: ephemeral, highly privileged, and shipping fast. A single default-deny spoke-gateway policy stops the exfiltration before the TCP handshake completes. That is the control the detection stack cannot provide.

The Architecture

The VCA uses two insertion layers, both transparent to the agent container:

Layer	What It Does
Egress enforcement	The AgentCore Runtime's VPC-mode Elastic Network Interface (ENI) is placed in a subnet whose default route points to the Aviatrix Spoke Gateway. Every outbound flow – tool calls, model invocations, MCP egress, Domain Name System (DNS) – traverses the gateway inline. Default-deny policy pack applies. No sidecar, no agent on the container.
Ingress enforcement	Two interface VPC endpoints – bedrock-agentcore (data plane) and bedrock-agentcore-control (control plane) – live in the same spoke. Client workloads invoking agents route through the Aviatrix transit. Distributed Cloud Firewall applies ingress allow rules keyed on the client VPC SmartGroup.
Identity and Access Management (IAM) containment	IAM condition keys on bedrock-agentcore:subnets and bedrock-agentcore:securityGroups prevent Runtime creation outside the approved landing zone. A PUBLIC-mode or foreign-subnet Runtime fails at the AWS API before any packet flows.
TLS decryption	URL-path enforcement blocks IoC patterns on sanctioned domains. All other traffic – ECR pulls, Bedrock SigV4 calls, MCP connections – remains encrypted end-to-end with original certificate chains intact.



Three Things Your Current Stack Can't Do

The VCA uses two insertion layers, both transparent to the agent container:

01 Put an enforcement point in front of Bedrock AgentCore egress

By default, AgentCore Runtime tool calls, model invocations, and MCP server connections route to the open internet through AWS-managed infrastructure with no inline inspection point your security team can see. Compromised agent traffic is indistinguishable from legitimate calls on the wire. AWS manages the service; AWS does not police what the agent sends outbound. Aviatrix is the firewall in front of it.

---

02 Block a supply-chain attack that hides malicious content on a sanctioned host

SNI-based allowlisting cannot distinguish a clean path on raw.githubusercontent.com from a poisoned one – the TLS endpoint is the same. Aviatrix applies selective transparent TLS decryption scoped tightly to a destination FQDN SmartGroup, inspects URL paths inside that scope only, and denies the compromised path while the legitimate path on the same host returns HTTP 200. Your current stack cannot make that distinction.

---

03 Prevent an agent Runtime from being created outside the landing zone in the first place

Without IAM condition keys on bedrock-agentcore:subnets and bedrock-agentcore:securityGroups, nothing stops a developer from spinning up a Runtime in PUBLIC mode or in a foreign subnet – bypassing every network control before a single packet flows. The VCA ships an IAM managed policy that denies the API call itself if the subnets or security groups aren't from the approved set. The breach path is closed at the AWS API, not detected after the fact.

Compliance Evidence

For HIPAA, PCI-DSS 4.0, SOC 2, FedRAMP, and EU AI Act environments, auditors require architectural proof of enforcement, not a policy document. CoPilot Distributed Cloud Firewall logs, IAM policy, Aviatrix SmartGroups, and the baseline, version-controlled Distributed Cloud Firewall pack provide continuous records and containment boundaries enforced by architecture. The proof of enforcement is the architecture itself – not a statement about the architecture.

Questions Worth Asking

- How many AI agents are running in your environment right now – and how many are fully inventoried?
- If a compromised agent attempted to exfiltrate data to an external domain, what in your stack would catch it?
- How are you currently proving to auditors that AI workload egress is controlled – not just described in policy?

What's Included

Deliverable	Detail
Insertion pattern	Architecture diagram and configuration notes showing how AgentCore Runtime session egress is projected into a customer VPC (AgentCore VPC mode) and how the AgentCore API is fronted by in-path PrivateLink consumer endpoints – both halves of AgentCore traffic landing in a single Aviatrix-attached spoke.
SmartGroup model	Five SmartGroups keyed to AgentCore's identity primitives: a subnet SmartGroup for Runtime ENIs, FQDN SmartGroups for data-plane and control-plane PrivateLink hostnames, and a destination FQDN SmartGroup scoped to supply-chain hosts where URL-path enforcement is required.
Baseline Distributed Cloud Firewall policy packt	Seven default-deny DCF policies covering: WebGroup-scoped allowlists for sanctioned model providers and tool-call destinations; a dedicated rule blocking DNS-tunneled exfiltration on UDP/53; a URL-filter deny backed by selective TLS decryption for Shai-Hulud-style supply-chain IoCs; and an IAM managed policy that denies Runtime creation in PUBLIC mode or outside the landing zone at the AWS API layer.
Bill of Materials	One Aviatrix Transit Gateway, two Aviatrix Spoke Gateways (AgentCore and client), two interface VPC endpoints (data plane and control plane), four WebGroups, five SmartGroups, seven DCF policies, the ECR repo, and a sample AgentCore Runtime. Idle cost approximately $1.10/hr. Controller 8.1+ required for baseline; Controller 9.0+ required for selective TLS decryption.
GitHub Repository	Full Terraform at aviatrix-blueprints/blueprints/agentcore-aws/ – deploys end-to-end in 25–30 minutes with one terraform apply, destroys cleanly with one command and leaves zero orphans. Includes a probe script and a Streamlit scenario UI with live demonstrations of all five OWASP LLM Top Ten and MITRE ATLAS scenarios covered by the VCA.

Get Started

The Validated Containment Architecture for AWS Bedrock AgentCore is available today. The Terraform blueprint, scenario UI, and deployment guide ship together.

- Aviatrix Distributed Cloud Firewall customers on Controller 8.1 or later can deploy the blueprint today.
- New customers can start with Aviatrix Secure Egress – replace the AgentCore VPC NAT gateway with a Spoke Gateway in egress mode.
- Existing AWS Transit Gateway customers can deploy Aviatrix in-path via Inspection for AWS Transit Gateway with zero routing changes.

Request a 30-minute architecture review

Walk through the enforcement model in your environment, map your current AgentCore Runtime inventory, and identify the egress paths you currently cannot see.

About Aviatrix

Aviatrix® is pioneering the Cloud Native Security Fabric – the architecture the Containment Era requires. The Cloud Native Security Fabric 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.