TASK

Objective

This task is designed to assess your ability to design and implement a secure, observable, and maintainable DevOps pipeline around a containerized application, with real-world cloud-native tooling.

Context

You are given a simple web service (a REST API or web app - you may use any minimal one of your choice, e.g., Python Flask, Node.js Express, or Go). You will build a CI/CD workflow around it, deploy it to Kubernetes, and integrate observability and security controls.

Requirements:

1. **Containerization**

    - Create a secure, minimal Dockerfile using multistage builds.
    - The image must not run as root.
    - No hardcoded secrets in code or image.

2. **CI/CD Pipeline**

    - Set up a pipeline using a tool of your choice (e.g., GitHub Actions, GitLab CI, Jenkins).
    - On push to the main branch:
        * Build the Docker image
        * Run a vulnerability scan (e.g., Trivy)
        * Push the image to a registry
        * Deploy the app to a Kubernetes cluster (KinD or Minikube is acceptable)

3. **Kubernetes Deployment**

    - Use Helm or Kustomize (your choice) to define manifests.
    - Include:
        * CPU and memory resource requests/limits
        * Readiness and liveness probes

4. **Secrets Management**

   - Store secrets securely (e.g., Kubernetes Secrets, SOPS, SealedSecrets, or external like Vault).

5. **Observability**

   - Add basic monitoring and logging:

     * Cloud Watch or Prometheus metrics

Bonus Points (Not Required, But Appreciated)

- Use Terraform to provision the cluster or registry

- Implement alerting (Slack/email/Webhook)

- Use ArgoCD or Flux for GitOps-style deployment

Submission Guidelines

- Push your code to a public or private Git repository (GitHub, GitLab, Bitbucket).

- Include a README.md that explains:

  * The architecture and decisions

  * How to run the project end-to-end

  * Known limitations and areas for improvement