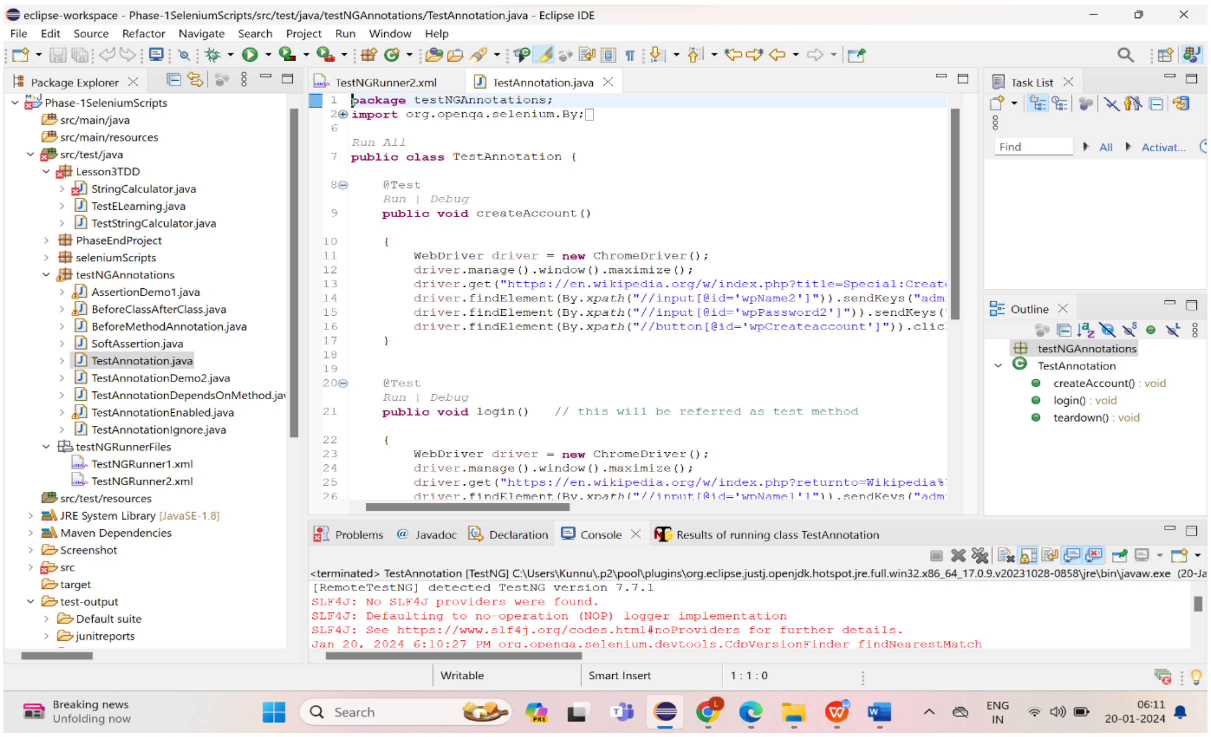
1. **Test Annotations :**

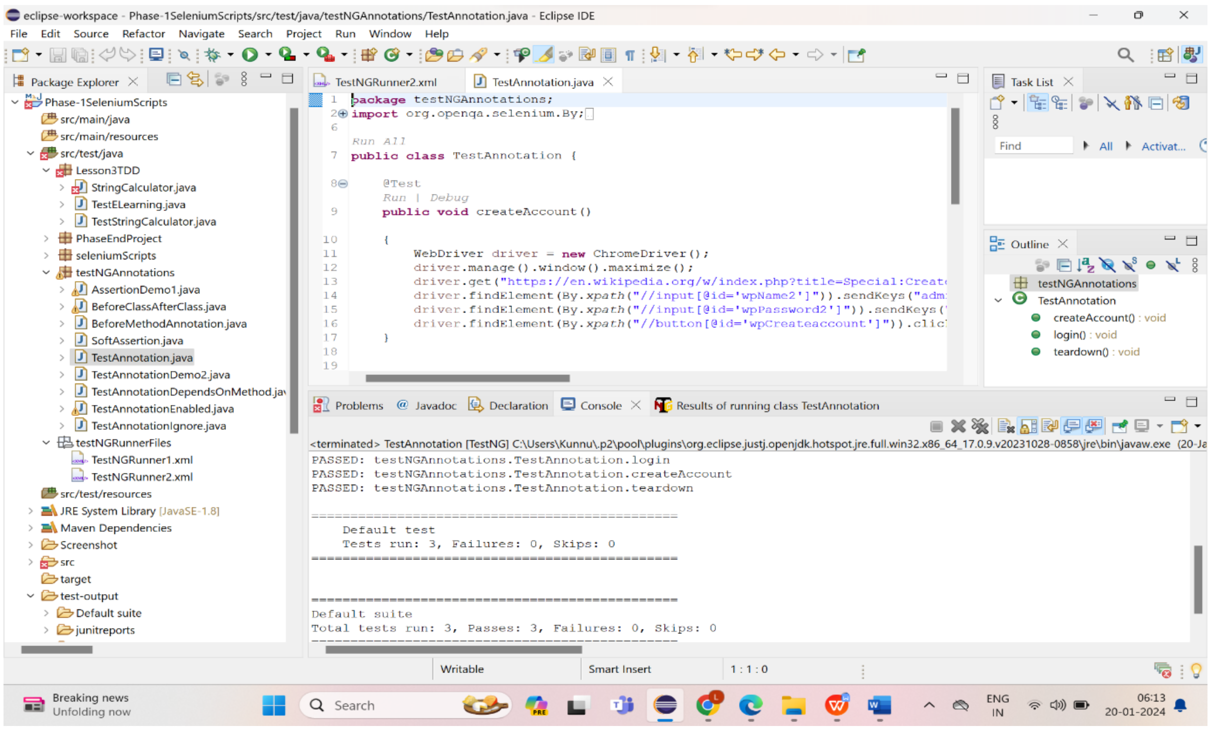
```
package testNGAnnotations;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import
org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

 public class TestAnnotation {

// in order to execute below script written in the
method use TestNG annotation

// ** TestNG annotation will be always written above
the Java method

@Test

public void createAccount()

{

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

driver.get("https://en.wikipedia.org/w/index.php?title=Special:Creat
eAccount&returnto=Wikipedia%3ASign+up&returntoquery=centralAuthA
utologinTried%3D1%26centralAuthError%3DNot%2Bcentrally%2Blogge
d%2Bin");


driver.findElement(By.xpath("//input[@id='wpName2']")).sendKeys(
"admin");


driver.findElement(By.xpath("//input[@id='wpPassword2']")).sendKeys("a
dmin@123");
```

```
        driver.findElement(By.xpath("//button[@id='wpCreateaccount']")).cli
ck();

        }


                @Test

                public void login()   // this will be referred as test method

        {

                WebDriver driver = new ChromeDriver();

                driver.manage().window().maximize();

        driver.get("https://en.wikipedia.org/w/index.php?returnto=Wikipedia
%3ASign+up&title=Special:UserLogin&centralAuthAutologinTried=1&ce
ntralAuthError=Not+centrally+logged+in");


        driver.findElement(By.xpath("//input[@id='wpName1']")).sendKeys(
"admin");


driver.findElement(By.xpath("//input[@id='wpPassword1']")).sendKeys("a
dmin@123");


        driver.findElement(By.xpath("//button[@id='wpLoginAttempt']")).clic
k();



        }


                @Test

                 public void teardown()

        {
            WebDriver driver = new ChromeDriver();

            driver.manage().window().maximize();

        driver.get("https://en.wikipedia.org/w/index.php?returnto=Wikipedia%3ASign+up&title=Special:UserLogin&centralAuthAutologinTried=1&centralAuthError=Not+centrally+logged+in");

            driver.close();



        }

}
```



```
package testNGAnnotations;
import org.openqa.selenium.By;

Run All
public class TestAnnotation {

    @Test
    Run | Debug
    public void createAccount()

    {
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en.wikipedia.org/w/index.php?title=Special:Creat
        driver.findElement(By.xpath("//input[@id='wpName2']")).sendKeys("adm
        driver.findElement(By.xpath("//input[@id='wpPassword2']")).sendKeys(
        driver.findElement(By.xpath("//button[@id='wpCreateaccount']")).clic
    }


    @Test
    Run | Debug
    public void login()    // this will be referred as test method

    {
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en.wikipedia.org/w/index.php?returnto=Wikipedia%
        driver.findElement(By.xpath("//input[@id='wpName1']")).sendKeys("adm
```

```
<terminated> TestAnnotation [TestNG] C:\Users\Kunnu\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe  (20-Ja
[RemoteTestNG] detected TestNG version 7.7.1
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Jan 20, 2024 6:10:27 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
```

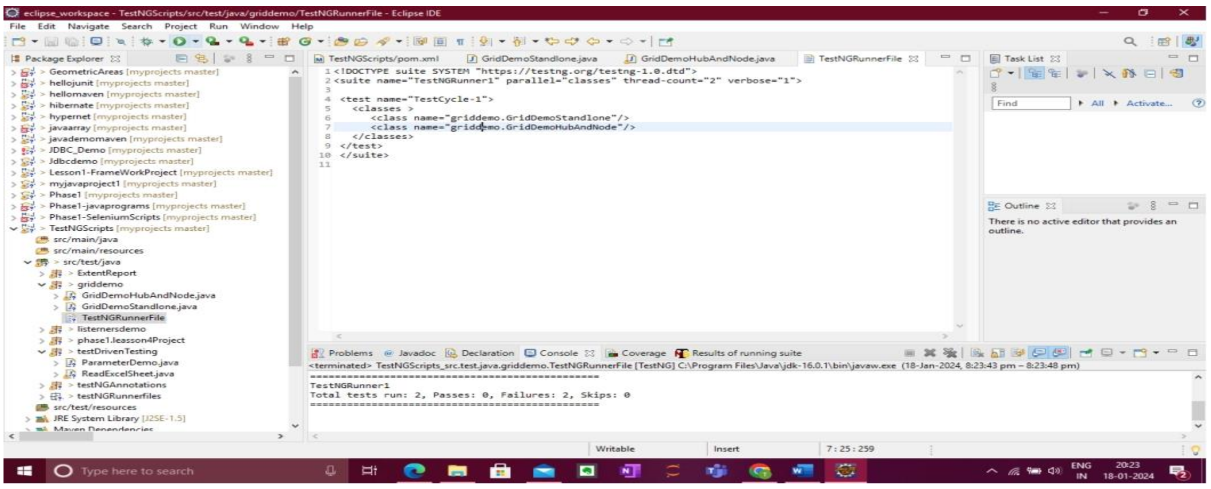


```
package testNGAnnotations;
import org.openqa.selenium.By;

public class TestAnnotation {

    @Test
    public void createAccount()

    {
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://en.wikipedia.org/w/index.php?title=Special:Creat
        driver.findElement(By.xpath("//input[@id='wpName2']")).sendKeys("adm
        driver.findElement(By.xpath("//input[@id='wpPassword2']")).sendKeys(
        driver.findElement(By.xpath("//button[@id='wpCreateaccount']")).clic
    }
```

```
<terminated> TestAnnotation [TestNG] C:\Users\Kunnu\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (20-Ja
PASSED: testNGAnnotations.TestAnnotation.login
PASSED: testNGAnnotations.TestAnnotation.createAccount
PASSED: testNGAnnotations.TestAnnotation.teardown

===============================================
    Default test
    Tests run: 3, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
===============================================
```

## 2.Test parallel execution :

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
    <suite name="TestNGRunner1" parallel="classes" thread-count="2" verbose="1">
    <test name="TestCycle-1">
      <classes >
          <class name="griddemo.GridDemoStandlone"/>
          <class name="griddemo.GridDemoHubAndNode"/>
      </classes>
     </test>
     </suite>
```



```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="TestNGRunner1" parallel="classes" thread-count="2" verbose="1">

