

Ibrahima Diallo
Kura Labs
Summer 2021
Instructors: Daniel Adeyanju
Tyrone Sanderson

Documentation - Deployment 3 - Python

Goal for this deployment:

1. Make a pipeline script to run a test build for the python application called Add2Vals using the source codes provided for this assignment.
2. If the build is successful, add features to the application, and rebuild.
3. Notice the failures, and fix the different issues that arise.

Procedure:

Step 1: I forked (https://github.com/kura-labs-org/DEPLOY03_TEST)

Then, I configured an EC2 on AWS, and follow the steps provided in this link to set up Jenkins: https://github.com/kura-labs-org/DEPLOY01_HELLO_WORLD/blob/main/Deployment%201.pdf

And, I reused the processes I applied to build the EC2 for the first and the second deployment. The reason for this repetitive work was to practice more with the tools.

Step 2: Then, I scanned the git repository. The first test build was successful.

Here is a screenshot.

The screenshot shows the Jenkins web interface for a job named 'Python Deployment3'. The left sidebar contains a navigation menu with options like 'Up', 'Status', 'Configure', 'Scan Repository Now', 'Scan Repository Log', 'Multibranch Pipeline Events', 'Delete Multibranch Pipeline', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'GitHub', 'Rename', 'Pipeline Syntax', and 'Credentials'. The main content area displays the job name 'Python Deployment3' and a table of branches. The table has columns for 'S' (Status), 'W' (Webhook), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The 'main' branch is listed with a success status, a duration of 2 min 17 sec, and a last build duration of 15 sec. There are also links for 'Legend', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

Step 3: I added a new feature that would multiply the two input arguments and return the result. In case one/both argument(s) is/are (a) strings, an error message will be returned.

In order to achieve this, I added new functions in the source codes provided previously, and I also had to update the `test_calc.py` for every function in order for the test to be successful.

- In `calc.py`, I added the function `def multiply(arg1, arg2)`
- In `add2vals.py`, I added this if statement to do the multiplication when the arguments are not strings.

if argnumbers == 2 and argnumbers != str

- In `test_calc.py`, I added these five functions in order to test for the multiplication feature.

```
def test_multiply_integers(self)

def test_multiply_floats(self)

def test_multiply_strings(self)

def test_multiply_string_and_integer(self)

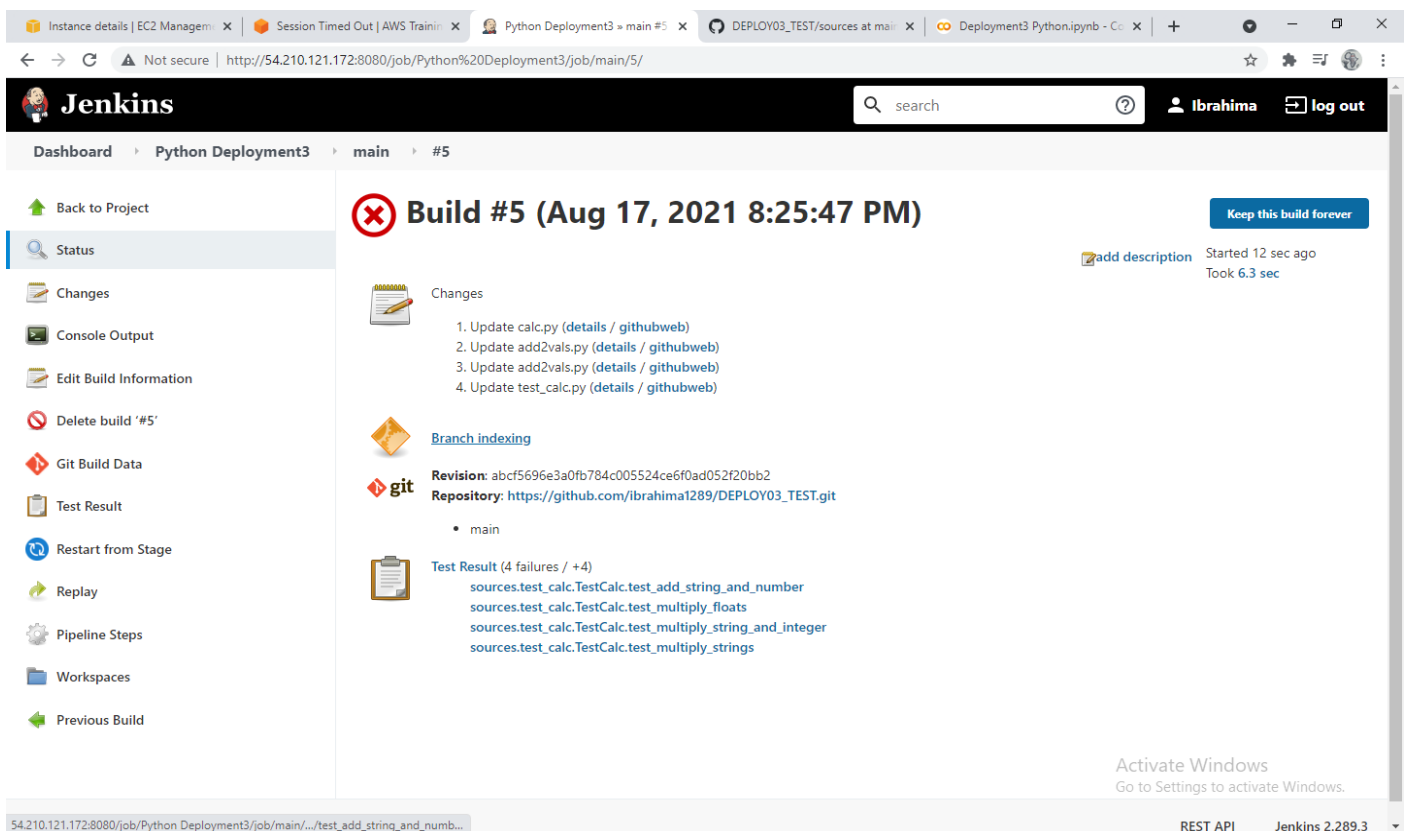
def test_multiply_string_and_number(self)
```

