

Unit Testing with Jenkins - pytest or JUnit

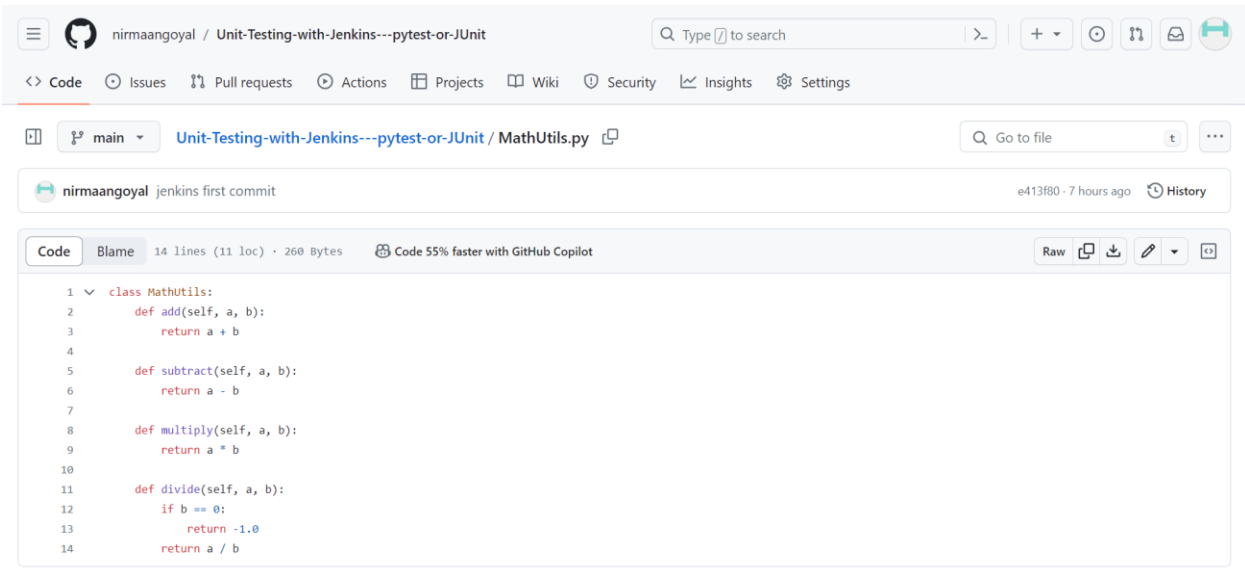
By:

Nirmaan Goyal

February, 4th 2024

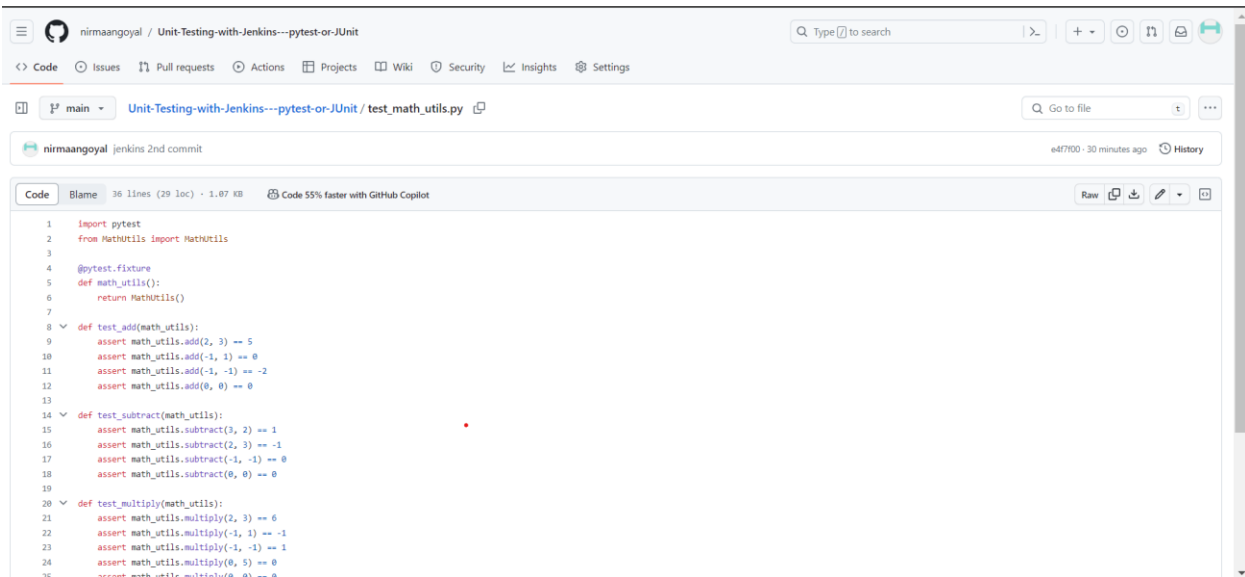
Link to my git repository: <https://github.com/nirmaangoyal/Unit-Testing-with-Jenkins---pytest-or-JUnit>