 <test name="TestCycle-1">
   <classes >
      <class name="griddemo.GridDemoStandlone"/>
      <class name="griddemo.GridDemoHubAndNode"/>
   </classes>
 </test>
 </suite>
```

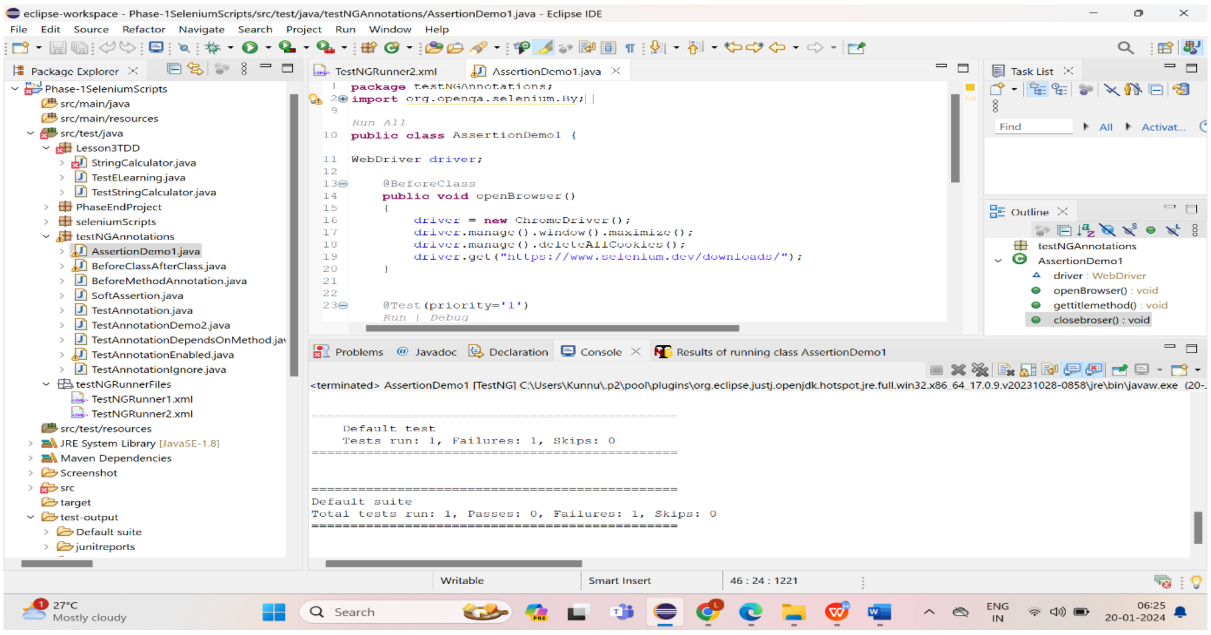
```
<terminated> TestNGScripts_src.test.java.griddemo.TestNGRunnerFile [TestNG] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (18-Jan-2024, 8:23:43 pm – 8:23:48 pm)
===============================================
TestNGRunner1
Total tests run: 2, Passes: 0, Failures: 2, Skips: 0
===============================================
```

## 3.Hard Assertion and soft Assertion :

### Hard Assertion:

```
 package testNGAnnotations;

        import org.openqa.selenium.By;

        import org.openqa.selenium.WebDriver;

        import org.openqa.selenium.chrome.ChromeDriver;

        import org.testng.Assert;

        import org.testng.annotations.AfterClass;

        import org.testng.annotations.BeforeClass;

        import org.testng.annotations.Test;

        public class HardAsseration {

        WebDriver driver;

       @BeforeClass

       public void openBrowser()

       {

              driver = new ChromeDriver();

              driver.manage().window().maximize();

              driver.manage().deleteAllCookies();

              driver.get("https://www.selenium.dev/downloads/");

       }

       @Test(priority='1')

       public void gettitlemethod() throws InterruptedException

       {

              String expectedTitle = "DownloadsSelenium";

              String actualTitle = driver.getTitle(); // Downloads | Selenium

              // we will check if expected title == actual title-> add assertions

              Assert.assertEquals(actualTitle, expectedTitle);
```

```
                    Thread.sleep(2000);

                    System.out.println("Assertion was passed");

                    //driver.findElement(By.xpath("(//div[@class='card-body px-0 text-center'])[3]/descendant::a[3]")).click();

                    System.out.println("click on the link");

                    }


          @AfterClass

          public void closebroser()

          {

                    driver.close();

          }

}
```



eclipse-workspace - Phase-1SeleniumScripts/src/test/java/testNGAnnotations/AssertionDemo1.java - Eclipse IDE

```
package testNGAnnotations;
import org.openqa.selenium.By;

Run All
public class AssertionDemo1 {

WebDriver driver;


    @BeforeClass
    public void openBrowser()
    {
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().deleteAllCookies();
        driver.get("https://www.selenium.dev/downloads/");
    }


    @Test(priority='1')
    Run | Debug
```

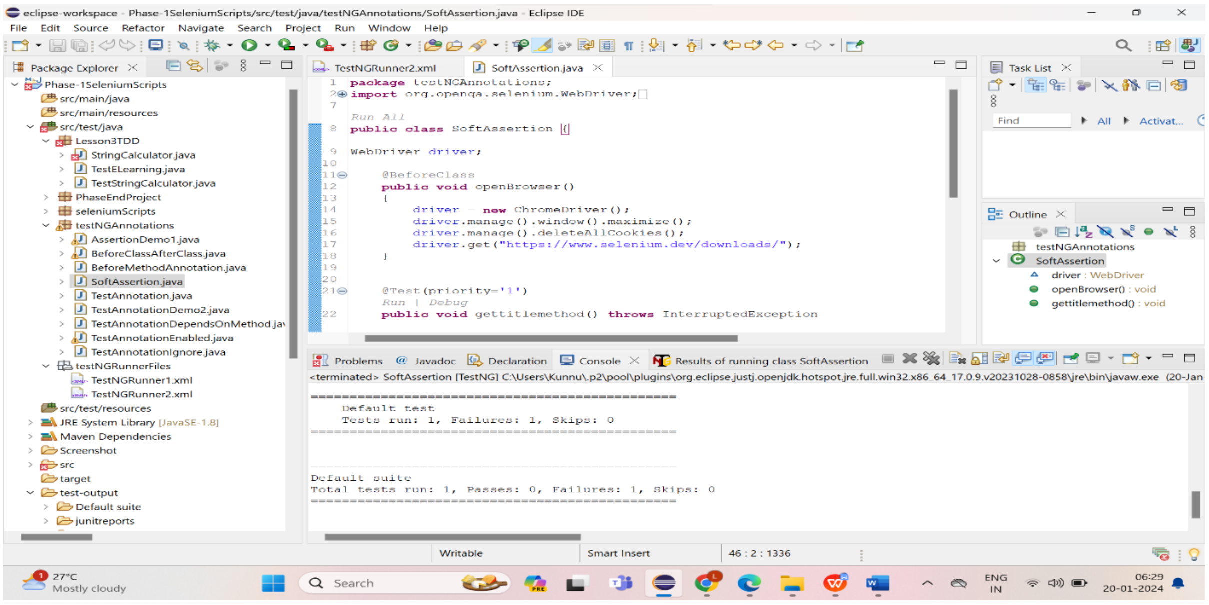
```
<terminated> AssertionDemo1 [TestNG] C:\Users\Kunnu\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (20-

===============================================
    Default test
    Tests run: 1, Failures: 1, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
===============================================
```

## Soft Assertion:

```
package testNGAnnotations;

            import org.openqa.selenium.WebDriver;
```

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

  import org.testng.annotations.BeforeClass;

  import org.testng.annotations.Test;

  import org.testng.asserts.SoftAssert;

  public class SoftAsseration {

  WebDriver driver;

 @BeforeClass

 public void openBrowser()

 {

         driver = new ChromeDriver();

         driver.manage().window().maximize();

         driver.manage().deleteAllCookies();

         driver.get("https://www.selenium.dev/downloads/");

 }

 @Test(priority='1')

 public void gettitlemethod() throws InterruptedException

 {

         SoftAssert sf = new SoftAssert();

         String expectedTitle = "DownloadsSelenium";

         String actualTitle = driver.getTitle(); // Downloads | Selenium

         // we will check if expected title == actual title-> add assertions

         sf.assertEquals(actualTitle, expectedTitle,"The title are not matching");// error will be
captured

         // but in case of soft assert.. further lines of code will continue to execute

         Thread.sleep(2000);

         System.out.println("Assertion was passed");

         //driver.findElement(By.xpath("(//div[@class='card-body px-0 text-
center'])[3]/descendant::a[3]")).click();

         System.out.println("click on the link");
```

```
                    sf.assertAll(); // print  all the assertion that have failed.

        }

}
```



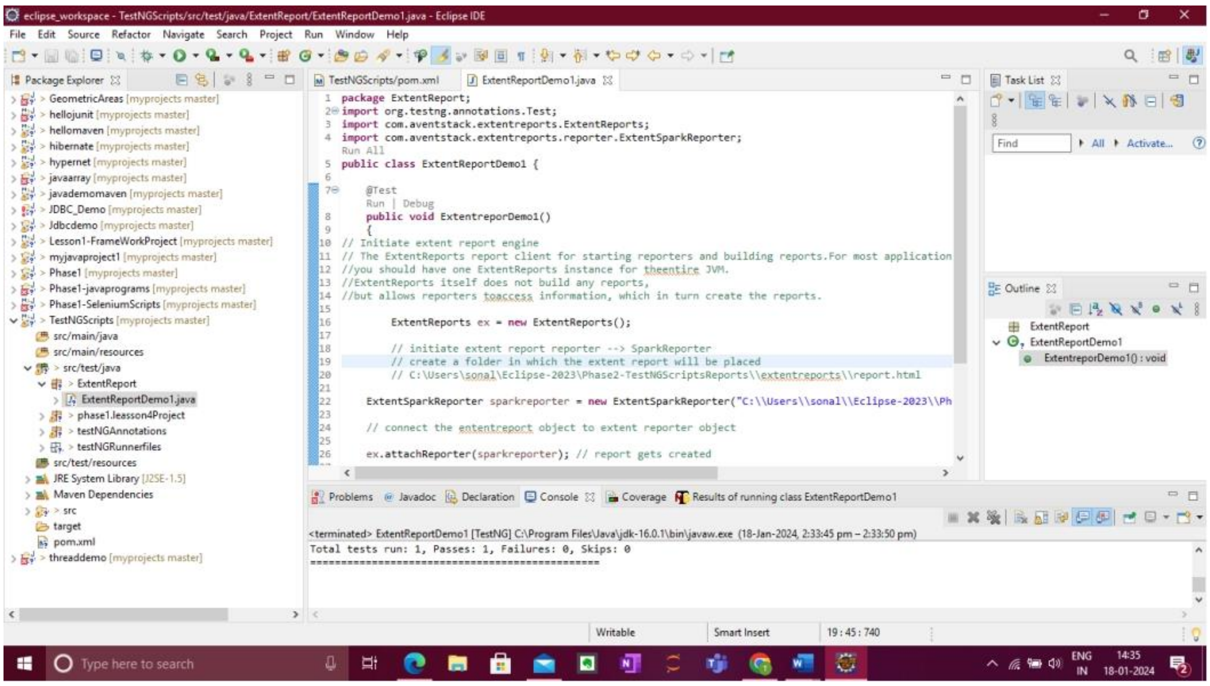
```
package testNGAnnotations;
import org.openqa.selenium.WebDriver;

public class SoftAssertion {

WebDriver driver;

    @BeforeClass
    public void openBrowser()
    {
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().deleteAllCookies();
        driver.get("https://www.selenium.dev/downloads/");
    }


    @Test(priority='1')
    public void gettitlemethod() throws InterruptedException
```

```
<terminated> SoftAssertion [TestNG] C:\Users\Kunnu\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe  (20-Jan

===============================================
    Default test
    Tests run: 1, Failures: 1, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
===============================================
```

## 4.Extent Report :

```
                    package extentReports;
                    import org.testng.annotations.Test;
                    import com.aventstack.extentreports.ExtentReports;
                    import com.aventstack.extentreports.reporter.ExtentSparkReporter;
                    public class ExtentReportDemo1 {
            @Test
                    public void ExtentreporDemo1()
                     {
                  // Initiate extent report engine
                 // The ExtentReports report client for starting reporters and building reports.For most
applications,
                 //you should have one ExtentReports instance for theentire JVM.
                 //ExtentReports itself does not build any reports,
                 //but allows reporters toaccess information, which in turn create the reports.
                 ExtentReports ex = new ExtentReports();
                    // initiate extent report reporter --> SparkReporter
                    // create a folder in which the extent report will be placed
                    // C:\Users\sonal\Eclipse-2023\Phase2-
TestNGScriptsReports\\extentreports\\report.html
                    ExtentSparkReporter sparkreporter = new
ExtentSparkReporter("C:\\Users\\sonal\\Eclipse-2023\\Phase2-
TestNGScriptsReports\\\\extentreports\\\\report.html");
          // connect the ententreport object to extent reporter object
          ex.attachReporter(sparkreporter); // report gets created
```

```
        ex.flush(); // generate the report in the required folder of the project

                }
}
```

