DevOps/SRE Engineer Practical Assessment

Objective

Your task is to set up a Kubernetes cluster, deploy an application with CI/CD, implement monitoring, and configure logging. This assessment evaluates your skills in infrastructure automation, deployment, and observability.

Time Allocation

- **Total Time:** 8 hours
- **Flexibility:** You may adjust time distribution as needed.

Tasks

1. Set Up a 3-Node Kubernetes Cluster Using Kubespray & Cilium

- Use **Kubespray** to deploy the Kubernetes cluster.
- Configure **Cilium** as the CNI.

2. Develop & Containerize a Simple Application

- Write a web application (any language you want) that:
  - Responds with a REST endpoint (e.g., `/health` returning `{"status": "ok"}`).
  - Exposes **Prometheus metrics** (e.g. GET count).
- Containerize the app with Docker and push it to a registry (Docker Hub, GitLab, etc).

4. Set Up CI/CD Pipeline

- Choose a CI/CD tool (GitHub Actions, GitLab CI, Jenkins, ArgoCD).
- Automate:
  - **Build & Push** (on code changes).
  - **Deploy** (to Kubernetes on `main` branch updates).
- Ensure pipeline logs are visible.

5. Monitoring & Logging

Monitoring (Mandatory)

- Deploy Prometheus or Victoria Metrics (Helm).
- Verify metrics are being scraped.

Logging (Mandatory)

- Deploy **EFK (Elasticsearch, Fluentd, Kibana)** or **ELK (Elasticsearch, Logstash, Kibana)** stack - single node is okay.
- Ensure you app logs are stored in Elasticsearch and queryable in Kibana.

Deliverables

1. **Code & Configurations** (Git repository with):
   - Application source code.
   - Dockerfile & Kubernetes manifests.
   - Kubespray inventory/config files.
   - CI/CD pipeline files.
   - Prometheus/EFK/ELK configurations.
2. **Documentation (README.md)** covering:
   - Steps to reproduce the setup.
   - How to access the deployed app, metrics, and logs.
   - Any challenges faced.

Evaluation Criteria

✅ **Cluster Setup** – Kubespray + Cilium working, nodes healthy.
✅ **App Deployment** – Running, accessible, metrics exposed.
✅ **CI/CD Pipeline** – Automated, functional.
✅ **Monitoring** – Prometheus scraping metrics (Grafana bonus).
✅ **Logging** – EFK/ELK collecting and storing logs.
✅ **Clean Code & Docs** – Reproducible, well-documented.

Good luck! Submit your work via Git repository, and we'll review it for further discussion. 🚀