Here is the link to the github repository containing the the source codes:

https://github.com/ibrahima1289/DEPLOY03_TEST/tree/main/sources

Step 4: After updating the source files on github, I sent a new build on Jenkins. This time, it was not successful.

Here is the screenshot of the failed test.



The screenshot shows the Jenkins web interface for a failed build. The browser tabs include 'Instance details | EC2 Managem...', 'Session Timed Out | AWS Traini...', 'Python Deployment3 » main #5', 'DEPLOY03_TEST/sources at mai...', and 'Deployment3 Python.ipynb - Co...'. The address bar shows 'http://54.210.121.172:8080/job/Python%20Deployment3/job/main/5/'. The Jenkins header shows the logo, a search bar, and the user 'Ibrahima' with a 'log out' button. The breadcrumb trail is 'Dashboard > Python Deployment3 > main > #5'. The left sidebar contains links: 'Back to Project', 'Status' (selected), 'Changes', 'Console Output', 'Edit Build Information', 'Delete build '#5'', 'Git Build Data', 'Test Result', 'Restart from Stage', 'Replay', 'Pipeline Steps', 'Workspaces', and 'Previous Build'. The main content area is titled 'Build #5 (Aug 17, 2021 8:25:47 PM)' with a red 'X' icon. A 'Keep this build forever' button is in the top right. Below the title, there's a 'Changes' section with a list of updates: '1. Update calc.py (details / githubweb)', '2. Update add2vals.py (details / githubweb)', '3. Update add2vals.py (details / githubweb)', and '4. Update test_calc.py (details / githubweb)'. A 'Branch indexing' section shows the 'Revision: abcf5696e3a0fb784c005524ce6f0ad052f20bb2' and 'Repository: https://github.com/ibrahima1289/DEPLOY03_TEST.git'. A 'Test Result' section shows '4 failures / +4' with links to specific test failures: 'sources.test_calc.TestCalc.test_add_string_and_number', 'sources.test_calc.TestCalc.test_multiply_floats', 'sources.test_calc.TestCalc.test_multiply_string_and_integer', and 'sources.test_calc.TestCalc.test_multiply_strings'. At the bottom right, there's a 'REST API' link and 'Jenkins 2.289.3' version information.

