Phase 4: APPIUM, AWS, SELENIUM DOCEKER GRID + JENKINS

BY: NEHA MITTAL

Learn Mobile Automation and Cloud Testing

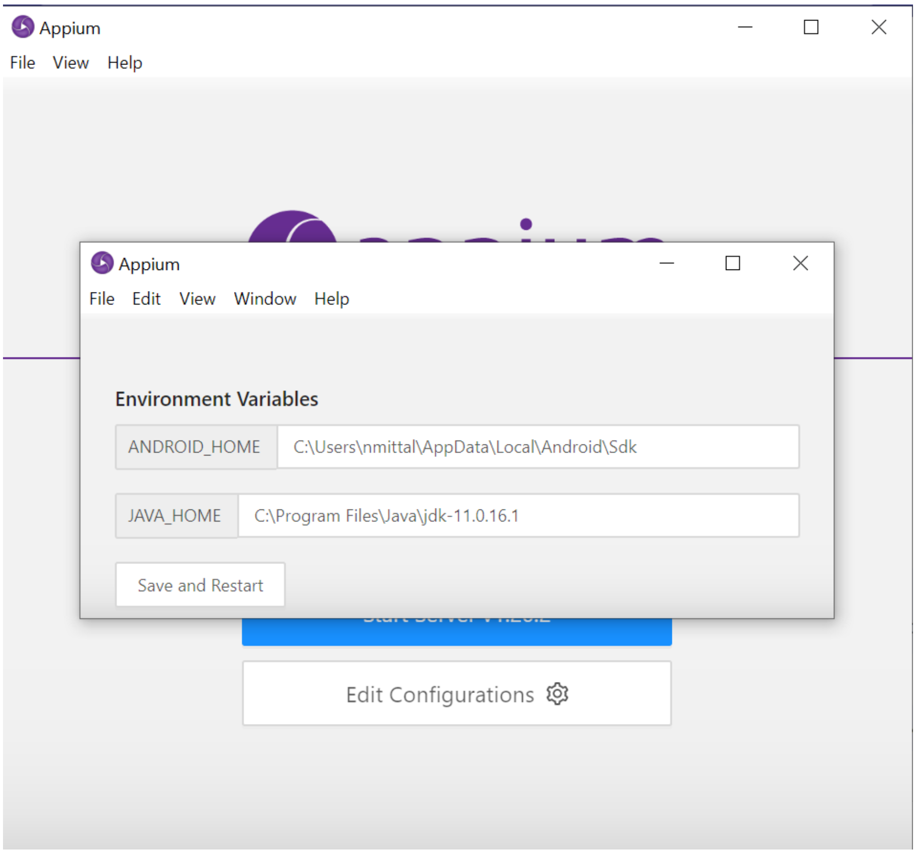
Table of Contents

Project Description:

A. Appium : Automate Aptoide App

Create a page object model project for “aptoide” mobile application automation where you have to install app then open app -> search amazon→ keep scrolling down till you find amazon fire tv app-> then click and install firetv app->verify that amazon fire tv app is installed-> then open fire tv from aptitode app-> switch to fire tv app-> click on more option on fire tv app-> then verify app version is correct → then again switch to aptitode app->click on stores-> click on login-> select agree terms-> select email method to login->put a dynamic address -> click send magic link-> then verify the content on check your mail screen→ then move to google chrome app-> open mailinator.com website->search for the email which you enter in aptitode app->open latest email and then verify the email is received and contains aptitode magic link(verify app login link is there and correct)

Configuration Settings



Environment Variables

ANDROID_HOME C:\Users\nmittal\AppData\Local\Android\Sdk

JAVA_HOME C:\Program Files\Java\jdk-11.0.16.1

Capabilities

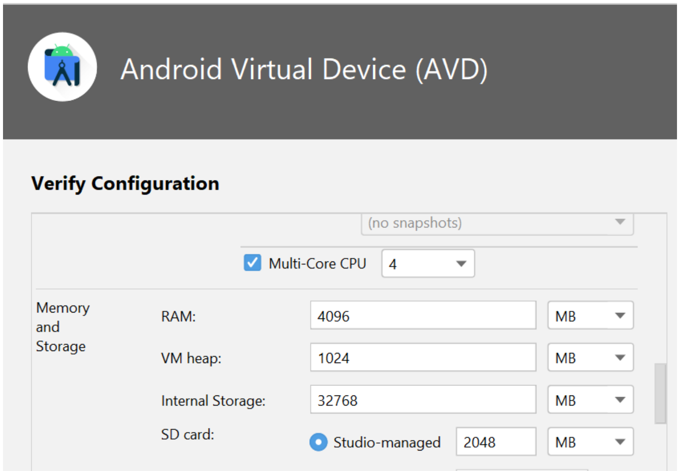
Virtual Device

```
{
  "platformName": "android",
  "appium:app": "C:\\Users\\nmittal\\Downloads\\apidemos.apk\\",
  "appium:deviceName": "emulator-5554",
  "appium:platformVersion": "10.0"
}
```

Steps to follow for Appium Project

Requirements:

1. Appium Installed
2. Android Studio
   a. Device: “Pixel 5 API 30”



Android Virtual Device (AVD)

Verify Configuration

(no snapshots)

☑ Multi-Core CPU 4

Memory and Storage

RAM: 4096 MB

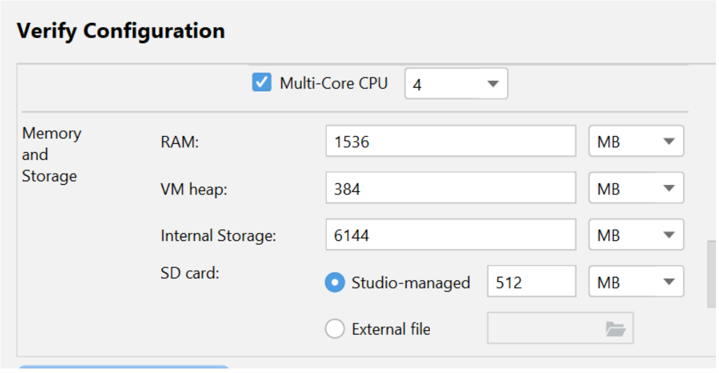
VM heap: 1024 MB

Internal Storage: 32768 MB

SD card: ● Studio-managed 2048 MB

Or

If more memory required



Verify Configuration

☑ Multi-Core CPU 4

Memory and Storage

RAM: 1536 MB

VM heap: 384 MB

Internal Storage: 6144 MB

SD card: ● Studio-managed 512 MB

○ External file

3. Inspector
4. Java Client

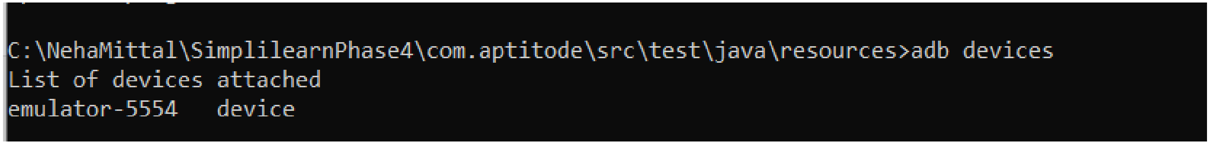
```
<!-- https://mvnrepository.com/artifact/io.appium/java-client -->
               <dependency>
                     <groupId>io.appium</groupId>
                     <artifactId>java-client</artifactId>
                     <version>8.3.0</version>
               </dependency>
```

5. Create Maven Project: **“com. aptoide”**
6. Add resources folder under **“src/test/java**”

7. Upload apk **"shoppingProject.apk"** file into it.
8. Open Visual Studio
   a. Create device "Pixel 3 XL API 30" (android 11.0) with required settings
   b. Play the device
9. Start Appium GUI server
10. Open Inspector
    a. Set the capabilities

```
{
  "platformName": "android",
  "appium:app": "C:\\NehaMittal\\SimplilearnPhase4\\com.aptitode\\src\\test\\java\\resources",
  "appium:deviceName": "Pixel 3 XL API 29",
  "appium:platformVersion": "10.0"
}
```

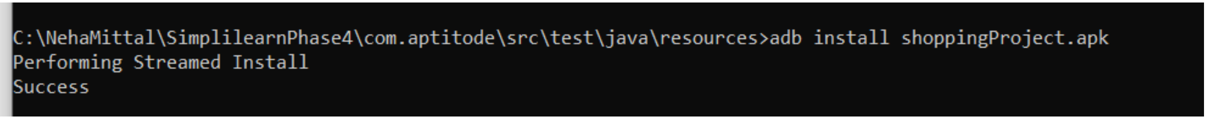
11. Go to apk location -> Open cmd-> type adb devices -> Enter



```
C:\NehaMittal\SimplilearnPhase4\com.aptitode\src\test\java\resources>adb devices
List of devices attached
emulator-5554   device
```

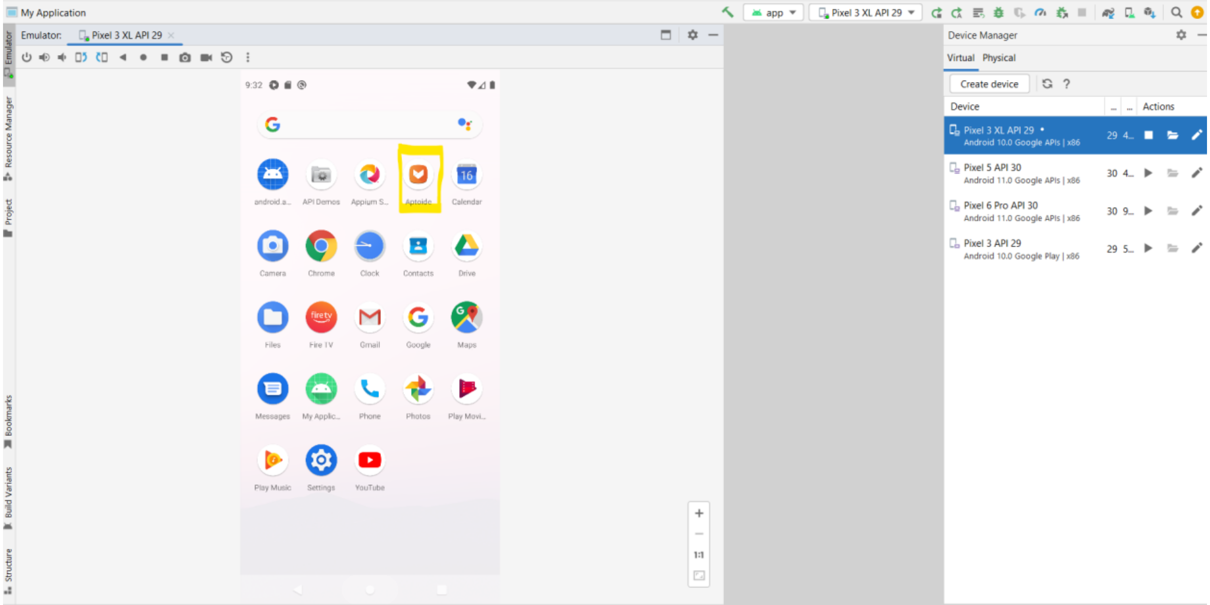
12. Install apk in device with command

adb install shoppingProject.apk



```
C:\NehaMittal\SimplilearnPhase4\com.aptitode\src\test\java\resources>adb install shoppingProject.apk
Performing Streamed Install
Success
```

13. Verify App should be installed on device



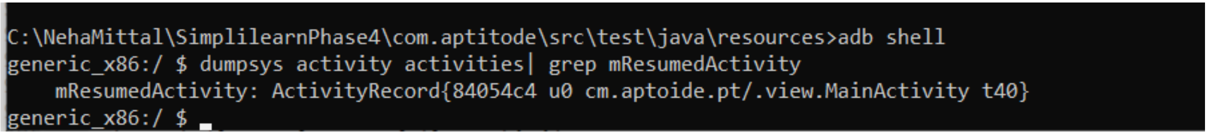
My Application
app
Pixel 3 XL API 29
Emulator: Pixel 3 XL API 29
9:32
android a... API Demos Appium S... Aptoide Calendar
Camera Chrome Clock Contacts Drive
Files Fire TV Gmail Google Maps
Messages My Applic... Phone Photos Play Movi...
Play Music Settings YouTube
Device Manager
Virtual Physical
Create device

Device	...	...	Actions
Pixel 3 XL API 29 • Android 10.0 Google APIs | x86	29	4...	
Pixel 5 API 30 Android 11.0 Google APIs | x86	30	4...	
Pixel 6 Pro API 30 Android 11.0 Google APIs | x86	30	9...	
Pixel 3 API 29 Android 10.0 Google Play | x86	29	5...	

