

Jenkins Pipeline Setup and Configuration and Build Results

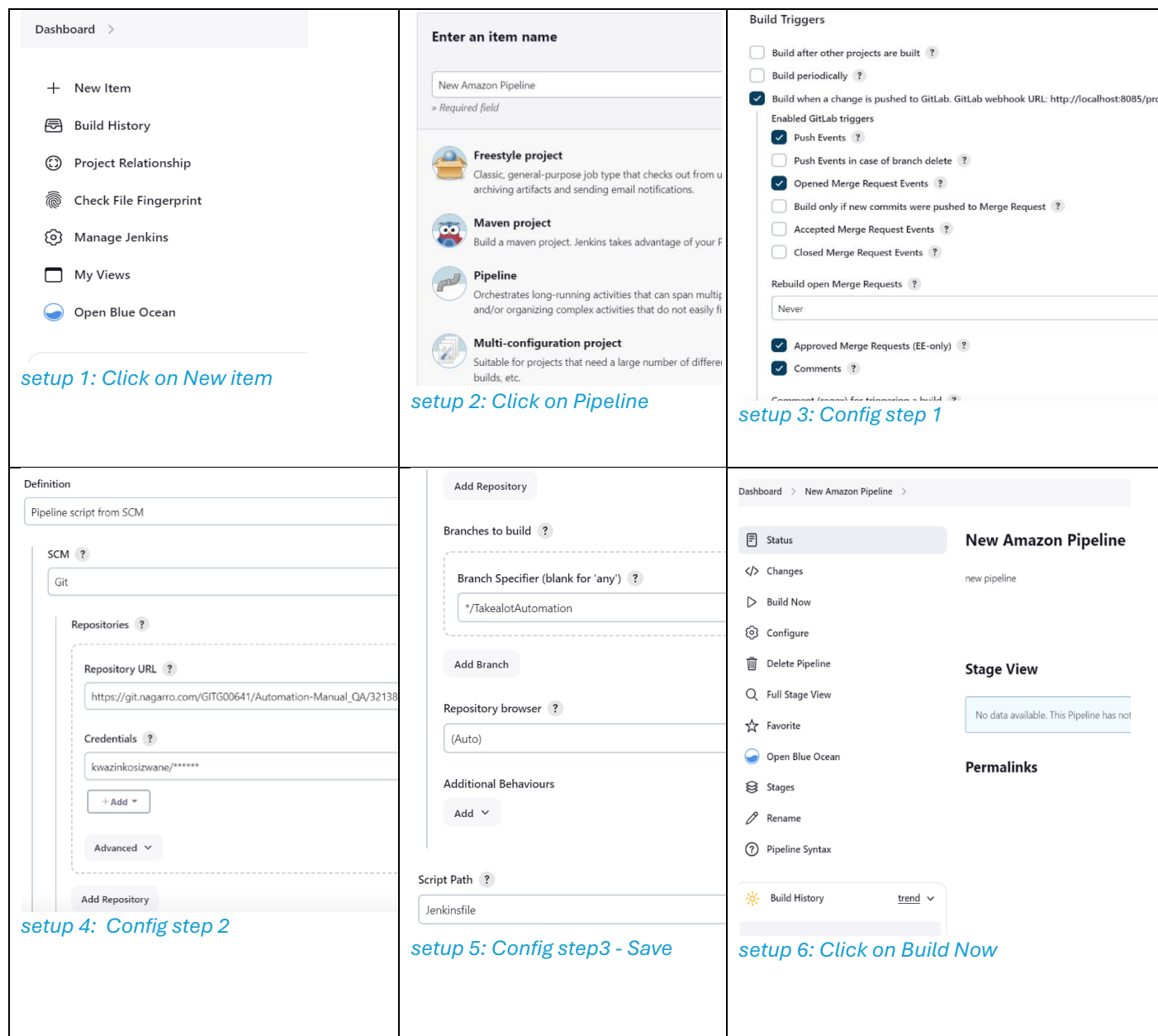


Figure 1: Pipeline Configuration Steps

The screenshots in figure 1 show the setup and configuration process of a pipeline.

Pipeline Results

Figure 2 shows screenshots of process of the testing framework.

