



Jenkins

Continuous Integration

Avinash Patel
Senior Manager

Agenda

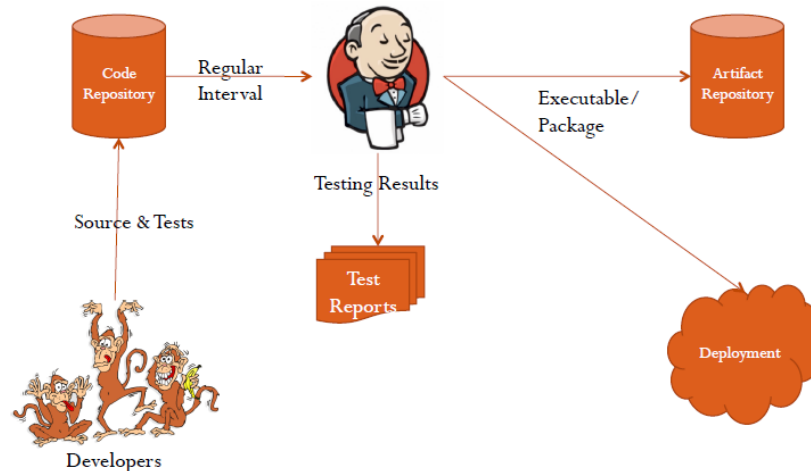
- Introduction – Jenkins
 - UI Walkthrough
 - Configure Git, Java, Maven and Email Notification
- Creating Jenkins Freestyle Job
 - Configure Build using batch commands & top level Maven goals
 - Configure Build using plugin based Maven goals
 - Configure Build, selenium test, deploy on server and notification
 - Automation of CI by defining poll SCM, hook trigger
- Creating Jenkins Pipeline Job
 - Construct groovy script to define simple pipeline job
 - Construct pipeline and monitor execution of defined stages

Jenkins - Introduction

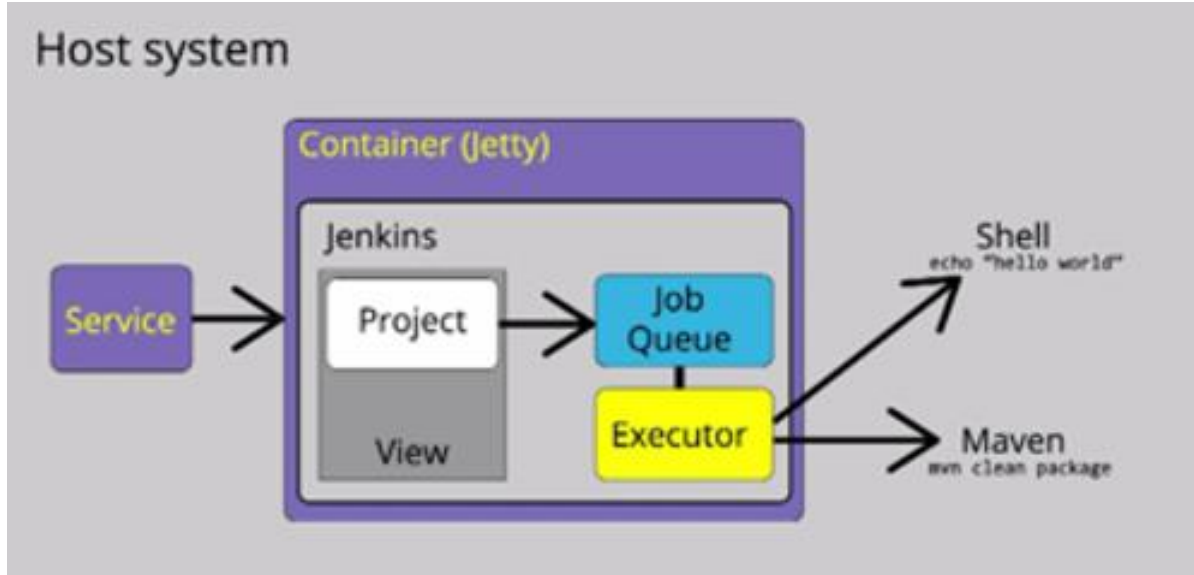
- Java based Continuous Build System
- Jenkins is a highly configurable system by itself
- Jenkins monitors the whole build process and provides reports and notifications to alert maintainers on success or errors
- Runs in servlet container
 - Glassfish, Tomcat

Benefits

- Immediate bug detection
- No integration step in the lifecycle
- A deployable system at any given point
- Record of evolution of the project



Jenkins Architecture



How Jenkins works - Setup

- It associates with a version control server
- It triggers builds through polling, periodic and building based on other projects
- It executes Maven targets, bash scripts etc.,
- It does Artifact archival
- It publishes javadocs and JUnit test results
- It does email notifications
- Through plugins it expands the functionality even further

Jenkins for a Developer

- **Easy to install**

- First download the file – jenkins.war
- Then run the command – java -jar jenkins.war

Note: You can alternatively use windows installer

- **Easy to use**

- First create a new job – You could initially checkout and build a small project
- Then check-in a change – watch it build
- Further create a test – watch it build and run
- Subsequently fix a test – check-in and watch it pass

- **Multi-technology**

- You can develop code with C, Java, C#, Python, Perl, SQL, etc.
- You can test with Junit, Nunit, MSTest, Selenium etc.

Jenkins Features

- Get source code from source code repository
- Trigger a build - Automatically build and test
- Post build, generate report & notify as Jenkins comes with basic reporting features that would help us to keep track of build status based on its last success and failure status
 - These features can be enhanced with use of pre-build plugins
 - Unit test coverage and Test result trending
- Deploy
- Distributed build

Jenkins terminologies

Term used in Jenkins	Description
Upstream project	A project can have one or several upstream projects, which means that current project is scheduled depending if the upstream project is built successfully or not. If the upstream project is successful the current project is added to the build queue. If the upstream project is broken the current project will not be added to the build queue.
Downstream project	A project can have one or several downstream projects. The downstream projects are added to the build queue if the current project is built successfully. It is an option to add the downstream project to the build queue even if the current project is unstable (default is off).
Stable build	A build is stable if it was built successfully and no publisher reports it as unstable.
Unstable build	A build is unstable if it was built successfully and one or more publishers report it unstable. For example if the JUnit publisher is configured and a test fails then the build will be marked unstable.
(Un)Stable project	A project is (un)stable if its most recent (completed) build is (un)stable.
Successful build	A build is successful when the compilation reported no errors.
Broken build Failed Build	A build is broken if it failed during building. That is, it is not successful.
Broken project	A project is broken if its most recent (completed) build is broken.
Slave	Slaves are computers that are set up to build projects for a master. Jenkins runs a separate program called "slave agent" on slaves. When slaves are registered to a master, a master starts distributing loads to slaves.
Publisher	A publisher is part of the build process other than compilation, for example JUnit test runs. A publisher may report <i>stable</i> or <i>unstable</i> result depending on the result of its processing. For example, if a JUnit test fails, then the whole JUnit publisher may report <i>unstable</i> .
Completed Build	A build is completed, if it was started and finished with any result, including failed builds.

Jenkins- Basics



Jenkins- Basic overview

- Once Jenkins service is started, open any browser with localhost:8080
- Creating the job
 - New item can be used for creating a job or project
- Running the jobs
 - Build now option is used to build and run the jobs
- Viewing output
 - Results can be viewed in the console output format as well
- Jenkins dashboard
 - Dashboard provides the complete view of all the jobs and their status


Jenkins- Basic overview


- Disabling jobs
 - Disable project option can be chosen to disable the jobs
- Deleting a job
 - Delete project can be used for deleting a job
- Updating and adding plugins
 - Manage plug in option can be used for updating and adding plugins


Jenkins configuration


- Manage Jenkins- configuration can be used


Jenkins > configuration


 New Item


 People


 Build History

 Project Relationship

 Check File Fingerprint

 Manage Jenkins

 My Views

 Credentials

Home directory

E:\Program Files (x86)\Jenkins

System Message

Advanced...

[Plain text] [Preview](#)

2

Labels

Utilize this node as much as possible

5

SCM checkout retry count

0

☐ Restrict project naming

Global properties

Save

Apply

Jenkins-Maven

- Create a new project and set MAVEN_HOME variable

List of Ant installations on this system

Maven

Maven installations

■ Maven	
Name	<input type="text" value="apache-maven"/>
MAVEN_HOME	<input type="text" value="E:\DEVOPS\apache-maven-3.3.9"/>
<input type="checkbox"/> Install automatically	?
	Delete Maven
Add Maven	

List of Maven installations on this system

Maven Project Configuration

Global MAVEN_OPTS [?](#)

Local Maven Repository [?](#)

☒ Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project. [?](#)

Build Tools

- **Java**
 - Maven (build-in), Ant, Gradle
- **.Net**
 - MSBuild, PowerShell
- **Shell script**
 - Python, Ruby, Groovy

Build Wrapper

- Build name (version no) setter
- Virtual machine (VMWare, Virtual Box)
- Set environment variable
- ClearCase release plugin

Build Notifier

- The various options are :
 - E-mail
 - Twitter
 - Jabber
 - IRC
 - RSS
 - Google calendar

Build Report

- **Static Code Analysis**
 - Checkstyle, PMD, Findbugs, Compiler Warning
- **Test Report & Code Coverage**
 - JUnit, TestNG, Cobertura, Clover
- **Open Tasks**

**Configure Jenkins with email
notification**

**Trigger a Freestyle job build through
Git hook with email notification**



Configure Jenkins

- Configure Jenkins

The screenshot shows the Jenkins dashboard interface. On the left sidebar, the 'Manage Jenkins' link is circled in blue. A blue arrow originates from this link and points towards the right side of the image, where the text 'Click On Manage Jenkins' is located. The main content area displays a table of build jobs. Below the table, there are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing 1 Idle and 2 Idle executors). The footer of the page indicates it was generated on Jan 13, 2018, at 2:18:59 PM IST, and mentions 'Jenkins ver. 2.89.2'.

S	W	Name	Last Success	Last Failure	Last Duration
		DemoWebApp	2 days 22 hr - #4	2 days 23 hr - #2	23 sec
		justdisplay	7 days 21 hr - #28	9 days 2 hr - #19	13 sec
		raghupipeline	4 days 21 hr - #16	4 days 21 hr - #11	14 sec

Legend RSS for all RSS for failures RSS for just latest builds

Click On Manage Jenkins

Configure Jenkins

- Configure Environment Variables like CLASSPATH , JAVA_HOME in Jenkins

The screenshot shows the Jenkins Configuration System interface. The browser address bar indicates the URL is `localhost:8080/configure`. The left sidebar shows the navigation menu with 'Jenkins' and 'configuration'. The main content area is titled 'Global properties' and has a sub-section 'Environment variables' which is checked and circled. Below this, there are two environment variables configured:

Name	Value	Action
CLASSPATH	.;C:\Program Files\Java\jdk1.7.0\lib;C:\Program Files\Java\jdk1.8.0_65\lib;C:\Users\rsethun	Delete
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_65	Delete

At the bottom of the environment variables section, there is an 'Add' button. Below this, there are sections for 'Tool Locations', 'Usage Statistics' (with a checkbox for sending anonymous usage statistics and crash reports), and 'Timestampers' (with fields for 'System clock time format' and 'Elapsed time format'). At the bottom of the configuration page, there are 'Save' and 'Apply' buttons. The Windows taskbar at the bottom shows the time as 12:19 PM on 1/13/2018.