14. Open Aptoide app
15. Check for package and activity name using command

    adb shell

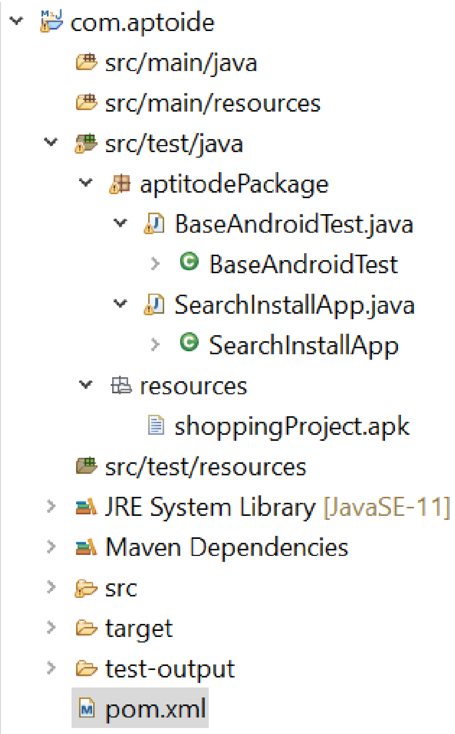
    dumpsys activity activities| grep mResumedActivity



```
C:\NehaMittal\SimplilearnPhase4\com.aptitode\src\test\java\resources>adb shell
generic_x86:/ $ dumpsys activity activities| grep mResumedActivity
    mResumedActivity: ActivityRecord{84054c4 u0 cm.aptoide.pt/.view.MainActivity t40}
generic_x86:/ $ 
```

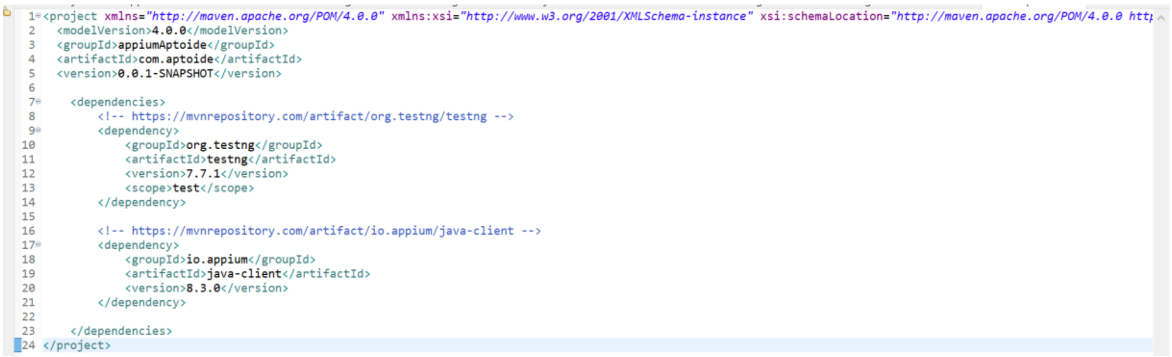
Activity Name: .view.MainActivity
Package Name: cm.aptoide.pt

Code:



```
com.aptoide
    src/main/java
    src/main/resources
    src/test/java
        aptitodePackage
            BaseAndroidTest.java
                BaseAndroidTest
            SearchInstallApp.java
                SearchInstallApp
        resources
            shoppingProject.apk
    src/test/resources
    JRE System Library [JavaSE-11]
    Maven Dependencies
    src
    target
    test-output
    pom.xml
```

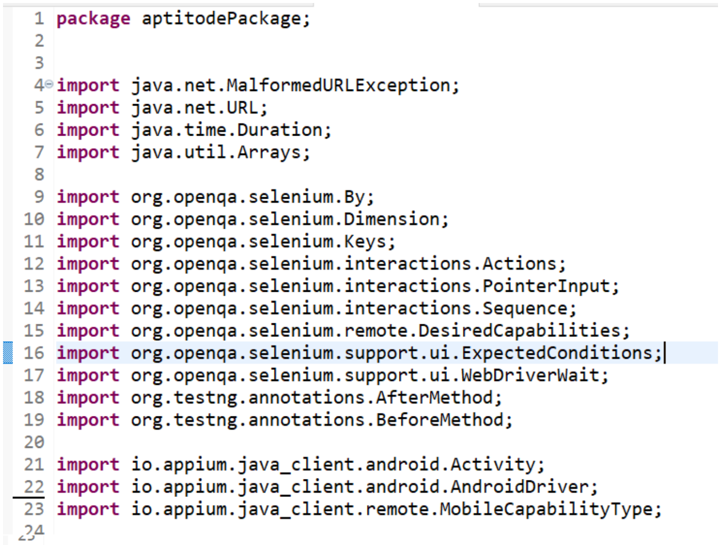
POM.xml



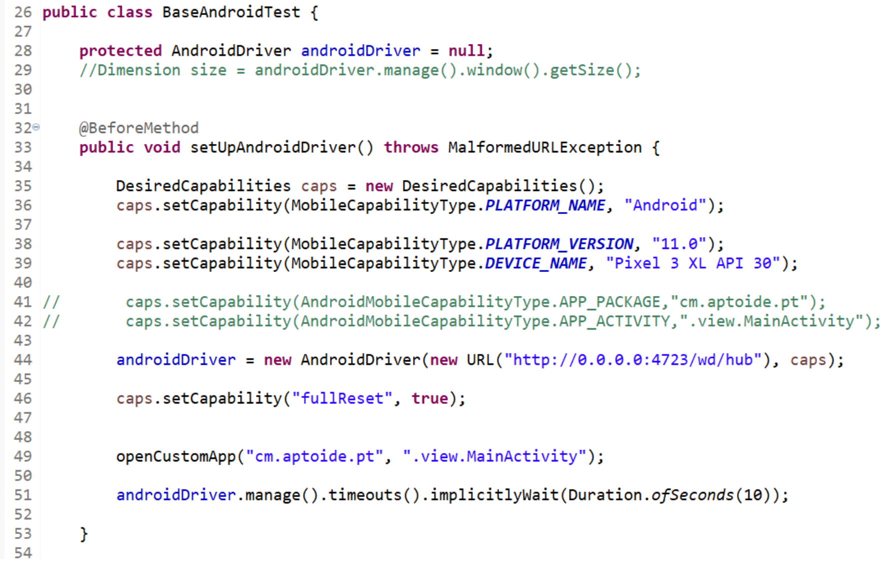
```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http
  <modelVersion>4.0.0</modelVersion>
  <groupId>appiumAptoide</groupId>
  <artifactId>com.aptoide</artifactId>
  <version>0.0.1-SNAPSHOT</version>

    <dependencies>
        <!-- https://mvnrepository.com/artifact/org.testng/testng -->
        <dependency>
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>7.7.1</version>
            <scope>test</scope>
        </dependency>

        <!-- https://mvnrepository.com/artifact/io.appium/java-client -->
        <dependency>
            <groupId>io.appium</groupId>
            <artifactId>java-client</artifactId>
            <version>8.3.0</version>
        </dependency>

    </dependencies>
</project>
```

src/test/java

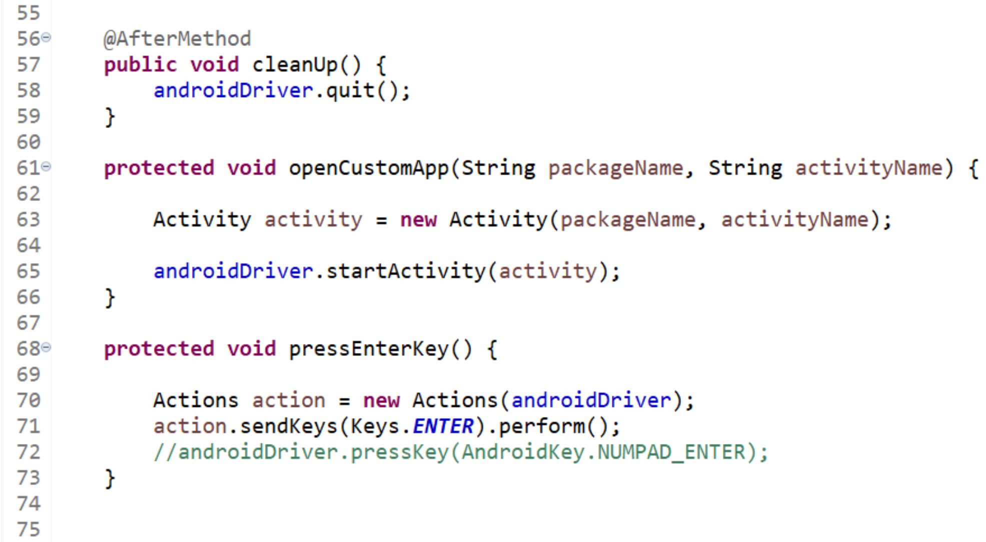
```java
public class BaseAndroidTest
```



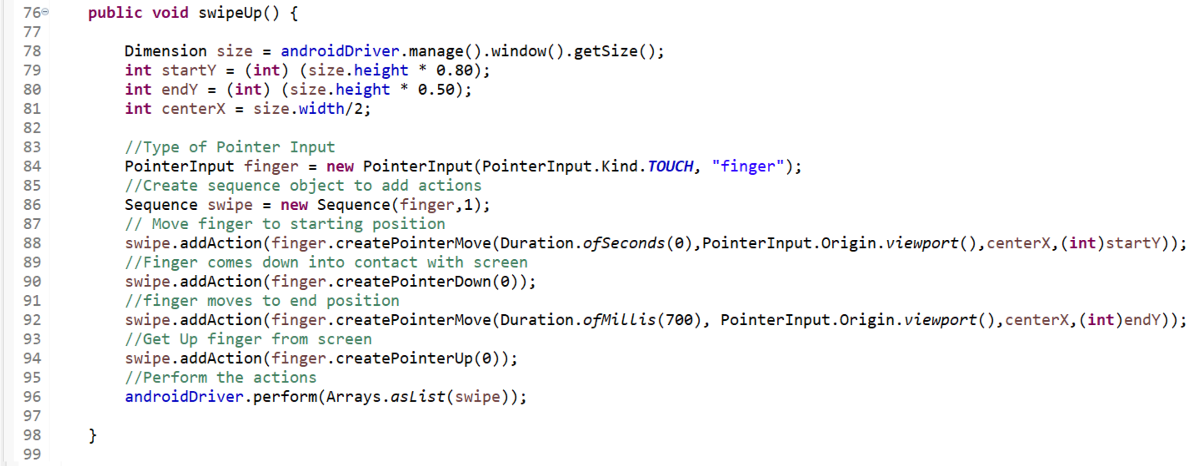
```java
package aptitodePackage;


import java.net.MalformedURLException;
import java.net.URL;
import java.time.Duration;
import java.util.Arrays;

import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Keys;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.interactions.PointerInput;
import org.openqa.selenium.interactions.Sequence;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;

import io.appium.java_client.android.Activity;
import io.appium.java_client.android.AndroidDriver;
import io.appium.java_client.remote.MobileCapabilityType;

```



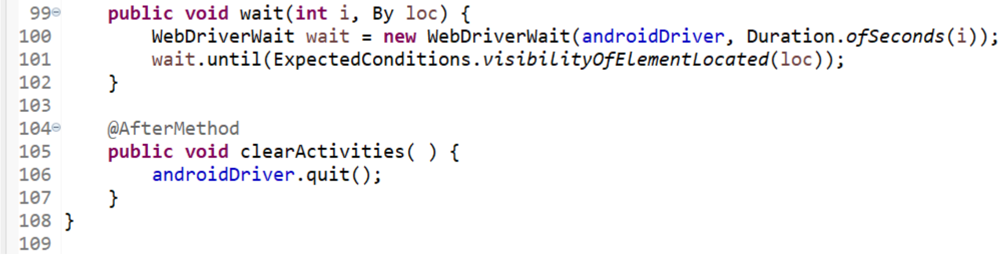
```java
public class BaseAndroidTest {

    protected AndroidDriver androidDriver = null;
    //Dimension size = androidDriver.manage().window().getSize();


    @BeforeMethod
    public void setUpAndroidDriver() throws MalformedURLException {

        DesiredCapabilities caps = new DesiredCapabilities();
        caps.setCapability(MobileCapabilityType.PLATFORM_NAME, "Android");

        caps.setCapability(MobileCapabilityType.PLATFORM_VERSION, "11.0");
        caps.setCapability(MobileCapabilityType.DEVICE_NAME, "Pixel 3 XL API 30");

//       caps.setCapability(AndroidMobileCapabilityType.APP_PACKAGE,"cm.aptoide.pt");
//       caps.setCapability(AndroidMobileCapabilityType.APP_ACTIVITY,".view.MainActivity");

        androidDriver = new AndroidDriver(new URL("http://0.0.0.0:4723/wd/hub"), caps);

        caps.setCapability("fullReset", true);


        openCustomApp("cm.aptoide.pt", ".view.MainActivity");

        androidDriver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

    }

```



```
    @AfterMethod
    public void cleanUp() {
        androidDriver.quit();
    }

    protected void openCustomApp(String packageName, String activityName) {

        Activity activity = new Activity(packageName, activityName);

        androidDriver.startActivity(activity);
    }

    protected void pressEnterKey() {

        Actions action = new Actions(androidDriver);
        action.sendKeys(Keys.ENTER).perform();
        //androidDriver.pressKey(AndroidKey.NUMPAD_ENTER);
    }

```

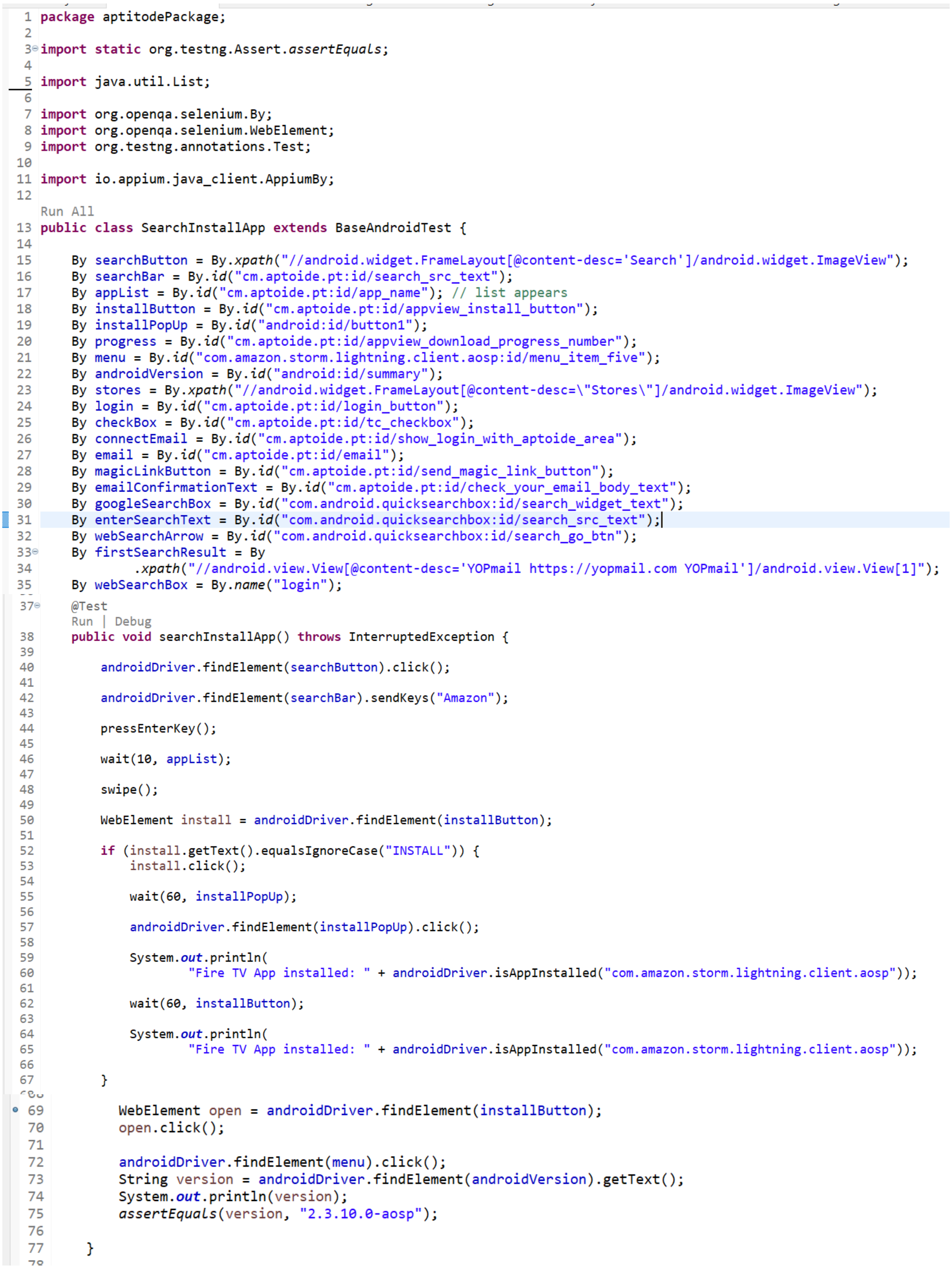


```
    public void swipeUp() {

        Dimension size = androidDriver.manage().window().getSize();
        int startY = (int) (size.height * 0.80);
        int endY = (int) (size.height * 0.50);
        int centerX = size.width/2;

        //Type of Pointer Input
        PointerInput finger = new PointerInput(PointerInput.Kind.TOUCH, "finger");
        //Create sequence object to add actions
        Sequence swipe = new Sequence(finger,1);
        // Move finger to starting position
        swipe.addAction(finger.createPointerMove(Duration.ofSeconds(0),PointerInput.Origin.viewport(),centerX,(int)startY));
        //Finger comes down into contact with screen
        swipe.addAction(finger.createPointerDown(0));
        //finger moves to end position
        swipe.addAction(finger.createPointerMove(Duration.ofMillis(700), PointerInput.Origin.viewport(),centerX,(int)endY));
        //Get Up finger from screen
        swipe.addAction(finger.createPointerUp(0));
        //Perform the actions
        androidDriver.perform(Arrays.asList(swipe));

    }

