MLOPS Assignment 1:- Report

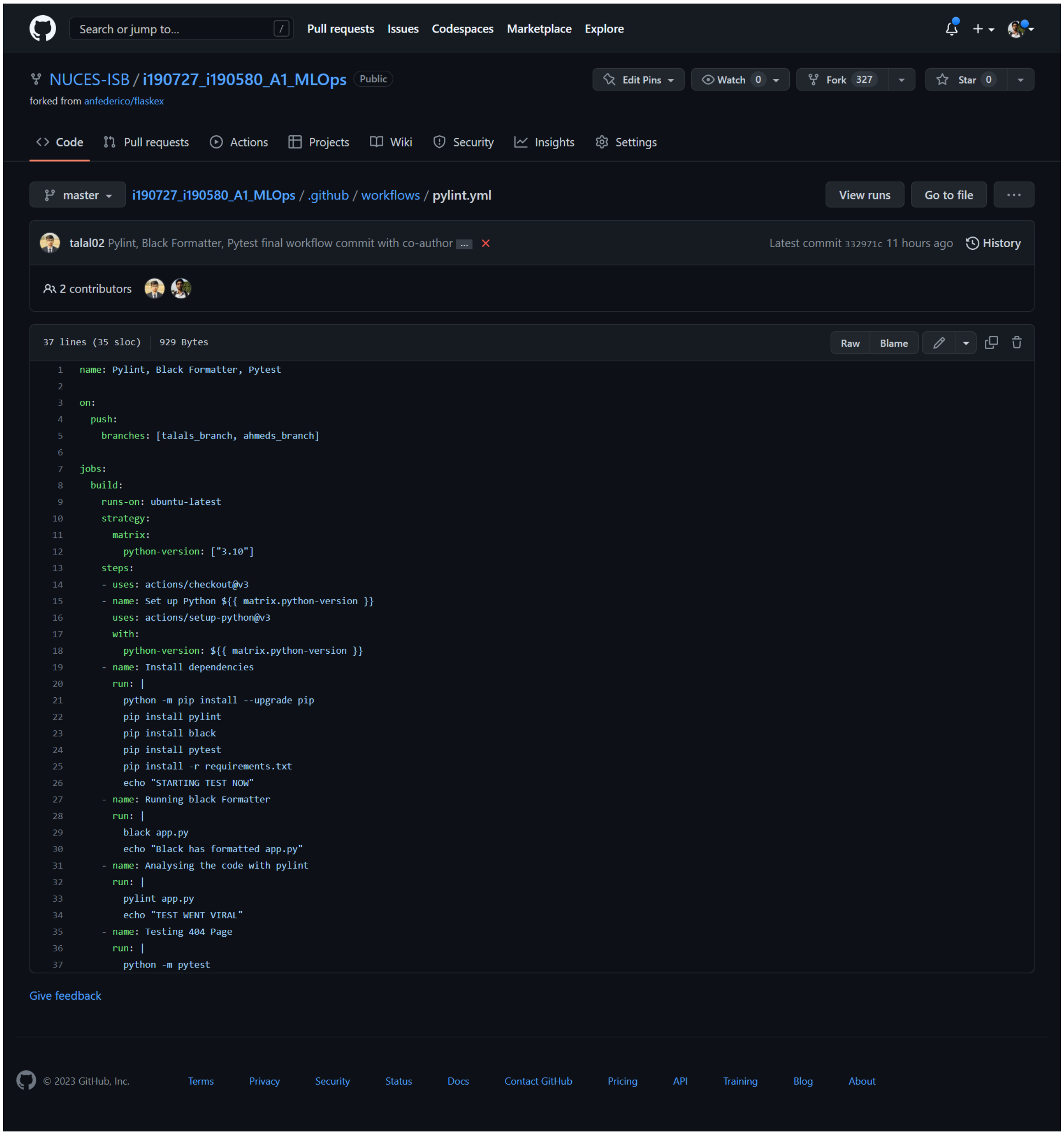
1. Introduction

Our repository	i190727_i190580_A1_MLOps
Original forked repository	anfederico/flaskex

The repository we forked is a simple flask example for quick prototypes and small applications. It contains the following features:

- Encrypted user authorization
- Database initialization
- New user signup
- User login/logout
- User settings
- Modern user interface
- Bulma framework
- Limited custom css/js
- Easily customizable

