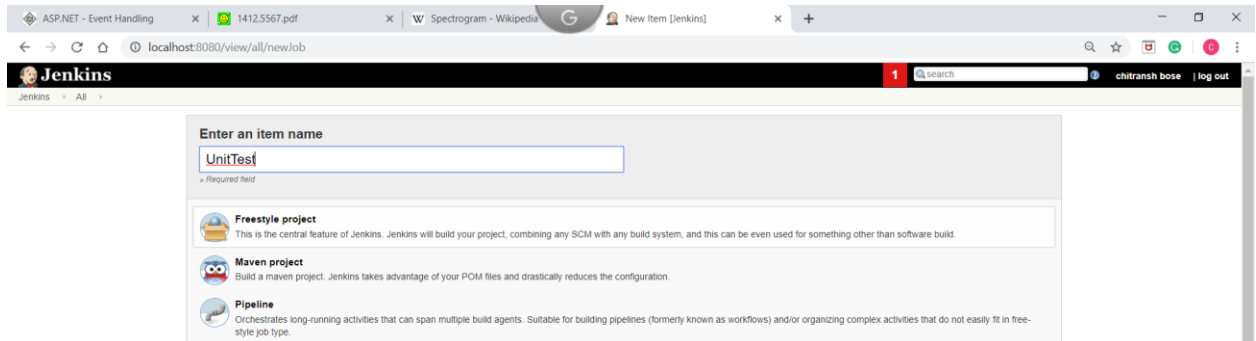
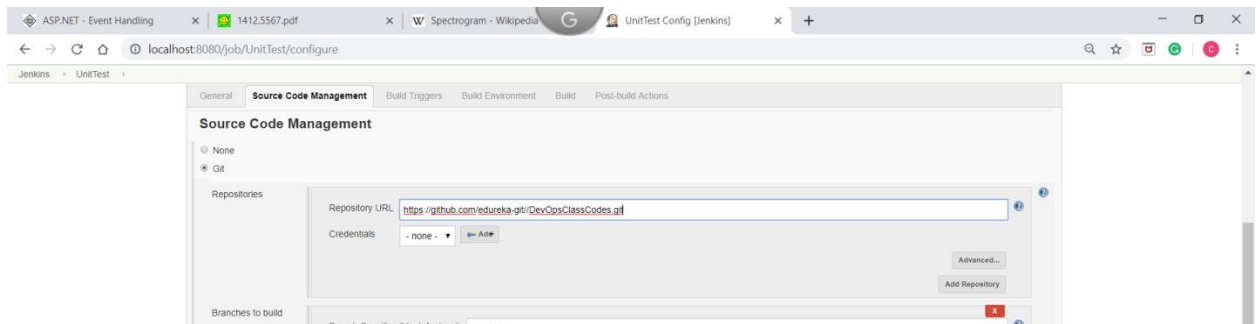


Task 2. I

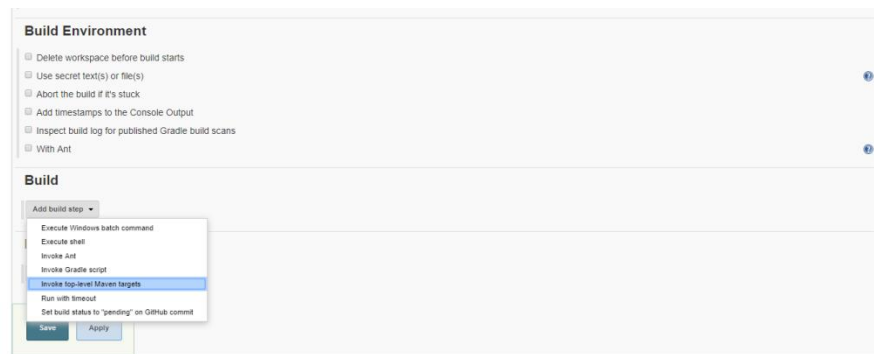
1. Create a freestyle job UnitTest.



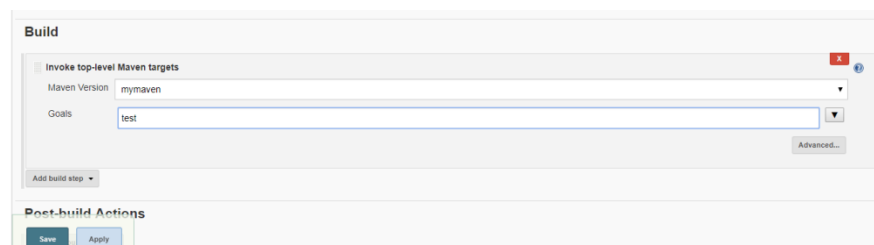
2. In configure, goto source code management and click on git and enter the url given.



3. GOTO build, then add build step and select top level maven targets.



4. Choose maven version as mymaven and goals as test.



- Build it after saving and check the console output.

```

Causeat com.edurekademo.utilities.ExceptionThrower.doYvY(ExceptionThrower.java:35)
Causeat com.edurekademo.utilities.ExceptionThrower.doXooX(ExceptionThrower.java:23)
[pool-1-thread-7] INFO com.edurekademo.utilities.TestLogger - java.lang.Exception: TEST MESSAGE
Causeat com.edurekademo.utilities.ExceptionThrower.doXooX(ExceptionThrower.java:43)
Causeat com.edurekademo.utilities.TestLogger.testGetErrorMessages(TestLogger.java:75)
**** java.lang.Exception: TEST MESSAGE
Causeat com.edurekademo.utilities.ExceptionThrower.doXooX(ExceptionThrower.java:43)
Causeat com.edurekademo.utilities.TestLogger.testGetErrorMessages(TestLogger.java:75)
[pool-1-thread-8] INFO com.edurekademo.utilities.TestLogger - java.io.IOException: TESTIOEXCEPTION
Causeat com.edurekademo.utilities.ExceptionThrower.doNothing(ExceptionThrower.java:18)
Causeat com.edurekademo.utilities.TestLogger.testGetErrorMessages(TestLogger.java:68)
[pool-1-thread-9] INFO com.edurekademo.utilities.TestLogger - java.lang.ArithmeticException: / by zero
Causeat com.edurekademo.utilities.TestLogger.testGetErrorMessages(TestLogger.java:30)
Tests run: sun.reflect.NativeMethodAccessorImpl.invoke(Native Method)
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.25 sec - in com.edurekademo.utilities.TestLogger
Running com.edurekademo.utilities.TestGenericComparator
Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.019 sec - in com.edurekademo.utilities.TestGenericComparator

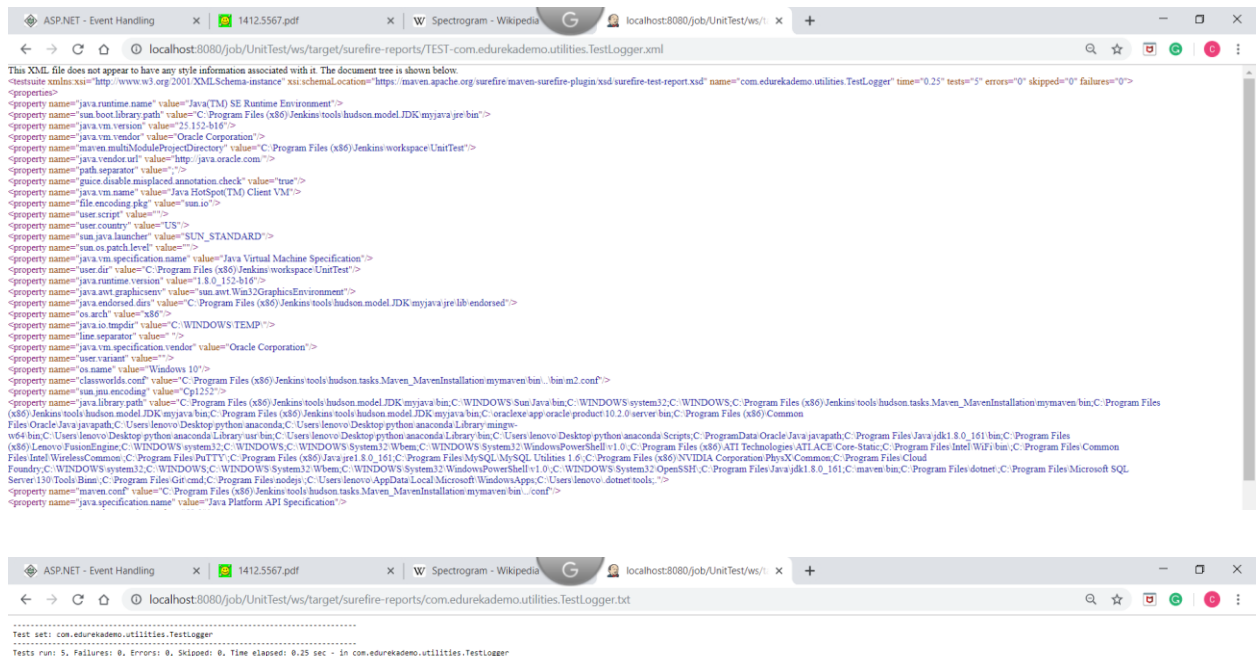
Results :

Tests run: 23, Failures: 0, Errors: 0, Skipped: 0

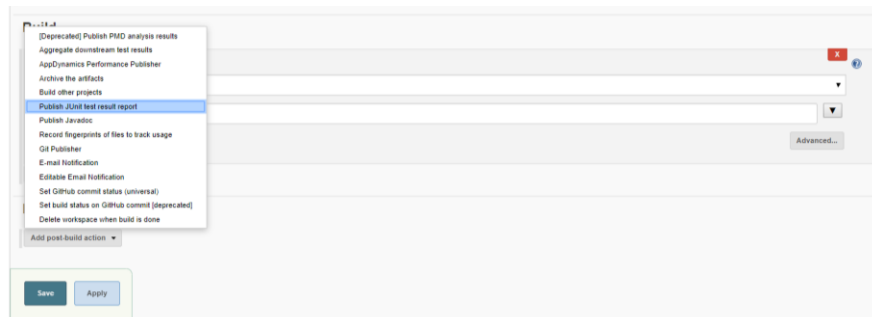
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 58.977 s
[INFO] Finished at: 2019-04-15T16:03:01+05:30
[INFO] -----
Finished: SUCCESS

