DevOps Pipeline

1. **GitHub-hosted code**
2. **GitHub Actions** to build Docker images from that code (frontend + backend)
3. **Push those images to GitLab Container Registry**
4. **SSH into EC2 (via GitHub Actions)** and:
   * Pull those GitLab-hosted Docker images
   * Use docker-compose to run the full stack (MySQL + backend + frontend)

**STEP 1: GitHub Code (Source of Truth)**

Your GitHub repo contains both:

/frontend ← React code

/backend ← Node.js + Express

/docker-compose.yml

**STEP 2: GitHub Actions Builds Docker Images (CI)**

In GitHub Actions workflow:

1. GitHub pulls the source code (from itself)
2. It **builds Docker images**:
   * frontend image from ./frontend/Dockerfile
   * backend image from ./backend/Dockerfile
3. Then it **pushes those images to GitLab Container Registry**:
   * registry.gitlab.com/<you>/incident-tracker-frontend
   * registry.gitlab.com/<you>/incident-tracker-backend

\*\* Use GitLab personal access token to push to GitLab.

**STEP 3: GitHub Actions Deploys to EC2 (CD)**

In the same GitHub Actions pipeline:

1. GitHub SSHes into the EC2 instance.
2. EC2 navigates to the GitHub project folder (which includes the docker-compose.yml file).

Checks if the IncidentTracker folder exists on the EC2 machine.

* If **yes**: it enters the folder and pulls the latest code from GitHub.
* If **no**: it clones the repo from GitHub and enters the new folder.
  + **cd IncidentTracker** happens, bringing you into the folder containing docker-compose.yml.

1. Logs in to GitLab Container Registry.
2. Pulls the latest Docker images as defined in the docker-compose.yml file:
   * frontend image from GitLab
   * backend image from GitLab
3. Deploys the containers by running:

**docker-compose pull**

(This command reads the docker-compose.yml inside IncidentTracker/ (the cloned repo) and pulls the specified image by the image tags in the compose file)

**docker-compose up -d** This spins up frontend + backend + MySQL containers together.

**In Summary:\**

GitHub repo (source code)

↓

GitHub Actions CI:

- Build Docker image (frontend + backend)

- Push to GitLab Container Registry

↓

GitHub Actions CD:

- SSH into EC2

- Pull Docker images from GitLab

- Run `docker compose up` to deploy app

**++Additional,**

Import a GitHub repo into GitLab

Enable GitLab Container Registry (if not already)

* Go to imported GitLab project:  
  **Settings → Packages & Registries → Container Registry** → Make sure it’s enabled.

Add .gitlab-ci.yml to GitLab Repo in the root. This builds the Docker image from your code and pushes it to GitLab Container Registry.  
----------------------------------------------

**How to install docker compose on your EC2,**  
# Install Docker

sudo yum update -y

sudo yum install -y docker

sudo systemctl start docker

sudo systemctl enable docker

sudo systemctl status docker

sudo usermod -aG docker ec2-user

# Docker Compose installation commands

sudo curl -L "https://github.com/docker/compose/releases/download/v2.33.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo mv /usr/local/bin/docker-compose /usr/bin/docker-compose

sudo chmod +x /usr/bin/docker-compose