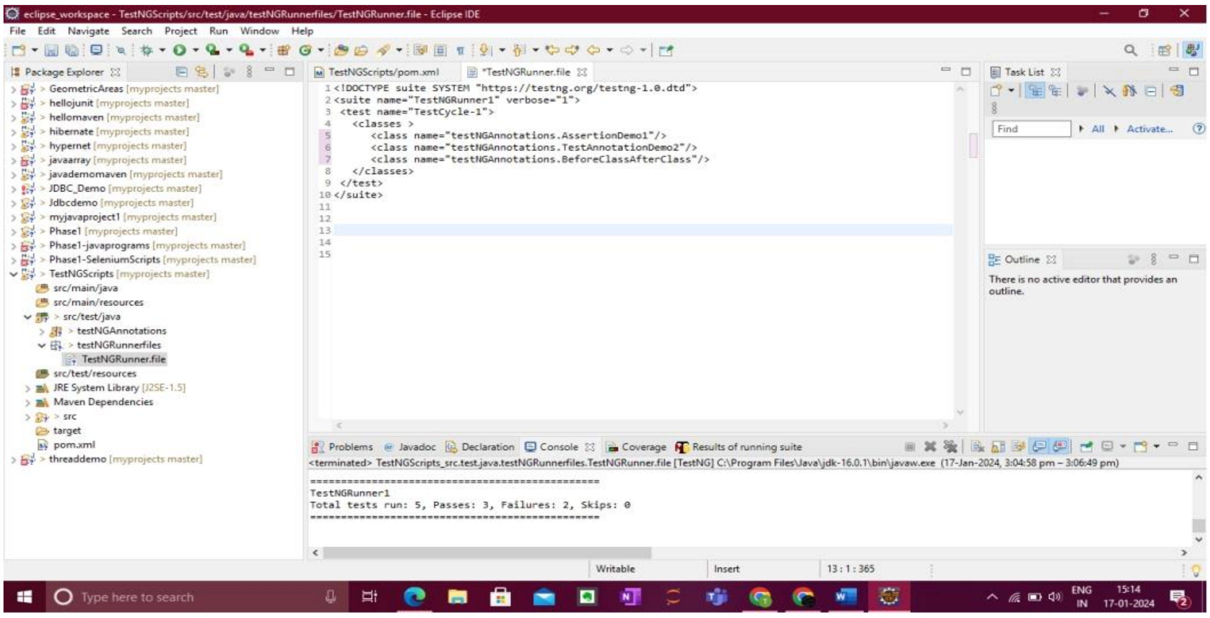


```
package ExtentReport;
import org.testng.annotations.Test;
import com.aventstack.extentreports.ExtentReports;
import com.aventstack.extentreports.reporter.ExtentSparkReporter;
Run All
public class ExtentReportDemo1 {

    @Test
    Run | Debug
    public void ExtentreporDemo1()
    {
// Initiate extent report engine
// The ExtentReports report client for starting reporters and building reports.For most application
//you should have one ExtentReports instance for theentire JVM.
//ExtentReports itself does not build any reports,
//but allows reporters toaccess information, which in turn create the reports.

        ExtentReports ex = new ExtentReports();

        // initiate extent report reporter --> SparkReporter
        // create a folder in which the extent report will be placed
        // C:\Users\sonal\Eclipse-2023\Phase2-TestNGScriptsReports\\extentreports\\report.html

    ExtentSparkReporter sparkreporter = new ExtentSparkReporter("C:\\Users\\sonal\\Eclipse-2023\\Ph

    // connect the ententreport object to extent reporter object

    ex.attachReporter(sparkreporter); // report gets created
```

```
<terminated> ExtentReportDemo1 [TestNG] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (18-Jan-2024, 2:33:45 pm – 2:33:50 pm)
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```

## 7.TestNG parser:

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

            <suite name="TestNGRunner1" verbose="1">

             <test name="TestCycle-1">

           <classes >

             <class name="testNGAnnotations.AssertionDemo1"/>

            <class name="testNGAnnotations.TestAnnotationDemo2"/>

            <class name="testNGAnnotations.BeforeClassAfterClass"/>

            </classes>

            </test>

          </suite>
```



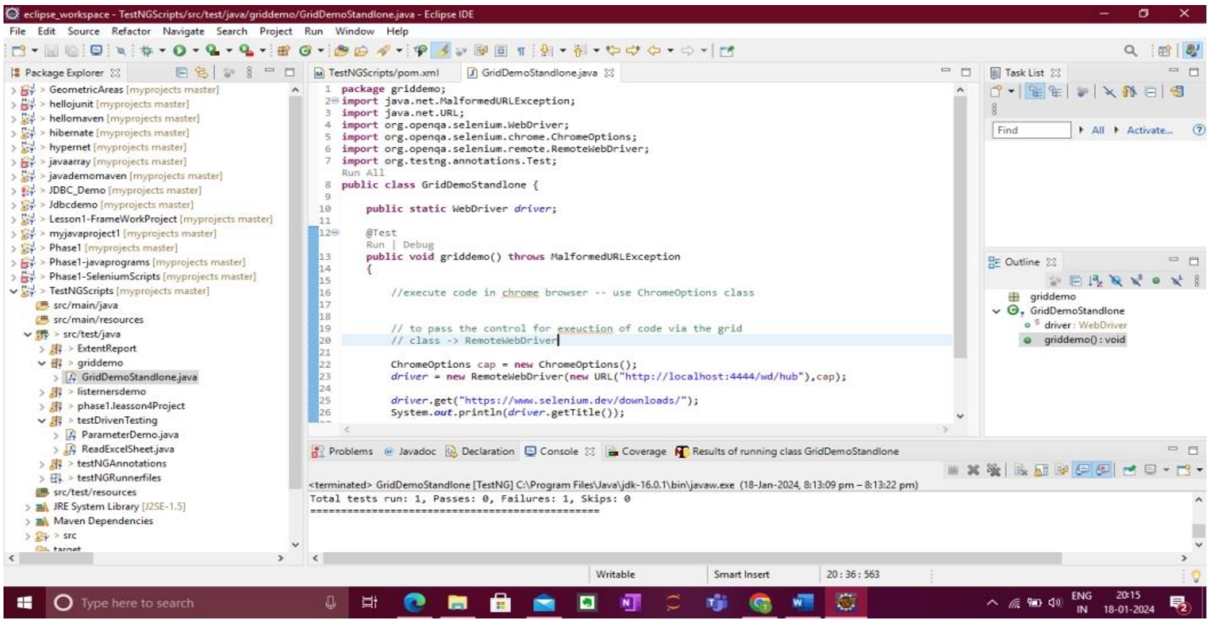
```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="TestNGRunner1" verbose="1">
 <test name="TestCycle-1">
   <classes >
      <class name="testNGAnnotations.AssertionDemo1"/>
      <class name="testNGAnnotations.TestAnnotationDemo2"/>
      <class name="testNGAnnotations.BeforeClassAfterClass"/>
   </classes>
 </test>
</suite>
```

```
<terminated> TestNGScripts_src.test.java.testNGRunnerfiles.TestNGRunner.file [TestNG] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (17-Jan-2024, 3:04:58 pm – 3:06:49 pm)
===============================================
TestNGRunner1
Total tests run: 5, Passes: 3, Failures: 2, Skips: 0
===============================================
```

## 8. Selenium Grid :

```
package griddemo;

    import java.net.MalformedURLException;

    import java.net.URL;

    import org.openqa.selenium.WebDriver;

    import org.openqa.selenium.chrome.ChromeOptions;

    import org.openqa.selenium.remote.RemoteWebDriver;

    import org.testng.annotations.Test;

   public class GridDemoStandlone {

  public static WebDriver driver;

 @Test

public void griddemo() throws MalformedURLException

{

        //execute code in chrome browser -- use ChromeOptions class

        // to pass the control for exeuction of code via the grid
```

```
            // class -> RemoteWebDriver

            ChromeOptions cap = new ChromeOptions();

            driver = new RemoteWebDriver(new
URL("http://localhost:4444/wd/hub"),cap);

            driver.get("https://www.selenium.dev/downloads/");

            System.out.println(driver.getTitle());

            }

}
```



```
package griddemo;
import java.net.MalformedURLException;
import java.net.URL;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.remote.RemoteWebDriver;
import org.testng.annotations.Test;
Run All
public class GridDemoStandlone {

    public static WebDriver driver;

    @Test
    Run | Debug
    public void griddemo() throws MalformedURLException
    {

        //execute code in chrome browser -- use ChromeOptions class


        // to pass the control for exeuction of code via the grid
        // class -> RemoteWebDriver

        ChromeOptions cap = new ChromeOptions();
        driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"),cap);

        driver.get("https://www.selenium.dev/downloads/");
        System.out.println(driver.getTitle());
```

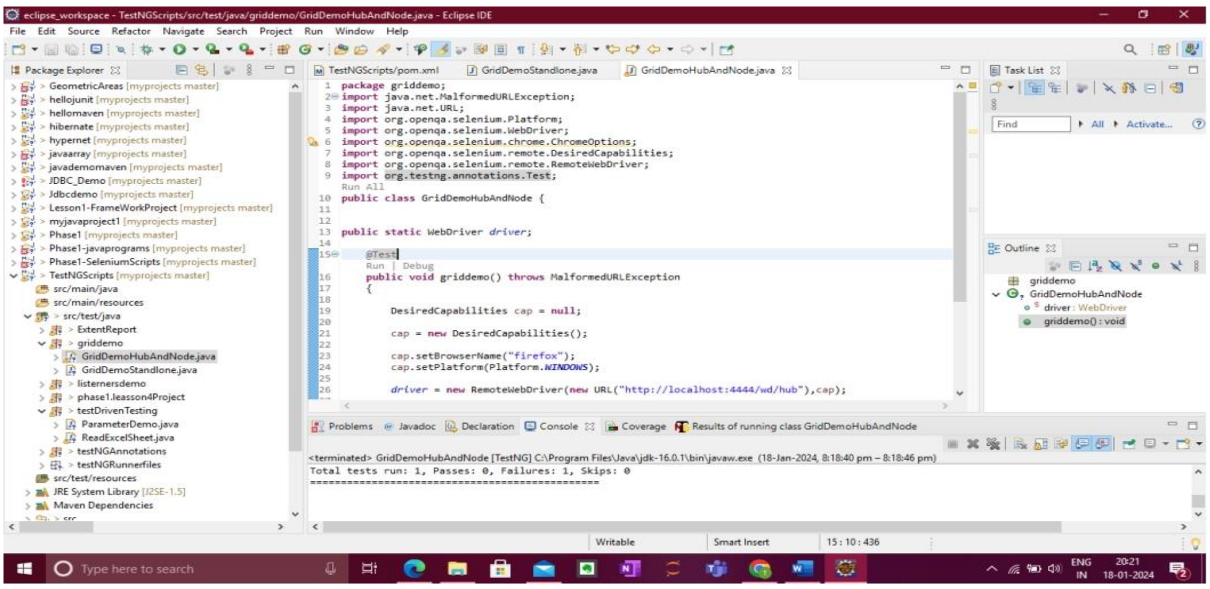
```
<terminated> GridDemoStandlone [TestNG] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (18-Jan-2024, 8:13:09 pm – 8:13:22 pm)
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
===============================================
```

eclipse_workspace - TestNGScripts/src/test/java/griddemo/GridDemoStandlone.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
TestNGScripts/pom.xml
GridDemoStandlone.java
Task List
Outline
Problems Javadoc Declaration Console Coverage Results of running class GridDemoStandlone
Writable
Smart Insert
20 : 36 : 563

## 9. Selenium Grid On Multiple Browsers :

```
package griddemo;

import java.net.MalformedURLException;

import java.net.URL;

import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeOptions;
```

```java
        import org.openqa.selenium.remote.DesiredCapabilities;

        import org.openqa.selenium.remote.RemoteWebDriver;

        import org.testng.annotations.Test;

        public class GridDemoHubAndNode {

        public static WebDriver driver;

      @Test

      public void griddemo() throws MalformedURLException

      {

            DesiredCapabilities cap = null;

            cap = new DesiredCapabilities();

            cap.setBrowserName("firefox");

            cap.setPlatform(Platform.WINDOWS);

           driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"),cap);

            driver.get("https://www.selenium.dev/downloads/");

           System.out.println(driver.getTitle());

            }

}
```



```java
package griddemo;
import java.net.MalformedURLException;
import java.net.URL;
import org.openqa.selenium.Platform;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
import org.testng.annotations.Test;
Run All
public class GridDemoHubAndNode {


public static WebDriver driver;

    @Test
    Run | Debug
    public void griddemo() throws MalformedURLException
    {

        DesiredCapabilities cap = null;

        cap = new DesiredCapabilities();

        cap.setBrowserName("firefox");
        cap.setPlatform(Platform.WINDOWS);

        driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"),cap);
```

