

# Module 4: Continuous Integration Using Jenkins

---

Demo Document

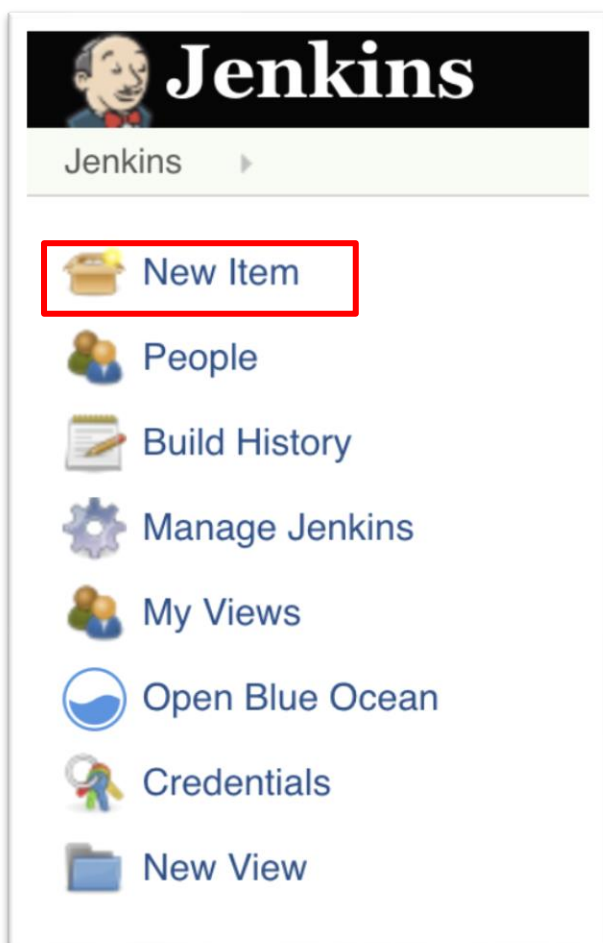
edureka!

**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.


## Demo 2: Build a Pipeline project using Groovy Script.


**Step1:** Create a Pipeline Project by navigating to Jenkins -> New Item





**Step-2** Create a Pipeline project for your job.


**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.

**External Job**  
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

**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.

**Step-3:** Go to your project and start configuring it.

**Step-4:** Go to the Advanced Project option and write your code within the Pipeline definition. The code is mentioned below if you want to copy and paste the code to generate the pipeline. Replace the “some email id” with the email\_id you want to send email to if the build fails.

**Advanced Project Options**

Advanced...

**Pipeline**

Definition: Pipeline script

Script

```
1 #!/usr/bin/env groovy
2
3 import hudson.model.*
4 import hudson.EnvVars
5 import groovy.json.JsonSlurperClassic
6 import groovy.json.JsonBuilder
7 import groovy.json.JsonOutput
8 import java.net.URL
9
10 try {
11
12     node{
13
14         stage('Checkout') {
15             git 'https://github.com/edureka-git/DevOpsClassCodes'
16
17         }
18     }
19 }
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Save Apply

```
#!/usr/bin/env groovy
```

```
import hudson.model.*
```

```
import hudson.EnvVars
```

```
import groovy.json.JsonSlurperClassic
```

```
import groovy.json.JsonBuilder
```

```
import groovy.json.JsonOutput
```

```
import java.net.URL
```

```
try {
```

```
node{
```

```
    stage('Checkout') {
```

```
        git 'https://github.com/edureka-git/DevOpsClassCodes'
```

```
}

stage('Build') {
    dir('') {
        sh 'mvn -B -V -U -e clean package'
    }
}

stage ('Email') {
    emailx attachLog: true, body: 'The status of the build can be obtained
from the build log attached', subject: 'The build update is ', to: 'some
email id'
}

stage('Deployment') {
    // Deployment
    script {
        echo "deployment"

        sh 'cp
/var/lib/jenkins/workspace/package_1/target/addressbook.war
/opt/tomcat/webapps/'
    }
}

stage('publish html report') {
    echo "publishing the html report"

    publishHTML([allowMissing: false, alwaysLinkToLastBuild:
false, keepAll: false, reportDir: '', reportFiles: 'index.html', reportName:
'HTML Report', reportTitles: ''])
}

stage('clean up') {
    echo "cleaning up the workspace"

    cleanWs()
}
```

```

} // node
} // try end
finally {

    (currentBuild.result != "ABORTED") && node("master") {

        // Send e-mail notifications for failed or unstable builds.
        // currentBuild.result must be non-null for this step to work.
        step([$class: 'Mailer',

            notifyEveryUnstableBuild: true,

            recipients: 'some email id',

            sendToIndividuals: true])

    }

}

```

**Step 5.** If you are getting any error on deployment stage, make sure you are copying the war file to webapps folder correctly i.e., provide the war file location and tomcat directory path correctly.

**Step 6.** Then save the code and click on build now to build the project.

**Step 7.** If you're getting some error related to html report then go to manage jenkins->manage plugins->available and install html publisher plugin.

**Step 8.** Go to the console output and check the build of your project.

**Jenkins** 2  shubham | log out

Jenkins > Devopsaac1 > #31

**Console Output**

```

Started by user shubham
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Devopsaac1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
> /usr/bin/git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> /usr/bin/git config remote.origin.url https://github.com/edureka-git/DevOpsClassCodes # timeout=10
Fetching upstream changes from https://github.com/edureka-git/DevOpsClassCodes
> /usr/bin/git --version # timeout=10
> /usr/bin/git fetch --tags --progress https://github.com/edureka-git/DevOpsClassCodes +refs/heads/*:refs/remotes/origin/* # timeout=10
> /usr/bin/git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> /usr/bin/git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision c986b9eb275ae50c5a7b9b1d757be5c58e38c0da (refs/remotes/origin/master)
> /usr/bin/git config core.sparsecheckout # timeout=10
> /usr/bin/git checkout -f c986b9eb275ae50c5a7b9b1d757be5c58e38c0da
> /usr/bin/git branch -a -v --no-abbrev # timeout=10
> /usr/bin/git branch -D master # timeout=10
> /usr/bin/git checkout -b master c986b9eb275ae50c5a7b9b1d757be5c58e38c0da
Commit message: "initial commit"
> /usr/bin/git rev-list --no-walk c986b9eb275ae50c5a7b9b1d757be5c58e38c0da # timeout=10
[Pipeline] }

```

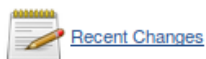
```

[Pipeline] stage
[Pipeline] { (publish html report)
[Pipeline] echo
publishing the html report
[Pipeline] publishHTML
[htmlpublisher] Archiving HTML reports...
[htmlpublisher] Archiving at PROJECT level /var/lib/jenkins/workspace/Devopsaac1 to /var/lib/jenkins/jobs/Devopsaac1/htmlreports/HTML_20Report
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (clean up)
[Pipeline] echo
cleaning up the workspace
[Pipeline] cleanWs
[WS-CLEANUP] Deleting project workspace...[WS-CLEANUP] done
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Devopsaac1
[Pipeline] {
[Pipeline] step
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

**Step-9:** Go back to your project and check the pipeline flow.

## Pipeline Devopsaac1



[Recent Changes](#)

### Stage View

	Checkout	Build	Email	Deployment	publish html report	clean up
Average stage times: (Average full run time: ~41s)	4s	11s	23s	338ms	239ms	111ms
#31 Oct23 17:04 No Changes	4s	15s	1min 0s	569ms	404ms	200ms