### **Take-Home Assignment: Software Engineer II - InfraOps**

Thank you for your interest in this role! This assignment assesses your ability to automate, containerize, and deploy applications efficiently using modern DevOps practices.

### **Objective**

Set up and deploy a simple application (Frontend **or** Backend) with a focus on containerization, CI/CD automation, and deployment using **Minikube**.

### **Task Details**

1. **Application Setup** (Minimal Development):
   * Choose between:
     + **Backend**: A REST API using **Spring Boot (Java)** with a single endpoint (e.g., GET /health or GET /books).
     + **Frontend**: A basic app using **Angular** or **React** displaying static data (e.g., a list of books/movies).
     + **Database**: A DB of your choice. Extra points for Mongo.
   * **Note**: Use a simple example—development is secondary to deployment and automation.
2. **Containerization**:
   * Write a **Dockerfile** to package the application into a container.
   * Ensure the container runs successfully.
3. **CI/CD Automation**:
   * Create a **GitHub Actions workflow** to:
     + Build the Docker image.
     + Push the image to a registry (e.g., Docker Hub or GitHub Container Registry).
4. **Kubernetes Deployment (Minikube)**:
   * Deploy the application to **Minikube** using Kubernetes manifests:
     + **Deployment**: Deploy your containerized application.
     + **Service**: Expose the application locally (e.g., NodePort or ClusterIP).
   * Verify the application is accessible on localhost.
5. **Documentation**:
   * Provide a **README.md** with:
     + Steps to build and run the application locally using Docker.
     + Steps to deploy the application on Minikube, including:
       - Required Minikube setup commands.
       - How to access the application.
     + Explanation of the CI/CD pipeline in GitHub Actions.
     + Any decisions, assumptions, or challenges faced.

### **Deliverables**

1. **Source Code** (minimal focus).
2. **Dockerfile**: For containerization.
3. **GitHub Actions workflow**: For automation.
4. **Kubernetes Manifests**: Deployment and Service YAML files.
5. **README.md**: Comprehensive setup instructions, CI/CD details, and decision justifications.
6. **GitHub Repository**: Public or accessible repository link with a clean commit history.
7. **Screenshots**: Screenshots or outputs verifying your solution.

### **Evaluation Criteria**

* **Automation**: Effective use of GitHub Actions for CI/CD.
* **Containerization**: Proper Dockerfile setup.
* **Kubernetes Deployment**: Working manifests and Minikube deployment.
* **Documentation**: Clear, step-by-step instructions.
* **Code Quality**: Minimal but functional code to demonstrate deployment.

### **Timeline**

* Submission deadline: **2 days** from receiving this assignment.

Please submit your **GitHub repository link** with all deliverables. If you have questions during the task, feel free to reach out.

We value originality and problem-solving over perfection. Please ensure you explain and justify every decision in your solution. We shall follow up with this during the final tech interview to discuss the same.

Good luck, and we look forward to reviewing your work! 🚀