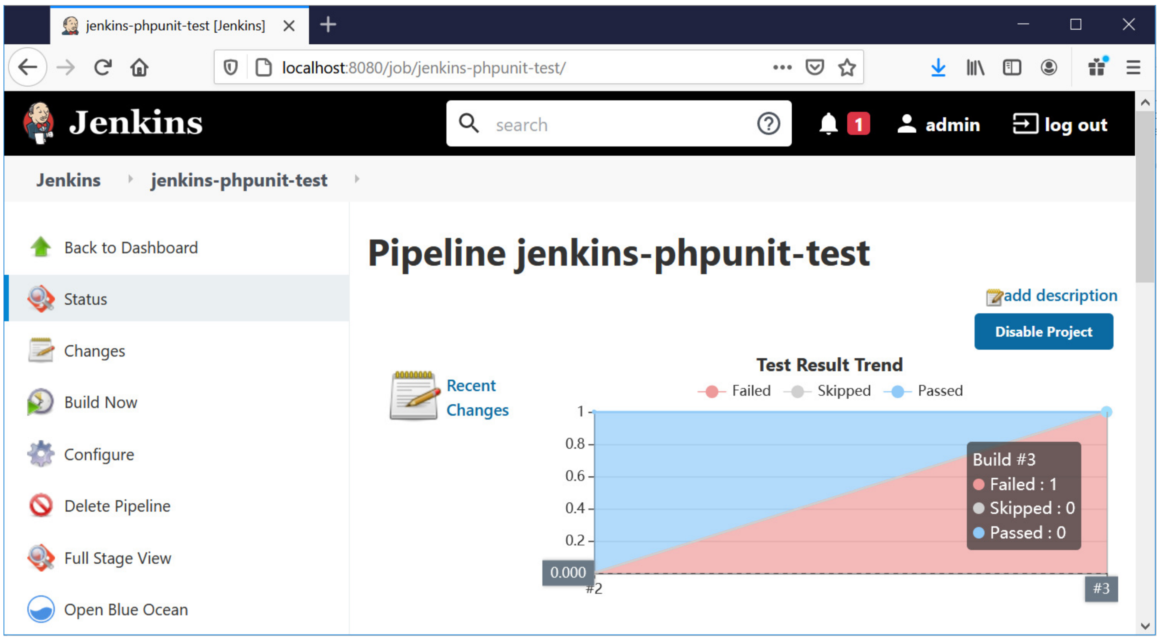
Integrating Jenkins with Automated Unit Testing

Overview



In Lecture 7, you are introduced to automated testing which is the current industry trend as promoted by Agile and DevOps/DevSecOps CI/CD process. In this Lab 7a, you are going to learn how to integrate automated unit testing into Jenkins Pipeline so that you are able to start incorporating it into your team project.

Outcomes

Upon completion of this lab, you should be able to:

- Write test cases for automated unit testing using xUnit (JUnit, PHPUnit, PyUnit, Mocha, etc) testing framework
- Create Jenkins pipeline to perform automated unit testing on a simple PHP application
- Start incorporating Jenkins Pipeline with automated testing into your team project

1: Reference

This lab is based on the video "Running PHPUnit tests after each commit (Get started with Jenkins part 5" by Simply Explained posted at https://www.youtube.com/watch?v=68cDNUz7uro. However, it was presented using different Jenkins environment which you may find it difficult to follow. Hence, this lab is re-written with enhancement using latest Jenkins Pipeline to help you understand how to integrate automated unit testing into your team project.

