Integrated Management Information System (MIS)

Enterprise Hybrid Infrastructure & Disaster Recovery Design

Role: DevOps / Infrastructure Architect

Date: February 24, 2026

Classification: Confidential / Enterprise Government Design

Table of Contents

1. Executive Summary & Scenario Overview

2. High-Level Architecture Design

3. Detailed Network & Infrastructure Design

4. High Availability (HA) Design

5. Disaster Recovery (DR) Strategy

6. Security & Compliance

7. CI/CD & Deployment Strategy

8. Monitoring & Observability

9. Capacity Planning & Sizing

1.1 Project Objective

The goal is to design a government-scale Integrated Management Information System (MIS) that balances the security of On-Premise infrastructure with the scalability and resilience of the Cloud. The system must serve 1,000+ concurrent users across multiple departments while maintaining a strict 99.9% uptime SLA and full regulatory compliance.

1.2 Scenario Requirements

- **Concurrency:** 1,000+ concurrent sessions at peak.
- **Architecture:** Hybrid (On-Prem + AWS).
- **Integrations:** Web + External API-based sinks.
- **Compliance:** Strict data protection and RBAC.
- **Reliability:** Mandatory Disaster Recovery (DR) with low RTO/RPO targets.

1.3 High-Level Solution

The proposed architecture utilizes a **Dual-Region Hybrid Model**. On-Premise serves as the primary data sovereign zone, while AWS provides elastic application scaling and a warm-standby Disaster Recovery environment. Orchestration is standardized on Kubernetes (K8s On-Prem and EKS on Cloud) to ensure deployment parity and seamless failover.

2.1 On-Premise Architecture (The Private Anchor)

The On-Premise environment is designed for sovereignty. It consists of: * **DMZ Layer:** Hardened F5 Load Balancers and Cisco Firewalls. * **Application Tier:** A self-managed Kubernetes cluster running the React SPA and FastAPI backend. * **Data Tier:** HA PostgreSQL cluster using streaming replication and Redis Sentinel for distributed caching.

2.2 Cloud Architecture (AWS - The Elastic Layer)

The AWS footprint focuses on high availability: * **VPC Design:** Three-tier VPC (Public, Private App, Private Data). * **Compute:** Amazon EKS for container orchestration across multiple Availability Zones (Multi-AZ). * **Storage:** Managed RDS (PostgreSQL) and S3 for document storage with Cross-Region Replication (CRR).

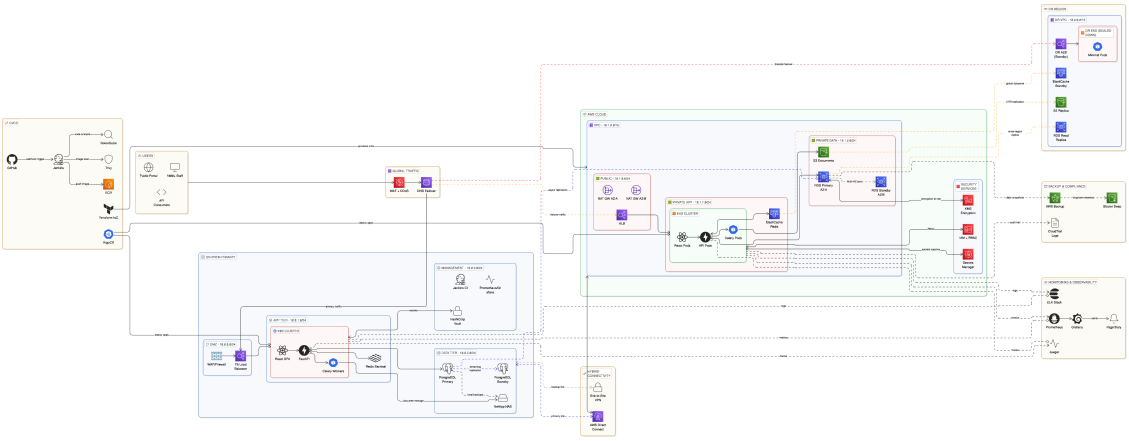
2.3 Hybrid Model (The Bridge)

Connectivity is established via **AWS Direct Connect** (Primary) and **Site-to-Site VPN** (Secondary/Backup). GSLB (Global Server Load Balancing) via Route 53 directs traffic based on proximity or health checks.

2.4 Disaster Recovery Architecture

The DR region (Secondary AWS Region) remains in a "Warm Standby" state. Database replicas are continuously updated via asynchronous replication, and application manifests are synced using GitOps (ArgoCD), allowing for rapid scaling during a failover event.

Figure 1: Enterprise Hybrid Architecture Overview



(Reference Architecture: Global Traffic -> Multi-Region Hub & Spoke)

3.1 CIDR Planning & Segmentation

To avoid IP overlap and ensure routing efficiency, the following CIDR scheme is implemented: * **On-Premise:** 10.0.0.0/16 * DMZ Subnet: 10.0.0.0/24 * App Subnet: 10.0.1.0/24 * Data Subnet: 10.0.2.0/24 * Management: 10.0.3.0/24 * **AWS Primary (Region A):** 10.1.0.0/16 * Public Subnet (ALB/NAT): 10.1.0.0/24 * Private App (EKS): 10.1.1.0/24 * Private Data (RDS): 10.1.2.0/24 * **AWS DR (Region B):** 10.2.0.0/16

3.2 Load Balancing & DMZ Strategy

- **External Tier:** AWS WAF + Route 53 Latency-based routing.
- **Internal Tier:**
  - **On-Prem:** F5 BIG-IP provides SSL termination and Layer 7 inspection.
  - **Cloud:** Application Load Balancer (ALB) integrated with AWS Certificate Manager (ACM).
- **DMZ Design:** All ingress traffic is forced through WAF/Firewall inspection. No direct internet access to Private App or Data tiers; outbound traffic is routed through HA NAT Gateways.