Configure Jenkins

- Steps to define E-mail Notification:

Configure System [Jenkins] X

localhost:8080/configure

Jenkins > configuration

Administrative monitors...

Jenkins Location

Jenkins URL ?
⚠ Please set a valid host name, instead of localhost

System Admin e-mail address ?

GitHub

GitHub Servers ?

GitHub Enterprise Servers

Pipeline Model Definition

Docker Label ?

Docker registry URL ?

Registry credentials

Global Pipeline Libraries

Shareable libraries available to any Pipeline like pipeline on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use...

Configure Jenkins

- Steps to define E-mail Notification:

The screenshot shows the Jenkins configuration page for 'Extended E-mail Notification'. The browser address bar indicates 'localhost:8080/configure'. The page has a sidebar with 'Jenkins' and 'configuration' links. The main content area contains the following fields and options:

- SMTP server:** smtp.office365.com
- Default user E-mail suffix:** @wipro.com
- Default Content Type:** Plain Text (text/plain) (with a dropdown arrow and an 'Advanced...' button circled in blue)
- ☐ Use List-ID Email Header
- ☐ Add 'Precedence: bulk' Email Header
- Default Recipients:** (empty text field)
- Reply To List:** (empty text field)
- Emergency reroute:** (empty text field)
- Excluded Recipients:** (empty text field)
- Default Subject:** \$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS
- Maximum Attachment Size:** (empty text field)
- Default Content:** \$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS

At the bottom of the form are 'Save' and 'Apply' buttons. A blue arrow points from the text 'Click on Advanced after entering SMTP server details' to the 'Advanced...' button.

Click on **Advanced** after entering SMTP server details

Configure Jenkins

- Steps to define E-mail Notification:

The screenshot shows the Jenkins configuration interface in a web browser. The browser's address bar displays 'localhost:8080/configure'. The page title is 'Configure System [Jenkins]'. The breadcrumb navigation shows 'Jenkins > configuration'. The 'Extended E-mail Notification' section is highlighted with a blue oval. This section contains the following fields and options:

- SMTP server:** smtp.office365.com
- Default user E-mail suffix:** @wipro.com
- ☒ **Use SMTP Authentication**
- User Name:** rsethum@wipro.com
- Password:** [masked with dots]
- ☐ **Use SSL**
- SMTP port:** 587
- Charset:** UTF-8
- Default Content Type:** Plain Text (text/plain)
- ☐ **Use List-ID Email Header**
- ☐ **Add 'Precedence: bulk' Email Header**
- Default Recipients:** [empty field]
- Reply To List:** [empty field]

At the bottom of the configuration section are two buttons: 'Save' and 'Apply'. The Windows taskbar at the bottom shows the system clock as 12:20 PM on 1/13/2018.

Configure Jenkins

Steps to define E-mail Notification:

The screenshot shows the Jenkins configuration interface in a web browser. The browser's address bar displays 'localhost:8080/configure'. The page title is 'Configure System [Jenkins]'. The breadcrumb navigation shows 'Jenkins > configuration'. The configuration page is divided into several sections. The 'E-mail Notification' section is highlighted with a red circle. It contains the following fields and options:

- Content Token Reference**: A text input field.
- SMTP server**: A text input field containing 'smtp.office365.com'.
- Default user e-mail suffix**: A text input field containing '@wipro.com'.
- Test configuration by sending test e-mail**: A checkbox that is currently unchecked.

Below the 'E-mail Notification' section are 'Save' and 'Apply' buttons. To the right of the 'SMTP server' and 'Default user e-mail suffix' fields is an 'Advanced...' button, which is also highlighted with a red circle. A blue arrow points from the text 'Click on Advanced after entering SMTP server details' to this button. The footer of the page indicates 'Page generated: Jan 13, 2018 12:17:34 PM IST' and provides links for 'REST API' and 'Jenkins ver. 2.89.2'.

Click on **Advanced** after entering SMTP server details

Configure Jenkins

- Steps to define E-mail Notification:

The screenshot shows the Jenkins configuration interface in a web browser. The address bar shows 'localhost:8080/configure'. The page title is 'Configure System [Jenkins]'. The breadcrumb navigation shows 'Jenkins > configuration'. The 'E-mail Notification' section is expanded, showing various configuration fields. A blue circle highlights the 'Save' button at the bottom left of the configuration section. A blue arrow points from the 'Save' button to the text 'Click on Save and complete the basic configuration'.

Content Token Reference

E-mail Notification

SMTP server: smtp.office365.com

Default user e-mail suffix: @wipro.com

☒ Use SMTP Authentication

User Name: rsethum@wipro.com

Password: [Redacted]

☐ Use SSL

SMTP Port: 587

Reply-To Address: raghavendran.sethumadhavan1@wipro.com

Charset: UTF-8

☐ Test configuration by sending test e-mail

Save Apply

Page generated: Jan 13, 2018 12:17:34 PM IST REST API Jenkins ver. 2.89.2

Click on **Save** and complete the basic configuration

Create Jenkins Freestyle Job

Create a New Job in Jenkins

The screenshot shows the Jenkins dashboard interface. The 'New Item' button is circled in blue. A blue arrow points from the 'New Item' button to the text 'Click on New Item to create a job'.

Jenkins Dashboard Data:

S	W	Name	Last Success	Last Failure	Last Duration
		DemoWebApp	2 days 22 hr - #4	2 days 23 hr - #2	23 sec
		justdisplay	7 days 21 hr - #28	9 days 2 hr - #19	13 sec
		raghupipeline	4 days 21 hr - #16	4 days 21 hr - #11	14 sec

Build Queue: No builds in the queue.

Build Executor Status:

- 1 Idle
- 2 Idle

Page generated: Jan 13, 2018 2:18:59 PM IST [REST API](#) [Jenkins ver. 2.89.2](#)

Click on New Item to create a job

Create Jenkins Freestyle Job

Create a New Job

New Item [Jenkins] x







localhost:8080/view/all/newJob

Jenkins 3 search

Jenkins > All >

Enter an item name

» Required field

-  **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.
-  **Pipeline**
Orchestrates long-running activities that can span multiple build slaves. 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.
-  **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
-  **GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
-  **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Configure Jenkins Freestyle Job

Configure a new Job

The screenshot shows the Jenkins web interface in a browser window. The address bar shows 'localhost:8080/job/firsttry/'. The Jenkins logo is in the top left, and a red box with the number '3' is in the top right. The left sidebar contains a list of links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', 'Configure' (highlighted with a red circle), and 'GitHub Hook Log'. The main content area is titled 'Project justdisplay' and shows 'Project name: firsttry'. There are links for 'add description' and 'Disable Project'. Below this, there are links for 'Workspace' and 'Recent Changes'. The 'Permalinks' section lists several build links: 'Last build (#28). 7 days 22 hr ago', 'Last stable build (#28). 7 days 22 hr ago', 'Last successful build (#28). 7 days 22 hr ago', 'Last failed build (#19). 9 days 2 hr ago', 'Last unsuccessful build (#19). 9 days 2 hr ago', and 'Last completed build (#28). 7 days 22 hr ago'. At the bottom left, there is a 'Build History' table with columns for build number and time. The table shows builds from #22 to #28, with the most recent build being #28 on Jan 5, 2018 at 4:20 PM. The Windows taskbar is visible at the bottom, showing the search bar and various application icons.

localhost:8080/job/firsttry/

Jenkins

3 search

Jenkins > justdisplay > [DISABLE AUTO REFRESH](#)

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[GitHub Hook Log](#)

Project justdisplay

Project name: firsttry

[add description](#)

[Disable Project](#)

[Workspace](#)

[Recent Changes](#)

Permalinks

- [Last build \(#28\). 7 days 22 hr ago](#)
- [Last stable build \(#28\). 7 days 22 hr ago](#)
- [Last successful build \(#28\). 7 days 22 hr ago](#)
- [Last failed build \(#19\). 9 days 2 hr ago](#)
- [Last unsuccessful build \(#19\). 9 days 2 hr ago](#)
- [Last completed build \(#28\). 7 days 22 hr ago](#)

Build History

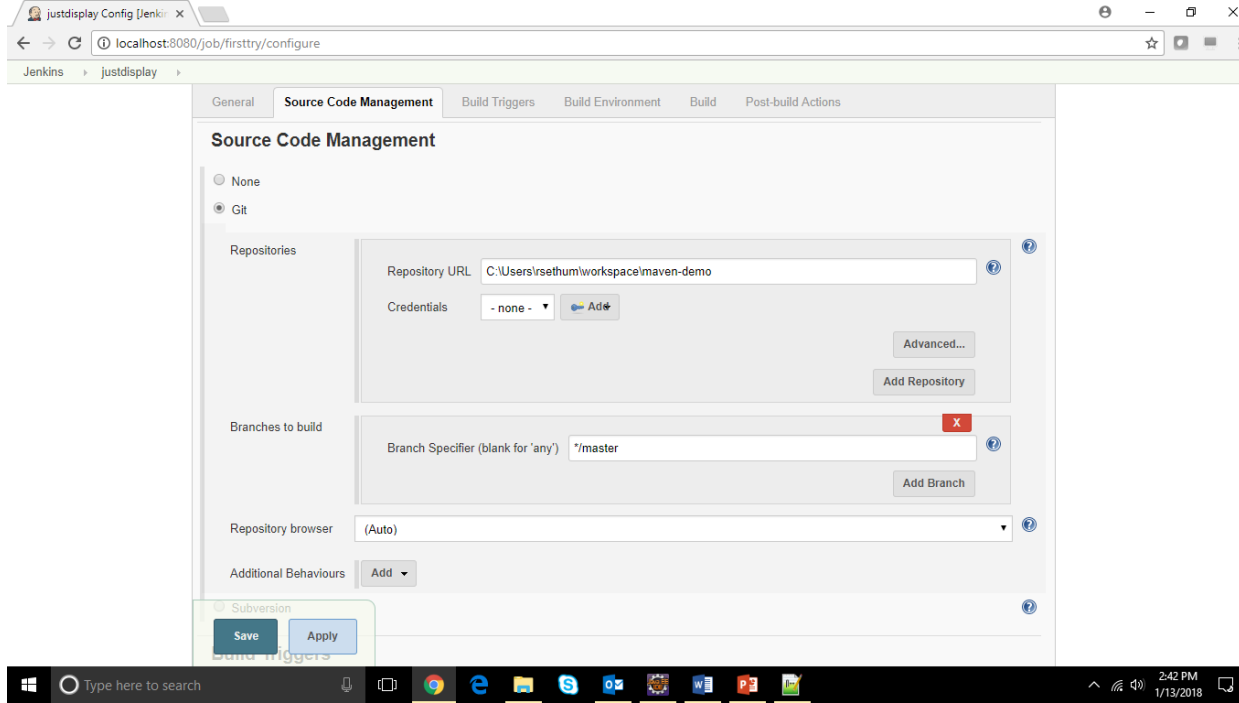
trend

Build	Time
#28	Jan 5, 2018 4:20 PM
#27	Jan 5, 2018 10:11 AM
#26	Jan 4, 2018 7:08 PM
#25	Jan 4, 2018 7:01 PM
#24	Jan 4, 2018 12:34 PM
#23	Jan 4, 2018 12:30 PM
#22	Jan 4, 2018 12:29 PM