```

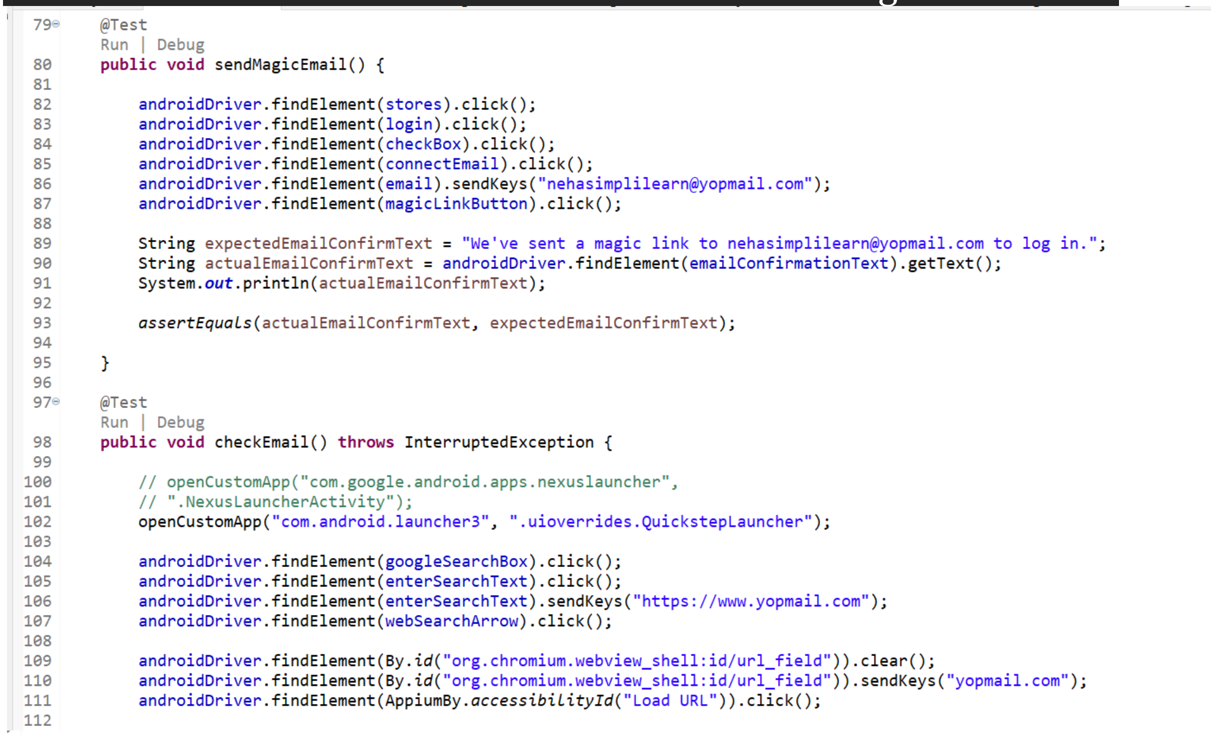


```
    public void wait(int i, By loc) {
        WebDriverWait wait = new WebDriverWait(androidDriver, Duration.ofSeconds(i));
        wait.until(ExpectedConditions.visibilityOfElementLocated(loc));
    }

    @AfterMethod
    public void clearActivities( ) {
        androidDriver.quit();
    }
}

```

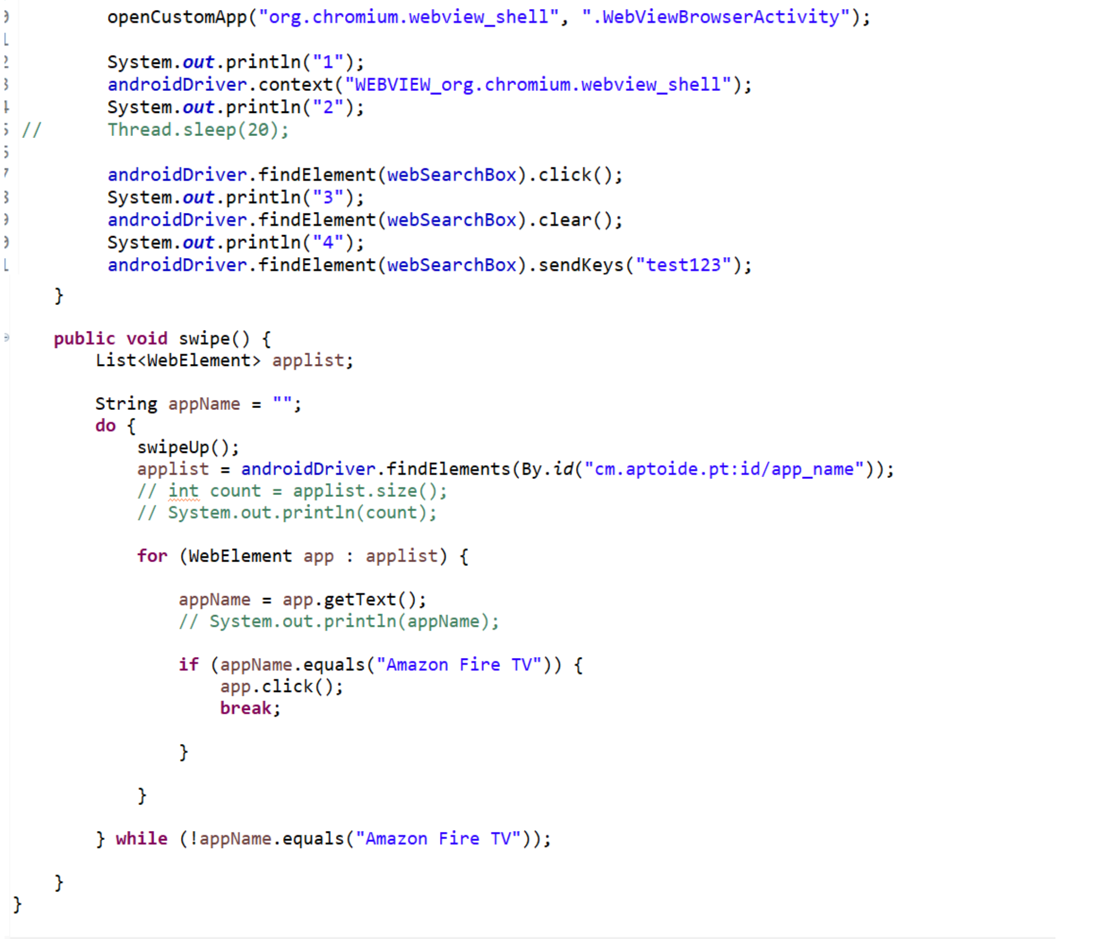
```
public class SearchInstallApp extends BaseAndroidTest
```



```
package aptitodePackage;

import static org.testng.Assert.assertEquals;

import java.util.List;

import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.testng.annotations.Test;

import io.appium.java_client.AppiumBy;

Run All
public class SearchInstallApp extends BaseAndroidTest {

    By searchButton = By.xpath("//android.widget.FrameLayout[@content-desc='Search']/android.widget.ImageView");
    By searchBar = By.id("cm.aptoide.pt:id/search_src_text");
    By appList = By.id("cm.aptoide.pt:id/app_name"); // list appears
    By installButton = By.id("cm.aptoide.pt:id/appview_install_button");
    By installPopUp = By.id("android:id/button1");
    By progress = By.id("cm.aptoide.pt:id/appview_download_progress_number");
    By menu = By.id("com.amazon.storm.lightning.client.aosp:id/menu_item_five");
    By androidVersion = By.id("android:id/summary");
    By stores = By.xpath("//android.widget.FrameLayout[@content-desc=\"Stores\"]/android.widget.ImageView");
    By login = By.id("cm.aptoide.pt:id/login_button");
    By checkBox = By.id("cm.aptoide.pt:id/tc_checkbox");
    By connectEmail = By.id("cm.aptoide.pt:id/show_login_with_aptoide_area");
    By email = By.id("cm.aptoide.pt:id/email");
    By magicLinkButton = By.id("cm.aptoide.pt:id/send_magic_link_button");
    By emailConfirmationText = By.id("cm.aptoide.pt:id/check_your_email_body_text");
    By googleSearchBox = By.id("com.android.quicksearchbox:id/search_widget_text");
    By enterSearchText = By.id("com.android.quicksearchbox:id/search_src_text");
    By webSearchArrow = By.id("com.android.quicksearchbox:id/search_go_btn");
    By firstSearchResult = By
            .xpath("//android.view.View[@content-desc='YOPmail https://yopmail.com YOPmail']/android.view.View[1]");
    By webSearchBox = By.name("login");
    @Test
    Run | Debug
    public void searchInstallApp() throws InterruptedException {

        androidDriver.findElement(searchButton).click();

        androidDriver.findElement(searchBar).sendKeys("Amazon");

        pressEnterKey();

        wait(10, appList);

        swipe();

        WebElement install = androidDriver.findElement(installButton);

        if (install.getText().equalsIgnoreCase("INSTALL")) {
            install.click();

            wait(60, installPopUp);

            androidDriver.findElement(installPopUp).click();

            System.out.println(
                    "Fire TV App installed: " + androidDriver.isAppInstalled("com.amazon.storm.lightning.client.aosp"));

            wait(60, installButton);

            System.out.println(
                    "Fire TV App installed: " + androidDriver.isAppInstalled("com.amazon.storm.lightning.client.aosp"));

        }
        WebElement open = androidDriver.findElement(installButton);
        open.click();

        androidDriver.findElement(menu).click();
        String version = androidDriver.findElement(androidVersion).getText();
        System.out.println(version);
        assertEquals(version, "2.3.10.0-aosp");

    }
```



```
    @Test
    Run | Debug
    public void sendMagicEmail() {

        androidDriver.findElement(stores).click();
        androidDriver.findElement(login).click();
        androidDriver.findElement(checkBox).click();
        androidDriver.findElement(connectEmail).click();
        androidDriver.findElement(email).sendKeys("nehasimplilearn@yopmail.com");
        androidDriver.findElement(magicLinkButton).click();

        String expectedEmailConfirmText = "We've sent a magic link to nehasimplilearn@yopmail.com to log in.";
        String actualEmailConfirmText = androidDriver.findElement(emailConfirmationText).getText();
        System.out.println(actualEmailConfirmText);

        assertEquals(actualEmailConfirmText, expectedEmailConfirmText);

    }

    @Test
    Run | Debug
    public void checkEmail() throws InterruptedException {

        // openCustomApp("com.google.android.apps.nexuslauncher",
        // ".NexusLauncherActivity");
        openCustomApp("com.android.launcher3", ".uioverrides.QuickstepLauncher");

        androidDriver.findElement(googleSearchBox).click();
        androidDriver.findElement(enterSearchText).click();
        androidDriver.findElement(enterSearchText).sendKeys("https://www.yopmail.com");
        androidDriver.findElement(webSearchArrow).click();

        androidDriver.findElement(By.id("org.chromium.webview_shell:id/url_field")).clear();
        androidDriver.findElement(By.id("org.chromium.webview_shell:id/url_field")).sendKeys("yopmail.com");
        androidDriver.findElement(AppiumBy.accessibilityId("Load URL")).click();

```



```
        openCustomApp("org.chromium.webview_shell", ".WebViewBrowserActivity");

        System.out.println("1");
        androidDriver.context("WEBVIEW_org.chromium.webview_shell");
        System.out.println("2");
//      Thread.sleep(20);

        androidDriver.findElement(webSearchBox).click();
        System.out.println("3");
        androidDriver.findElement(webSearchBox).clear();
        System.out.println("4");
        androidDriver.findElement(webSearchBox).sendKeys("test123");
    }

    public void swipe() {
        List<WebElement> applist;

        String appName = "";
        do {
            swipeUp();
            applist = androidDriver.findElements(By.id("cm.aptoide.pt:id/app_name"));
            // int count = applist.size();
            // System.out.println(count);

            for (WebElement app : applist) {

                appName = app.getText();
                // System.out.println(appName);

                if (appName.equals("Amazon Fire TV")) {
                    app.click();
                    break;

                }

            }

        } while (!appName.equals("Amazon Fire TV"));

    }
}
```

B. AWS

- Create an AWS account in console.
- Set up user/policies in IAM to perform actions with respect to service.
- Set up SSH and the user should be able to connect the ec2 machine from their local (Mac/Linux).
- Create an IAM user that can perform to create EC2 instance, S3 bucket, creating EBS volume, and mounting in AWS console.

Submit recorded video of aws tasks along names of ec2 machine, security groups, s3 and iam.

