# DevOps Engineering Challenge

Create a minimalistic local Kubernetes environment for running a **NodeJS** application that connects to a database, retrieves the "Hello World" string from it and returns it as a reply to an HTTP request. There should be two infrastructure environments, Dev and Prod.

### Requirements

All resources should be managed in your code (i.e. **GitOps** approach).

The application itself is expected to be deployed directly **without** using pre-existing images (like the images publicly available from DockerHub).

Publicly-available images are allowed to be used for the underlying “infrastructure” pieces, like DB or the CI tool.

Dev and Prod environments must be configured using a template tool (e.g. HELM, Kustomize, jsonnet, etc.)

Keep the solution simple, but make sure to apply best practices where you see them fit.

### Tools

The following tools are expected to be used for completing this challenge:

- Container management: **Kubernetes.**
- Cluster management: **Minikube.**
- Deployable artifact: Docker image created from the provided **Dockerfile**.
- Deployment method: Kubernetes **YAML** files **/ HELM.**
- Infrastructure environment templating: **HELM / Kustomize / jsonnet.**
- CI/CD: **GitlabCI / Jenkins** pipeline (can be run as a separate pod on the Minikube "cluster").
- Scripting language: **Bash, Python, Ruby or Go.**

### Deliverables

A link to a publicly available git repo with:

- **Dockerfile** (one or several, depending on your design) containing the build of a deployable artifact, like a Docker image with an application inside.
- Kubernetes **manifests / *HELM*** charts to deploy this app to a Kubernetes cluster.
- ***GitlabCI/Jenkins*** pipeline which streamlines the process of building and deploying (nice to have).
- Create **README.md** with step-by-step instructions on how to get from cloning your public repo to opening your hello-world application running in Minikube in a web browser and seeing the "Hello World" phrase (no need for a fancy web page, a line of text is sufficient).

What we will look for when checking out your solution:

- Functionality and working state of the setup and the application you are deploying.
- Cleanliness and simplicity of the automated infrastructure.
- Clearness and structure of the documentation.