The Compliant Cloud Perimeter (AWS + Terraform)

Executive Summary

This project demonstrates the transition from manual, error-prone cloud configuration to **Infrastructure as Code (IaC)** with a Security-First mindset, the essence of DevSecOps. Instead of deploying a standard out-of-the-box environment, yours truly has engineered a hardened AWS infrastructure designed to satisfy Governance, Risk, and Compliance (GRC) requirements.

By utilizing **Terraform**, I have automated the enforcement of security controls, ensuring that the environment is auditable, repeatable, and resilient against common misconfigurations.

Technical Architecture & GRC Alignment

The project is built on four core pillars, each mapping directly to industry standard frameworks like NIST or ISO 27001.

1. Network Isolation (Custom VPC)

> **What:** A tailored Virtual Private Cloud featuring distinct public and private subnets across multiple Availability Zones.
>
> **Why:** Standard default VPCs are often too open. This architecture ensures that sensitive workloads are isolated from the public internet, reducing the attack surface.

2. Security Groups as Code (Micro-Segmentation)

> **What:** Strict ingress/egress rules defined within Terraform.
>
> **Why:** We operate on a **Zero Trust** principle. There are no `0.0.0.0/0` (open to the world) rules here; only specific ports and protocols are permitted for identified sources.

3. Identity & Access Management (Least Privilege)

> **What:** Automated creation of IAM Roles and Instance Profiles for EC2, replacing the dangerous practice of using hardcoded access keys.
>
> **Why:** This aligns with the **Principle of Least Privilege (PoLP)**. The infrastructure only has the permissions it absolutely needs to function, and nothing more.

4. Continuous Monitoring (Audit Trail)

> **What:** Integration of **AWS Config** and **CloudWatch Logging**.
>
> **Why:** In a SOC or GRC role, visibility is everything. This setup ensures that every change is logged and the environment is constantly monitored for compliance drift.

Why This Matters

I recognize that modern security is about finding bugs as well as building systems that are secure by default. This project proves my ability to:

- Translate high-level compliance requirements into technical reality.
- Manage cloud resources efficiently using industry-standard tools.
- Maintain a high level of technical rigor.