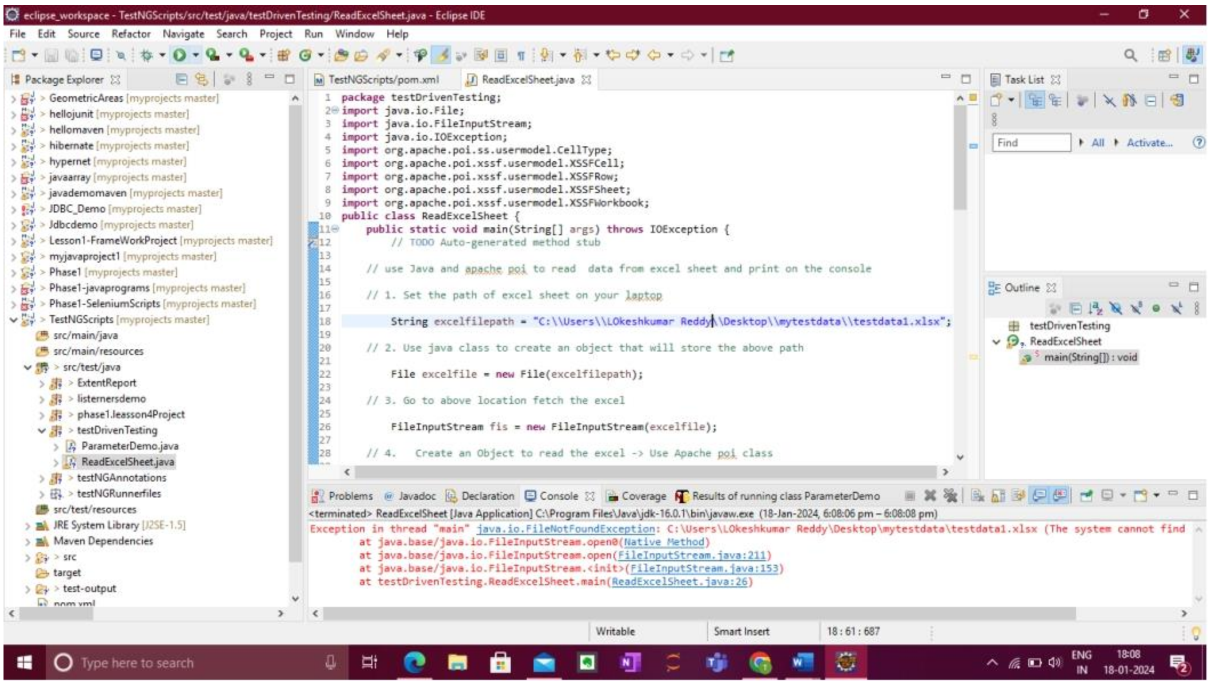
```
<terminated> GridDemoHubAndNode [TestNG] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (18-Jan-2024, 8:18:40 pm – 8:18:46 pm)
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
===============================================
```

## 12. Excel Sheet Read in Selenium :

```
package testDrivenTesting;

  import java.io.File;

  import java.io.FileInputStream;

  import java.io.IOException;

  import org.apache.poi.ss.usermodel.CellType;

  import org.apache.poi.xssf.usermodel.XSSFCell;

  import org.apache.poi.xssf.usermodel.XSSFRow;

  import org.apache.poi.xssf.usermodel.XSSFSheet;

  import org.apache.poi.xssf.usermodel.XSSFWorkbook;

 public class ReadExcelSheet {

 public static void main(String[] args) throws IOException {

   // TODO Auto-generated method stub

   // use Java and apache poi to read data from excel sheet and print on
the console

   // 1. Set the path of excel sheet on your laptop

   String excelfilepath = "C:\\Users\\Lokeshkumar
\\Desktop\\mytestdata\\testdata1.xlsx";

   // 2. Use java class to create an object that will store the above path

   File excelfile = new File(excelfilepath);

   // 3. Go to above location fetch the excel

   FileInputStream fis = new FileInputStream(excelfile);

   // 4.Create an Object to read the excel -> Use Apache poi class

   XSSFWorkbook workbook = new XSSFWorkbook(fis);

   // 5. From the workbook, fetch the sheet
```

```
                XSSFSheet sheet = workbook.getSheet("Sheet1");

                //6. Count the number of rows with data  in the sheet

                int rows= sheet.getLastRowNum();

                System.out.println("Number of rows in the sheet " + rows);

                // 7. Count number of columns with data

                // there is no method to count the number of columns

                // we need to use logic: go to 1st row, count the each cell with data =>
number of columns with data

                int col =        sheet.getRow(1).getLastCellNum();

                System.out.println("Number of columns in the sheet " + col);

                // 8. Go to each row, each column and get the cell data

                // write 2 for loop to go to every row , every cell and get data

                for      (int r =0;r<rows;r++)

        {

                XSSFRow row = sheet.getRow(r);

                // loop to go to each cell of the row

                for(int c=0; c<col;c++)

                {

                        XSSFCell cell = row.getCell(c);

                        CellType celltype = cell.getCellType();

                        switch(celltype)

                        {

                        case STRING:

                        System.out.print(cell.getStringCellValue());

                        break;

                        case NUMERIC:
```

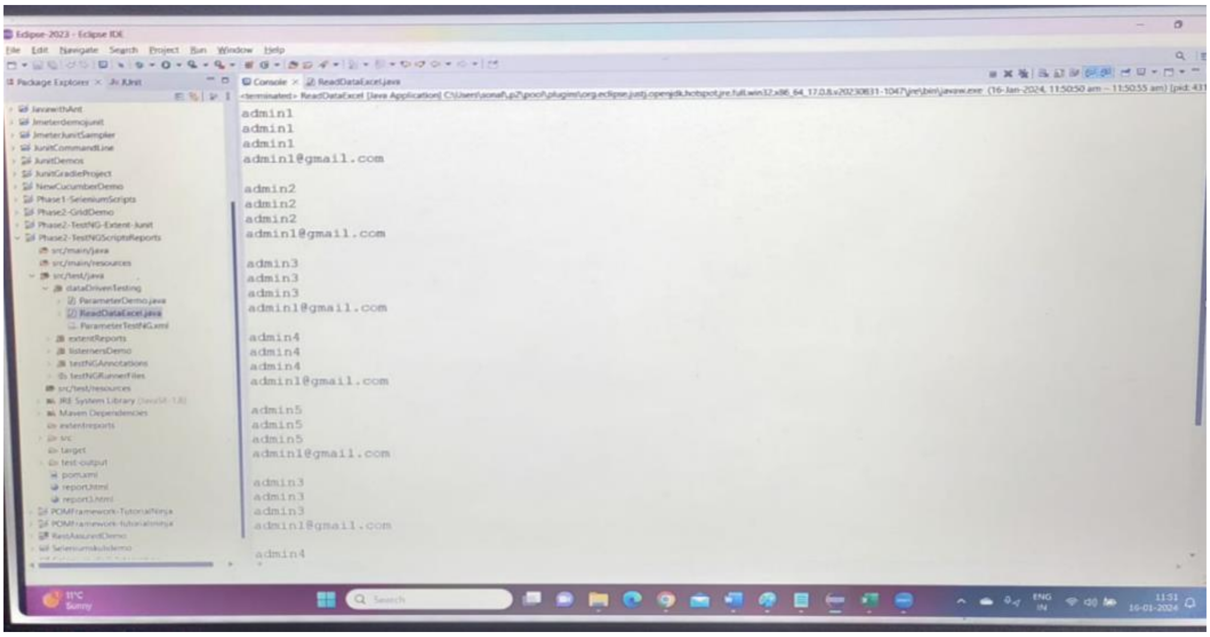
```
                System.out.print(cell.getNumericCellValue());

                break;

            }

                System.out.println(" ");

                }

                  System.out.println("");

            }

                  workbook.close();



            }

}
```



eclipse_workspace - TestNGScripts/src/test/java/testDrivenTesting/ReadExcelSheet.java - Eclipse IDE

```java
package testDrivenTesting;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.xssf.usermodel.XSSFCell;
import org.apache.poi.xssf.usermodel.XSSFRow;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
public class ReadExcelSheet {
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub

    // use Java and apache poi to read  data from excel sheet and print on the console

    // 1. Set the path of excel sheet on your laptop

        String excelfilepath = "C:\\Users\\LOkeshkumar Reddy\\Desktop\\mytestdata\\testdata1.xlsx";

    // 2. Use java class to create an object that will store the above path

        File excelfile = new File(excelfilepath);

    // 3. Go to above location fetch the excel

        FileInputStream fis = new FileInputStream(excelfile);

    // 4.   Create an Object to read the excel -> Use Apache poi class
```

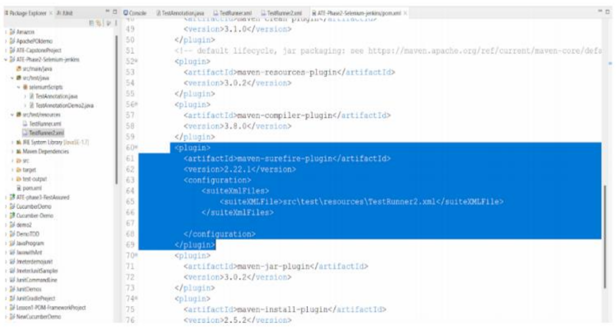
```
<terminated> ReadExcelSheet [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (18-Jan-2024, 6:08:06 pm – 6:08:08 pm)
Exception in thread "main" java.io.FileNotFoundException: C:\Users\LOkeshkumar Reddy\Desktop\mytestdata\testdata1.xlsx (The system cannot find
	at java.base/java.io.FileInputStream.open0(Native Method)
	at java.base/java.io.FileInputStream.open(FileInputStream.java:211)
	at java.base/java.io.FileInputStream.<init>(FileInputStream.java:153)
	at testDrivenTesting.ReadExcelSheet.main(ReadExcelSheet.java:26)
```



```
admin1
admin1
admin1
admin1@gmail.com

admin2
admin2
admin2
admin1@gmail.com

admin3
admin3
admin3
admin1@gmail.com

admin4
admin4
admin4
admin1@gmail.com

admin5
admin5
admin5
admin1@gmail.com

admin3
admin3
admin3
admin1@gmail.com

admin4
```

## 13. Selenium With Maven :

```
<plugin>
   <artifactId>maven-surefire-plugin</artifactId>
     <version>2.22.1</version>
       <configuration>
         <suiteXmlFiles>
            <suiteXMLFile>src\test\resources\TestRunner2.xml</suiteXMLFile>
         </suiteXmlFiles>
      </configuration>

</plugin>
```

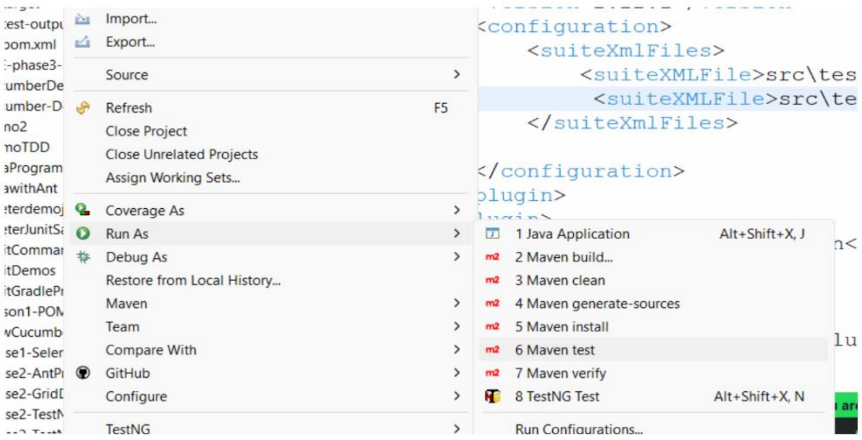


```
        <version>3.1.0</version>
      </plugin>
      <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/defa
      <plugin>
        <artifactId>maven-resources-plugin</artifactId>
        <version>3.0.2</version>
      </plugin>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
      </plugin>
      <plugin>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>2.22.1</version>
        <configuration>
            <suiteXmlFiles>
                <suiteXMLFile>src\test\resources\TestRunner2.xml</suiteXMLFile>
            </suiteXmlFiles>

        </configuration>
      </plugin>
      <plugin>
        <artifactId>maven-jar-plugin</artifactId>
        <version>3.0.2</version>
      </plugin>
      <plugin>
        <artifactId>maven-install-plugin</artifactId>
        <version>2.5.2</version>
```

Save the POM file.

