**Setting up a pipeline on Jenkins for continuous integration testing of a simple front-end web-based application requires the following steps:**

To set up a pipeline on Jenkins for continuous integration testing using Selenium WebDriver, Java, TestNG, and Maven, the following steps: can be taken:

* Install the Jenkins plugin for Selenium and configure the necessary dependencies in the Jenkins file.
* Set up a new Jenkins job and configure it to pull code from the GitHub repository containing the sample application.
* In the build step, add a Maven goal to compile and test the code using the TestNG framework.
* Configure post-build actions to archive the test results and send notifications on build status.
* Set up the Jenkins job to trigger a build automatically on changes pushed to the GitHub repository, or set up a schedule for regular builds.
* Set up a Selenium Grid to run tests on different browsers and platforms in parallel.
* Set up the Jenkins job to integrate with Selenium Grid for running the tests
* Configure the job to trigger the build only when a pull request is made, or when a new branch is pushed to the repository.
* In summary, you would use Jenkins to pull code from a Github repository, use maven to build the code and test it using TestNG framework,
* integrate Selenium Grid and run the tests on different browsers, and trigger the builds automatically or on schedule.

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

public class CalculatorTest {

private WebDriver driver;

@Test

public void testAddition() {

driver = new ChromeDriver();

driver.get("https://www.online-calculator.com/");

WebElement num1 = driver.findElement(By.xpath("//input[@name='f']"));

num1.sendKeys("5");

WebElement num2 = driver.findElement(By.xpath("//input[@name='l']"));

num2.sendKeys("3");

WebElement addButton = driver.findElement(By.xpath("//input[@value='+']"));

addButton.click();

WebElement result = driver.findElement(By.xpath("//input[@name='res']"));

Assert.assertEquals(result.getAttribute("value"), "8");

driver.quit();

}

}

**By setting up a pipeline on Jenkins for continuous integration testing, the development team can ensure that the application is thoroughly tested and working properly before it is deployed to a production environment.**