Videos have been submitted with following title:

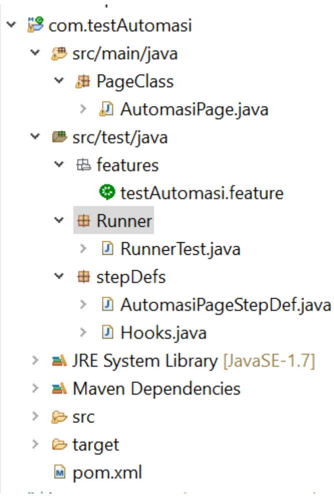
- aws -Account.mp4
- Creating_Connecting_EC2_ServerInstance_UsingPutty_FromLocal.mp4
- IAM_NewUser_AssinedToGroup_withPermissions.mp4

C. Selenium-Docker Grid + Jenkins: Using BDD Framework

Create a page object model BDD/cucumber project for testautomasi search course flow-> open testautomasi.com-> click on E-learning-> search (Search input from table) in course search->verify that the title of course contains “J-meter”-> click on course more link->click on buy now and verify message “Unknown column 'graduation' in 'field list'.

Run this BDD cucumber project on selenium docker image using ec2 on a scheduled run from Jenkins

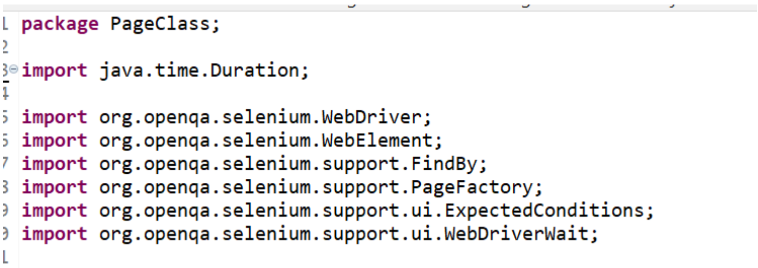
Code:



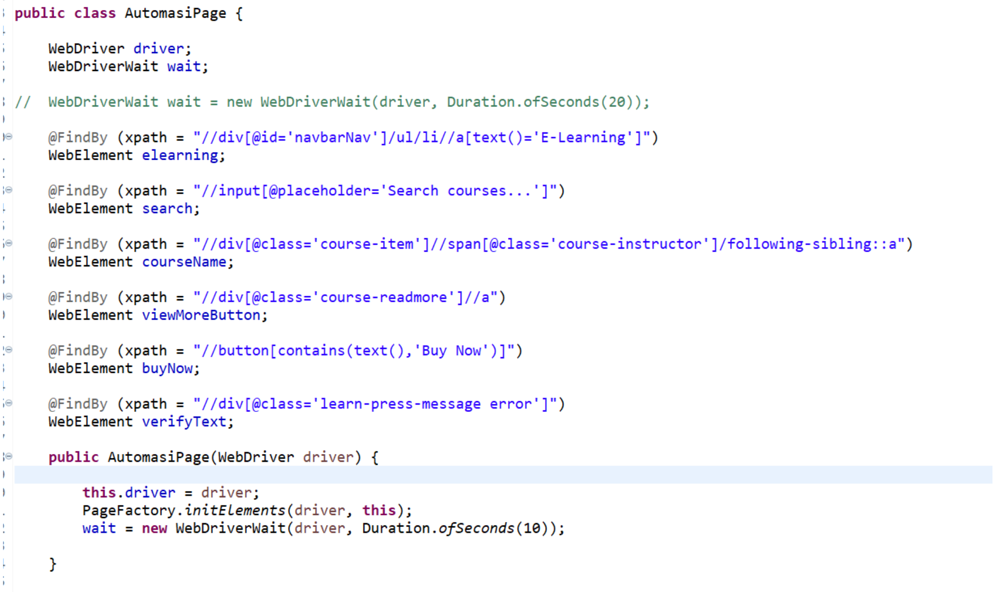
```
com.testAutomasi
  src/main/java
    PageClass
      AutomasiPage.java
  src/test/java
    features
      testAutomasi.feature
    Runner
      RunnerTest.java
    stepDefs
      AutomasiPageStepDef.java
      Hooks.java
  JRE System Library [JavaSE-1.7]
  Maven Dependencies
  src
  target
  pom.xml
```

src/main/java

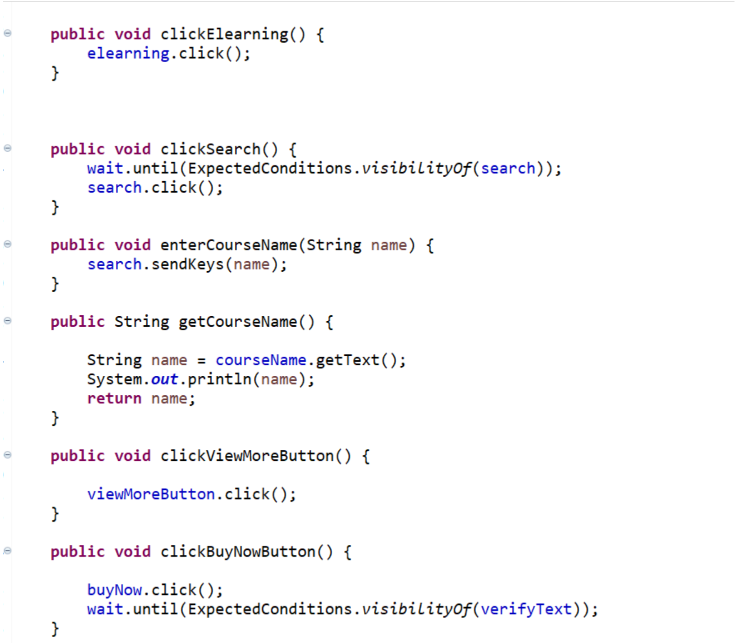
public class AutomasiPage



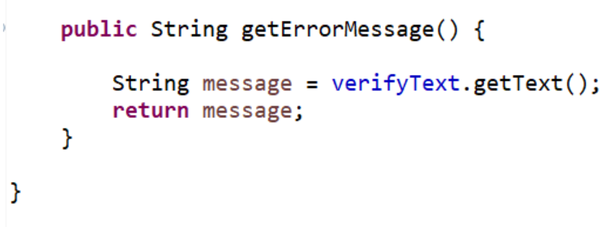
```java
package PageClass;

import java.time.Duration;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
```



```java
public class AutomasiPage {

    WebDriver driver;
    WebDriverWait wait;

//  WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(20));

    @FindBy (xpath = "//div[@id='navbarNav']/ul/li//a[text()='E-Learning']")
    WebElement elearning;

    @FindBy (xpath = "//input[@placeholder='Search courses...']")
    WebElement search;

    @FindBy (xpath = "//div[@class='course-item']//span[@class='course-instructor']/following-sibling::a")
    WebElement courseName;

    @FindBy (xpath = "//div[@class='course-readmore']//a")
    WebElement viewMoreButton;

    @FindBy (xpath = "//button[contains(text(),'Buy Now')]")
    WebElement buyNow;

    @FindBy (xpath = "//div[@class='learn-press-message error']")
    WebElement verifyText;

    public AutomasiPage(WebDriver driver) {

        this.driver = driver;
        PageFactory.initElements(driver, this);
        wait = new WebDriverWait(driver, Duration.ofSeconds(10));

    }
```



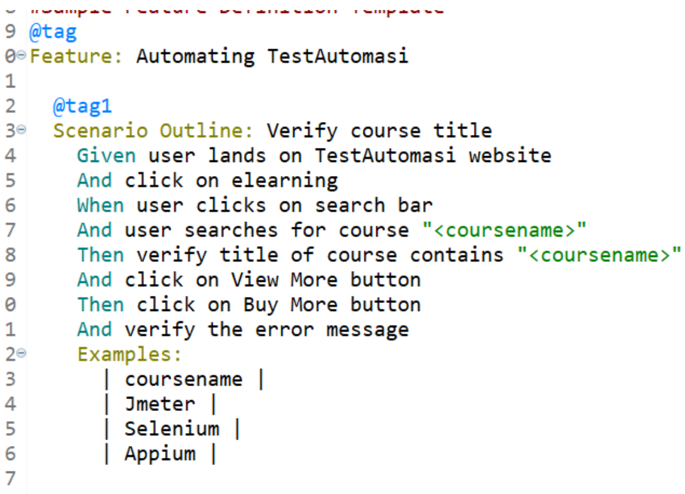
```java
    public void clickElearning() {
        elearning.click();
    }


    public void clickSearch() {
        wait.until(ExpectedConditions.visibilityOf(search));
        search.click();
    }

    public void enterCourseName(String name) {
        search.sendKeys(name);
    }

    public String getCourseName() {

        String name = courseName.getText();
        System.out.println(name);
        return name;
    }

    public void clickViewMoreButton() {

        viewMoreButton.click();
    }

    public void clickBuyNowButton() {

        buyNow.click();
        wait.until(ExpectedConditions.visibilityOf(verifyText));
    }
```



```
    public String getErrorMessage() {

        String message = verifyText.getText();
        return message;
    }

}
```

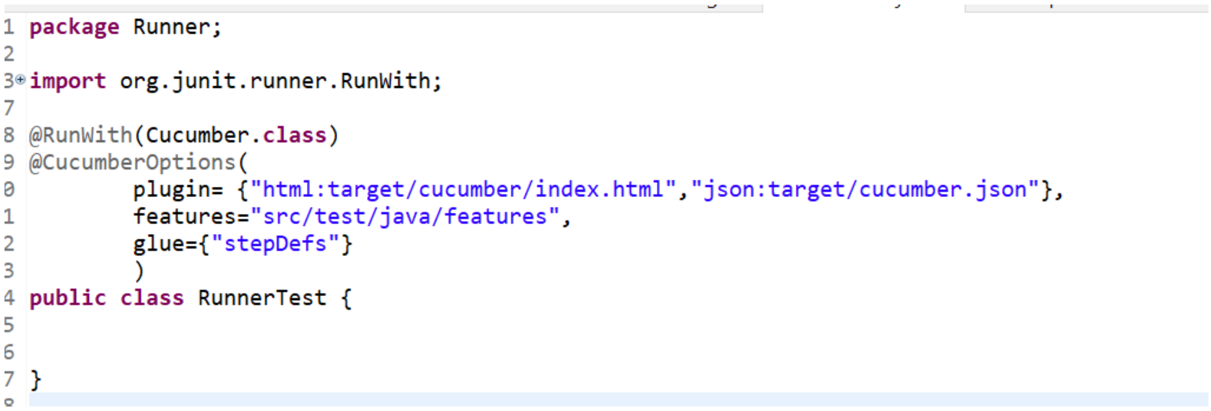
src/test/java

features : testAutomasi.feature



```
#Sample Feature Definition Template
@tag
Feature: Automating TestAutomasi

  @tag1
  Scenario Outline: Verify course title
    Given user lands on TestAutomasi website
    And click on elearning
    When user clicks on search bar
    And user searches for course "<coursename>"
    Then verify title of course contains "<coursename>"
    And click on View More button
    Then click on Buy More button
    And verify the error message
    Examples:
      | coursename |
      | Jmeter |
      | Selenium |
      | Appium |
```

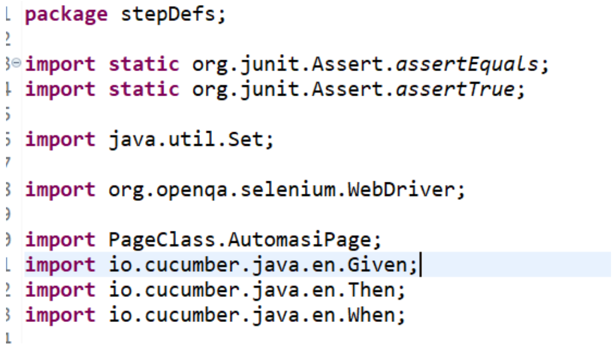
Runner: RunnerTest.java



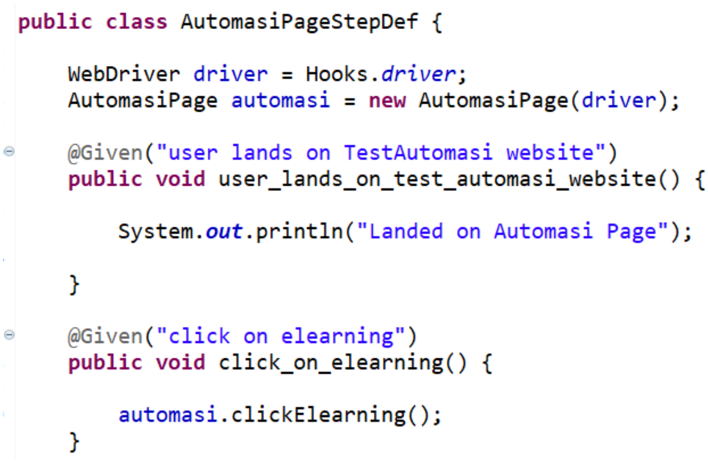
```
package Runner;

import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(
        plugin= {"html:target/cucumber/index.html","json:target/cucumber.json"},
        features="src/test/java/features",
        glue={"stepDefs"}
        )
public class RunnerTest {


}
```