2. Github Actions

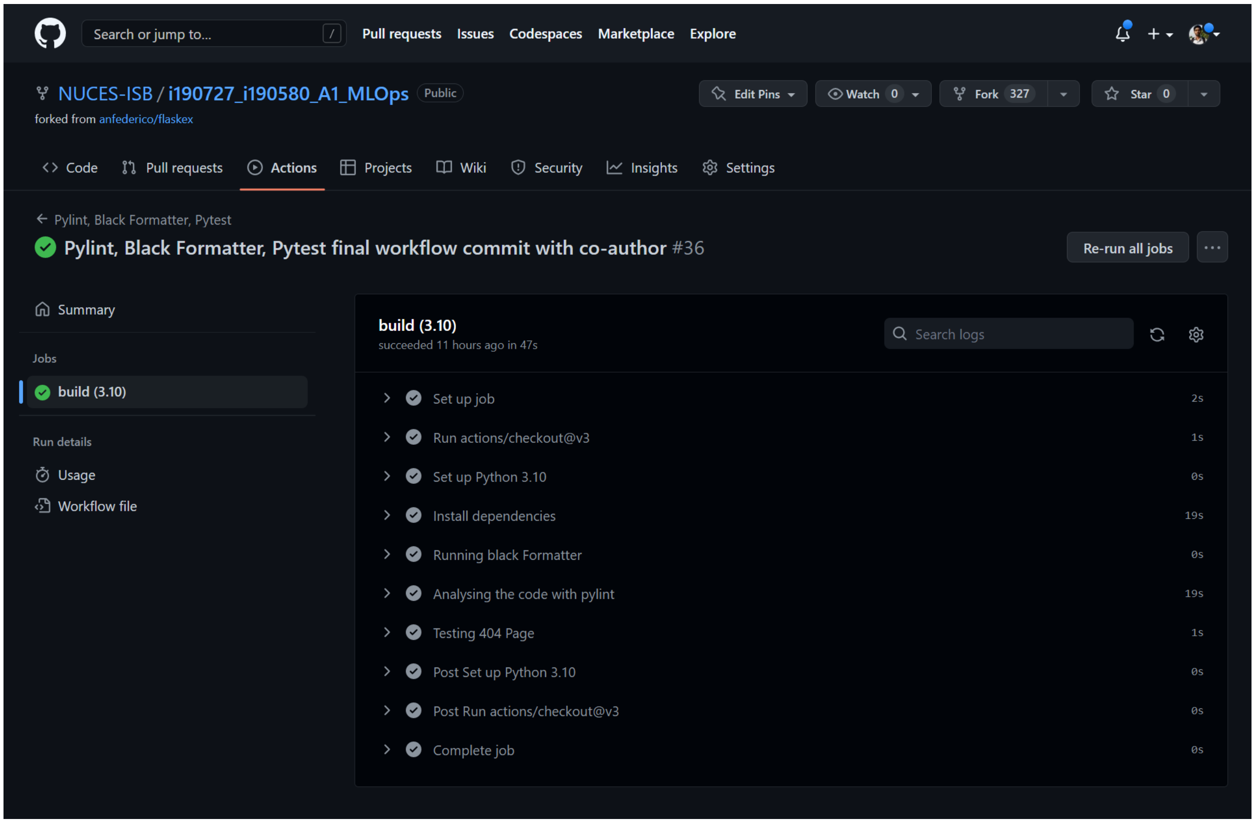


```yaml
name: Pylint, Black Formatter, Pytest

on:
  push:
    branches: [talals_branch, ahmeds_branch]

jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: ["3.10"]
    steps:
    - uses: actions/checkout@v3
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v3
      with:
        python-version: ${{ matrix.python-version }}
    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install pylint
        pip install black
        pip install pytest
        pip install -r requirements.txt
        echo "STARTING TEST NOW"
    - name: Running black Formatter
      run: |
        black app.py
        echo "Black has formatted app.py"
    - name: Analysing the code with pylint
      run: |
        pylint app.py
        echo "TEST WENT VIRAL"
    - name: Testing 404 Page
      run: |
        python -m pytest
```

The main workflow file we defined (as seen above) contains 3 main jobs.

