

lab-mlops-deploy-prod-to-dev

Gatekeeper & Developer Workflow

Process Overview:

In this exercise, we followed a real-world software development workflow:

- As the Developer, I created a Python project, set up a virtual environment, installed the necessary packages, and saved them in a requirements.txt file. I then initialized a Git repository, committed the code, pushed it to GitHub, and created a Pull Request (PR).
- As the Gatekeeper, I reviewed the PR submitted by my partner. After confirming that the code was clean and functional, I merged it, pulled the changes using `git pull origin main`, created a new virtual environment using Python 3.12.9, installed the dependencies, and ran the project to ensure everything worked correctly.
- We then switched roles and repeated the process to make sure each participant experienced both sides of the collaboration.

Challenges Faced & Solutions:

- **Multiple Python Versions:** There were multiple Python versions installed on the system, which caused some confusion. I solved this by checking the version using `python --version` and explicitly using the path to Python 3.12.9.
- **Wrong Python Version Used in venv:** At first, the virtual environment was created using the wrong Python version. I corrected this by specifying the exact version when running the `venv` command.
- **TensorFlow Installation Error:** While installing TensorFlow, I encountered a Windows error related to long file paths. I enabled long path support through system settings to resolve it.
- **No Output on Script Run:** When running `python your_script.py`, there was no output. I checked the script and realized there were no print statements.

Final Outcome:

The workflow was completed successfully on both sides. The project ran without issues, and the experience provided a solid understanding of collaborative development, code review practices, version control with Git, and environment setup.

lab-mlops-deploy-prod-to-dev

```
C:\Users\zhang > conda env create --name=python3.11.4
Collecting environment information...
[+] Microsoft Corporation. All rights reserved.

C:\Users\zhang> python --version
Python 3.11.4

C:\Users\zhang> C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV
[+] Microsoft Corporation. All rights reserved.

C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV> git pull origin main
From https://github.com/microsoft/LAB-RLPS-Deployment-from-PROD-to-Dev
   branch      main       -> FETCH_HEAD

C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV> pip install --no-warn-deps python3.12 --version
python3.12 is not recognized as an internal or external command,
operable program or batch file.

C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV> python --help
Python 3.12.9

C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV> python --help
Python 3.12.9

C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV> python Scripts\activate
env(C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV)python --version
Python 3.12.9

env(C:\Users\zhang\Downloads\LAB-RLPS-Deployment-from-PROD-to-DEV)pip install -r requirements.txt
Collecting absl-py==2.2 (from -r requirements.txt [line 1])
  Using cached absl_py-2.2.0-py3-none-any.whl.metadata (2.6 kB)
Collecting astunparse==6.3 (from -r requirements.txt [line 2])
  Using cached astunparse-6.3-py3-none-any.whl.metadata (4.4 kB)
Collecting certifi==2023.11.8 (from -r requirements.txt [line 3])
  Using cached certifi-2023.11.8-py3-none-any.whl.metadata (2.1 kB)
Collecting charset-normalizer==3.4.1 (from -r requirements.txt [line 4])
  Using cached charset_normalizer-3.4.1-py3-none-any.whl.metadata (3.6 kB)
Collecting contourpy==3.1 (from -r requirements.txt [line 5])
  Using cached contourpy-3.1-py38-gpl-win_amd64.whl.metadata (5.4 kB)
Collecting cycler==0.12.1 (from -r requirements.txt [line 6])
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (2.8 kB)
Collecting fonttools==4.52.1 (from -r requirements.txt [line 7])
  Using cached fonttools-4.52.1-py3-none-any.whl.metadata (876 bytes)
Collecting matplotlib==3.8 (from -r requirements.txt [line 8])
  Using cached matplotlib-3.8-py38-gpl-win_amd64.whl.metadata (184 kB)
Collecting numpy==2.0.0 (from -r requirements.txt [line 9])
  Using cached numpy-2.0.0-py38-gpl-win_amd64.whl.metadata (1.3 kB)
Collecting pandas==2.2.0 (from -r requirements.txt [line 10])
  Using cached pandas-2.2.0-py38-gpl-win_amd64.whl.metadata (4.4 kB)
Collecting pillow==10.4.0 (from -r requirements.txt [line 11])
  Using cached pillow-10.4.0-py38-gpl-win_amd64.whl.metadata (164 bytes)
Collecting pyparsing==3.1.2 (from -r requirements.txt [line 12])
  Using cached pyparsing-3.1.2-py3-none-any.whl.metadata (1.4 kB)
```

[illegible]