Mathutils.py



```
1 class MathUtils:
2     def add(self, a, b):
3         return a + b
4
5     def subtract(self, a, b):
6         return a - b
7
8     def multiply(self, a, b):
9         return a * b
10
11    def divide(self, a, b):
12        if b == 0:
13            return -1.0
14        return a / b
```

Test_math_utils.py



```
1 import pytest
2 from MathUtils import MathUtils
3
4 @pytest.fixture
5 def math_utils():
6     return MathUtils()
7
8 def test_add(math_utils):
9     assert math_utils.add(2, 3) == 5
10    assert math_utils.add(-1, 1) == 0
11    assert math_utils.add(-1, -1) == -2
12    assert math_utils.add(0, 0) == 0
13
14 def test_subtract(math_utils):
15     assert math_utils.subtract(3, 2) == 1
16     assert math_utils.subtract(2, 3) == -1
17     assert math_utils.subtract(-1, -1) == 0
18     assert math_utils.subtract(0, 0) == 0
19
20 def test_multiply(math_utils):
21     assert math_utils.multiply(2, 3) == 6
22     assert math_utils.multiply(-1, 1) == -1
23     assert math_utils.multiply(-1, -1) == 1
24     assert math_utils.multiply(0, 5) == 0
25     assert math_utils.multiply(0, 0) == 0
```

My Jenkins pipeline configuration for checkout, build, test and posting the result in a user-friendly manner.

****post stage will always happen irrespective of success or failure.**

```
pipeline {
    agent any

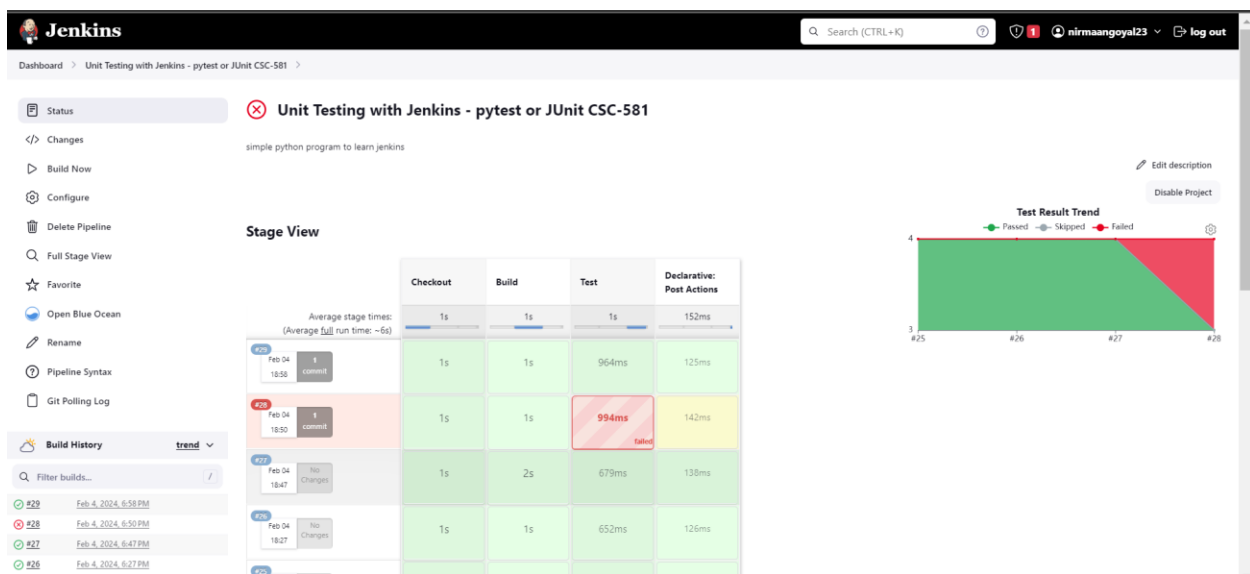
    stages {
        stage('Checkout') {
            steps {
                checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs:
[[url: 'https://github.com/nirmaangoyal/Unit-Testing-with-Jenkins---pytest-or-JUnit.git']])
            }
        }
        stage('Build') {
            steps {
                git branch: 'main', url: 'https://github.com/nirmaangoyal/Unit-Testing-with-Jenkins---
pytest-or-JUnit.git'
                bat 'python3 MathUtils.py'
            }
        }
        stage('Test') {
            steps {
                bat 'python3 -m pytest --junitxml=results.xml'
            }
        }
    }
    post {
        always {
```

```

junit '**/results.xml'
}
}
}