1. It runs the Black code formatter to format the python code files according to a predefined standard.
2. It runs Pylint to check whether the code formatting is according to the pep-8 standard.
3. It runs a unit test (that we created ourselves) that tests whether the 404 Page not found error works correctly

The workflow eventually ran successfully on our repository:

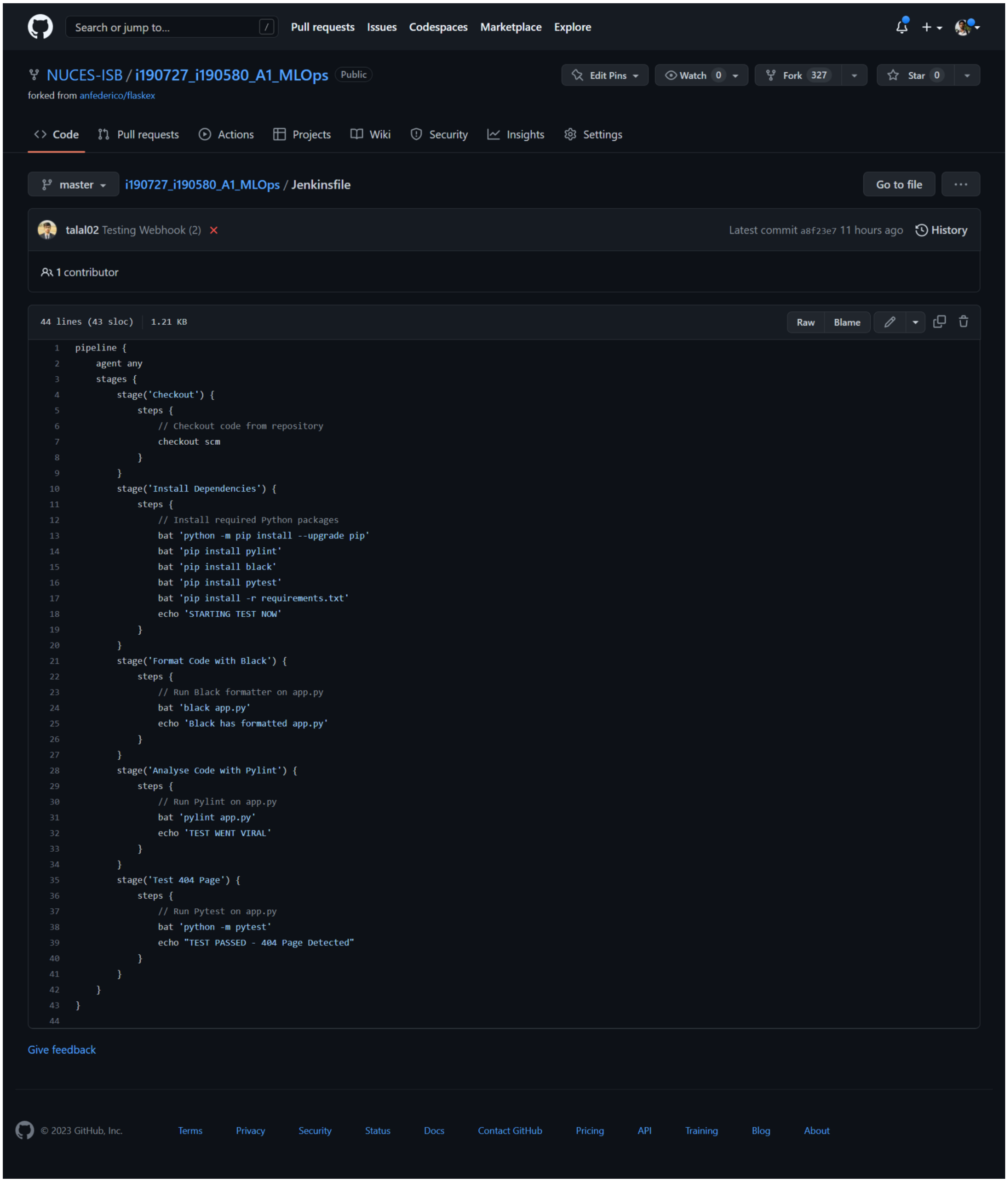


Pylint, Black Formatter, Pytest final workflow commit with co-author #36

build (3.10)
succeeded 11 hours ago in 47s

```
Set up job                         2s
Run actions/checkout@v3            1s
Set up Python 3.10                 0s
Install dependencies               19s
Running black Formatter            0s
Analysing the code with pylint     19s
Testing 404 Page                   1s
Post Set up Python 3.10            0s
Post Run actions/checkout@v3       0s
Complete job                       0s
```