```

6. Check the detailed report at `workspace/targets/surefire-reports`.



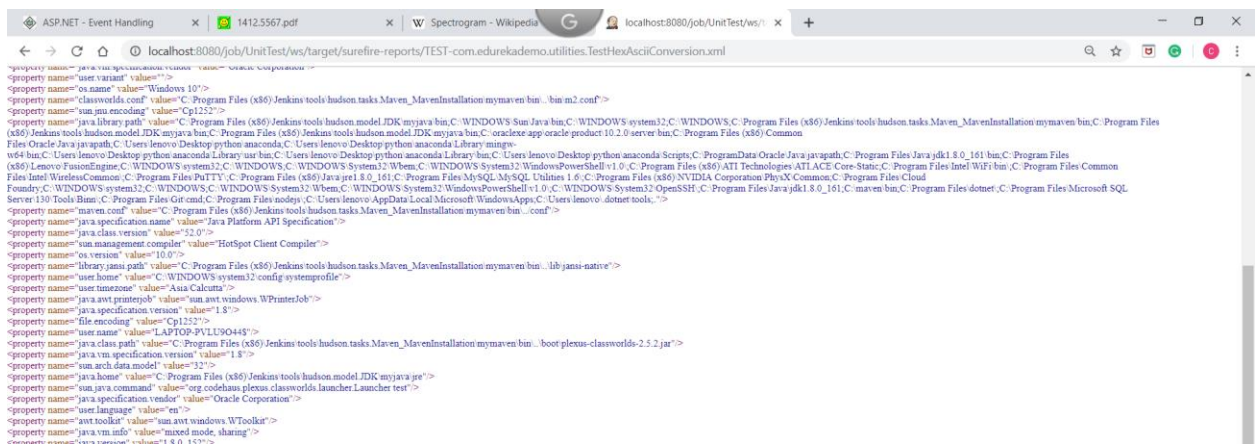
7. Configure the job and go to Post Build Actions and add post-build action as Publish Junit test result report.



8. Give the path in Test report XMLs as target/surefire-reports/*.xml

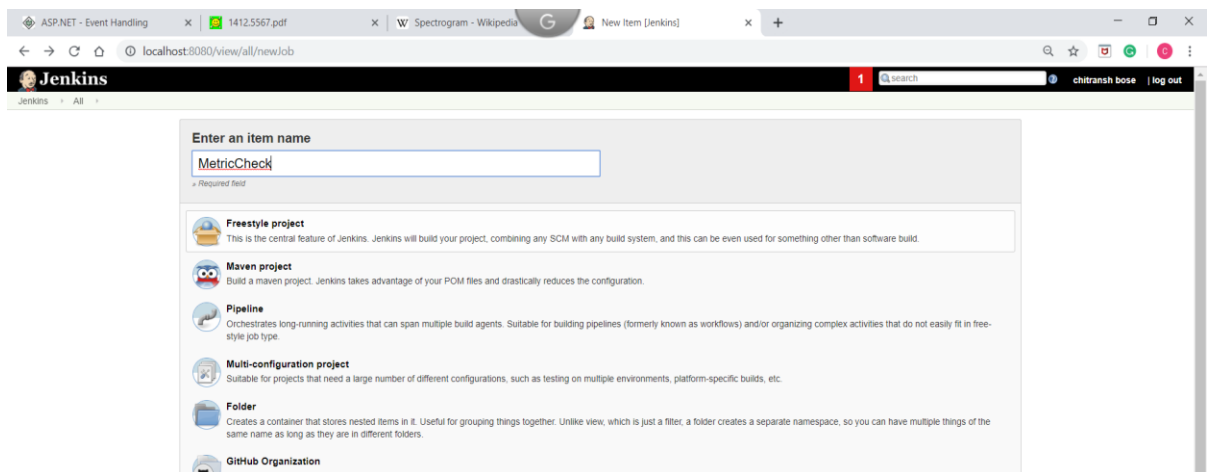
The screenshot shows the 'Post-build Actions' configuration in Jenkins. Under the 'Publish JUnit test result report' section, the 'Test report XMLs' field is set to 'target/surefire-reports/*.xml'. Below this, there is a 'Fileset Includes' section with a text box containing 'Fileset Includes: setting that specifies the generated raw XML report files, such as 'myproject/target-test-reports.xml'. There are also checkboxes for 'Retain long standard output/error' and 'Allow empty results'. The 'Health report amplification factor' is set to '1.0'. At the bottom, there are 'Save' and 'Apply' buttons.

9. Save and build it and check the results.

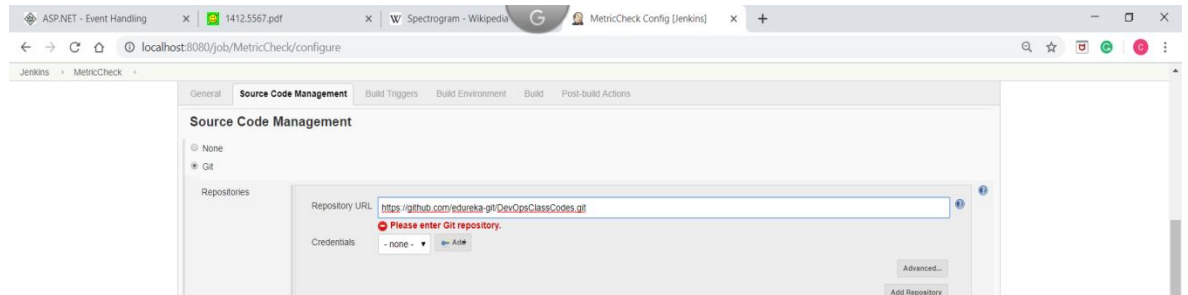


Task 2.m

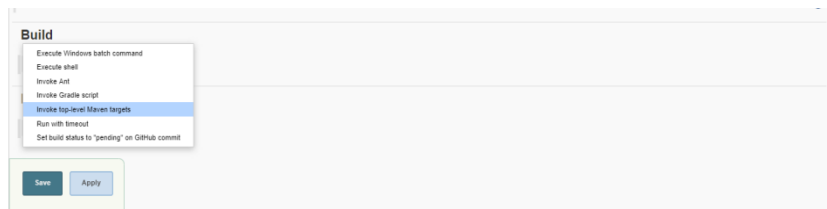
1. Create a job “MetricCheck”.



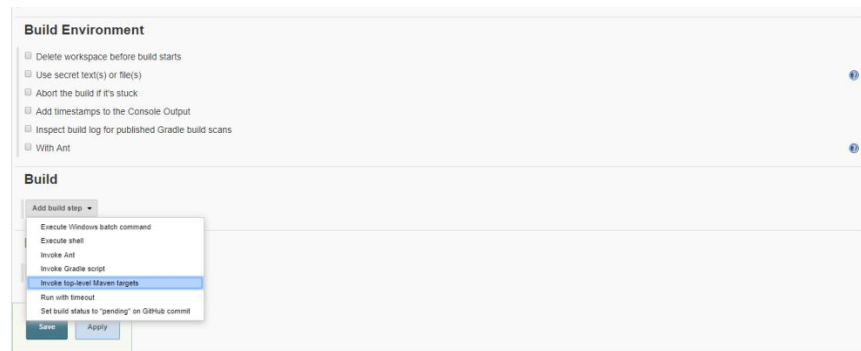
2. Goto source code management and choose git and enter the url given.