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and a user profile for 'Ibrahima' with a 'log out' button. The breadcrumb trail is 'Dashboard > Python Deployment3 > main > #5'. The left sidebar contains links to 'Back to Project', 'Status', 'Changes', 'Console Output' (which is selected), 'View as plain text', 'Edit Build Information', 'Delete build '#5'', 'Git Build Data', 'Test Result', 'Restart from Stage', 'Replay', 'Pipeline Steps', 'Workspaces', and 'Previous Build'. The main content area is titled 'Console Output' with a red 'X' icon. It displays the following log output:

```
Branch indexing
28:25:47 Connecting to https://api.github.com using ibrahima1289/*****
Obtained Jenkinsfile from abcf5696e3a0fb784c005524ce6f0ad052f20bb2
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Python_Deployment3_main
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: NONE
using credential Python-Deployment3-id
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Python_Deployment3_main/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/ibrahima1289/DEPLOY03_TEST.git # timeout=10
Fetching without tags
Fetching upstream changes from https://github.com/ibrahima1289/DEPLOY03_TEST.git
> git --version # timeout=10
> git --version # 'git version 2.32.0'
using GIT_ASKPASS to set credentials
> git fetch --no-tags --force --progress -- https://github.com/ibrahima1289/DEPLOY03_TEST.git +refs/heads/main:refs/remotes/origin/main #
timeout=10
Checking out Revision abcf5696e3a0fb784c005524ce6f0ad052f20bb2 (main)
> git config core.sparsecheckout # timeout=10
> git checkout -f abcf5696e3a0fb784c005524ce6f0ad052f20bb2 # timeout=10
Commit message: "Update test_calc.py"
> git rev-list --no-walk 7c92e928f76af4e3618cb53fba650e563dc4b216 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
```

An 'Activate Windows' watermark is visible in the bottom right corner of the console output area.

Step 5 Error Troubleshooting: I found three reasons why the test failed

1. First, after creating the Jenkins file, I had some issues there, but I figured that the access token that I was using was not the right one. So I created a new one and I was able to connect into the github repository.

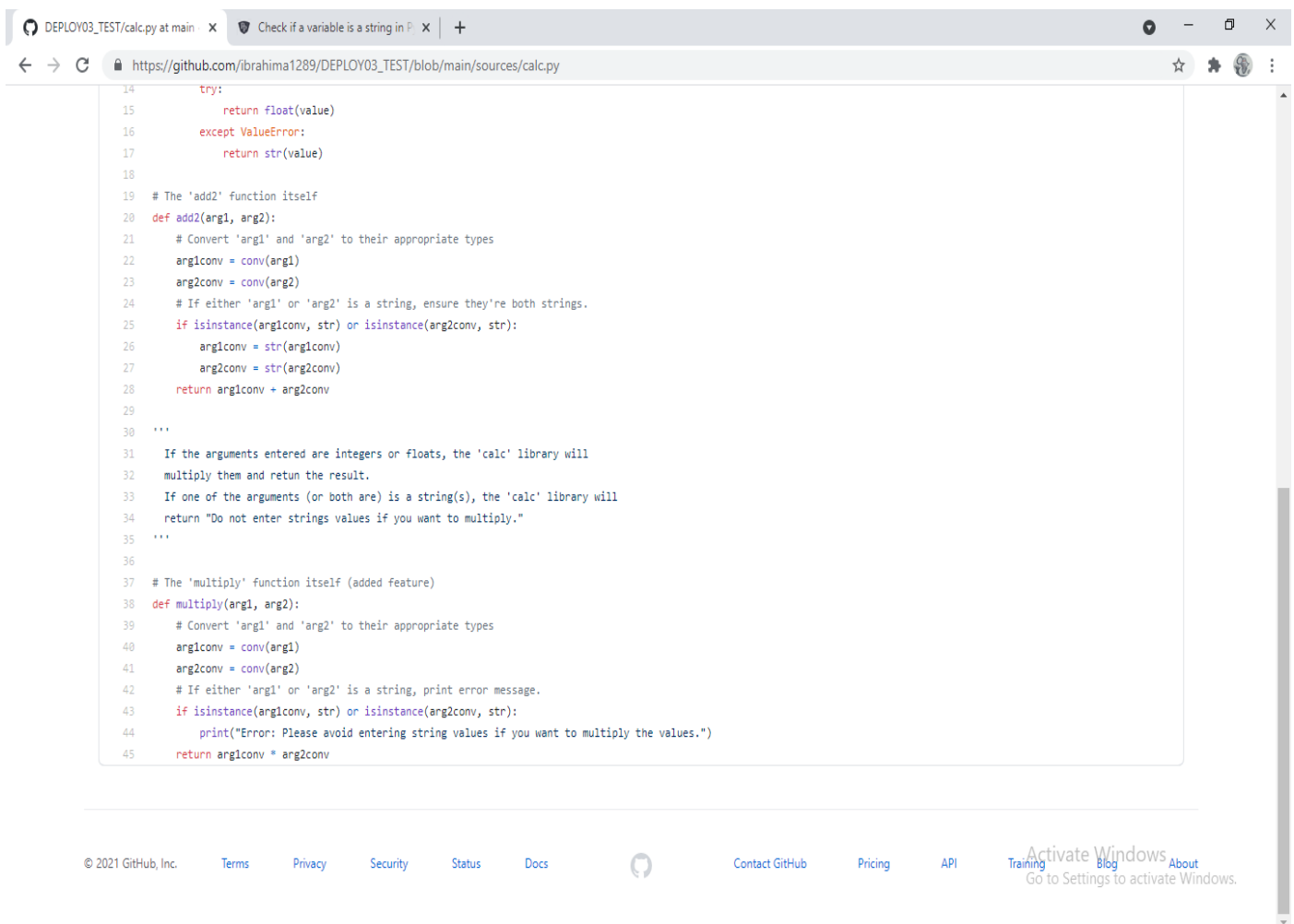
2. After reading the console output, I noticed that only the features I have added failed the test (see screenshot below).

