AVIATRIX

TECHNICAL BRIEF

Secure Enterprise GitHub Pipelines

Powered by Aviatrix — Self-Hosted Runner Edition

Aviatrix Distributed Cloud Firewall · Reference Architecture for Self-Hosted GitHub Actions Runners

Aviatrix Validated Containment Architectures are lab-tested containment deployment blueprints for the enterprise platforms you are actually running. They are ship-ready, policy-included, and validated before they arrive. This Validated Containment Architecture covers Secure Enterprise GitHub Pipelines for self-hosted runners.

Threat Context

When using the native egress path, every GitHub Actions runner operates as an ungoverned egress path – there is no egress firewall, no traffic logging, and no destination policy.

The CI/CD exfiltration trifecta describes the three conditions that create an exfiltration vector: runners with access to secrets and credentials ($GITHUB_TOKEN, cloud keys, SSH keys), exposure to untrusted third-party code (public Actions, npm/PyPI packages), and the ability to reach any internet destination. GitHub Actions pipelines stack the first two legs by design. This architecture breaks the third.

> **LAB-VALIDATED THREAT SCENARIO**
>
> A self-hosted runner executes npm install, which fetches a supply chain-compromised package. The package reads $MY_SECRET, then attempts HTTPS to a rogue webhook endpoint to exfiltrate data. The runner has permission for the credential, not the destination. Aviatrix sees the destination, applies the DCF WebGroup policy, and blocks the egress. The attempt is logged with full attribution. The secrets have nowhere to go.

Prerequisites

Before configuring egress enforcement for self-hosted runners, verify the following are in place:

- **Self-hosted runners in VPC/VNET:** Aviatrix Spoke Gateway in the runner VPC, private subnets with route table 0.0.0.0/0 pointing to Spoke Gateway Private IP, single_ip_snat = true on the gateway.
    - Zero bypass risk: the runner has no public IP and the route table is managed outside the runner's trust boundary.

What Ships on Day One

The following deliverables ship with the Validated Containment Architecture for Secure Enterprise GitHub Pipelines – Self-Hosted Runners:

Self-hosted runners

- Creates a VNET/VPC, deploys Aviatrix Spoke Gateway, configures routing to send runner traffic to the Aviatrix Spoke Gateway, configures egress policies and starts enforcement.

A README is also included with a step-by-step deployment guide, test scenarios (permit passes, block fires, live policy update), cleanup instructions, and troubleshooting reference.

Runner Deployment Layout

Runner Type	Deployment Context	Enforcement Method
Self-Hosted Runners (VNET, VPC ...)	Inside a subnet	VPC route table 0.0.0.0/0 → Aviatrix Spoke Gateway. Network-level enforcement, zero bypass risk.

Policy Evaluation Order

Every packet from every self-hosted runner traverses three tiers in order:

Tier 1: Named Permit/Deny Rules

- CIDR permit for any required cloud service endpoints (no-SNI traffic requires CIDR, not FQDN)
- Cloud infrastructure egress permit (ECR/ACR, blob storage, container registries) scoped to runner subnet CIDR
- Default-deny to Default ThreatGroup