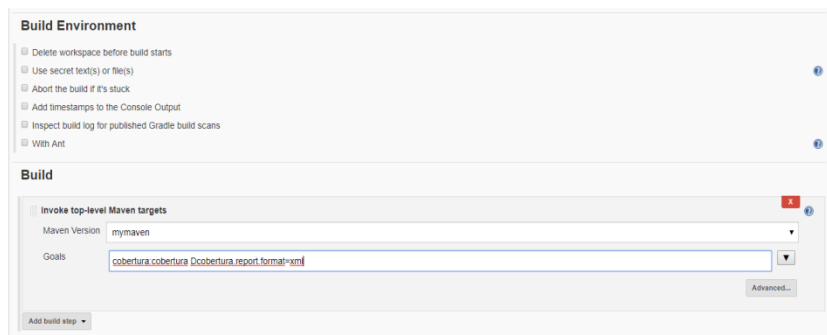
3. GOTO build, then add build step and select top level maven targets.



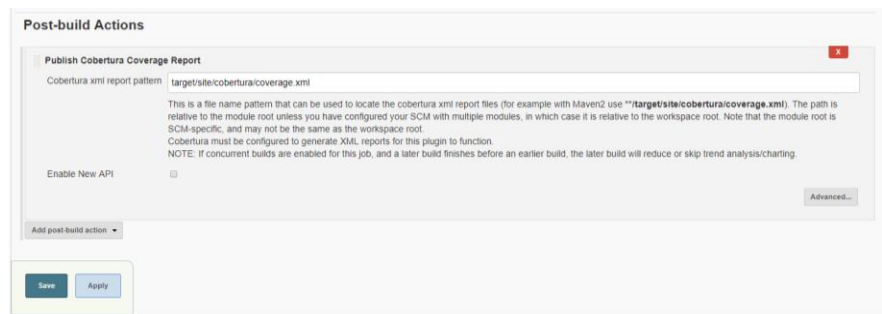
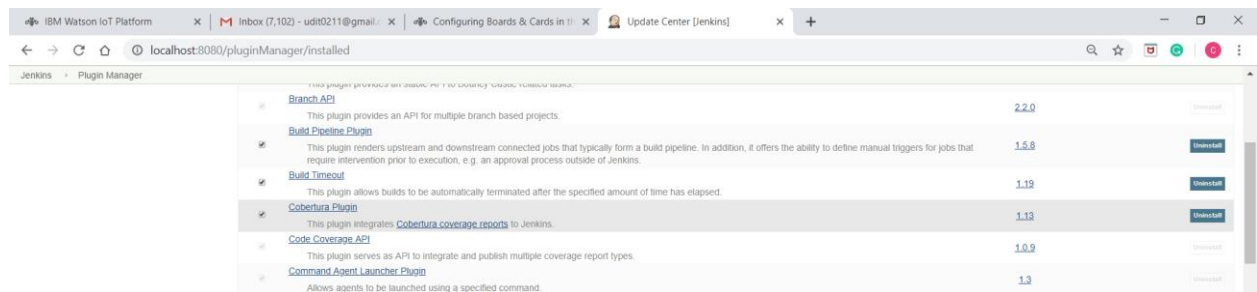
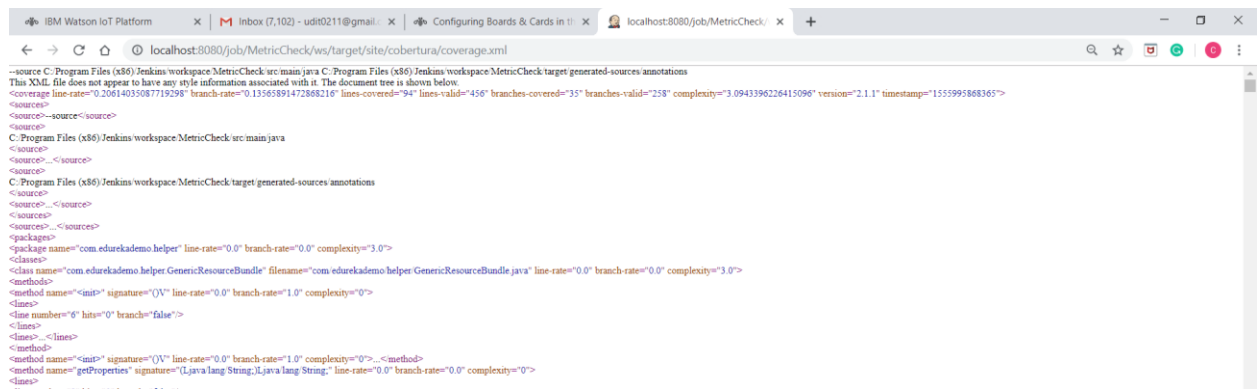
4. GOTO build, then add build step and select top level maven targets.

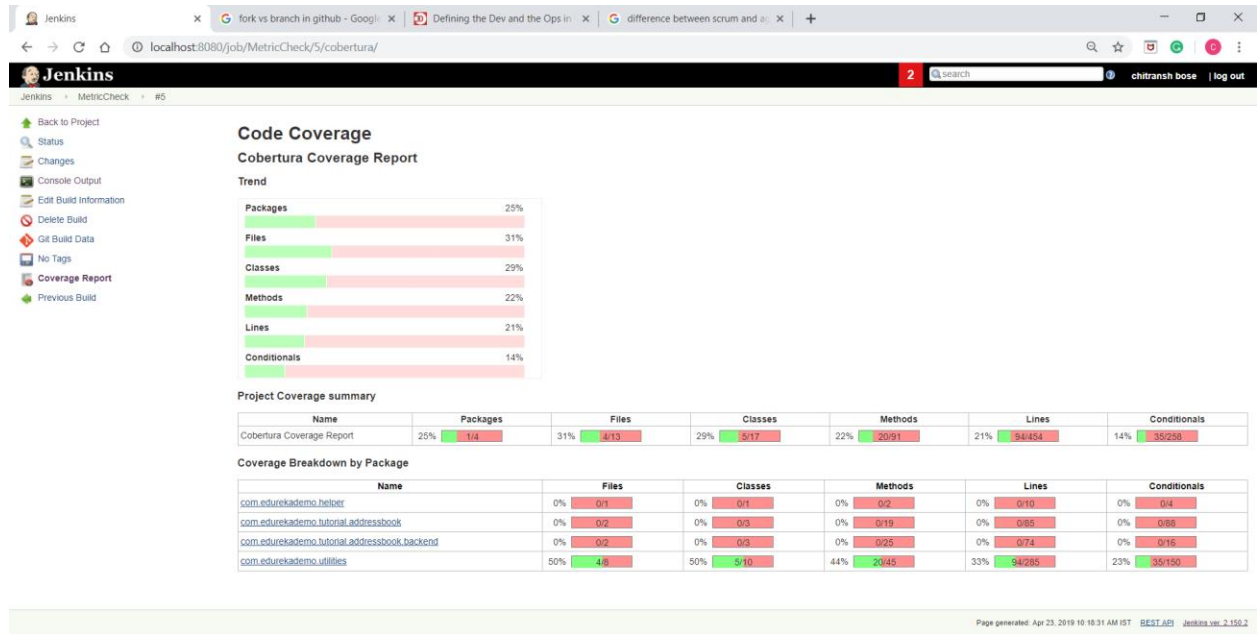


5. Choose maven version as mymaven and goals as cobertura:cobertura Dcobertura.report.format=xml.



6. Save and build it and check the output at workspace/target/site/cobertura/coverage.xml





Task 2.n

1. Create a freestyle job “Package”.

Enter an item name

Package

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

2. Configure the project and goto source code management and choose git and enter the url given.

Source Code Management

☐ None
☒ Git

Repositories

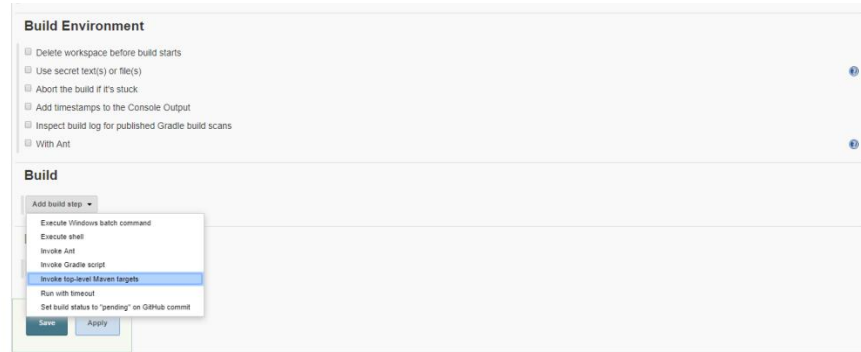
Repository URL:

