**Assignment-1 [Total Marks - 25]**

**M1: MLOps Foundations**

**Objective**: Understand the basics of MLOps and implement a simple CI/CD pipeline.

**Tasks**:

1. **Set Up a CI/CD Pipeline**:

• Use a CI/CD tool like GitHub Actions or GitLab CI to set up a pipeline for a sample machine learning project.

• Include stages for linting, testing, and deploying a simple machine learning model.

2. **Version Control**:

• Implement version control for your project using Git.

Here are the contributors for the git repository:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

• Demonstrate branching, merging, and pull requests.

Here you can see that there are 3 branches:

1. feature/v0.1
2. main
3. origin/feature/v0.1.1

Team members are contributing to the branches as per their scope of work

A screenshot of a computer

Description automatically generated

For merging and pull request <Gaurav> to raise PR and ask someone to approve.

**Deliverables**:

• A report detailing the CI/CD pipeline stages.

• Screenshots or logs showing successful runs of the pipeline.

• A Git repository link with branches and merge history.

**M2: Process and Tooling**

**Objective**: Gain hands-on experience with popular MLOps tools and understand the processes they support.

**Tasks**:

1. **Experiment Tracking**:

• Use MLflow to track experiments for a machine learning project.

• Record metrics, parameters, and results of at least three different model training runs.

2. **Data Versioning**:

• Use DVC (Data Version Control) to version control a dataset used in your project.

• Show how to revert to a previous version of the dataset.

**Deliverables**:

• MLflow experiment logs with different runs and their results.

• A DVC repository showing different versions of the dataset.

**M3: Model Experimentation and Packaging**

**Objective**: Train a machine learning model, perform hyperparameter tuning, and package the model for deployment.

**Tasks**:

1. **Hyperparameter Tuning**:

• Use a library like Optuna or Scikit-learn’s GridSearchCV to perform hyperparameter tuning on a chosen model.

• Document the tuning process and the best parameters found.

2. **Model Packaging**:

• Package the best-performing model using tools like Docker and Flask.

• Create a Dockerfile and a simple Flask application to serve the model.

These are the screen shot which shows docker build. In the screen shot below docker build -t shuklagauravjn/ml-ops-group44-v-01:latest .

Leads to building of docker container.

