

- 1 --Create and manage user roles etc
- 2 33. Login to jenkins
- 3 34. click manage jenkins
- 4 35. click manage users
- 5 36. from left pane click Create User
- 6 37. Enter relevant data.
- 7 do this for say 2 users user1 and user2
- 8
- 9 ---manage access and roles
- 10 38. click manage jenkins
- 11 39. manage plugins
- 12 40. click Available
- 13 41. Search Role-based authorization strategy
- 14 45. select and click on install without restart
- 15 46. After plugin is installed Goto Manage Jenkins
- 16 47. configure global security
- 17 48. Under Authorization Select Role Based Strategy
- 18 49. Markup Formatter Select Safe Html
- 19 50. Apply and Save
- 20 51. Logout as admin
- 21 52 Try Logging in as user1 or user2
- 22 53. logout and login as admin
- 23 54. Manage Jenkins
- 24 55. click Manage and assign roles
- 25 56. Click Manage Roles
- 26 57. Role To Add Enter employees click add button
- 27 58. Select Read a ccess from overall and all access from view
- 28 59. For Item Role create roles developer/tester with patterns .\*Dev/. \*Test  
respectively Add both the roles and patterns by clicking on add each time u create  
the role
- 29 60. Assign all access to developer/tester roles
- 30 61. Click Apply and save
- 31 62. Click Assign Roles
- 32 63. under global roles, in user/group to add enter user1 and click add, enter user2  
and click add
- 33 64. select employees checkbox for both
- 34 65. Under item roles section, in user/group to add enter user1 and user2 adding each
- 35 66. select developer checkbox for user1 and tester checkbox for user2.
- 36 67. click on apply and save
- 37 68. Goto dashboard top left corner
- 38 69. Click on new Item
- 39 70. Name Project1Dev
- 40 71. Select Freestyle project
- 41 72. Click OK
- 42 73. Click Apply Save
- 43 74. Click on new Item
- 44 75. Name Project1Test
- 45 76. Select Freestyle project

