Azure Network Security

Audience: {'primary': 'Cloud security architects and network engineers', 'secondary': ['Platform engineers', 'Auditors'], 'assumed_level': 'Intermediate-to-advanced Azure knowledge'} **Job ID:** c97bbf1c-a48b-4798-9638-c178008bd38e **Generated:** 2026-02-05

Table of Contents

1. Scope, Threat Model, and Reference Architectures (s01)

1.1 Objectives

- Define scope and non-goals for Azure network security guidance in this document.
- Establish a practical threat model for Azure networking.
- Introduce reference architectures used throughout, with control placement expectations.

1.2 When to use

- You should use this section to baseline your network security program before implementing controls such as Azure Firewall (Azure Firewall), Network Security Group (NSG) (Network Security Group (NSG)), Azure DDoS Protection Standard (Azure DDoS Protection Standard), and Private Endpoint (Private Endpoint).
- You should reference this section during architecture reviews, threat modeling workshops, landing zone decisions, and audit scoping.

1.3 Design decisions

- **Control boundaries**: Treat Azure as operating under a shared responsibility model; Microsoft secures the cloud platform, while you secure configuration, identity, and traffic policies for your tenant/subscriptions and workloads.
- **Two planes**:
  - **Control plane** (Control plane): Governed primarily by identity, RBAC, policy, and management network access.
  - **Data plane** (Data plane): Governed by routing, segmentation, L3/L4 filtering, L7 inspection, and workload endpoints.
- **Default reference topology**: Prefer **hub-spoke** (Hub-spoke) or **Azure Virtual WAN (vWAN)** (Azure Virtual WAN (vWAN)) for centralized inspection, consistent routing, and shared services.
- **Threat model categories** to explicitly cover in design reviews:
  - External attacks (north-south)
  - Lateral movement (east-west)
  - Insider misuse (privileged access)

    - Supply chain (CI/CD, images, dependencies, partner connectivity)
    - Misconfiguration (routes, NSGs, firewall policies, DNS, private endpoints)

Security rationale: Separating control plane from data plane decisions reduces blast radius and prevents “secure data plane, exposed management” failures (e.g., permissive management access paths, excessive RBAC, or ungoverned changes to routes/firewall rules).

1.4 Implementation notes

1.4.1 Scope (in-scope)

- **Network topology and routing**: Hub-spoke and vWAN patterns, route tables and UDR (User-Defined Route), Forced tunneling.
- **Segmentation and filtering**: NSG, ASG (Application Security Group), Azure Firewall, NVA patterns.
- **Perimeter controls**: WAF (WAF), DDoS, ingress/egress design.
- **Private access to PaaS**: Private Link, Private Endpoint, Azure Private DNS Zone, service networking approaches.
- **Hybrid connectivity**: ExpressRoute, VPN, BGP (BGP (Border Gateway Protocol)).
- **Operational telemetry**: Logging to Log Analytics / Microsoft Sentinel, flow and firewall logs, incident evidence.

1.4.2 Non-goals (out-of-scope, but adjacent)

- Application secure coding, SAST/DAST tooling selection, and detailed workload hardening baselines (covered elsewhere).
- Vendor-specific SD-WAN/NVA feature-by-feature evaluations (patterns are covered; product selection is contextual).
- Full compliance framework mapping (handled in the compliance/audit section later, using this section’s scoping as an input).

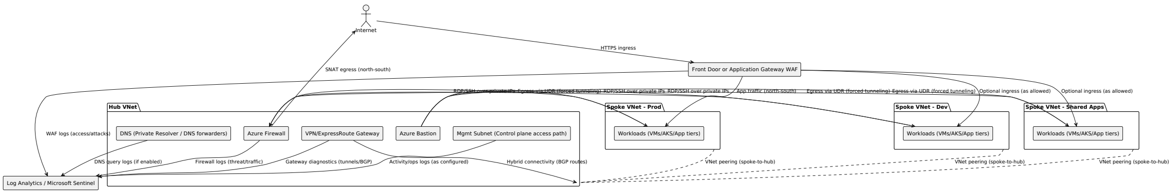
1.4.3 Reference architectures (used throughout)

- **Hub-spoke**: Centralizes shared services (DNS, egress inspection, hybrid gateways, management access) while isolating workloads in spokes.
- **vWAN**: Centralizes routing and connectivity via managed hubs; suitable for many branches/sites and multi-region connectivity standardization.

- **Landing zones**: Subscription management groups, policy baselines, and network resource organization patterns that enforce guardrails consistently.

  Anti-pattern: Building “flat VNets” with ad-hoc peering and inconsistent route tables. This commonly produces undocumented east-west paths, inconsistent egress inspection, and audit gaps in rule ownership.

1.4.4 Diagram: Hub-Spoke Reference Architecture with Centralized Inspection


Internet
HTTPS ingress
SNAT egress (north-south)
Front Door or Application Gateway WAF
RDP/SSH over private IPs
Egress via UDR (forced tunneling)
RDP/SSH over private IPs
RDP/SSH over private IPs
App traffic (north-south)
Egress via UDR (forced tunneling)
Egress via UDR (forced tunneling)
Optional ingress (as allowed)
Optional ingress (as allowed)
Hub VNet
Spoke VNet - Prod
Spoke VNet - Dev
Spoke VNet - Shared Apps
WAF logs (access/attacks)
DNS (Private Resolver / DNS forwarders)
Azure Firewall
VPN/ExpressRoute Gateway
Azure Bastion
Mgmt Subnet (Control plane access path)
Workloads (VMs/AKS/App tiers)
Workloads (VMs/AKS/App tiers)
Workloads (VMs/AKS/App tiers)
DNS query logs (if enabled)
Firewall logs (threat/traffic)
Gateway diagnostics (tunnels/BGP)
Activity/ops logs (as configured)
Hybrid connectivity (BGP routes)
VNet peering (spoke-to-hub)
VNet peering (spoke-to-hub)
VNet peering (spoke-to-hub)
Log Analytics / Microsoft Sentinel

1.5 Validation

You should validate scope and reference architecture decisions using:

1. **Architecture decision records (ADRs)** confirming topology choice (hub-spoke vs vWAN), inspection strategy, and plane separation.
2. **Threat modeling output** mapping assets to threats across:
   1. External (north-south)
   2. East-west lateral movement
   3. Insider/privileged misuse
   4. Supply chain paths
   5. Misconfiguration scenarios
3. **Connectivity verification**:
   - Spokes have deterministic egress (expected route next hop = firewall/NVA).
   - Ingress only via approved edge (WAF/Front Door) and not via direct public IPs on workloads (unless explicitly scoped).
4. **Telemetry verification**:
   - Logging destinations exist and are immutable enough for your audit posture (workspace retention/locks as applicable).

1.6 Pitfalls

- Treating NSGs as a complete perimeter strategy (they are L3/L4 stateful filters, not centralized policy or full egress control).

- Allowing management access from broad networks instead of a dedicated management path (e.g., Azure Bastion or controlled jump hosts).
- Omitting DNS and private name resolution planning, which breaks Private Link adoption later.
- Designing without explicit ownership: who can change routes, firewall policy, NSG rules, and private endpoints.

  Security rationale: Many Azure network incidents are governance and routing failures (unexpected egress paths, unmanaged peerings, permissive default routes) rather than “missing a tool.” Clear ownership and deterministic routing are primary risk reducers.

1.7 Audit evidence

You should retain the following artifacts for audit and internal assurance:

- Scope statement: in-scope resource types, subscriptions, VNets, regions, and connectivity boundaries.
- Threat model documentation: assumptions, threat categories, and mitigations mapped to controls.
- Network diagrams with control placement (hub-spoke/vWAN) and traffic flows (ingress, egress, hybrid).
- Policy and guardrail evidence: Azure Policy assignments and exemptions (if used), plus change control records.
- Logging evidence: diagnostic settings enabled for firewall/WAF/gateways, workspace retention settings, Sentinel data connectors (if applicable), and sample logs.

1.8 Controls checklist

- [ ] Scope and non-goals documented and approved (architecture governance).
- [ ] Threat model completed covering external, east-west, insider, supply chain, and misconfiguration.
- [ ] Reference topology selected (hub-spoke or vWAN) with inspection and routing principles defined.
- [ ] Control plane access path defined (e.g., Bastion/mgmt subnet) and separated from workload data plane.
- [ ] Central logging defined (Log Analytics/Sentinel) with required diagnostic settings in scope.

2. Azure Networking Security Fundamentals

2.1 Objectives

- Refresh the core Azure networking building blocks through a security lens.
- Clarify **data plane** vs **control plane** implications for network security.
- Establish baseline practices for segmentation, routing intent, private service access, and secure administration consistent with hub-spoke patterns.

2.2 When to use

- You are designing or reviewing an Azure landing zone networking baseline (hub-spoke) and need security-oriented defaults.
- You are preparing for an audit where you must evidence routing intent, segmentation enforcement, and private access posture.
- You are troubleshooting suspected misconfiguration issues (e.g., unintended Internet egress, public endpoint fallback, overly permissive rules).

2.3 Design decisions

- **Data plane focus**: Prioritize controls that directly affect runtime traffic (e.g., Network Security Group (NSG), UDR (User-Defined Route), Azure Firewall, Private Endpoint); include **control plane** governance (e.g., Azure Policy) only to prevent or detect data plane drift.
- **Reference architecture**: Use **hub-spoke** where the hub centralizes shared security services (firewall, DNS, bastion) and spokes isolate workloads.
- **Threat-model-driven baseline**:
    - External: minimize public exposure; enforce WAF where applicable.
    - East-west traffic: segmentation and least privilege; avoid flat networks.
    - Insider: restrict who can change routing/NSGs/firewall policy; evidence change control.
    - Supply chain: treat NVAs/images as sensitive; validate provenance where used.
    - Misconfiguration: prevent “allow any/any”, unintended UDR bypass, and DNS public fallback.

Security rationale: Treat routing, DNS, and segmentation as first-class security controls. In Azure, a single mis-scoped NSG rule, UDR, or DNS configuration can create unintended exposure or exfiltration paths without any code change.

2.4 Implementation notes

2.4.1 Data plane vs control plane (security relevance)

- **Data plane**: Packet flows between clients, workloads, and services (north-south and east-west). Controls include:
  - Subnet boundaries, NSG rules, ASG (Application Security Group) membership
  - UDR steering (e.g., forced tunneling to Azure Firewall)
  - Private connectivity to PaaS via Private Link and Azure Private DNS Zone
- **Control plane**: Management operations via Azure APIs/ARM.
  - Network-impacting control plane actions (create/modify NSG/UDR/peering/firewall policy) must be restricted and audited.
  - Use Azure Policy to prevent known-bad configurations (e.g., public IPs on sensitive subnets, permissive NSGs), and produce compliance evidence.

2.4.1.1 Procedure: classify a change request (decision tree)

1. Identify the change target:
   1. If it changes **NSG/UDR/Firewall/DNS resolution/private endpoints** → treat as **data plane-impacting**.
   2. If it only changes **RBAC/Policy/Locks** → treat as **control plane** (but may indirectly protect data plane).
2. Determine blast radius:
   1. Subnet/VNet-wide changes (subnet NSG, route table) → higher risk than NIC-scoped.
3. Require evidence:
   1. For data plane-impacting: effective routes, rule diffs, and traffic validation outputs (see **Validation** and **Audit evidence**).

Anti-pattern: Treating NSG/UDR/DNS changes as “routine” without pre/post validation. These changes frequently cause either silent exposure (breakout) or silent outage (DNS private resolution failure).

2.4.2 Core building blocks (security lens)

2.4.2.1 Azure Virtual Network (VNet) and subnets

- Use separate subnets for trust boundaries (e.g., web/app/data, shared services, management).
- Align subnets to enforcement points:
  - Subnet-level NSGs for consistent baseline controls.
  - UDRs per subnet to ensure egress steering (forced tunneling) where required.

2.4.2.2 Routing: system routes vs UDR

- **System routes** provide default intra-VNet, peering, and Internet egress behavior.
- **UDR** overrides system routes to enforce routing intent:
  - Forced tunneling: 0.0.0.0/0 → Azure Firewall (hub) as next hop for spokes.
  - Explicit routes for private ranges to avoid asymmetric routing in hybrid scenarios.

  Security rationale: UDRs are the primary mechanism to prevent unmanaged Internet egress and to centralize inspection, which reduces exfiltration paths and improves detection coverage.

2.4.2.3 Service Endpoint vs Private Link (high-level comparison)

- Service Endpoint:
  - Extends VNet identity to certain Azure services over the service's public endpoint.
  - Security posture depends on service-side ACLs and correct subnet scoping.
- Private Link with Private Endpoint:
  - Brings the service into your VNet via private IP.
  - Usually the preferred option for preventing public endpoint exposure and reducing public Internet dependency.

Tradeoffs (options):

- Private Link:
  - Pros: private IP, stronger "no public path" posture, better alignment with zero-trust segmentation.

    - Cons: requires Private DNS planning; can introduce DNS complexity across VNets.
- Service Endpoints:
    - Pros: simpler DNS; may be sufficient for constrained scenarios.
    - Cons: public endpoint still used; misconfigurations can allow public access.

Anti-pattern: Implementing Private Endpoints without Private DNS resolution, causing clients to resolve public records and “fail open” to public endpoints during outages or misconfiguration.

2.4.3 DNS resolution paths (Azure-provided, custom DNS, Private DNS zones)

- Use **Azure Private DNS Zone** for Private Endpoint name resolution.
- Decide and document DNS authority:
    - Azure-provided DNS is simplest within a single VNet but can be insufficient for multi-VNet private endpoint resolution patterns.
    - Custom DNS (e.g., in hub) may be required for centralized governance, conditional forwarding, and hybrid integration.

2.4.3.1 Procedure: implement Private DNS for Private Endpoints (baseline)

1. Create or reuse the appropriate **Azure Private DNS Zone** for the PaaS private link domain (assumption: standard Azure private link zones; adapt to your services).
2. Link the Private DNS Zone to:
    1. The spoke VNet(s) hosting clients.
    2. The hub VNet if you centralize DNS forwarders there.
3. Ensure the Private Endpoint NIC registers/has an A record in the zone (or create records explicitly where required).
4. Validate resolution from workload subnets using the expected FQDN (fully qualified domain name) and confirm it resolves to a private IP.

Security rationale: Correct private DNS resolution prevents accidental traversal to public endpoints, reducing exposure to data exfiltration and misrouting risks.

2.4.4 Identity and access in networking (RBAC, PIM, management groups, policy)

Conceptually (third-person): Control plane permissions determine who can change the enforcement points (NSG/UDR/firewall policy). This is a common insider and misconfiguration risk.

Procedurally (second-person), you should:

1. Separate roles:
   1. Network operators (routing/peering).
   2. Security operators (firewall policy/WAF/NSG baseline).
   3. Workload owners (limited to their spokes where appropriate).
2. Use least privilege RBAC on:
   1. Route tables (UDR)
   2. NSGs/ASGs
   3. Azure Firewall and policy objects
   4. Private DNS zones and Private Endpoints
3. Use PIM (Privileged Identity Management) where applicable for just-in-time elevation (assumption: your tenant has PIM enabled; adapt to your governance model).
4. Apply **Azure Policy** guardrails at management group/subscription scope to:
   1. Deny public IP creation on protected subnets (where applicable).
   2. Deny permissive NSG patterns (e.g., inbound Any from Internet to management ports).
   3. Audit missing flow logs / diagnostics (where supported).

   Anti-pattern: Granting broad “Network Contributor” at subscription scope to workload teams, enabling them to bypass forced tunneling or open inbound exposure without centralized review.

2.4.5 Encryption basics (TLS, IPsec, MACsec) and certificate lifecycle (network-relevant)

- **TLS**: Primary for application-layer encryption. Network controls (WAF, firewall) may observe metadata without decryption unless **TLS inspection** is explicitly configured (typically Azure Firewall Premium).
- **IPsec**: Used for site-to-site VPN and some NVA scenarios; relevant for hybrid data plane confidentiality/integrity.
- **MACsec**: Where applicable (typically service-provider/express connectivity contexts), provides L2 encryption; treat as hybrid design-specific.

Operationally, you should:

1. Document where decryption occurs (if at all), and the privacy/compliance basis.
2. Manage certificates (issuance, rotation, revocation) for:
    1. TLS termination points (Front Door/WAF, Application Gateway WAF, internal ingress)
    2. TLS inspection (if used), including trust distribution to clients

    Security rationale: Encryption design is part of the routing and inspection story; inspection points without clear certificate governance create outage and compliance risks.

2.5 Validation

You should validate baseline networking security with repeatable checks:

1. **Resource inventory**
    1. Enumerate VNets, subnets, NSGs, ASGs, route tables, peerings, Azure Firewall instances/policies, Private Endpoints, and Private DNS zone links.
2. **Effective route checks (forced tunneling/no breakout)**
    1. For representative NICs in spoke subnets, verify effective routes send 0.0.0.0/0 to the hub inspection/egress point (where forced tunneling is required).
3. **Ingress/egress tracing**
    1. Validate north-south ingress path (e.g., WAF → workload) and ensure backend exposure is private as intended.
    2. Validate spoke egress path (spoke → firewall → Internet) including SNAT behavior assumptions.
4. **DNS resolution validation**
    1. From workload subnets, resolve PaaS FQDNs to private IPs where Private Endpoints exist; confirm there is no public fallback.
5. **Logging coverage**
    1. Confirm NSG flow logs (where enabled), firewall logs, and relevant diagnostics stream to Log Analytics/Sentinel with retention aligned to audit requirements.

2.6 Pitfalls

- Overlapping address spaces across VNets that later block peering or force NAT workarounds, weakening segmentation intent.
- Permissive NSG defaults (“allow all within VNet”) applied broadly without ASG-based microsegmentation.
- UDR misapplication causing asymmetric routing, intermittent connectivity, or unintended Internet breakout.
- Private Endpoint created without Private DNS zone link to all client VNets, causing resolution failures or public endpoint usage.
- Hub overloading (too many shared services without scale planning), creating a single choke point without capacity controls (including SNAT planning where applicable).

2.7 Audit evidence

You should retain and be able to produce:

1. **Architecture and intent**
    1. Hub-spoke diagram(s) and address plan
    2. Routing intent statement (forced tunneling expectations, exceptions)
2. **Configuration exports**
    1. NSG rules (including ASG references)
    2. Route tables (UDR) and association to subnets
    3. VNet peering settings (allow forwarded traffic, gateway transit, etc.)
    4. Azure Firewall policy/rules (if deployed)
    5. Private Endpoint list and Private DNS zone links/records
3. **Governance artifacts**
    1. Azure Policy assignments and compliance results related to networking security
    2. RBAC role assignments for network/security resource scopes (with change history where available)
4. **Operational validation records**
    1. Effective route screenshots/exports for representative subnets
    2. DNS resolution test outputs
    3. Logging/retention settings and sample queries/alerts mapped to threats (external, east-west, insider, misconfiguration)

3. s03 — Network Segmentation Strategy

3.1 Objectives

- Define segmentation models suitable for Azure networking **data plane** controls.
- Provide patterns for isolating environments, applications, and tiers using **Azure Virtual Network (VNet)**, subnets, **Network Security Group (NSG)**, and **ASG (Application Security Group)**.
- Establish decision points for centralized vs distributed east-west enforcement and inspection.

3.2 When to use

- You are designing or refactoring a landing zone network with multiple workloads, environments (dev/test/prod), or compliance boundaries.
- You need to reduce blast radius for:
    - North-south threats (Internet-to-app and app-to-Internet).
    - East-west lateral movement between workloads.
    - Insider and misconfiguration risks (overly-permissive rules, route leakage).
- You are introducing **Private Endpoint** / **Private Link** and must prevent public endpoint fallback through segmentation and DNS controls.

3.3 Design decisions

- **Primary segmentation dimensions (choose explicitly and document):**
    - Environment (dev/test/prod) as a hard boundary where required.
    - Workload/application boundary (app A vs app B).
    - Tier boundary (ingress/app/data/management).
    - Tenant boundary (multi-tenant isolation).
    - Compliance boundary (regulated vs non-regulated workloads).
- **Macrosegmentation approach (topology):**
    - Hub-spoke baseline: shared controls in hub, workloads in spokes; controlled peering; optional forced tunneling to **Azure Firewall**.
- **Microsegmentation approach (filtering model):**
    - Subnet-level NSGs as baseline.

