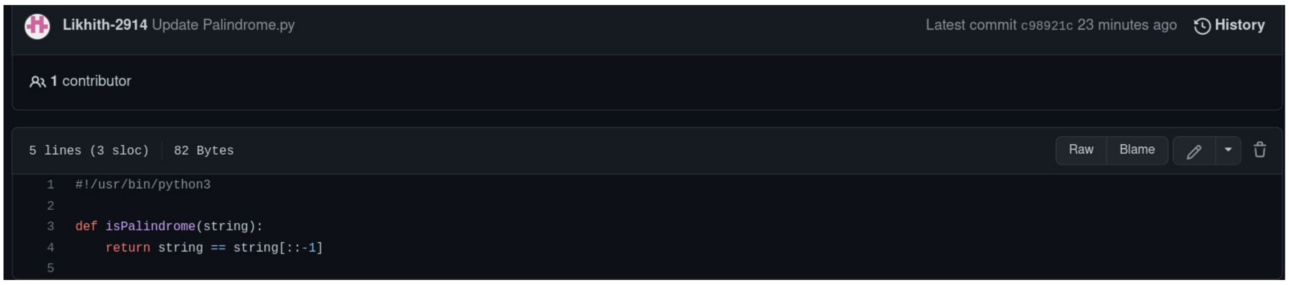
Jenkins Project – Graded Assignment 6

N V Sai Likhith
IMT2020118

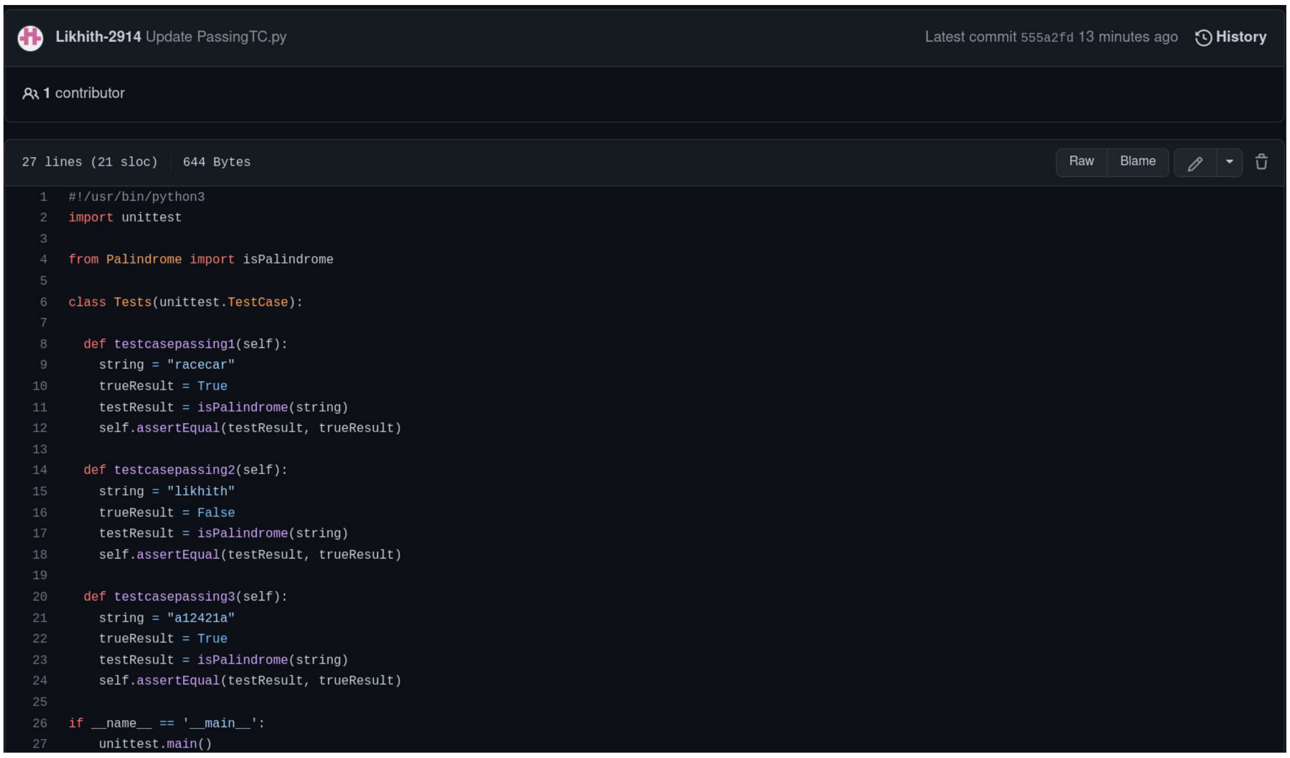
1. `isPalindrome(string):` returns true if the string is a palindrome.



