AVIATRIX

TECHNICAL BRIEF

Contain AWS Bedrock AgentCore

Powered by Aviatrix + AWS

Aviatrix Distributed Cloud Firewall · Reference Architecture for AWS Bedrock AgentCore

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

Threat Context

AWS Bedrock AgentCore is one of the leading platforms for creating and deploying AI agents at scale, but it presents a challenge for security teams: ungoverned access to the internet. AgentCore Runtime Virtual Private Cloud (VPC) mode drops the per-session micro-Virtual Machine (VM) Elastic Network Interface (ENI) in customer-controlled subnets. If you haven't set an explicit egress policy, the subnet default route permits any internet destination.

This configuration creates three problems for security teams:

- **Unconstrained egress:** any outbound TCP/UDP from the Runtime subnet, including to attacker-controlled domains and unsanctioned model providers.
- **Supply-chain IoC paths on sanctioned domains:** raw.githubusercontent.com and similar hosts are legitimately required by agents – SNI-only allowlisting cannot distinguish a compromised path from a legitimate one.
- **Control-plane drift:** a Runtime created in PUBLIC mode has no customer ENI and bypasses every customer enforcement point by construction.

LAB-VALIDATED THREAT SCENARIO

The Aviatrix lab validated this architecture against six Streamlit scenario cards: LLM01 prompt-injection tool-abuse exfiltration, LLM02 DNS-tunneled exfiltration, LLM05 compromised MCP source, LLM05 URL-path variant on raw.githubusercontent.com (Shai-Hulud IoC pattern), LLM08 shadow routing to unsanctioned models, and control-plane drift to PUBLIC mode. Each scenario card maps the attack to OWASP LLM Top Ten and MITRE ATLAS entries, the Distributed Cloud Firewall rule that closes it, and the CoPilot view where the operator confirms the block.

Prerequisites

Before configuring Distributed Cloud Firewall (DCF) enforcement, verify the following are in place:

- **Aviatrix Controller 8.1 or later.** Controller 9.0 required for TLS decryption on supply-chain FQDN.
- **Aviatrix Spoke Gateway deployed in the AgentCore landing-zone VPC:** minimum one High Availability (HA) pair (two gateways), c5.xlarge or equivalent.
- **AgentCore Runtime configured for VPC mode.** A PUBLIC-mode Runtime has no customer ENI and does not traverse DCF.
- **VPC Flow Logs enabled on the AgentCore landing-zone VPC.** Required for AgentGuard Shadow AI Discovery and for CoPilot DCF Monitor ingestion.
- **IAM role for the Aviatrix Controller** with permissions to read VPC and subnet metadata in the landing-zone account.
- **Per-session microVM ENIs must land in a dedicated Runtime subnet** whose default route points to the Aviatrix Spoke Gateway.
- **Two interface VPC endpoints provisioned in the landing-zone VPC:** com.amazonaws.[region].bedrock-agentcore and com.amazonaws.[region].bedrock-agentcore-control.

> **CONTROLLER VERSION NOTE**
>
> Controller 8.1 delivers all baseline DCF enforcement: subnet SmartGroups, FQDN WebGroups - used for model providers and MCP servers, default-deny policy pack, and CoPilot DCF logging. TLS decryption for URL-path enforcement on supply-chain hosts requires Controller 9.0. The blueprint repository is versioned for both. Deploying on 8.1 produces a fully functional containment deployment; the supply-chain URL-filter rule activates automatically when the controller is upgraded to 9.0.

What Ships on Day One

The architecture is one repository, one Terraform module, one set of validated controls. Everything required to deploy on a fresh AWS account is in the box.

