INDEPENDENT HANDS-ON PROJECT

Secure Multi-AZ VPC Foundation

Networking, private access, observability and cost-aware egress

Portfolio track	Independent Hands-on Engineering
Evidence level	Terraform fmt, init and validate passed
Author	Saurabh Dindokar

Evidence boundary: This is a public-safe personal implementation lab. It is not client production work, and no live deployment is claimed.

Overview

A reusable Terraform network foundation with public ingress subnets, private application subnets, isolated data subnets, controlled outbound routing, VPC Flow Logs, an S3 gateway endpoint and private Systems Manager endpoints.

Engineering Problem

Application platforms need predictable network boundaries before compute or databases are introduced. The lab addresses subnet separation, routing, administrative access, audit visibility and the cost trade-off between no NAT, a shared NAT Gateway and one NAT Gateway per Availability Zone.

Architecture

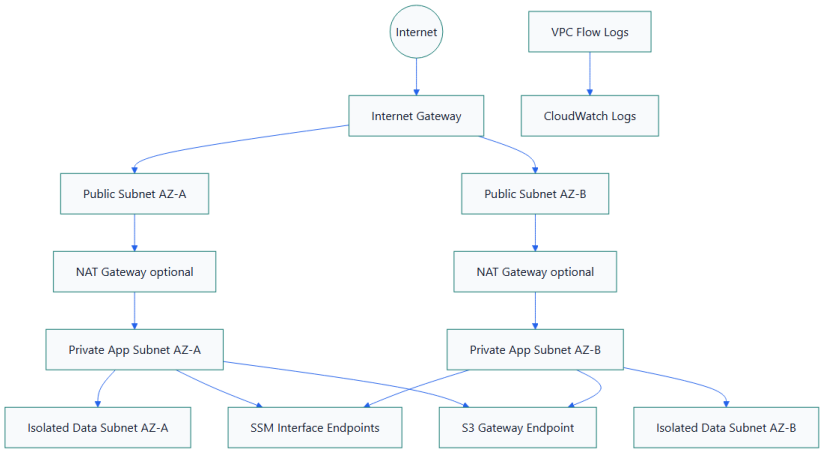

Figure 1. Secure Multi-AZ VPC Foundation architecture

What I Implemented

- Built a reusable network module with VPC, subnet, route-table, NAT and endpoint options.

- Separated public, private application and isolated data CIDR ranges across two Availability Zones.
- Kept public IP assignment disabled by default and avoided inbound SSH by designing for Systems Manager access.
- Added VPC Flow Logs with explicit CloudWatch retention and private S3 routing through a gateway endpoint.
- Created development and production examples with different availability and cost choices.

Important Technical Decisions

- NAT mode defaults to none so validation cannot create accidental hourly and data-processing charges.
- Production can use one NAT Gateway per Availability Zone to remove a single-AZ egress dependency.
- Data subnets receive no default internet route.
- Systems Manager endpoints support administration without opening SSH ingress.

Security Controls

- Private and isolated subnet boundaries reduce direct internet exposure.
- Endpoint security-group paths are explicit, and administration is designed around Systems Manager.
- Flow Logs provide network-flow evidence for troubleshooting and review.

Reliability and Operations

- Subnets span two Availability Zones.
- Production NAT can be deployed per Availability Zone.
- The module exposes outputs needed by later ALB, compute and database layers.

Cost and Cleanup Guardrails

- NAT Gateways and interface endpoints can incur hourly and data-processing charges.
- The development example keeps NAT disabled; the production example makes higher availability explicit.
- CloudWatch log retention is bounded instead of unlimited.

Validation Evidence

The following local checks passed on 2026-08-13:

```
terraform fmt -check -recursive
terraform init -backend=false
terraform validate
```

Scope boundary: Format, initialization and validation passed on 2026-08-13. No Terraform plan, AWS apply or live connectivity test is claimed.

Delivery and Verification Runbook

- 1. Select non-overlapping CIDRs and Availability Zones
- 2. Review NAT and endpoint cost
- 3. Validate Terraform

- 4. Review plan in a sandbox
- 5. Verify routes, DNS, Flow Logs and SSM
- 6. Destroy and verify chargeable resources

Technology Stack

Terraform | AWS VPC | Subnets | Route Tables | NAT Gateway | VPC Endpoints | Systems Manager | CloudWatch Logs

Key Learnings

- Infrastructure evidence must distinguish code validation, plan review and live deployment.
- Security, reliability, cost and cleanup decisions should be documented before apply.
- A useful platform lab includes verification and rollback thinking, not only resource declarations.