Credentials:

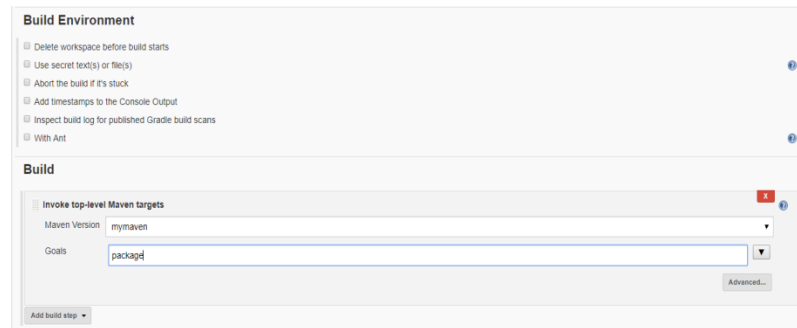
Branches to build

Branch Specifier (blank for 'any'):

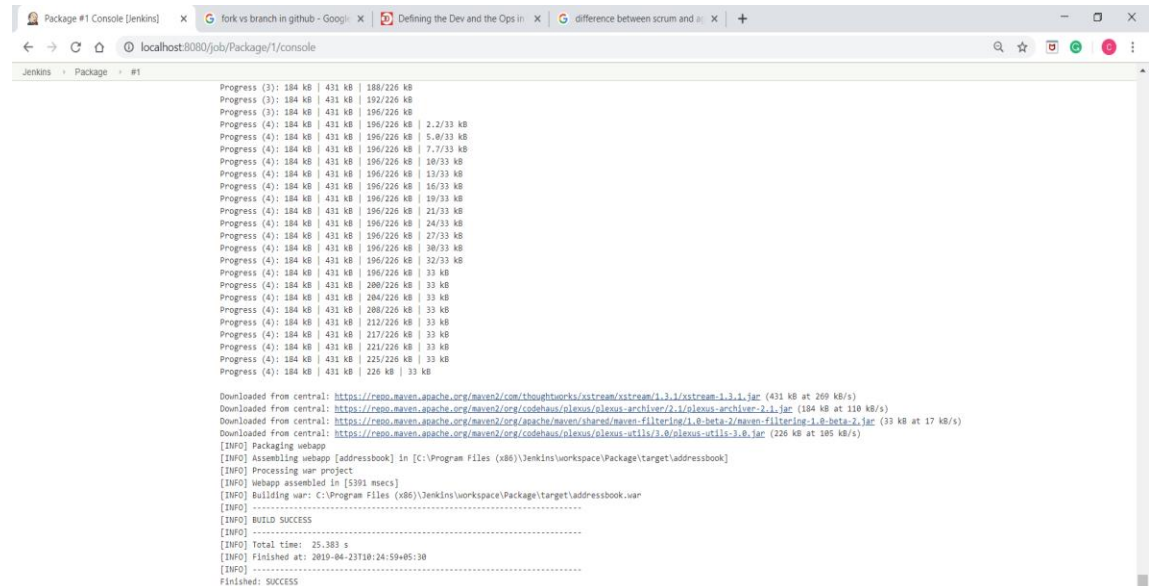
3. GOTO build, then add build step and select top level maven targets.



4. Choose maven version as mymaven and goals as package.

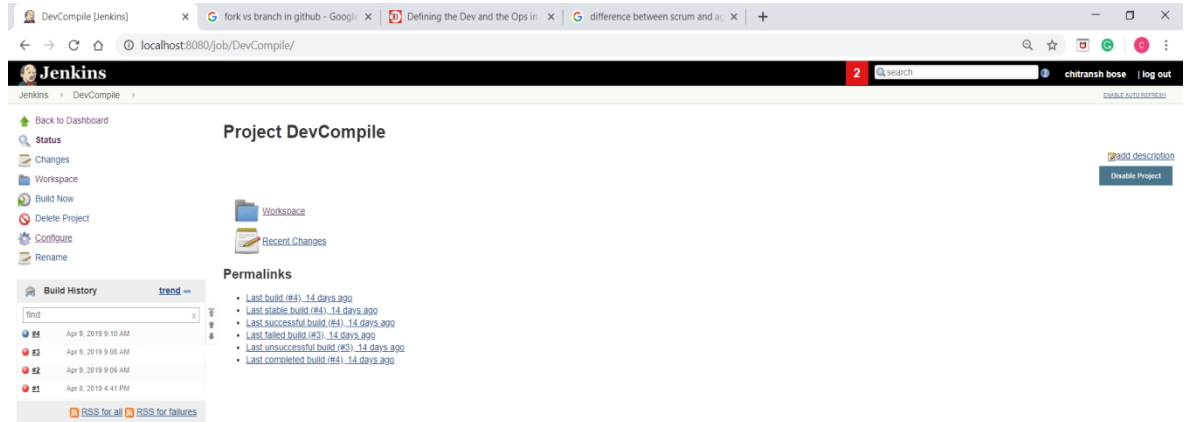


5. Save and build the project and check the console.

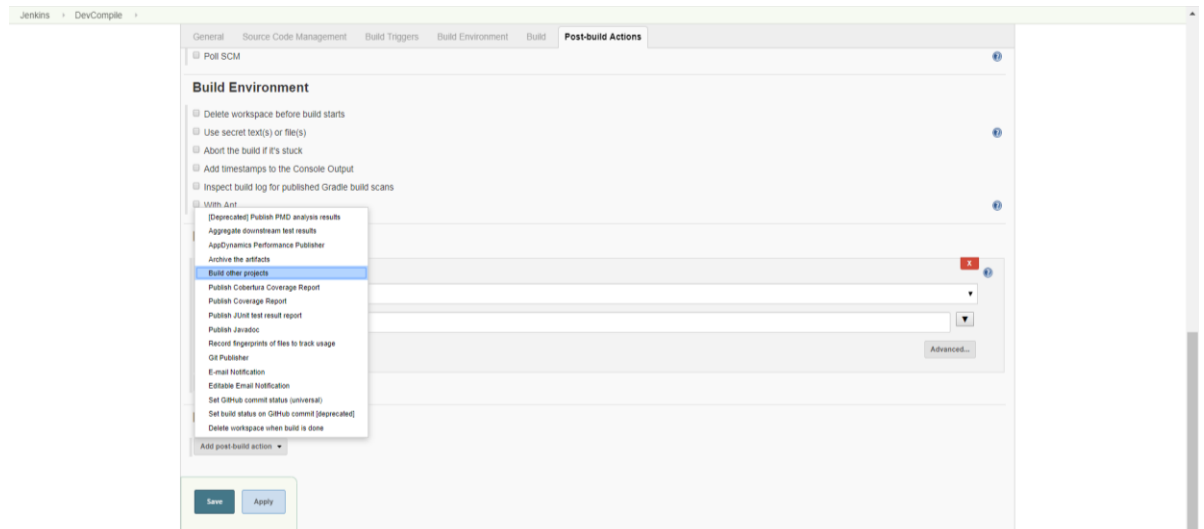


Task 2.o

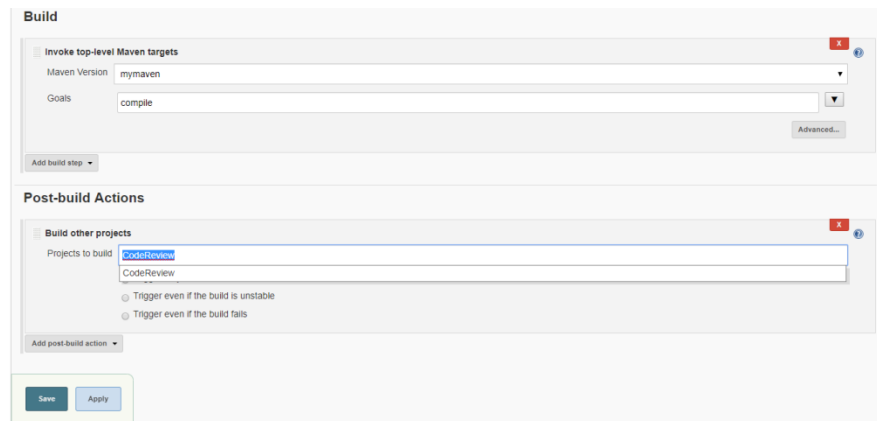
1. Goto to job DevCompile.