localhost:8080/job/firsttry/configure

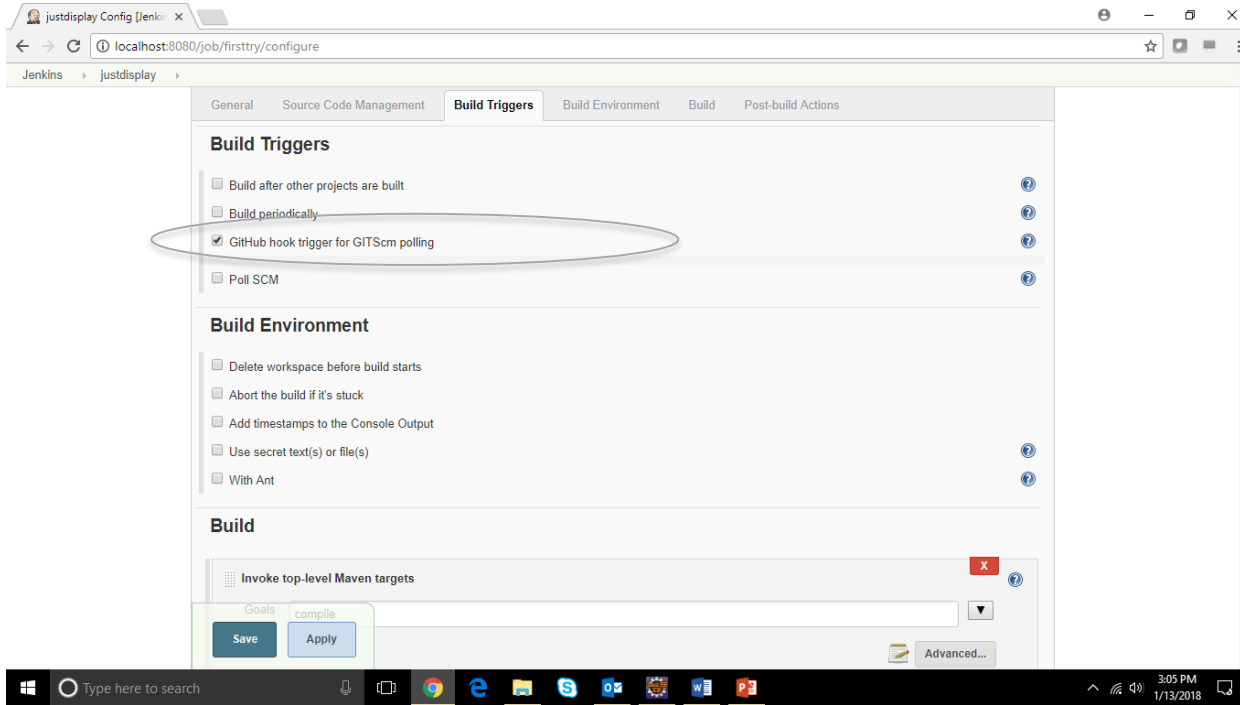
Configure Jenkins Freestyle Job

Declare that the source will be pulled from a Git repository



Configure Jenkins Freestyle Job

Declare that the source will be pulled from a Git repository



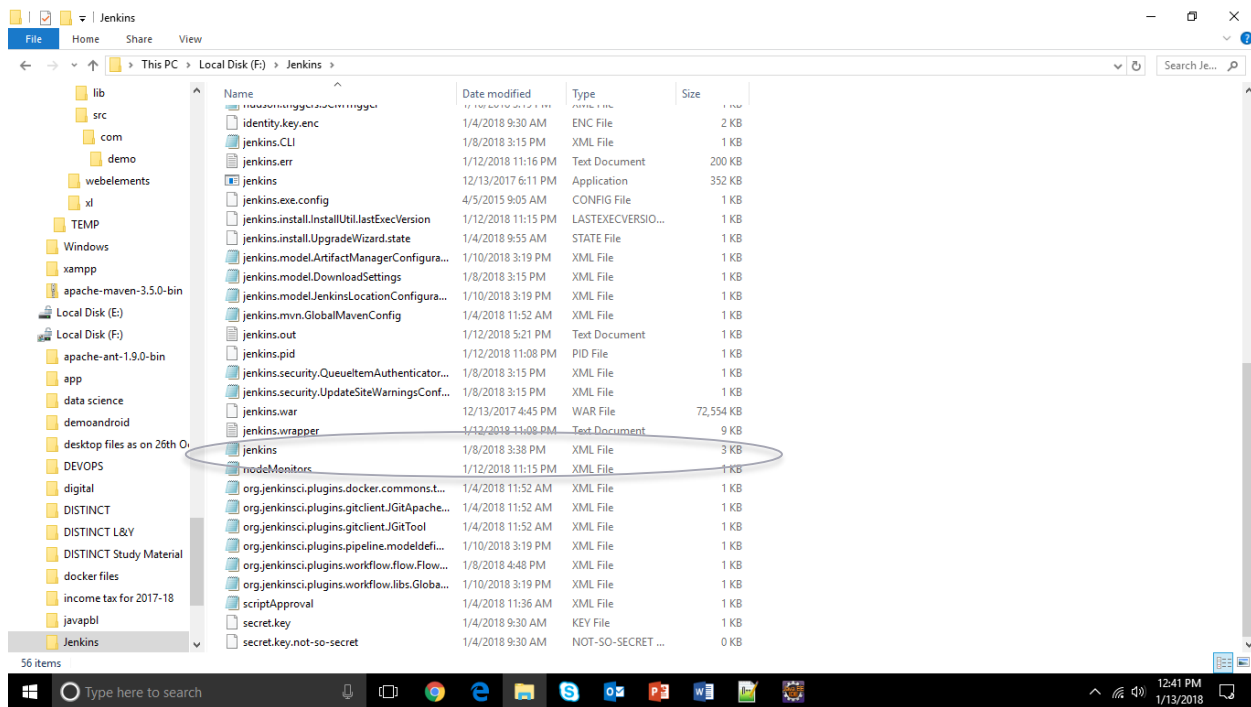
Create Jenkins Freestyle Job

- Configure Build using top level Maven goal with customizable email notification trigger at Job Level:

The screenshot shows the Jenkins configuration interface for a job named 'justdisplay'. The 'Build' tab is active, displaying the 'Invoke top-level Maven targets' step with the goal 'compile' circled. Below this, the 'Post-build Actions' section shows the 'E-mail Notification' step with the recipient 'raghavedendran.sethumadhavan1@wipro.com' circled. The 'Save' and 'Apply' buttons are visible at the bottom of the configuration area.

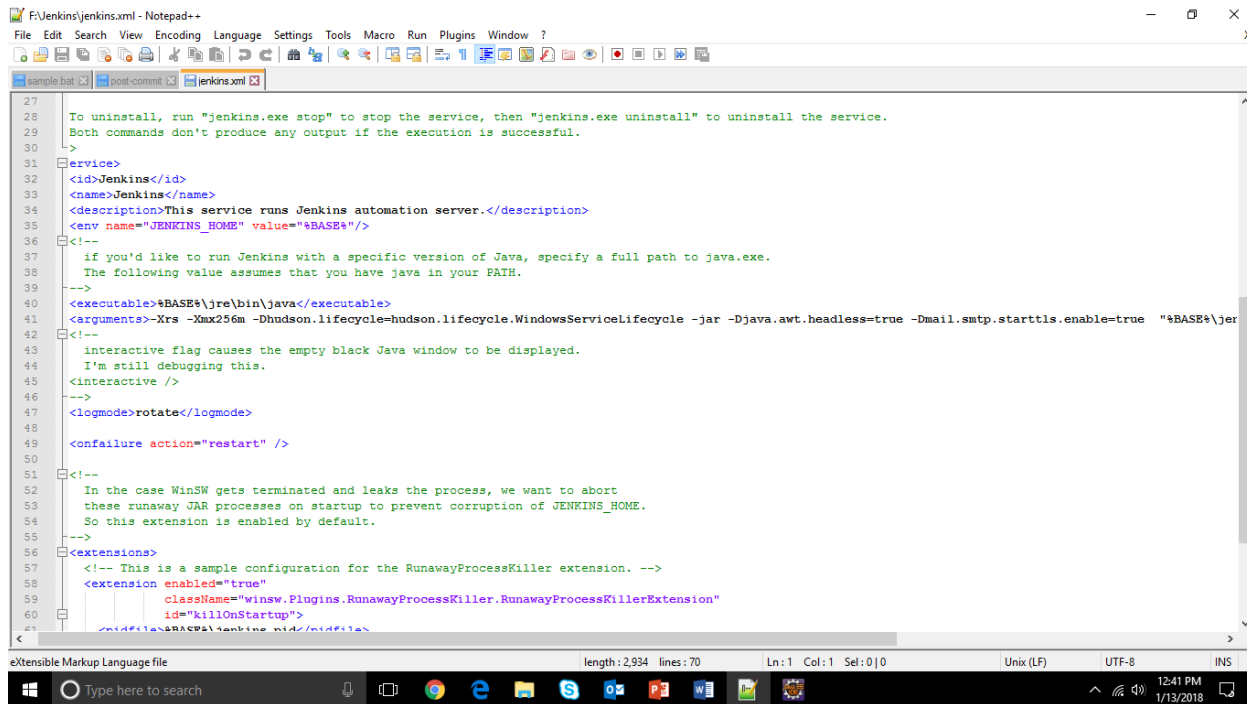
Update Jenkins.xml

- Make Changes in Jenkins.xml :