  - ASG-based NSG rules for workload identity grouping (avoid IP brittleness).
- **East-west inspection posture (tradeoffs):**
  - Distributed enforcement: NSGs per subnet/ASG; simpler latency profile; weaker centralized visibility.
  - Centralized inspection: steer traffic through **Azure Firewall** via **UDR (User-Defined Route)**; better consistency/visibility; higher routing complexity and SNAT planning.

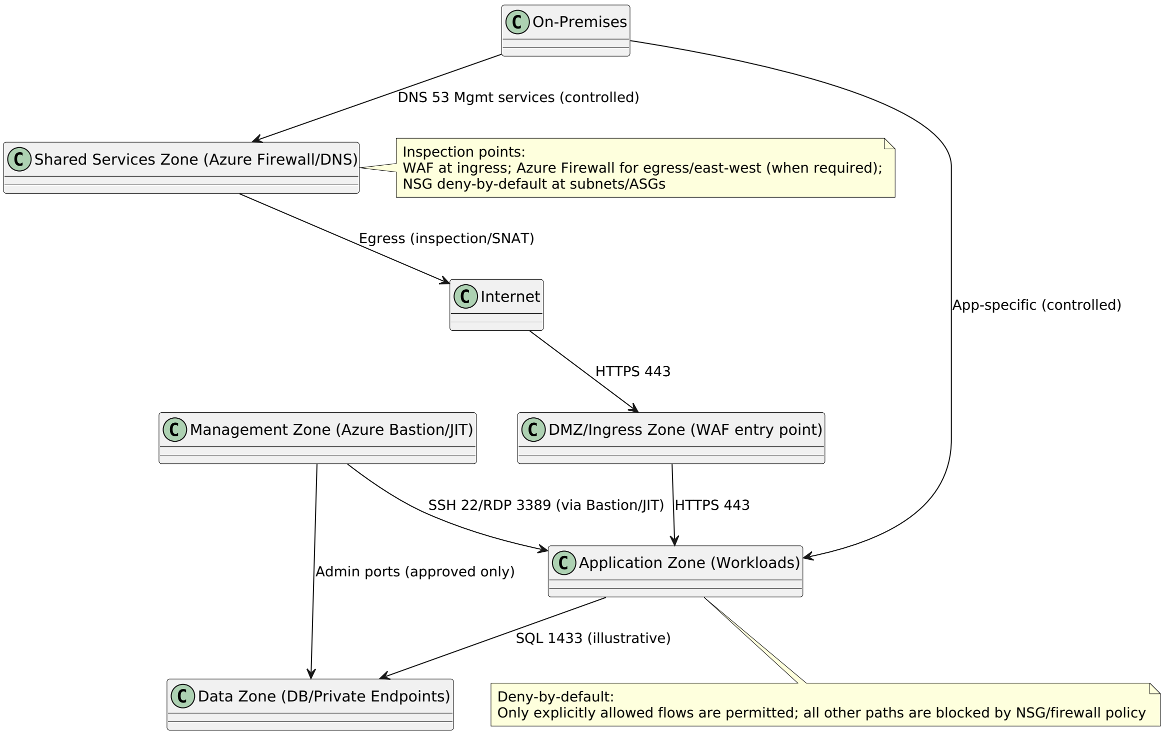
> [!SECURITY RATIONALE] You should define segmentation around likely lateral movement paths (east-west) and misconfiguration risk, not only around IP address convenience. Threat actors routinely pivot across flat networks; the primary goal of segmentation is to constrain reachable attack surfaces and privileged management paths.

3.4 Implementation notes

3.4.1 Segmentation zone model (reference)

Use a consistent zone taxonomy and map it to VNets/subnets and controls:

- **Internet**: untrusted source.
- **DMZ/Ingress zone**: HTTP(S) ingress components (e.g., WAF entry point, application gateway tier).
- **Application zone**: app services/VMs/AKS worker subnets (as applicable).
- **Data zone**: databases, caches, storage access subnets (prefer **Private Endpoint**).
- **Management zone**: admin entry points (e.g., **Azure Bastion**, jump tooling).
- **Shared Services zone**: DNS, update repos, central logging forwarders, hub firewall.
- **On-Premises**: corporate network over VPN/ExpressRoute (not a trust boundary; treat as “controlled external”).


On-Premises
DNS 53 Mgmt services (controlled)
Shared Services Zone (Azure Firewall/DNS)
Inspection points:
WAF at ingress; Azure Firewall for egress/east-west (when required);
NSG deny-by-default at subnets/ASGs
Egress (inspection/SNAT)
Internet
App-specific (controlled)
HTTPS 443
Management Zone (Azure Bastion/JIT)
DMZ/Ingress Zone (WAF entry point)
SSH 22/RDP 3389 (via Bastion/JIT)
HTTPS 443
Admin ports (approved only)
Application Zone (Workloads)
SQL 1433 (illustrative)
Data Zone (DB/Private Endpoints)
Deny-by-default:
Only explicitly allowed flows are permitted; all other paths are blocked by NSG/firewall policy

3.4.2 Subnet and address planning (non-overlapping CIDRs)

1. You should allocate CIDR blocks per environment and per region to avoid overlap (required for future peering and M&A scenarios).
2. You should allocate subnets per zone/tier with growth buffer (avoid frequent subnet resizing).
3. You should reserve dedicated subnets for:
    - Azure Firewall (AzureFirewallSubnet / AzureFirewallManagementSubnet where applicable).
    - Azure Bastion (AzureBastionSubnet).
    - Private Endpoint subnets (if you use subnet policies/UDR patterns that differ from workload subnets).

Assumptions (adapt as required):

- Hub VNet has shared services and security controls.
- Each spoke VNet maps to a workload or bounded group of workloads.
- No overlapping RFC1918 ranges across peered VNets.

> [!ANTI-PATTERN] Reusing the same CIDR ranges across dev/test/prod because “they will never connect” frequently breaks later when you need peering, Private Link resolution, or centralized inspection.

3.4.3 NSG/ASG microsegmentation baseline (deny-by-default)

You should implement NSGs with:

- Explicit allow rules for required flows between zones/tier subnets and ASGs.
- Explicit outbound control (do not rely on default outbound allow for sensitive subnets).
- Optional explicit denies for high-risk lateral paths (e.g., dev-to-prod, app-to-management).

Procedure (illustrative):

1. You should define ASGs per role (e.g., `asg-app-web`, `asg-app-api`, `asg-data-sql`) and assign NICs accordingly.
2. You should attach NSGs to subnets (preferred for consistency) and use ASG references in rules.
3. You should author rules by traffic intent:
    1. Ingress: DMZ to App (HTTPS 443).
    2. App to Data (DB ports only).
    3. Management to targets (via Bastion/JIT; constrain sources).
    4. Egress: App to approved dependencies (prefer via firewall/proxy patterns).
4. You should validate **effective security rules** on representative NICs for each tier.

    [!SECURITY RATIONALE] ASG-based rules reduce operational drift and “IP sprawl” errors, which are a leading cause of accidental exposure and unintended lateral reachability.

3.4.4 Macrosegmentation with hub-spoke and centralized inspection (when required)

If you require centralized egress control or east-west inspection:

- You should use forced tunneling from spoke subnets with a `0.0.0.0/0` UDR next hop to the hub firewall private IP.
- You should selectively steer east-west flows through Azure Firewall only where required (e.g., between regulated and non-regulated spokes).

Procedure (forced tunneling, illustrative):

1. You should create/identify the hub firewall (Azure Firewall) in the hub VNet.

2. You should create a route table per spoke (or per subnet class) and add:
    - `0.0.0.0/0` → next hop `Virtual appliance` → `<Azure Firewall private IP>`
3. You should associate the route table to the relevant spoke subnets.
4. You should validate effective routes on NICs and validate firewall SNAT capacity assumptions (port utilization) for peak egress.

    [!SECURITY RATIONALE] Centralized egress reduces data exfiltration paths, enforces consistent FQDN/URL controls (where used), and improves auditability of outbound dependencies—at the cost of routing complexity and throughput/SNAT planning.

3.5 Validation

- **Reachability tests (data plane):**
    - Verify intended flows succeed (e.g., DMZ→App 443, App→Data 1433).
    - Verify unintended flows fail (e.g., App→Mgmt, Dev→Prod, Spoke→Spoke where not allowed).
- **Azure checks:**
    - Review **effective security rules** on NICs for each tier.
    - Review **effective routes** on NICs where UDRs/forced tunneling are implemented.
- **Logging readiness (for later monitoring sections):**
    - Ensure NSG Flow Logs / firewall logs are enabled where mandated by policy (record in audit evidence).

3.6 Pitfalls

- Allowing broad intra-VNet or intra-subnet traffic (“flat” network) and assuming VNet isolation equals security filtering.
- Using IP-based rules instead of ASGs, resulting in brittle rule maintenance and rule sprawl.
- Forcing all east-west traffic through centralized inspection without capacity planning (latency/throughput/SNAT exhaustion).
- Mixing management access paths with workload subnets (no dedicated management zone controls).

3.7 Audit evidence

You should retain evidence that demonstrates intent, enforcement, and change control:

- Network segmentation design document (zones, boundaries, allowed flows matrix).
- Inventory exports:
  - VNets, subnets, peerings.
  - NSGs and rule sets (including priorities) and ASG membership mapping.
  - Route tables (UDRs) and subnet associations.
  - Azure Firewall policy/rules (if used for inspection/egress).
- Validation artifacts:
  - Screenshots or exports of effective routes/effective security rules for sample NICs per tier.
  - Test records showing allowed/blocked flows (timestamped).

3.7.1 Controls checklist (s03)

- [ ] Zones and segmentation dimensions are explicitly defined and mapped to VNets/subnets.
- [ ] CIDR plan avoids overlap across environments/regions and supports future peering.
- [ ] NSGs enforce deny-by-default with explicit allows; outbound is controlled for sensitive tiers.
- [ ] ASGs are used for workload grouping to avoid IP-based brittleness.
- [ ] Forced tunneling (if used) is validated with effective routes and SNAT capacity planning.
- [ ] Evidence collection (configs + validation artifacts) is captured for audit and change reviews.

4. Perimeter and Ingress/Egress Security (s04)

4.1 Objectives

- Harden north-south traffic paths (Internet/on-premises to Azure workloads).

- Establish secure egress controls and outbound governance with auditable decision points.
- Reduce misconfiguration risk through standardized patterns (hub-spoke, forced tunneling, centralized policy).

4.2 When to use

- You expose any HTTP/HTTPS workload to the Internet (API, web apps, AKS ingress).
- You require centralized outbound control (malware callback prevention, data exfiltration controls, regulatory egress allowlists).
- You operate hub-spoke networks and need consistent routing intent enforcement via UDR (User-Defined Route).
- You must evidence perimeter controls and monitoring for audits (logging, rule hygiene, change control).

4.3 Design decisions

- **Ingress edge choice (L7):** Azure Front Door (global) vs Application Gateway WAF (regional/VNet).
- **Perimeter stateful firewall (L3-L7):** Azure Firewall with Firewall Policy vs NVA (only when required features exceed native).
- **DDoS posture:** enable Azure DDoS Protection Standard on public IP–exposed VNets/resources.
- **Egress governance model:** forced tunneling through Azure Firewall vs direct internet egress (exception-driven).
- **SNAT strategy:** Azure Firewall SNAT vs NAT Gateway (capacity, cost, and scaling considerations).
- **TLS inspection:** none vs selective vs broad (privacy, certificate governance, and operational maturity).

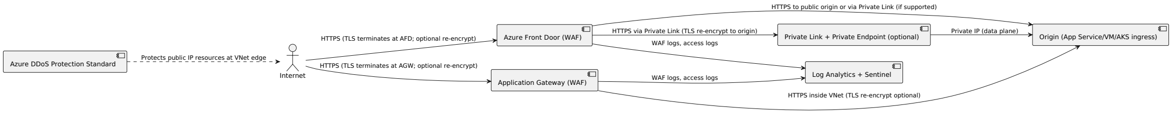
  Security rationale: You reduce **external (north-south)** exposure by terminating/inspecting at managed edges (WAF) and centralized firewalls, and reduce **misconfiguration** by standardizing ingress/egress chokepoints with consistent logging.

4.4 Implementation notes

4.4.1 Ingress: WAF patterns (Front Door vs Application Gateway)

Use the following decision points to select an ingress model:

1. If you need **global anycast entry**, global failover, and edge caching/CDN features, you should prefer **Azure Front Door (WAF)**.
2. If you need **regional/VNet-close L7 control**, private backends without global edge requirements, or tight integration with VNet routing, you should prefer **Application Gateway (WAF)**.
3. If you require **private origin** access from the edge, you should use **Private Link/Private Endpoint** patterns where supported (and validate private DNS to prevent public fallback).
4. If you need **end-to-end TLS**, you should terminate TLS at WAF **and** re-encrypt to the origin (separate cert lifecycle and SNI/hostname validation).


Azure DDoS Protection Standard
Protects public IP resources at VNet edge
Internet
HTTPS (TLS terminates at AFD; optional re-encrypt)
HTTPS (TLS terminates at AGW; optional re-encrypt)
Azure Front Door (WAF)
Application Gateway (WAF)
HTTPS to public origin or via Private Link (if supported)
HTTPS via Private Link (TLS re-encrypt to origin)
WAF logs, access logs
Private Link + Private Endpoint (optional)
Private IP (data plane)
Origin (App Service/VM/AKS ingress)
WAF logs, access logs
Log Analytics + Sentinel
HTTPS inside VNet (TLS re-encrypt optional)

Operational notes (illustrative; adapt to your environment):

- You should enable WAF in **Prevention** mode for mature apps; use **Detection** during initial tuning with a defined promotion window.
- You should standardize: allowed methods, request body limits, file upload constraints, and bot protections (as applicable).
- You should log WAF events and access logs to **Log Analytics** and forward to **Microsoft Sentinel** for correlation.

  Anti-pattern: Exposing origins publicly “for convenience” while also placing WAF in front, without enforcing origin restrictions (e.g., allowing direct Internet access to the backend). This creates a bypass path and weakens your north-south control.

4.4.2 DDoS Protection Standard placement and practices

1. You should enable **Azure DDoS Protection Standard** on the **VNet** that contains Internet-facing public IP resources (e.g., Application Gateway, Azure Firewall public IPs, public load balancers).
2. You should integrate DDoS telemetry with your SIEM and define an incident runbook (contacts, escalation, evidence collection).

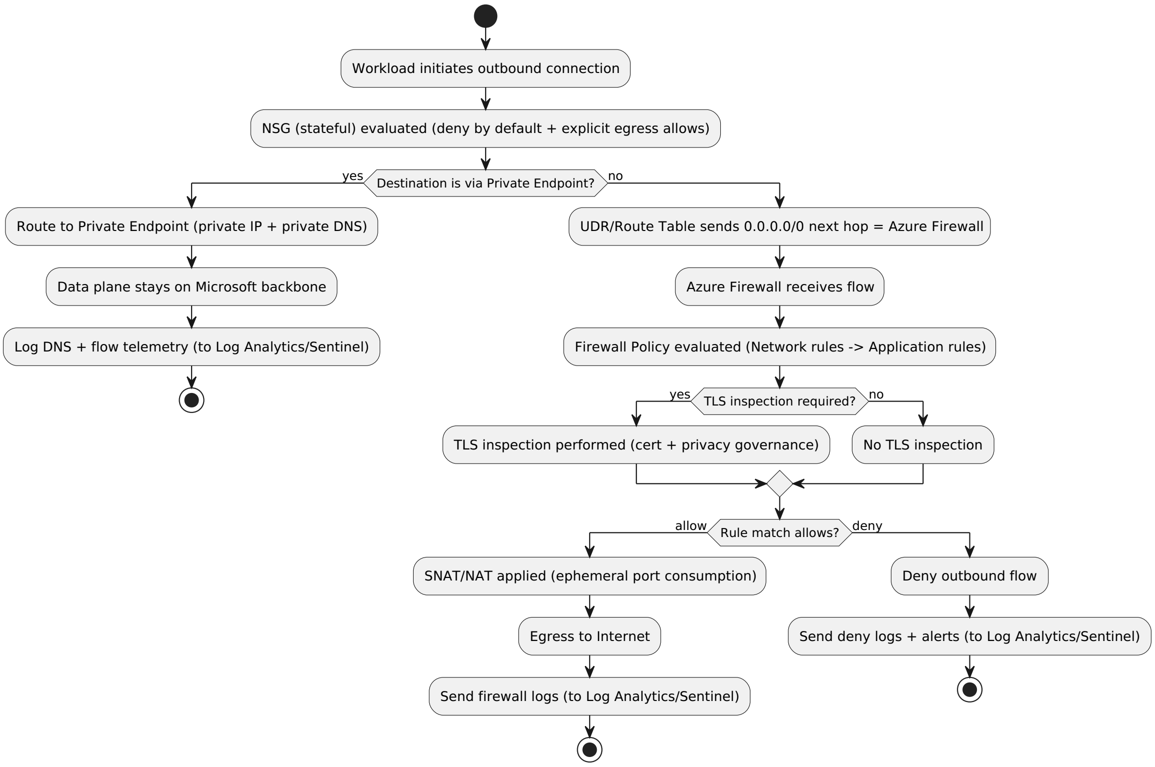
3. You should validate that protected resources are actually in the protected VNet(s) and that teams do not deploy unmanaged public IPs outside the protected scope.

   Security rationale: DDoS Protection Standard provides adaptive tuning and attack telemetry, reducing risk from **external** volumetric attacks and improving auditability via centralized analytics.

4.4.3 Egress: forced tunneling through Azure Firewall with policy layers

Baseline pattern (hub-spoke):

- Spoke subnets route `0.0.0.0/0` to the hub firewall using a UDR (User-Defined Route).
- Azure Firewall enforces policy (application/network rules; optional TLS inspection) and performs SNAT for Internet-bound flows.
- Approved Azure PaaS access should prefer **Private Endpoint** (exception path that does not traverse the Internet).


Workload initiates outbound connection
NSG (stateful) evaluated (deny by default + explicit egress allows)
yes
Destination is via Private Endpoint?
no
Route to Private Endpoint (private IP + private DNS)
Data plane stays on Microsoft backbone
Log DNS + flow telemetry (to Log Analytics/Sentinel)
UDR/Route Table sends 0.0.0.0/0 next hop = Azure Firewall
Azure Firewall receives flow
Firewall Policy evaluated (Network rules -> Application rules)
yes
TLS inspection required?
no
TLS inspection performed (cert + privacy governance)
No TLS inspection
allow
Rule match allows?
deny
SNAT/NAT applied (ephemeral port consumption)
Egress to Internet
Send firewall logs (to Log Analytics/Sentinel)
Deny outbound flow
Send deny logs + alerts (to Log Analytics/Sentinel)

Implementation notes (illustrative; adapt):

1. You should create a **hub Azure Firewall** and attach a **Firewall Policy** with:
    - Application rules using FQDN-based controls (including built-in “FQDN tags” where appropriate).
    - Network rules for non-HTTP/S protocols that must egress.
    - Explicit deny rules and logging for blocked traffic.
2. You should apply forced tunneling by attaching a **route table** to spoke subnets with:
    - `0.0.0.0/0` next hop = Azure Firewall private IP.
    - More specific routes for private address spaces as required (to avoid hairpinning internal traffic).
3. You should define an exception process for **Private Endpoint** adoption:
    - Prefer Private Endpoint for Azure PaaS to reduce Internet exposure.
    - Validate private DNS resolution returns the private endpoint IP to prevent public endpoint fallback.
4. You should plan SNAT capacity:
    - Estimate concurrent outbound connections per workload.
    - Monitor SNAT port utilization where available and design for scale-out (additional firewall IPs, architecture changes, or NAT Gateway where appropriate).

Security rationale: Forced tunneling reduces **insider** and **supply chain** egress risk by creating a single, auditable enforcement point, and reduces **misconfiguration** by preventing direct-to-Internet “shadow egress.”

4.4.4 NAT Gateway vs Azure Firewall SNAT (scaling considerations)

- **Azure Firewall SNAT**
    - Pros: integrated with policy enforcement; single choke point for inspection.
    - Cons: SNAT port exhaustion risk if very high connection churn; requires capacity planning.
- **NAT Gateway**
    - Pros: highly scalable outbound SNAT for subnets; predictable egress IPs.
    - Cons: does not provide L7 policy enforcement; typically complements (not replaces) firewall-based governance.

Anti-pattern: Using NAT Gateway to “solve” SNAT exhaustion while allowing workloads to bypass Azure Firewall. This improves connectivity but degrades outbound governance and auditability.