Furthermore, I read in the console output an error where I **printed** the error message for the string multiplication as this:

```
print('Error: Please avoid entering string values if you want to multiply the values.')
```

I switch it into a **return** statement:

```
return 'Error: Please avoid entering string values if you want to multiply the values.'
```

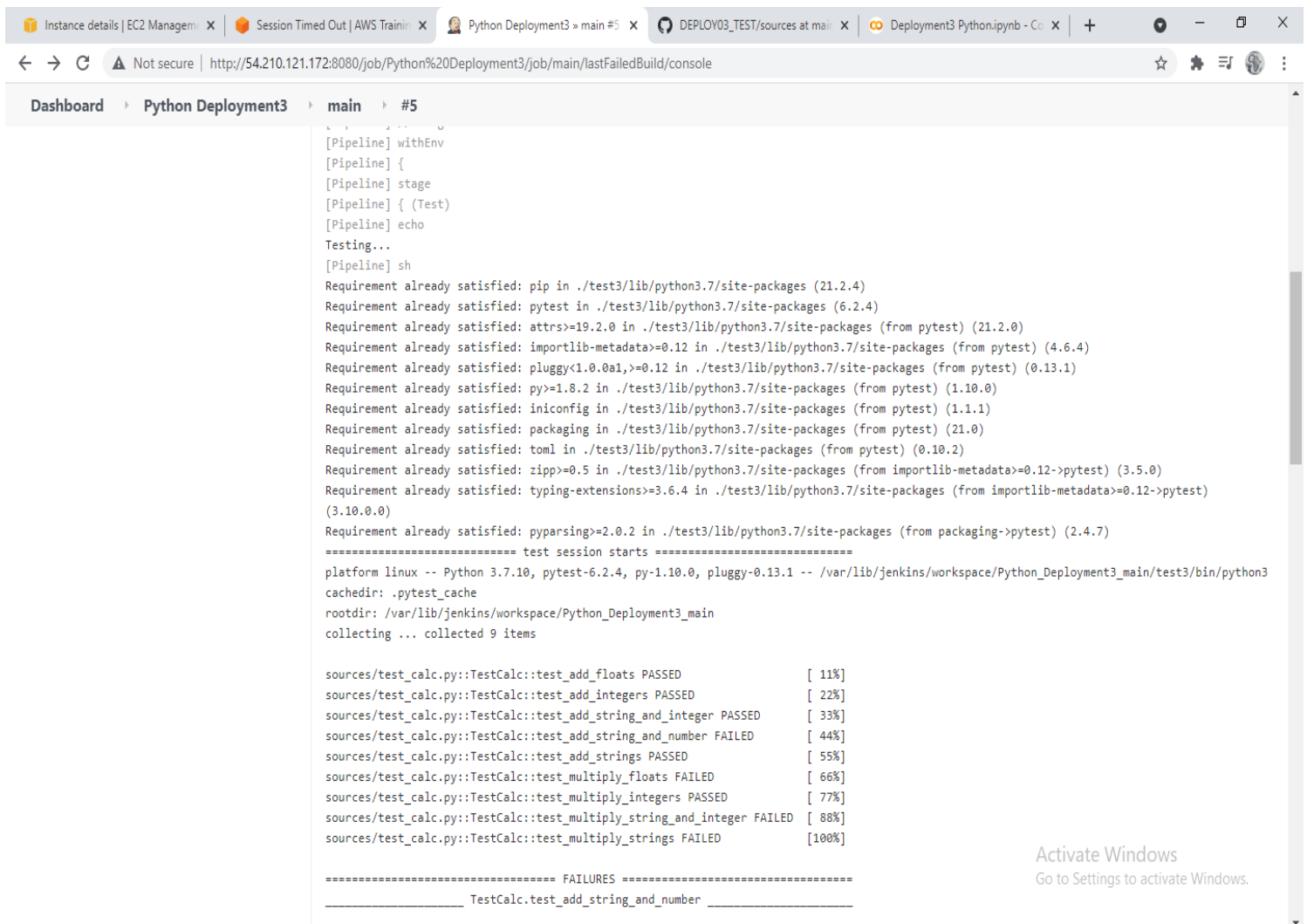


3. I forgot to change the names of the new functions for the `test_calc.py` code for the multiplication. Here is how I updated the name (in blue) of the functions.

Note: Keeping the naming consistent is an absolute necessity in order for the program to work properly.

```
def test_add_integers(self) =====> def test_multiply_integers(self)
def test_add_floats(self) =====> def test_multiply_floats(self)
def test_add_strings(self) =====> def test_multiply_strings(self)
def test_add_string_and_integer(self) ==> def test_multiply_string_and_integer(self)
def test_add_string_and_number(self) => def test_multiply_string_and_number(self)
```

The next two screenshots are from the failed build shown on the console output.



Dashboard ▸ Python Deployment3 ▸ main ▸ #5

```
(from packaging->pytest) (2.4.7)
===== test session starts =====
platform linux -- Python 3.7.10, pytest-6.2.4, py-1.10.0, pluggy-0.13.1 --
/var/lib/jenkins/workspace/Python_Deployment3_main/test3/bin/python3
cachedir: .pytest_cache
rootdir: /var/lib/jenkins/workspace/Python_Deployment3_main
collecting ... collected 9 items

sources/test_calc.py::TestCalc::test_add_floats PASSED [ 11%]
sources/test_calc.py::TestCalc::test_add_integers PASSED [ 22%]
sources/test_calc.py::TestCalc::test_add_string_and_integer PASSED [ 33%]
sources/test_calc.py::TestCalc::test_add_string_and_number FAILED [ 44%]
sources/test_calc.py::TestCalc::test_add_strings PASSED [ 55%]
sources/test_calc.py::TestCalc::test_multiply_floats FAILED [ 66%]
sources/test_calc.py::TestCalc::test_multiply_integers PASSED [ 77%]
sources/test_calc.py::TestCalc::test_multiply_string_and_integer FAILED [ 88%]
