# **Technical Report**

## 1. Architectural Justification

The system is designed to deploy a **simple web application** along with a **Prometheus/Grafana observability stack** on AWS, using **Terraform, Ansible, and Kubernetes**.

**Key Components and Rationale:**

* **AWS EKS (Kubernetes)**  
  Managed Kubernetes ensures high availability, automatic upgrades, and easy scaling of pods. Supports multi-AZ deployments for fault tolerance and integrates with IAM Roles for Service Accounts (IRSA) to provide fine-grained access to AWS resources.
* **VPC with Public and Private Subnets**  
  Public subnets host ALB for external traffic; private subnets host EKS worker nodes and internal services. NAT Gateways allow private subnets to access the internet securely. Multi-AZ architecture improves resilience.
* **ECR for Container Images**  
  Centralized image repository ensures version control, image scanning, and easy CI/CD integration.
* **Prometheus + Grafana**  
  Prometheus collects metrics from all pods and nodes; Grafana provides visualization, offering end-to-end observability for monitoring performance, resource utilization, and potential issues.

**Trade-offs and Future Improvements:**

* Managed EKS simplifies operations but may increase costs compared to self-managed Kubernetes.
* Current setup uses basic instance types; production may benefit from mixed instance types for cost/performance optimization.
* Future improvements: integrate logging with centralized storage (e.g., S3 or OpenSearch), enable alerting via SNS, and implement autoscaling policies for workloads and cluster nodes.

## 2. Security Rationale

**Implemented Measures:**

* **AWS IAM**  
  Principle of least privilege applied; GitHub Actions deploys through a dedicated role. IRSA grants pods access to AWS resources without embedding credentials.
* **Secrets Management**  
  Kubernetes secrets stored securely, optionally encrypted with KMS. Environment-specific secrets injected dynamically via CI/CD pipeline.
* **Network Security**  
  Public/private subnet segregation prevents direct access to internal services. Security groups restrict traffic to required ports only. NAT Gateways avoid exposing internal nodes to the internet.
* **Container Security**  
  ECR image scanning enabled to detect vulnerabilities. Resource requests/limits prevent resource starvation and noisy neighbor issues.

**Future Security Enhancements:**

* Integrate **Pod Security Policies** or **OPA/Gatekeeper** to enforce security policies.
* Enable **AWS GuardDuty** and **CloudTrail** monitoring for auditing.
* Encrypt Kubernetes secrets at rest and in transit using KMS or HashiCorp Vault.
* Enable network segmentation using **Cilium** or **Calico** for fine-grained pod communication controls.

## 3. CI/CD Pipeline Design

**Design Overview:**

* Implemented via **GitHub Actions**, fully environment-agnostic using Terraform workspaces.
* Steps include:
  1. **Terraform Init/Plan/Apply**: Creates all required AWS resources including ECR, VPC, and EKS.
  2. **Docker Build & Push**: Builds environment-specific images and pushes them to ECR.
  3. **Kubernetes Deployment**: Dynamically replaces image placeholders in manifests and applies resources.
  4. **Smoke-Test**: Validates deployment functionality end-to-end.

**Rationale and Benefits:**

* **Repeatability**: Terraform and Kubernetes manifests ensure consistent deployments.
* **Scalability**: Modular Terraform and Kubernetes designs allow adding more environments, nodes, or services with minimal changes.
* **Reliability**: Automated smoke-test ensures deployments are verified before success.
* **Environment Agnostic**: CI/CD uses Terraform workspaces and dynamic image URIs, allowing the same codebase to deploy dev/staging/prod.

**Future Optimizations:**

* Integrate **parallel deployment stages** for faster delivery.
* Use **Helm** for more flexible Kubernetes templating and versioning.
* Add automated **rollback and canary deployment strategies**.
* Monitor CI/CD pipeline itself for failures and performance metrics.