Likhith-2914 Update Palindrome.py
Latest commit c98921c 23 minutes ago
History
1 contributor
5 lines (3 sloc) 82 Bytes
Raw
Blame

```python
#!/usr/bin/python3

def isPalindrome(string):
    return string == string[::-1]

```

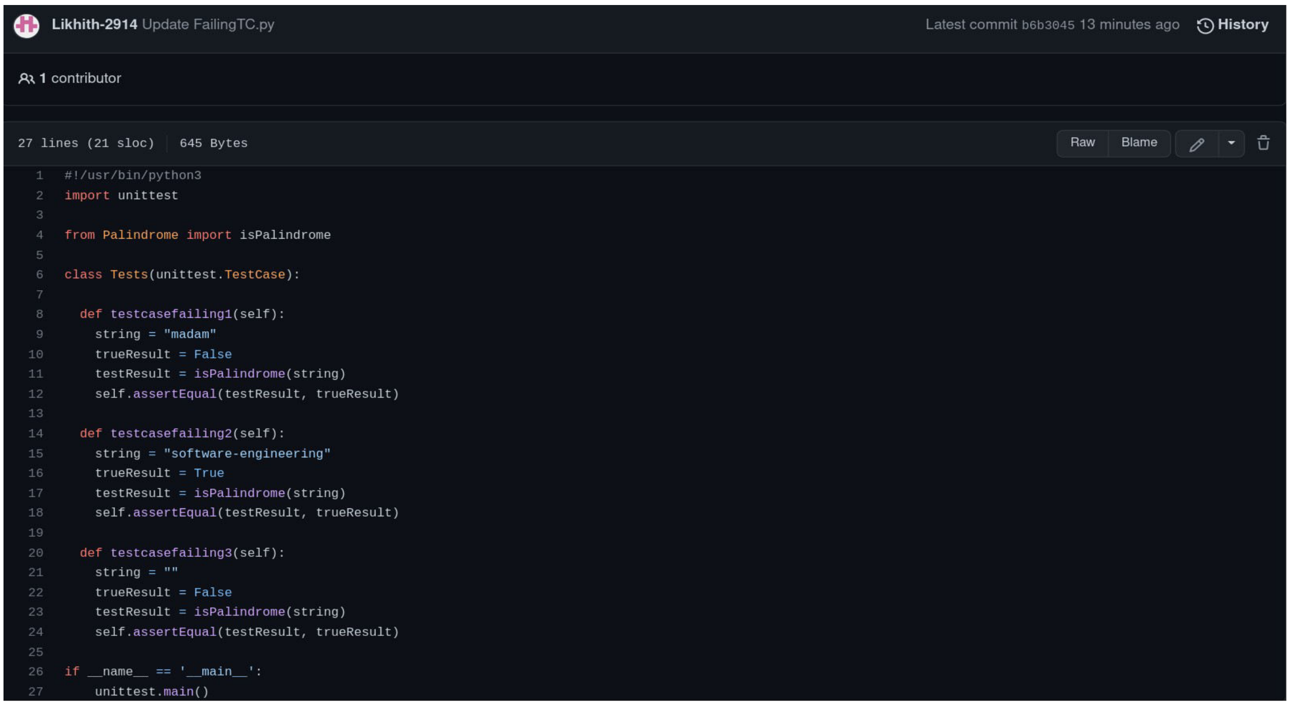
2. `testcasespassing<x>():` checks the testcases that are true (Eg: racecar is a palindrome)