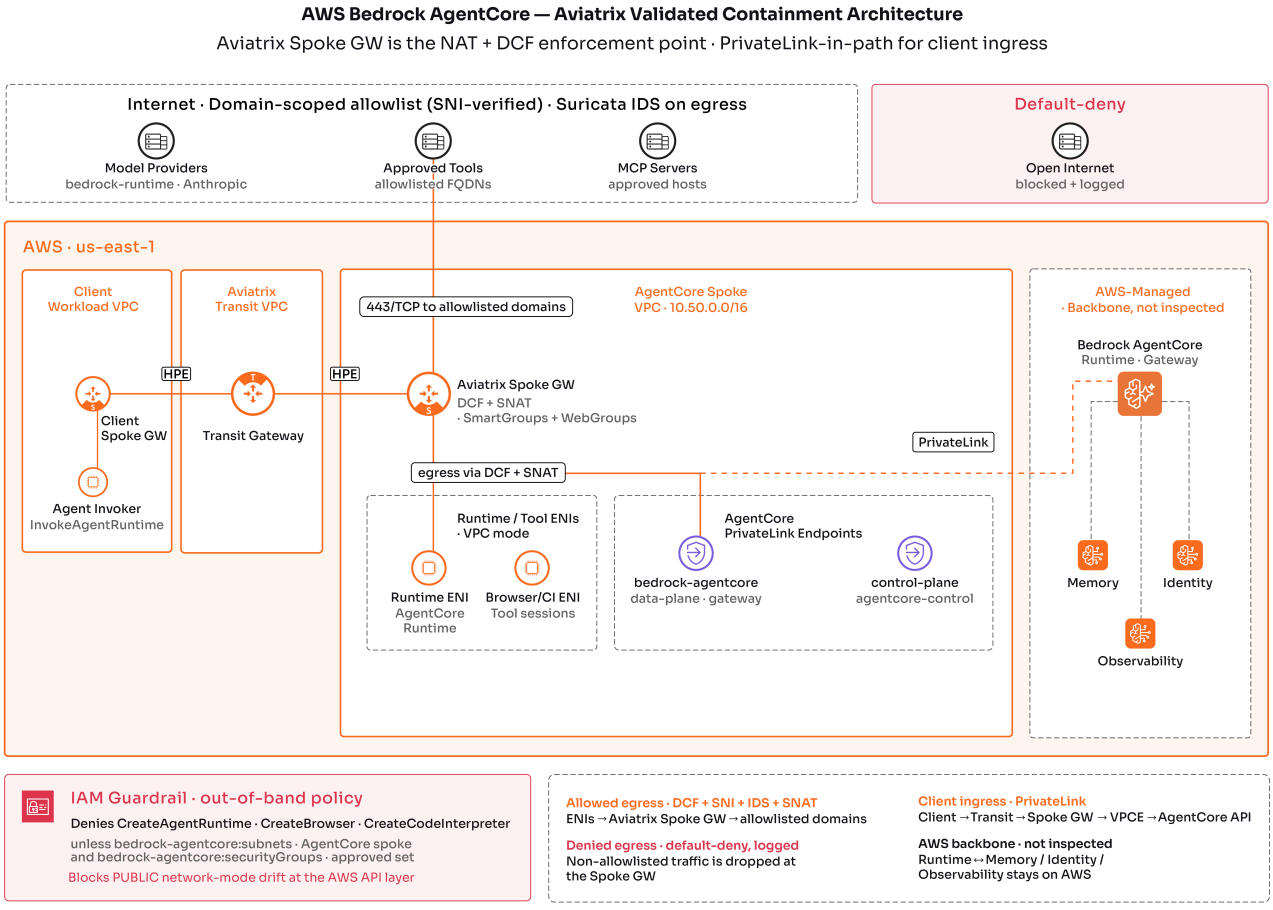
Insertion pattern. Architecture diagram and configuration notes showing exactly where Distributed Cloud Firewall enforcement goes in the AgentCore topology. Pattern 1 (Spoke Gateway in-path for Runtime egress) combined with Pattern 3 (Private Link consumer endpoints for the AgentCore API), in a single Aviatrix-attached landing zone.

SmartGroup model. A subnet SmartGroup for the AgentCore Runtime ENIs. FQDN WebGroups for the data-plane and control-plane PrivateLink host names. A VPC SmartGroup for the client spoke. A destination FQDN WebGroup scoped tightly to supply-chain hosts where URL-path enforcement is required.

