

# Project Instructions: Heart Disease Prediction MLOps

---

This document provides the full technical guide for setting up and running the MLOps pipeline.

---

## 1. Local Environment Setup

Follow these steps to initialize your environment using Conda.

### Setup Conda Environment

```
# Create the conda environment with Python 3.9
conda create -n mlops_env python=3.9 -y
```

```
# Activate the environment
conda activate mlops_env
```

```
#Install Dependencies
# Install the required packages via pip within the conda environment
pip install -r requirements.txt
```

---

## 2. Training & Experiment Tracking (MLflow)

The training process is instrumented with MLflow to track hyperparameters and model performance metrics.

**Execute Training:** Run the pipeline to process data and train models (Logistic Regression and Random Forest):

```
python scripts/train.py
```

**Review Experiments** Launch the MLflow UI to compare runs:

```
mlflow ui
```

- Visit <http://localhost:5000> to see the comparison table and accuracy scores.
- 

## 3. CI/CD Pipeline (GitHub Actions)

The project uses GitHub Actions to automate code quality and testing.

## Workflow Logic

- **Linting:** Every push triggers **Ruff** to ensure PEP8 compliance.
- **Testing: Pytest** runs suite-level tests on API endpoints.
- **Security:** The pipeline only builds the Docker image if all tests pass.

To view the pipeline status, navigate to the **Actions** tab in your GitHub repository.

---

## 4. Kubernetes Deployment & Monitoring

The API is deployed as a high-availability service on Kubernetes (Docker Desktop).

### Local Deployment

Apply the manifests to your local cluster:

```
# Deploy Pods and Service
kubectl apply -f kubernetes/deployment.yaml
kubectl apply -f kubernetes/service.yaml
```

### Verification Commands

Check the status of your deployment:

```
# Ensure 2/2 pods are 'Running'
kubectl get pods

# Find the External IP/Port for the API
kubectl get svc heart-disease-service
```

### Monitoring (Prometheus Metrics)

The API exposes real-time operational metrics via the Prometheus instrumentator. You can view these by navigating to:

- **Metrics Endpoint:** <http://localhost/metrics>
  - **Health Check:** <http://localhost/health>
- 

## 5. API Usage Example

You can test the running Kubernetes API using **curl** to ensure it is serving predictions:

```
curl -X 'POST' \
  'http://localhost/predict' \
  -H 'Content-Type: application/json' \
  -d '{
    "age": 52,
    "sex": 1,
    "cp": 0,
    "trestbps": 125,
    "chol": 212,
    "fbs": 0,
    "restecg": 1,
    "thalach": 168,
    "exang": 0,
    "oldpeak": 1.0,
    "slope": 2,
    "ca": 2,
    "thal": 3
  }'
```