3 - Jenkins



talal02 Testing Webhook (2) ✗

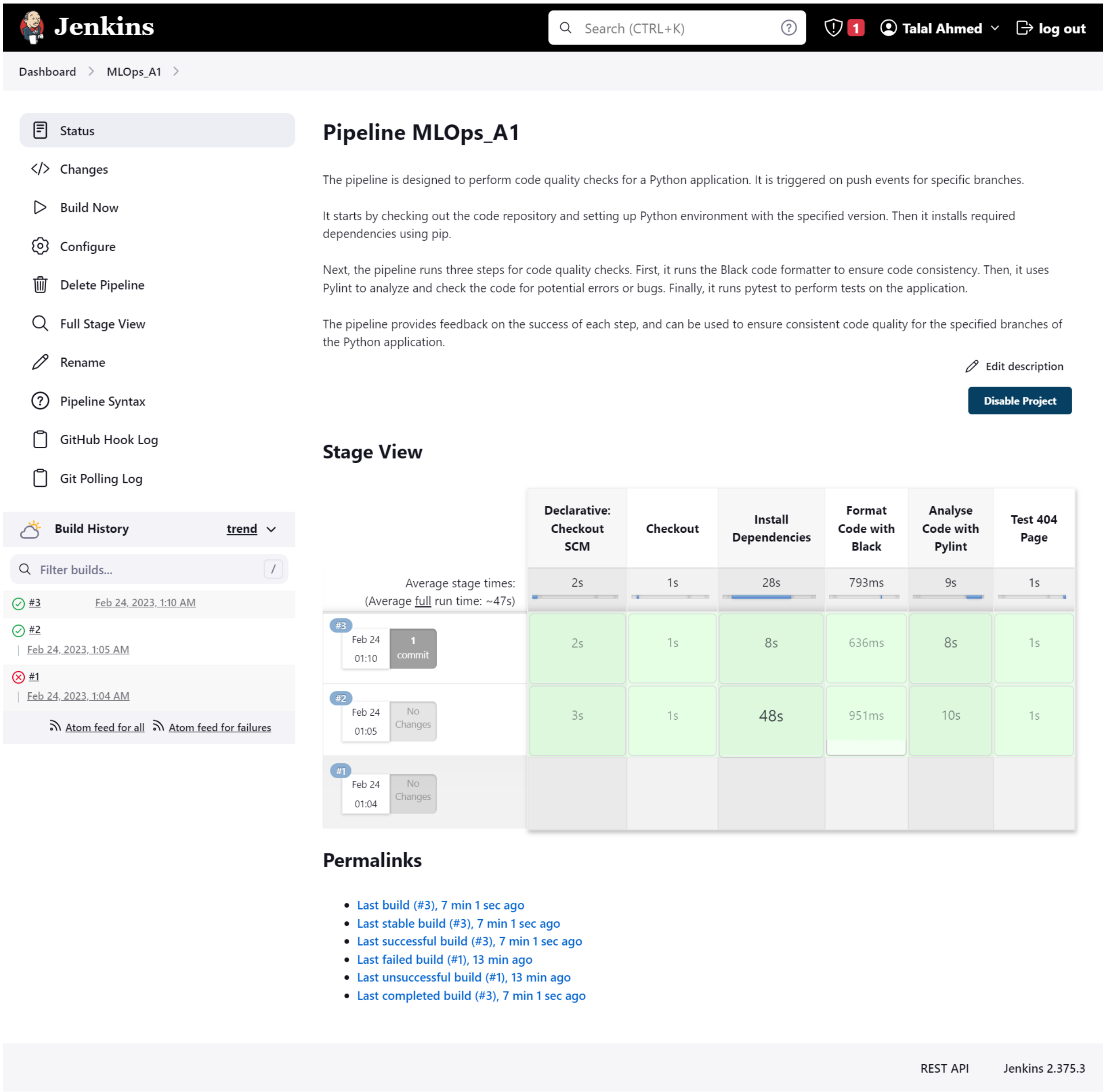
Latest commit a8f23e7 11 hours ago

44 lines (43 sloc) 1.21 KB

```
pipeline {
    agent any
    stages {
        stage('Checkout') {
            steps {
                // Checkout code from repository
                checkout scm
            }
        }
        stage('Install Dependencies') {
            steps {
                // Install required Python packages
                bat 'python -m pip install --upgrade pip'
                bat 'pip install pylint'
                bat 'pip install black'
                bat 'pip install pytest'
                bat 'pip install -r requirements.txt'
                echo 'STARTING TEST NOW'
            }
        }
        stage('Format Code with Black') {
            steps {
                // Run Black formatter on app.py
                bat 'black app.py'
                echo 'Black has formatted app.py'
            }
        }
        stage('Analyse Code with Pylint') {
            steps {
                // Run Pylint on app.py
                bat 'pylint app.py'
                echo 'TEST WENT VIRAL'
            }
        }
        stage('Test 404 Page') {
            steps {