Baseline Distributed Cloud Firewall policy pack. Seven policies, ordered for first-match: allow client spoke to data-plane PrivateLink; allow client spoke to control-plane PrivateLink; deny supply-chain IoC paths via URL filter (priority 29, decryption-enabled); allow Runtime subnet to sanctioned model providers; allow Runtime subnet to sanctioned tool destinations; allow Runtime subnet to approved AWS control-plane domains; allow Runtime subnet to sanctioned MCP servers; deny outbound UDP/53 (DNS exfil); default-deny everything else. Every rule logs to CoPilot DCF Monitor with a human-readable name.

Bill of materials. A single pair of Aviatrix Spoke Gateways in the AgentCore VPC. One landing zone. Aviatrix licenses per landing zone – one base, one advanced visibility, one advanced security. The architecture is sized in landing zones, not in agents, sessions, or invocations.

GitHub repository. aviatrix-blueprints/blueprints/agentcore-aws. Infrastructure as code for the full deployment. A probe script that runs the containment tests from the client-spoke EC2 over SSM. A Streamlit scenario UI that runs the OWASP LLM Top Ten and MITRE ATLAS scenarios live against the deployed environment – including LLM01 prompt-injection tool-abuse exfil, LLM02 DNS-tunneled exfil, LLM05 compromised MCP source, LLM05's URL-path variant on raw.githubusercontent.com, LLM08 shadow routing to unsanctioned models, and the control-plane drift scenario. Each scenario card maps the attack to the OWASP and MITRE entries, the Distributed Cloud Firewall rule that closes it, and the CoPilot view where the operator confirms the block.



Every flow into and out of the AgentCore landing zone reaches the Aviatrix Spoke Gateway in-path. Sanctioned models, sanctioned MCP servers, and approved AWS APIs are explicitly permitted by SmartGroup. Unsanctioned hosts – catbox.moe, unknown model providers, IoC URL paths – fall to the default-deny rule and are logged with a human-readable rule name.

SmartGroup and WebGroup Design

The policy model uses five SmartGroup objects and the Aviatrix-managed AI WebGroups. All objects are defined in the Aviatrix Controller and referenced by name in the DCF policy pack.

Object	Type / Scope / Purpose
agentcore-runtime -subnet	Subnet SmartGroup – matches the Runtime subnet CIDR in the landing-zone VPC. Source identity for all egress rules targeting the Runtime.
agentcore -dataplane-fqdn	FQDN WebGroup– matches the bedrock-agentcore PrivateLink hostname for the landing-zone region. Destination identity for the ingress allow rule from the client spoke.
agentcore-control plane-fqdn	FQDN WebGroup– matches the bedrock-agentcore-control PrivateLink hostname. Destination identity for the control-plane ingress allow rule.
client-spoke-vpc	VPC SmartGroup – matches the client VPC that invokes AgentCore sessions. Source identity for ingress allow rules to the PrivateLink endpoints.
supply-chain-fqdn-group	FQDN WebGroup– scoped to supply-chain hosts where URL-path enforcement is required (raw.githubusercontent.com and equivalents). Used exclusively in the priority-29 URL-filter rule with decrypt_policy = DECRYPT_ALLOWED. All other rules set DECRYPT_NOT_ALLOWED explicitly.
avx-ai-llm-providers (managed)	Aviatrix-managed AI WebGroup – curated, auto-updated destination list covering all major LLM API providers. Used in the allow-sanctioned-models rule.
avx-ai-mcp-agent-platforms (managed)	Aviatrix-managed AI WebGroup – curated destination list for sanctioned MCP server platforms. Used in the allow-sanctioned-mcp rule.

CRD-defined WebGroups for per-MCP-server scoping are separate from the above and live in the Kubernetes cluster namespace, not on the Aviatrix Controller. They are referenced by the MCP server's FirewallPolicy CRD and are not required for the base AgentCore VCA.