4.4.5 TLS inspection considerations

If you enable TLS inspection:

- You should scope it to high-risk egress categories first (e.g., unknown destinations), with explicit exclusions for privacy-sensitive endpoints.
- You should implement certificate lifecycle governance (issuance, rotation, compromise handling).
- You should document legal/privacy basis and retain evidence of approvals.

4.5 Validation

You should validate both **routing intent** and **policy enforcement**:

1. **Ingress**
    1. Confirm WAF is in the expected mode (Detection/Prevention) and ruleset versions are controlled.
    2. Confirm origins are not directly reachable from the Internet (where required), and origin access restrictions are enforced.
    3. Confirm logs are arriving in Log Analytics/Sentinel and include WAF actions (allow/block) and relevant dimensions (client IP, rule ID).
2. **Egress**
    1. Confirm effective routes on representative NICs/subnets show `0.0.0.0/0` via Azure Firewall (forced tunneling).
    2. Confirm firewall rule hit counts align with expected application behavior.
    3. Confirm deny events are logged and alerting is configured for anomalous spikes.
    4. Confirm Private Endpoint name resolution returns private IPs from the correct Azure Private DNS Zone.

4.6 Pitfalls

- Overly permissive outbound rules (e.g., “Allow Internet any/any”) that defeat forced tunneling’s purpose.
- Relying on VNet isolation alone (a Azure Virtual Network (VNet) is not a firewall).

- Creating bypass paths:
    - Public origin exposure behind WAF.
    - Spoke subnets without the egress UDR.
    - Private Endpoint deployed without DNS linkage, causing public endpoint fallback.
- Inadequate SNAT planning leading to intermittent outbound failures under load (appears as random timeouts).
- Enabling TLS inspection without certificate, privacy, and exception governance.

4.7 Audit evidence

You should retain evidence that supports **design intent**, **enforcement**, and **monitoring**:

- Architecture diagrams showing ingress/egress chokepoints (hub-spoke, forced tunneling, WAF placement).
- Azure Firewall Policy exports (rules, rule collections, change history) and justification for broad allows/exceptions.
- Route table (UDR) configuration and “effective routes” captures for sampled workloads.
- WAF policy configuration (mode, managed rules, exclusions) and evidence of tuning approvals.
- DDoS Protection Standard enablement scope and attack analytics (if applicable).
- Centralized logging configuration:
    - Diagnostic settings for WAF/Azure Firewall to Log Analytics.
    - Sentinel data connectors, analytics rules, and retention settings.
- Change control records for perimeter rule changes (who/what/when/why), aligned to your governance process.

4.7.1 Controls checklist (s04)

- [ ] WAF deployed on the selected ingress edge (Front Door or Application Gateway) with defined tuning/promotion process.
- [ ] Origins protected against direct access (no bypass path) consistent with the chosen pattern.
- [ ] Azure DDoS Protection Standard enabled for VNets hosting Internet-facing public IP resources.
- [ ] Forced tunneling implemented via UDR for spoke subnets; exceptions are explicit and documented.

- [ ] Azure Firewall uses Firewall Policy with deny-by-default posture and logged allow/deny decisions.
- [ ] SNAT capacity planning performed; monitoring/alerts exist for egress failures and anomalous denies.
- [ ] Private Endpoint exception path validated with private DNS resolution and logging to Log Analytics/Sentinel.

5. s05 — Private Access to PaaS: Private Link, Service Endpoints, and Service Networking

5.1 Objectives

- You should select and implement private access patterns to Azure PaaS that minimize public endpoint exposure.
- You should prevent data exfiltration via unintended public paths and DNS fallback.
- You should implement repeatable DNS, approval, and logging patterns for Private Link at scale.

5.2 When to use

- Use **Private Endpoint** (Private Link) when you require:
  - Private IP connectivity to PaaS over the Microsoft backbone, with the ability to **disable public network access** on the PaaS resource.
  - Stronger data exfiltration resistance (no reliance on public endpoint routing semantics).
  - Consistent access across **hub-spoke** and hybrid networks with controlled DNS.
- Use **Service Endpoint** when you require:
  - Simpler enablement for supported services and accept that traffic targets the **public endpoint** with service-side ACL scoping.
  - No per-resource private IP mapping (i.e., you do not need Private Link semantics).

- Use a **service networking** pattern (e.g., shared DNS resolvers, centralized firewall egress, and standardized approvals) when:
    - You operate many spokes/subscriptions and need centralized governance and consistent resolution.

5.3 Design decisions

- **Private Link vs Service Endpoint**
    - **Private Endpoint** tradeoffs:
        - Pros: private IP, supports disabling public access, reduces public dependency, supports cross-VNet/hybrid with DNS.
        - Cons: requires correct **Azure Private DNS Zone** design/linking; introduces approval/governance overhead; per-endpoint IP management.
    - **Service Endpoint** tradeoffs:
        - Pros: simpler DNS (public name), subnet-based ACLs, low operational overhead.
        - Cons: still uses public endpoint semantics; harder to prove “no public exposure” in audits; weaker against DNS/route misconfigurations.
- **DNS authority and split-horizon**
    - You should decide whether name resolution is anchored in:
        - **Azure DNS Private Resolver** (preferred for hybrid-capable standardization), or
        - Custom DNS forwarders/appliances in the hub.
    - You should standardize on split-horizon DNS where the same PaaS FQDN resolves to:
        - Private endpoint IP inside approved VNets, and
        - Public IP outside (or no resolution if you intentionally block).
- **Approval and ownership model**
    - You should define who can create private endpoints, who can approve them on the PaaS side, and how exceptions are handled (cross-subscription/tenant).
- **Exfiltration posture**
    - You should decide whether to enforce “private-only” by:
        - Disabling public network access on PaaS where supported, and
        - Enforcing forced tunneling egress via **Azure Firewall** to reduce direct Internet reachability from spokes.

Security rationale: Private Endpoint + correct private DNS + disabled public access reduces misconfiguration and insider-driven exfiltration paths by removing reliance on public endpoints and ensuring traffic stays on intended private routes.

5.4 Implementation notes

5.4.1 1) Private Endpoint and Private DNS Zone baseline

Assumptions (adapt as required):

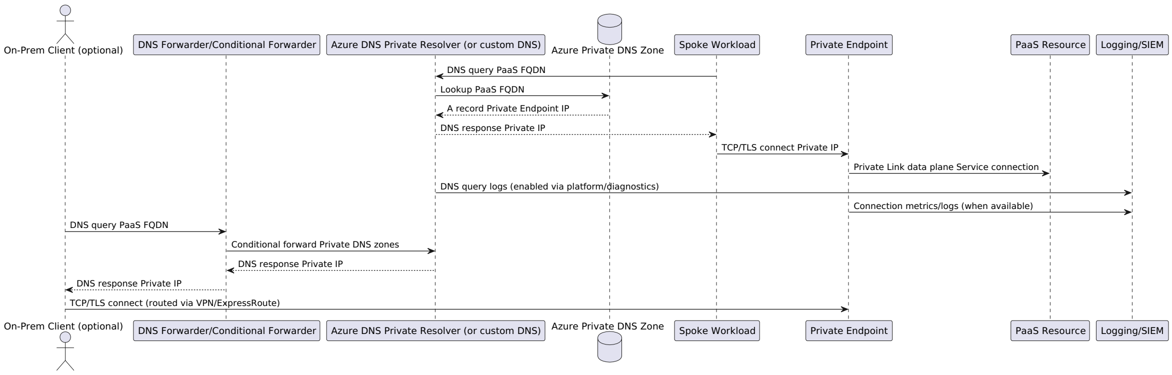
- Hub-spoke topology with centralized DNS (Azure DNS Private Resolver or custom DNS) in the hub.
- Spokes are peered to hub; workloads initiate PaaS connections (data plane).
- You will use **Azure-native logging** (e.g., Diagnostic settings to Log Analytics/Sentinel) where available.

Procedure (recommended baseline):

1. You should create a **Private Endpoint** in the client VNet (or a dedicated “endpoints” subnet) for the target PaaS resource.
2. You should create or reuse the correct **Azure Private DNS Zone** for the service (service-specific zone name).
3. You should link the private DNS zone to:
    1. All spoke VNets that contain clients that must resolve the private endpoint, and
    2. The hub VNet if it hosts DNS forwarders/resolvers.
4. You should ensure DNS forwarders (if used) conditionally forward the relevant private DNS zones to Azure DNS Private Resolver (or the authoritative resolver) to support hybrid.
5. You should disable public network access on the PaaS resource (where supported) and remove broad IP-based firewall exceptions.
6. You should validate DNS resolution and connectivity from each consuming subnet, and validate there is no public fallback.

    Anti-pattern: Creating private endpoints without linking the corresponding Azure Private DNS Zone to all client VNets, causing intermittent failures or silent public endpoint fallback when public access remains enabled.

5.4.2 2) Hybrid-capable DNS resolution sequence (common failure mode)


On-Prem Client (optional)
DNS Forwarder/Conditional Forwarder
Azure DNS Private Resolver (or custom DNS)
Azure Private DNS Zone
Spoke Workload
Private Endpoint
PaaS Resource
Logging/SIEM
DNS query PaaS FQDN
Lookup PaaS FQDN
A record Private Endpoint IP
DNS response Private IP
TCP/TLS connect Private IP
Private Link data plane Service connection
DNS query logs (enabled via platform/diagnostics)
Connection metrics/logs (when available)
DNS query PaaS FQDN
Conditional forward Private DNS zones
DNS response Private IP
DNS response Private IP
TCP/TLS connect (routed via VPN/ExpressRoute)
On-Prem Client (optional)
DNS Forwarder/Conditional Forwarder
Azure DNS Private Resolver (or custom DNS)
Azure Private DNS Zone
Spoke Workload
Private Endpoint
PaaS Resource
Logging/SIEM

5.4.3 3) Service Endpoints (when acceptable)

Procedure (high-level):

1. You should enable **Service Endpoint** on the specific subnet(s) requiring access.
2. You should configure the PaaS resource firewall to allow only the authorized VNet/subnet.
3. You should keep forced tunneling and egress controls aligned with the expectation that the destination remains the public endpoint.
4. You should validate that name resolution stays public and that access is denied from unauthorized subnets.

   Security rationale: Service Endpoints reduce exposure compared to “allow all networks,” but they do not remove public endpoint semantics; use them when operational simplicity outweighs the stronger isolation of Private Link.

5.4.4 4) Hardening PaaS for “private-only” access

- You should disable **public network access** where supported.
- You should scope access using:
  - Private Endpoint approvals, and
  - Resource firewall rules only as needed (avoid broad IP allowlists).
- You should ensure outbound Internet access from clients is inspected/controlled (e.g., forced tunneling to Azure Firewall) to reduce alternate exfil paths.

5.4.5 5) Cross-subscription / cross-tenant governance considerations

- You should standardize:
    - Who can create private endpoints (consumer side).
    - Who can approve private endpoint connections (provider side).
    - Naming/tagging conventions for endpoint ownership and purpose.
- You should use **Azure Policy** guardrails where it materially affects data plane security (e.g., deny creation of public PaaS exposure, require private endpoint for certain services, require diagnostics).

5.5 Validation

You should validate per consuming VNet/subnet and per PaaS resource:

1. **DNS resolution**
    1. Query the PaaS FQDN from the workload subnet.
    2. Confirm the A record returns the **private endpoint IP**, not a public IP.
2. **Connectivity path**
    1. Confirm TCP/TLS succeeds to the private endpoint IP.
    2. Confirm no required path traverses Internet egress (especially in hybrid).
3. **Public access posture**
    1. Confirm public network access is disabled (if supported/required).
    2. Confirm attempts to access via public endpoint fail from unapproved networks.
4. **Routing alignment (data plane)**
    1. Confirm forced tunneling/UDRs do not inadvertently blackhole required Private Link paths.
    2. Confirm effective routes and NSG rules allow the intended client-to-private-endpoint flows only.
5. **Logging**
    - Confirm diagnostic settings/telemetry are enabled for:
        - DNS resolver/forwarders (query logs where available),
        - Private endpoint connection events (where exposed),
        - PaaS resource access logs (service dependent).

5.6 Pitfalls

- **DNS zone not linked** to all client VNets, causing resolution mismatch across spokes.

- Leaving **public network access enabled** “temporarily” and never removing it, enabling silent fallback and audit gaps.
- Over-reliance on IP allowlists instead of Private Link approvals and private-only posture.
- Creating private endpoints in ad hoc subnets without NSG/route hygiene, complicating incident response.
- Not accounting for hybrid DNS forwarding, causing on-prem clients to resolve public records.

5.7 Audit evidence

You should retain and be able to produce:

- Inventory of Private Endpoints per PaaS resource, including:
    - Subscription/resource group, VNet/subnet, and connection approval state.
- Azure Private DNS Zone configuration:
    - Zone list, VNet links, and record sets for private endpoints.
- PaaS resource network configuration evidence:
    - Public network access disabled (where applicable), firewall rules scoped.
- Validation outputs:
    - DNS query results showing private IP resolution from representative spokes/on-prem.
    - Connectivity tests demonstrating private endpoint usage.
- Logging configuration:
    - Diagnostic settings and retention policies for relevant DNS and PaaS logs.
- Change control records for endpoint creation/approval and DNS link changes.

5.7.1 Controls checklist (s05)

- [ ] Private Endpoint used for PaaS requiring private-only access; Service Endpoint justified where used.
- [ ] Correct Azure Private DNS Zone created and linked to all consuming VNets (and hub DNS resolver VNet).
- [ ] Hybrid DNS forwarding configured (conditional forwarders) where on-prem access is required.
- [ ] Public network access disabled on PaaS where supported; broad IP allowlists removed.

- [ ] NSG/UDR allow only required client-to-private-endpoint flows; forced tunneling does not break Private Link.
- [ ] Diagnostics/logging enabled for DNS resolution path and PaaS access; evidence retained for audit.

6. Hybrid Connectivity Security (VPN, ExpressRoute, SD-WAN/NVAs)

6.1 Objectives

- Secure hybrid **data plane** connectivity between on-premises and Azure, minimizing exposure to interception, route hijack, and lateral movement.
- Define resilient routing and inspection patterns for **north-south traffic** (on-prem ↔ Azure) and spillover **east-west traffic** (spoke ↔ spoke via hub).
- Establish implementation controls for gateways, **BGP (Border Gateway Protocol)** routing, and NVA/firewall high availability (HA) that produce auditable evidence.

6.2 When to use

- You have on-premises networks that must access Azure workloads or PaaS privately, and you need predictable routing and segmentation across connectivity types.
- You require dual connectivity (e.g., **ExpressRoute** primary with site-to-site **VPN** backup) with controlled failover behavior and inspection continuity.
- You deploy SD-WAN or other **NVA (Network Virtual Appliance)** patterns that impact routing intent, encryption, and availability.

6.3 Design decisions

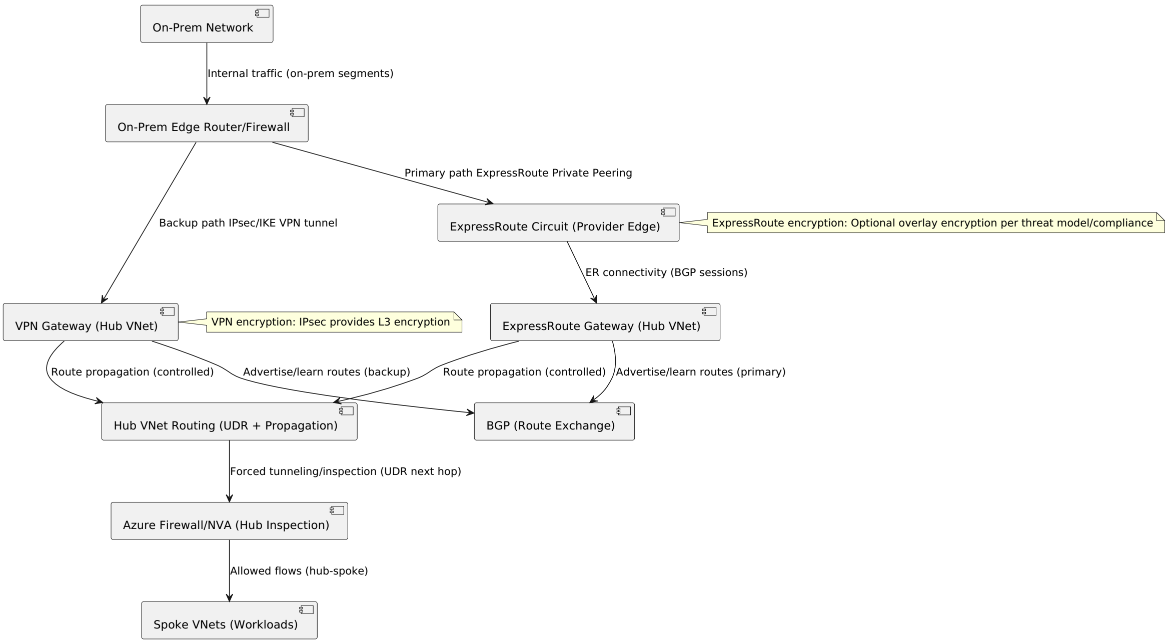
- **Connectivity option selection**
    - **Site-to-site VPN (IPsec)**: Encrypted over Internet; faster to deploy; typically lower bandwidth and higher jitter than ExpressRoute.
    - **ExpressRoute**: Private connectivity via provider; not Internet-routed; encryption is not inherent at L3 and may require overlay encryption depending on threat model and compliance.

- **Peering model (ExpressRoute)**
  - Prefer **ExpressRoute Private Peering** for on-prem ↔ VNet private IP connectivity.
  - Use Microsoft Peering only for approved scenarios; treat as expanded exposure surface and apply strict route filters and egress controls.
- **Routing intent and inspection**
  - Hub-spoke reference: terminate gateways in the **hub VNet**, steer traffic through **Azure Firewall** or NVA for centralized inspection and logging.
  - Decide whether on-prem ↔ spoke traffic must be inspected (recommended when threat model includes insider/lateral movement).
- **Failover strategy**
  - Decide on routing preference and convergence targets for ExpressRoute↔VPN failover using BGP attributes/communities and controlled propagation.
  - Decide how to prevent asymmetric routing during partial failures (a primary cause of stateful firewall drops).

Security rationale: Hybrid links are high-value trust boundaries. Treat routing control (BGP advertisements, propagation, and default route behavior) as a security control, not only a networking function, because misrouting can bypass inspection and enable data exfiltration.

6.4 Implementation notes

6.4.1 Reference topology (hub with ExpressRoute + VPN failover)


On-Prem Network
Internal traffic (on-prem segments)
On-Prem Edge Router/Firewall
Primary path ExpressRoute Private Peering
Backup path IPsec/IKE VPN tunnel
ExpressRoute Circuit (Provider Edge)
ExpressRoute encryption: Optional overlay encryption per threat model/compliance
ER connectivity (BGP sessions)
VPN Gateway (Hub VNet)
VPN encryption: IPsec provides L3 encryption
ExpressRoute Gateway (Hub VNet)
Route propagation (controlled)
Advertise/learn routes (backup)
Route propagation (controlled)
Advertise/learn routes (primary)
Hub VNet Routing (UDR + Propagation)
BGP (Route Exchange)
Forced tunneling/inspection (UDR next hop)
Azure Firewall/NVA (Hub Inspection)
Allowed flows (hub-spoke)
Spoke VNets (Workloads)

6.4.2 Site-to-site VPN security (IPsec/IKE)

1. You should enforce modern cryptographic suites (e.g., IKEv2, AES-GCM where supported) and disable legacy algorithms.
2. You should use strong pre-shared keys (PSK) or certificate-based authentication where feasible, and rotate keys on a defined schedule.
3. You should constrain traffic selectors (or policy-based rules when required) to only necessary prefixes; avoid “any-to-any” if it causes unnecessary trust expansion.
4. You should log tunnel events and renegotiations and retain logs according to your audit retention requirements.

   Security rationale: VPN is often the “break-glass path” during ExpressRoute issues. If its cipher suites, authentication, or route scope are weaker, attackers may target it as the easier entry point.

6.4.2.1 Anti-pattern

- Allowing broad on-prem prefixes over VPN “temporarily,” then never removing them, effectively creating an unsegmented hybrid flat network.