                // Run Pytest on app.py
                bat 'python -m pytest'
                echo "TEST PASSED - 404 Page Detected"
            }
        }
    }
}

```

The same 3 jobs were also carried out in a jenkinsfile as seen above.

Those were also executed successfully:



Pipeline MLOps_A1

The pipeline is designed to perform code quality checks for a Python application. It is triggered on push events for specific branches.

It starts by checking out the code repository and setting up Python environment with the specified version. Then it installs required dependencies using pip.

Next, the pipeline runs three steps for code quality checks. First, it runs the Black code formatter to ensure code consistency. Then, it uses Pylint to analyze and check the code for potential errors or bugs. Finally, it runs pytest to perform tests on the application.

The pipeline provides feedback on the success of each step, and can be used to ensure consistent code quality for the specified branches of the Python application.

Build History

- ✓ #3 Feb 24, 2023, 1:10 AM
- ✓ #2 Feb 24, 2023, 1:05 AM
- ✗ #1 Feb 24, 2023, 1:04 AM

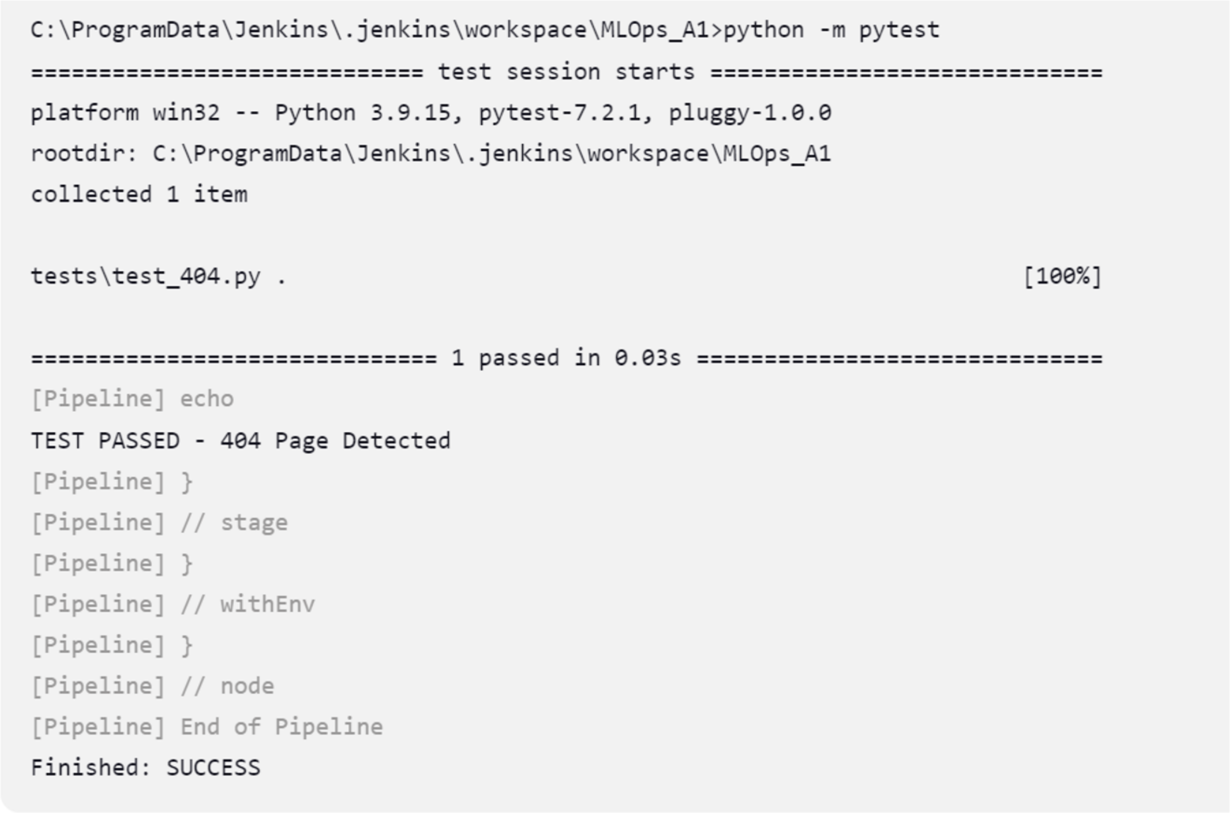
Atom feed for all Atom feed for failures