Update Jenkins.xml

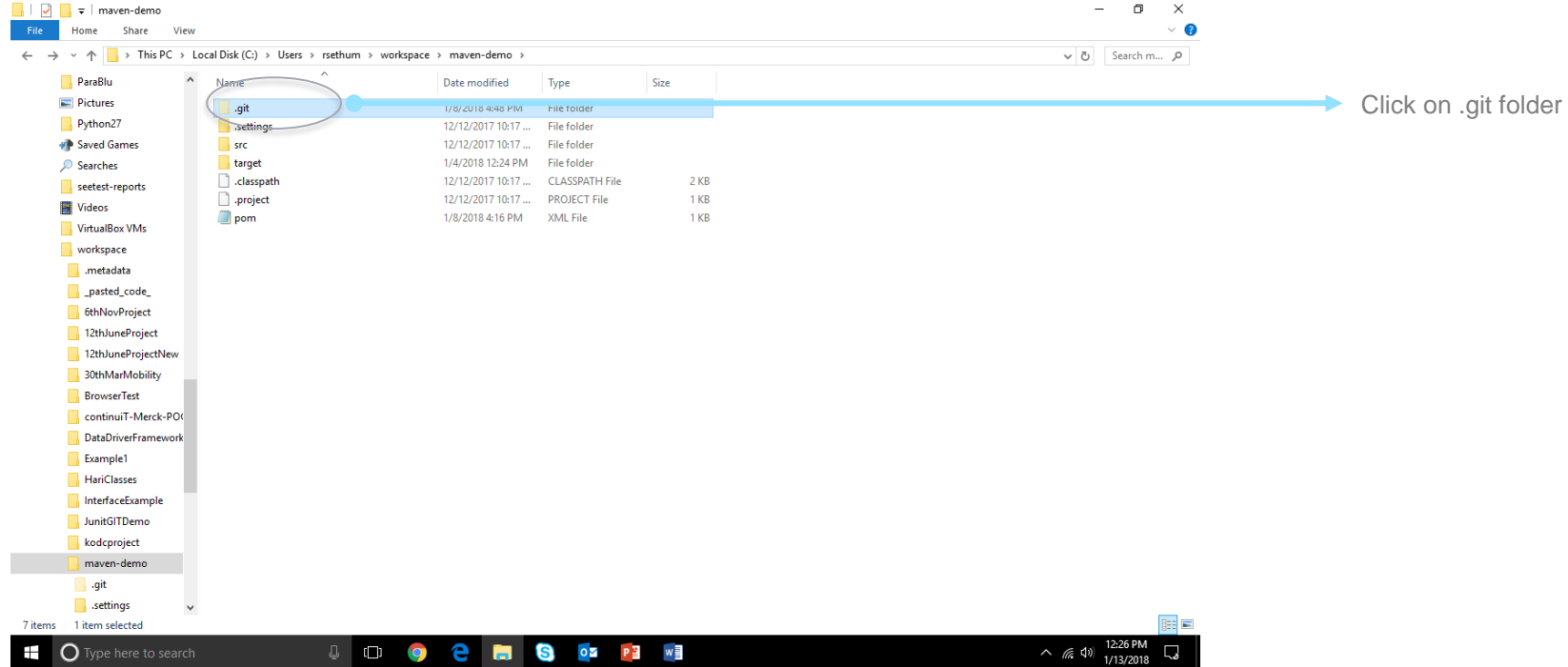
- Make Changes in Jenkins.xml :



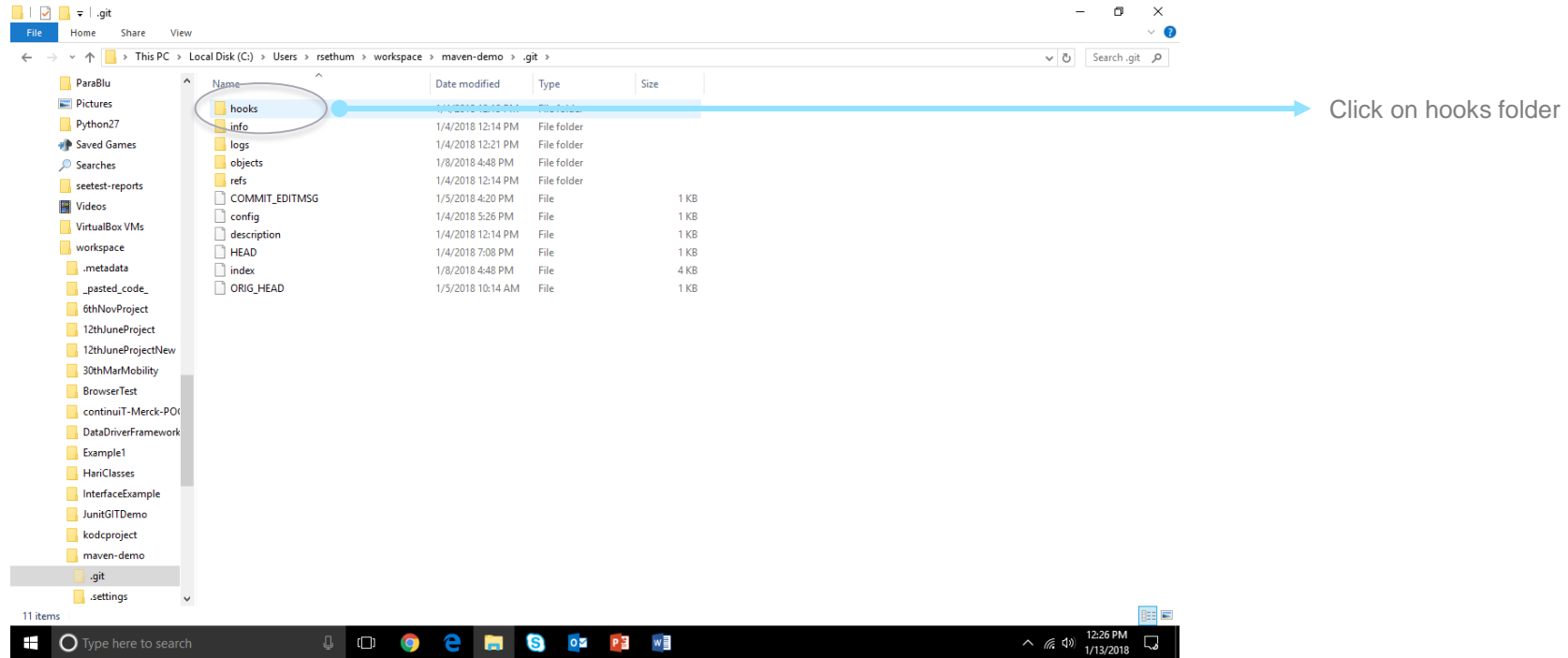
```
27
28 To uninstall, run "jenkins.exe stop" to stop the service, then "jenkins.exe uninstall" to uninstall the service.
29 Both commands don't produce any output if the execution is successful.
30 >
31 <service>
32 <id>Jenkins</id>
33 <name>Jenkins</name>
34 <description>This service runs Jenkins automation server.</description>
35 <env name="JENKINS_HOME" value="%BASE%" />
36 <!--
37 if you'd like to run Jenkins with a specific version of Java, specify a full path to java.exe.
38 The following value assumes that you have java in your PATH.
39 -->
40 <executable>%BASE%\jre\bin\java</executable>
41 <arguments>-Xrs -Xmx256m -Dhudson.lifecycle=jenkins.lifecycle.WindowsServiceLifecycle -jar -Djava.awt.headless=true -Dmail.smtp.starttls.enable=true "%BASE%\jer
42 <!--
43 interactive flag causes the empty black Java window to be displayed.
44 I'm still debugging this.
45 <interactive />
46 -->
47 <logmode>rotate</logmode>
48
49 <onfailure action="restart" />
50
51 <!--
52 In the case WinSW gets terminated and leaks the process, we want to abort
53 these runaway JAR processes on startup to prevent corruption of JENKINS_HOME.
54 So this extension is enabled by default.
55 -->
56 <extensions>
57 <!-- This is a sample configuration for the RunawayProcessKiller extension. -->
58 <extension enabled="true"
59     className="winsw.Plugins.RunawayProcessKiller.RunawayProcessKillerExtension"
60     id="killOnStartup">
61 </extension>
62 </extensions>
63 </service>
```

Add the below two parameters in
<Arguments> tag as indicated:
-Djava.awt.headless=true
-Dmail.smtp.starttls.enable=true