stepDefs

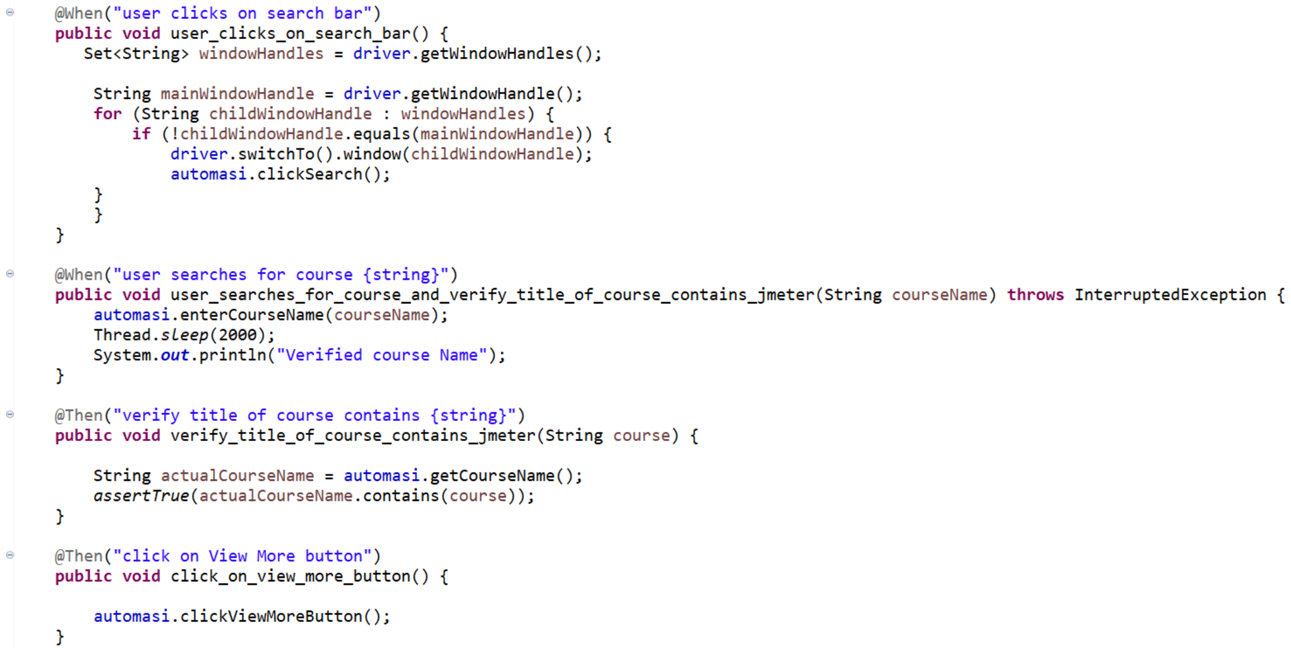
1. `public class AutomasiPageStepDef`



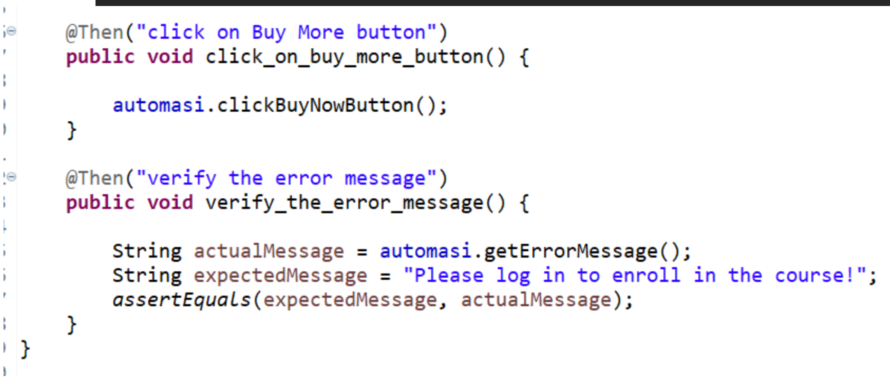
```java
package stepDefs;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;

import java.util.Set;

import org.openqa.selenium.WebDriver;

import PageClass.AutomasiPage;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
```



```java
public class AutomasiPageStepDef {

    WebDriver driver = Hooks.driver;
    AutomasiPage automasi = new AutomasiPage(driver);

    @Given("user lands on TestAutomasi website")
    public void user_lands_on_test_automasi_website() {

        System.out.println("Landed on Automasi Page");

    }

    @Given("click on elearning")
    public void click_on_elearning() {

        automasi.clickElearning();
    }
```

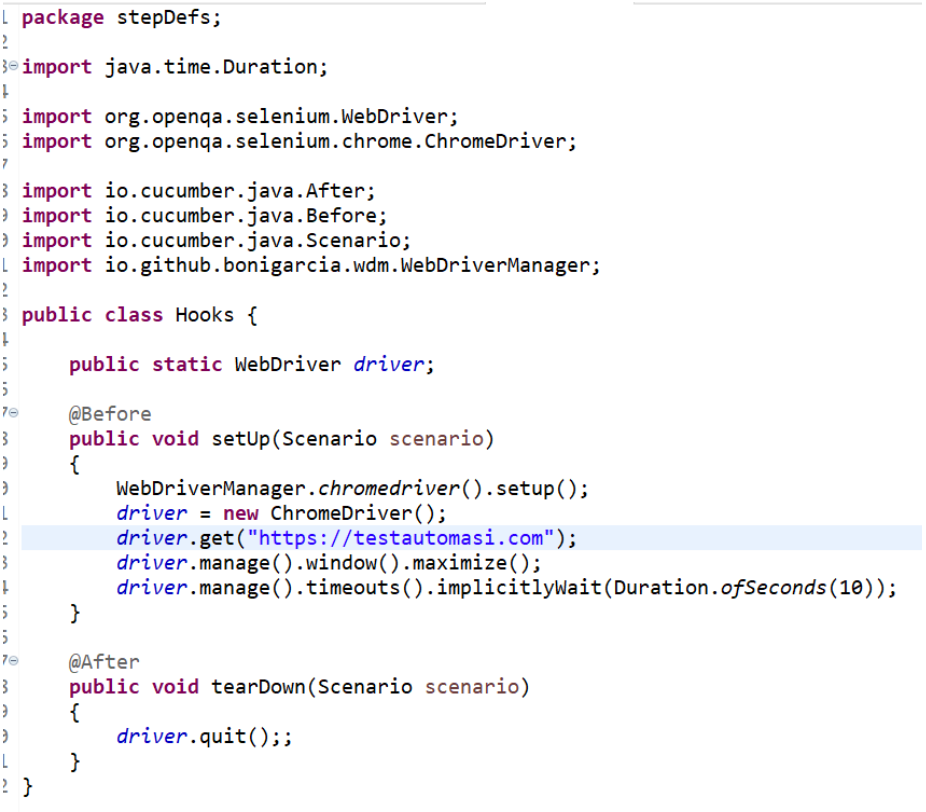


```java
    @When("user clicks on search bar")
    public void user_clicks_on_search_bar() {
       Set<String> windowHandles = driver.getWindowHandles();

        String mainWindowHandle = driver.getWindowHandle();
        for (String childWindowHandle : windowHandles) {
            if (!childWindowHandle.equals(mainWindowHandle)) {
                driver.switchTo().window(childWindowHandle);
                automasi.clickSearch();
        }
        }
    }

    @When("user searches for course {string}")
    public void user_searches_for_course_and_verify_title_of_course_contains_jmeter(String courseName) throws InterruptedException {
        automasi.enterCourseName(courseName);
        Thread.sleep(2000);
        System.out.println("Verified course Name");
    }

    @Then("verify title of course contains {string}")
    public void verify_title_of_course_contains_jmeter(String course) {

        String actualCourseName = automasi.getCourseName();
        assertTrue(actualCourseName.contains(course));
    }

    @Then("click on View More button")
    public void click_on_view_more_button() {

        automasi.clickViewMoreButton();
    }
```



```
    @Then("click on Buy More button")
    public void click_on_buy_more_button() {

        automasi.clickBuyNowButton();
    }

    @Then("verify the error message")
    public void verify_the_error_message() {

        String actualMessage = automasi.getErrorMessage();
        String expectedMessage = "Please log in to enroll in the course!";
        assertEquals(expectedMessage, actualMessage);
    }
}
```

2. Hooks.java



```
package stepDefs;

import java.time.Duration;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import io.cucumber.java.After;
import io.cucumber.java.Before;
import io.cucumber.java.Scenario;
import io.github.bonigarcia.wdm.WebDriverManager;

public class Hooks {

    public static WebDriver driver;

    @Before
    public void setUp(Scenario scenario)
    {
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        driver.get("https://testautomasi.com");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
    }

    @After
    public void tearDown(Scenario scenario)
    {
        driver.quit();;
    }
}
```

POM.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>

        <groupId>TestAutomasi_Phase4</groupId>
        <artifactId>com.testAutomasi</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <packaging>jar</packaging>
```

```
    <name>com.testAutomasi</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-junit -
->
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-junit</artifactId>
            <version>7.11.1</version>
        </dependency>

        <dependency>
            <groupId>info.cukes</groupId>
            <artifactId>cucumber-picocontainer</artifactId>
            <version>1.2.5</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>4.9.0</version>
        </dependency>


        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-
picocontainer -->
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-picocontainer</artifactId>
            <version>7.9.0</version>

        </dependency>


        <!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java --
>
        <dependency>
            <groupId>io.cucumber</groupId>
            <artifactId>cucumber-java</artifactId>
            <version>7.9.0</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/net.masterthought/cucumber-
reporting -->
        <dependency>
            <groupId>net.masterthought</groupId>
            <artifactId>cucumber-reporting</artifactId>
            <version>5.7.4</version>
        </dependency>
        <!--
https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
        <dependency>
            <groupId>io.github.bonigarcia</groupId>
            <artifactId>webdrivermanager</artifactId>
            <version>5.3.1</version>
        </dependency>

    </dependencies>

    <build>
        <plugins>

            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>3.0.0-M7</version>
                <configuration>
                    <includes>
                        <include>**/RunnerTest*.java</include>
                    </includes>
                    <testFailureIgnore>true</testFailureIgnore>
                </configuration>
            </plugin>
            <!--
https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-compiler-plugin
-->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.10.1</version>
            </plugin>

        </plugins>
    </build>
</project>
```

Running this bdd cucumber project on selenium docker image using ec2 on a scheduled run from jenkins

First, update the default Ubuntu packages lists for upgrades with the following command:
sudo apt-get update

Then, run the following command to install JDK 11:
sudo apt-get install openjdk-11-jdk

Now, we will install Jenkins itself. Issue the following four commands in sequence to initiate the installation from the Jenkins repository:

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo
tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```

Once that’s done, start the Jenkins service with the following command:
sudo systemctl start jenkins.service

To confirm its status, use:
sudo systemctl status jenkins

With Jenkins installed, we can proceed with adjusting the firewall settings. By default, Jenkins will run on port 8080.

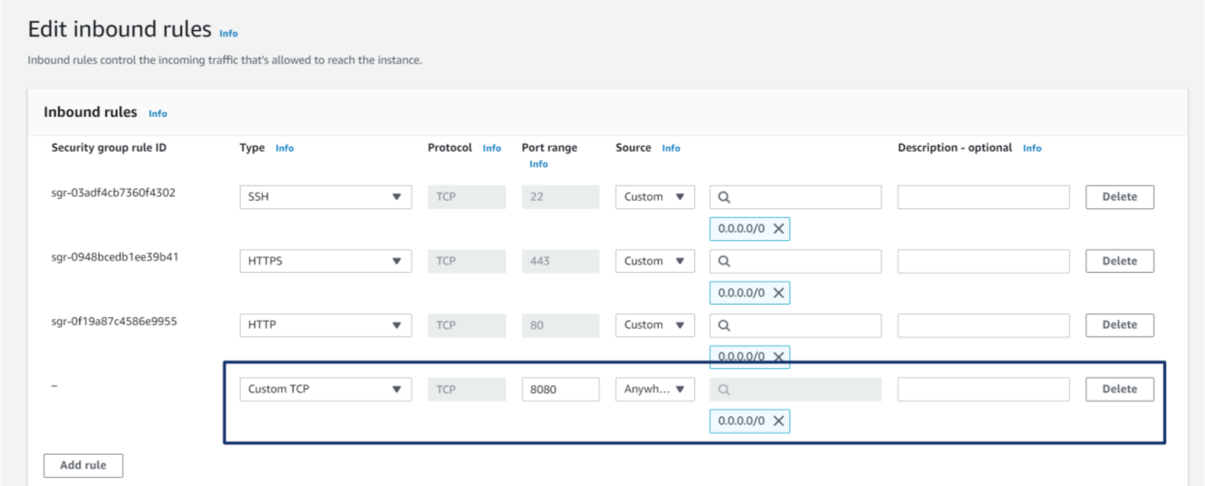
In order to ensure that this port is accessible, we will need to configure the built-in Ubuntu firewall (ufw). To open the 8080 port and enable the firewall, use the following commands:

sudo ufw allow 8080
sudo ufw enable

Once done, test whether the firewall is active using this command:
sudo ufw status

With the firewall configured, it’s time to set up Jenkins itself. Type in the IP of your EC2 along with the port number. The Jenkins setup wizard will open.
http://52.87.180.242:8080/

If Jenkins does not open. Make sure to add highlighted Inbound rule for 8080

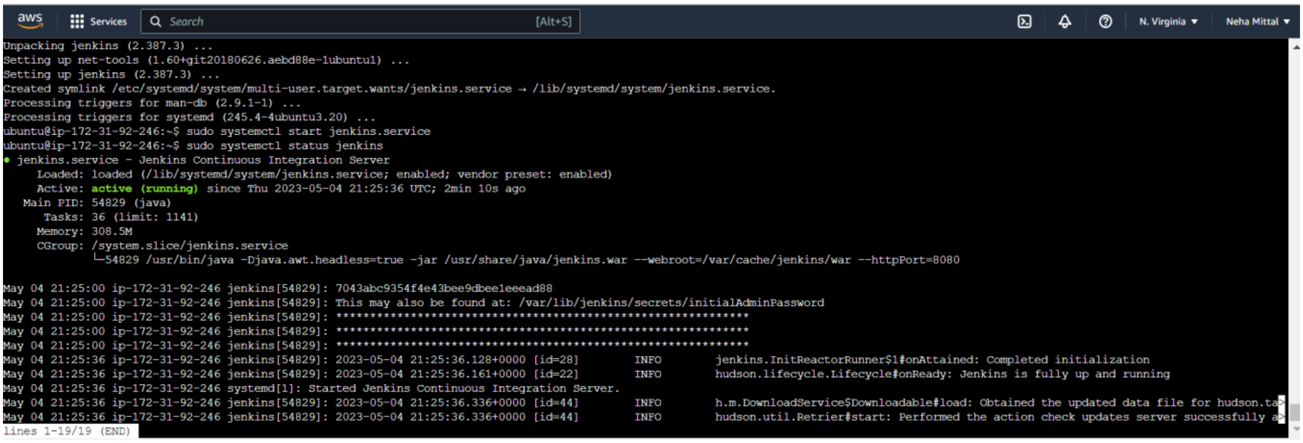


Edit inbound rules

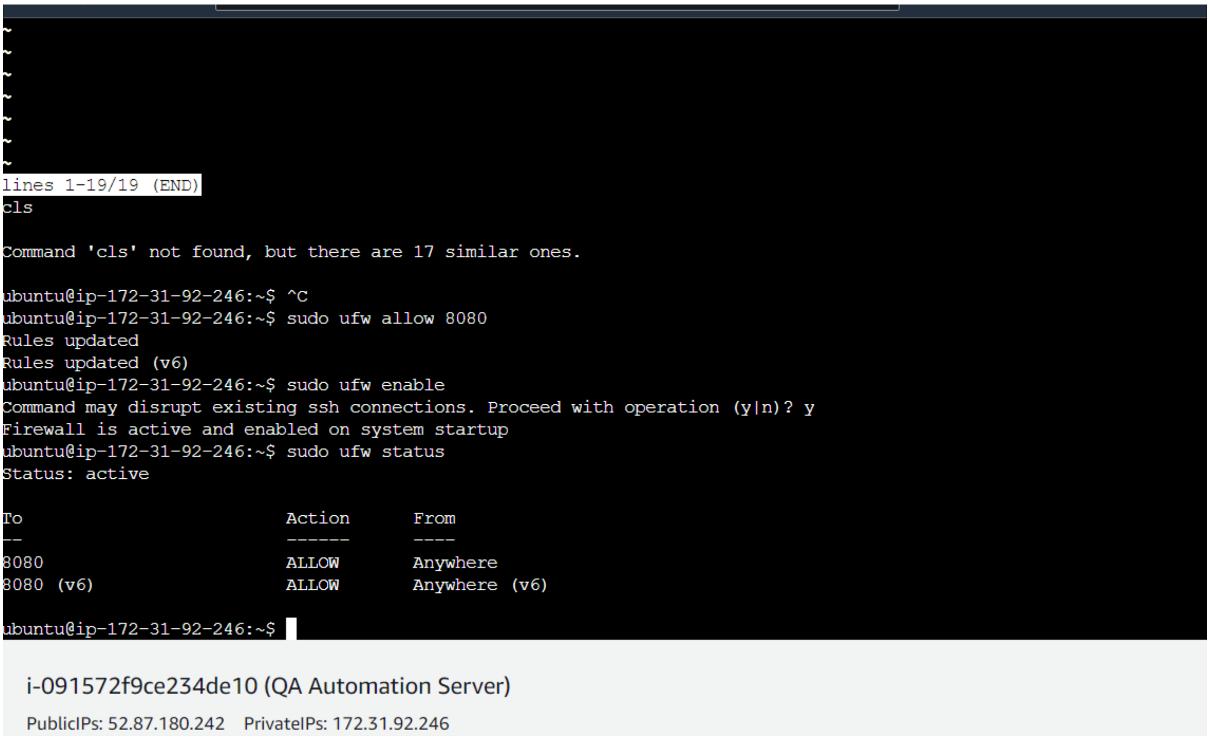
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-03adf4cb7360f4302	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0948bcedb1ee39b41	HTTPS	TCP	443	Custom	0.0.0.0/0		Delete
sgr-0f19a87c4586e9955	HTTP	TCP	80	Custom	0.0.0.0/0		Delete
–	Custom TCP	TCP	8080	Anywh...	0.0.0.0/0		Delete



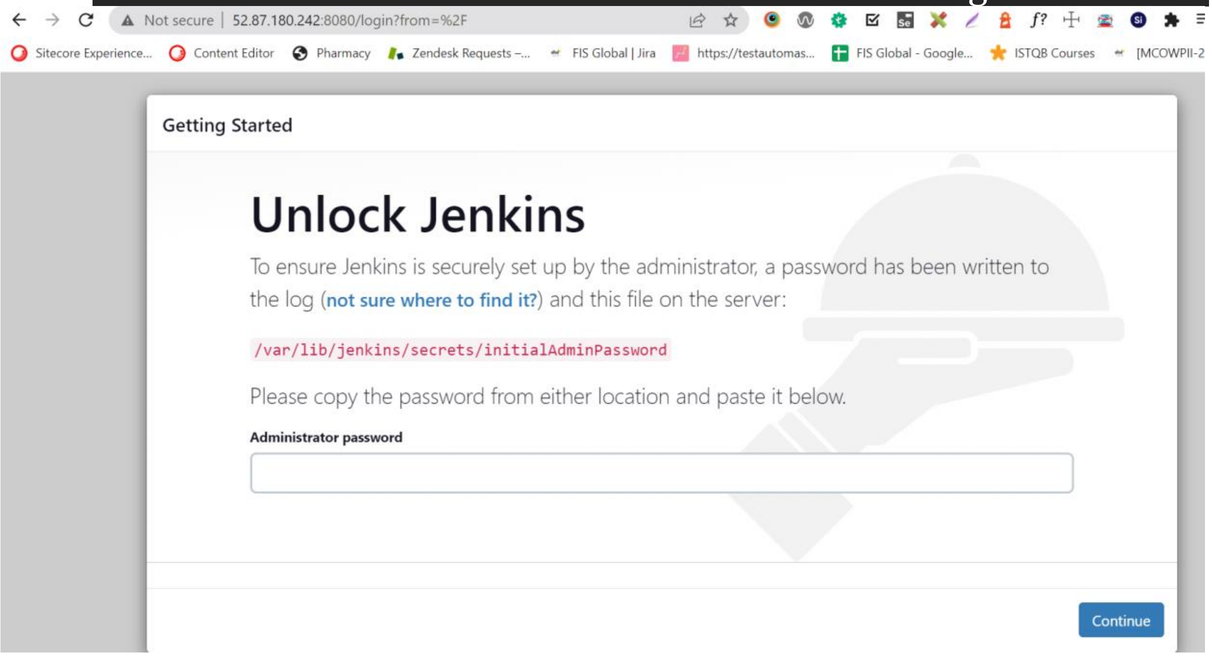
```
Unpacking jenkins (2.387.3) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up jenkins (2.387.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.20) ...
ubuntu@ip-172-31-92-246:~$ sudo systemctl start jenkins.service
ubuntu@ip-172-31-92-246:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-05-04 21:25:36 UTC; 2min 10s ago
   Main PID: 54829 (java)
      Tasks: 36 (limit: 1141)
     Memory: 308.5M
     CGroup: /system.slice/jenkins.service
             └─54829 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