Likhith-2914 Update PassingTC.py
Latest commit 555a2fd 13 minutes ago
History
1 contributor
27 lines (21 sloc) 644 Bytes
Raw
Blame

```python
#!/usr/bin/python3
import unittest

from Palindrome import isPalindrome

class Tests(unittest.TestCase):

  def testcasepassing1(self):
    string = "racecar"
    trueResult = True
    testResult = isPalindrome(string)
    self.assertEqual(testResult, trueResult)

  def testcasepassing2(self):
    string = "likhith"
    trueResult = False
    testResult = isPalindrome(string)
    self.assertEqual(testResult, trueResult)

  def testcasepassing3(self):
    string = "a12421a"
    trueResult = True
    testResult = isPalindrome(string)
    self.assertEqual(testResult, trueResult)

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

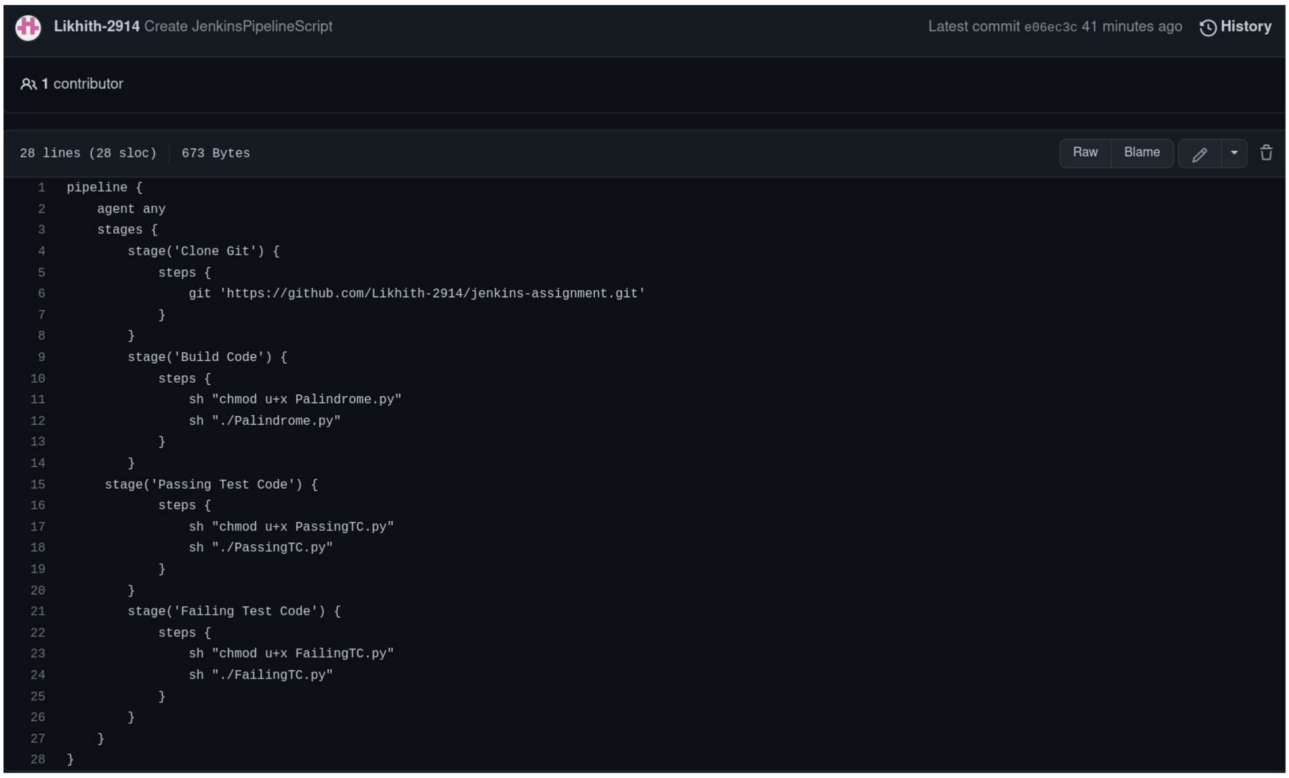
3. `testcasesfailing<x>():` checks the testcases that are false (Eg: madam is a palindrome)


Likhith-2914 Update FailingTC.py
Latest commit b6b3045 13 minutes ago
History
1 contributor
27 lines (21 sloc) | 645 Bytes
Raw
Blame