Create a hook in Git repository

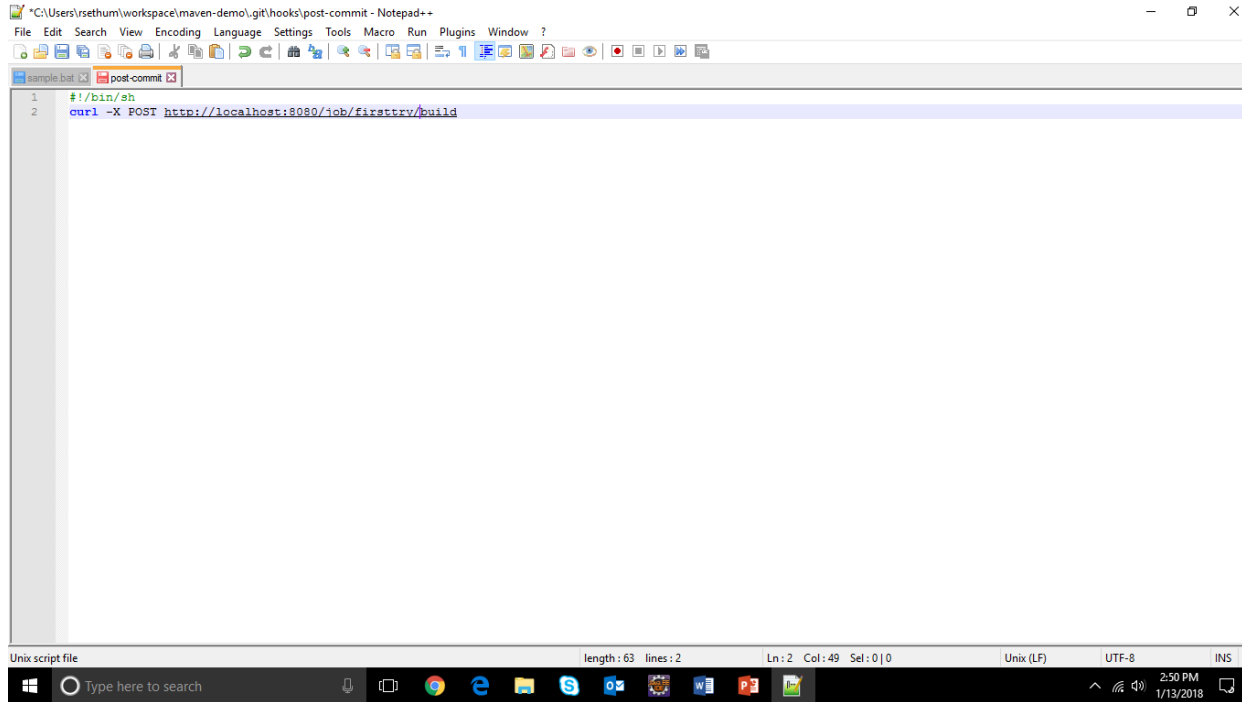


Create a hook in Git repository



Create a hook in Git repository

- Create a post-commit file

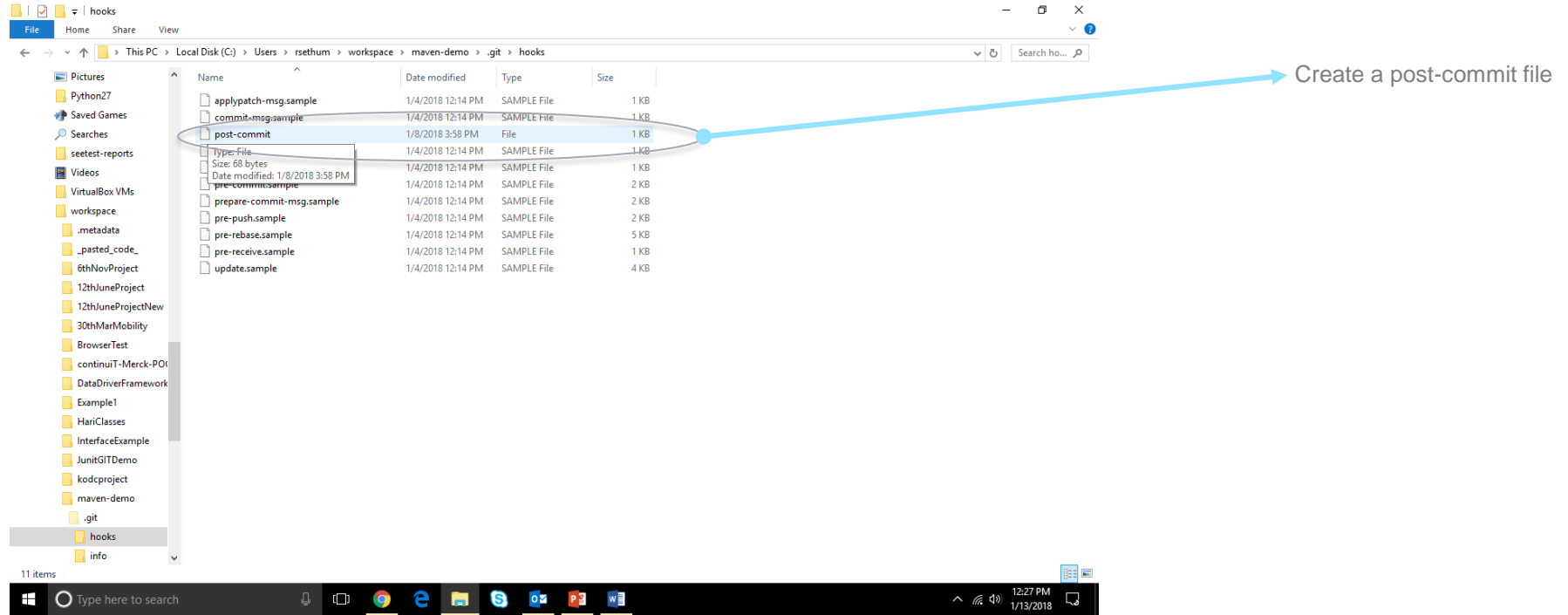


The screenshot shows a Notepad++ window titled "C:\Users\rsethum\workspace\maven-demo\git\hooks\post-commit - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. The toolbar contains various icons for file operations. The editor has two tabs: "sample.txt" and "post-commit", with "post-commit" being the active tab. The script content is as follows:

```
1 #!/bin/sh
2 curl -X POST http://localhost:8080/job/firestrv/build
```

The status bar at the bottom indicates "Unix script file", "length: 63 lines: 2", "Ln: 2 Col: 49 Sel: 0 | 0", "Unix (LF)", "UTF-8", and "INS". The Windows taskbar at the very bottom shows the search bar, task view icon, and several application icons (Chrome, Edge, File Explorer, Slack, Teams, etc.), along with the system clock showing 2:50 PM on 1/13/2018.

Create a hook in Git repository



Auto trigger of build by committing changes in Git

- After the configuration, by committing changes in Git repository, the build will be auto triggered as define

The screenshot shows the Jenkins web interface in a browser window. The address bar shows 'localhost:8080/job/firsttry/'. The Jenkins logo is in the top left, and a red status bar shows '3' builds. The left sidebar contains navigation links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now' (highlighted with a red circle), 'Delete Project', 'Configure', and 'GitHub Hook Log'. The main content area is titled 'Project justdisplay' with 'Project name: firsttry'. It includes links for 'add description', 'Disable Project', 'Workspace', and 'Recent Changes'. Below this is a 'Permalinks' section with a list of build links. At the bottom left is a 'Build History' table.

Build History

Build Number	Timestamp
#28	Jan 5, 2018 4:20 PM
#27	Jan 5, 2018 10:11 AM
#26	Jan 4, 2018 7:08 PM
#25	Jan 4, 2018 7:01 PM
#24	Jan 4, 2018 12:34 PM
#23	Jan 4, 2018 12:30 PM
#22	Jan 4, 2018 12:29 PM

Permalinks

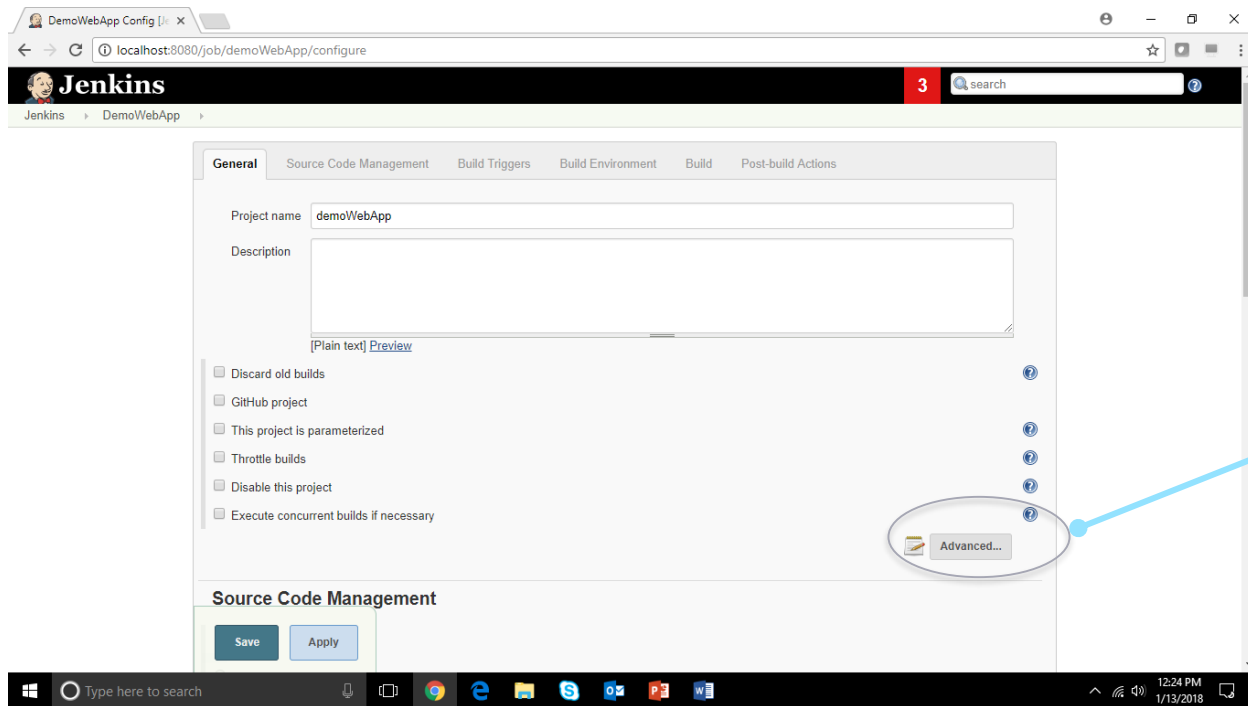
- [Last build \(#28\). 7 days 22 hr ago](#)
- [Last stable build \(#28\). 7 days 22 hr ago](#)
- [Last successful build \(#28\). 7 days 22 hr ago](#)
- [Last failed build \(#19\). 9 days 2 hr ago](#)
- [Last unsuccessful build \(#19\). 9 days 2 hr ago](#)
- [Last completed build \(#28\). 7 days 22 hr ago](#)

Build a selenium test job using batch file



Build a selenium test with a batch file

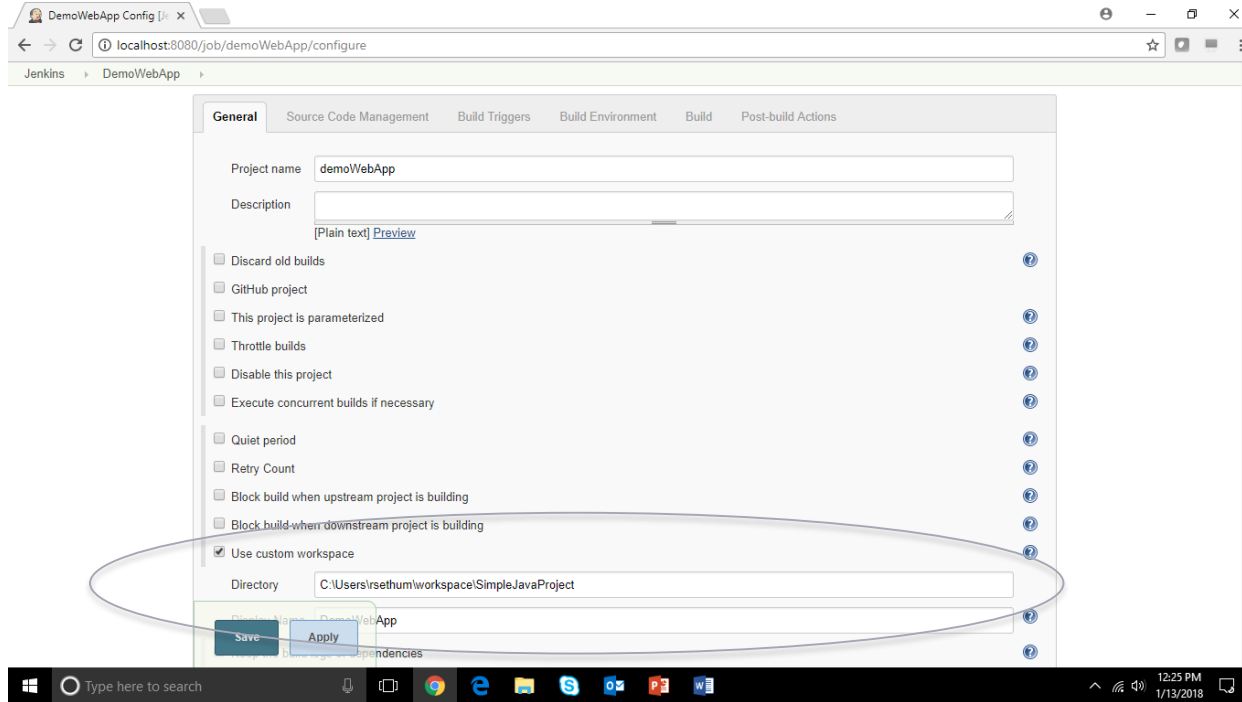
- Declare a customer workspace in a Freestyle job newly created:



Click on Advanced

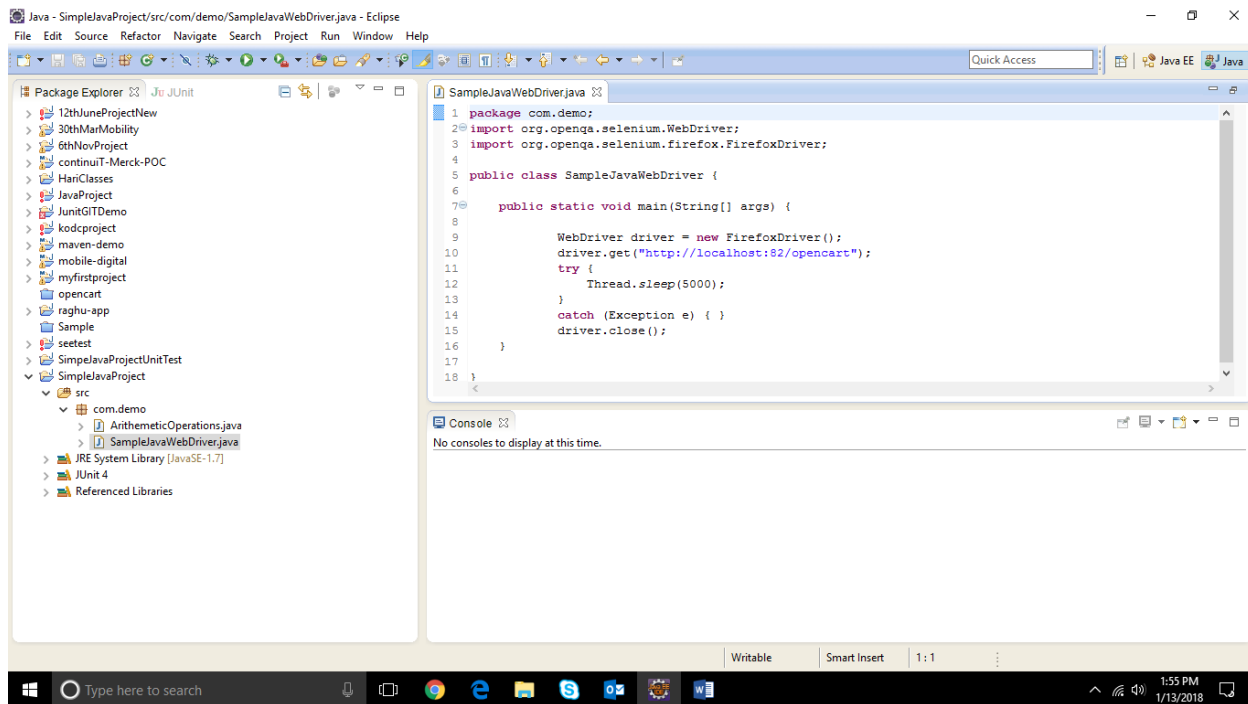
Build a selenium test with a batch file

- **Declare a customer workspace:**



Build a selenium test with a batch file

- Write a selenium test in Eclipse IDE



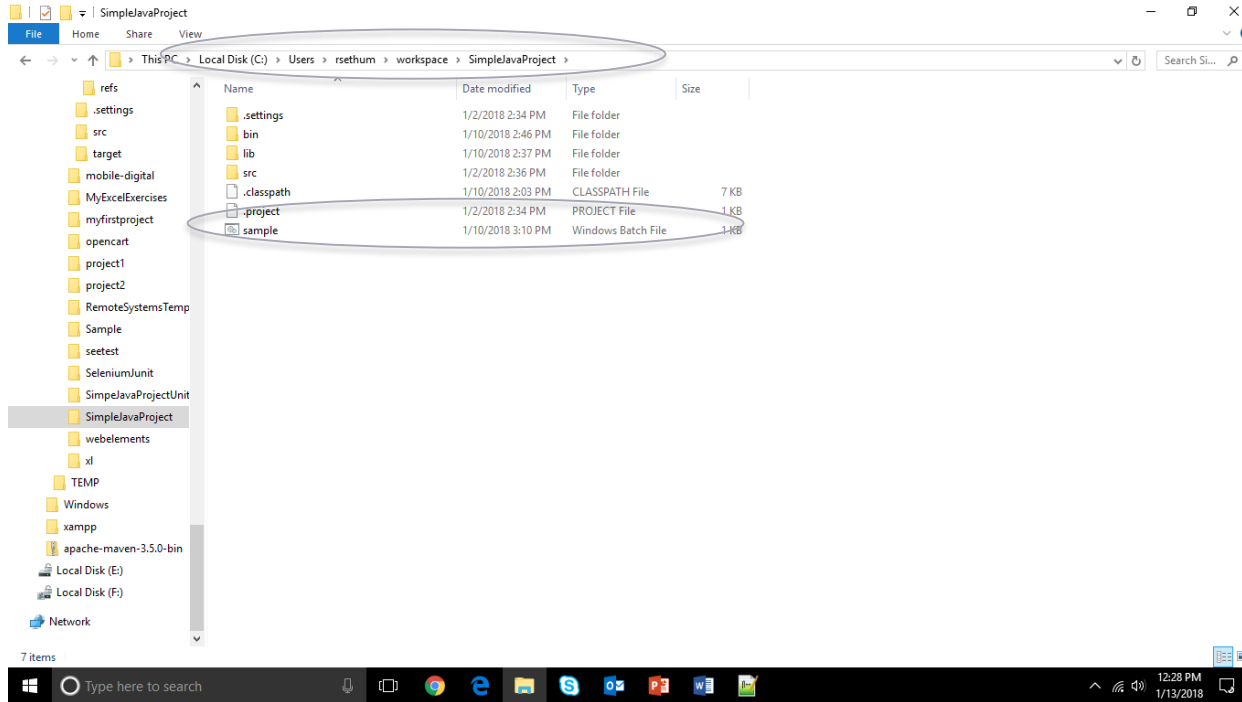
Build a selenium test with a batch file

- Write a batch file

```
1 java com.demo.SampleJavaWebDriver
```

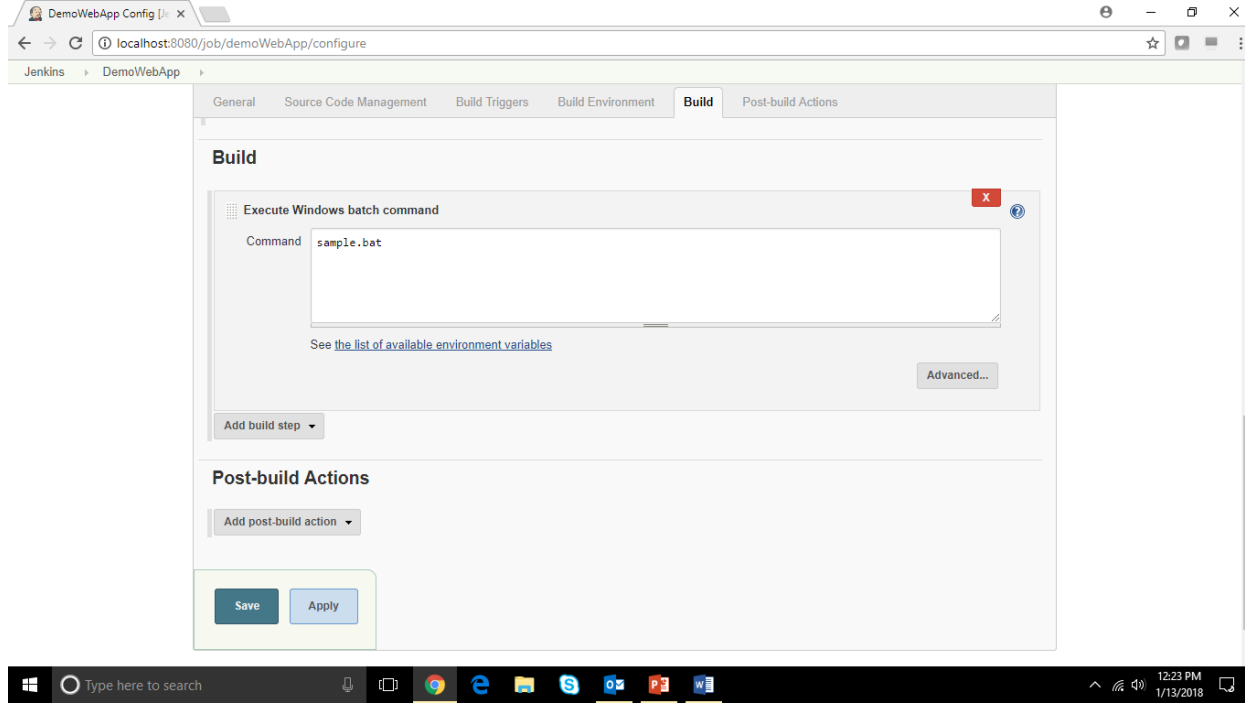
Build a selenium test with a batch file

- Store the batch file in the project folder defined as customer workspace



Build a selenium test with a batch file

- In the build section of the job, mention the name of the batch file created and then perform Build now operation in Jenkins.