3.3 Container Orchestration (Kubernetes)

Both environments utilize K8s to abstract infrastructure. * **Worker Nodes:** Auto-scaling groups to handle bursty 1,000+ user loads. * **Ingress:** NGINX Ingress Controller for fine-grained path routing.

4.1 Achieving 99.9% Uptime

99.9% uptime permits approx. 8.77 hours of downtime per year. This is achieved through: * **Redundancy:** No single point of failure (N+1 at every layer). * **Multi-AZ Strategy:** Cloud components are spread across three Availability Zones.

4.2 Database & Cache HA

- PostgreSQL:
  - **On-Prem:** Primary with Synchronous Standby to ensure zero data loss.
  - **Cloud:** Multi-AZ RDS deployments with automatic failover.
- Redis HA:
  - **On-Prem:** Redis Sentinel managing master-slave promotion.
  - **Cloud:** Amazon ElastiCache (Redis OSS) in Cluster Mode with Multi-AZ.

4.3 Stateless App Scaling

Backend services (FastAPI) are designed to be stateless. Sessions are stored in Redis. This allows the Horizontal Pod Autoscaler (HPA) to scale replicas based on CPU/Memory utilization dynamically.

5.1 Recovery Objectives

- **RPO (Recovery Point Objective):** < 15 minutes (asynchronous data lag).
- **RTO (Recovery Time Objective):** < 1 hour (automated infrastructure spin-up).

5.2 Mode of Operation

Active-Passive (Warm Standby): The On-Prem/Primary Cloud serves 100% of traffic. The DR region maintains a scaled-down EKS cluster and a live DB Read-Replica.

5.3 Failover & Governance

1. **Detection:** Route 53 Health Checks identify Primary region failure.
2. **Promotion:** DR PostgreSQL Read-Replica is promoted to Primary.
3. **Traffic Shift:** DNS records update to point to the DR ALB.
4. **Scale-up:** HPA automatically scales the DR EKS worker nodes to handle production load.

5.4 DR Drills & Validation

- **Quarterly Drills:** Simulated regional failure to test automated Terraform/ArgoCD recovery.
- **Automation:** Use of AWS Resilience Hub to audit and simulate outages.

6.1 IAM & RBAC

- **Strategy:** Least Privilege.
- **Implementation:** Kubernetes RBAC integrated with LDAP (On-Prem) and AWS IAM (Cloud) using IAM Roles for Service Accounts (IRSA).

6.2 Data Security

- **At Rest:** AES-256 encryption via AWS KMS and On-Prem Hardware Security Modules (HSM).
- **In Transit:** TLS 1.3 enforced for all internal and external communication.
- **Secrets Management:** HashiCorp Vault (On-Prem) and AWS Secrets Manager (Cloud). Secrets are injected into containers as environment variables or sidecars.

6.3 Zero-Trust Principles

- All internal traffic is authenticated.
- Micro-segmentation enforced via K8s Network Policies (Cilium or Calico).

7.1 Pipeline Overview

A unified Jenkins pipeline handles the build process: 1. **CI Phase:** Code analysis (SonarQube) -> Security Scan (Trivy) -> Unit Tests. 2. **Artifact:** Docker images pushed to Amazon ECR (Cloud) and Private Nexus/Harbor (On-Prem). 3. **CD Phase:** ArgoCD (GitOps) monitors the Git repo for manifest changes and syncs the state to K8s/EKS clusters.

7.2 Deployment Strategy

Blue-Green Deployment: New versions are deployed to a "Green" stack. Traffic is shifted only after automated smoke tests pass.

7.3 Infrastructure as Code (IaC)

100% of the Cloud and Hybrid connectivity is managed via **Terraform**. This ensures the DR region is an exact replica of the Production environment.

8.1 Toolset

- **Metrics:** Prometheus + Grafana for real-time dashboarding.
- **Logging:** ELK Stack (Elasticsearch, Logstash, Kibana) for centralized log aggregation.
- **Tracing:** Jaeger / AWS X-Ray for distributed tracing across microservices.
- **Alerting:** PagerDuty integration for critical threshold breaches.

8.2 Observability Metrics

- **SLIs/SLOs:** Error rate < 0.1%, App Latency < 200ms (p99).
- **Error Budgets:** Managed within quarterly cycles to balance feature velocity with stability.

9.1 Sizing Assumptions (1,000 Concurrent Users)

- **Web/API Nodes:** 8-12 Pods (2 vCPU, 4GB RAM each).
- **Database:** 8 vCPU, 64GB RAM (SSD optimized).
- **Cache:** 16GB RAM Redis Cluster.

9.2 5-Year Growth Model

- **Data Growth:** Estimated 200GB/year for structured data + 1TB/year for document storage (S3/NAS).
- **Scaling Strategy:** Horizontal scaling for App Tier; Vertical scaling + Sharding for the Data Tier.

9.3 Potential Single Points of Failure (SPOF)

- **Action:** Direct Connect is redundant with Site-to-Site VPN. PostgreSQL is Multi-AZ/Standby. WAF is global via AWS Edge.