```python
#!/usr/bin/python3
import unittest

from Palindrome import isPalindrome

class Tests(unittest.TestCase):

  def testcasefailing1(self):
    string = "madam"
    trueResult = False
    testResult = isPalindrome(string)
    self.assertEqual(testResult, trueResult)

  def testcasefailing2(self):
    string = "software-engineering"
    trueResult = True
    testResult = isPalindrome(string)
    self.assertEqual(testResult, trueResult)

  def testcasefailing3(self):
    string = ""
    trueResult = False
    testResult = isPalindrome(string)
    self.assertEqual(testResult, trueResult)

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

2. `Pipeline script:`

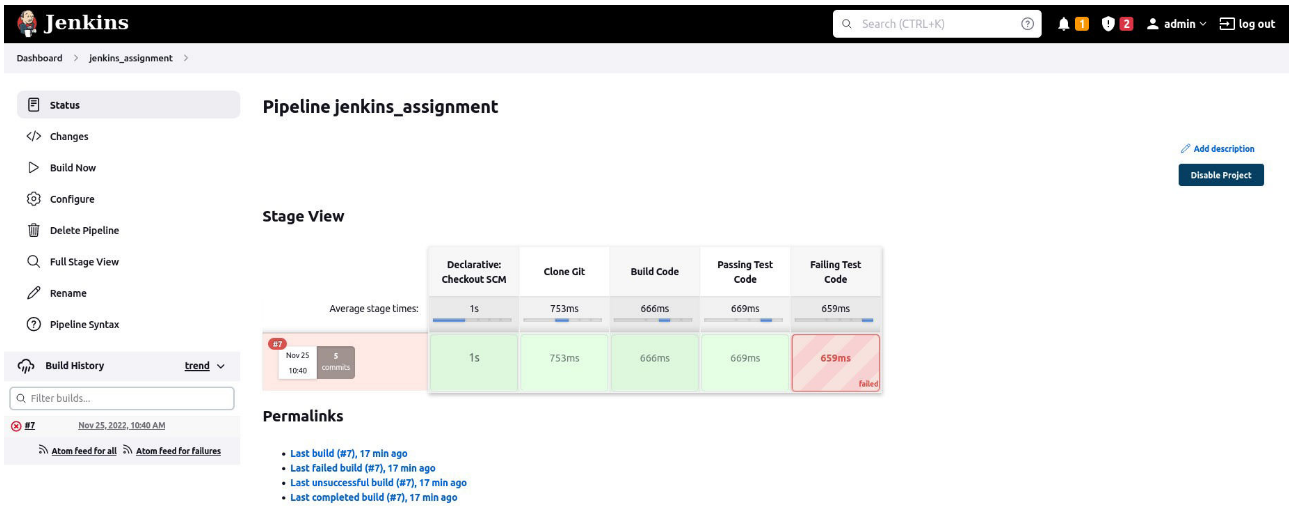
contains 4 stages where each stage deals with seperate actvity as in the script. Ideally the 4th stage must fail as it has the script of `failingTC.py` which has test cases that fail