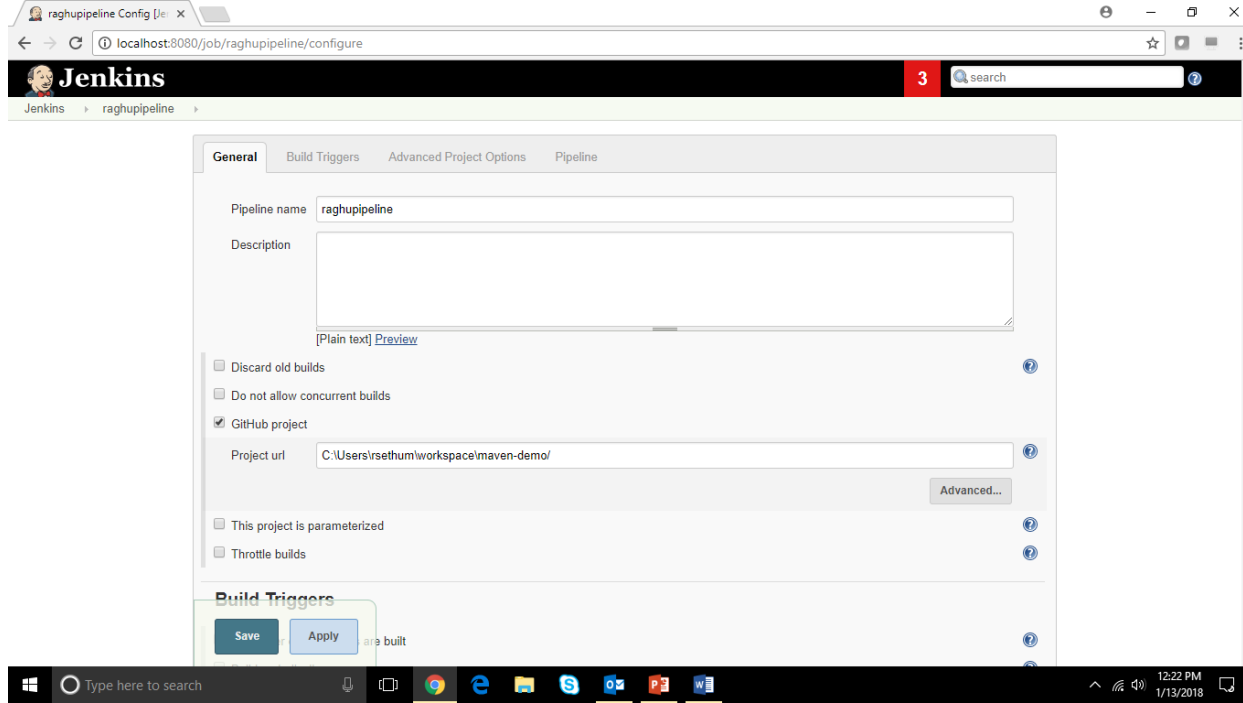


Create Jenkins Pipeline Job



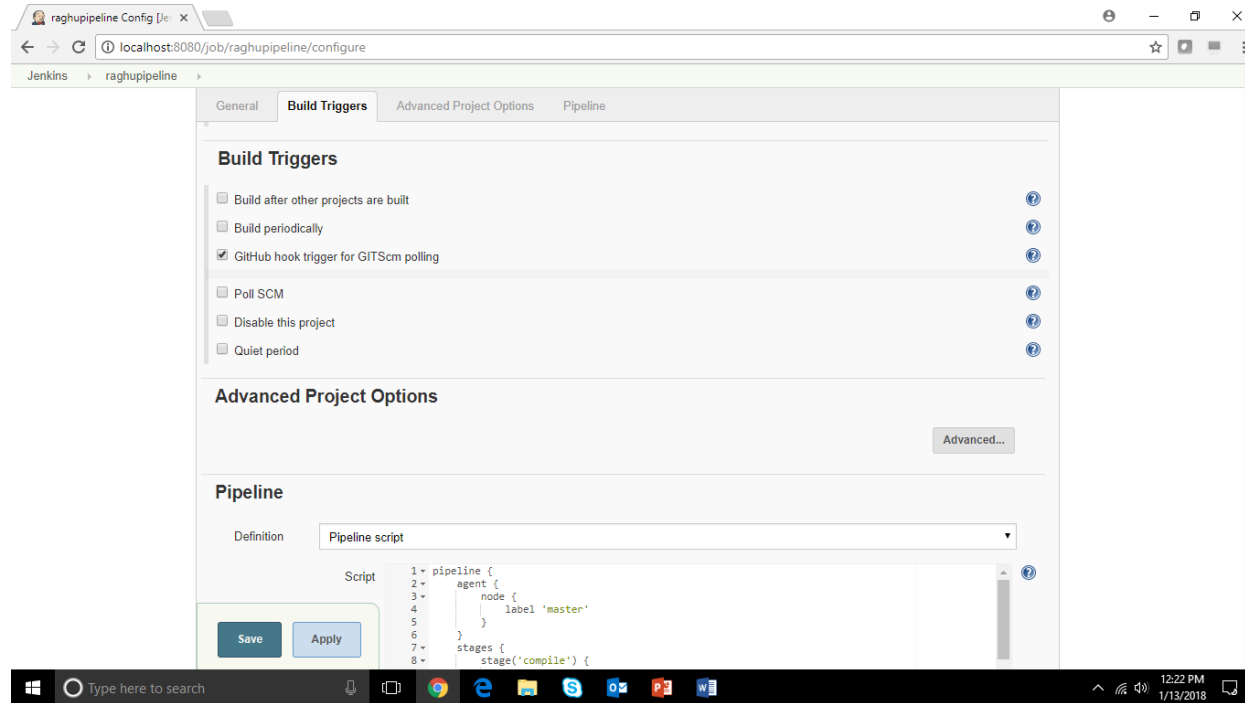
Create Jenkins Pipeline Job

- Create a pipeline job in jenkins



Create Jenkins Pipeline Job

- Configure the job to perform build whenever there is a new commit in Git repository



Create Jenkins Pipeline Job

- Write a simple Groovy script

The screenshot shows the Jenkins web interface for configuring a pipeline job. The browser address bar shows 'localhost:8080/job/raghupipeline/configure'. The 'Pipeline' tab is selected, and the 'Definition' is set to 'Pipeline script'. The Groovy script is as follows:

```
1 pipeline {  
2   agent {  
3     node {  
4       label 'master'  
5     }  
6   }  
7   stages {  
8     stage('compile') {  
9       steps {  
10        bat 'mvn -f c:/Users/rsethum/workspace/maven-demo/pom.xml compile'  
11        echo "Welcome to compile"  
12      }  
13    }  
14  }  
15  stage('Unit Testing') {  
16    // ...  
17  }
```

Below the script, the checkbox 'Use Groovy Sandbox' is checked. There are 'Save' and 'Apply' buttons at the bottom left of the configuration area. The Windows taskbar at the bottom shows the time as 12:22 PM on 1/13/2018.

Create Jenkins Pipeline Job

- Write a groovy script (Continued)

```
pipeline {  
    agent {  
        node {  
            label 'master'  
        }  
    }  
    stages {  
        stage('compile') {  
            steps {  
                bat 'mvn -f c:/Users/rsethum/workspace/maven-demo/pom.xml compile'  
                echo "Welcome to compile"  
            }  
        }  
    }  
}
```

Create Jenkins Pipeline Job

- Write a groovy script (Continued)

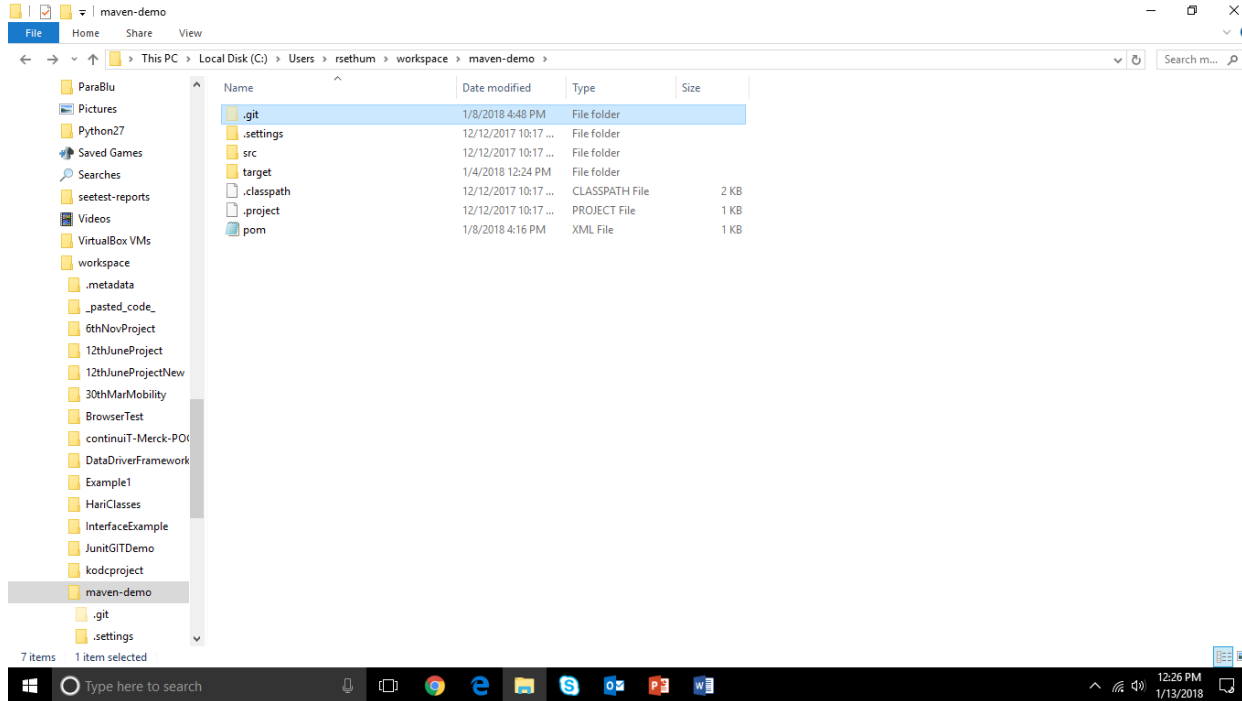
```
    stage('Unit Testing') {
    steps {

        bat 'mvn -f c:/Users/rsethum/workspace/maven-demo/pom.xml test'
        echo "Welcome to Unit Testing"
    }
}

post {
    always {
        echo 'it works always'
    }
    success {
        echo 'It will get executed only if successful'
    }
    failure {
        echo 'It will get executed only if failed'
    }
}
}
```