Now execute the maven command

Right click on project → go to run as → go to maven test → code will run

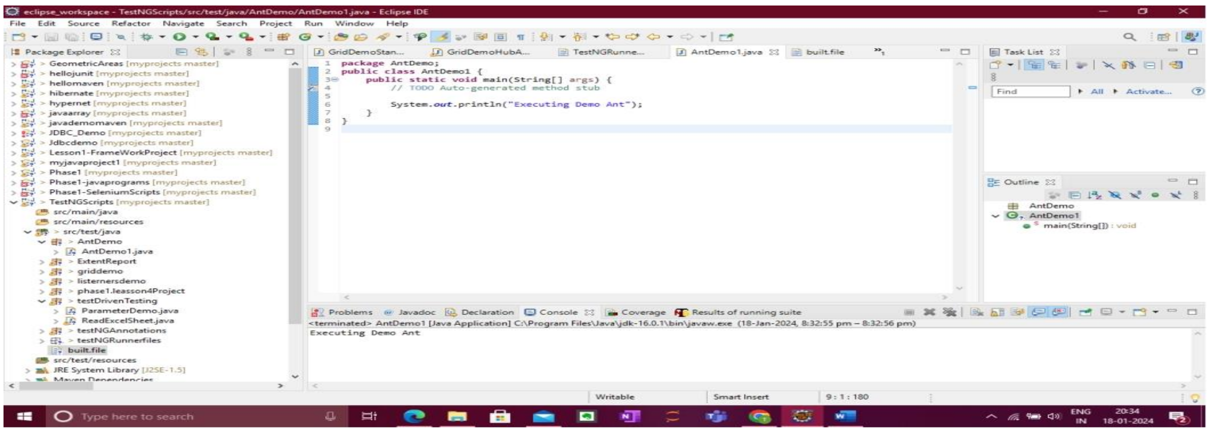


```
<configuration>
    <suiteXmlFiles>
        <suiteXMLFile>src\tes
         <suiteXMLFile>src\te
    </suiteXmlFiles>

</configuration>
plugin>
```

Whenever a maven command is executed, it will generate output files and place it in Target folder of the project

## 14. Selenium With Ant :

```java
package AntDemo;

    public class AntDemo1 {

  public static void main(String[] args) {

        // TODO Auto-generated method stub

        System.out.println("Executing Demo Ant");

    }

}
```

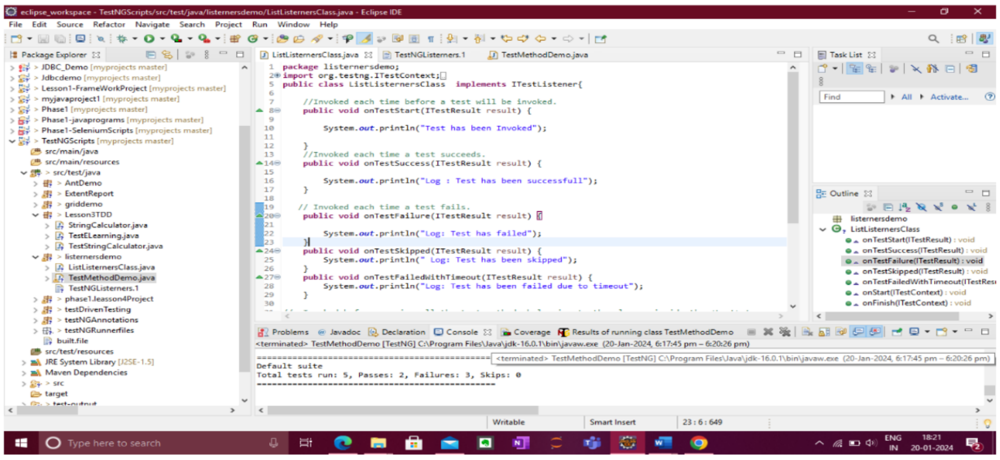


```java
package AntDemo;
public class AntDemo1 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.out.println("Executing Demo Ant");
    }
}
```

```
<terminated> AntDemo1 [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (18-Jan-2024, 8:32:55 pm – 8:32:56 pm)
Executing Demo Ant
```

## 15. Selenium in listerners :

```
package listernersdemo;
import org.testng.ITestContext;
import org.testng.ITestListener;
import org.testng.ITestResult;
public class ListListernersClass implements ITestListener{
//Invoked each time before a test will be invoked.
public void onTestStart(ITestResult result) {
System.out.println("Test has been Invoked");
}
//Invoked each time a test succeeds.
public void onTestSuccess(ITestResult result) {
System.out.println("Log : Test has been successfull");
}
// Invoked each time a test fails.
public void onTestFailure(ITestResult result) {
System.out.println("Log: Test has failed");
}
public void onTestSkipped(ITestResult result) {
System.out.println(" Log: Test has been skipped");
}
public void onTestFailedWithTimeout(ITestResult result) {
System.out.println("Log: Test has been failed due to timeout");
}
// Invoked before running all the test methods belonging to the classes
inside the <test> tag
// and calling all their Configuration methods.
public void onStart(ITestContext context) {
System.out.println("The Main test has started");
}
//Invoked after all the test methods belonging to the classes inside the
<test> tag have run
// and all their Configuration methods have been called.
public void onFinish(ITestContext context) {
System.out.println("The Main test has Completed");
}

}
```



```
package listernersdemo;
import org.testng.ITestContext;
public class ListListernersClass  implements ITestListener{

    //Invoked each time before a test will be invoked.
    public void onTestStart(ITestResult result) {

        System.out.println("Test has been Invoked");

    }
    //Invoked each time a test succeeds.
    public void onTestSuccess(ITestResult result) {

        System.out.println("Log : Test has been successfull");
    }

   // Invoked each time a test fails.
    public void onTestFailure(ITestResult result) {

        System.out.println("Log: Test has failed");
    }
    public void onTestSkipped(ITestResult result) {
        System.out.println(" Log: Test has been skipped");
    }
    public void onTestFailedWithTimeout(ITestResult result) {
        System.out.println("Log: Test has been failed due to timeout");
    }
```

```
<terminated> TestMethodDemo [TestNG] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (20-Jan-2024, 6:17:45 pm – 6:20:26 pm)
===============================================
Default suite
Total tests run: 5, Passes: 2, Failures: 3, Skips: 0
===============================================
```

## 16. Artifactory installed :

Install Jfrog Artifactory in Lab

Use the SL lab machine to install and set up Jfrog Artifactoy

Connect to the lab -> go to the terminal

Execute below commands:

```
================================

# sudo su -

# mkdir myartifactory

# cd myartifactory

# wget https://jfrog.bintray.com/artifactory/jfrog-artifactory-oss-6.9.6.zip

# unzip jfrog-artifactory-oss-6.9.6.zip

# cd jfrog-artifactory-oss-6.9.6

# cd bin

# ./artifactory.sh start
```

Go to your lab browser and give the URL

localhost:8081

You will be on the Jfrog Artifactoy page

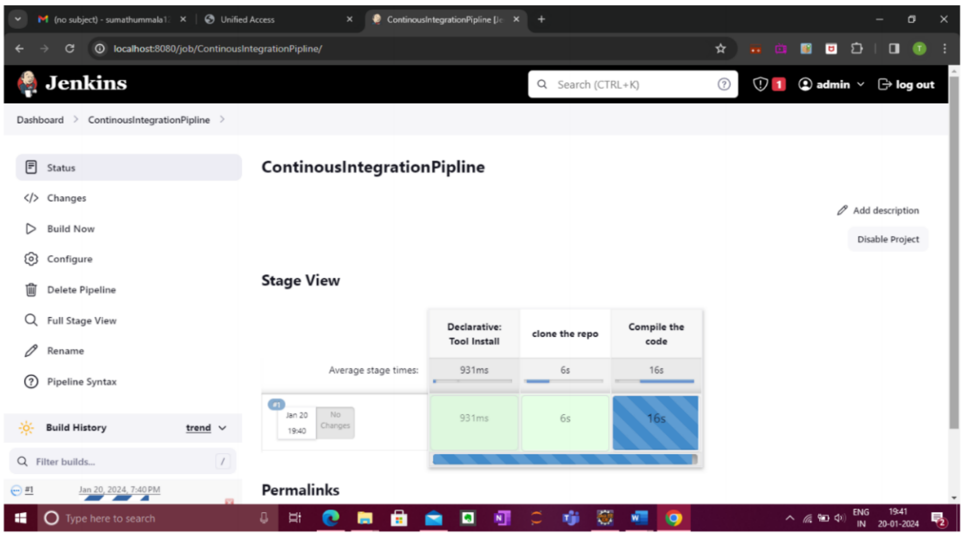
Login with below credentials:

Username: admin

Password: password

## 17. CI/CD pipline With Maven :

```
pipeline{
tools{
maven 'mymaven'
}
// where to run the pipeline // agent means a server/virtual machine
// any here means--current windows server
agent any
// In pipline we want to exeucte many jobs -> called stages
// pipeline = set of stages/set of task
stages{
// each task/job represents a stage
stage('Clone the repo'){
steps{
git 'https://github.com/Sonal0409/ATE_Phase2-Selenium-Jenkins-
Jan24.git'
}
}
stage('Execute the tests'){
steps{
bat 'mvn test'
//bat : you are running the command using windows command
line(batch)
}
}
}

}
```



ContinousIntegrationPipline

Stage View

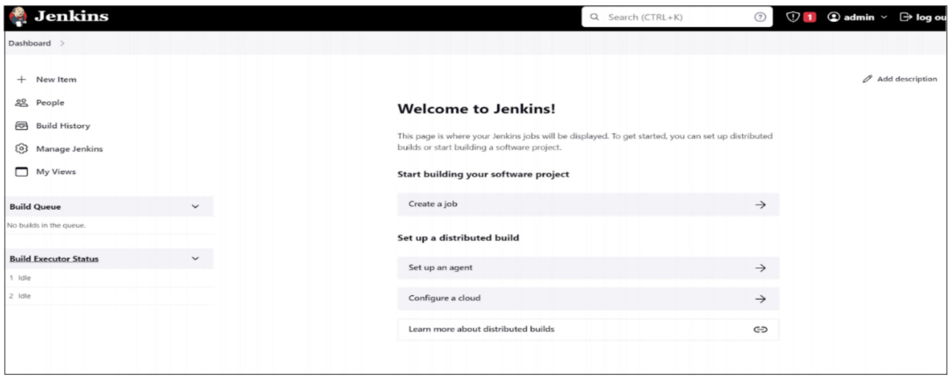
| | Declarative: Tool Install | clone the repo | Compile the code |
|---|---|---|---|
| Average stage times: | 931ms | 6s | 16s |
| #1 Jan 20 19:40 No Changes | 931ms | 6s | 16s |

## 18. CI/CD pipline With Selenium Webdriver :

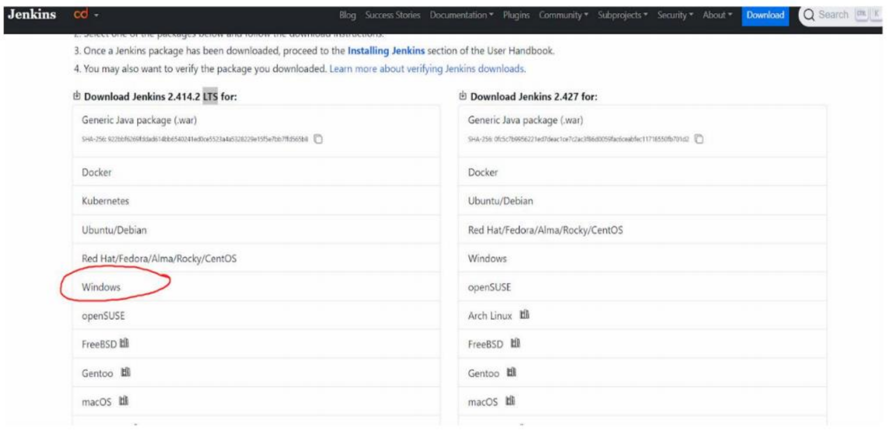
```
pipeline{
tools{
maven 'mymaven'
}
// where to run the pipeline
// agent means a server/virtual machine
// any here means--current windows server
agent any
// In pipline we want to exeucte many jobs -> called stages
// pipeline = set of stages/set of task
stages{
// each task/job represents a stage
stage('Clone the repo'){
steps{
git 'https://github.com/Sonal0409/ATE_Phase2-Selenium-Jenkins-
Jan24.git'
}
}
stage('Execute the tests'){
steps{
bat 'mvn test'
//bat : you are running the command using windows command
line(batch)
}
}
}

}
```

## 19. Selenium integration with Jenkins :

we will install the LTS version. Click on Windows

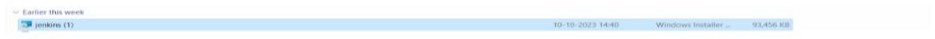

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

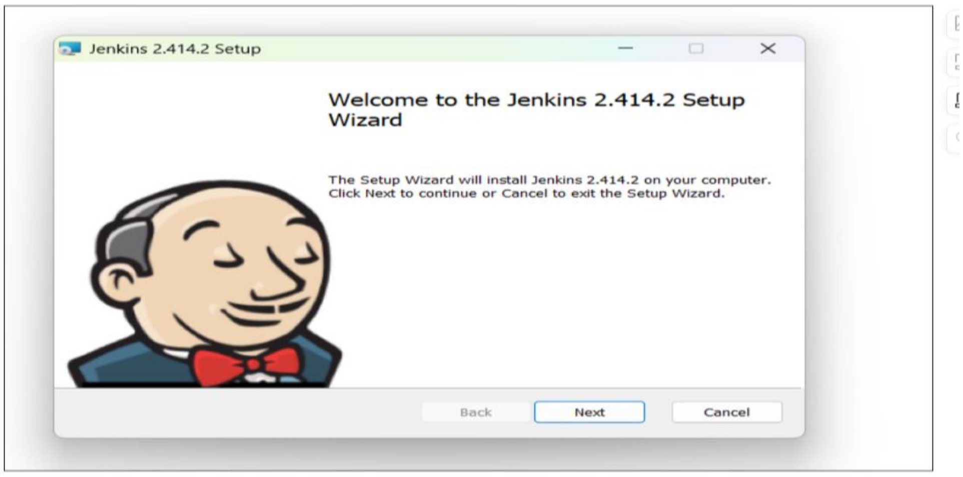

3. Once a Jenkins package has been downloaded, proceed to the **Installing Jenkins** section of the User Handbook.
4. You may also want to verify the package you downloaded. Learn more about verifying Jenkins downloads.

