CS 303P Software Engineering Lab: Jenkins Demo Report

Harshadeep Donapati

December 1, 2022

1 Folder Structure

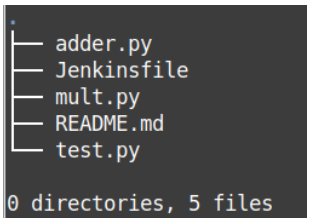

Figure 1: Folder structure.

`adder.py`
This file contains a function to add 2 numbers `x` and `y` and returns the result as `x + y`.

```
def add(x, y):
    return x + y
```

Figure 2: Code for adder.py

`mult.py`
This file contains a function to add 2 numbers `x` and `y` and returns the result as `x * y`.

```
def multiply(x, y):
    return x * y
```

Figure 3: Code for mult.py

`test.py`

- This file contains unit test cases which are written using python's inbuilt `unittest` testing library.

- There are 5 test cases. 2 test cases for addition, and other 3 test cases for multiplication.

- These test cases are written in such a way that first three test cases pass and the last two test cases fail.

```python
import unittest
from adder import add
from mult import multiply


class Test(unittest.TestCase):
    def test_addition_1(self):
        lst = [1, 2]
        result = 3
        self.assertEqual(add(lst[0], lst[1]), result)

    def test_addition_2(self):
        lst = [2, 3]
        result = 5
        self.assertEqual(add(lst[0], lst[1]), result)

    def test_multiplication_1(self):
        lst = [7, 7]
        result = 49
        self.assertEqual(multiply(lst[0], lst[1]), result)

    def test_multiplication_2(self):
        lst = [8, 8]
        result = 72
        self.assertEqual(multiply(lst[0], lst[1]), result)

    def test_multiplication_3(self):
        lst = [10, 9]
        result = 100
        self.assertEqual(multiply(lst[0], lst[1]), result)


if __name__ == '__main__':
    unittest.main()
```

Figure 4: Code for test.py

`Jenkinsfile`

- The pipeline consists of 4 stages.
- First stage is cloning my github repository.(Link added later.)
- Second stage is building the code `adder.py`
- Third stage is building the code `mult.py`
- Last stage is testing the written test cases with `test.py`

```
pipeline {
    agent any
    stages {
        stage('Clone Git') {
            steps {
                git 'https://github.com/harsha-deep/Jenkins-Demo.git'
            }
        }
        stage('Code Build add') {
            steps {
                sh 'chmod u+x adder.py'
                sh 'python3 adder.py'
            }
        }
        stage('Code Build mult') {
            steps {
                sh 'chmod u+x mult.py'
                sh 'python3 mult.py'
            }
        }
        stage('Test Code') {
            steps {
                sh 'chmod u+x test.py'
                sh 'python3 test.py'
            }
        }
    }
}
```

Figure 5: Code for Jenkinsfile

2 Results and Analysis

```
...FF
======================================================================
FAIL: test_multiplication_2 (__main__.Test)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/var/lib/jenkins/workspace/Jenkins Demo Test 1/test.py", line 25, in test_multiplication_2
    self.assertEqual(multiply(lst[0], lst[1]), result)
AssertionError: 64 != 72

======================================================================
FAIL: test_multiplication_3 (__main__.Test)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/var/lib/jenkins/workspace/Jenkins Demo Test 1/test.py", line 30, in test_multiplication_3
    self.assertEqual(multiply(lst[0], lst[1]), result)
AssertionError: 90 != 100

----------------------------------------------------------------------
Ran 5 tests in 0.001s

FAILED (failures=2)
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

Figure 6: Jenkins console output

From the above figure, we can see that 5 tests were executed. Out of those, `test_multiplication_2` and `test_multiplication_3` were failed.

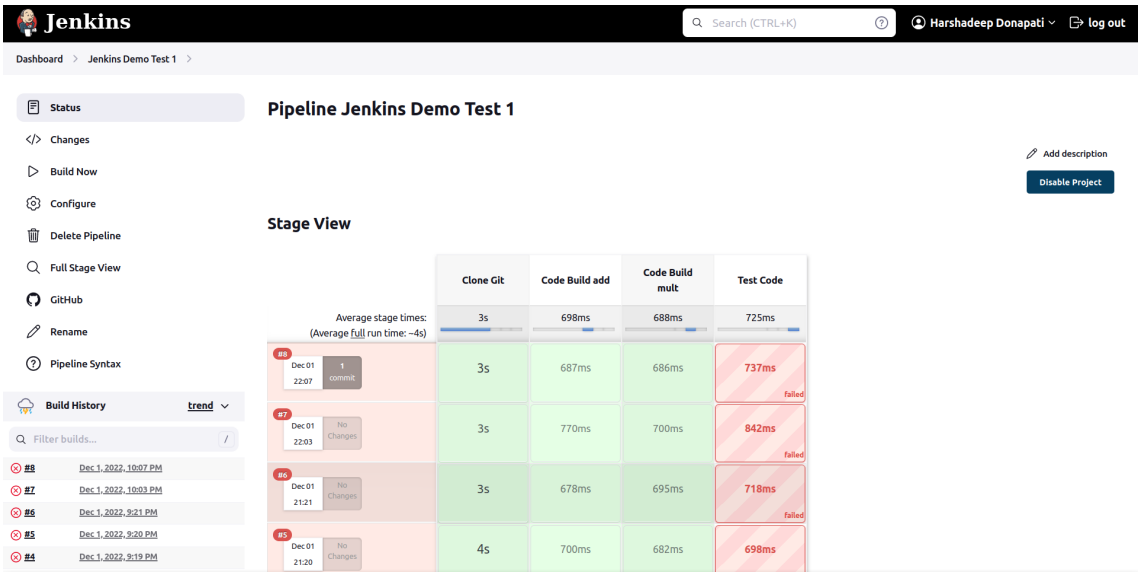

Figure 7: Jenkins App screenshot.

3 Important Links

- `https://github.com/harsha-deep/Jenkins-Demo/tree/master`
  This is the github repository link. Please refer to the `master` branch of the repository.

4 References

- `https://stackoverflow.com/q/3371255/15069364`
  This stackoverflow discussion has helped me a lot to write test cases in python.