May 04 21:25:00 ip-172-31-92-246 jenkins[54829]: 7043abc9354f4e43bee9dbee1eeead88
May 04 21:25:00 ip-172-31-92-246 jenkins[54829]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
May 04 21:25:00 ip-172-31-92-246 jenkins[54829]: *************************************************************
May 04 21:25:00 ip-172-31-92-246 jenkins[54829]: *************************************************************
May 04 21:25:00 ip-172-31-92-246 jenkins[54829]: *************************************************************
May 04 21:25:36 ip-172-31-92-246 jenkins[54829]: 2023-05-04 21:25:36.128+0000 [id=28]        INFO        jenkins.InitReactorRunner$1#onAttained: Completed initialization
May 04 21:25:36 ip-172-31-92-246 jenkins[54829]: 2023-05-04 21:25:36.161+0000 [id=22]        INFO        hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
May 04 21:25:36 ip-172-31-92-246 systemd[1]: Started Jenkins Continuous Integration Server.
May 04 21:25:36 ip-172-31-92-246 jenkins[54829]: 2023-05-04 21:25:36.336+0000 [id=44]        INFO        h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.ta>
May 04 21:25:36 ip-172-31-92-246 jenkins[54829]: 2023-05-04 21:25:36.336+0000 [id=44]        INFO        hudson.util.Retrier#start: Performed the action check updates server successfully a>
lines 1-19/19 (END)
```



```
~
~
~
~
~
~
~
lines 1-19/19 (END)
cls

Command 'cls' not found, but there are 17 similar ones.

ubuntu@ip-172-31-92-246:~$ ^C
ubuntu@ip-172-31-92-246:~$ sudo ufw allow 8080
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-92-246:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ubuntu@ip-172-31-92-246:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
8080                       ALLOW       Anywhere
8080 (v6)                  ALLOW       Anywhere (v6)