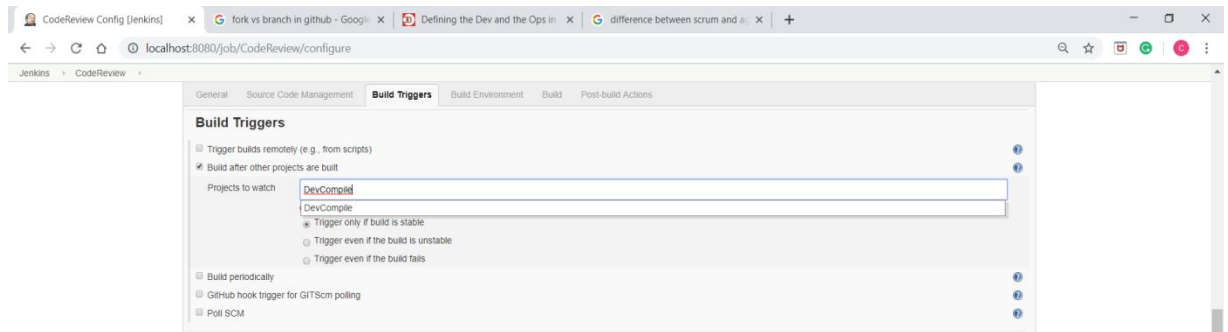
2. Goto Post Build Actions and add post build actions and choose build other project.



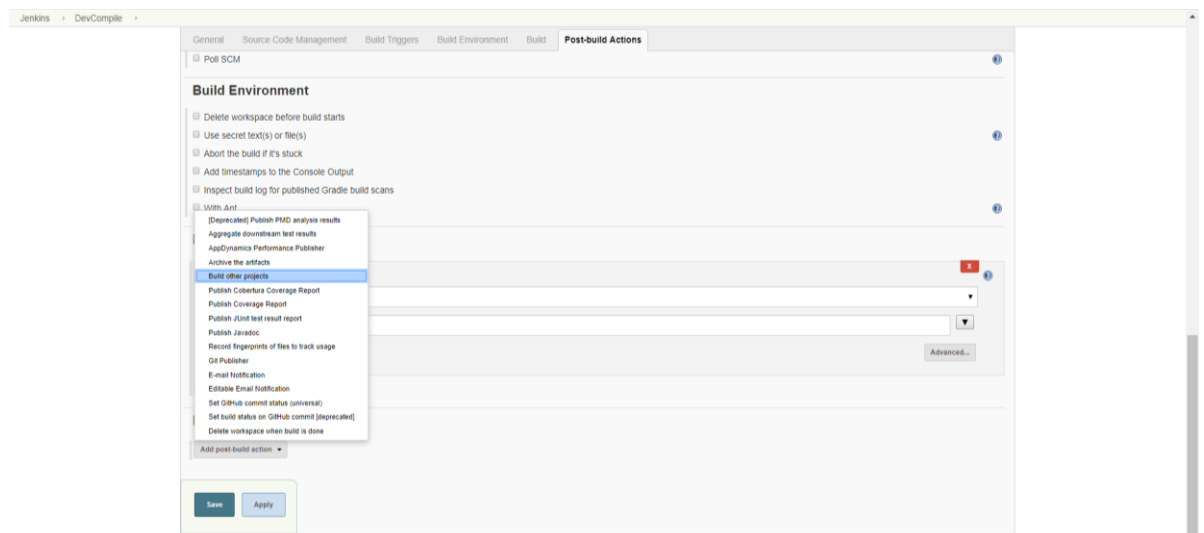
3. Project to build : CodeReview and save.



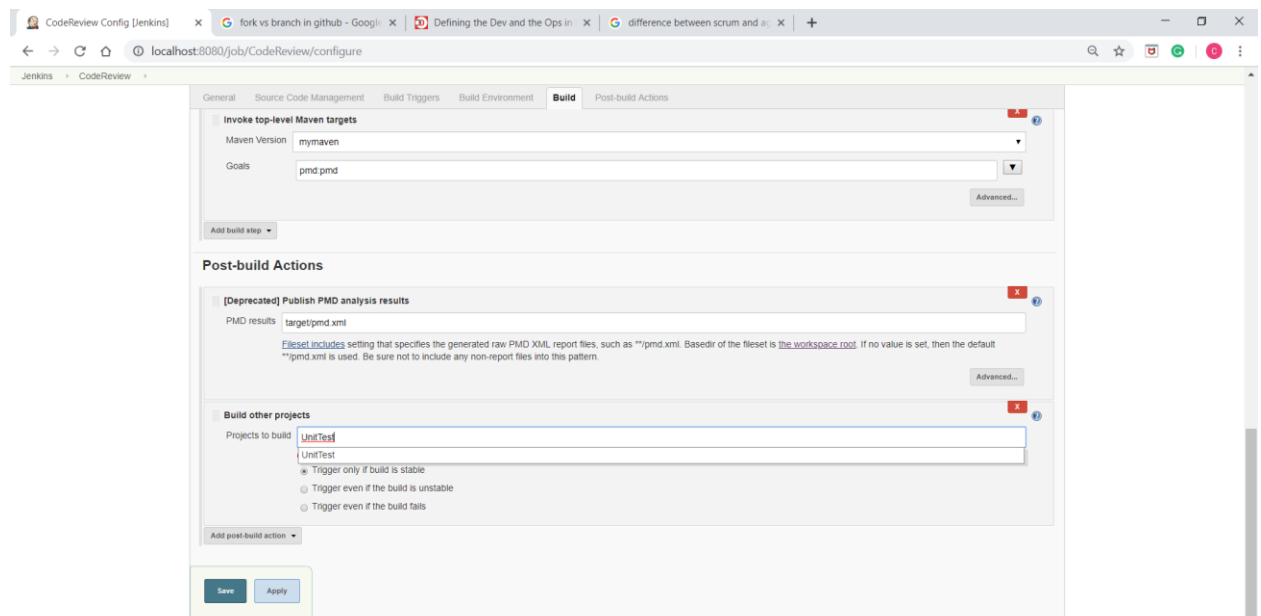
4. Goto CodeReview project and go to build triggers. Check the build after other projects are build and in Project to watch: DevCompile.



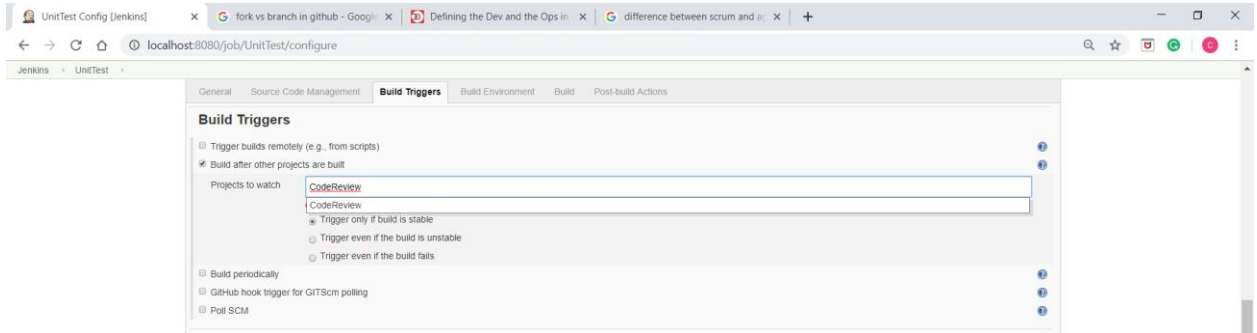
5. Goto Post Build Actions and add post build actions and choose build other project.