6.4.3 ExpressRoute security (Private Peering and encryption overlays)

- **Route control**
    - You should use BGP route filters and explicit prefix lists to constrain what is advertised and accepted (least route principle).
    - You should explicitly decide whether to propagate a default route (`0.0.0.0/0`) from on-prem into Azure; default route propagation can unintentionally force all Azure egress on-prem (or bypass hub inspection if misconfigured).
- **Encryption decision**
    - ExpressRoute provides private connectivity but not automatic payload encryption at L3.
    - If your threat model includes provider insider risk, regulatory requirements for encryption-in-transit, or sensitive workloads, you should implement overlay encryption (e.g., IPsec over ExpressRoute, or application-level TLS where appropriate).

Security rationale: ExpressRoute reduces exposure to Internet-based interception but does not eliminate insider and misconfiguration threats; overlay encryption mitigates confidentiality risks when required.

6.4.4 BGP, propagation, and asymmetric routing prevention

1. You should design for *symmetry through stateful inspection points* (Azure Firewall/NVA). Ensure return paths follow the same inspection state table.
2. You should implement clear route preference logic for primary/backup:
    - Prefer ExpressRoute (primary) and de-prefer VPN (backup) using BGP attributes (as supported) and consistent on-prem and Azure gateway configuration.
3. You should avoid uncontrolled transitive routing between spokes:
    - In hub-spoke, only allow spoke-to-spoke via hub when explicitly required, and ensure it is inspected and filtered.
4. You should document route propagation boundaries:
    - Identify which route tables (UDR) are associated to which subnets and how propagation from gateways is controlled.

6.4.4.1 Anti-pattern

- Mixing gateway route propagation and UDRs without deterministic rules, resulting in hidden “escape paths” that bypass Azure Firewall/NVA inspection.

6.4.5 NVA / SD-WAN placement and high availability

- **Placement**
  - You should place NVAs in the hub VNet (shared services/security) unless a workload requires dedicated isolation (then use a dedicated spoke with controlled peering and UDR).
- **HA**
  - You should deploy NVAs in an HA pair or scale set pattern aligned to vendor guidance, across availability zones when supported.
  - You should ensure UDR next hops and health probes fail over without creating blackholes or asymmetric return paths.
- **Key management**
  - You should manage NVA credentials, API tokens, and certificates using an approved secrets system and rotate per policy.

  Security rationale: NVAs often become chokepoints. HA misconfigurations are not only availability risks; they can trigger emergency bypass routes that reduce security.

6.5 Validation

You should validate hybrid connectivity security with repeatable tests (runbooks) and capture outputs for audit:

1. **Routing intent validation**
   1. Verify effective routes on representative spoke subnets (system + UDR + propagated).
   2. Confirm on-prem prefixes are learned via intended gateway (ExpressRoute primary; VPN backup).
2. **Inspection continuity**
   1. Test that on-prem ↔ spoke traffic traverses Azure Firewall/NVA (e.g., via firewall logs and flow diagnostics).
   2. Confirm explicit denies for prohibited east-west paths are enforced (e.g., dev→prod).

3. **Failover testing**
    1. Simulate ExpressRoute unavailability (provider-approved procedure) and confirm VPN takes over within your RTO/RPO targets.
    2. Confirm reversion to ExpressRoute does not cause asymmetric routing or session drops beyond acceptable thresholds.
4. **Encryption verification**
    - VPN: confirm IPsec parameters negotiated match the standard.
    - ExpressRoute overlay (if used): confirm tunnel establishment and that sensitive traffic uses the encrypted path.

6.6 Pitfalls

- Allowing default route propagation from on-prem without confirming how it interacts with forced tunneling and Azure Firewall/NVA.
- Underestimating **SNAT** needs at centralized egress/inspection points when hybrid traffic patterns change during failover.
- Failing to align on-prem routing policy with Azure UDR/propagation, causing asymmetric routing during maintenance or partial outages.
- Treating ExpressRoute as “encrypted by default” and omitting overlay encryption where compliance requires it.

6.7 Audit evidence

You should retain the following artifacts (minimum set) to support auditability and change control:

- Architecture diagrams and routing intent statements (what routes are allowed, where inspection occurs).
- Gateway configurations:
    - VPN: IKE/IPsec policies, authentication method, key rotation records.
    - ExpressRoute: peering configuration, accepted/advertised prefixes, route filters.
- Route evidence:
    - Effective route exports from representative subnets (hub and spokes).
    - BGP tables/learned routes from gateways and on-prem edge.
- Inspection and logging:
    - Azure Firewall/NVA policy snapshots, rule change history, and logs proving traffic traversal.
- Failover test records:
    - Test plan, timestamps, observed convergence, and issues/remediations.

6.8 Controls checklist (s06)

- [ ] ExpressRoute and VPN routing preference is explicitly defined and tested (primary/backup) with documented convergence targets.
- [ ] BGP advertisements/acceptance are least-privilege (only required prefixes) with documented ownership and approvals.
- [ ] On-prem ↔ spoke traffic inspection path is deterministic (UDR/propagation documented) and validated via logs.
- [ ] Asymmetric routing risks are analyzed and mitigated (stateful inspection points aligned with return paths).
- [ ] VPN cryptographic policy and key management meet current standards; rotation is evidenced.
- [ ] ExpressRoute encryption requirements are assessed; overlay encryption is implemented where required and validated.
- [ ] NVA/SD-WAN is deployed with HA and tested failover without security bypass.
- [ ] Audit artifacts (effective routes, BGP tables, firewall/NVA logs, change records) are collected and retained per policy.

7. s07 — Identity, Access, and Governance for Network Controls

7.1 Objectives

- You should implement least privilege for network administration across **Azure Virtual Network (VNet)**, **Network Security Group (NSG)**, **Azure Firewall**, routing (**UDR (User-Defined Route)**), and **Private Endpoint** / **Azure Private DNS Zone** operations.
- You should enforce preventative guardrails (primarily **Azure Policy**) that materially reduce network security misconfiguration in the **data plane**.
- You should establish auditable separation of duties (SoD) for change control on network security-critical resources.

7.2 When to use

- You operate a **hub-spoke** or **Azure Virtual WAN (vWAN)** environment with centralized security (e.g., **Azure Firewall** forced tunneling) and multiple workload teams.

- You need to reduce risk from:
    - **Insider**: unauthorized NSG/UDR/firewall policy change enabling exfiltration or lateral movement.
    - **Misconfiguration**: public exposure, “Any-Any”, missing forced tunneling, Private DNS failures causing public fallback.
    - **Supply chain**: CI/CD pipeline misuse or unauthorized IaC changes impacting network controls.
    - **External**: weakened ingress/egress controls (e.g., bypassing WAF, creating public endpoints).
- You are preparing for audit readiness and need clear evidence for “who can change what, where, and how changes are controlled”.

7.3 Design decisions

- **Scope model (recommended)**:
    - Assign broad read-only roles at **Management Group** for visibility (auditor/SOC).
    - Assign network change roles at **Resource Group** boundaries aligned to architecture:
        - `rg-hub-networking` (firewall, gateways, central DNS, bastion)
        - `rg-spoke-<app>-networking` (spoke VNets, subnets, NSGs, UDR associations)
        - `rg-shared-dns` (Private DNS Zones and links) where centralized
- **Separation of duties** (minimum set):
    - Firewall policy owners are distinct from workload operators who attach route tables / NSGs.
    - DNS owners (Private DNS zones/links) are controlled to prevent private name resolution bypass.
    - Log access is granted without configuration change rights.
- **Privileged elevation**:
    - Require **PIM (Privileged Identity Management)** activation for privileged roles that can alter security posture (firewall policy, UDR default route, NSG denies, Private DNS links).
- **Policy guardrails**:
    - Use Azure Policy “deny” or “deployIfNotExists” where feasible for high-impact network controls (e.g., disallow public IP on NICs in spokes; require NSG on subnets; require diagnostic settings).

- **Change path**:
    - Prefer Infrastructure-as-Code (IaC) with protected pipelines over portal changes for repeatability and audit evidence (details of SDLC are out of scope; only network-impacting controls are specified here).

  Security rationale: Central network controls (NSG/UDR/Azure Firewall/Private DNS) are high-leverage security boundaries. Least privilege + PIM + policy reduces the probability and blast radius of insider abuse and misconfiguration that could enable Internet exposure, forced-tunneling bypass, or lateral movement.

  Anti-pattern: Granting “Owner” or “Contributor” at subscription scope to network operators or workload teams. This commonly leads to ad-hoc NSG exceptions, route-table bypass of inspection, and untracked public endpoint creation.

7.4 Implementation notes

7.4.1 RBAC model (roles, scopes, responsibilities)

You should define personas and map them to minimal Azure roles at the narrowest scope that supports operations:

- **Network Security Admin** (privileged; PIM required)
    - Scope: Hub networking RG and shared DNS RG (or equivalent)
    - Can modify: Azure Firewall policies/rules, DNAT/SNAT-related configuration, NSG standards, UDR templates, Private DNS zone links (where centralized)
- **Network Operator** (non-privileged or PIM for limited actions)
    - Scope: Spoke networking RGs
    - Can modify: subnet-to-NSG associations, UDR associations, local NSG rules within approved patterns (no “Any-Any”)
- **Platform Engineer** (workload/platform delivery)
    - Scope: Workload RGs; limited networking actions only when required (e.g., attach ASG references, request Private Endpoint via controlled process)
- **Auditor** (read-only)
    - Scope: Management Group / Subscriptions
    - Can read: config state and policy assignments; cannot access sensitive payloads

- **SOC Analyst** (read-only + log analytics)
  - Scope: Management Group / Subscriptions for config read; Log Analytics/Sentinel for log queries
  - Can read: activity logs, resource diagnostics, flow logs; cannot change enforcement points

Custom role guidance (implementation-oriented)

- You should create custom roles when built-in roles are too permissive (common for route tables, firewall policy, Private DNS).
- You should explicitly exclude actions that allow bypass:
  - Route changes that could remove `0.0.0.0/0` forced tunneling UDRs
  - Creation/attachment of Public IP to NICs in spoke workloads
  - Modifying Private DNS zone links that could cause public DNS fallback

  Security rationale: Most security-impacting network events are caused by a small set of actions (modify UDR default routes, widen NSG rules, weaken firewall policy, detach Private DNS). Custom roles reduce accidental privilege that can silently create breakout paths.

7.4.2 PIM activation and break-glass

You should implement PIM and emergency access with explicit constraints:

1. Configure PIM for privileged RBAC assignments:
   1. Make `Network Security Admin` eligible (not permanent) at hub/shared scopes.
   2. Require approval and justification; set time-bound activation.
   3. Enforce MFA requirements consistent with your identity baseline (identity specifics are out of scope).
2. Define a break-glass process for network security incidents:
   1. Use a dedicated emergency role assignment path (time-boxed) for firewall policy restoration or routing repair.
   2. Store runbooks and escalation paths in an audited system.
   3. Ensure break-glass actions are logged and reviewed post-incident.

   Anti-pattern: Permanent privileged access for “convenience” to avoid PIM friction. Audits commonly flag this, and it increases insider and compromised-account risk.

7.4.3 Azure Policy guardrails for network security-critical resources

You should assign Azure Policy at Management Group or subscription scope for controls that prevent high-impact misconfiguration (examples are illustrative; adapt to your naming, regions, and architecture).

Key policy intents (threat-driven):

- **Misconfiguration / external exposure**
    - Deny Public IP creation in spoke subscriptions (or restrict to approved RGs)
    - Deny NIC public IP association for workloads
    - Restrict allowed locations for network resources (limit sprawl)
- **East-west / lateral movement**
    - Require NSG on subnets (or deny subnet creation without NSG association)
    - Enforce baseline NSG rules (e.g., explicit denies for dev→prod patterns) where feasible via initiative/policy-as-code
- **Inspection / exfiltration**
    - Deny route table changes that remove forced tunneling (where implementable; otherwise detect via audit policy + alert)
    - Require Azure Firewall diagnostics and threat-intel logs to central workspace
- **Private access**
    - Require Private Endpoint for specified PaaS types (where business-approved)
    - Require Private DNS zone group configuration for Private Endpoints (where supported) to prevent public DNS fallback

> **Security rationale:** Preventative “deny” policy is the fastest way to stop accidental public exposure and inspection bypass, which are common precursors to external compromise and data exfiltration.

7.4.4 Resource locks, tagging, and management group structure (network-impacting only)

You should apply governance that directly reduces network security drift:

1. Apply resource locks (`CanNotDelete`) to:
    1. Hub firewall resources and firewall policy objects
    2. Central route tables used for forced tunneling

    3. Private DNS zones used for Private Endpoint name resolution
2. Enforce tags required for audit and ownership routing:
    - `Owner`, `DataClassification`, `Environment`, `CostCenter`, `Criticality`
3. Use Management Groups to separate environments (e.g., Prod vs NonProd) when it improves policy assignment and SoD.

    Anti-pattern: Locking entire RGs without an operational exception process. This often causes emergency, unreviewed “unlock” operations that reduce auditability.

7.4.5 IaC change control for network controls (network-impacting aspects only)

You should constrain network changes to controlled pipelines where possible:

1. Store IaC (Bicep/Terraform) in version control with protected branches.
2. Require pull-request review by a separate approver group for changes to:
    - NSG rules, ASG references, UDRs, Azure Firewall policy/rules, Private DNS zones/links
3. Implement a policy to block direct portal changes for privileged resources where feasible, or at minimum detect and alert on them using Activity Log.

    Security rationale: Pipeline-based changes reduce supply chain and insider risk by ensuring traceability, review, and repeatability of security boundary modifications.

7.5 Validation

You should validate both **authorization** and **effective security posture**:

1. RBAC verification (who can change what):
    1. Export role assignments at Management Group/subscription/RG scopes.
    2. Validate no broad “Owner/Contributor” grants exist for non-security personas at high scopes.
2. PIM verification:
    1. Confirm privileged roles are eligible, require approval, and are time-bound.
    2. Validate activation and audit events are retained.
3. Policy verification:
    1. Confirm policy assignments are in place at intended scopes.

   2. Test denial for public IP creation in a spoke scope (non-production).
   3. Validate deployIfNotExists created diagnostic settings for firewall/NSG logs (where used).
4. Drift/bypass validation (data-plane relevant outcomes):
   1. Confirm forced tunneling remains effective by checking **effective routes** on representative spoke NICs.
   2. Confirm Private Endpoint name resolution uses **Azure Private DNS Zone** (no public fallback) from a spoke VM test host.

7.6 Pitfalls

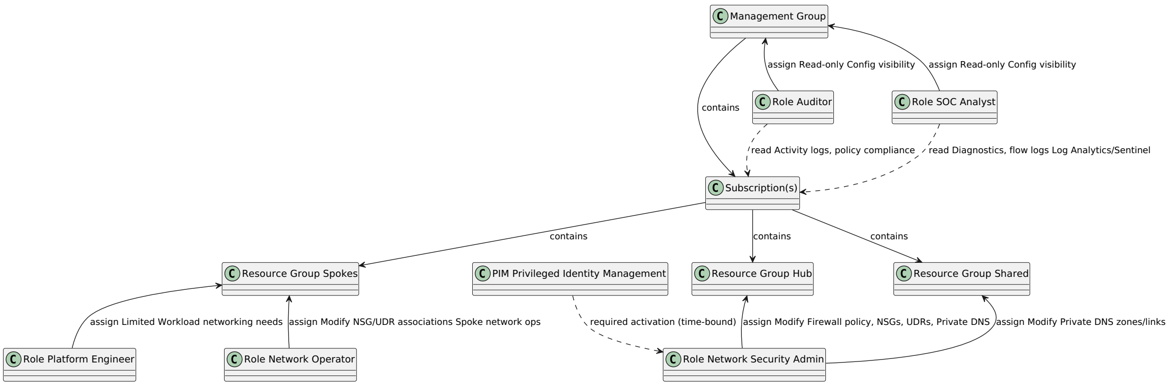
- Over-scoping roles at subscription level “to simplify operations”, causing uncontrolled ability to:
  - Remove forced tunneling UDRs
  - Introduce `Any/Any` NSG rules
  - Alter Azure Firewall policy to allow direct Internet egress
- Treating Azure Policy as “optional guidance” (audit-only) for controls that should be preventative.
- Splitting Private DNS ownership across many teams without clear authority, leading to broken zone links and public endpoint fallback.
- Relying solely on portal activity without enforcing pipeline review for security boundary changes.

7.7 Audit evidence

You should be able to produce the following on request:

- RBAC exports:
  - Role assignments by scope (Management Group/subscription/RG) for network-related roles
  - Custom role definitions (JSON) demonstrating least privilege boundaries
- PIM configuration:
  - Eligible role settings, approval requirements, activation logs, and review attestations
- Azure Policy artifacts:
  - Policy/initiative definitions, assignments, exemptions, and compliance reports
- Change-control evidence:
  - Pull requests and approvals for changes to NSG/UDR/Azure Firewall/Private DNS

    - Activity Log queries showing privileged operations and their initiators
- Logging access model:
    - Proof that Auditor/SOC have read access to configurations and logs without modify rights


Management Group
contains
assign Read-only Config visibility
assign Read-only Config visibility
Role Auditor
Role SOC Analyst
read Activity logs, policy compliance
read Diagnostics, flow logs Log Analytics/Sentinel
Subscription(s)
contains
contains
contains
Resource Group Spokes
PIM Privileged Identity Management
Resource Group Hub
Resource Group Shared
assign Limited Workload networking needs
assign Modify NSG/UDR associations Spoke network ops
required activation (time-bound)
assign Modify Firewall policy, NSGs, UDRs, Private DNS
assign Modify Private DNS zones/links
Role Platform Engineer
Role Network Operator
Role Network Security Admin

7.7.1 Controls checklist (s07)

- [ ] RBAC assignments are scoped to RG where possible; subscription/MG scope is justified and reviewed.
- [ ] Privileged network roles (e.g., firewall policy/UDR default route/Private DNS links) require PIM activation with approval and time limits.
- [ ] Azure Policy denies or deploys guardrails for public exposure and required diagnostics at the correct scope.
- [ ] Resource locks protect hub firewall, critical route tables, and Private DNS zones with an exception process.
- [ ] Network security boundary changes are routed through audited IaC pipelines with peer review and traceable approvals.
- [ ] Auditor and SOC have read-only access to configurations and logs; they cannot modify enforcement points.

8. Network Security Controls: NSG, ASG, Azure Firewall, and NVA Patterns

8.1 Objectives

- You should implement layered Azure networking data plane controls for **north-south traffic**, **east-west traffic**, and **egress** while minimizing misconfiguration risk.
- You should standardize rule patterns (NSG/ASG, Azure Firewall Policy, NVA steering) with repeatable validation and audit evidence.

8.2 When to use

- You should use this section when designing or hardening a **hub-spoke** environment where:
    - Spokes require **microsegmentation** using NSG and ASG (Application Security Group).
    - The hub provides centralized inspection/egress using Azure Firewall (including Premium features) and/or an NVA (Network Virtual Appliance).
    - You need operational safety controls: change control, drift detection, and log-based verification.

8.3 Design decisions

- **Layering model**
    - Use NSG + ASG (Application Security Group) for *workload-local* enforcement (microsegmentation, subnet/NIC boundaries).
    - Use Azure Firewall for *centralized* control (forced tunneling, SNAT/DNAT, URL filtering, IDPS).
    - Use NVAs when you require features not met by Azure Firewall (e.g., specific compliance-certified inspection, SD-WAN), and accept added operational overhead.
- **Default posture**
    - You should implement **deny-by-default** with explicit allows for required flows, including **explicit egress** allows.

- **Routing as enforcement**
    - You should treat routing intent (e.g., UDR (User-Defined Route) for forced tunneling) as part of the security control plane because it determines whether traffic reaches inspection/enforcement points.
- **Threat-model alignment**
    - External (north-south): perimeter controls + DNAT publishing patterns.
    - East-west: NSG/ASG tiering and explicit lateral denies.
    - Insider/misconfiguration: JIT, least privilege on rule changes, policy guardrails, and logging/retention.

  Security rationale: Layering NSG (workload-local) with centralized inspection (Azure Firewall or NVA) reduces blast radius for east-west movement and creates deterministic egress control and auditability for north-south and outbound traffic.

8.4 Implementation notes

8.4.1 NSG + ASG microsegmentation baseline

8.4.1.1 Objectives

- You should implement a reusable NSG/ASG pattern to constrain east-west flows by role/tier, not IPs.
- You should ensure rules are testable, logged, and resilient to scaling and redeployments.

