MLOps Deployment from DEV to PROD

Objective

The objective of this lab was to simulate the process of deploying code from a **development environment (DEV)** to a **production environment (PRD)** in an MLOps pipeline. In this lab, I performed the roles of both the **Developer** and the **Gatekeeper**, following the steps to commit code, push it to GitHub, and review and run it as the Gatekeeper.

Steps Followed

---

1. Developer Role: Setting Up the Project

Project Setup and Dependencies

- I started by **creating a new Python project** and setting up a **virtual environment (venv)** to isolate project dependencies. This ensured that my project could run in different environments without conflicts.
    - I used pip to install necessary packages such as numpy, pandas, and scikit-learn, and created a requirements.txt file to document all dependencies.

Git Repository Setup

- I cloned the remote GitHub repository to my local machine using the following command:

  ```
  git clone https://github.com/RaghadAlmutairi/lab-mlops-deploy-prod-to-dev.git
  ```

- I initialized the Git repository in the local project folder:

  ```
  git init
  ```

Commit and Push Code

- I added all files (excluding the venv directory) to Git and committed the changes:

  ```
  git add .
  ```

  ```
  git commit -m "Set up virtual environment and add requirements.txt"
  ```

- After committing, I pushed the changes to the remote GitHub repository:

  ```
  git push origin main
  ```

- This completed the task of **pushing the code to the remote repository** as a Developer.

Creating a Pull Request (PR)

- Once my code was pushed, I created a **Pull Request (PR)** in GitHub to request the Gatekeeper to review and merge my changes. I followed the instructions in the lab to ensure that my PR was clear and properly formatted.

---

2. Gatekeeper Role: Reviewing and Running the Code

Pulling the Latest Code

- As the Gatekeeper, I first **reviewed the Pull Request** created by the Developer. Once the code was reviewed and verified, I **merged the PR**.
- I pulled the latest changes to my local machine using the command:

  ```
  git pull origin main
  ```

Setting Up the Environment

- I created a new **virtual environment (venv)** on my local machine:

  ```
  python -m venv venv
  ```

- Then, I activated the virtual environment and installed the required dependencies using:

  ```
  pip install -r requirements.txt
  ```

Running the Project

- After setting up the environment, I ran the code to ensure that everything worked correctly and there were no errors.
- Since the project ran successfully, I was able to provide feedback to the Developer confirming that everything was set up properly.

---

3. Swapping Roles

After completing the tasks as both the **Developer** and **Gatekeeper**, I swapped roles with my partner to gain hands-on experience with both sides of the process. This allowed me to see both the development and deployment aspects of the pipeline.

Challenges Faced and How I Solved Them

1. Git Configuration Issues

- **Challenge:** When I first tried to commit my changes, Git prompted me for my email and username configuration, which I had not set up yet.
- **Solution:** I ran the following commands to configure my Git username and email:

  ```
  git config --global user.email "you@example.com"
  git config --global user.name "RaghadAlmutairi"
  ```

After setting this up, I was able to commit my changes successfully.

2. Authentication Issues When Pushing Code

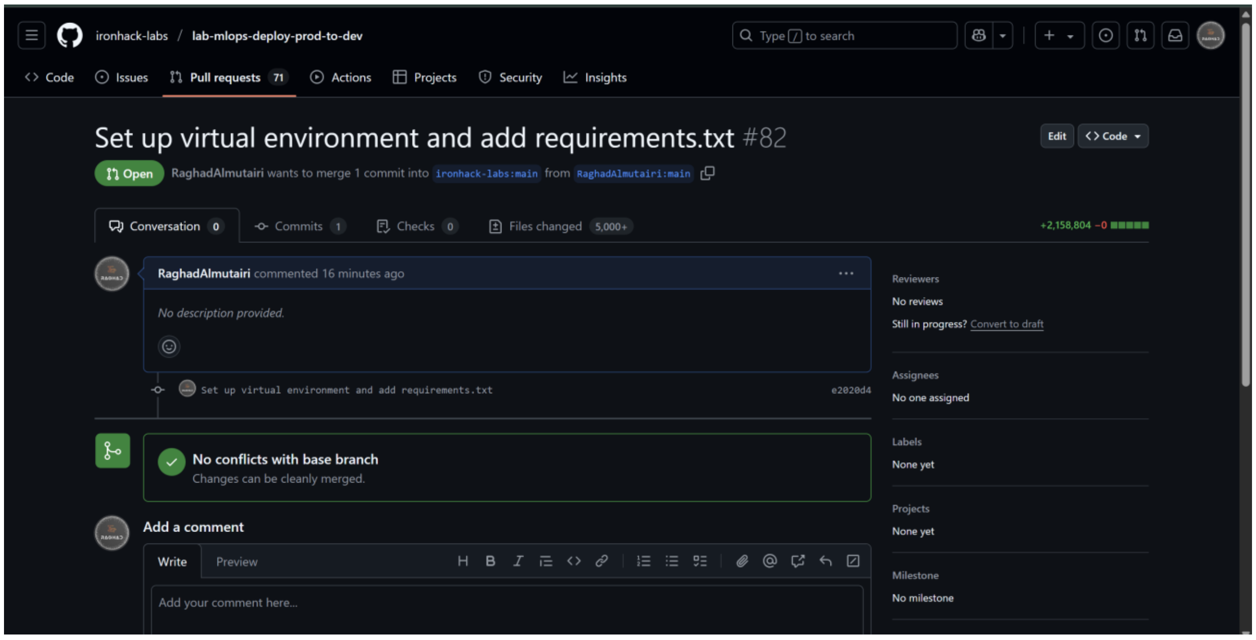
- **Challenge:** While pushing my changes to GitHub, I encountered an authentication error.
- **Solution:** I used **VS Code's GitHub integration** to authenticate my GitHub account and push the code without issues. This ensured that my GitHub credentials were properly stored and used.