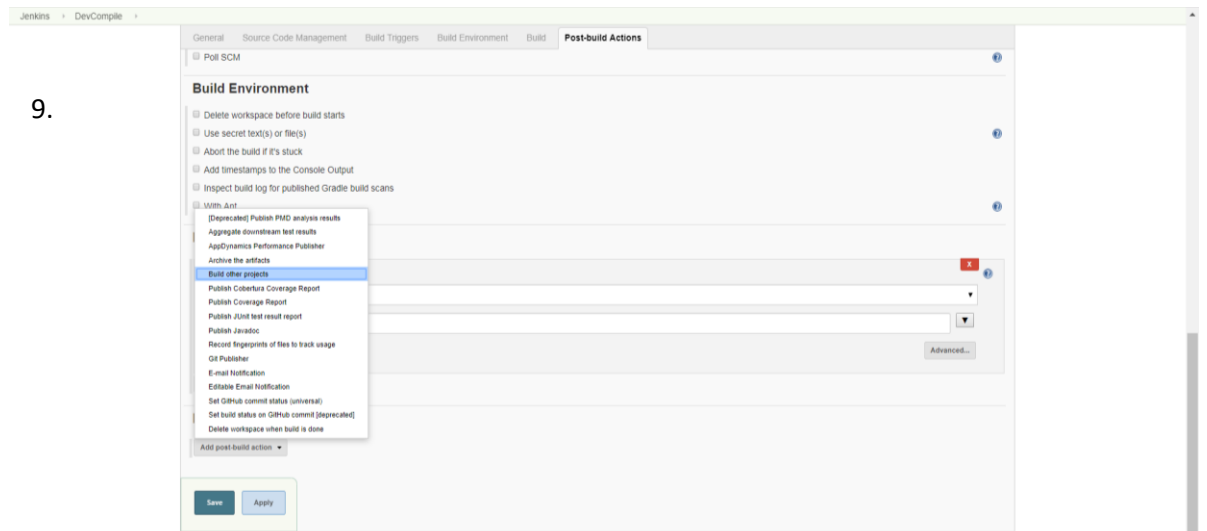
6. Project to build : UnitTest and save.



7. Goto UnitTest project and go to build triggers. Check the build after other projects are build and in Project to watch: CodeReview.

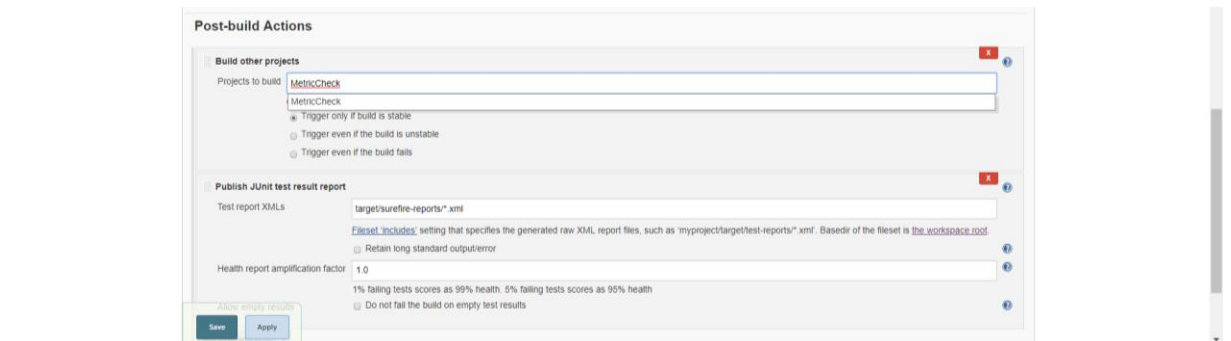


8. Goto Post Build Actions and add post build actions and choose build other project.

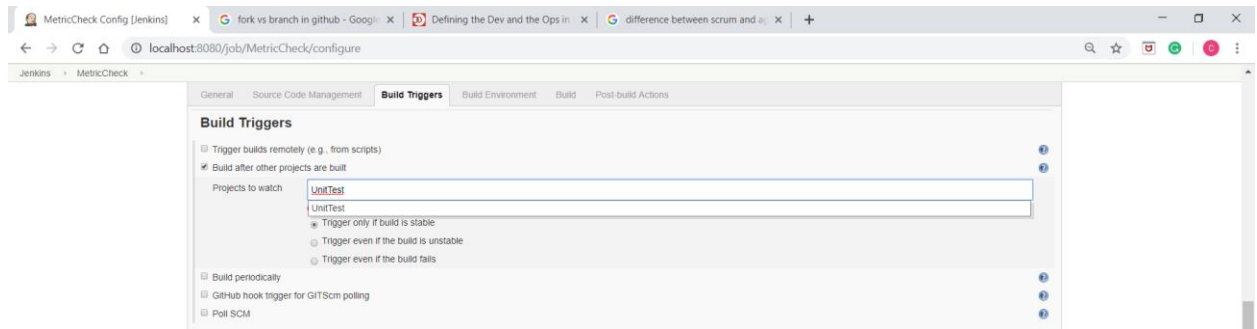


- 9.

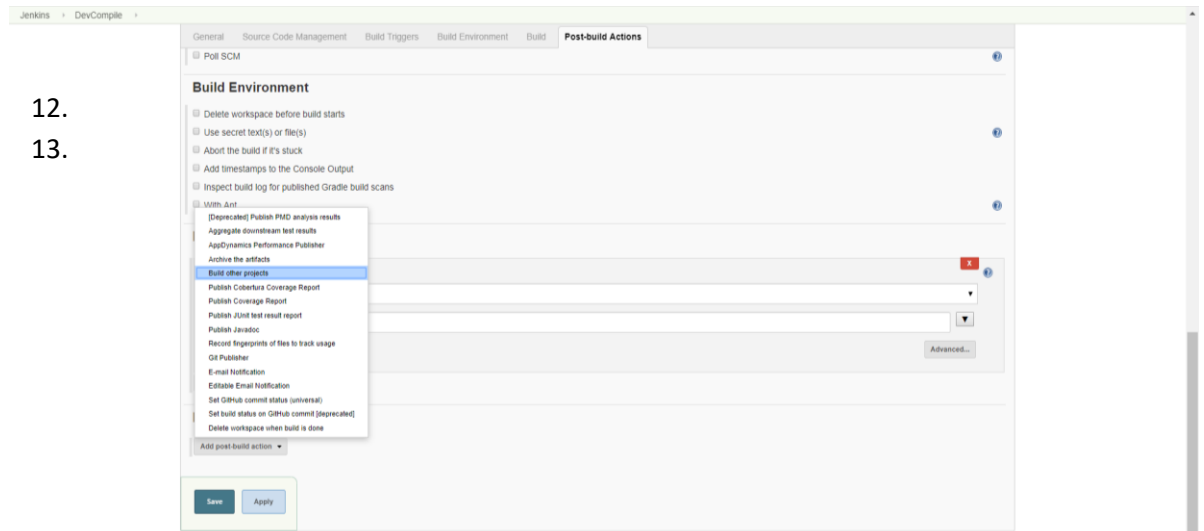
9. Project to build : MetricCheck and save.



10. Goto MetricCheck project and go to build triggers. Check the build after other projects are build and in Project to watch: UnitTest.



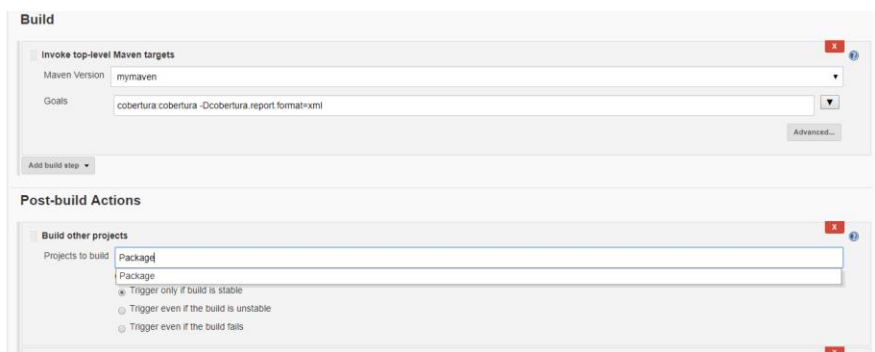
11. Goto Post Build Actions and add post build actions and choose build other project.



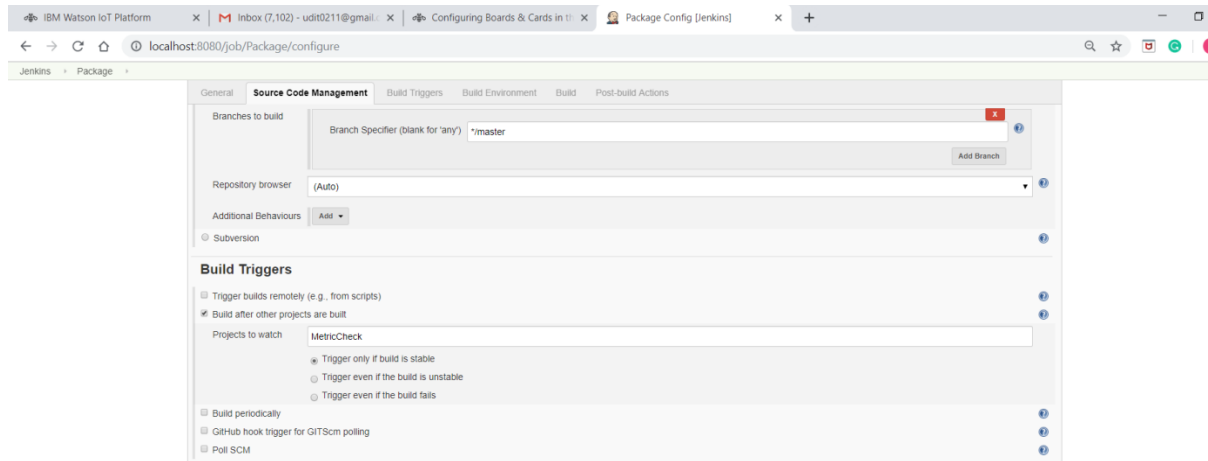
12.