sources/test_calc.py::TestCalc::test_multiply_strings FAILED [100%]

===== FAILURES =====
_____ TestCalc.test_add_string_and_number _____

self = <test_calc.TestCalc testMethod=test_add_string_and_number>

    def test_add_string_and_number(self):
        """
        Test the addition of a string and a float returns an error
        """
        result = calc.add2('abc', '5.5')
> self.assertEqual(result, 'Error: Please avoid entering string values if you
want to multiply the values.')
E   AssertionError: 'abc5.5' != 'Error: Please avoid entering string values if
want to multiply the values.'
E   - abc5.5
E   + Error: Please avoid entering string values if you want to multiply the
values.

sources/test_calc.py:84: AssertionError
```

```
51 def test_multiply_integers(self):
52     """
53     Test that the multiplication of two integers returns the correct total
54     """
55     result = calc.multiply(1, 2)
56     self.assertEqual(result, 2)
57
58 def test_multiply_floats(self):
59     """
60     Test that the multiplication of two floats returns the correct result
61     """
62     result = calc.add2('10.5', 2)
63     self.assertEqual(result, 21)
64
65 def test_multiply_strings(self):
66     """
67     Test the multiplication of two strings returns an error
68     """
69     result = calc.multiply('abc', 'def')
70     self.assertEqual(result, 'Error: Please avoid entering string values if you want to multiply
71
72 def test_multiply_string_and_integer(self):
73     """
74     Test the addition of a string and an integer returns an error
75     """
76     result = calc.add2('abc', 3)
77     self.assertEqual(result, 'Error: Please avoid entering string values if you want to multiply
78
79 def test_add_string_and_number(self):
80     """
81     Test the addition of a string and a float returns an error
82     """
83     result = calc.add2('abc', '5.5')
84     self.assertEqual(result, 'Error: Please avoid entering string values if you want to multiply
85
86 if __name__ == '__main__':
87     unittest.main()
```

Activate Windows
Go to Settings to activate Windows.