```

Screenshot of my build history with stage view.



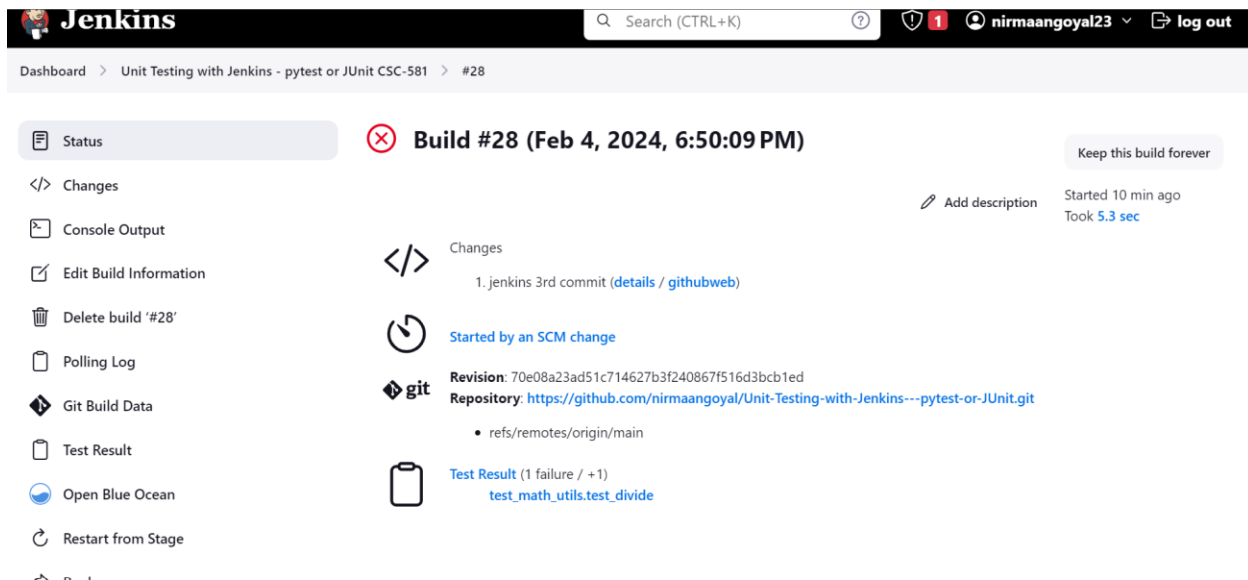
Results.xml file for more detailed view

The screenshot shows the Jenkins interface for a specific build, 'Build #29 (Feb 4, 2024, 6:58:09 PM)'. The build status is 'Success' (green checkmark). The left sidebar contains a list of actions: Status, Changes, Console Output, Edit Build Information, Delete build '#29', Polling Log, Git Build Data, Test Result, Open Blue Ocean, and Restart from Stage. The main content area displays build details:

- Changes:** 1. jenkins 4rd commit (details / githubweb)
- Started by an SCM change**
- git Revision:** 26815abadb693aab06c519d7e6cede86cde7ad20
- Repository:** <https://github.com/nirmaangoyal/Unit-Testing-with-Jenkins---pytest-or-JUnit.git>
- Refs/remotes/origin/main**
- Test Result:** (no failures)

Additional information includes 'Keep this build forever', 'Started 1 min 37 sec ago', and 'Took 4.3 sec'.