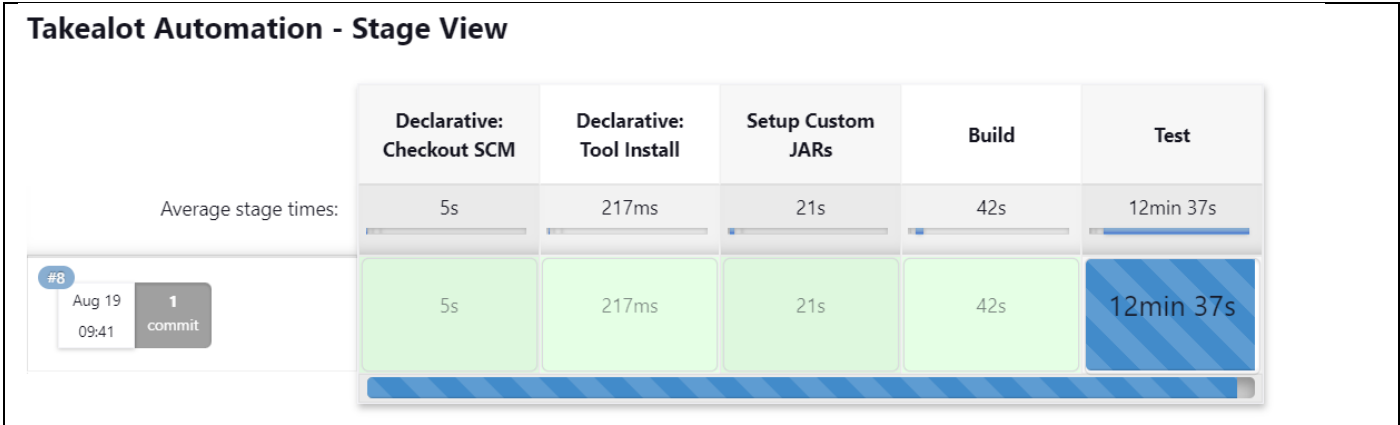


Figure 2: Build process running

There are 6 stages involved in this pipeline, as shown in figure 3.

- Declarative: Checkout SCM - In this stage Jenkins clones the project from gitlab to its own environment.
- Declarative: Tool Install – Jenkins installs all the required dependencies, specified in the pom file.
- Setup Custom Jars – In this stage Jenkins installs the required custom jars, specified in the Jenkins file. The following code shows how the installation of the custom jars was specified in the Jenkins file.

```
stage('Setup Custom JARs') {
  steps {
    script {
      bat "mvn install:install-file -Dfile=libs/wait-utils-0.0.1-SNAPSHOT.jar -DgroupId=com.kwazi -DartifactId=wait-utils -Dversion=0.0.1-SNAPSHOT -Dpackaging=jar -DgeneratePom=true"
      bat "mvn install:install-file -Dfile=libs/file-utils-0.0.1-SNAPSHOT.jar -DgroupId=com.kwazi -DartifactId=file-utils -Dversion=0.0.1-SNAPSHOT -Dpackaging=jar -DgeneratePom=true"
      bat "mvn install:install-file -Dfile=libs/logs-report-util-0.0.1-SNAPSHOT.jar -DgroupId=com.kwazi -DartifactId=logs-report-util -Dversion=0.0.1-SNAPSHOT -Dpackaging=jar -DgeneratePom=true"
    }
  }
}
```

- Build – This stage is responsible for the compilation of the testing framework.
- Test – In the testing stage Jenkins executes the “mvn test” command and runs the tests specified in the test.xml file.
- Clean up – cleans up

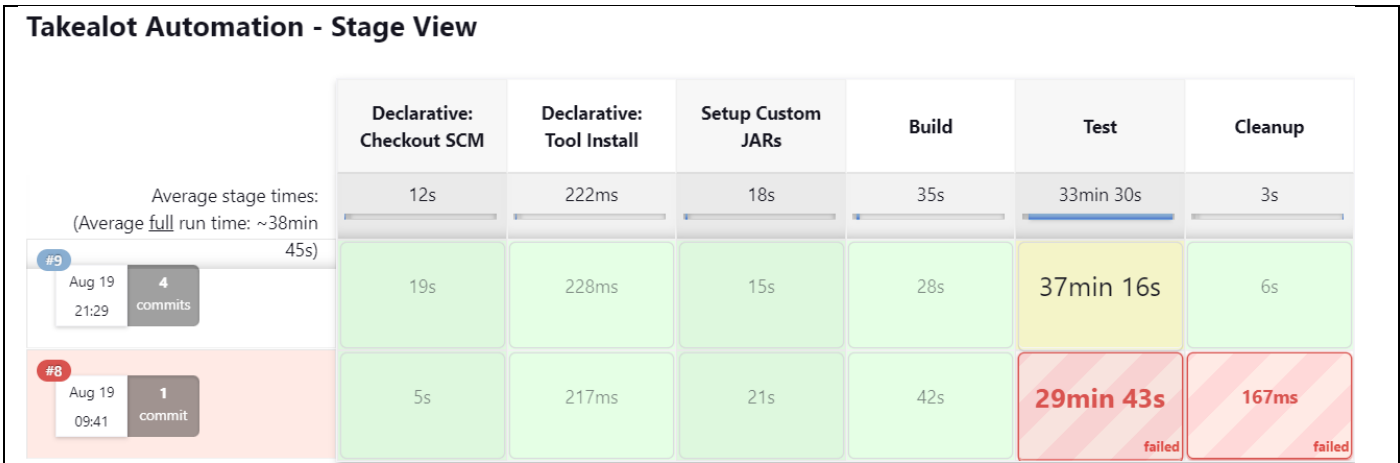


Figure 3: Complete pipeline build

As shown in figure 3, the testing stage for build 8 failed. The reason for this is that there are test cases that failed. The following table shows the logs from Jenkins which shows the test cases that failed. Build 9

shows the test stage is unstable, hence the yellow colour. Instead of terminating the pipeline when some test cases fail as in build 8 Jenkins is configured to mark the stage unstable and continue with other stages.