8.4.1.2 When to use

- You should use NSG/ASG microsegmentation when workloads are VM- or NIC-based and you need tier-based control (e.g., web/app/data).
- You should use it in spokes even when a hub firewall exists; the firewall does not replace local segmentation.

8.4.1.3 Design decisions

- **ASG-centric rules:** Prefer ASG (Application Security Group) references over IP/CIDR where possible.
- **Subnet vs NIC assignment:** Apply NSGs at subnet for consistency; apply at NIC only for exceptions with documented justification.
- **Priorities:** Reserve priority ranges (e.g., 100–199 platform/ingress, 200–399 east-west tier allows, 400–499 management, 900–999 explicit denies).

- **Service tags:** Prefer Azure service tags where applicable, but do not assume they replace Private Link segmentation needs.

8.4.1.4 Implementation notes

1. **Define tiers and ASGs**
    1. You should create ASGs per role (e.g., `asg-web`, `asg-app`, `asg-data`) and ensure all relevant NICs are members.
    2. You should document ownership and lifecycle rules for ASG membership (e.g., enforced via IaC).
2. **Create NSGs and attach**
    1. You should create an NSG per tier subnet (or per spoke standard) and attach to the corresponding subnet.
    2. You should avoid mixing unrelated workloads in the same subnet if they require distinct policies.
3. **Author explicit allow rules (least privilege)**
    1. You should allow only required ports between ASGs (e.g., web→app 443; app→data 1433).
    2. You should restrict sources to the minimal ASG/subnet scope.
4. **Author explicit egress controls**
    1. You should allow egress only to required destinations (e.g., hub firewall IP/subnet, required Private Endpoints).
    2. You should add explicit denies for prohibited lateral paths (e.g., dev→prod) using high-priority deny rules.
5. **Enable logging**
    1. You should enable NSG flow logs and send them to a central Log Analytics workspace (and SIEM such as Microsoft Sentinel if used).
    2. You should standardize retention to meet audit and incident response requirements.

Anti-pattern: Relying on default NSG rules (or "it's in a VNet") for segmentation. A Azure Virtual Network (VNet) provides routing isolation, not enforcement.

8.4.1.5 Validation

1. You should verify **effective security rules** on representative NICs in each tier (portal/CLI) and confirm "first match wins" behaves as intended.
2. You should perform connectivity tests aligned to the rule matrix:
    1. Internet→web (only intended ports).
    2. web→app (only intended ports).
    3. app→data (only intended ports).

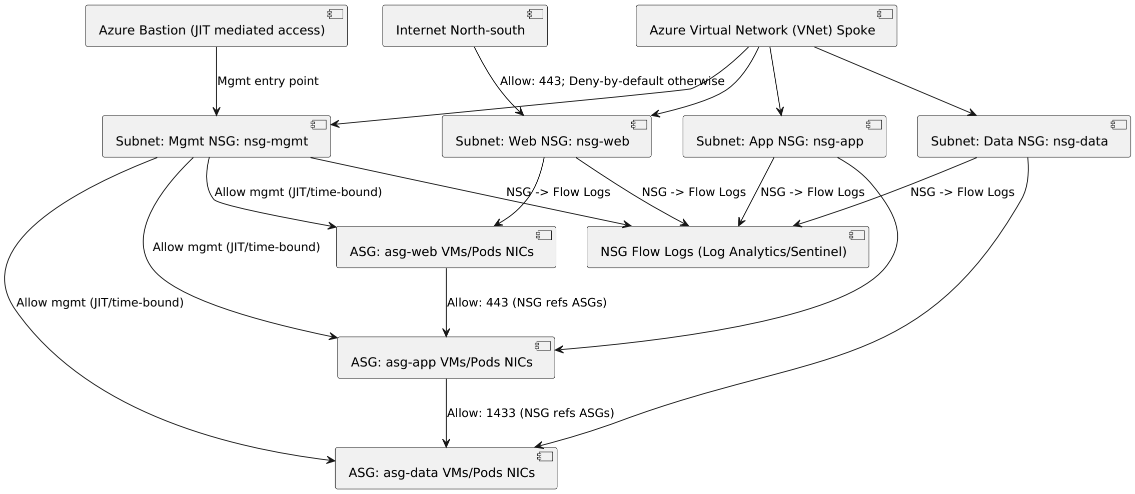
    4. mgmt→tiers (only via approved management path).
3. You should validate NSG flow logs show:
    1. Allowed tier flows.
    2. Denied lateral attempts (intentional negative tests).

8.4.1.6 Pitfalls

- Overlapping rules where a broad allow at higher priority nullifies later denies.
- ASG membership drift when deployments are not IaC-managed.
- Missing egress rules leading to implicit outbound behavior that bypasses intended inspection when routing changes.

8.4.1.7 Audit evidence

- Exported NSG rule sets (JSON/IaC), including priority conventions and change history.
- Evidence of ASG membership governance (IaC pipeline logs, deployment records).
- NSG flow log configuration and retention settings; sample queries showing allow/deny verification.


Azure Bastion (JIT mediated access)
Internet North-south
Azure Virtual Network (VNet) Spoke
Mgmt entry point
Allow: 443; Deny-by-default otherwise
Subnet: Mgmt NSG: nsg-mgmt
Subnet: Web NSG: nsg-web
Subnet: App NSG: nsg-app
Subnet: Data NSG: nsg-data
Allow mgmt (JIT/time-bound)
NSG -> Flow Logs
NSG -> Flow Logs
NSG -> Flow Logs
NSG -> Flow Logs
Allow mgmt (JIT/time-bound)
ASG: asg-web VMs/Pods NICs
NSG Flow Logs (Log Analytics/Sentinel)
Allow mgmt (JIT/time-bound)
Allow: 443 (NSG refs ASGs)
ASG: asg-app VMs/Pods NICs
Allow: 1433 (NSG refs ASGs)
ASG: asg-data VMs/Pods NICs

8.4.1.8 Controls checklist (s08: NSG/ASG)

- [ ] You enforce deny-by-default with explicit ingress *and egress* allows.
- [ ] You reference ASG (Application Security Group) instead of IPs where feasible.
- [ ] You reserve priority ranges and prevent broad allows overriding targeted denies.

- [ ] You enable NSG flow logs centrally with defined retention and query validation.
- [ ] You have documented exceptions (NIC-level NSGs, temporary rules) with expiry/change control.

8.4.2 Azure Firewall patterns (Policy, SNAT/DNAT, Premium features)

8.4.2.1 Objectives

- You should centralize egress control and inbound publishing with auditable policy structure.
- You should enable advanced inspection (when required) while controlling privacy and operational risk.

8.4.2.2 When to use

- You should use Azure Firewall when you need centralized:
  - Forced tunneling egress allowlisting and logging.
  - DNAT publishing to internal workloads (non-HTTP/S scenarios) or as part of a broader perimeter stack.
  - IDPS and URL filtering (Premium), optionally TLS inspection with governance.

8.4.2.3 Design decisions

- **Firewall Policy structure**
  - Use hierarchical Firewall Policy (parent/child) to reuse global baselines and allow spoke-specific overrides.
  - Use rule collection groups to separate concerns:
    - Platform baseline (DNS/NTP/management).
    - Workload egress allowlists.
    - Inbound DNAT (if used).
    - Threat protection (IDPS categories, URL filtering).
- **Forced tunneling**
  - You should steer `0.0.0.0/0` from spoke subnets via UDR (User-Defined Route) to Azure Firewall, and ensure the firewall performs **SNAT**.
  - You should plan SNAT port utilization for peak outbound concurrency.

- **Premium features governance**
    - Enable IDPS with tuned mode (alert/deny) aligned to environment criticality.
    - Use TLS inspection only with approved certificate handling, privacy review, and explicit bypasses for sensitive categories where required.

Security rationale: Centralized egress control and inspection reduces exposure to supply-chain and C2 (command-and-control) traffic, and provides consistent logging for investigations across spokes.

8.4.2.4 Implementation notes

1. **Deploy and integrate**
    1. You should deploy Azure Firewall in the hub and attach it to the hub virtual network per Microsoft reference.
    2. You should ensure spokes have route tables that force egress via the firewall (forced tunneling).
2. **Create Firewall Policy and baselines**
    1. You should create a parent policy with shared rule collection groups (DNS, required Azure control traffic where applicable).
    2. You should create child policies per environment/spoke class for workload-specific egress rules.
3. **Egress allowlisting**
    1. You should prefer FQDN-based rules where stable and compatible; otherwise use IP groups with controlled updates.
    2. You should document all outbound dependencies and owners (application teams).
4. **DNAT (if required)**
    1. You should implement DNAT only for explicitly approved inbound services and prefer WAF-based publishing for HTTP/HTTPS where possible (to reduce bypass).
    2. You should ensure NSGs on target subnets allow only the firewall as source for inbound DNAT flows.
5. **Enable Premium protections (as required)**
    1. You should enable IDPS and validate rule modes and alert routing.
    2. If enabling TLS inspection, you should implement certificate lifecycle management and explicit bypass categories.

Anti-pattern: Using Azure Firewall without forced tunneling (or without UDR coverage for all relevant subnets), resulting in partial inspection and inconsistent audit logs.

8.4.2.5 Validation

1. You should validate **effective routes** for each spoke subnet show `0.0.0.0/0` next hop to the firewall (or hub security provider in vWAN scenarios).
2. You should validate SNAT behavior by confirming outbound public IP identity and ensuring return traffic symmetry.
3. You should validate firewall logs:
    1. Application/network rule hits for allowed egress.
    2. Denies for blocked categories/domains (negative tests).
    3. IDPS alerts align to expected tuning.

8.4.2.6 Pitfalls

- SNAT exhaustion under high outbound concurrency (symptoms: intermittent outbound failures).
- Over-broad FQDN rules (e.g., `*.cloudprovider.com`) that undermine egress intent.
- DNAT exposure without compensating controls (lack of WAF for HTTP/S, weak NSG constraints).

8.4.2.7 Audit evidence

- Firewall Policy exports (parent/child) and rule collection group definitions.
- Route table exports showing forced tunneling coverage (including exception documentation).
- Firewall diagnostic settings and log retention; sample incident-trace queries proving visibility.

8.4.2.8 Controls checklist (s08: Azure Firewall)

- [ ] You enforce forced tunneling with validated UDR (User-Defined Route) coverage.
- [ ] You structure Firewall Policy using parent/child and rule collection groups with ownership.
- [ ] You document and test SNAT capacity assumptions and monitoring thresholds.
- [ ] You restrict DNAT and prevent origin bypass (NSG source constraint to firewall).
- [ ] You enable and retain firewall logs; you test allow/deny cases routinely.

8.4.3 NVA patterns (steering, high availability, and operational safety)

8.4.3.1 Objectives

- You should deploy NVAs with predictable routing, health, and failover behavior.
- You should minimize misconfiguration and drift while keeping audit-ready evidence.

8.4.3.2 When to use

- You should use an NVA (Network Virtual Appliance) when:
  - Required inspection/SD-WAN features are not available in Azure Firewall for your constraints.
  - You need vendor-specific capabilities and can operate patching, scaling, and HA.

8.4.3.3 Design decisions

- **HA model**
  - Active/active when supported with symmetric routing and consistent state handling.
  - Active/passive when simplicity and deterministic failover outweigh utilization.
- **Scaling**
  - Prefer scale sets when vendor supports it; otherwise, use a documented scale unit and capacity model.
- **Steering**
  - Use UDR (User-Defined Route) to direct specific prefixes/egress to NVAs; avoid ambiguous route precedence.

  Security rationale: NVAs can increase inspection depth but also expand the customer-managed attack surface (patching, credentials, management plane). Treat NVA operations as a high-risk control requiring strict change control and monitoring.

8.4.3.4 Implementation notes

1. **Placement and segmentation**
    1. You should place NVAs in the hub (or dedicated security spoke) with dedicated subnets and tightly scoped NSGs.
2. **Routing**
    1. You should create route tables for spoke subnets that set next hop to the NVA (or NVA load-balanced frontend, where applicable).
    2. You should ensure return paths are symmetric (avoid asymmetric routing that breaks stateful inspection).
3. **Health and failover**
    1. You should implement health probes and automated failover per vendor reference.
    2. You should test failure scenarios (instance down, interface down, route withdrawal if applicable).
4. **Operations**
    1. You should patch NVAs on a defined cadence and record versions.
    2. You should lock down NVA management access via Azure Bastion and/or JIT, and record administrative actions.

    Anti-pattern: Steering traffic to an NVA without proving return-path symmetry; this commonly causes intermittent drops that are misdiagnosed as application instability.

8.4.3.5 Validation

1. You should validate effective routes for representative subnets and confirm next hop is correct for each intended prefix.
2. You should perform failover testing and capture packet/flow evidence showing continuity or controlled degradation.
3. You should validate NVA logs are centralized and time-synchronized for correlation.

8.4.3.6 Pitfalls

- Asymmetric routing with stateful NVAs.
- NVA management exposure (public IPs, overly permissive NSGs).
- Drift between intended and deployed route tables during incremental changes.

8.4.3.7 Audit evidence

- NVA architecture decision record (why NVA vs Azure Firewall), including threat model justification.
- Route tables and health probe configuration exports.
- Patch/change records and centralized log proof.

8.4.3.8 Controls checklist (s08: NVA)

- [ ] You document the justification for NVA use and the shared responsibility boundaries.
- [ ] You implement and test HA/failover with symmetric routing.
- [ ] You lock down NVA management access (Bastion/JIT) and remove public exposure.
- [ ] You centralize NVA logs with retention and correlation readiness.
- [ ] You operate NVAs with patching/version control and change approvals.

9. s09 — Routing Security and Traffic Engineering

9.1 Objectives

- Reduce risk of route leaks and unintended exposure across north-south and east-west traffic paths.
- Engineer deterministic, symmetric, and auditable data plane routing, aligned to hub-spoke inspection and segmentation.
- Minimize misconfiguration risk by standardizing **UDR (User-Defined Route)** intent and limiting uncontrolled propagation.

9.2 When to use

- You operate a **hub-spoke** topology where you must force inspection for Internet egress (**forced tunneling**) and/or east-west flows.
- You use centralized inspection (e.g., **Azure Firewall**) and must ensure symmetric routing to avoid dropped stateful flows.
- You have hybrid connectivity using **BGP (Border Gateway Protocol)** (VPN/ExpressRoute) and must control route advertisements/propagation.
- You use **Azure Virtual WAN (vWAN)** secure hubs and need consistent “routing intent” to prevent bypass.

9.3 Design decisions

- **Inspection placement**
  - Use **Azure Firewall** in the hub for centralized L3/L4 and application-aware policy enforcement; optionally use Firewall Premium for IDPS/TLS inspection where governance supports it.
  - Keep **VNet provides routing/isolation; it is not a firewall**—routing must steer traffic to enforcement points (NSG/Azure Firewall/WAF).
- **East-west steering model**
  - Prefer hub inspection for spoke-to-spoke flows where lateral movement risk is material (threat model: **east-west**, **insider**, **misconfiguration**).
  - Decide whether to steer *all* spoke-to-spoke via hub or only selected prefixes (exception routes) based on latency/cost vs risk.
- **Routing authority and propagation**
  - Decide where BGP-learned routes are allowed to propagate (hub only vs into spokes) to reduce route leak blast radius.
  - Standardize route table hygiene (naming, ownership, drift control) and limit who can modify UDRs (control plane governance that materially impacts data plane security).
- **Symmetry requirements**
  - For stateful devices (Azure Firewall, NVAs), you must ensure **both directions** traverse the same device instance/path (avoid asymmetric routing).

Security rationale: Deterministic routing is a primary control against **misconfiguration** and **east-west** bypass. UDR-enforced steering ensures traffic reaches stateful inspection and logging points; symmetry preserves state and prevents “silent” policy bypass via alternate return paths.

9.4 Implementation notes

9.4.1 UDR patterns (forced tunneling, spoke-to-spoke via hub, exceptions)

1. **Forced tunneling (spoke egress via hub)**
   1. Create a route table per spoke subnet category (e.g., `rt-spoke-app`, `rt-spoke-data`) to avoid unintended coupling.

   2. Add `0.0.0.0/0 -> Virtual appliance` next hop set to **Azure Firewall** private IP in hub.
   3. Associate the route table to the spoke subnets that require forced tunneling.
   4. Confirm **SNAT** requirements: Azure Firewall performs SNAT for outbound unless explicitly configured otherwise; plan capacity to avoid port exhaustion.
2. **Spoke-to-spoke via hub inspection**
   1. In **Spoke A**, add UDR(s) for Spoke B CIDR(s) pointing to Azure Firewall private IP.
   2. In **Spoke B**, add UDR(s) for Spoke A CIDR(s) pointing to Azure Firewall private IP.
   3. Ensure peering settings allow forwarded traffic (where required) and that return routes do not follow a different path.
3. **Exception routes (bypass inspection with explicit justification)**
   1. Define exception prefixes narrowly (single PaaS endpoint prefixes are typically *not* stable; prefer **Private Endpoint** with private IP where possible).
   2. Document the exception threat acceptance (e.g., performance requirement) and add explicit monitoring compensating controls.
   3. Implement exceptions as higher-specificity routes (longest prefix match) and verify they do not reintroduce public fallback paths.

   Anti-pattern: Adding `0.0.0.0/0` to a spoke route table without validating all required platform dependencies (e.g., DNS forwarders, update repositories, monitoring agents) leads to outages and emergency “temporary” broad bypass rules that persist.

9.4.2 BGP security considerations (prefix filtering and route hygiene)

- You should treat BGP-learned routes as potentially unsafe inputs (threat model: **misconfiguration**, **supply chain** via partner connectivity).
- You should:
  - Implement prefix allowlists on on-prem/edge devices and constrain advertised prefixes to the minimal necessary.
  - Avoid propagating broad/overlapping routes into spokes; prefer learning routes centrally in the hub and explicitly steering via UDRs.
  - Regularly review effective routes for unexpected `0.0.0.0/0`, RFC1918 overlaps, or overly general summaries that could attract traffic unintentionally.

Security rationale: Prefix filtering reduces the chance that an incorrect advertisement (accidental or malicious) reroutes traffic around inspection or into unintended trust zones (route leak).

9.4.3 Azure Virtual WAN secure hub patterns and routing intent

- When using **Azure Virtual WAN (vWAN)**:
  - You should use secure hub constructs (Azure Firewall in hub) and routing intent/policies to enforce consistent inspection.
  - You should keep spoke/VNet routing intent aligned with the same principles: deterministic steering, minimal propagation, and explicit exceptions.
- If mixing vWAN and classic hub-spoke peering, you should document the routing domain boundaries and validate symmetry across both constructs.

9.4.4 Asymmetric routing detection and mitigation

- You should assume asymmetric routing will break stateful inspection (Azure Firewall/NVAs) and complicate incident response.
- Mitigations:
  - Enforce symmetric UDRs in both spokes for all steered prefixes.
  - Avoid “dual exit” designs (e.g., some subnets default to NAT Gateway while others default to firewall) unless you can prove symmetry per flow.
  - Use effective route analysis and flow logs to confirm both directions traverse the intended hop.

9.4.5 Zero-trust implications for routing decisions

- Routing is not authorization; it is traffic steering. In a zero-trust model:
  - You should steer traffic through enforcement points (Azure Firewall/NSG) and enforce explicit allowlists.
  - You should segment by explicit dimensions (workload/tier/environment) and ensure routing does not create implicit trust via broad reachability.

9.5 Validation

You should validate routing controls at three layers: intent, effective state, and runtime behavior.

1. **Intent validation (configuration)**
    1. Confirm route table associations match the subnet inventory (no “orphan” subnets without intended UDRs).
    2. Confirm UDR prefix specificity and next hop IPs (Azure Firewall private IP) are correct and version-controlled (IaC).
2. **Effective route validation (data plane)**
    1. Use NIC effective routes to confirm UDR overrides system routes as intended (including longest-prefix-match behavior).
    2. Confirm there is no unexpected route propagation into spokes (especially from BGP) that introduces alternative paths.
3. **Runtime validation (traffic)**
    1. Generate controlled test flows between spokes and verify logs show traversal through Azure Firewall (and expected rule hit).
    2. Validate symmetry: test bidirectional sessions (e.g., TCP) and confirm stateful continuity (no intermittent resets).
    3. Validate DNS paths where routing impacts Private Link resolution (ensure no public fallback due to missing **Azure Private DNS Zone** links).