Test result with a failed case.



The screenshot shows the Jenkins interface for Build #28. The top navigation bar includes the Jenkins logo, a search bar, and user information. The main content area is divided into a left sidebar with navigation links (Status, Changes, Console Output, etc.) and a main panel. The main panel displays the build status as 'Failed' (indicated by a red 'X' icon) and provides details about the build, including the revision, repository, and test results. The test results section shows a failure in the 'test_math_utils.test_divide' test.

Jenkins Search (CTRL+K) nirmaangoyal23 log out

Dashboard > Unit Testing with Jenkins - pytest or JUnit CSC-581 > #28

Build #28 (Feb 4, 2024, 6:50:09 PM) Keep this build forever

Changes

1. jenkins 3rd commit (details / githubweb)

Started by an SCM change

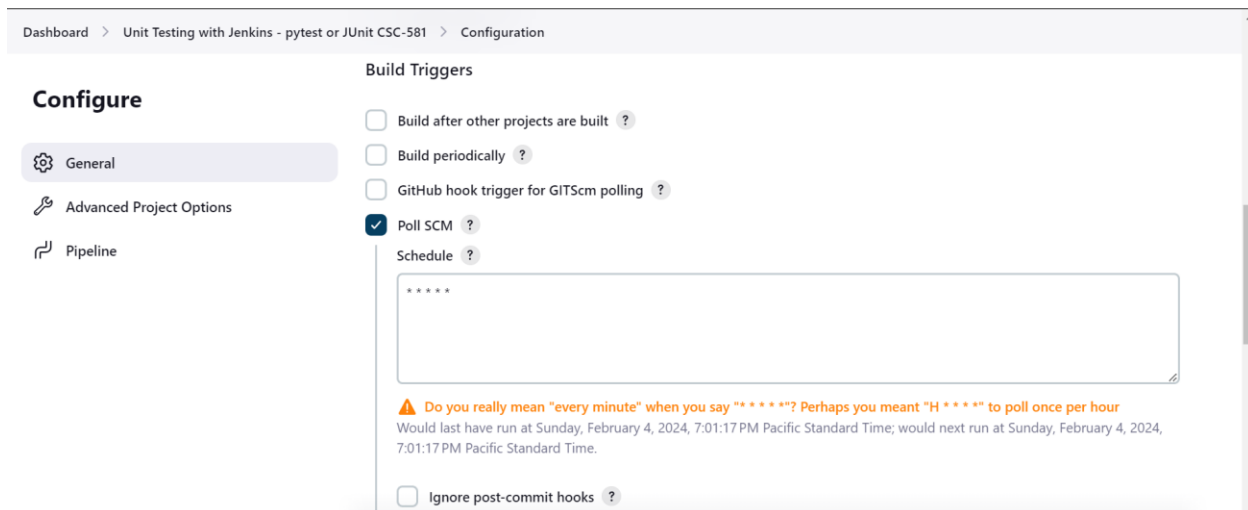
Revision: 70e08a23ad51c714627b3f240867f516d3bcb1ed
Repository: <https://github.com/nirmaangoyal/Unit-Testing-with-Jenkins---pytest-or-JUnit.git>

refs/remotes/origin/main

Test Result (1 failure / +1)
[test_math_utils.test_divide](#)

Trigger the Jenkins job to run automatically whenever changes are pushed to git.

**I am polling every minute to check if any changes have occurred in my git repository.



The screenshot shows the Jenkins Configuration page for the 'Unit Testing with Jenkins - pytest or JUnit CSC-581' job. The 'Build Triggers' section is expanded, showing options for triggering the build. The 'Poll SCM' option is selected, and the 'Schedule' field is set to '* * * * *'. A warning message is displayed below the schedule field, advising the user to use a more specific schedule if they mean 'every minute'.

Dashboard > Unit Testing with Jenkins - pytest or JUnit CSC-581 > Configuration

Configure

General

Advanced Project Options

Pipeline

Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☒ Poll SCM ?

Schedule ?

* * * * *

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour
Would last have run at Sunday, February 4, 2024, 7:01:17 PM Pacific Standard Time; would next run at Sunday, February 4, 2024, 7:01:17 PM Pacific Standard Time.

☐ Ignore post-commit hooks ?