Likhith-2914 Create JenkinsPipelineScript
Latest commit e06ec3c 41 minutes ago
History
1 contributor
28 lines (28 sloc) | 673 Bytes
Raw
Blame

```groovy
pipeline {
    agent any
    stages {
        stage('Clone Git') {
            steps {
                git 'https://github.com/Likhith-2914/jenkins-assignment.git'
            }
        }
        stage('Build Code') {
            steps {
                sh "chmod u+x Palindrome.py"
                sh "./Palindrome.py"
            }
        }
     stage('Passing Test Code') {
            steps {
                sh "chmod u+x PassingTC.py"
                sh "./PassingTC.py"
            }
        }
        stage('Failing Test Code') {
            steps {
                sh "chmod u+x FailingTC.py"
                sh "./FailingTC.py"
            }
        }
    }
}
```

Build Result:

The below picture shows the result obtained after building the pipeline script. We can observe that except failing Test code everything passed. It had failed because it has test cases that fail.



Pipeline jenkins_assignment

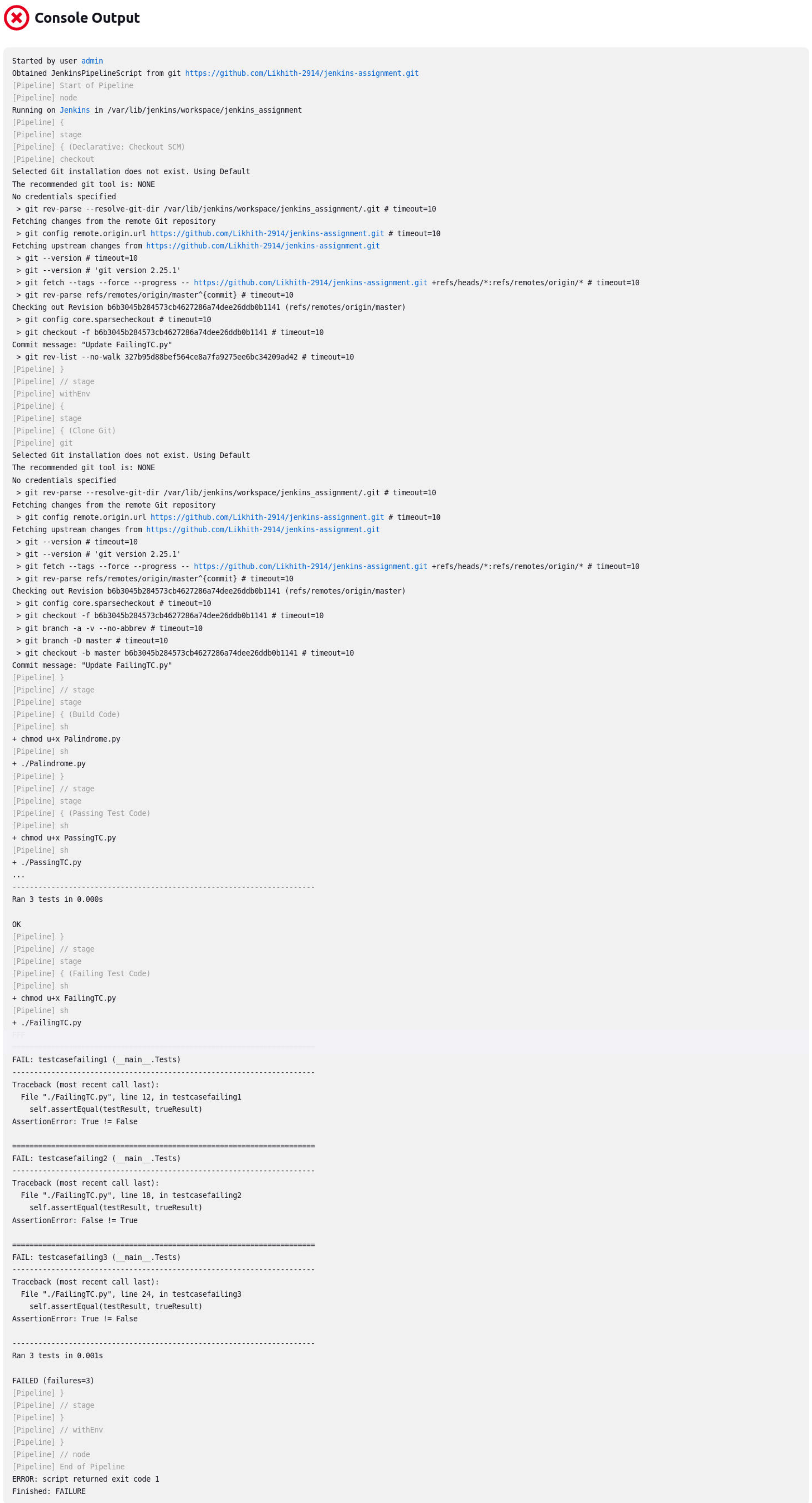
Stage View

	Declarative: Checkout SCM	Clone Git	Build Code	Passing Test Code	Failing Test Code
Average stage times:	1s	753ms	666ms	669ms	659ms
#7 Nov 25 10:40 5 commits	1s	753ms	666ms	669ms	659ms failed

Permalinks

- Last build (#7), 17 min ago
- Last failed build (#7), 17 min ago
- Last unsuccessful build (#7), 17 min ago
- Last completed build (#7), 17 min ago

In the next page, the entire console output is given



Console Output

```
Started by user admin
Obtained JenkinsPipelineScript from git https://github.com/Likhith-2914/jenkins-assignment.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/jenkins_assignment
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/jenkins_assignment/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Likhith-2914/jenkins-assignment.git # timeout=10
Fetching upstream changes from https://github.com/Likhith-2914/jenkins-assignment.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- https://github.com/Likhith-2914/jenkins-assignment.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision b6b3045b284573cb4627286a74dee26ddb0b1141 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f b6b3045b284573cb4627286a74dee26ddb0b1141 # timeout=10
Commit message: "Update FailingTC.py"