ubuntu@ip-172-31-92-246:~$
```

i-091572f9ce234de10 (QA Automation Server)

PublicIPs: 52.87.180.242   PrivateIPs: 172.31.92.246



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

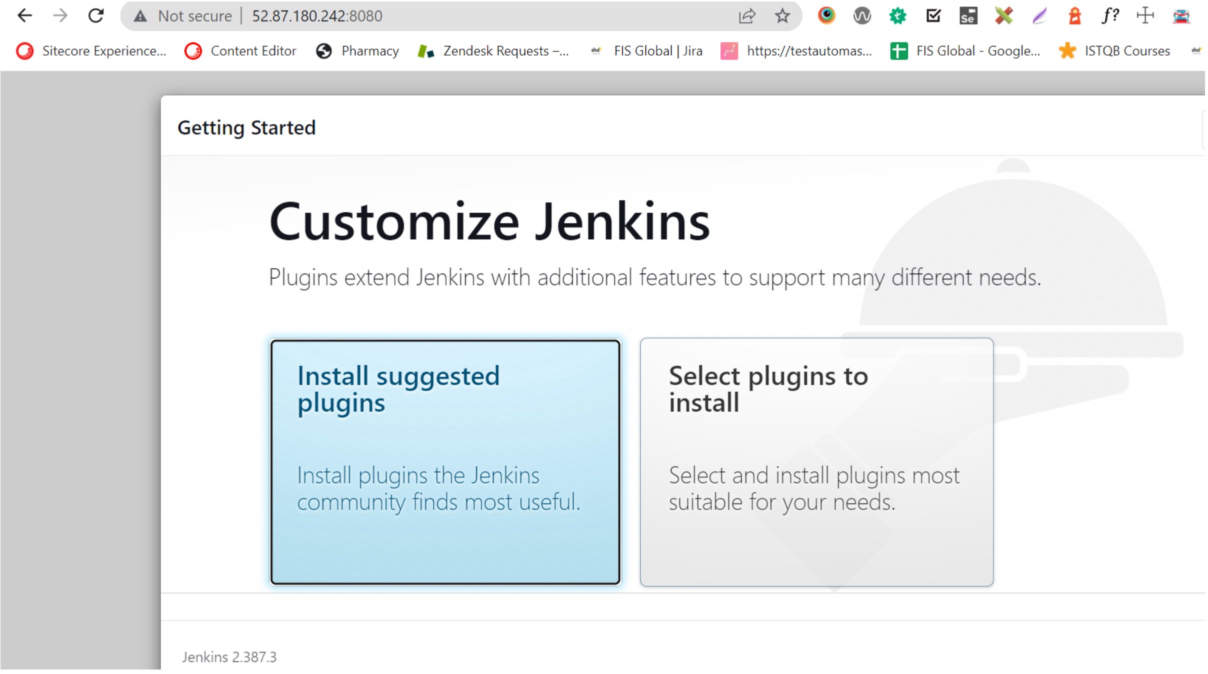
Administrator password

To check the initial password, use the cat command as indicated below:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

7043abc9354f4e43bee9dbee1eeead88

All Set! You can now start automating...



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

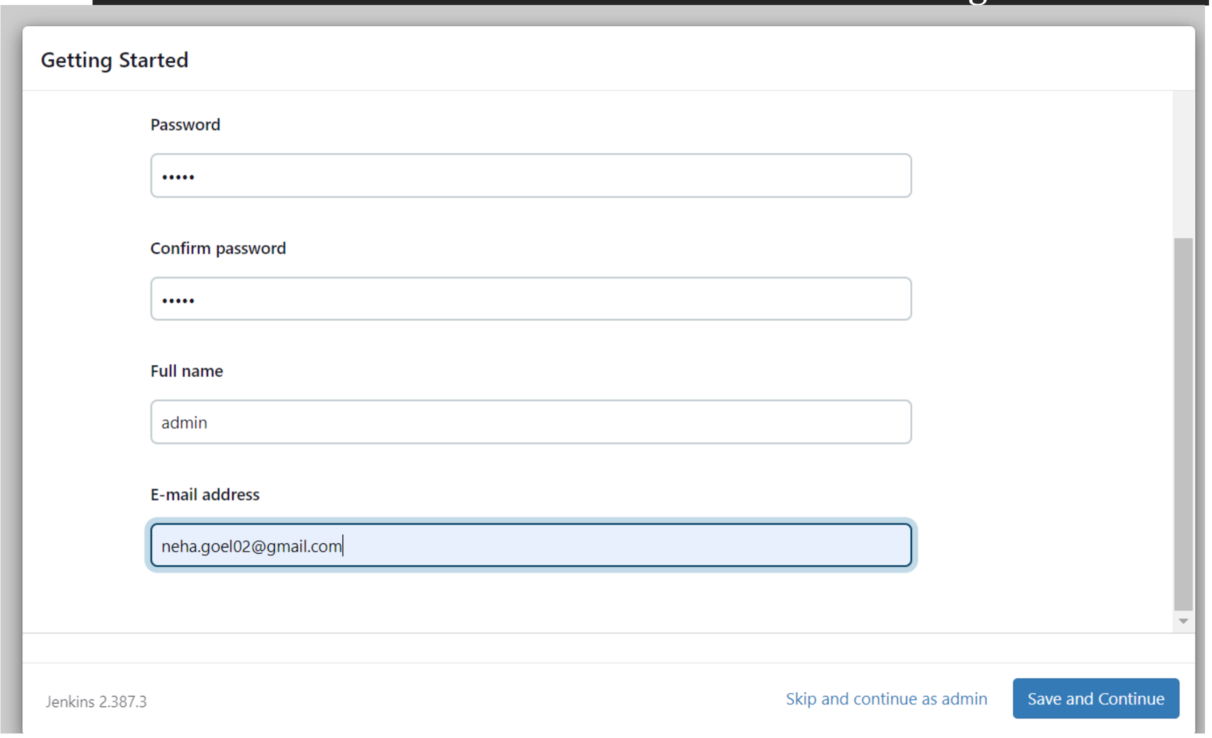
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.387.3



Getting Started

Password

•••••

Confirm password

•••••

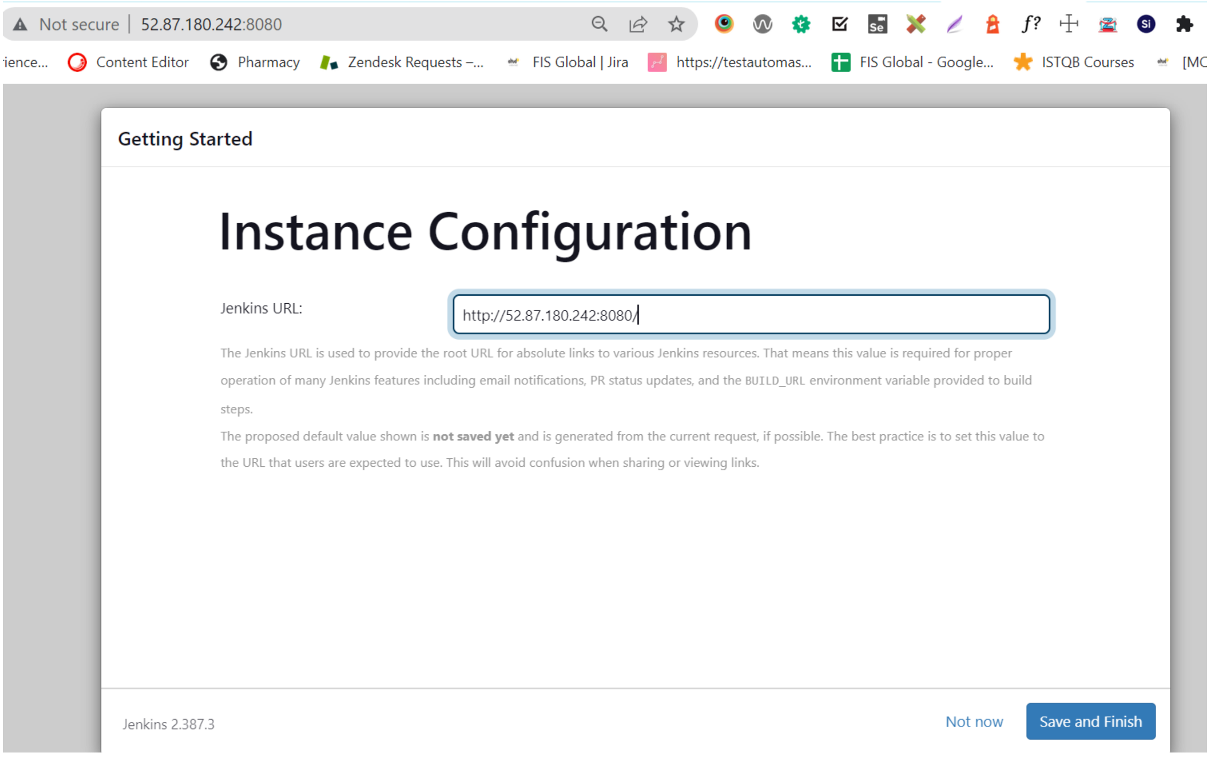
Full name

admin

E-mail address

neha.goel02@gmail.com

Jenkins 2.387.3



Getting Started

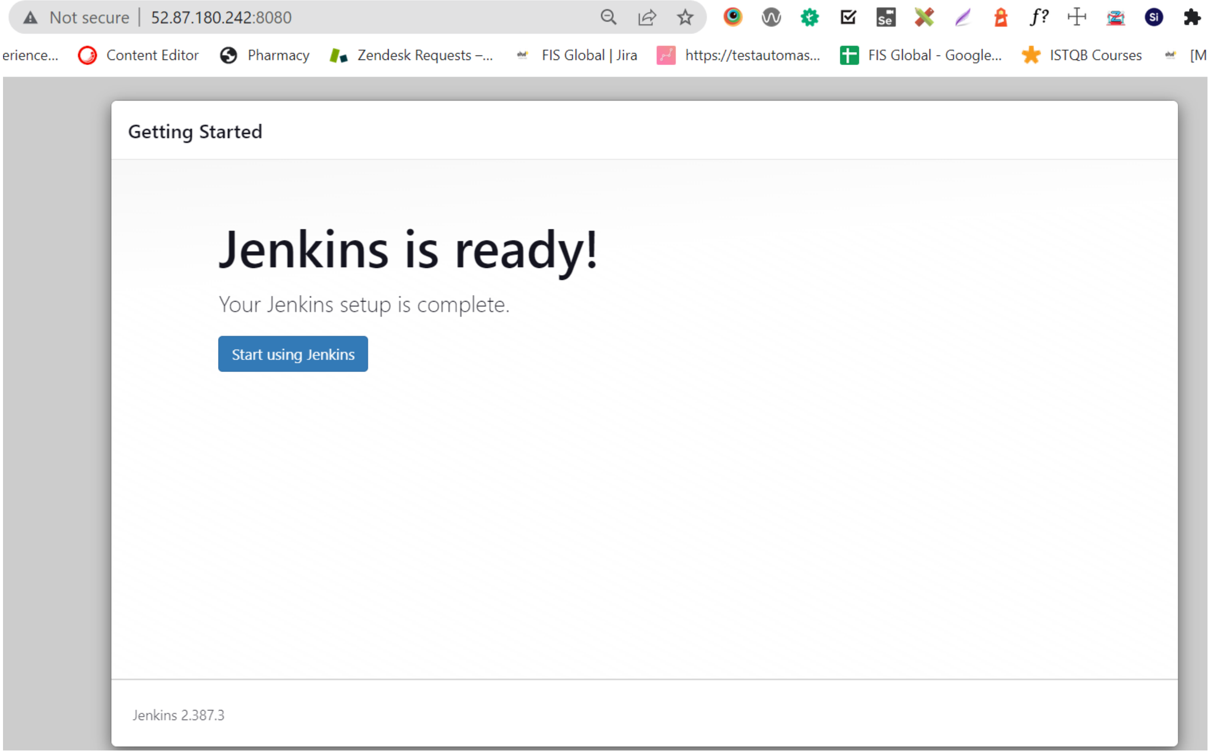
Instance Configuration

Jenkins URL: http://52.87.180.242:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.387.3



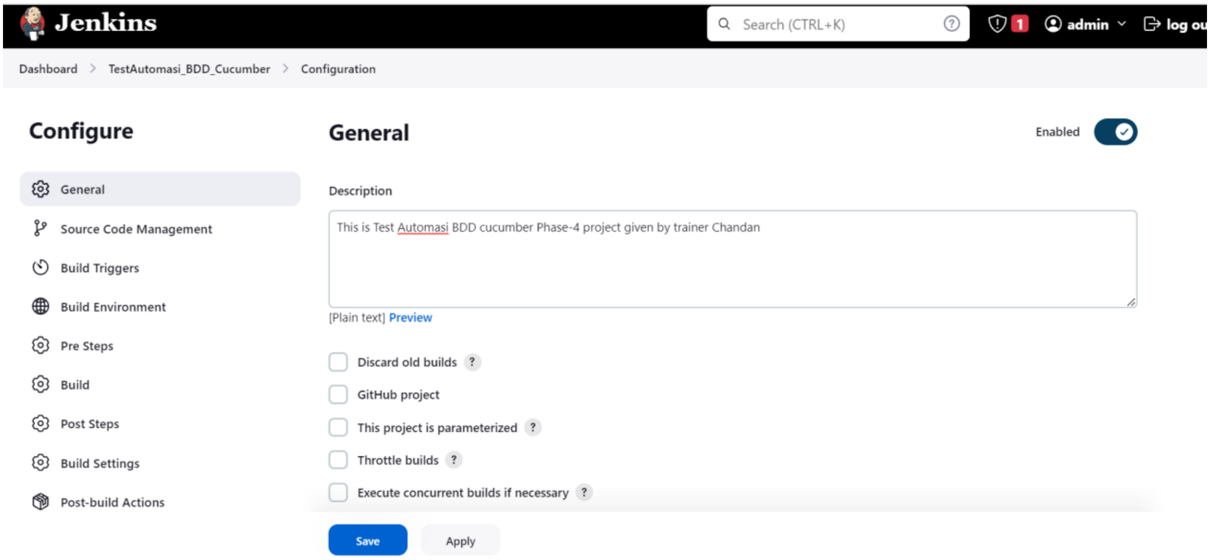
Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

Jenkins 2.387.3



General

Enabled

Description

This is Test Automasi BDD cucumber Phase-4 project given by trainer Chandan

[Plain text] Preview

□ Discard old builds ?

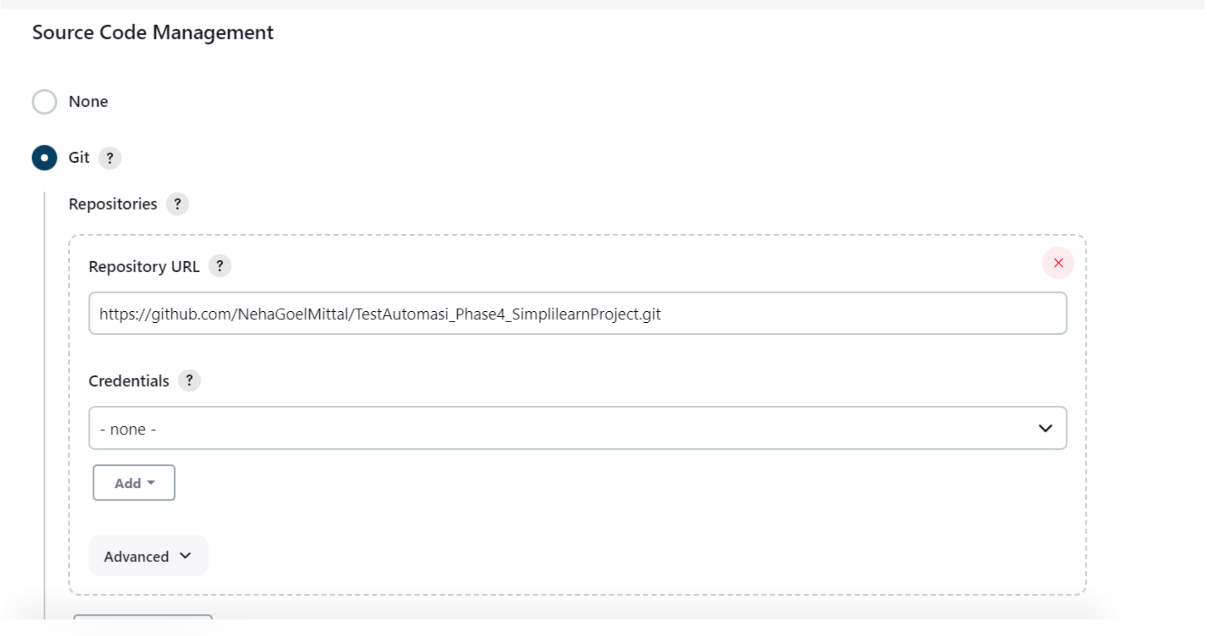
□ GitHub project

□ This project is parameterized ?

□ Throttle builds ?

□ Execute concurrent builds if necessary ?

Save Apply



Source Code Management

○ None

● Git ?

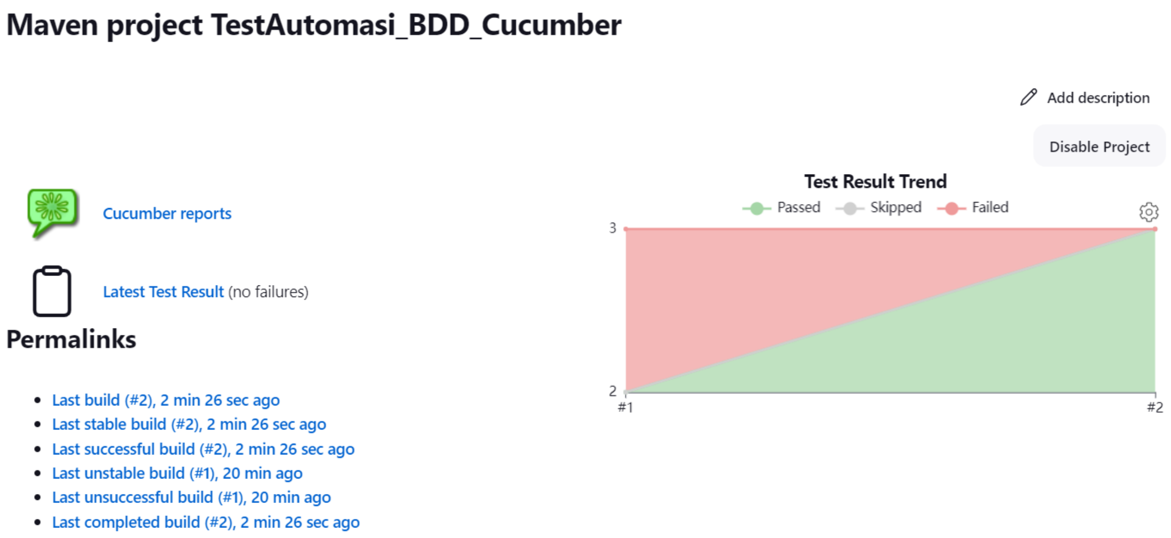
Repositories ?

Repository URL ?

https://github.com/NehaGoelMittal/TestAutomasi_Phase4_SimplilearnProject.git

Credentials ?

- none -

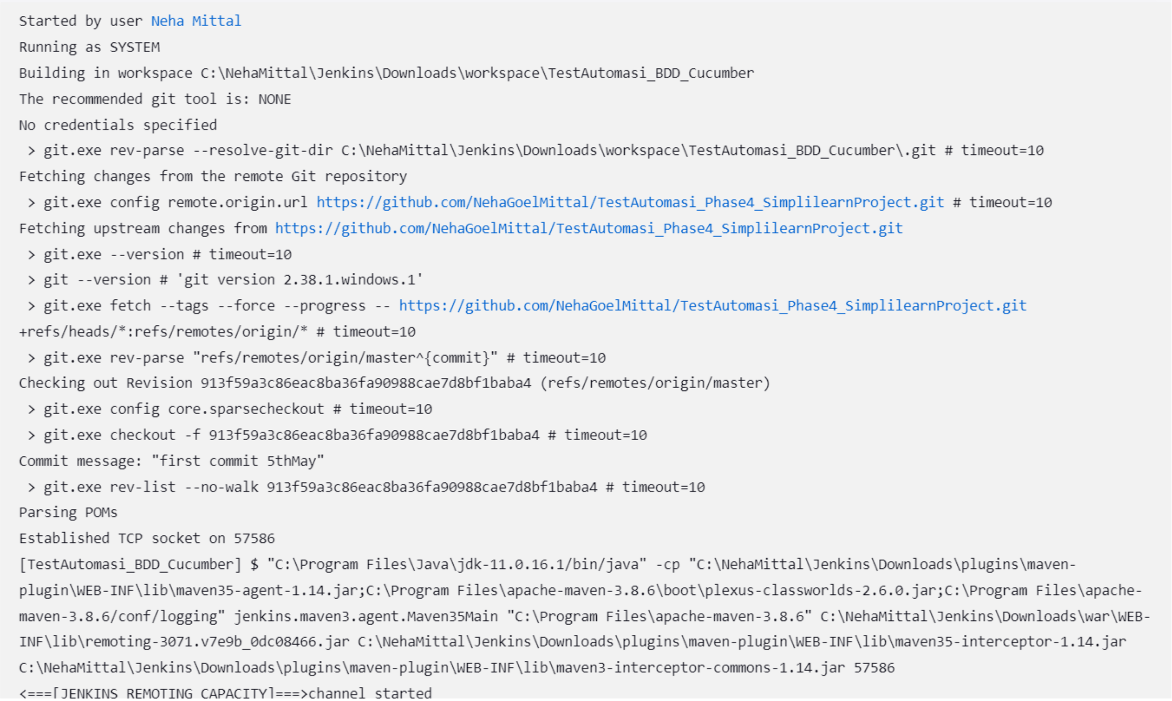


Maven project TestAutomasi_BDD_Cucumber

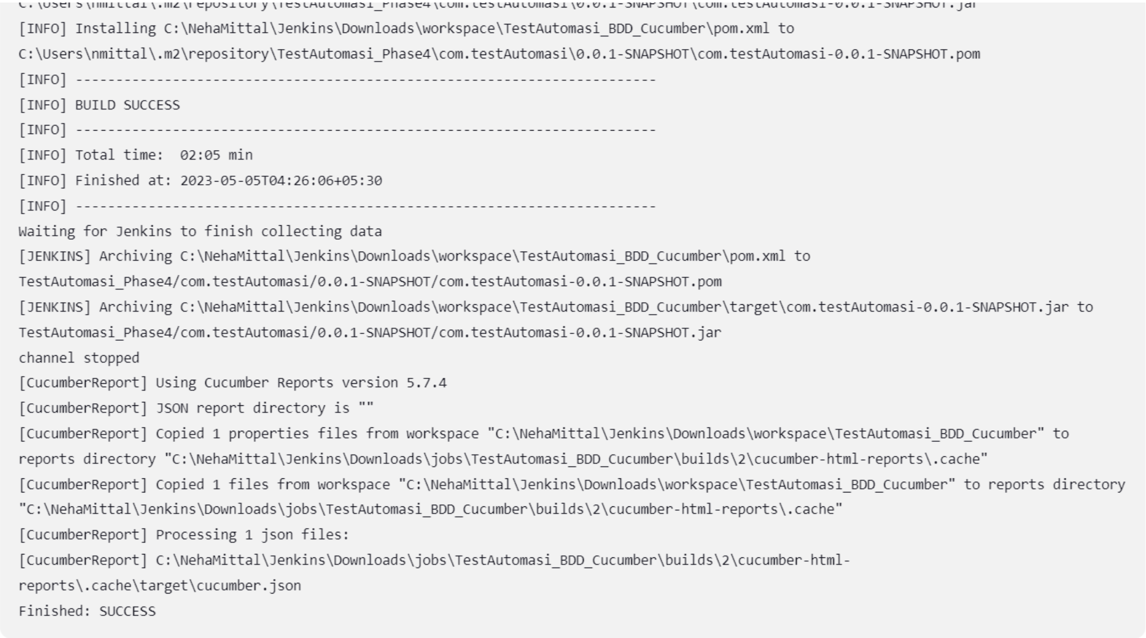
Latest Test Result (no failures)

Permalinks

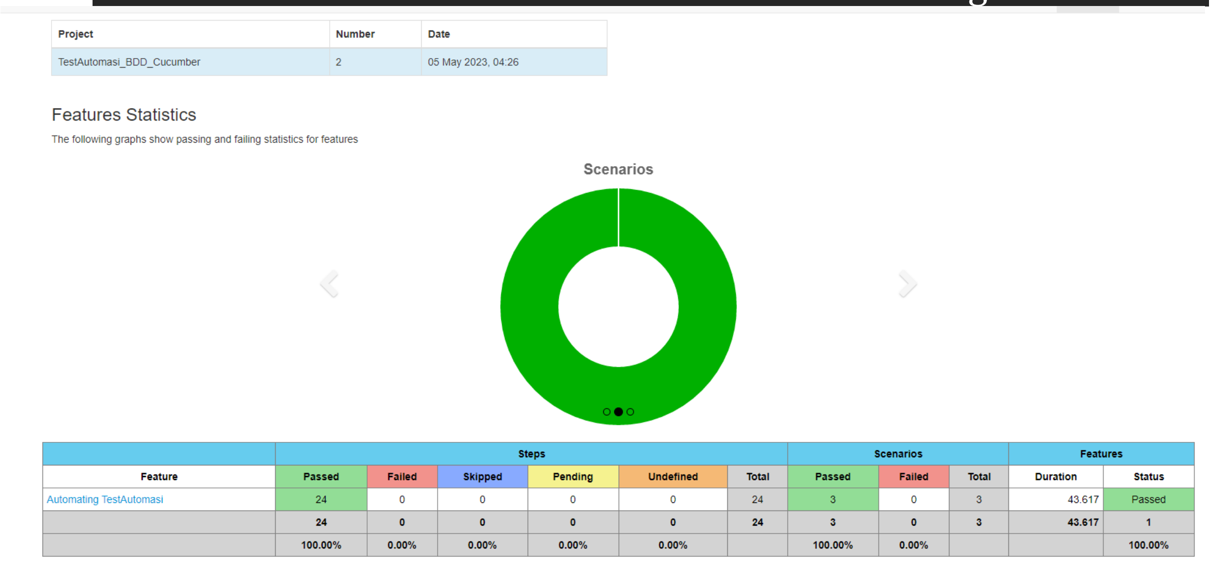
- Last build (#2), 2 min 26 sec ago
- Last stable build (#2), 2 min 26 sec ago
- Last successful build (#2), 2 min 26 sec ago
- Last unstable build (#1), 20 min ago
- Last unsuccessful build (#1), 20 min ago
- Last completed build (#2), 2 min 26 sec ago