Distributed Cloud Firewall Policy Pack

Seven rules in priority order. First match wins. Every rule is logging-enabled and writes to CoPilot DCF Monitor with a human-readable name. Monitor mode is enabled by default for all rules; promote to enforcement rule by rule as validated against production traffic.

Pri	Rule Name	Source	Destination	Action / Notes
10	allow-client-dataplane	client-spoke-vpc	agentcore-dataplane-fqdn	PERMIT · TCP 443 · Ingress to PrivateLink data plane from client VPC
15	allow-client-controlplane	client-spoke-vpc	agentcore-controlplane-fqdn	PERMIT · TCP 443 · Ingress to PrivateLink control plane from client VPC
20	allow-sanctioned-models	gentcore-runtime-subnet	avx-ai-llm-providers	PERMIT · TCP 443 · Egress to sanctioned LLM API providers (Aviatrix-managed, auto-updated)
25	allow-sanctioned-tools	agentcore-runtime-subnet	Custom tool FQDN list	PERMIT · TCP 443 · Egress to explicitly approved tool endpoints (operator-maintained)
29	deny-supply-chain-ioc	agentcore-runtime-subnet	supply-chain-fqdn-group	DENY · URL filter · decrypt_policy=DECRYPT_ALLOWED · Blocks IoC URL paths on sanctioned domains; legitimate paths return 200
30	allow-aws-control-plane	agentcore-runtime-subnet	AWS service FQDNs	PERMIT · TCP 443 · ECR, STS, SSM, CloudWatch endpoints · DECRYPT_NOT_ALLOWED explicit
35	allow-sanctioned-mcp	agentcore-runtime-subnet	avx-ai-mcp-agent-platforms	PERMIT · TCP 443 · Egress to sanctioned MCP server platforms (Aviatrix-managed, auto-updated)
40	deny-dns-exfil	agentcore-runtime-subnet	ANY	DENY · UDP/TCP 53 to non-VPC-resolver IPs · Blocks DNS tunneling
1000	default-deny	agentcore-runtime-subnet	ANY	DENY · All protocols · Final default-deny. Every match logged to CoPilot FlowIQ.

IAM Policy — Control-Plane Containment

The IAM managed policy uses condition keys to prevent AgentCore Runtime creation outside the approved landing zone. Attach to the IAM role used by developers and CI/CD pipelines that create AgentCore resources.

```
{
 "Effect": "Deny",
 "Action": [
  "bedrock-agentcore:CreateAgentRuntime",
  "bedrock-agentcore:CreateAgentRuntimeEndpoint",
  "bedrock-agentcore:UpdateAgentRuntime"
 ],
 "Resource": "*",
 "Condition": {
  "StringNotEquals": {
   "bedrock-agentcore:subnets":
    ["subnet-<approved-id-az1>", "subnet-<approved-id-az2>"],
   "bedrock-agentcore:securityGroups":
    ["sg-<approved-sg-id>"]
  }
 }
}
```

A PUBLIC-mode Runtime (NetworkMode omitted or set to PUBLIC) has no subnets or security groups in the request context. The Deny condition evaluates to true and the API call fails with an explicit deny before any packet flows. This closes the control-plane escape vector without requiring any network-layer change.

TLS Decryption — URL-Path Enforcement

TLS decryption in this VCA is scoped leveraged to demonstrate deeper control via a URL-typed WebGroup. The decision is explicit and documented.

Configuration Element	Detail
supply-chain-fqdn-group	Contains raw.githubusercontent.com and equivalent supply-chain hosts where URL-path enforcement is required. Operator-maintained; seeded from the VCA blueprint repository.
MITM CA provisioning	The Aviatrix MITM CA is provisioned to AWS Secrets Manager during terraform apply. For full supply-chain enforcement including egress TLS from the Runtime container itself, add a fetch step to the Dockerfile that retrieves and installs the CA bundle. A five-line script is provided in the blueprint repository.