Stage View

	Declarative: Checkout SCM	Checkout	Install Dependencies	Format Code with Black	Analyse Code with Pylint	Test 404 Page
Average stage times: (Average full run time: ~47s)	2s	1s	28s	793ms	9s	1s
#3 Feb 24 01:10 1 commit	2s	1s	8s	636ms	8s	1s
#2 Feb 24 01:05 No Changes	3s	1s	48s	951ms	10s	1s
#1 Feb 24 01:04 No Changes						

Permalinks

- Last build (#3), 7 min 1 sec ago
- Last stable build (#3), 7 min 1 sec ago
- Last successful build (#3), 7 min 1 sec ago
- Last failed build (#1), 13 min ago
- Last unsuccessful build (#1), 13 min ago
- Last completed build (#3), 7 min 1 sec ago

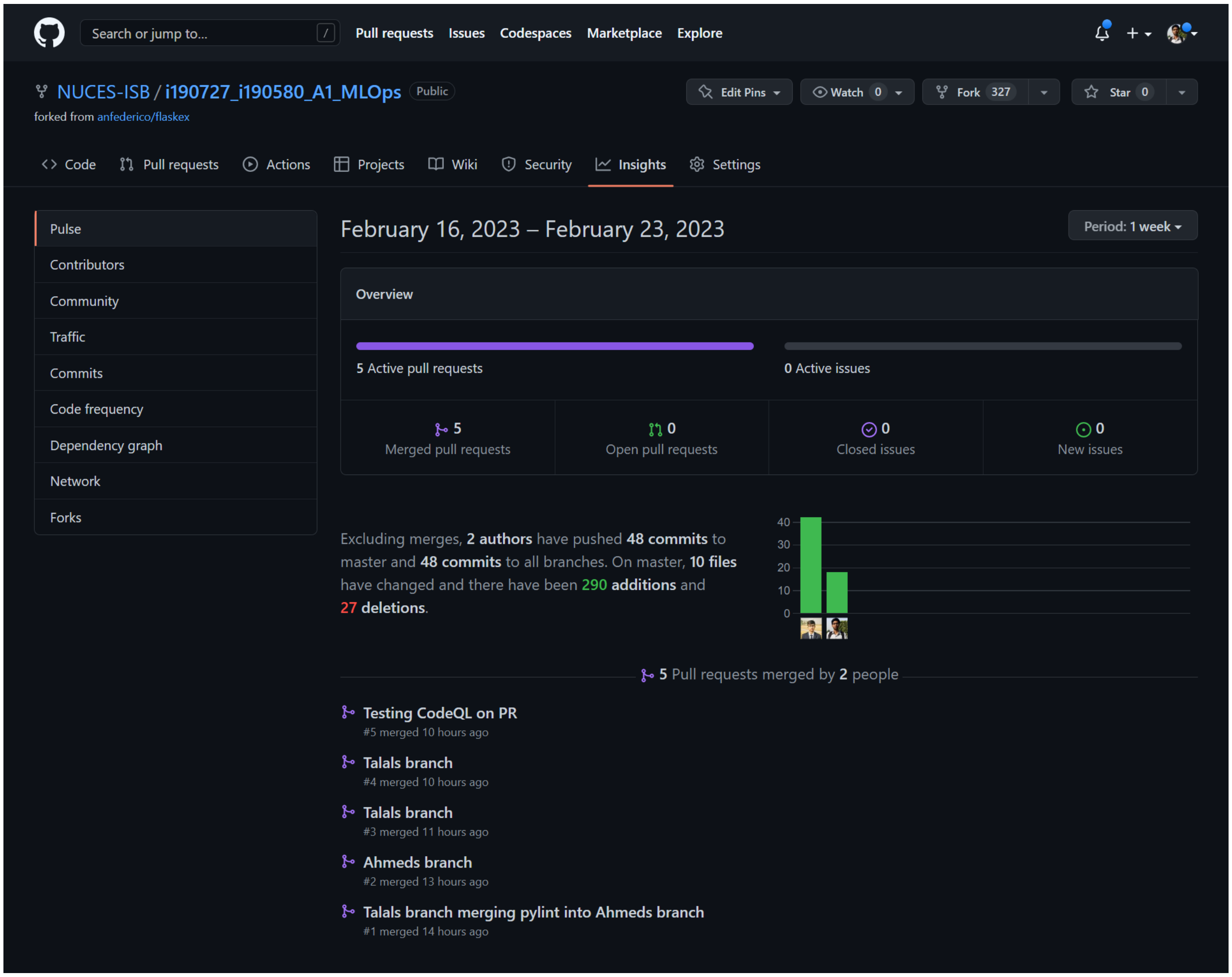


```
C:\ProgramData\Jenkins\.jenkins\workspace\MLOps_A1>python -m pytest
============================= test session starts ==============================
platform win32 -- Python 3.9.15, pytest-7.2.1, pluggy-1.0.0
rootdir: C:\ProgramData\Jenkins\.jenkins\workspace\MLOps_A1
collected 1 item

tests\test_404.py .                                                       [100%]

============================== 1 passed in 0.03s ===============================
[Pipeline] echo
TEST PASSED - 404 Page Detected
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

4 - How we did our work