2: Pre-requisite

Git

Download the provided `jenkins-phpunit-test.zip` file which contains a simple PHP app, a corresponding unit testing file and a Jenkinsfile. Unzip and initialize it as a Git repository using the command:

```
git init
```

Next, commit with the commands:

```
git add .
```

then

```
git commit -m "Add initial files"
```

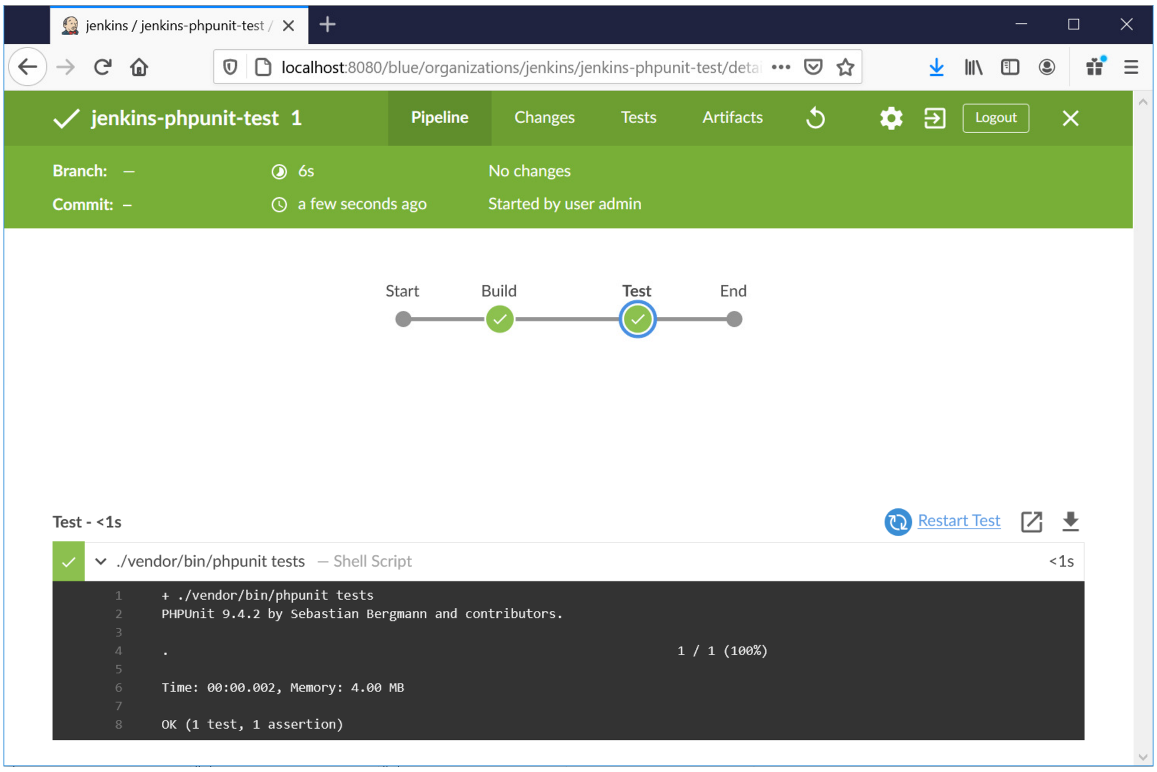
3: Integrating Automated Unit Testing into Jenkins Pipeline

Create new Jenkins Pipeline

Create a new Pipeline for the simple PHP app in Section 2. (Hopefully you are now familiar with creating a Jenkins pipeline after trying Lab-X05.)

Run Jenkins Pipeline with Unit Testing

a) Run the Jenkins pipeline which should pass the unit test indicated by OK (1 test, 1 assertion) as shown:



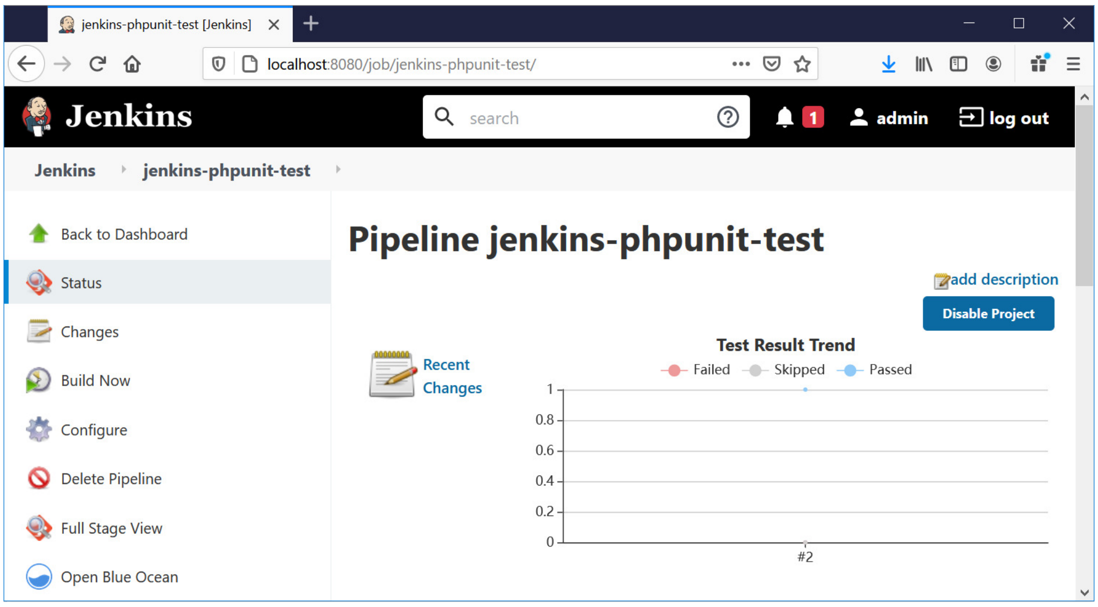
```
+ ./vendor/bin/phpunit tests
PHPUnit 9.4.2 by Sebastian Bergmann and contributors.

.                                                                   1 / 1 (100%)

Time: 00:00.002, Memory: 4.00 MB

OK (1 test, 1 assertion)
```

Enhancing with Test Reporting

b) Update the Jenkinsfile to include test reporting so that your final Jenkinsfile will be similar as follows:

```
pipeline {
    agent {
        docker {
            image 'composer:latest'
        }
    }
    stages {
        stage('Build') {
            steps {
                sh 'composer install'
            }
        }
        stage('Test') {
            steps {
                sh './vendor/bin/phpunit --log-junit logs/unitreport.xml -c tests/phpunit.xml tests'
            }
        }
    }
    post {
        always {
            junit testResults: 'logs/unitreport.xml'
        }
    }
}
```

c) Commit the update and re-run the Jenkins pipeline. Once completed, exit from Blue Ocean UI and return to Jenkins classic UI where you will now see the Test Result Trend with 1 test passed (indicated in blue dot) for Build #2 as shown:



d) Next, let's intentionally include a bug in the original program by changing the function turnWheel() in GumballMachine.php as follows:

From:

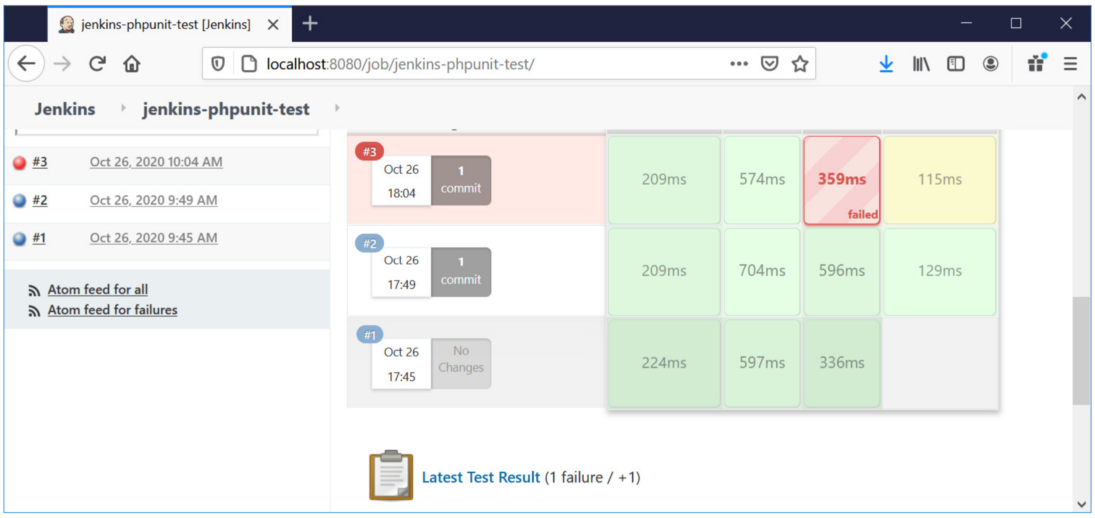
```
    public function turnWheel() {
        $this->setGumballs($this->getGumballs()-1);
    }
```