13.

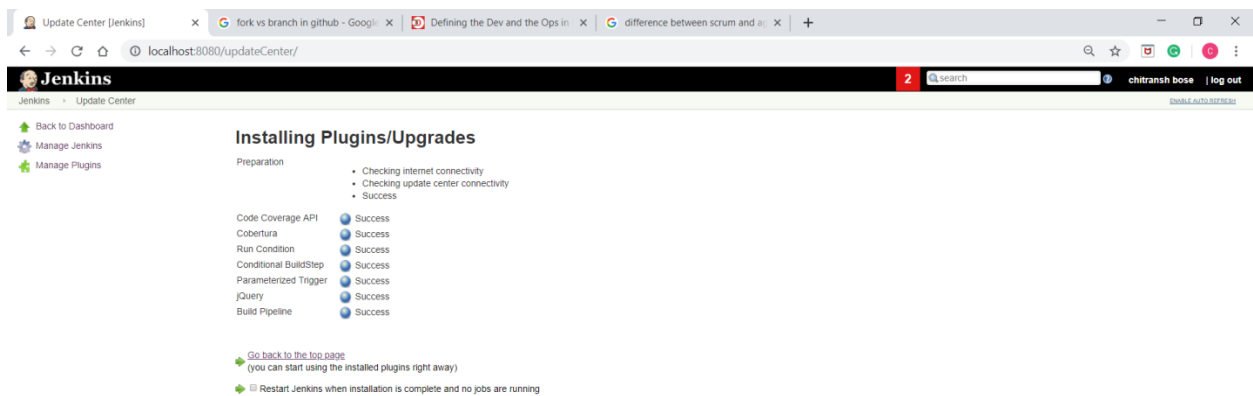
12. Project to build : Package and save.



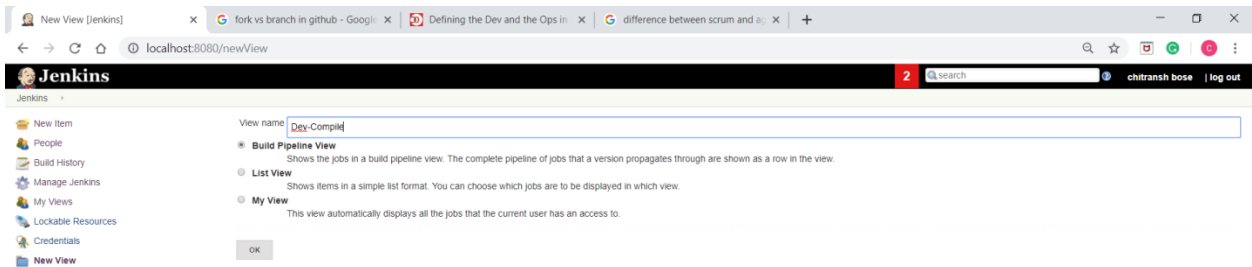
13. Goto Package project and go to build triggers. Check the build after other projects are build and in Project to watch: MetricCheck.



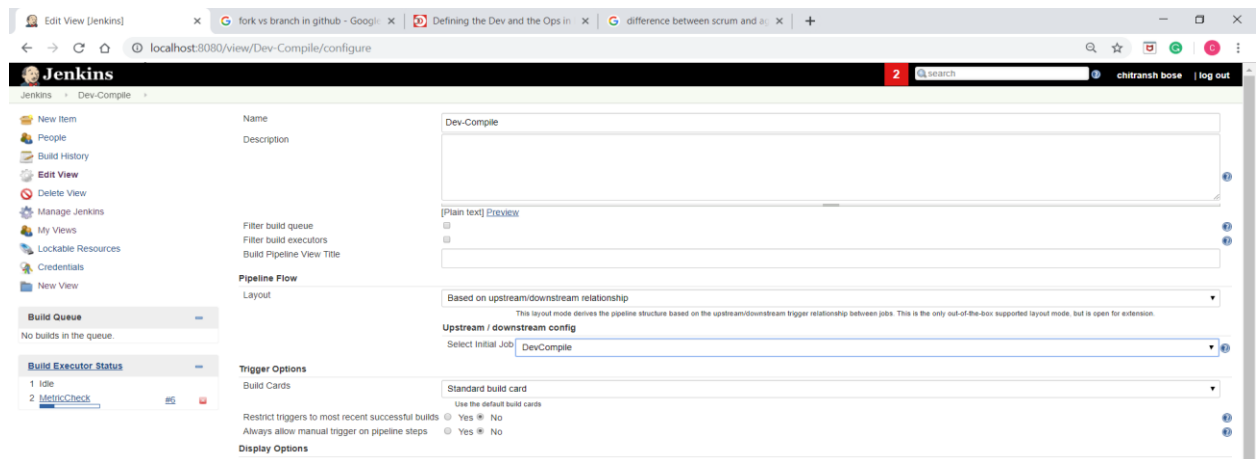
14. Install pipeline plugin.



15. Click on new view name it Dev-Compile and select build pipeline view.



16. Select initial job as DevCompile.



17. Output.

Dev-Compile [Jenkins]

fork vs branch in github - Google

Defining the Dev and the Ops in

difference between scrum and ag

localhost:8080/view/Dev-Compile/

Jenkins

2

search

chitransh bose

log out

JenkinsDev-Compile

ENABLE AUTO REFRESH

Build Pipeline

Run

History

Configure

Add Step

Delete

Manage

Pipeline

#5

#5 DevCompile

Apr 23, 2019 10:32:34 AM

18 sec

chitransh

#7 CodeReview

Apr 23, 2019 10:32:53 AM

53 sec

#3 UnitTest

Apr 23, 2019 10:33:53 AM

13 sec

#6 MetricCheck

Apr 23, 2019 10:34:13 AM

15 sec

Package

19A

19B

Ques1.

```
package selProject;

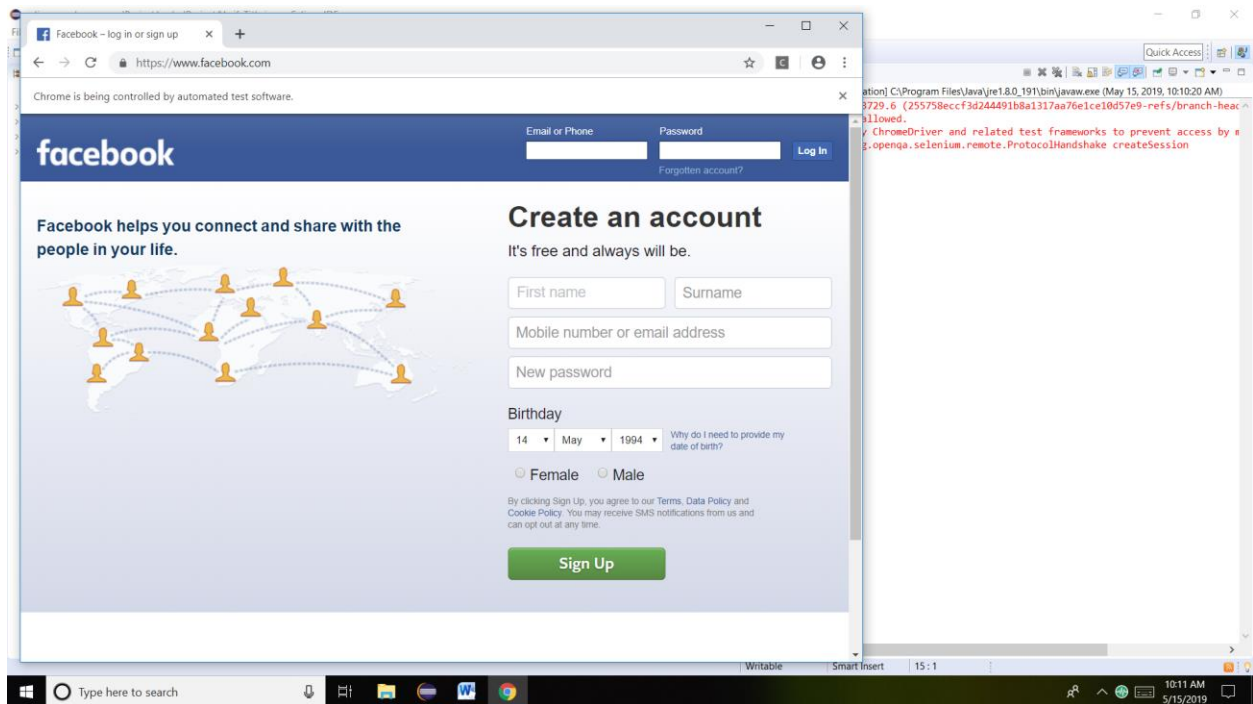
import org.openqa.selenium.chrome.ChromeDriver;

public class VerifyTitle {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.setProperty("webdriver.chrome.driver", "chromedriver.exe");
        ChromeDriver driver=new ChromeDriver();
        driver.get("http://www.facebook.com");
    }

}
```

Output:



Ques.