| **Download Jenkins 2.414.2 LTS for:** | **Download Jenkins 2.427 for:** |
|---|---|
| Generic Java package (.war) SHA-256: 922bbf6269fddad614bb6540241ed0ce5523a4a5328229e15f5e7bb7ffd565b8 | Generic Java package (.war) SHA-256: 0fc5c7b9956221ed7deac1ce7c2ac3f86d0059facfceabfec11718550fb701d2 |
| Docker | Docker |
| Kubernetes | Ubuntu/Debian |
| Ubuntu/Debian | Red Hat/Fedora/Alma/Rocky/CentOS |
| Red Hat/Fedora/Alma/Rocky/CentOS | Windows |
| Windows | openSUSE |
| openSUSE | Arch Linux |
| FreeBSD | FreeBSD |
| Gentoo | Gentoo |
| macOS | macOS |

As you will click on Windows, it will automatically install Jenkins in your downloads folder

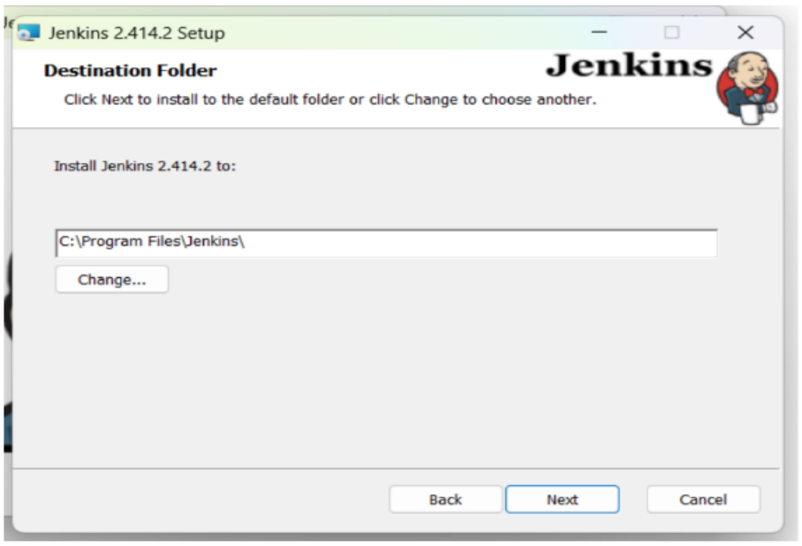

Earlier this week
jenkins (1)    10-10-2023 14:40    Windows Installer ...    93,456 KB

Double click on Jenkins Windows installer to start the instillation.


Jenkins 2.414.2 Setup

Welcome to the Jenkins 2.414.2 Setup Wizard

The Setup Wizard will install Jenkins 2.414.2 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

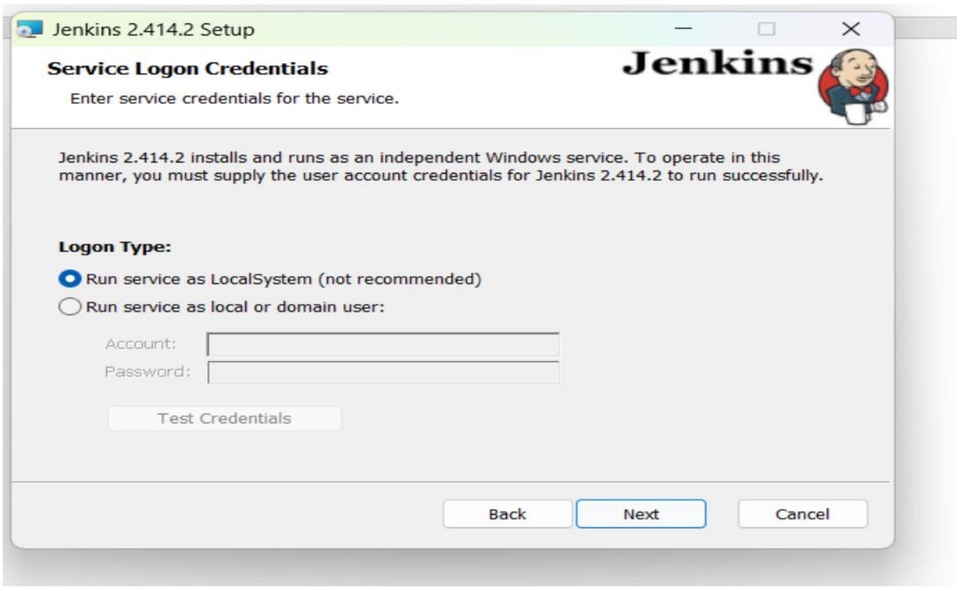

Jenkins 2.414.2 Setup

**Destination Folder**

Click Next to install to the default folder or click Change to choose another.

Install Jenkins 2.414.2 to:

C:\Program Files\Jenkins\


Jenkins 2.414.2 Setup

**Service Logon Credentials**

Enter service credentials for the service.

Jenkins 2.414.2 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.414.2 to run successfully.

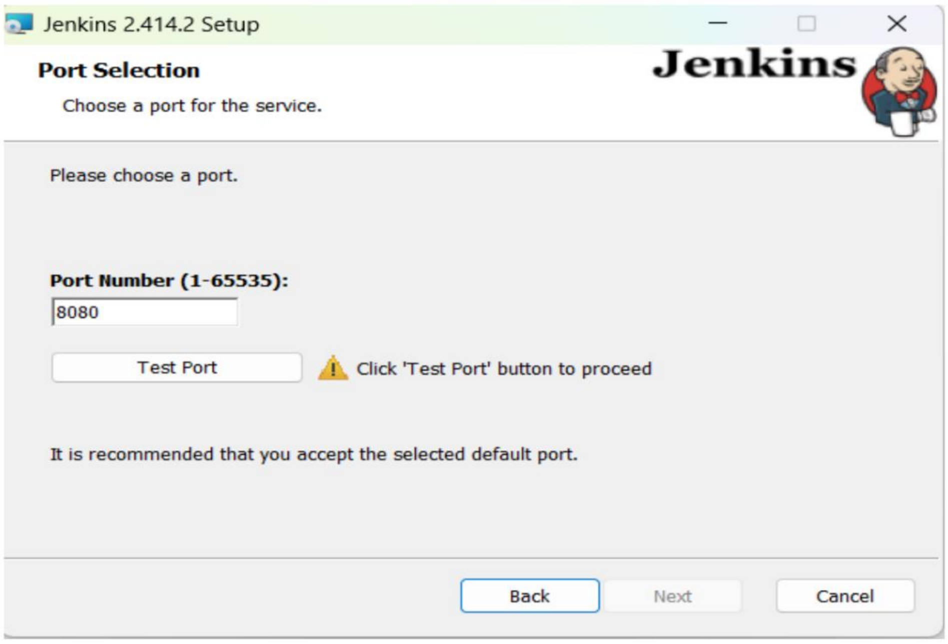
**Logon Type:**

[✓ Run service as LocalSystem (not recommended)]

Run service as local or domain user:

Account:

Password:


Jenkins 2.414.2 Setup

**Port Selection**

Choose a port for the service.

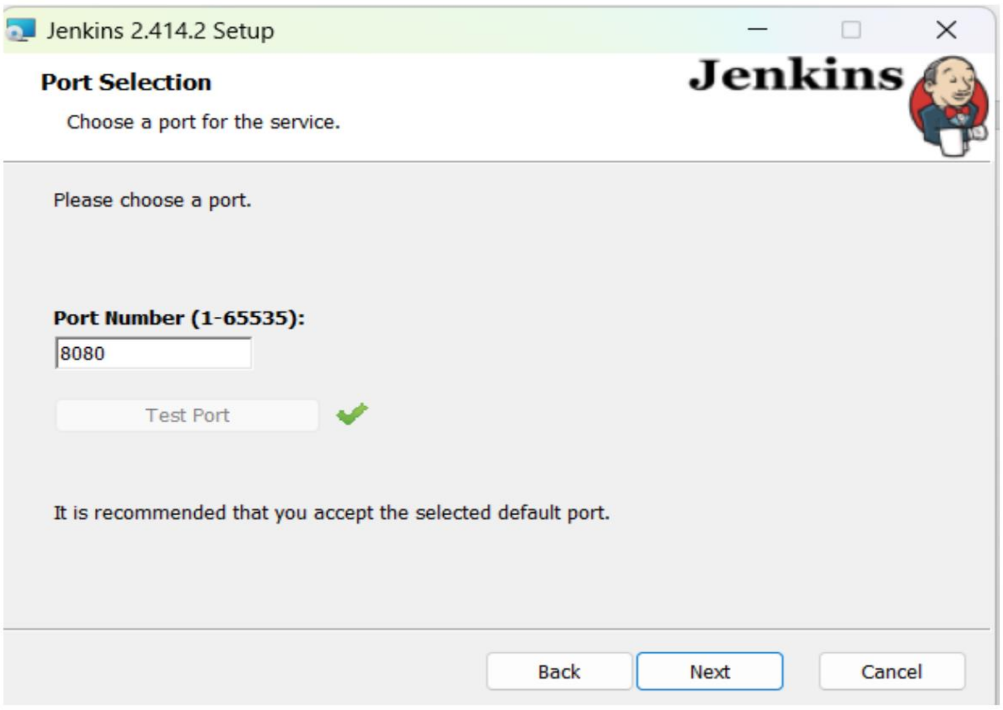
Please choose a port.

**Port Number (1-65535):**

8080

Click 'Test Port' button to proceed

It is recommended that you accept the selected default port.



**Port Selection**

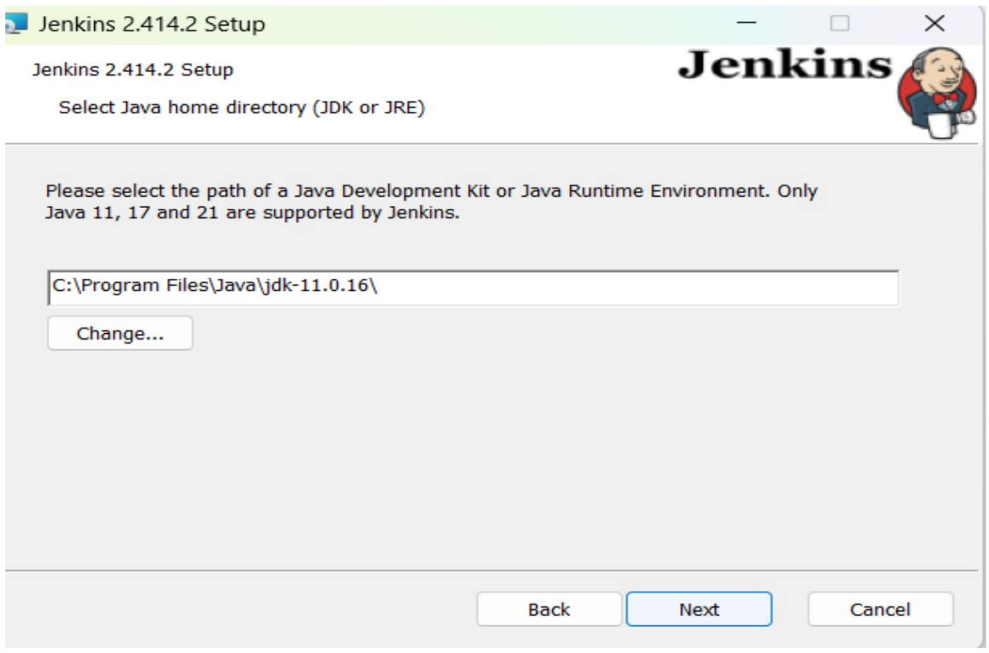
Choose a port for the service.

Please choose a port.

**Port Number (1-65535):**

8080

It is recommended that you accept the selected default port.

