**I. Create a Dockerfile**

* **Action**: Define the environment and application dependencies in a Dockerfile.
* **Command**: Ensure the Dockerfile is properly structured with all necessary instructions (e.g., FROM, COPY, RUN, CMD).

**II. Build the Docker Image**

* **Action**: Build the Docker image locally.
* **Command**:

docker build -t <image-name>:<tag> <path-to-dockerfile>

* **Example**:

docker build -t my-app:latest .

**III. Create a Repository in ECR**

* **Action**: Create an AWS Elastic Container Registry (ECR) repository to store the Docker image.
* **Command**:

aws ecr create-repository --repository-name <your-repository-name> --region <your-region>

* **Example**:

aws ecr create-repository --repository-name my-app --region eu-west-1

**IV. Push the Image into the Repository**

* **Action**: Push the locally built Docker image to the ECR repository.
* **Steps**:
  1. **Authenticate Docker with ECR**:

aws ecr get-login-password --region <your-region> | docker login --username AWS --password-stdin <account-id>.dkr.ecr.<region>.amazonaws.com

* 1. **Tag the image for ECR**:

docker tag <image-name>:<tag> <account-id>.dkr.ecr.<region>.amazonaws.com/<your-repository-name>:<tag>

* 1. **Push the image**:

docker push <account-id>.dkr.ecr.<region>.amazonaws.com/<your-repository-name>:<tag>

Build terraform s3 backend

 **then, deploy your Terraform infrastructure**:

* This creates your AWS resources (VPC, EKS cluster, RDS database)
* Sets up the basic CloudWatch log groups and metrics
* Establishes SNS topics for alerts

 **Then, deploy your Kubernetes application manifests in this order**:

configure kubectl to use the EKS cluster.

aws eks update-kubeconfig --name topsurvey-dev-eks-cluster --region eu-west-1

Update the ecr uri in the deployment.yaml manifest file

kubectl apply -f kubernetes/manifest-files/

kubectl apply -f namespace.yaml

kubectl apply -f secrets.yaml

kubectl apply -f deployment.yaml

kubectl apply -f service.yaml

kubectl apply -f ingress.yaml

 **then, deploy the CloudWatch monitoring components**:

bash

Copy

# Navigate to your monitoring directory

cd monitoring/

# Make the script executable

chmod +x install-container-insights.sh

# Run the installation script

./install-container-insights.sh

**Then, Build and deploy your frontend application**

Create a route53 record

Update ingress file with the alb certificate

Manifests files: backend > deployment = change image with the ecr uri

Explicar que uso ssm