We made 2 branches in addition to the master branch, one for each team member, and performed some work in each branch. In the end, my branch was pulled in Talal's branch and Talal's branch was pulled into master branch.

Summary of our work:



NUCES-ISB / i190727_i190580_A1_MLOps Public

forked from anfederico/flaskex

February 16, 2023 – February 23, 2023

Overview

5 Active pull requests

0 Active issues

5 Merged pull requests

0 Open pull requests

0 Closed issues

0 New issues

Excluding merges, **2 authors** have pushed **48 commits** to master and **48 commits** to all branches. On master, **10 files** have changed and there have been **290 additions** and **27 deletions**.

5 Pull requests merged by **2** people

Testing CodeQL on PR
#5 merged 10 hours ago

Talals branch
#4 merged 10 hours ago

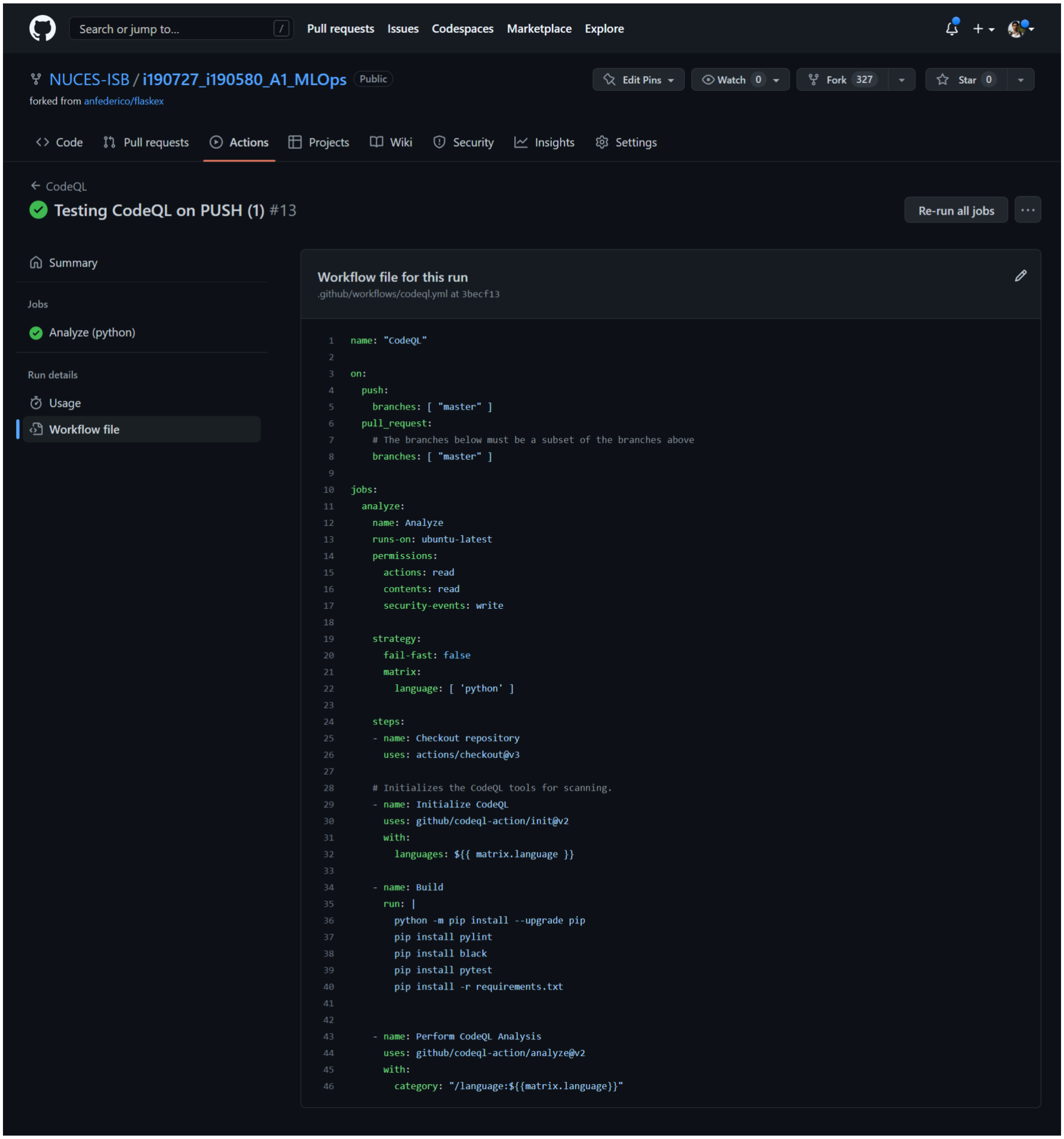
Talals branch
#3 merged 11 hours ago

Ahmeds branch
#2 merged 13 hours ago

Talals branch merging pylint into Ahmeds branch
#1 merged 14 hours ago

5 CodeQL

We used CodeQL to check the security status of our repository. CodeQL was available in predefined suggested GitHub Actions and Github automatically created a default CodeQL file for us.



Testing CodeQL on PUSH (1) #13

Workflow file for this run

.github/workflows/codeql.yml at 3becf13