Step 6. After I fixed the failed test build, the new build was successful.

See screenshots of the successful test build below.

The screenshot shows the Jenkins web interface for a job named 'Python Deployment3'. The top navigation bar includes the Jenkins logo, a search bar, and the user 'Ibrahima' with a 'log out' button. The left sidebar contains various navigation links: 'Up', 'Status', 'Configure', 'Scan Repository Now', 'Scan Repository Log', 'Multibranch Pipeline Events', 'Delete Multibranch Pipeline', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'GitHub', 'Rename', 'Pipeline Syntax', and 'Credentials'. The main content area displays the 'Python Deployment3' job details. It shows a table of build history for the 'main' branch, with columns for 'S' (Status), 'W' (Webhook), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The table shows a successful build #6 with a duration of 6.2 sec. Below the table, there are links for 'Icon: S M L', 'Legend', and 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. A 'Disable Multibranch Pipeline' button is visible in the top right corner of the main content area. At the bottom right, there is a 'Activate Windows' watermark.

S	W	Name	Last Success	Last Failure	Last Duration
✓	🔗	main	11 sec - #6	17 min - #5	6.2 sec

We can see here that all of the added functions (features) and the old ones passed the test.

Instance details | EC2 Managemen...

Session Timed Out | AWS Trainin...

Python Deployment3 » main #6 |

Deployment3 Python.ipynb - Co...

Not secure | http://54.210.121.172:8080/job/Python%20Deployment3/job/main/lastSuccessfulBuild/console

Dashboard ▾ Python Deployment3 ▾ main ▾ #6

Requirement already satisfied: typing-extensions>=3.6.4 in ./test3/lib/python3.7/site-packages (from importlib-metadata>=0.12->pytest) (3.10.0.0)
Requirement already satisfied: pyparsing>=2.0.2 in ./test3/lib/python3.7/site-packages (from packaging->pytest) (2.4.7)
===== test session starts =====
platform linux -- Python 3.7.10, pytest-6.2.4, py-1.10.0, pluggy-0.13.1 -- /var/lib/jenkins/workspace/Python_Deployment3_main/test3/bin/python3
cachedir: .pytest_cache
rootdir: /var/lib/jenkins/workspace/Python_Deployment3_main
collecting ... collected 9 items













sources/test_calc.py::TestCalc::test_add_floats PASSED [11%]
sources/test_calc.py::TestCalc::test_add_integers PASSED [22%]
sources/test_calc.py::TestCalc::test_add_string_and_integer PASSED [33%]
sources/test_calc.py::TestCalc::test_add_string_and_number PASSED [44%]
sources/test_calc.py::TestCalc::test_add_strings PASSED [55%]
sources/test_calc.py::TestCalc::test_multiply_floats PASSED [66%]
sources/test_calc.py::TestCalc::test_multiply_integers PASSED [77%]
sources/test_calc.py::TestCalc::test_multiply_string_and_integer PASSED [88%]
sources/test_calc.py::TestCalc::test_multiply_strings PASSED [100%]

- generated xml file: /var/lib/jenkins/workspace/Python_Deployment3_main/test-reports/results.xml -
===== 9 passed in 0.04s =====
Post stage
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline

GitHub has been notified of this commit's build result

Finished: SUCCESS

Activate Windows
Go to Settings to activate Windows.

Build	Time Since ↑	Status	
 Python Deployment3 » main #6	2 min 10 sec	back to normal	
 Python Deployment3 » main #5	19 min	broken since this build	
 Python Deployment3 » main #4	1 hr 9 min	stable	
 Python Deployment3 » main #3	1 hr 9 min	stable	
 Python Deployment3 » main #2	5 hr 42 min	stable	
 Python Deployment3 » main #1	2 days 23 hr	stable	

Step 6. Finally, initiate a pull request to the [kura_labs_org/DEPLOY_3_TESTING](#) repository with all the documentations and screenshots, `test_calc.py` with the added test, and `add2vals.py` with the added feature or component. (Including this document).