9.6 Pitfalls

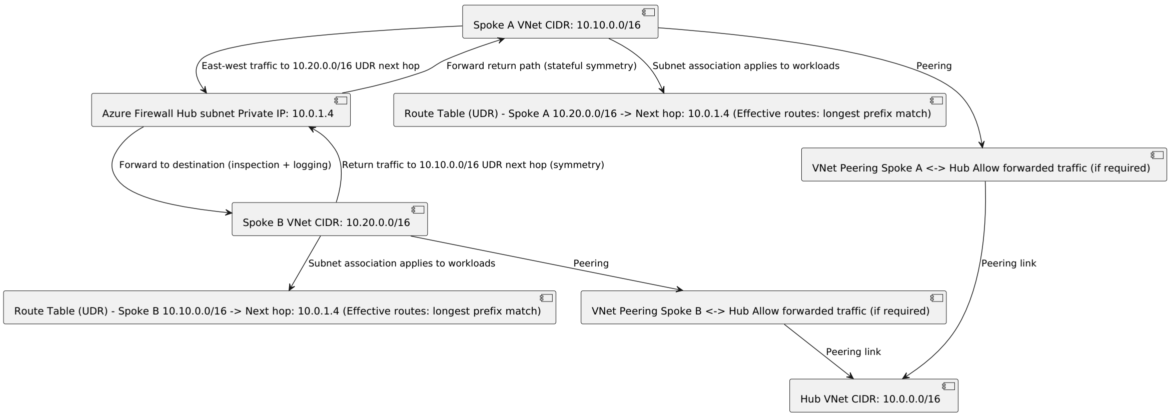
- **Route leaks** from BGP or mis-scoped propagation causing bypass of hub inspection.
- **Asymmetric routing** where only one direction is steered through Azure Firewall/NVA.
- **Over-broad exceptions** (e.g., bypassing inspection for large prefixes) that become de facto backdoors.
- **SNAT exhaustion** on Azure Firewall when forced tunneling increases outbound concurrency; symptoms often appear as intermittent outbound failures.
- **Hidden dependencies** (DNS, time sync, update endpoints) failing after forced tunneling, leading to emergency changes outside change control.

9.7 Audit evidence

You should retain evidence that demonstrates deterministic routing intent, controlled change, and operational verification:

- Route tables (UDRs) exported from Azure (resource JSON) showing:
    - `0.0.0.0/0` forced-tunnel routes (where applicable)
    - Spoke-to-spoke routes to firewall next hop
    - Exception routes and their justifications (ticket/change record linkage)
- Effective route reports for representative NICs per subnet category.
- Azure Firewall logs demonstrating inspected east-west and north-south flows (retain per policy).
- Change control artifacts:
    - IaC pull requests/approvals
    - Deployment logs and rollback plans
- BGP/prefix filtering configuration from edge devices (where in scope) and periodic route review reports.

9.8 Diagram — Routing with UDRs for Spoke-to-Spoke via Hub Inspection


Spoke A VNet CIDR: 10.10.0.0/16
East-west traffic to 10.20.0.0/16 UDR next hop
Forward return path (stateful symmetry)
Subnet association applies to workloads
Peering
Azure Firewall Hub subnet Private IP: 10.0.1.4
Route Table (UDR) - Spoke A 10.20.0.0/16 -> Next hop: 10.0.1.4 (Effective routes: longest prefix match)
Forward to destination (inspection + logging)
Return traffic to 10.10.0.0/16 UDR next hop (symmetry)
VNet Peering Spoke A <-> Hub Allow forwarded traffic (if required)
Spoke B VNet CIDR: 10.20.0.0/16
Subnet association applies to workloads
Peering
Peering link
Route Table (UDR) - Spoke B 10.10.0.0/16 -> Next hop: 10.0.1.4 (Effective routes: longest prefix match)
VNet Peering Spoke B <-> Hub Allow forwarded traffic (if required)
Peering link
Hub VNet CIDR: 10.0.0.0/16

9.8.1 Controls checklist (s09)

- [ ] You enforce forced tunneling (`0.0.0.0/0`) via **UDR** where required, with documented exceptions.
- [ ] You steer spoke-to-spoke prefixes symmetrically through **Azure Firewall** (or approved NVA) for east-west inspection.
- [ ] You validate effective routes on representative NICs and monitor for unexpected propagation/route leaks.

- [ ] You implement BGP prefix filtering and minimize route propagation into spokes.
- [ ] You capacity-plan **SNAT** for Azure Firewall under forced tunneling and verify via telemetry.
- [ ] You retain auditable artifacts: route table exports, effective route evidence, firewall logs, and change approvals.

10. s10 — DNS Security and Private Resolution

10.1 Objectives

- Secure name resolution for hybrid environments and Private Endpoint traffic paths.
- Prevent DNS-based data exfiltration, public DNS fallback, and misrouting caused by zone/forwarding errors.

10.2 When to use

- You deploy **Private Endpoint** (see glossary: Private Endpoint / Private Link) and must ensure PaaS Fully Qualified Domain Names (FQDNs) resolve to private IPs from all client networks.
- You operate **hub-spoke** (see glossary: Hub-spoke) with shared services in a hub VNet and want centralized, supportable DNS.
- You require hybrid resolution between on-premises and Azure using conditional forwarding with controlled east-west paths.

10.3 Design decisions

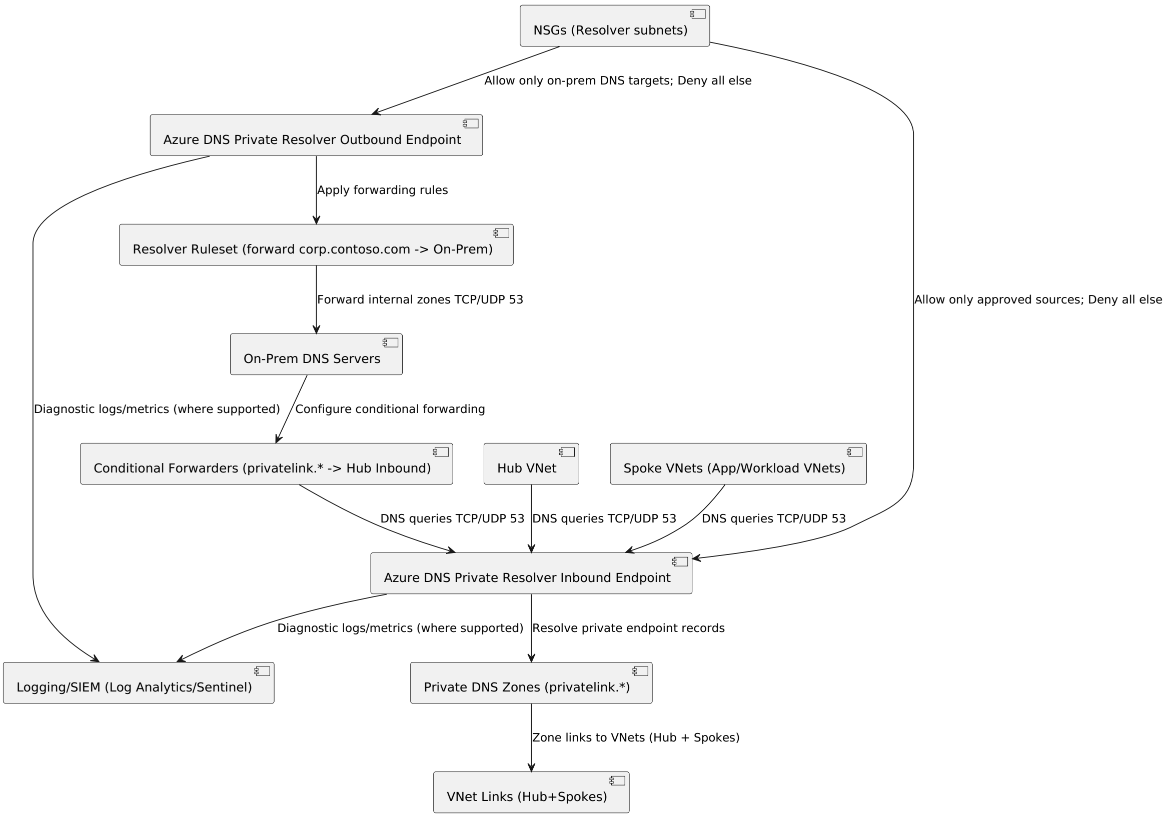
- **DNS model**
  - **Azure-provided DNS (168.63.129.16)** for basic VNet resolution (limited control; no custom conditional forwarding).
  - **Custom DNS servers** (IaaS) for full control, but you assume patching, HA, and audit responsibility.
  - **Azure DNS Private Resolver** (recommended) for managed inbound/outbound forwarding in Azure (see glossary: Azure DNS Private Resolver is implied by section; use as managed alternative to custom forwarders).

- **Private Endpoint DNS strategy**
  - Prefer **Azure Private DNS Zone** (see glossary: Azure Private DNS Zone) per service (`privatelink.<service-domain>`) linked to every VNet that originates queries.
  - Decide whether to centralize zones in the hub subscription/resource group or distribute per application landing zone (central is simpler; distributed reduces blast radius but increases governance overhead).
- **Hybrid conditional forwarding**
  - On-prem DNS should forward Azure private endpoint zones (e.g., `privatelink.*`) to the **inbound endpoint**.
  - Azure should forward on-prem internal zones (e.g., `corp.contoso.com`) via **outbound endpoint** rulesets to on-prem resolvers.
- **Threat-model alignment**
  - **Misconfiguration**: incorrect zone links/records causing public fallback or split-brain issues.
  - **Insider**: unauthorized DNS rule/zone changes enabling exfiltration via malicious resolvers.
  - **East-west traffic**: resolver paths enabling lateral discovery if resolver subnets are not restricted.
  - **External (north-south)**: DNS egress to Internet resolvers bypassing intended inspection/egress controls.

[!NOTE] **Security rationale**
You should treat DNS as a tier-0 dependency for Private Link: incorrect private resolution commonly leads to **silent public endpoint use**, undermining the intent of Private Endpoint and egress controls.

10.4 Implementation notes

10.4.1 Reference architecture: Hub-based Azure DNS Private Resolver (hybrid)


NSGs (Resolver subnets)
Allow only on-prem DNS targets; Deny all else
Azure DNS Private Resolver Outbound Endpoint
Apply forwarding rules
Resolver Ruleset (forward corp.contoso.com -> On-Prem)
Forward internal zones TCP/UDP 53
Allow only approved sources; Deny all else
On-Prem DNS Servers
Diagnostic logs/metrics (where supported)
Configure conditional forwarding
Conditional Forwarders (privatelink.* -> Hub Inbound)
Hub VNet
Spoke VNets (App/Workload VNets)
DNS queries TCP/UDP 53
DNS queries TCP/UDP 53
DNS queries TCP/UDP 53
Azure DNS Private Resolver Inbound Endpoint
Diagnostic logs/metrics (where supported)
Resolve private endpoint records
Logging/SIEM (Log Analytics/Sentinel)
Private DNS Zones (privatelink.*)
Zone links to VNets (Hub + Spokes)
VNet Links (Hub+Spokes)

10.4.2 Procedures (implementation)

1. **Create resolver subnets in the Hub VNet**
    1. Create dedicated subnets for inbound and outbound endpoints (do not share with workloads).
    2. Apply **Network Security Group (NSG)** (see glossary: Network Security Group (NSG)) to each resolver subnet.
2. **Deploy Azure DNS Private Resolver**
    1. Create a resolver resource in the hub.
    2. Create an **inbound endpoint** in the inbound subnet.
    3. Create an **outbound endpoint** in the outbound subnet.
3. **Create Private DNS Zones for Private Endpoints**
    1. For each Azure PaaS type used with Private Endpoint, create the correct `privatelink.*` zone.
    2. Link the zone to all client VNets (hub and spokes) that must resolve private IPs.

    3. Enable auto-registration only where appropriate (most Private Endpoint scenarios rely on explicit record creation).
4. **Configure forwarding**
    1. On-prem DNS: configure conditional forwarders for `privatelink.*` zones to the resolver **inbound endpoint IP(s)**.
    2. Azure: create a resolver ruleset that forwards on-prem/internal zones (e.g., `corp.contoso.com`) via the resolver **outbound endpoint** to on-prem DNS IPs.
    3. Associate the ruleset to the VNets that need hybrid resolution.
5. **Control DNS egress**
    1. Ensure workloads use intended resolvers (Azure-provided DNS or custom) and cannot bypass to arbitrary Internet resolvers.
    2. If you use forced tunneling (see glossary: Forced tunneling), ensure DNS paths are consistent with **UDR (User-Defined Route)](#)** and firewall policies.

> [!NOTE] **Security rationale**
> You should restrict resolver subnets with NSGs to known sources/targets to reduce **east-west discovery** and prevent misuse as an exfiltration channel (e.g., forwarding to unauthorized resolvers).

10.4.3 NSG baseline for resolver endpoint subnets (illustrative)

Assumptions (adapt as required):

- You have a hub VNet with separate inbound/outbound resolver endpoint subnets.
- Your on-prem DNS IPs and spoke address ranges are known and stable.
- You maintain deny-by-default posture.

Options/tradeoffs:

- Allow DNS from **all spokes** vs allow only from specific spoke subnets:
    - Broad allow is operationally simpler but increases blast radius.
    - Narrow allow is more secure but increases change overhead.

Minimum rules (conceptual):

- Inbound subnet NSG:
    - Allow TCP/UDP 53 from on-prem DNS forwarders and authorized spokes/hub.
    - Deny TCP/UDP 53 from all other sources.

- Outbound subnet NSG:
    - Allow TCP/UDP 53 to on-prem DNS targets.
    - Deny TCP/UDP 53 to Internet destinations.

  > [!WARNING] **Anti-pattern**
  > Linking Private DNS Zones only to the hub VNet and assuming spokes will resolve via peering. VNet peering does **not** provide transitive DNS; each client VNet must either link the zone or query a resolver that can resolve it.

10.5 Validation

You should validate both correctness (private resolution) and containment (no bypass).

1. **Private Endpoint resolution**
    1. From a spoke workload, query the PaaS FQDN and confirm it resolves to the private endpoint IP (not public).
    2. Repeat from on-prem clients if hybrid is in scope.
2. **Zone link coverage**
    1. Enumerate all VNets that originate queries to private endpoints.
    2. Verify each is linked to required **Azure Private DNS Zone**(s) or is configured to use resolvers that can resolve them.
3. **Forwarding correctness**
    1. Validate on-prem conditional forwarders point to inbound endpoint IPs.
    2. Validate outbound ruleset forwards only intended internal zones to on-prem DNS.
4. **Failure-mode testing**
    1. Temporarily disable a zone link in a test environment and confirm the failure mode is detected (alerts/health checks) rather than silently falling back to public resolution.
    2. Validate that forwarder loops do not occur (e.g., on-prem forwards a zone to Azure while Azure forwards same zone back on-prem).

10.6 Pitfalls

- **Public DNS fallback** due to missing zone links or incorrect client DNS settings; this undermines Private Link traffic intent.
- **Split-brain DNS** when the same zone exists on-prem and in Azure without a clear authoritative boundary.

- **Forwarder loops** between on-prem forwarders and Azure outbound rulesets.
- **Stale records** after private endpoint recreation; clients cache old IPs (plan TTL and cache flush procedures).
- **Over-permissive resolver NSGs**, enabling lateral movement and abuse of DNS forwarding.
- **Unmonitored change paths** (e.g., ad hoc zone edits) leading to misrouting and incident-response ambiguity.

10.7 Audit evidence

You should retain evidence that demonstrates intent, enforcement, and monitoring:

- Inventory of:
  - Azure DNS Private Resolver resources (inbound/outbound endpoints, IPs, subnets).
  - Private DNS Zones and VNet links (hub + spokes).
  - Resolver rulesets and forwarding targets.
- Configuration exports:
  - NSG rules applied to resolver endpoint subnets.
  - On-prem conditional forwarder configuration (change ticket + config snapshot).
- Operational records:
  - Change control approvals for DNS zone/link/ruleset modifications.
  - Logs/diagnostics configuration for resolver (where supported) and surrounding controls (e.g., firewall logs for DNS egress).
- Validation artifacts:
  - Test results showing private endpoint FQDNs resolve to private IPs from spokes and on-prem.
  - Evidence of alerting for anomalous DNS patterns where feasible.

10.7.1 Controls checklist (s10)

- [ ] Private Endpoint service-specific **Azure Private DNS Zone** created and named correctly (`privatelink.*`).
- [ ] Zone is linked to every VNet that requires private resolution (or those VNets query a resolver that can resolve it).
- [ ] Azure DNS Private Resolver deployed in hub with dedicated inbound/outbound subnets.
- [ ] NSGs applied to resolver subnets with explicit allowlists and deny-by-default.

- [ ] On-prem conditional forwarders configured for `privatelink.*` to inbound endpoint IPs.
- [ ] Outbound ruleset forwards only approved internal zones to on-prem DNS targets; no forwarder loops.
- [ ] Documented validation proves private resolution and no public fallback.
- [ ] DNS-related changes are governed (e.g., Azure Policy guardrails where materially relevant) and auditable via change records and configuration snapshots.

11. s11 — Protection for Compute and Platform Endpoints

11.1 Objectives

- You should reduce VM, AKS, and PaaS-adjacent endpoint exposure on the **data plane** while preserving operational access patterns.
- You should align endpoint controls with segmentation intent (macrosegmentation via hub-spoke + **UDR (User-Defined Route)**, microsegmentation via **NSG (Network Security Group)** + **ASG (Application Security Group)**).
- You should produce audit-grade evidence that endpoints are private-by-default, egress is controlled, and logs are retained and reviewable.

11.2 When to use

- You operate IaaS VMs and require management access without exposing RDP/SSH to the Internet.
- You run **AKS** (Azure Kubernetes Service) and need controlled egress, restricted API server access, and private dependencies.
- You consume PaaS (e.g., ACR, Key Vault, Storage) from compute and want to prevent public endpoint fallback using **Private Endpoint** and **Azure Private DNS Zone**.
- You must mitigate threats across: external (north-south), east-west (lateral), insider, supply chain, and misconfiguration.

11.3 Design decisions

- **Management access pattern (VMs)**
  - Prefer **Azure Bastion** for RDP/SSH without public IPs on target VMs.
  - Use **JIT (Just-In-Time) VM access** when organizational processes require time-bound port exposure (still avoid persistent public exposure).
  - Decide whether to centralize management in a dedicated “management subnet” (hub) or per-spoke (only when latency/ sovereignty requires).
- **AKS networking controls**
  - Choose a CNI and network policy strategy (e.g., Azure CNI + Azure Network Policy, or Calico) and treat it as complementary to NSGs (cluster microsegmentation vs VNet-level guardrails).
  - Decide AKS API server exposure: private cluster vs authorized IP ranges (data-plane impact: who can reach the API endpoint).
  - Choose egress control: forced tunneling to hub **Azure Firewall** via **UDR**, and explicitly allow required destinations.
- **Private dependencies**
  - Prefer **Private Endpoint** (Private Link) for ACR/Key Vault/Storage over **Service Endpoint** where you need private IP semantics and reduced public exposure.
  - Decide DNS ownership model: centralized private DNS in hub with links to spokes vs per-spoke zones (centralized is typically more auditable and consistent).

Security rationale: Eliminating public management endpoints, forcing egress through centralized inspection, and using Private Link with validated private DNS materially reduces north-south exposure, lateral movement paths, and misconfiguration risk (e.g., unintended public access), while improving auditability through centralized policy/logging.

11.4 Implementation notes

11.4.1 VM access hardening (RDP/SSH elimination)

Assumptions (adapt as required):

- Hub-spoke is in use; hub contains shared services including **Azure Firewall** and optionally **Azure Bastion**.

- Workload VMs reside in spoke subnets protected by NSGs (deny-by-default).

Procedure

1. You should remove public IPs from workload VM NICs unless a documented exception exists.
2. You should deploy **Azure Bastion** in the hub (or a dedicated management VNet) and use it for browser-based RDP/SSH.
3. You should enforce NSG rules on workload subnets:
    1. Allow management traffic only from the Bastion subnet (or approved management subnet).
    2. Deny inbound from Internet and untrusted spokes (explicit deny where high risk, e.g., dev→prod).
4. You should control outbound from workload subnets:
    1. Use **UDR** to force `0.0.0.0/0` to **Azure Firewall** for inspection/allowlisting.
    2. Monitor firewall **SNAT** utilization to prevent port exhaustion.
5. You should enable flow and firewall logging to Log Analytics/Sentinel for access traceability.