```
name: "CodeQL"

on:
  push:
    branches: [ "master" ]
  pull_request:
    # The branches below must be a subset of the branches above
    branches: [ "master" ]

jobs:
  analyze:
    name: Analyze
    runs-on: ubuntu-latest
    permissions:
      actions: read
      contents: read
      security-events: write

    strategy:
      fail-fast: false
      matrix:
        language: [ 'python' ]

    steps:
    - name: Checkout repository
      uses: actions/checkout@v3

    # Initializes the CodeQL tools for scanning.
    - name: Initialize CodeQL
      uses: github/codeql-action/init@v2
      with:
        languages: ${{ matrix.language }}

    - name: Build
      run: |
        python -m pip install --upgrade pip
        pip install pylint
        pip install black
        pip install pytest
        pip install -r requirements.txt


    - name: Perform CodeQL Analysis
      uses: github/codeql-action/analyze@v2
      with:
        category: "/language:${{matrix.language}}"
```

6 CodeQL Results

The vulnerabilities that were suggested by CodeQl included the following:

1. Flask app is run in debug mode.Running a Flask application with debug mode enabled may allow an attacker to gain access through the Werkzeug debugger. It suggested us to ensure that Flask applications that are run in a production environment have debugging disabled.
2. Including a resource from an untrusted source or using an untrusted channel may allow an attacker to include arbitrary code in the response. When including an external resource (for example, a `script` element or an `iframe` element) on a page, it is important to ensure that the received data is not malicious. (This is referring to `<script src="https://cdnjs.cloudflare.com/ajax/libs/jqueryui/1.12.1/jquery-ui.min.js"></script>`)

—- End —-