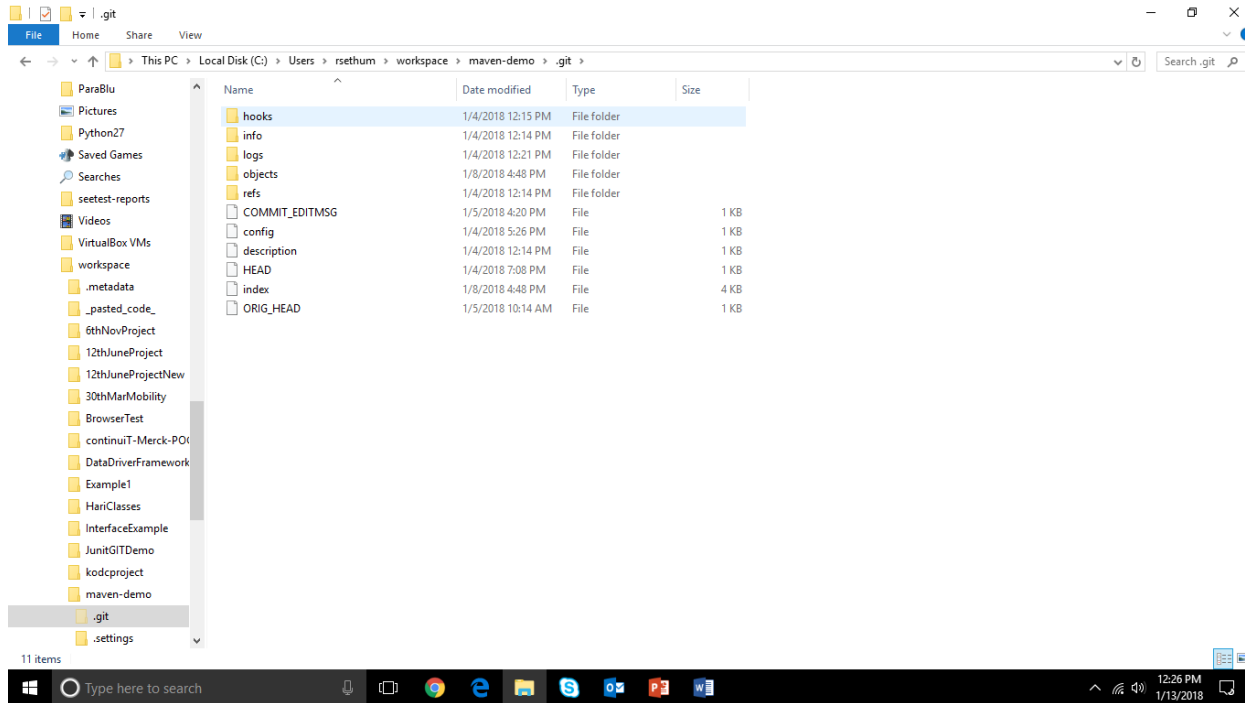
Create Jenkins Pipeline Job

- In the git repository create the post-commit file



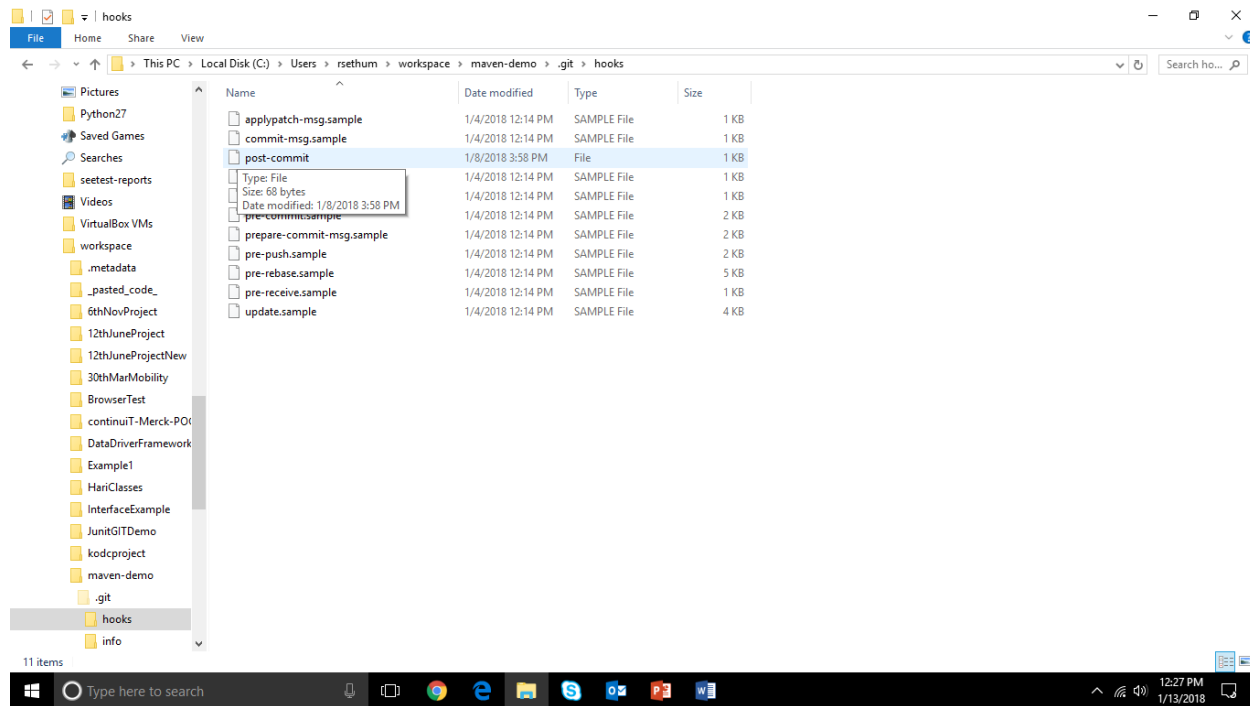
Create Jenkins Pipeline Job

- In the git repository create the post-commit file (Continued)



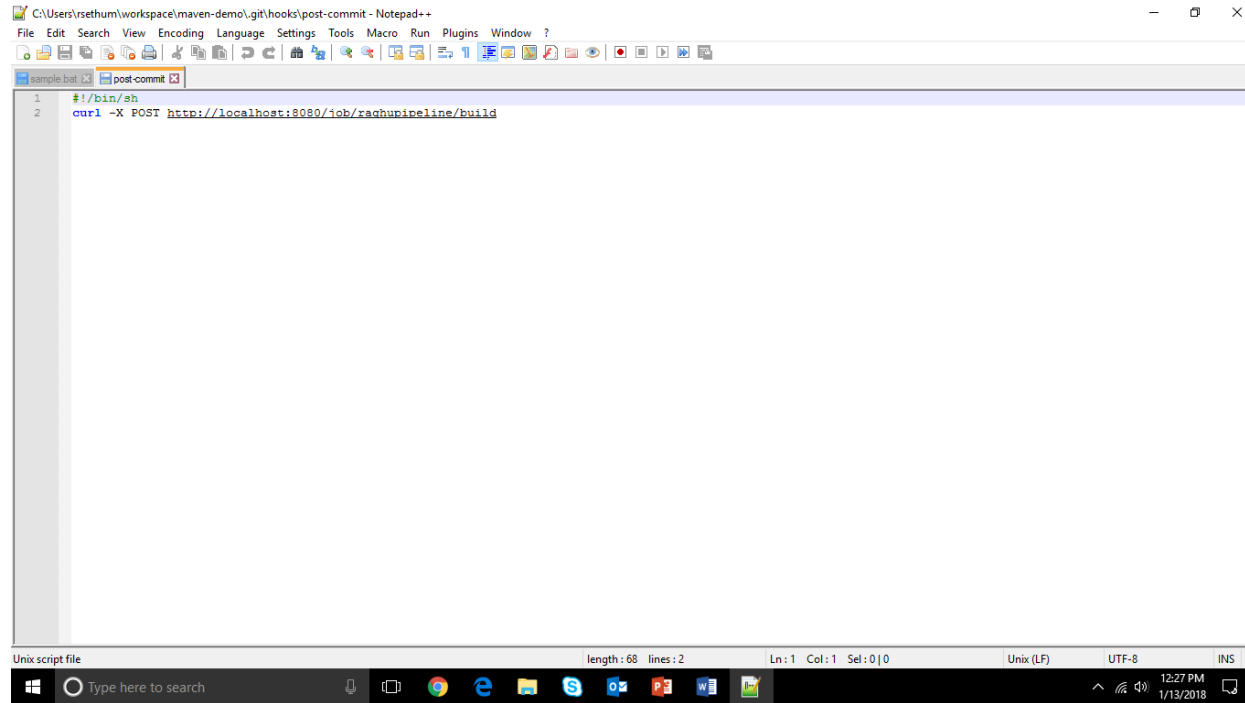
Create Jenkins Pipeline Job

- In the git repository create the post-commit file (Continued)



Create Jenkins Pipeline Job

- In the git repository create the post-commit file (Continued)



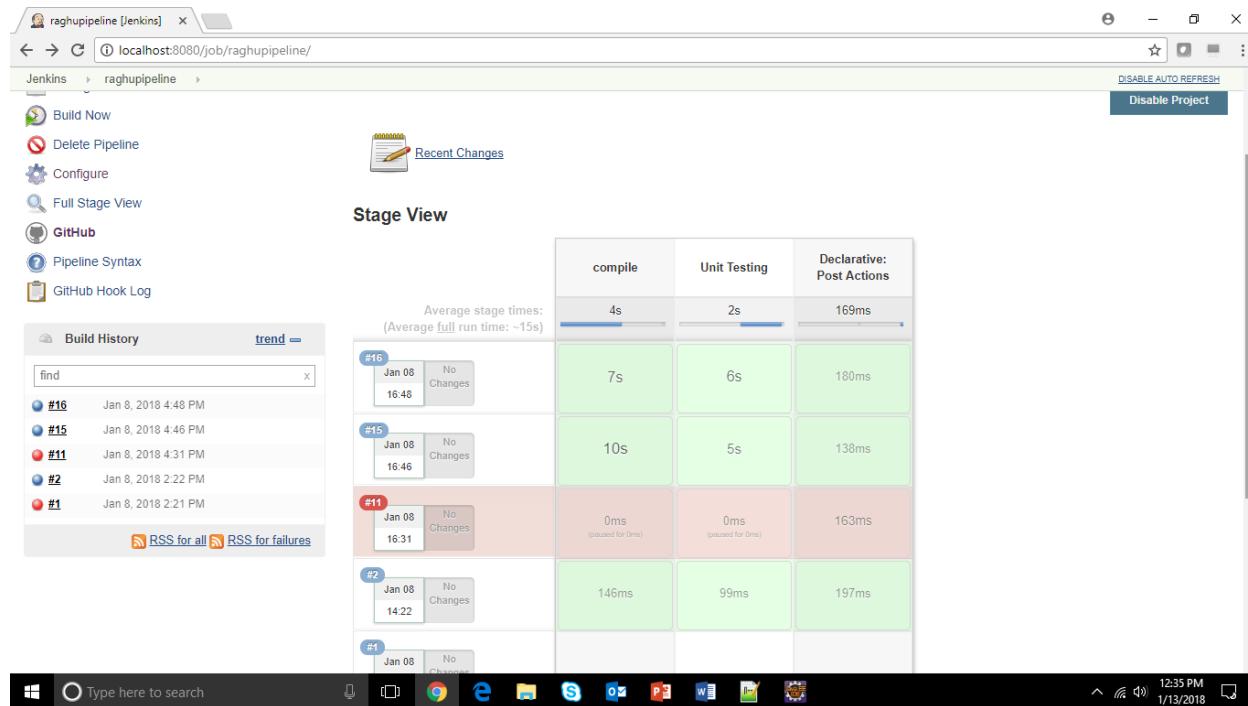
The screenshot shows a Notepad++ window titled "C:\Users\rsethum\workspace\maven-demo\git\hooks\post-commit - Notepad++". The window contains a file named "post-commit" with the following content:

```
1 #!/bin/sh
2 curl -X POST http://localhost:8080/job/raghuipipeline/build
```

The status bar at the bottom indicates "Unix script file", "length: 68 lines: 2", "Ln: 1 Col: 1 Sel: 0 | 0", "Unix (LF)", "UTF-8", and "INS". The Windows taskbar is visible at the bottom with the search bar and various application icons.

Create Jenkins Pipeline Job

Commit a new change in the Git repository. It would trigger a new build. You can monitor different stages of build process as demonstrated in the below picture



Create Jenkins Pipeline Job

You could also view the console output as given below



The screenshot shows the Jenkins web interface in a browser window. The address bar shows the URL `localhost:8080/job/raghupipeline/16/console`. The Jenkins header is visible with a red status indicator showing '3' and a search bar. The left sidebar contains navigation links: 'Back to Project', 'Status', 'Changes', 'Console Output' (selected), 'View as plain text', 'Edit Build Information', 'Delete Build', 'Replay', 'Pipeline Steps', and 'Previous Build'. The main content area is titled 'Console Output' and displays the following log:

```
Started by user anonymous
[Pipeline] node
Running on jenkins in F:\Jenkins\workspace\raghupipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (compile)
[Pipeline] bat
[raghupipeline] Running batch script

F:\Jenkins\workspace\raghupipeline>mvn -f c:/Users/rsethum/workspace/maven-demo/pom.xml compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building maven-demo 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ maven-demo ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ maven-demo ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to c:\Users\rsethum\workspace\maven-demo\target\classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 3.826 s
[INFO]
```

The Windows taskbar at the bottom shows the system clock as 12:36 PM on 1/13/2018.

References:

1. <http://www.javaworld.com/javaworld>



Thank You

Avinash Patel

Senior Manager

avinash.patel@wipro.com