 > git rev-list --no-walk 327b95d88bef564ce8a7fa9275ee6bc34209ad42 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone Git)
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/jenkins_assignment/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Likhith-2914/jenkins-assignment.git # timeout=10
Fetching upstream changes from https://github.com/Likhith-2914/jenkins-assignment.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- https://github.com/Likhith-2914/jenkins-assignment.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision b6b3045b284573cb4627286a74dee26ddb0b1141 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f b6b3045b284573cb4627286a74dee26ddb0b1141 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D master # timeout=10
 > git checkout -b master b6b3045b284573cb4627286a74dee26ddb0b1141 # timeout=10
Commit message: "Update FailingTC.py"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Code)
[Pipeline] sh
+ chmod u+x Palindrome.py
[Pipeline] sh
+ ./Palindrome.py
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Passing Test Code)
[Pipeline] sh
+ chmod u+x PassingTC.py
[Pipeline] sh
+ ./PassingTC.py
...
----------------------------------------------------------------------
Ran 3 tests in 0.000s

OK
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Failing Test Code)
[Pipeline] sh
+ chmod u+x FailingTC.py
[Pipeline] sh
+ ./FailingTC.py
FFF
======================================================================
FAIL: testcasefailing1 (__main__.Tests)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "./FailingTC.py", line 12, in testcasefailing1
    self.assertEqual(testResult, trueResult)
AssertionError: True != False

======================================================================
FAIL: testcasefailing2 (__main__.Tests)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "./FailingTC.py", line 18, in testcasefailing2
    self.assertEqual(testResult, trueResult)
AssertionError: False != True

======================================================================
FAIL: testcasefailing3 (__main__.Tests)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "./FailingTC.py", line 24, in testcasefailing3
    self.assertEqual(testResult, trueResult)
AssertionError: True != False

----------------------------------------------------------------------
Ran 3 tests in 0.001s

FAILED (failures=3)
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```