Table 1: Summary of results and test case failures

Results Summary	<div>[ERROR] Tests run: 44, Failures: 8, Errors: 0, Skipped: 0</div>
Errors	<div>0[ERROR] Failures:</div> <div>[ERROR] CartPageTest.verifyCartIsEmpty:63 The cart is not empty. expected [true] but found [false]</div> <div>[ERROR] CartPageTest.verifyItemDetailsInCart:113 The cart item should be present in the cart. expected object to not be null</div> <div>[ERROR] CartPageTest.verifyMoveItemToWishlist:162 The cart item should be present in the cart before moving to wishlist. expected object to not be null</div> <div>[ERROR] CartPageTest.verifyRemovalOfItemFromCart:147 The cart item should be present in the cart before removal. expected object to not be null</div> <div>[ERROR] ProductsPageTest.verifyProductDetails:179 Product price does not match expected [690.00] but found [850.00]</div> <div>[ERROR] ProductsPageTest.verifyProductDetails:179 Product price does not match expected [27595.00] but found [27082.00]</div> <div>[ERROR] RegistrationPageTest.verifyRegistration:90 » ElementClickIntercepted element click intercepted: Element <input name="first_name" type="text" id="register_customer_first_name" autocomplete="on" value=""> is not clickable at point (512, 96). Other element would receive the click: <div class="ab-page-blocker" style="background-color: rgba(51, 51, 51, 0.75);"></div></div>

Figure 4 shows the stage view of the pipeline build. If all 44 test cases were to pass, all the stages would have a green check mark.

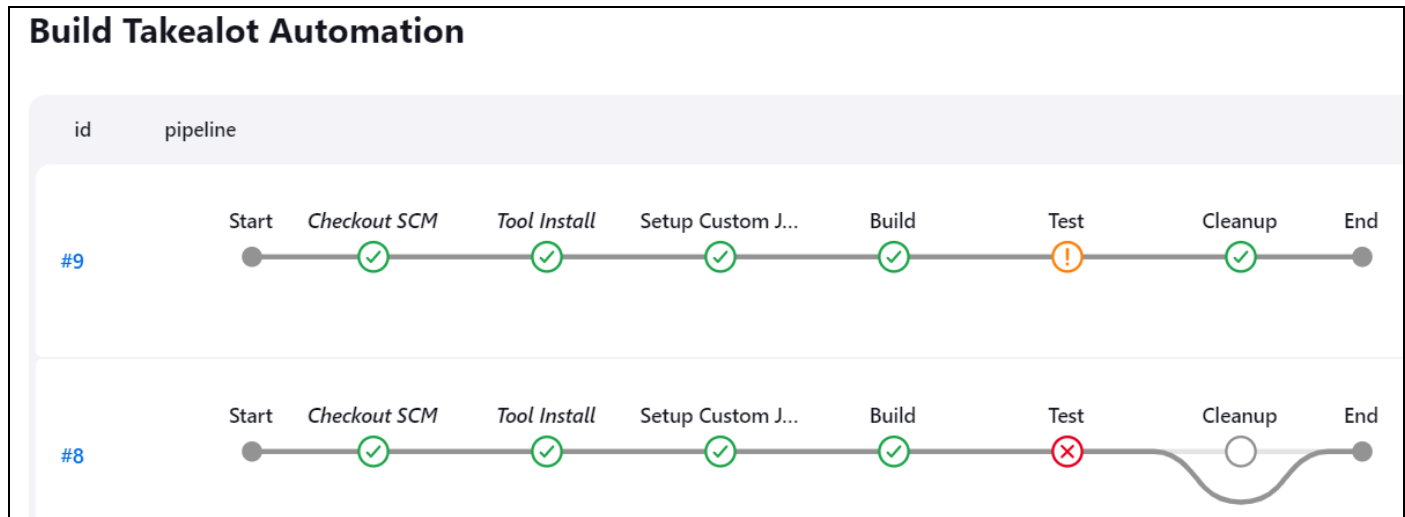


Figure 4: Pipeline stages

A complete log file for the Takealot framework generated by Jenkins is attached in the same directory as this document.