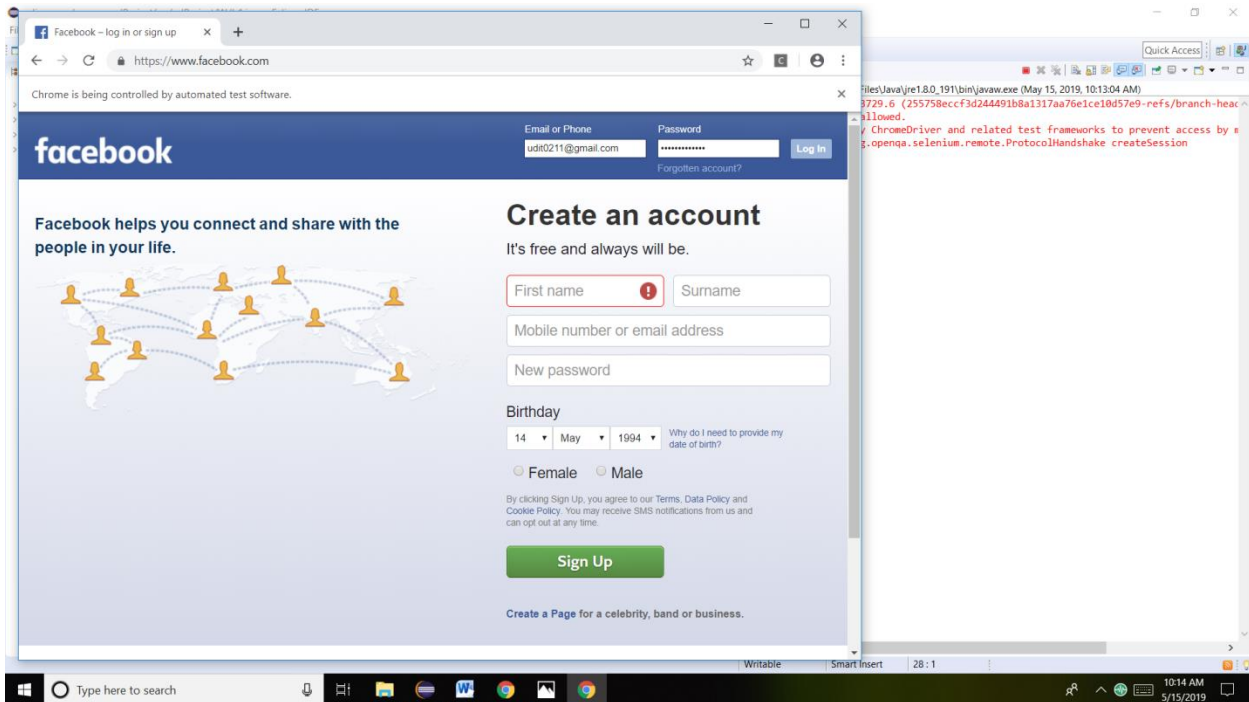
```
package selProject;

import org.openqa.selenium.By;
import org.openqa.selenium.Chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;

public class Wdb1 {

    public static void main(String[] args) {
        ChromeOptions options = new ChromeOptions();
        //options.addArguments("disable-infobars");
        System.setProperty("webdriver.chrome.driver", "chromedriver.exe");
        ChromeDriver driver=new ChromeDriver();
        driver.get("http://www.facebook.com");
        driver.findElement(By.id("email")).sendKeys("udit0211@gmail.com");
        driver.findElement(By.id("pass")).sendKeys("*****");
        driver.findElement(By.id("u_0_2")).click();
    }
}
```

Output:



Ques.

```
package selProject;
```

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

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

```
public class Wdd2 {
```

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

```
        // TODO Auto-generated method stub
```

```
        System.setProperty("webdriver.chrome.driver", "chromedriver.exe");
```

```
        //System.setProperty("webdriver.edge.driver",
```

```
        "MicrosoftWebDriver.exe");
```

```
        //EdgeDriver driver=new EdgeDriver();
```

```
        ChromeDriver driver=new ChromeDriver();
```

```
        driver.get("http://www.google.com");
```

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

```
        //driver.findElement(By.linkText("Images")).click();
```

```
        driver.findElement(By.xpath("//a[@href='https://www.google.co.in/imghp?hl=en&tab=wi']")).click();
```

```
        Thread.sleep(5000);
```

```
        driver.navigate().back();
```

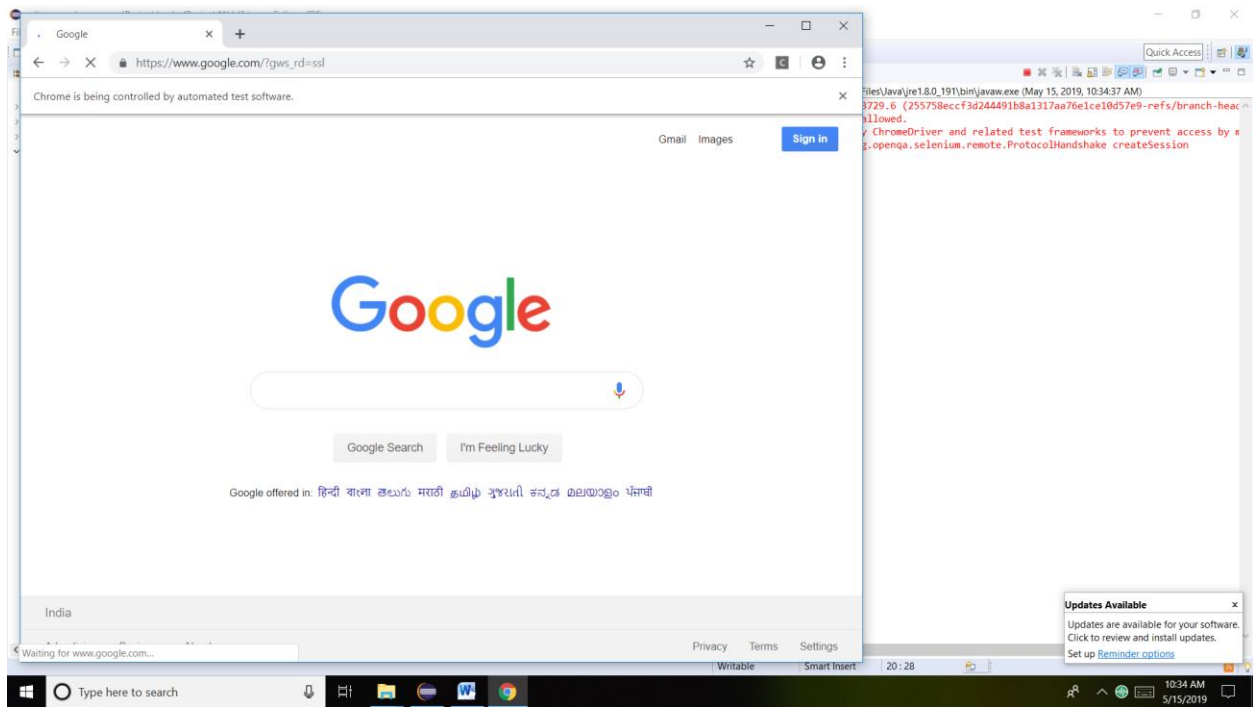
```
        Thread.sleep(4000);
```

```
        driver.navigate().forward();
```

```
    }
```

```
}
```


Output1.



Output2.