Conclusion:

In this lab, I successfully completed the requirements as both the **Developer** and **Gatekeeper**, following the outlined steps:

- **As the Developer**, I set up the virtual environment, installed dependencies, committed my changes, pushed the code to GitHub, and created a pull request.
- **As the Gatekeeper**, I reviewed the pull request, set up the environment, installed dependencies, and tested the project.

This exercise allowed me to gain hands-on experience in the **MLOps** deployment process, understanding how to manage version control, set up isolated environments, and collaborate effectively in a team. The experience deepened my understanding of **CI/CD** practices and the importance of good communication between developers and gatekeepers.



Set up virtual environment and add requirements.txt #82

Open RaghadAlmutairi wants to merge 1 commit into ironhack-labs:main from RaghadAlmutairi:main

Conversation 0 · Commits 1 · Checks 0 · Files changed 5,000+ · +2,158,804 −0

RaghadAlmutairi commented 16 minutes ago

No description provided.

Set up virtual environment and add requirements.txt e2020d4

No conflicts with base branch
Changes can be cleanly merged.

Add a comment

Write · Preview

Add your comment here...

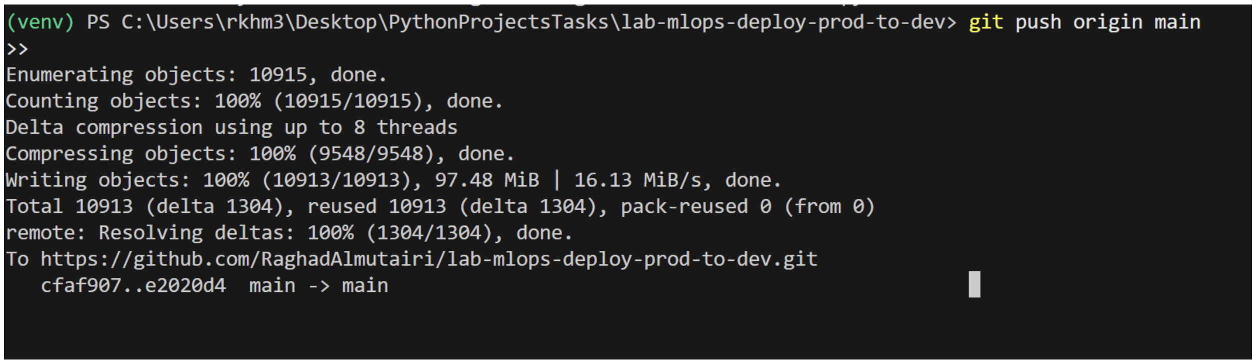
Reviewers: No reviews
Still in progress? Convert to draft

Assignees: No one assigned

Labels: None yet

Projects: None yet

Milestone: No milestone



```
(venv) PS C:\Users\rkhm3\Desktop\PythonProjectsTasks\lab-mlops-deploy-prod-to-dev> git push origin main
>>
Enumerating objects: 10915, done.
Counting objects: 100% (10915/10915), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9548/9548), done.
Writing objects: 100% (10913/10913), 97.48 MiB | 16.13 MiB/s, done.
Total 10913 (delta 1304), reused 10913 (delta 1304), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1304/1304), done.
To https://github.com/RaghadAlmutairi/lab-mlops-deploy-prod-to-dev.git
   cfaf907..e2020d4  main -> main
```