To:

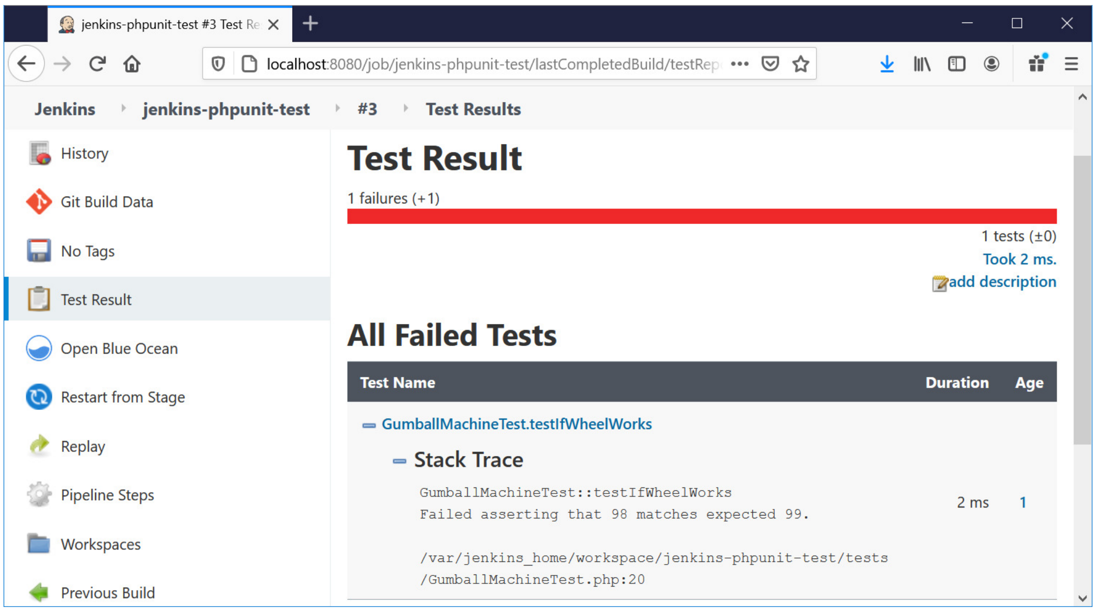
```
    public function turnWheel() {
        $this->setGumballs($this->getGumballs()-2);
    }
```

e) Commit the change and re-run the Jenkins pipeline which will result in failing the unit test for Build #3 as shown on the Test Result Trend graph on the first page of this hand-out.

f) Scroll down the page where you will see the Latest Test Result icon at the bottom as shown:



g) Click the Latest Test Result icon which will shown you information of the failed test.



Test Result

1 failures (+1)

1 tests (±0)
Took 2 ms.
add description

All Failed Tests

Test Name	Duration	Age
GumballMachineTest.testIfWheelWorks		
Stack Trace	2 ms	1

```
GumballMachineTest::testIfWheelWorks
Failed asserting that 98 matches expected 99.

/var/jenkins_home/workspace/jenkins-phpunit-test/tests
/GumballMachineTest.php:20
```

Hope that you have a better understanding of how to implement automated unit testing now. So try to start incorporating it into your team project. Enjoy!

END OF DOCUMENT