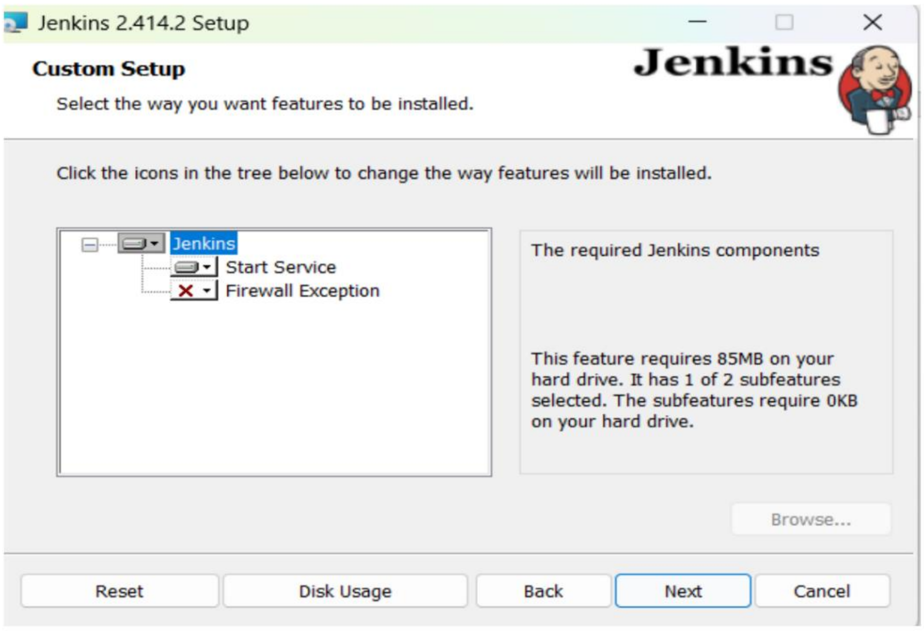


Jenkins 2.414.2 Setup

Select Java home directory (JDK or JRE)

Please select the path of a Java Development Kit or Java Runtime Environment. Only Java 11, 17 and 21 are supported by Jenkins.

C:\Program Files\Java\jdk-11.0.16\


Jenkins 2.414.2 Setup

**Custom Setup**

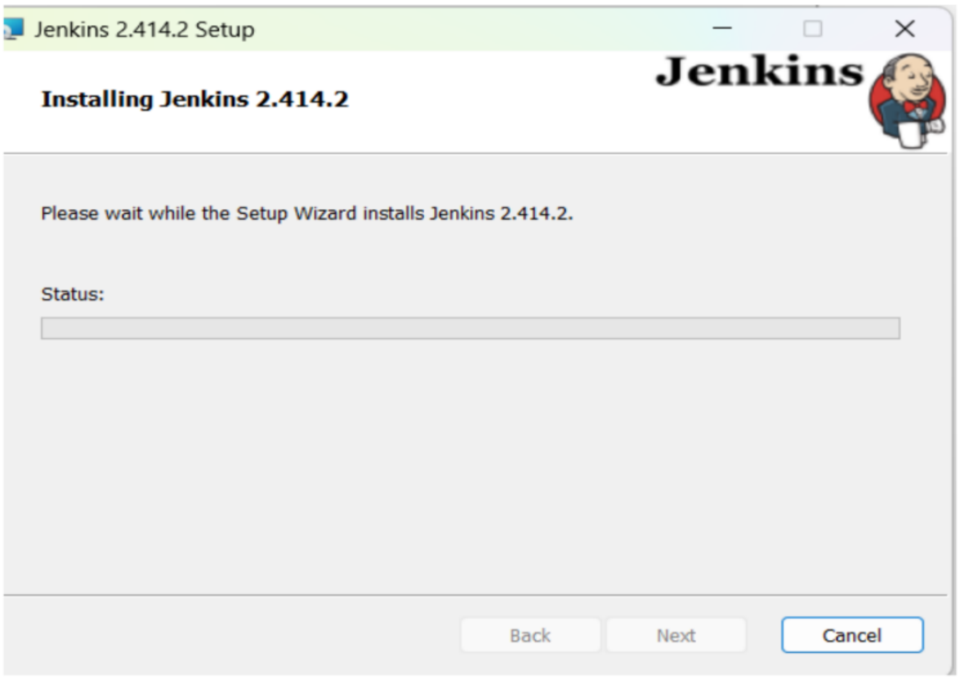
Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- Jenkins
  - Start Service
  - Firewall Exception

The required Jenkins components

This feature requires 85MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.

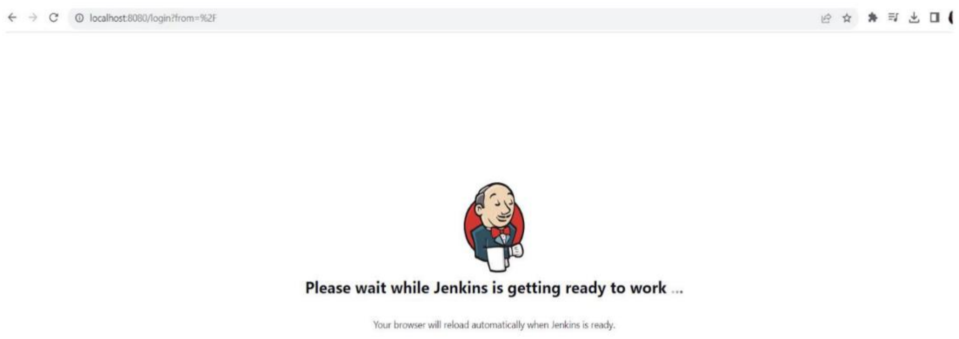

Jenkins 2.414.2 Setup

**Installing Jenkins 2.414.2**

Please wait while the Setup Wizard installs Jenkins 2.414.2.

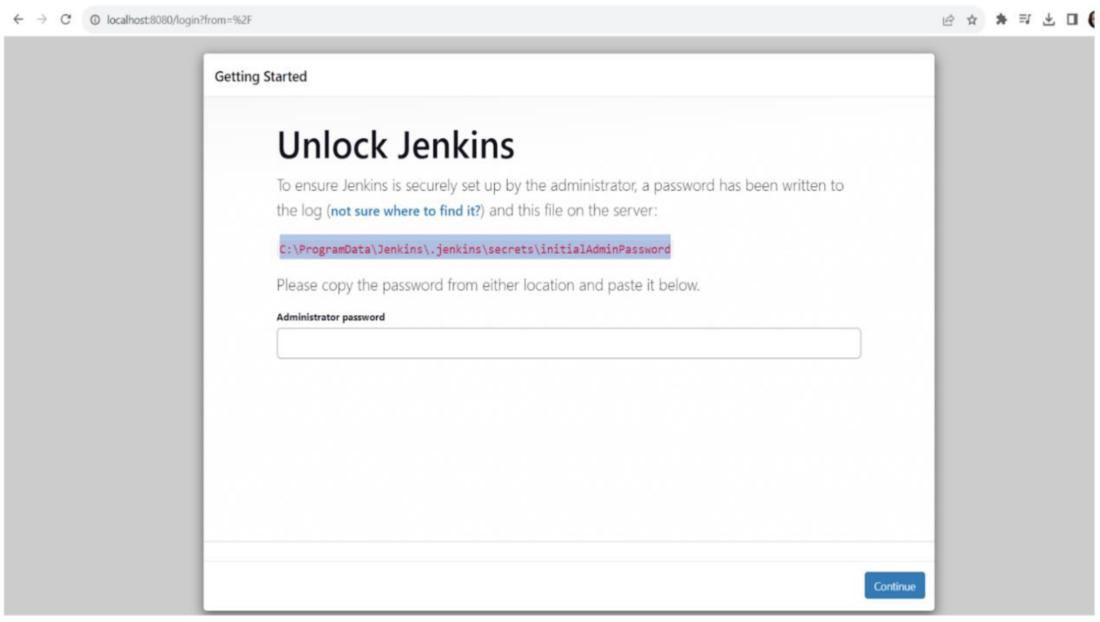
Status:

Installation will complete in sometime


localhost:8080/login?from=%2F

Please wait while Jenkins is getting ready to work ...

Your browser will reload automatically when Jenkins is ready.


localhost:8080/login?from=%2F

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:

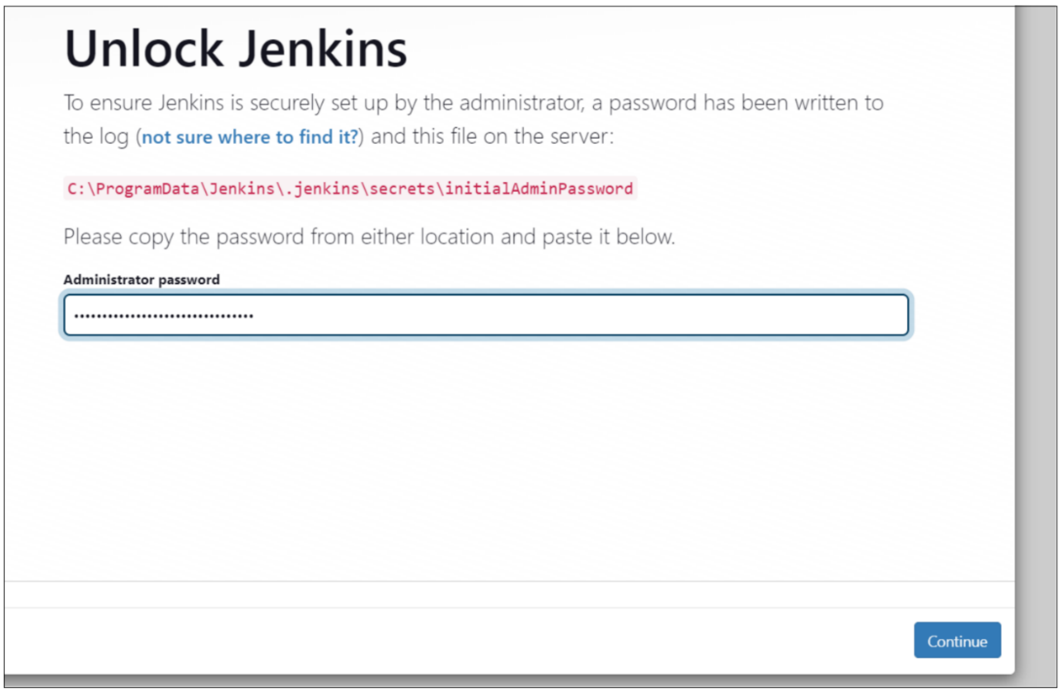
```
C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword
```

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

Administrator password

Go to this path: C:\ProgramData\Jenkins\.jenkins\secrets
You will get a file: initialAdminPassword