    Anti-pattern: Leaving RDP/SSH reachable via public IP "temporarily" and relying on credentials alone. This increases external and insider risk and makes exposure drift likely.

11.4.2 AKS endpoint protection (egress, API server, private dependencies)

Assumptions (adapt as required):

- AKS nodes are in a spoke subnet; hub hosts **Azure Firewall**.
- Private endpoints are created in a designated private endpoint subnet (spoke or shared services), with private DNS zones linked to all client VNets.

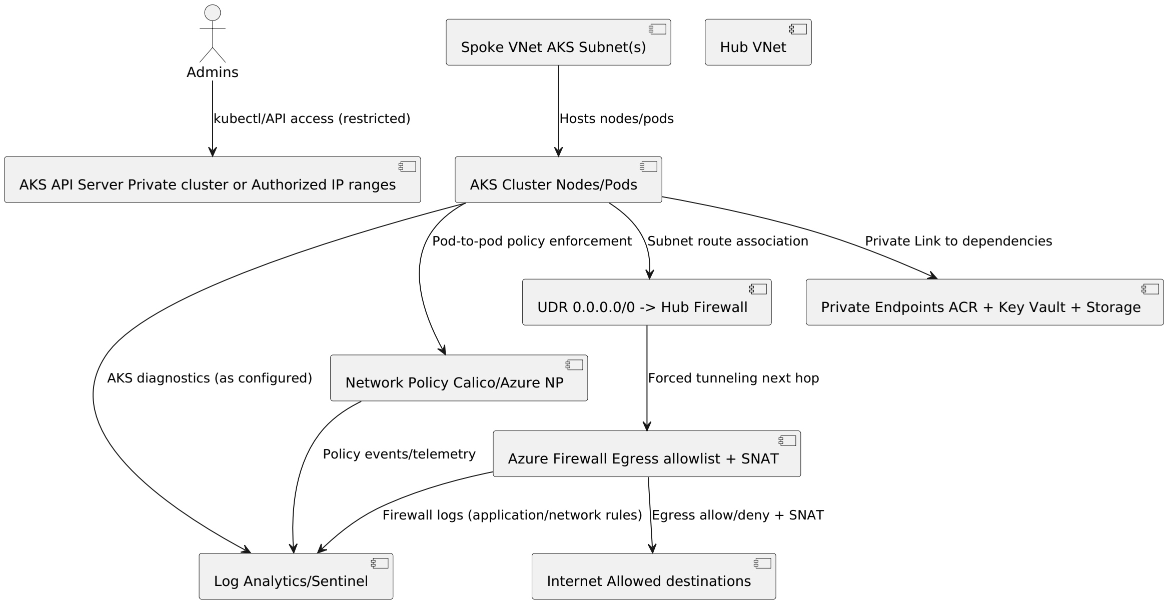
Procedure

1. You should enforce cluster-to-Internet egress via hub firewall:
    1. Associate a route table with the AKS subnet.
    2. Add UDR `0.0.0.0/0 -> Virtual appliance (Azure Firewall private IP)` for forced tunneling.
    3. Implement firewall allowlists for required FQDNs/IPs (minimize broad outbound).

2. You should restrict AKS API server access:
    1. Prefer private cluster where operationally feasible, or
    2. Configure authorized IP ranges (only approved admin networks).
3. You should apply network policy inside the cluster (Calico/Azure NP) to reduce pod-to-pod lateral movement beyond what NSGs can express.
4. You should consume platform dependencies via Private Link:
    1. Create **Private Endpoint** for ACR, Key Vault, Storage.
    2. Configure and link the appropriate **Azure Private DNS Zone** to all VNets that host AKS nodes and build agents.
    3. Validate name resolution returns private endpoint IPs to prevent public fallback.
5. You should log:
    - Firewall logs (egress decisions, DNAT/SNAT outcomes where relevant)
    - AKS control-plane/diagnostic logs as required to investigate network policy and egress denials
    - Private DNS query patterns where available (or validate via periodic synthetic checks)

    Security rationale: Combining VNet-level forced tunneling (macrosegmentation + inspection) with in-cluster network policy (microsegmentation) addresses east-west and misconfiguration threats that either layer alone will not fully mitigate.

11.4.3 Diagram — AKS Network Security: Controlled Egress and Private Dependencies


Admins
Spoke VNet AKS Subnet(s)
Hub VNet
kubectl/API access (restricted)
Hosts nodes/pods
AKS API Server Private cluster or Authorized IP ranges
AKS Cluster Nodes/Pods
Pod-to-pod policy enforcement
Subnet route association
Private Link to dependencies
UDR 0.0.0.0/0 -> Hub Firewall
Private Endpoints ACR + Key Vault + Storage
AKS diagnostics (as configured)
Network Policy Calico/Azure NP
Forced tunneling next hop
Policy events/telemetry
Azure Firewall Egress allowlist + SNAT
Firewall logs (application/network rules)
Egress allow/deny + SNAT
Log Analytics/Sentinel
Internet Allowed destinations

11.5 Validation

- **VMs**
    - You should confirm no public IPs are attached to workload VM NICs (inventory + policy compliance).
    - You should validate NSG effective rules allow management only from Bastion/management subnet and deny Internet inbound.
    - You should validate effective routes show `0.0.0.0/0` to the firewall where forced tunneling is required.
- **AKS**
    - You should validate AKS subnet effective routes include UDR to firewall and that egress succeeds only to allowlisted destinations.
    - You should validate API server restriction is enforced (private endpoint reachability or authorized IP range enforcement).
    - You should validate Private Endpoint DNS: resolve ACR/Key Vault/Storage FQDNs from AKS nodes/pods and confirm private IP results (no public fallback).
- **Logging**
    - You should validate firewall logs and relevant diagnostics are arriving in Log Analytics/Sentinel with required retention.

11.6 Pitfalls

- **SNAT exhaustion** on Azure Firewall when forced tunneling all cluster egress; you should capacity-plan and monitor SNAT utilization (symptom: intermittent outbound failures).
- **Private DNS mislinking** causing public endpoint fallback for Private Link services; you should treat DNS linkage/conditional forwarding as required for security, not optional.
- **Over-reliance on NSGs** for pod-level controls; NSGs operate at subnet/NIC granularity, so you should still enforce in-cluster network policy.
- **Origin bypass in platform exposure**: if you publish services, you should ensure ingress patterns (WAF/ILB/private ingress) cannot be bypassed by direct public access paths.

11.7 Audit evidence

- Azure Policy assignments and compliance results for:
    - Deny/limit public IPs on NICs (where applicable)
    - Require diagnostics on Azure Firewall and relevant resources

    - Guardrails enforcing Private Endpoint usage/DNS linkage (where feasible)
- NSG configurations and effective security rules for workload and management subnets (exported snapshots with timestamps).
- Route tables/UDRs showing forced tunneling (`0.0.0.0/0 -> Azure Firewall`) and effective routes from representative NICs.
- Azure Firewall policy/rule collections and logs demonstrating egress allowlisting decisions.
- Private Endpoint resources + **Azure Private DNS Zone** records and VNet links; validation outputs showing private resolution.
- Log Analytics/Sentinel workbooks/queries showing management access, denied egress attempts, and investigation trails.

11.8 Controls checklist (s11)

- [ ] Workload VMs have no public IPs; management is via **Azure Bastion** and/or time-bound **JIT (Just-In-Time) VM access**.
- [ ] NSGs are deny-by-default and use **ASG (Application Security Group)** where appropriate; inbound from Internet is denied.
- [ ] Spoke egress is forced through hub **Azure Firewall** via **UDR (User-Defined Route)**; SNAT capacity is monitored.
- [ ] AKS API server exposure is restricted (private cluster or authorized IP ranges) and reviewed periodically.
- [ ] AKS network policy (Calico/Azure NP) is enforced for pod-level segmentation.
- [ ] ACR/Key Vault/Storage access uses **Private Endpoint** with validated **Azure Private DNS Zone** resolution (no public fallback).
- [ ] Firewall/flow/diagnostic logs are centralized to Log Analytics/Sentinel with retention and access controls suitable for audit.

12. s12 — Observability: Logs, Flows, and Security Monitoring

12.1 Objectives

- You should enable actionable visibility into network security posture and events across **north-south traffic** and **east-west traffic**.

- You should support threat detection and audit evidence collection with centralized logging, retention controls, and repeatable queries.
- You should reduce **misconfiguration** and improve response time by monitoring rule effectiveness (e.g., denied flows, rule hits, WAF blocks).

12.2 When to use

- You operate a **hub-spoke** environment with centralized inspection (e.g., **Azure Firewall** and **forced tunneling** via **UDR (User-Defined Route)**).
- You rely on **NSG (Network Security Group)** controls for microsegmentation and need evidence of enforcement and traffic patterns.
- You expose HTTP/HTTPS through **WAF** (Application Gateway WAF or Azure Front Door WAF) and must tune and prove efficacy.
- You require centralized detection/response via Microsoft Sentinel and audit-ready retention/immutability.

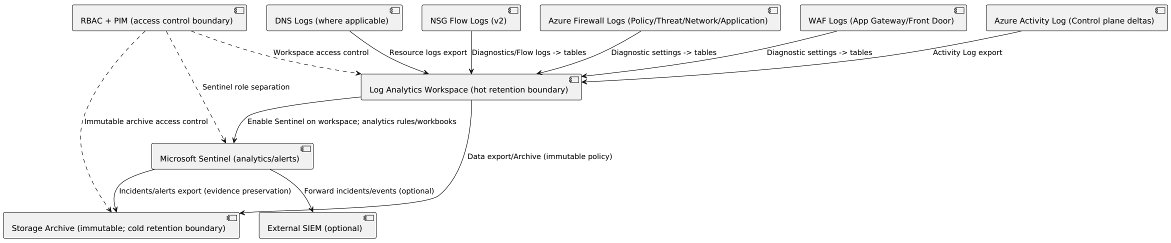
12.3 Design decisions

- **Centralize analytics** in one or few **Log Analytics workspaces** aligned to tenant/subscription boundaries and data residency.
- **Separate hot analytics vs cold archive**:
  - Hot: Log Analytics + Microsoft Sentinel for detection and triage.
  - Cold: Storage archive with immutability for audit/legal hold and cost optimization.
- **Decide per log source** which data is required for:
  - Security detection (near-real-time).
  - Troubleshooting (short-term, verbose).
  - Audit evidence (long-term, immutable).
- **Access control model**:
  - Least privilege via Azure RBAC on workspaces, Sentinel, and storage.
  - Privileged actions gated via PIM (Privileged Identity Management) for log settings, diagnostic settings, and data export.

  Security rationale: Centralizing network telemetry (flows, firewall, WAF, and control-plane deltas) reduces detection gaps for **external (north-south)**, **east-west (lateral movement)**, **insider**, and **misconfiguration** threats, while supporting evidence integrity through immutability and access controls.

12.4 Implementation notes

12.4.1 Architecture: end-to-end logging pipeline


RBAC + PIM (access control boundary)
DNS Logs (where applicable)
NSG Flow Logs (v2)
Azure Firewall Logs (Policy/Threat/Network/Application)
WAF Logs (App Gateway/Front Door)
Azure Activity Log (Control plane deltas)
Workspace access control
Resource logs export
Diagnostics/Flow logs -> tables
Diagnostic settings -> tables
Diagnostic settings -> tables
Activity Log export
Sentinel role separation
Log Analytics Workspace (hot retention boundary)
Immutable archive access control
Enable Sentinel on workspace; analytics rules/workbooks
Microsoft Sentinel (analytics/alerts)
Data export/Archive (immutable policy)
Incidents/alerts export (evidence preservation)
Forward incidents/events (optional)
Storage Archive (immutable; cold retention boundary)
External SIEM (optional)

12.4.2 Log source configuration (data-plane focus, control-plane where material)

1. You should enable **NSG flow logs v2** on all workload and shared-services NSGs.
2. You should enable Azure Firewall diagnostic logs and metrics on the hub firewall(s).
3. You should enable WAF logs (and metrics) on Application Gateway WAF and/or Azure Front Door WAF.
4. You should export **Azure Activity Log** to Log Analytics to detect material configuration changes impacting the data plane (NSG/route table/firewall/WAF/DNS/Private Endpoint changes).
5. You should enable DNS-related logging **where available** for your DNS pattern (e.g., DNS Private Resolver query logs, firewall DNS proxy logs), and document gaps explicitly.

Assumptions (adapt as needed):

- You use at least one centralized Log Analytics workspace per environment/landing zone.
- Diagnostic settings are deployed via IaC (Bicep/Terraform) and protected by Azure Policy where possible.
- You have data residency requirements defined for workspace region and storage account region.

12.4.3 Microsoft Sentinel integration

1. You should enable Microsoft Sentinel on the workspace that receives network security telemetry.

2. You should connect data sources via supported connectors (Log Analytics-based sources are typically “connected” by virtue of being in the workspace; ensure correct solutions/content are enabled).
3. You should implement detections mapped to threat model categories:
    - **External (north-south):** WAF blocks/spikes, inbound DNAT anomalies, known bad IPs.
    - **East-west (lateral movement):** unusual intra-spoke flows, denied east-west, new high-volume ports.
    - **Insider:** configuration changes + traffic anomalies from admin subnets.
    - **Supply chain:** unexpected egress to new domains/IPs from build agents.
    - **Misconfiguration:** sudden increase in allowed inbound, public exposure drift, missing logs.
4. You should use Sentinel workbooks for operational visibility (top talkers, denied flows, firewall rule hits, WAF top rules, blocked geos).

12.4.4 Retention, immutability, and evidence handling

- You should define and document:
    - Workspace hot retention (operational + detection needs).
    - Archive retention in immutable storage (audit/regulatory needs).
    - Incident evidence retention (cases, alerts, supporting logs).
- You should implement immutability in storage (e.g., immutable blob policies) and restrict deletion/alteration permissions.
- You should treat log settings as security controls:
    - Protect diagnostic settings changes with least privilege and PIM.
    - Consider Azure Policy to require diagnostic settings on NSG/Firewall/WAF resources (control-plane governance only where it prevents data-plane visibility regression).

  Security rationale: Immutable archival and strict access controls mitigate **insider** and **misconfiguration** risks (e.g., tampering with evidence, disabling diagnostics) and strengthen audit defensibility.

12.4.5 Rule efficacy and tuning loops (operational procedures)

1. You should perform weekly rule-hit reviews:
    1. Identify Azure Firewall rules with no hits (candidate for removal) and rules with unexpected hits (candidate for tightening).
    2. Identify NSG denies/allows that contradict intended segmentation.
    3. Identify WAF rules generating high false positives and tune via exclusions with documented justification.

2. You should baseline normal traffic patterns per environment and alert on deviations:
    - New outbound destinations (by FQDN/IP category where feasible).
    - Port/protocol drift.
    - High-volume denied traffic (scans, misroutes, broken dependencies).

   Anti-pattern: Treating “logging enabled” as sufficient without ongoing review; this produces high cost, low signal, and weak audit narratives because you cannot show controls are monitored and improved.

12.5 Validation

You should validate end-to-end telemetry using a mix of configuration checks and data checks.

1. **Configuration validation (control-plane checks that protect data-plane visibility):**

    1. Confirm diagnostic settings exist for NSG/Firewall/WAF and point to the intended Log Analytics workspace.
    2. Confirm Activity Log export is enabled to the workspace.
    3. Confirm storage archive configuration (if used) and immutability policy status.
    4. Confirm RBAC assignments and PIM activation requirements for log configuration changes.

2. **Data validation (data-plane evidence):**

    1. Generate a known test flow across an NSG boundary and confirm it appears in flow logs.
    2. Trigger a firewall-logged event (e.g., permitted outbound via an application rule) and confirm records arrive.
    3. Trigger a WAF event in a controlled test app and confirm WAF logs/metrics.
    4. Confirm Sentinel can query the workspace and create an alert from a test analytic rule.

12.6 Pitfalls

- Missing NSG flow logs on “internal-only” subnets, causing blind spots for **east-west traffic** and lateral movement.

- Centralized egress via Azure Firewall without monitoring **SNAT** utilization signals; port exhaustion can look like random outages and won't be diagnosable without the right telemetry.
- WAF tuning without change control; exclusions become permanent and silently reduce protection.
- Over-permissive access to workspaces/storage, enabling log tampering and weakening audit confidence.

  Anti-pattern: Storing long-term audit logs only in Log Analytics without an immutable archive strategy, then discovering retention/cost constraints during an audit or incident.

12.7 Audit evidence

You should be able to produce the following on request:

- Inventory of enabled diagnostic settings for NSGs, Azure Firewall, WAF resources, and Activity Log export (resource IDs, destinations, categories).
- Proof of log arrival and continuity:
  - Sample queries showing recent events per source.
  - Evidence of coverage across hub and spokes (representative NSGs/subnets).
- Sentinel configuration artifacts:
  - Enabled analytics rules, incidents, workbooks (exported configurations where possible).
- Retention and immutability evidence:
  - Workspace retention settings (and any archival/export configuration).
  - Storage immutability policy configuration and RBAC showing restricted delete/modify rights.
- Change control evidence:
  - Activity Log entries showing who changed logging settings (or proof of no changes), with PIM activation records where applicable.

12.7.1 Controls checklist (s12)

- [ ] NSG flow logs v2 enabled for all NSGs with consistent destination and retention.
- [ ] Azure Firewall diagnostics enabled (logs + metrics) to Log Analytics.
- [ ] WAF diagnostics enabled (logs + metrics) to Log Analytics; tuning process documented.

- [ ] Azure Activity Log exported to Log Analytics for material network-security changes.
- [ ] Sentinel enabled on the workspace; detections aligned to threat model categories.
- [ ] Retention defined for hot analytics and cold immutable archive; deletion rights restricted via RBAC + PIM.
- [ ] Operational review loop implemented (rule hits, denies, anomalies) with evidence of execution.

13. Secure Operations, Incident Response, and Continuous Improvement

13.1 Objectives

- You should operationalize Azure networking data plane controls (e.g., NSG for Network Security Group, Azure Firewall, UDR for User-Defined Route) using repeatable, auditable processes.
- You should prepare for and execute incident response actions that affect network paths and enforcement points, with controlled rollback.
- You should continuously improve control quality using metrics tied to the threat model (external (north-south), east-west (lateral movement), insider, supply chain, misconfiguration).

13.2 When to use

- You operate a hub-spoke architecture with forced tunneling (spokes route `0.0.0.0/0 -> Azure Firewall`) and require rapid containment options.
- You need clear separation between “outage” runbooks and “security incident” runbooks to reduce mis-triage.
- You must produce audit evidence for changes to network enforcement (NSG, Azure Firewall policies, routing) and for incident response actions.

13.3 Design decisions

- **Runbooks are split by intent**:
  - **Outage runbooks**: restore service safely; avoid expanding exposure.

    - **Security incident runbooks**: contain first (stop lateral movement/exfiltration), then eradicate and recover.
- **Containment is layered** (apply the least-disruptive effective control first, escalate as needed):
    - NSG rule changes (subnet/NIC) for microsegmentation and emergency deny.
    - Azure Firewall policy changes for centralized egress/ingress blocks (including DNAT removal if needed).
    - Routing isolation (UDR/peering controls) to isolate a spoke/subnet when compromise is suspected.
- **Version-controlled configuration**:
    - Azure Firewall Policy, NSG rules, route tables, and Private Link/DNS artifacts should be managed via infrastructure-as-code (IaC) with immutable releases and break-glass exception paths.
- **Evidence-first operations**:
    - Every containment action should produce artifacts (change request, before/after effective rules/routes, log queries, timestamps) suitable for audit and forensics.

Security rationale: Separating outage vs incident workflows reduces the probability of “availability-first” actions that inadvertently widen blast radius (e.g., broad allow rules) during active compromise (misconfiguration + insider/lateral movement threat categories).

13.4 Implementation notes

13.4.1 Runbook structure: outage vs security incident

You should standardize both runbook types to include:

- Scope (subscription/resource groups, hub vs spoke, affected subnets/ASGs).
- Preconditions (required roles, approvals, break-glass usage criteria).
- Step-by-step actions (numbered, with explicit rollback).
- Validation checks (effective rules/routes + traffic verification).
- Evidence capture steps (queries, exports, screenshots where necessary).

Decision tree (triage):

1. Determine if symptoms indicate **availability degradation** (packet loss, latency, route flap) or **security** (unexpected east-west, C2 egress, abnormal DNS, suspicious flows).

