

## Continuous Integration - Using Github Actions - Setting up workflow

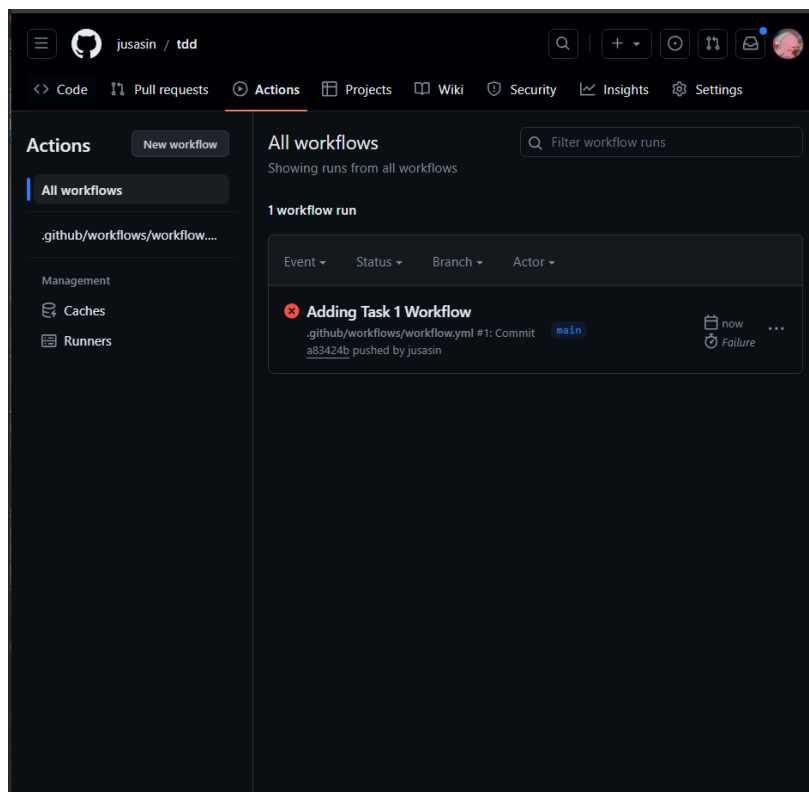
Justin Vence Llamas

Link to GitHub: <https://github.com/jusasin/tdd>

### Task 1:

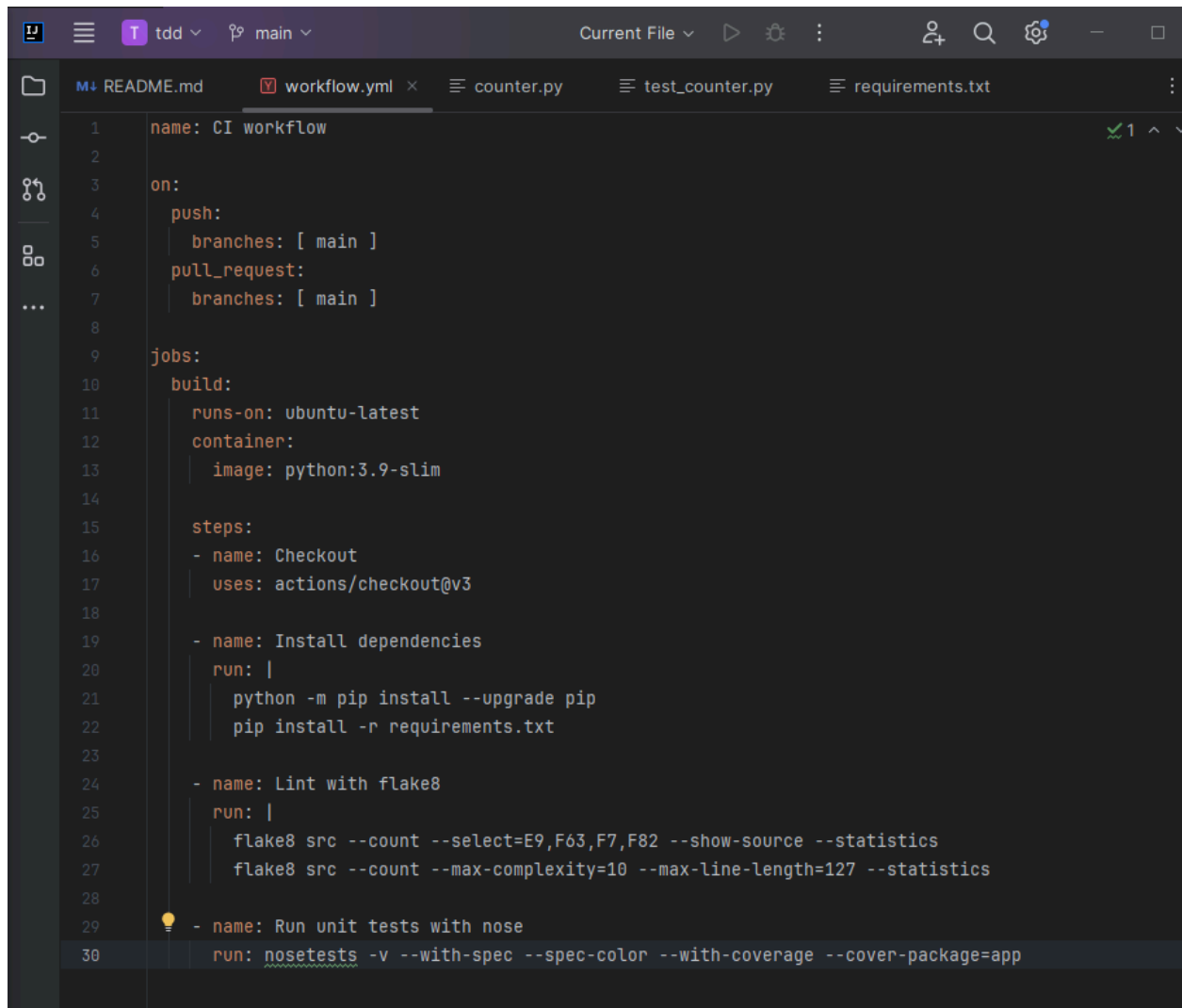
```
1  name: CI workflow
2
3  on:
4    push:
5      branches: [ main ]
6    pull_request:
7      branches: [ main ]
8
9  jobs:
10   build:
11     runs-on: ubuntu-latest
12     container:
13       image: python:3.9-slim
```

- Picture of the code I did for task 1



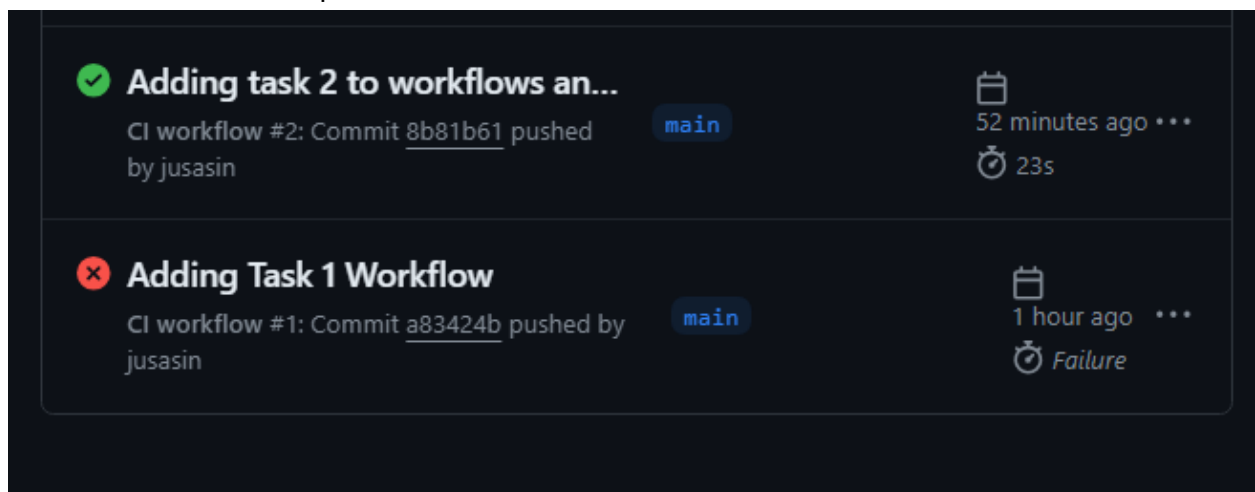
- Picture of the Actions tab from my GitHub