Open it with notepad => you will get a password copy it and paste on browser



Unlock Jenkins

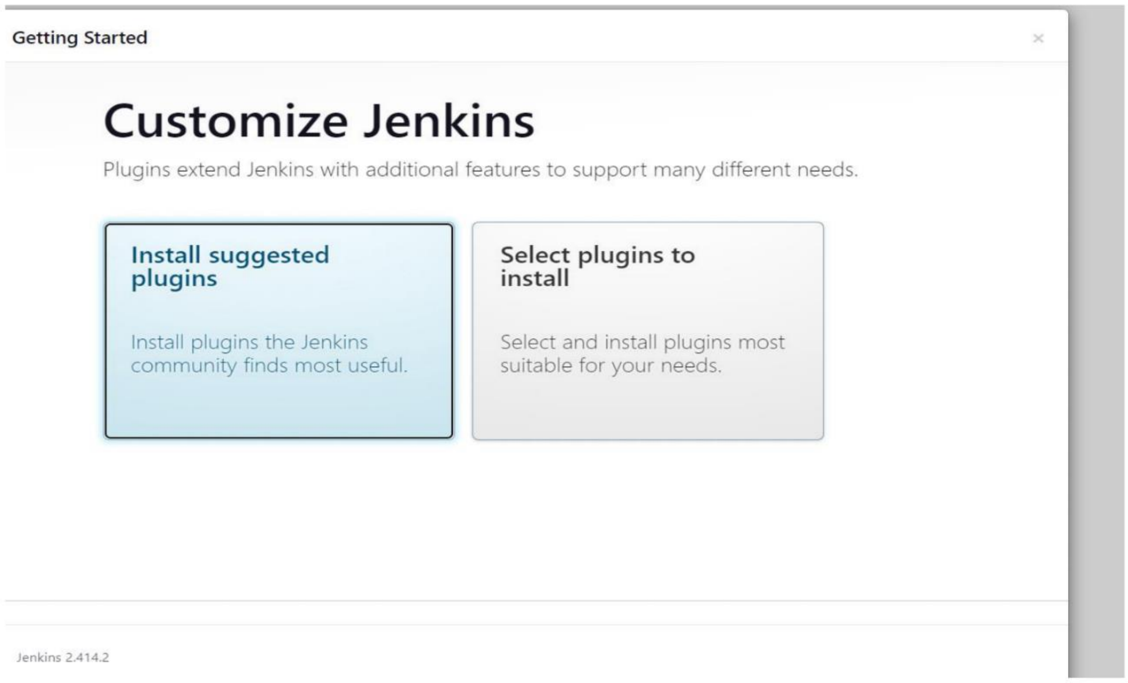
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword
```

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

Administrator password

································



Getting Started

Customize Jenkins

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

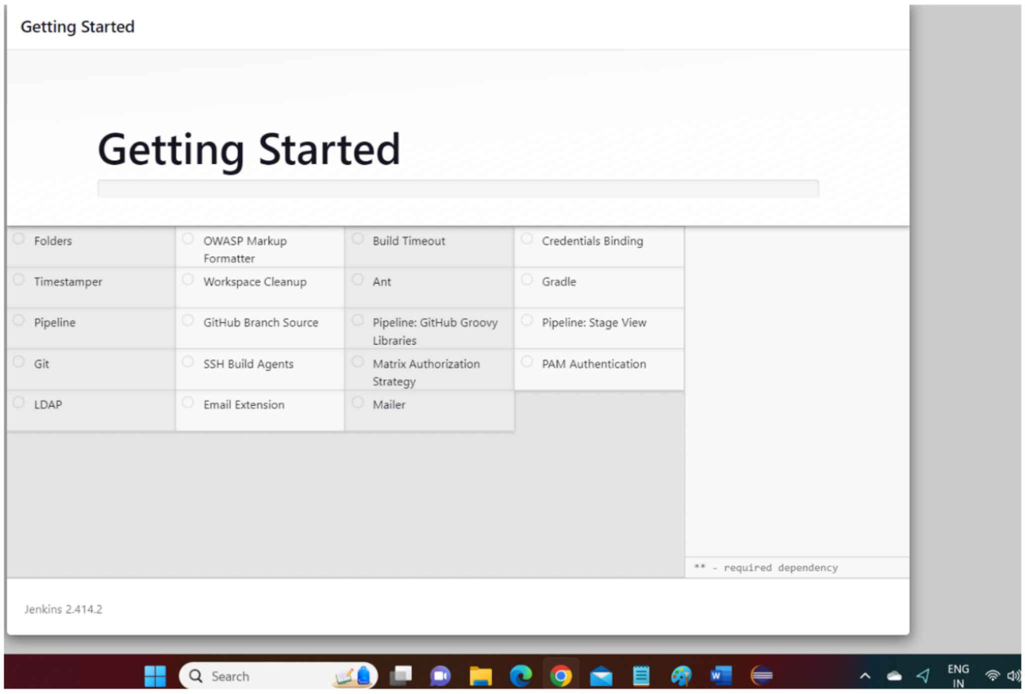
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.414.2

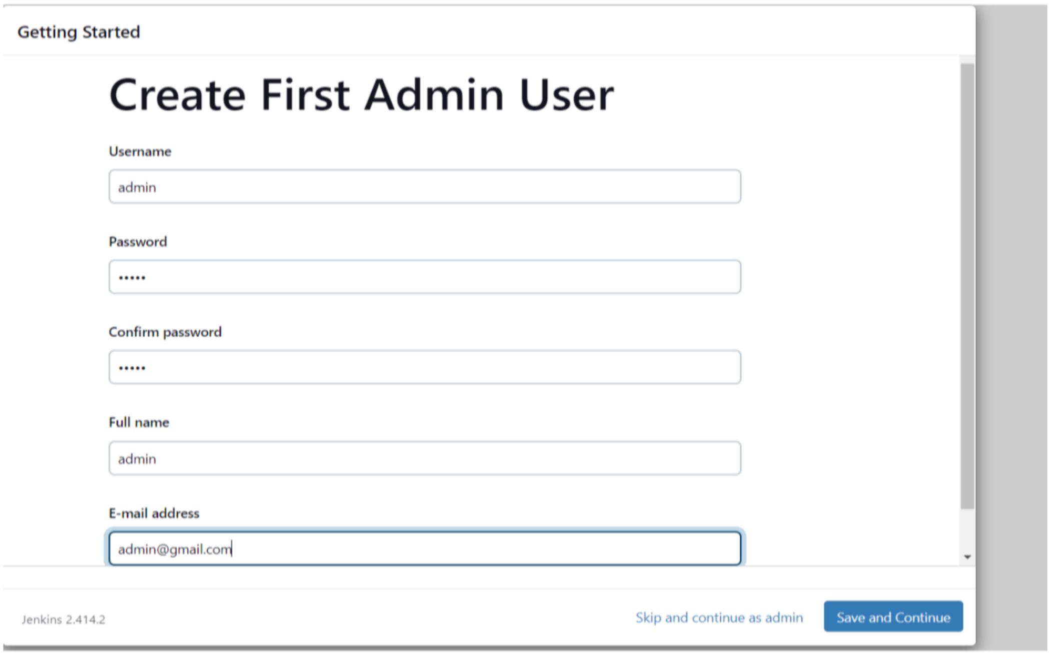

Getting Started

Getting Started

| | | | |
|---|---|---|---|
| Folders | OWASP Markup Formatter | Build Timeout | Credentials Binding |
| Timestamper | Workspace Cleanup | Ant | Gradle |
| Pipeline | GitHub Branch Source | Pipeline: GitHub Groovy Libraries | Pipeline: Stage View |
| Git | SSH Build Agents | Matrix Authorization Strategy | PAM Authentication |
| LDAP | Email Extension | Mailer | |

** - required dependency

Jenkins 2.414.2


Getting Started

Create First Admin User

Username: admin

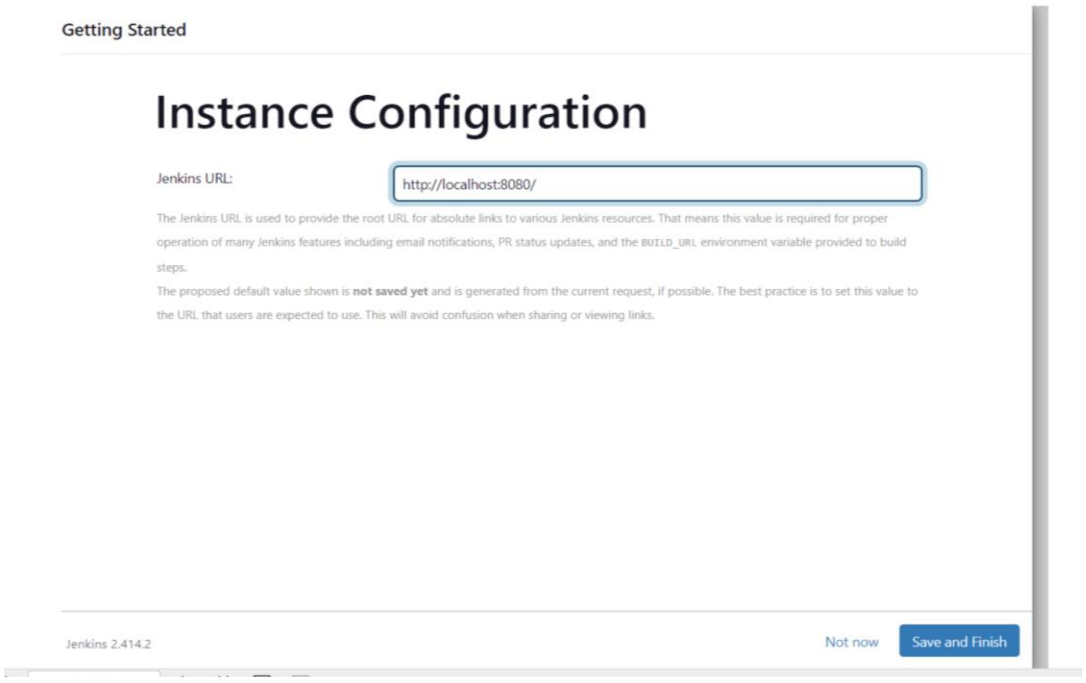
Password: •••••

Confirm password: •••••

Full name: admin

E-mail address: admin@gmail.com

Jenkins 2.414.2


Getting Started

Instance Configuration

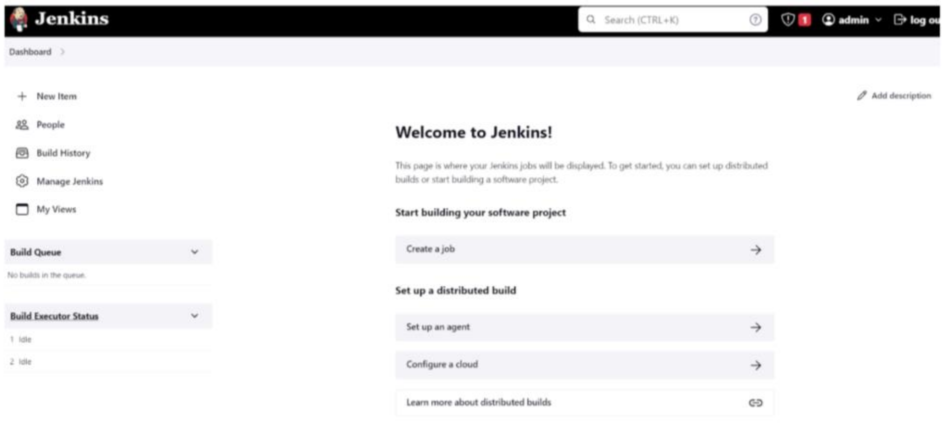
Jenkins URL: http://localhost:8080/

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 2.414.2

Click on save and finish.



No builds in the queue.

1 Idle

2 Idle

Welcome to Jenkins!

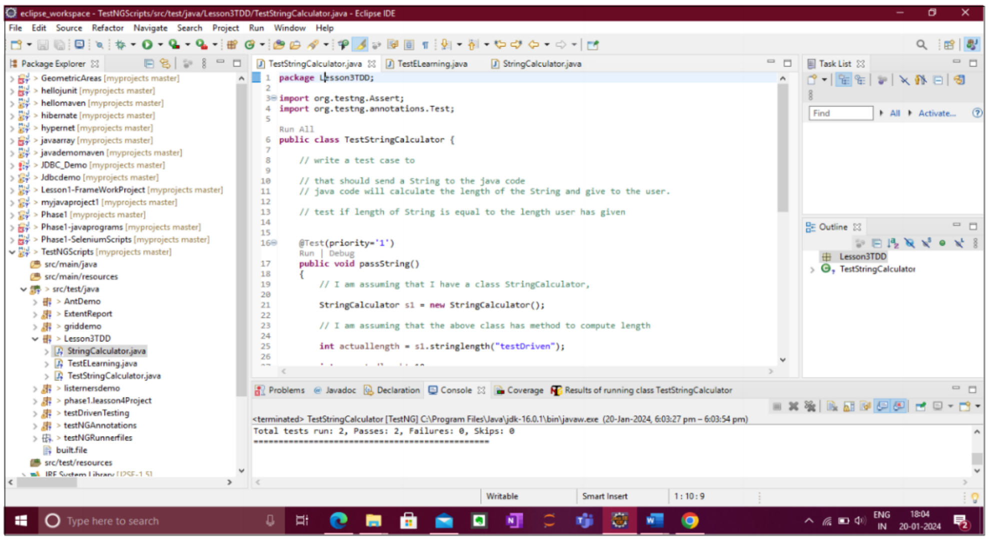
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

## 20. TDD With TestNG :

```
 package Lesson3TDD;
import org.testng.Assert;
import org.testng.annotations.Test;
public class TestStringCalculator {
// write a test case to
// that should send a String to the java code
// java code will calculate the length of the String and give to the user.
// test if length of String is equal to the length user has given
@Test(priority='1')
public void passString()
{
// I am assuming that I have a class StringCalculator,
StringCalculator s1 = new StringCalculator();
// I am assuming that the above class has method to compute length
int actuallength = s1.stringlength("testDriven");
int expectedlenght=10;
// using testNg assertion I am comparing the length of the string
Assert.assertEquals(actuallength, expectedlenght);
}
// The calculator should be able to add 2 strings
@Test(priority='2')
public void TestaddString()
{
// I am assuming that I have a class StringCalculator,
StringCalculator str = new StringCalculator();
// I am assuming that the above class has method to concatinate 2
strings
String actualString= str.addstring("selenium","tool");
String expectedString = "SELENIUMTOOL";
Assert.assertEquals(actualString,expectedString);
}

}
```



eclipse_workspace - TestNGScripts/src/test/java/Lesson3TDD/TestStringCalculator.java - Eclipse IDE

```
package Lesson3TDD;

import org.testng.Assert;
import org.testng.annotations.Test;

Run All
public class TestStringCalculator {

    // write a test case to

    // that should send a String to the java code
    // java code will calculate the length of the String and give to the user.

    // test if length of String is equal to the length user has given


    @Test(priority='1')
    Run | Debug
    public void passString()
    {
        // I am assuming that I have a class StringCalculator,

        StringCalculator s1 = new StringCalculator();

        // I am assuming that the above class has method to compute length

        int actuallength = s1.stringlength("testDriven");
```

```
<terminated> TestStringCalculator [TestNG] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (20-Jan-2024, 6:03:27 pm – 6:03:54 pm)
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================
```