2. If unclear, treat as security until proven otherwise (containment actions should be reversible and scoped).

   Anti-pattern: Using an outage runbook to “temporarily allow all egress” from a spoke to restore function without verifying whether the triggering symptom is exfiltration blocking.

13.4.2 Emergency access and containment mechanisms

You should pre-stage containment primitives with known blast radius:

1. **Quarantine NSG pattern (subnet-level)**

   - Create an “emergency quarantine” rule set that:
     - Denies all inbound from VNet and peered VNets except from a management subnet (e.g., Azure Bastion) if needed.
     - Denies all outbound except to evidence/management endpoints required for response (e.g., log collectors, update repos via approved egress).
   - Prefer referencing ASG (Application Security Group) where possible to avoid brittle IP targeting.

2. **Isolation spoke pattern (routing + peering controls)**

   - For high-risk compromise, you should isolate an entire spoke by:
     - Removing or disabling routes that enable transit to other spokes/hub services, or
     - Applying UDRs to blackhole (`next hop: None`) east-west prefixes while keeping minimal management reachability.
   - Ensure forced tunneling remains consistent with containment intent (avoid accidental direct Internet egress).

3. **Azure Firewall rapid blocks (centralized policy)**

   - You should implement an “Incident Response” rule collection group with:
     - Highest priority deny rules for known-bad destinations, ports, and suspicious east-west patterns.
     - Fast rollback mechanism (policy version or deployment slot) to revert after validation.
   - For inbound, remove/disable DNAT rules that expose suspected compromised origins.

Security rationale: Layered containment reduces time-to-contain for east-west (lateral movement) while limiting reliance on any single control that may be mis-scoped or bypassed (misconfiguration threat category).

13.4.3 Backups and configuration versioning

You should implement:

- **IaC state protection**:
  - Remote state with immutability/soft-delete where supported; separate break-glass state changes from standard pipelines.
- **Azure Firewall Policy versioning**:
  - Treat policy updates as releases: tag, record diffs, and keep a revertable prior version.
- **Exportable snapshots** (illustrative; adapt to your tooling):
  - Periodic export of NSG effective rules, route tables, Azure Firewall policy JSON for point-in-time reconstruction.

Anti-pattern: Making emergency portal edits to Azure Firewall policy without capturing “before” state and without a defined rollback package.

13.4.4 Penetration testing considerations (network controls)

You should establish rules of engagement that:

1. Identify permitted test sources/destinations and time windows.
2. Pre-register test IPs/FQDNs in Azure Firewall/WAF allowlists only if required and time-bounded.
3. Define monitoring expectations (what alerts should fire, which should be suppressed, and how suppression is documented).
4. Require post-test cleanup verification (remove temporary rules; validate effective routes and policies).

13.4.5 Metrics/KPIs for continuous improvement

You should track:

- **Policy compliance**: % of spokes with forced tunneling UDRs correctly applied; % of subnets with explicit egress NSG posture.
- **Rule churn**: number of NSG/Azure Firewall rule changes per week; emergency changes vs planned.

- **Alert fidelity**: ratio of true positives to total network alerts; top noisy detections.
- **Time-to-contain (TTC)**: detection timestamp to containment confirmation (traffic stopped).
- **Misconfiguration rate**: incidents caused by incorrect UDR/NSG/Azure Firewall changes, measured via post-change reviews.

13.5 Validation

You should validate containment and operational readiness using repeatable checks:

1. **Effective enforcement**

    1. Confirm NSG effective security rules at NIC and subnet scopes (deny rules are taking precedence per priority).
    2. Confirm effective routes for impacted subnets (UDR is applied as intended; no unintended Internet egress path).
    3. Confirm Azure Firewall logs show denies for the intended traffic (and no unexpected allows).

2. **Traffic validation**

    1. Execute controlled connectivity tests between suspected source and target (east-west) and to known external endpoints (north-south) as applicable.
    2. Validate that business-critical paths are either intentionally preserved or intentionally severed per incident decision.

3. **Rollback validation**

    1. Revert containment in a staged manner (e.g., firewall policy rollback before route re-open) and re-test connectivity.
    2. Confirm alerts return to baseline and no residual temporary rules exist.

13.6 Pitfalls

- Containment blocks business-critical services without pre-defined exception handling, causing prolonged outage.
- Forced tunneling assumptions break during incident actions (e.g., removing UDRs restores direct Internet egress).

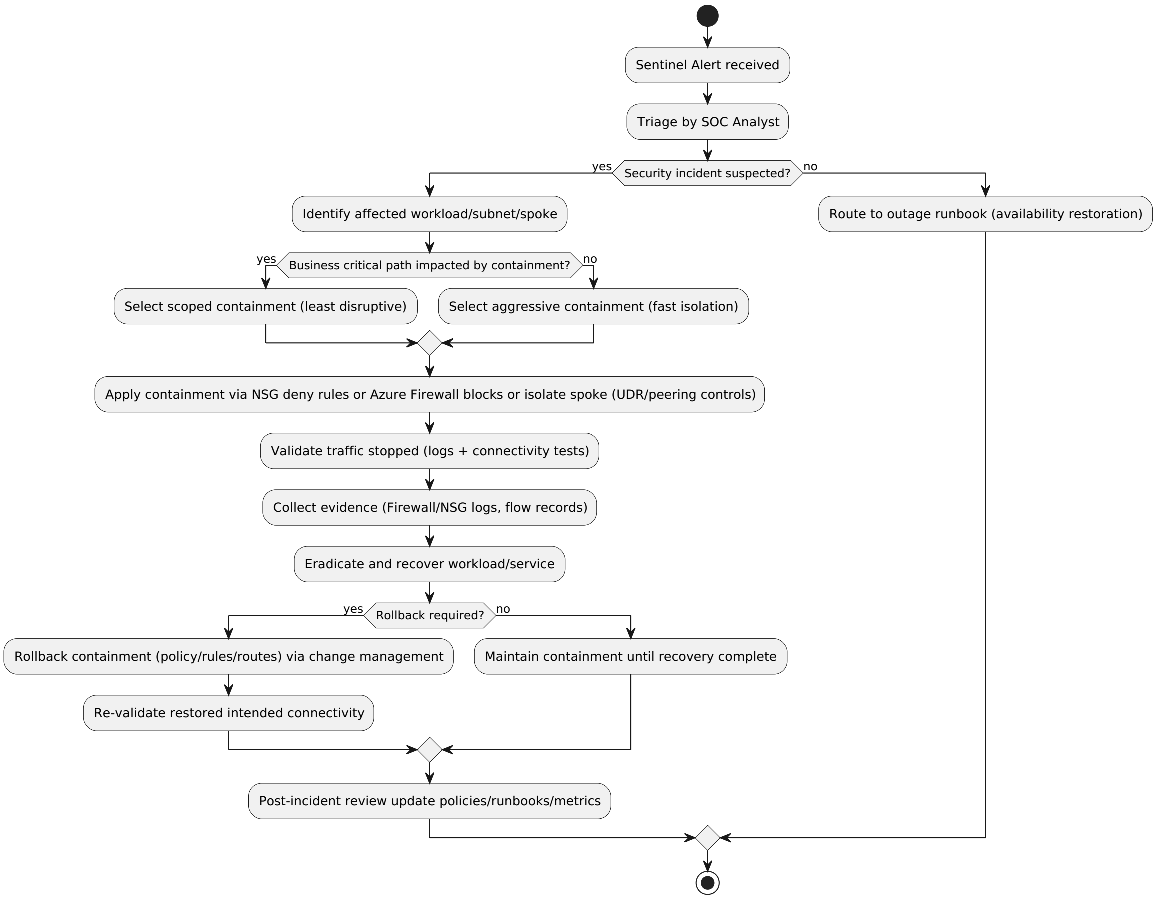
- DNS private resolution issues cause “public DNS fallback” for Private Link if Azure Private DNS Zone links are incomplete, undermining containment expectations for PaaS access paths.
- Emergency changes are made without evidence capture, impeding forensics and audit defensibility.

  Anti-pattern: Isolating a spoke by removing peering without preserving a management path, then losing the ability to collect host/network evidence.

13.7 Audit evidence

You should retain and be able to produce:

- Change records for NSG/Azure Firewall/UDR modifications (request/approval, executor identity, timestamps).
- “Before/after” snapshots:
  - NSG rules and effective rules for impacted NICs/subnets.
  - Route tables and effective routes for impacted subnets.
  - Azure Firewall policy version/diff and deployed rule collection priorities.
- Logs proving action and outcome:
  - NSG flow logs (where enabled), Azure Firewall logs (network/application), and relevant analytics queries in Microsoft Sentinel.
- Incident timeline:
  - Alert ID, triage notes, containment decisions (business criticality), validation results, rollback actions, and post-incident improvements.


Sentinel Alert received
Triage by SOC Analyst
Security incident suspected?
yes
no
Identify affected workload/subnet/spoke
Route to outage runbook (availability restoration)
Business critical path impacted by containment?
yes
no
Select scoped containment (least disruptive)
Select aggressive containment (fast isolation)
Apply containment via NSG deny rules or Azure Firewall blocks or isolate spoke (UDR/peering controls)
Validate traffic stopped (logs + connectivity tests)
Collect evidence (Firewall/NSG logs, flow records)
Eradicate and recover workload/service
Rollback required?
yes
no
Rollback containment (policy/rules/routes) via change management
Maintain containment until recovery complete
Re-validate restored intended connectivity
Post-incident review update policies/runbooks/metrics

13.7.1 Controls checklist (s13)

- [ ] You maintain separate, versioned outage and security incident runbooks for network controls.
- [ ] You pre-stage quarantine NSG patterns and an Azure Firewall “Incident Response” rule group with defined priority and rollback.
- [ ] You can isolate a spoke/subnet using UDR/peering controls while preserving a documented management/evidence path.
- [ ] You capture “before/after” effective rules/routes and retain Firewall/NSG logs in a centralized store (e.g., Log Analytics/Microsoft Sentinel) with audit-aligned retention.
- [ ] You measure TTC, rule churn, and misconfiguration-driven incidents and feed outcomes into continuous improvement.

14. s14 — Compliance Mapping and Audit Readiness

14.1 Objectives

- You should map Azure networking **data plane** controls to common compliance expectations (ISO 27001, SOC 2, PCI DSS) in a repeatable structure.
- You should produce auditor-ready evidence for segmentation, ingress/egress control, private access patterns, and monitoring/retention.
- You should document statements for network boundaries, data residency impacts, and shared responsibility.

14.2 When to use

- You are preparing for (or responding to) ISO 27001 certification audits, SOC 2 Type II examinations, PCI DSS assessments, or internal risk reviews.
- You need to demonstrate that hub-spoke segmentation, **forced tunneling**, **Azure Firewall**, **Network Security Group (NSG)**, and **Private Endpoint** controls are operating effectively.
- You are standardizing evidence collection across subscriptions/regions/workloads.

14.3 Design decisions

- **Control mapping granularity**
  - **High-level mapping** (recommended for most audits): map to control families (network security, change management, logging/monitoring, access control) with pointers to evidence.
  - **Requirement-by-requirement mapping** (use for PCI DSS or strict customer assurance): map each requirement to a specific Azure control, configuration baseline, and evidence query/export.
- **Evidence sources of truth**
  - Prefer **Azure Policy** compliance state for guardrails that materially impact network security (e.g., public IP restrictions, diagnostic settings enforcement).
  - Prefer centralized telemetry in Log Analytics/Microsoft Sentinel for operational effectiveness (Firewall/NSG/WAF logs).

- **Architecture narrative**
    - Anchor your narrative on **hub-spoke** with routing intent enforced by **UDR (User-Defined Route)** and inspection at **Azure Firewall**.
    - State explicit assumptions and exceptions (e.g., workloads that cannot be forced-tunneled; sanctioned direct egress).

  Security rationale: Auditors typically validate (1) design intent, (2) configuration enforcement, and (3) operating effectiveness over time. Using Policy + centralized logs + route/rule inventories provides coverage across all three with minimal manual evidence.

14.4 Implementation notes

14.4.1 1) Control mapping structure (audit-friendly template)

You should structure your mapping table using these columns (minimum viable set):

- Compliance framework and clause (e.g., ISO 27001 A.8.x, SOC 2 CCx.x, PCI DSS 1.x)
- Control intent (plain language)
- Azure networking control(s) (data plane focus)
- Implementation location (subscription, hub VNet, spoke VNets)
- Configuration baseline reference (e.g., “Firewall Policy v1.3”, “NSG Baseline 2026-01”)
- Evidence source(s) and retrieval steps
- Test procedure (what the auditor can repeat)
- Exception handling (approved deviations; expiry; compensating controls)

14.4.2 2) High-level mapping (typical examples)

You should map common expectations as follows (illustrative; adapt to your framework version and scoping):

- **Network boundary and ingress/egress control**
    - Controls: Azure Firewall, forced tunneling via UDR, NSG deny-by-default, Azure Front Door/WAF (if applicable)
    - Threats: external (north-south), misconfiguration
- **Internal segmentation (east-west)**
    - Controls: hub-spoke isolation, NSG microsegmentation, **ASG (Application Security Group)** targeting, limited peering, UDR constraints

    - Threats: east-west traffic, insider
- **Private access to PaaS**
    - Controls: **Private Endpoint** (Private Link), **Azure Private DNS Zone** linkage, disable/limit public network access on PaaS where feasible
    - Threats: supply chain, misconfiguration, data exfiltration paths
- **Logging/monitoring and retention**
    - Controls: NSG flow logs, Azure Firewall logs, WAF logs, gateway diagnostics, centralized Log Analytics/Sentinel retention controls
    - Threats: detection gaps, insider, incident response readiness
- **Change control and configuration enforcement (only where it impacts data plane security)**
    - Controls: Azure Policy, resource locks where appropriate, Activity Log review for network changes
    - Threats: misconfiguration, unauthorized change

Anti-pattern: Treating screenshots as primary evidence. Screenshots are non-repeatable and time-bound; prefer exportable compliance states (Policy) and queryable logs with immutable retention controls.

14.4.3 3) Evidence sources and how you should collect them

You should standardize evidence collection per control type:

1. **Azure Policy compliance (guardrails)**

    - Evidence: policy assignments, initiatives, compliance results, remediation tasks
    - Use for: “public IP prohibited,” “diagnostic settings required,” “Private Endpoint required” (where adopted)

2. **Azure Activity Log (control plane change traceability)**

    - Evidence: write operations for route tables, NSGs, Firewall Policy, public IP creation, private endpoint changes
    - Use for: change management sampling and segregation-of-duties narratives (who changed what, when)

3. **Azure Firewall logs**

    - Evidence: application/network rule hits, DNAT usage, threat intelligence hits (if enabled), IDPS/TLS inspection outcomes (if applicable)
    - Use for: operating effectiveness of forced tunneling and egress controls

4. **NSG flow logs**

    - Evidence: allowed/denied flows by 5-tuple, verified segmentation behavior over time
    - Use for: east-west controls validation and incident reconstruction

5. **WAF logs (Azure Front Door WAF / Application Gateway WAF, if in scope)**

    - Evidence: blocked requests, matched rules, tuning history
    - Use for: L7 protection effectiveness

6. **Routing intent and effective routes**

    - Evidence: route table exports + “effective routes” snapshots for representative NICs/subnets
    - Use for: proving forced tunneling and next-hop enforcement

7. **Private Link and DNS correctness**

    - Evidence: private endpoint inventory, private DNS zone links, DNS resolution test results showing private IP mapping
    - Use for: preventing public DNS fallback

14.4.4 4) Segregation of duties and privileged access reviews (network-relevant)

You should provide evidence that administrative capability over network security controls is restricted and reviewed:

- Role assignments for network management scopes (subscription/resource group for hub networking)
- Evidence of periodic access reviews (where implemented) for roles that can modify:
    - NSGs/ASGs
    - Route tables/UDRs
    - Azure Firewall/Firewall Policy
    - Private Endpoints and Private DNS Zone links
- Change approval linkage (ticket IDs) for sampled modifications

    Security rationale: Many audit findings stem from excessive privileges that allow silent weakening of segmentation/egress controls; role scoping and review evidence addresses insider and misconfiguration risk.

14.4.5 5) Data residency and network boundary statements

You should document:

- Where traffic is inspected (region of hub Azure Firewall, central logging workspace region)
- Where logs are stored and retained (workspace location, retention/immutability settings)
- Whether any components are global (e.g., Azure Front Door is global; document implications for request handling and logging)
- A boundary statement defining:
    - In-scope VNets/subscriptions
    - Ingress points (Front Door/WAF, VPN/ExpressRoute, published public IPs)
    - Egress points (Azure Firewall SNAT, approved direct egress exceptions)

14.4.6 6) Auditor-friendly diagrams and narratives (recommended set)

You should maintain (and version-control) these diagrams and one-page narratives:

- Hub-spoke network overview (hub services: firewall, gateways, DNS, bastion; spokes and peering/vWAN)
- Ingress and egress flows (north-south) showing inspection points and logging
- East-west segmentation view (NSG/ASG intent per tier)
- Private access to PaaS flow (Private Endpoint + Private DNS Zone resolution path)

Assumption: diagrams should reflect actual deployed subscriptions/regions; redact sensitive IP ranges if required, but preserve routing/security intent.

14.5 Validation

You should validate audit readiness using repeatable checks:

1. **Forced tunneling enforcement**

    1. Export route tables from spokes and confirm `0.0.0.0/0` points to Azure Firewall (or approved egress).

   2. Capture “effective routes” for representative workload NICs to confirm next-hop matches intent.

2. **NSG deny-by-default posture**

   1. Confirm inbound and outbound rules have explicit allows and a final deny (or rely on implicit deny with explicit allow patterns).
   2. Validate that high-risk outbound is explicitly controlled (not “Allow Internet Any”).

3. **Private Endpoint and DNS**

   1. Query private endpoint NIC IPs and associated DNS zone records.
   2. Perform name resolution tests from spoke workloads to verify private IP responses (no public fallback).

4. **Operating effectiveness of inspection and monitoring**

   1. Query Azure Firewall logs for egress denies/allows over an audit window.
   2. Query NSG flow logs for sampled east-west denies consistent with segmentation design.
   3. Validate log retention meets audit requirement (e.g., 90/180/365 days) and is consistently applied.

14.6 Pitfalls

- Missing **Azure Private DNS Zone** links to all relevant **Azure Virtual Network (VNet)** instances, causing intermittent resolution to public endpoints.
- Incomplete egress story: forced tunneling configured, but firewall policy allows overly broad outbound, undermining intent.
- Evidence fragmentation: logs stored in multiple workspaces/subscriptions without consistent retention and access controls.
- Overreliance on control plane controls (Policy) without data plane operating effectiveness (flow/firewall logs).

  Anti-pattern: Declaring “no public exposure” while allowing ad-hoc public IP creation in spokes (even if “not used”). Auditors often test existence, not intent.

14.7 Audit evidence

You should assemble an “evidence pack” per audit period with:

- **Architecture and scope**
  - Current hub-spoke diagram set (versioned)
  - In-scope inventory: VNets, subnets, peerings, route tables, NSGs, Azure Firewall instances/policies, Private Endpoints, Private DNS zones
- **Configuration enforcement**
  - Azure Policy assignments and compliance exports for network guardrails
  - Exception register with approvals and expiry dates
- **Change traceability**
  - Activity Log exports for network changes (sampled) with ticket references
- **Operating effectiveness**
  - Azure Firewall log queries/exports demonstrating egress control and threat detections (where enabled)
  - NSG flow log samples showing segmentation behavior
  - Private Endpoint DNS validation results
- **Access control**
  - Role assignment exports for network admin roles and evidence of periodic review

14.8 Controls checklist (s14)

- [ ] You maintain a documented boundary statement covering ingress, egress, and east-west segmentation intent.
- [ ] You can export inventories for NSG/ASG, UDRs, Azure Firewall/Firewall Policy, Private Endpoints, and Private DNS zones for the audit window.
- [ ] You use Azure Policy compliance exports for network-relevant guardrails and track exceptions.
- [ ] You retain and can query Azure Firewall logs and NSG flow logs centrally for the required retention period.
- [ ] You can demonstrate forced tunneling via UDRs and effective routes for representative workloads.
- [ ] You can prove Private Endpoint name resolution to private IPs (no public DNS fallback).
- [ ] You provide role assignment and access review evidence for network control administrators.

- [ ] You have a repeatable evidence pack process (export steps, queries, storage, and versioning).