## Task 2:



```
1 name: CI workflow
2
3 on:
4   push:
5     branches: [ main ]
6   pull_request:
7     branches: [ main ]
8
9 jobs:
10  build:
11    runs-on: ubuntu-latest
12    container:
13      image: python:3.9-slim
14
15    steps:
16      - name: Checkout
17        uses: actions/checkout@v3
18
19      - name: Install dependencies
20        run: |
21          python -m pip install --upgrade pip
22          pip install -r requirements.txt
23
24      - name: Lint with flake8
25        run: |
26          flake8 src --count --select=E9,F63,F7,F82 --show-source --statistics
27          flake8 src --count --max-complexity=10 --max-line-length=127 --statistics
28
29      - name: Run unit tests with nose
30        run: nosetests -v --with-spec --spec-color --with-coverage --cover-package=app
```

- Picture of the steps from task 2.



- Adding my workflow to GitHub, picture of the action tab.

### Creating Delete Test Case:

```
def test_delete_a_counter(self):
    """Test to delete a counter"""
    #Create a counter
    result = self.client.post('/counters/deleteCounter')
    #Check that it was created successfully
    self.assertEqual(result.status_code, status.HTTP_201_CREATED)
    #Delete the created counter
    deleteResult = self.client.delete('/counters/deleteCounter')
    #Check that there is no counter that exists
    self.assertEqual(deleteResult.status_code, status.HTTP_204_NO_CONTENT)
    #delete a counter that doesnt even exists
    notReal = self.client.delete('counters/fakeCounter')
    #check that it is not found
    self.assertEqual(notReal.status_code, status.HTTP_404_NOT_FOUND)
```

- For the test case I created a new counter and made sure that it was created successfully. Then I tried to delete the counter I already had and checked that it no longer exists afterwards. Then I tried to delete a counter that did not exist and checked that by using NOT\_FOUND status code.

```
Justin@Justin MINGW64 ~/CS472/cs472-group4/tdd (main)
$ nosetests

Counter tests
- It should create a counter
- Test to delete a counter (FAILED)
- It should return an error for duplicates
- Test to read a counter
- Tests to update Counters

=====
FAIL: Test to delete a counter
=====
```

- Picture of running nosetests of the new test, it had an error code with 405 != 204 and we can see that my delete test counter failed.

```
@app.route('/counters/<name>', methods=['DELETE'])
def delete_counter(name):
    """Delete a counter"""
    app.logger.info(f"Request to delete counter: {name}")
    if name not in COUNTERS:
        return {"Message": f"Counter {name} not found"}, status.HTTP_404_NOT_FOUND
    del COUNTERS[name]
    return '', status.HTTP_204_NO_CONTENT
```

- Adding the new function for delete. Really just copied and paste most of the actions from create and changed a few things up.

```
Justin@Justin MINGW64 ~/CS472/cs472-group4/tdd (main)
$ nosetests

Counter tests
- It should create a counter
- Test to delete a counter
- It should return an error for duplicates
- Test to read a counter
- Tests to update Counters

Name          Stmts  Miss  Cover   Missing
-----
src\counter.py    27     0   100%
src\status.py     6     0   100%
-----
TOTAL             33     0   100%
-----
Ran 5 tests in 0.189s

OK
```

- Picture of adding the new function and showing that it passed and there is 100% coverage on all the lines.