```
Started by user Neha Mittal
Running as SYSTEM
Building in workspace C:\NehaMittal\Jenkins\Downloads\workspace\TestAutomasi_BDD_Cucumber
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\NehaMittal\Jenkins\Downloads\workspace\TestAutomasi_BDD_Cucumber\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/NehaGoelMittal/TestAutomasi_Phase4_SimplilearnProject.git # timeout=10
Fetching upstream changes from https://github.com/NehaGoelMittal/TestAutomasi_Phase4_SimplilearnProject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.38.1.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/NehaGoelMittal/TestAutomasi_Phase4_SimplilearnProject.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 913f59a3c86eac8ba36fa90988cae7d8bf1baba4 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 913f59a3c86eac8ba36fa90988cae7d8bf1baba4 # timeout=10
Commit message: "first commit 5thMay"
 > git.exe rev-list --no-walk 913f59a3c86eac8ba36fa90988cae7d8bf1baba4 # timeout=10
Parsing POMs
Established TCP socket on 57586
[TestAutomasi_BDD_Cucumber] $ "C:\Program Files\Java\jdk-11.0.16.1/bin/java" -cp "C:\NehaMittal\Jenkins\Downloads\plugins\maven-
plugin\WEB-INF\lib\maven35-agent-1.14.jar;C:\Program Files\apache-maven-3.8.6\boot\plexus-classworlds-2.6.0.jar;C:\Program Files\apache-
maven-3.8.6/conf/logging" jenkins.maven3.agent.Maven35Main "C:\Program Files\apache-maven-3.8.6" C:\NehaMittal\Jenkins\Downloads\war\WEB-
INF\lib\remoting-3071.v7e9b_0dc08466.jar C:\NehaMittal\Jenkins\Downloads\plugins\maven-plugin\WEB-INF\lib\maven35-interceptor-1.14.jar
C:\NehaMittal\Jenkins\Downloads\plugins\maven-plugin\WEB-INF\lib\maven3-interceptor-commons-1.14.jar 57586
<===[JENKINS REMOTING CAPACITY]===>channel started
```



```
C:\Users\nmittal\.m2\repository\TestAutomasi_Phase4\com.testAutomasi\0.0.1-SNAPSHOT\com.testAutomasi-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\NehaMittal\Jenkins\Downloads\workspace\TestAutomasi_BDD_Cucumber\pom.xml to
C:\Users\nmittal\.m2\repository\TestAutomasi_Phase4\com.testAutomasi\0.0.1-SNAPSHOT\com.testAutomasi-0.0.1-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  02:05 min
[INFO] Finished at: 2023-05-05T04:26:06+05:30
[INFO] ------------------------------------------------------------------------
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving C:\NehaMittal\Jenkins\Downloads\workspace\TestAutomasi_BDD_Cucumber\pom.xml to
TestAutomasi_Phase4/com.testAutomasi/0.0.1-SNAPSHOT/com.testAutomasi-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving C:\NehaMittal\Jenkins\Downloads\workspace\TestAutomasi_BDD_Cucumber\target\com.testAutomasi-0.0.1-SNAPSHOT.jar to
TestAutomasi_Phase4/com.testAutomasi/0.0.1-SNAPSHOT/com.testAutomasi-0.0.1-SNAPSHOT.jar
channel stopped
[CucumberReport] Using Cucumber Reports version 5.7.4
[CucumberReport] JSON report directory is ""
[CucumberReport] Copied 1 properties files from workspace "C:\NehaMittal\Jenkins\Downloads\workspace\TestAutomasi_BDD_Cucumber" to
reports directory "C:\NehaMittal\Jenkins\Downloads\jobs\TestAutomasi_BDD_Cucumber\builds\2\cucumber-html-reports\.cache"
[CucumberReport] Copied 1 files from workspace "C:\NehaMittal\Jenkins\Downloads\workspace\TestAutomasi_BDD_Cucumber" to reports directory
"C:\NehaMittal\Jenkins\Downloads\jobs\TestAutomasi_BDD_Cucumber\builds\2\cucumber-html-reports\.cache"
[CucumberReport] Processing 1 json files:
[CucumberReport] C:\NehaMittal\Jenkins\Downloads\jobs\TestAutomasi_BDD_Cucumber\builds\2\cucumber-html-
reports\.cache\target\cucumber.json
Finished: SUCCESS
```

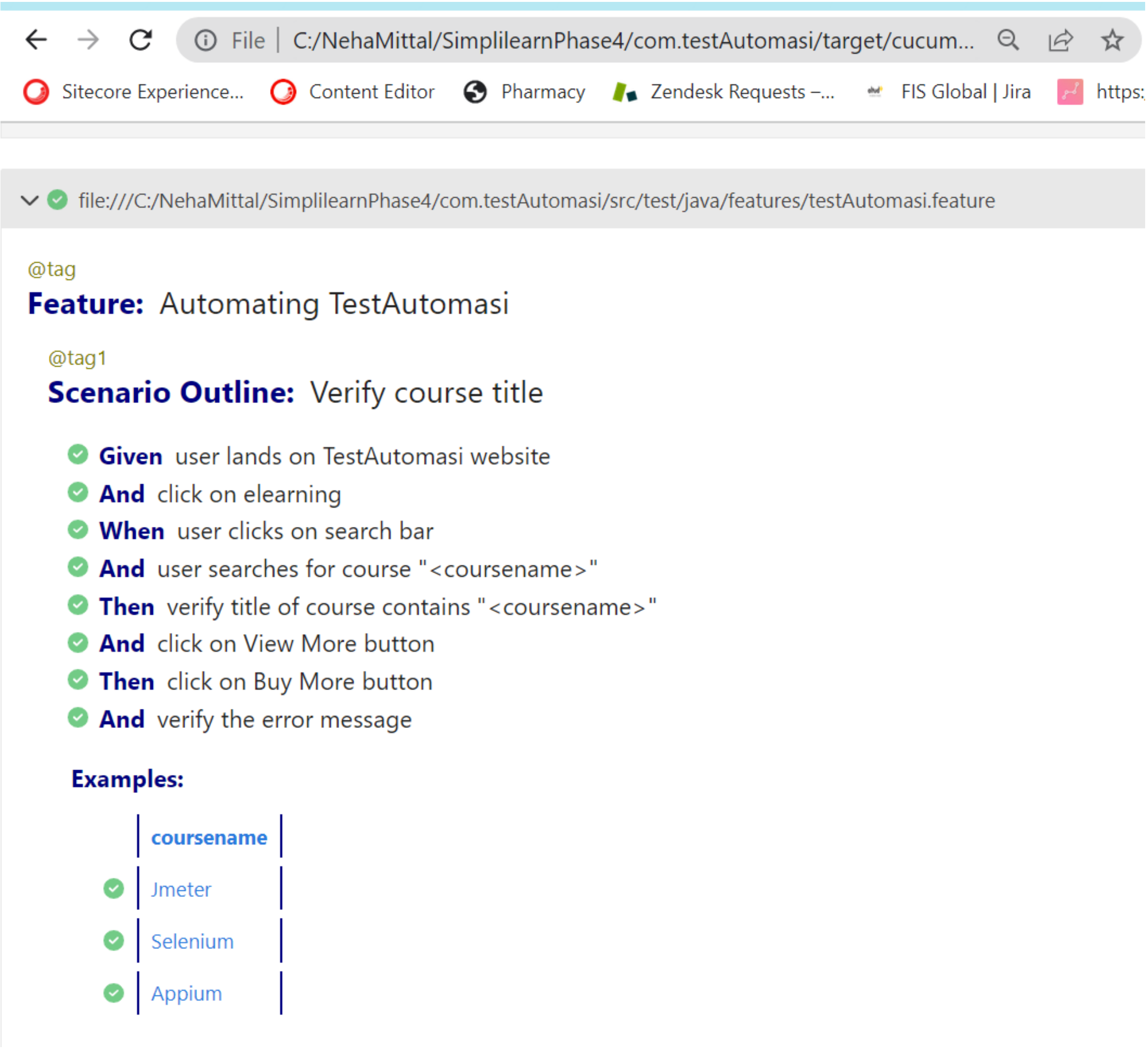


Project	Number	Date
TestAutomasi_BDD_Cucumber	2	05 May 2023, 04:26

Features Statistics

The following graphs show passing and failing statistics for features

	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Automating TestAutomasi	24	0	0	0	0	24	3	0	3	43.617	Passed
	24	0	0	0	0	24	3	0	3	43.617	1
	100.00%	0.00%	0.00%	0.00%	0.00%		100.00%	0.00%			100.00%



file:///C:/NehaMittal/SimplilearnPhase4/com.testAutomasi/src/test/java/features/testAutomasi.feature

@tag

Feature: Automating TestAutomasi

@tag1

Scenario Outline: Verify course title

- ✓ **Given** user lands on TestAutomasi website
- ✓ **And** click on elearning
- ✓ **When** user clicks on search bar
- ✓ **And** user searches for course "<coursename>"
- ✓ **Then** verify title of course contains "<coursename>"
- ✓ **And** click on View More button
- ✓ **Then** click on Buy More button
- ✓ **And** verify the error message

Examples:

	coursename
✓	Jmeter
✓	Selenium
✓	Appium