A screenshot of a computer program

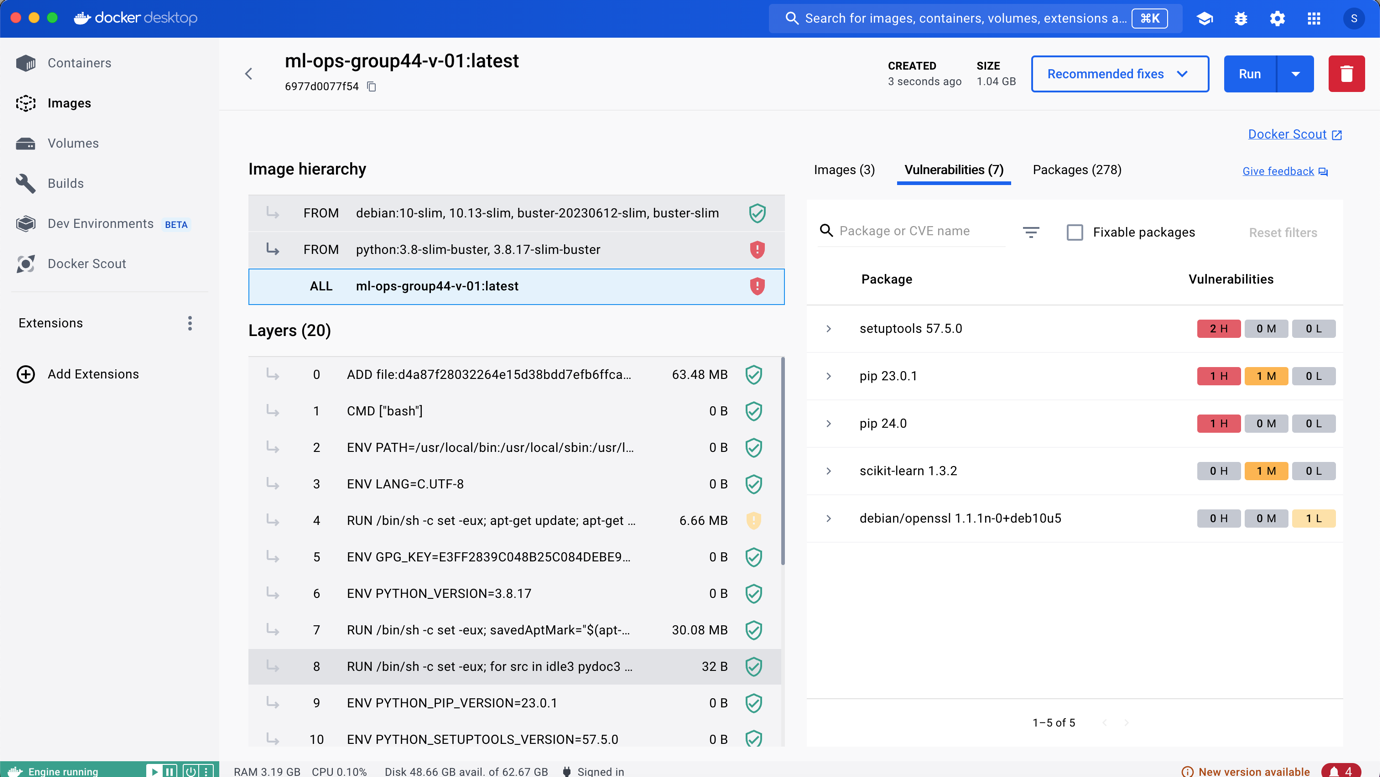
Description automatically generated

Once the docker container is built, it’s visible in docker desktop on mac.

A screenshot of a computer

Description automatically generated

The screen shot below shows the docker image and possible vulnerabilities.



Once the docker container was run from docker desktop, here is the log:A screenshot of a computer

Description automatically generated

Later this docker image was pushed to docker hub using the following command:

docker login -u "shuklagauravjn" -p "XSDSDwewe#" docker.io

docker push shuklagauravjn/ml-ops-group44-v-01:latest

A screenshot of a computer program

Description automatically generated

So it can be seen in hub.docker.com

A screenshot of a computer

Description automatically generated

**Deliverables**:

• A report on hyperparameter tuning results.

• A Dockerfile and Flask application code.

• Screenshots of the model running in a Docker container.

**M4: Model Deployment & Orchestration (Optional)**

**Objective**: Deploy a machine learning model and orchestrate its operations using Kubernetes.

**Tasks**:

1. **Model Deployment**:

• Deploy the Dockerized model from M3 to a cloud platform like AWS, Azure, or GCP.

• Use a platform service like AWS ECS, Azure AKS, or Google Kubernetes Engine (GKE).

2. **Orchestration**:

• Set up a Kubernetes cluster.

For local mac following things were done:

brew install minikube

minikube start

kubectl create deployment ml-ops-group44-v-01 --image=shuklagauravjn/ml-ops-group44-v-01:latest

kubectl expose deployment ml-ops-group44-v-01 --type=NodePort --port=8080

kubectl get services ml-ops-group44-v-01

minikube service ml-ops-group44-v-01

A screenshot of a computer

Description automatically generated

• Deploy the model using Kubernetes and create a Helm chart for managing deployments.

**Deliverables**:

• A link to the deployed model endpoint.

• Kubernetes configuration files and Helm chart.

• A report detailing the deployment and orchestration process.

**M5: Final Deliverables**

• A zip file containing:

• Code

• Data

• Model

• A one-page summary that includes:

• Description of the work completed

• Justification for the choices made

• A screen recording (maximum 5 minutes) that:

* Explains the work done
* Shows the results