Operational Safety Properties

Three structural properties ensure the deployment is safe to operate in production environments and won't disrupt agent activity:

Property	Implementation Detail
Monitor before enforce	Every DCF rule in the policy pack deploys in watch mode by default: same rule, same dataplane, logging only. Promote to enforcement by toggling the Enforce flag. Rollback is the same flag in reverse. The dataplane never reloads.
One VPC at a time	Every deployment step is scoped to the AgentCore landing-zone VPC. A policy regression on this VPC has zero impact on any other VPC in the fabric. Back-out is one action: disable spoke-gateway insertion on the AgentCore VPC. Nothing else is touched.
Policy propagation in ~100ms	DCF rule changes propagate via a kernel-level eBPF map update. The dataplane never reloads. A rule change or full back-out completes across the landing zone in under a second with no service disruption.

Known Constraints

Three structural properties ensure the deployment is safe to operate in production environments and won't disrupt agent activity:

Constraint	Workaround / Notes
AgentCore PUBLIC mode	A PUBLIC-mode Runtime has no customer ENI and does not traverse DCF. The IAM condition-key policy is the only control for this mode. Ensure the IAM policy is attached before allowing AgentCore Runtime creation in the account.
GitHub-hosted runners	If tool calls route through GitHub-hosted runners (not self-hosted), the runner's egress does not traverse the AgentCore VPC. Contain GitHub Actions AI pipelines separately using VCA-04.
MITM CA in Runtime container	TLS decryption for egress HTTPS from the Runtime container requires the Aviatrix MITM CA in the container trust store. Without it, the agent sees certificate errors on decrypted flows. The CA is provisioned to Secrets Manager; the fetch script is in the repository.
AgentCore Browser, Code Interpreter, Gateway	This VCA release covers AgentCore Runtime only. Browser, Code Interpreter, and Gateway are deferred. The MITM CA is provisioned to accommodate them in a follow-on release.

DNS resolver requirement	The deny-dns-exfil rule blocks UDP/TCP 53 to non-VPC-resolver IPs. The AgentCore Runtime subnet must be configured with a VPC DHCP option set that points to the AWS-provided resolver (169.254.169.253 or the VPC+2 address). Custom resolvers must be pre-approved in the rule.

Appendix — Cloud-Specific Domain

The following AWS service domains must be reachable from the AgentCore Runtime subnet. Add to a custom WebGroup and reference in the allow-aws-control-plane rule. All entries require HTTPS (TCP 443) only.

Domain	Purpose
bedrock-agentcore.[region] .amazonaws.com	AgentCore data-plane API (PrivateLink endpoint hostname – resolves to VPC endpoint IP)
bedrock-agentcore-control. [region].amazonaws.com	AgentCore control-plane API (PrivateLink endpoint hostname)
***.dkr.ecr.[region]. amazonaws.com**	ECR image pull for the AgentCore microVM bootstrap layer and the agent container image
sts.amazonaws.com	STS AssumeRole for the Runtime execution role and tool-call credential vending
ssm.[region].amazonaws.com	AWS Systems Manager (SSM) for probe-script access via Session Manager
logs.[region].amazonaws.com	CloudWatch Logs for Runtime execution logs and CoPilot telemetry
secretsmanager.[region]. amazonaws.com	Secrets Manager access for MITM CA retrieval by the Runtime container (if TLS decryption is enabled)
bedrock.[region].amazonaws. com	Bedrock model invocation API endpoint (required if the agent invokes Bedrock models directly; not required if routing through bedrock-agentcore endpoint only)

Domain syntax for FQDN WebGroups: exact hostnames or leading wildcards (*.dkr.ecr.[region].amazonaws.com). Bare * is rejected by the Controller. Substitute [region] with the AWS region identifier (for example, us-east-1).

Aviatrix Validated Containment Architecture for AWS Bedrock AgentCore removes the unknown from agentic deployment by enforcing policy at the network layer. **Ask your Aviatrix account team** for a guided deployment focused on one AgentCore landing zone.

Explore Validated Containment Architectures for other AI platforms.

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.