Tier 2: Per-Runner-Group WebGroup Rules

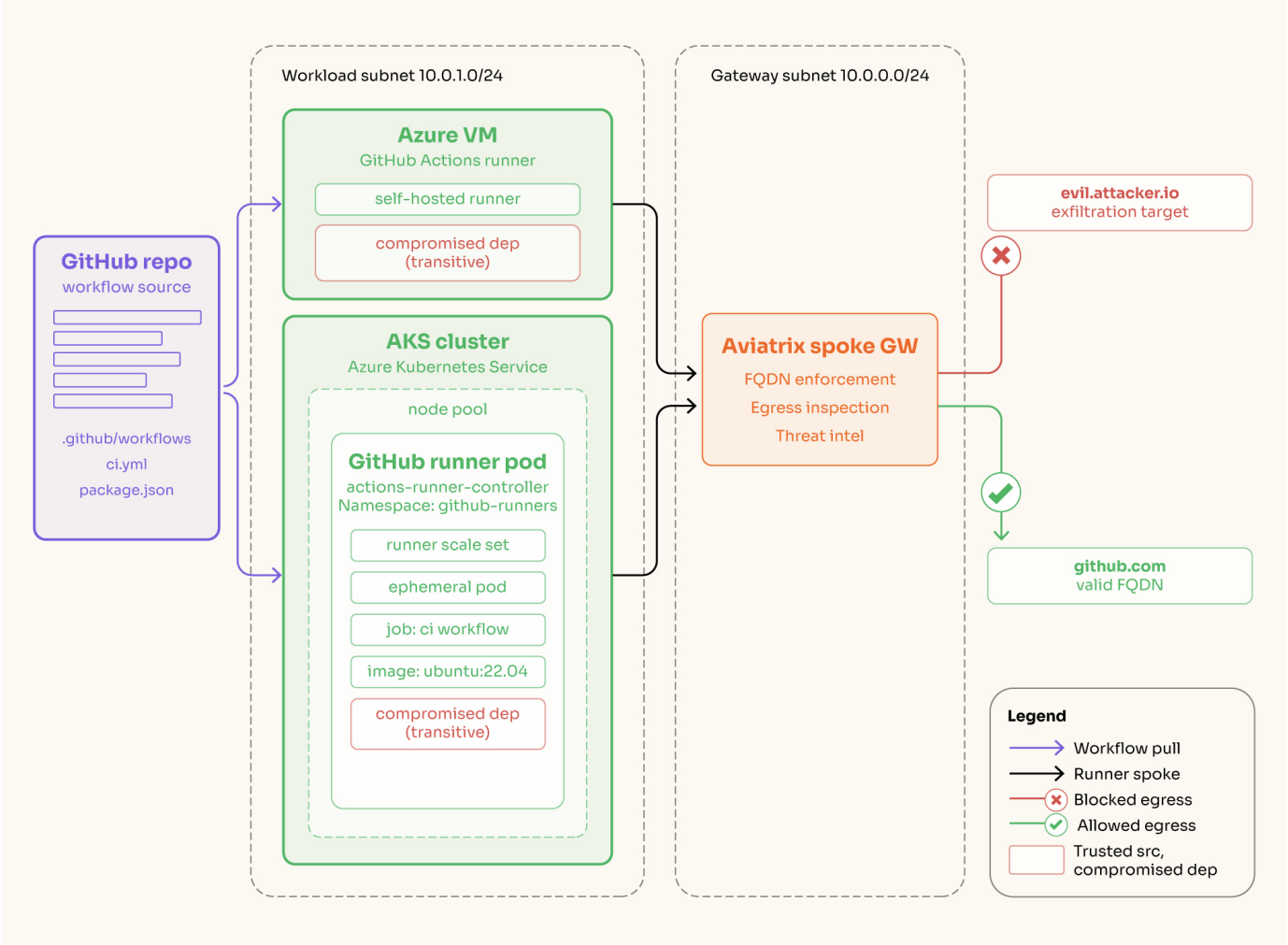
SmartGroups match runner VMs by subnet or tag. WebGroups define approved egress destinations by FQDN per runner group. Different runner groups carry different policies. If no rule matches a runner, traffic falls through to Tier 3.

Tier 3: Default Deny

POST_RULES default action: DENY + LOG. Catches everything not explicitly permitted in Tier 1 or Tier 2. This is not a fallback; it is the posture. Must be configured as a default action rule, not as a named policy entry.

WHY VPC ROUTING PROVIDES ZERO BYPASS RISK

VPC routing (private subnet, route table → Spoke Gateway) enforces egress at the network layer outside the runner entirely. No software process on the runner can circumvent the route table. The runner has no public IP. Enforcement is owned by the cloud network infrastructure, not by any agent or process within the runner's trust boundary.



Known Constraints

Constraint	Workaround / Notes
Route table dependency on cloud provider	Ensure Spoke Gateway Private IP is used as the route target and single_ip_snat = true is set on the gateway. Validate routing after any VPC/VNET topology change.

Aviatrix Validated Containment Architecture for Secure Enterprise GitHub Pipelines removes the unknown from CI/CD deployment by enforcing egress policy at the network layer for self-hosted runners.

Ask your Aviatrix account team for a guided deployment.

Aviatrix Validated Containment Architecture for Enterprise GitHub Pipelines 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.