46 77. Click OK  
47 78. Click Apply Save  
48 79. logout and login as user1 and user2 see the dashboard :) :)  
49 -----  
50 --- Trigger a build for a project  
51 --- If not already started Start Jenkins java -jar jenkins.war  
52 --- Browse to localhost:8080/ login with username/password  
53 --- Click on New Item in the left side pane  
54 --- Name Project2Dev  
55 --- select freestyle project  
56 --- click on OK  
57 --- On the page that opens In GENERAL tab  
58 --- Description : <h2>This is Project 2 For Dev</h2>  
59 --- Source code Management tab Select None  
60 --- Build Trigger tab Select Build Periodically set cron as \* \* \* \* \*  
61 --- select Add build step dropdown and select windows batch command and type in  
command Date /t in text box  
62 --- Post-Build Actions tab : Do nothing As of Now  
63 --- Click Apply and save  
64 --- Goto dashboard  
65 --- Select dropdown of the project and select build observe the changes to the  
columns wherever you see them (refresh browser)  
66 --- Goto your project --- configure --- change the windows command from date to  
dat /t apply and save  
67 --- Build the project again and observe the changes in column icons  
68 --- Build a few more times observe changes in icons  
69  
70 --- Remote Build of Project  
71 --- Goto Project2Dev --- Configure --- Build trigger --- uncheck the existing ---  
Check Trigger Bulds remotely  
72 --- Authentication Token : Am12Ey34a (This could be anything)  
73 --- Apply and save  
74 --- copy the url : JENKINS\_URL/job/Project2Dev/build?token=TOKEN\_NAME  
75 --- Modify it as : <http://localhost:8080/job/Project2Dev/build?token=Am12Ey34a>  
76 --- Invoke it from browser  
77 --- In the jenkins browser you will see that Build Queue has 1 entry (If not  
refreshed) refresh this browser  
78 --- Explore Build History from left pane  
79  
80 --- Build After other projects are built  
81 --- create a new freestyle project Project3Dev  
82 --- Goto build Trigger --- Check Build After other projects are built --- enter  
project names separated by commas : Project2Dev  
83 --- Select the radio : Trigger only if build is stable  
84 --- Goto build --- Add build step --- windows batch command--- type time /t  
85 --- Apply and Save  
86 --- create a new freestyle project Project4Dev  
87 --- Click on OK --- Apply and Save

```
88 --- Go back to Project3Dev
89 --- Configure
90 --- Post-build Action
91 --- Add Post-Build Action --- Build Other Project --- Type Project4Dev
92 --- Apply and Save
93 --- Goto dashboard in browser
94 --- Click on Project3Dev see that there is no build history for Project
95 --- Also observe the Upstream and downstream Projects
96 --- Do the same for Project4Dev
97 --- Now Trigger the build for Project2Dev (Either remotely or locally)
98 --- Observe the Build Queue (You might need to refresh the page)
99
100 -----
101 -----
102 --- Build After other projects are built
103 --- create a new freestyle project Project3Dev
104 --- Goto build Trigger --- Check Build After other projects are built --- enter
    project names separated by commas : Project2Dev
105 --- Select the radio : Trigger only if build is stable
106 --- Goto build --- Add build step --- windows batch command--- type time /t
107 --- Apply and Save
108 --- create a new freestyle project Project4Dev
109 --- Click on OK --- Apply and Save
110 --- Go back to Project3Dev
111 --- Configure
112 --- Post-build Action
113 --- Add Post-Build Action --- Build Other Project --- Type Project4Dev
114 --- Apply and Save
115 --- Goto dashboard in browser
116 --- Click on Project3Dev see that there is no build history for Project
117 --- Also observe the Upstream and downstream Projects
118 --- Do the same for Project4Dev
119 --- Now Trigger the build for Project2Dev (Either remotely or locally)
120 --- Observe the Build Queue (You might need to refresh the page)
121
122 --- Git-Jenkins Integration
123 --- Create a Github/bitbucket account and a remote repo GIT-DO also create
    .gitignore file for java template
124 (REFER TO GIT TRAINING MATERIAL)
125 --- Clone the remote-repo with the local-repo (Below is folder structure)
126 C:\work\sapient\
127     \Devops
128     \Git-DO
129     \.git
130 --- Ensure GIT is installed and the current ws is in local repo
131 --- start eclipse and create new java project in the ws within git repo
132 (I.E. folder hierarchy is as follows)
```

```
133 C:\work\sapient\  
134     \Devops  
135         \Git-DO  
136             \.git -->Local repo  
137             \javaws --> Your ws  
138                 \Project1Dev --> Your Project  
139 Replace the .gitignore contents with following  
140 # Compiled class file  
141 *.class  
142  
143 # Log file  
144 *.log  
145  
146 # BlueJ files  
147 *.ctxt  
148  
149 # Mobile Tools for Java (J2ME)  
150 .mtj.tmp/  
151  
152 # Package Files #  
153 *.jar  
154 *.war  
155 *.nar  
156 *.ear  
157 *.zip  
158 *.tar.gz  
159 *.rar  
160  
161 # virtual machine crash logs, see  
    http://www.java.com/en/download/help/error\_hotspot.xml  
162 hs_err_pid*  
163  
164 # eclipse specific  
165 *.pydevproject  
166 .project  
167 .metadata  
168 bin/**  
169 tmp/**  
170 tmp/**/*  
171 *.tmp  
172 *.bak  
173 *.swp  
174 *~.nib  
175 local.properties  
176 .classpath  
177 .settings/  
178 .loadpath  
179
```

```

180 # External tool builders
181 .externalToolBuilders/
182
183 # Locally stored "Eclipse launch configurations"
184 *.launch
185
186 --- open the git bash or cmd in GIT-DO folder (this is local repo)
187 --- run git status
188 --- Observe the status
189 --- run git add .
190 --- Create a com.ameya.TestMain.java as follows
191 ackage com.ameya;
192
193 public class TestMain {
194
195     public static void main(String[] args) {
196         System.out.println(" Hello Wolrd");
197
198     }
199
200 }
201 --- run git status
202 --- run git add .
203 --- run git commit -m "Added TestMain.java with Helloworld sysout"
204 --- run git push origin master
205
206 --- goto jenkins --- manage jenkins --- configure system --- Git Plugin --- provide
name and email
207 --- goto jenkins --- manage jenkins --- global tools configuration --- path to git
must be path to your git installation folder/bin/git.exe
208 --- goto Project1Dev --- Configure --- Source code Management --- select Git
209 --- Add repositories : https://github.com/amsalways/Git-DO.git
210 ---Credentials --- Add --- Jenkins --- username : bitbucket username password
bitbucket password --- click add
211 --- In dropdown for credentials select newly added credentials
212 --- goto Build Triggers
213 --- select poll SCM
214 --- enter cron expression as * * * * *
215 ---Add Build Step --- windows batch command--- add below as per your path
216 cd C:\work\sapient\Devops\GIT-DO\javaws\Project1Dev\src\com\ameya
217 javac -d C:\work\sapient\Devops\GIT-DO\javaws\Project1Dev\bin *.java
218 cd C:\work\sapient\Devops\GIT-DO\javaws\Project1Dev\bin
219 java com.ameya.TestMain
220 ---Add PostBuild Action --- Build Other Projects --- Type Project2Dev ---Trigger
only if build is stable
221 --- Apply and save
222 ----- Now Goto your java project
223 --- Add following NewTestClass.java

```

```

224 package com.ameya;
225
226 public class NewTestClass {
227     public NewTestClass() {
228         System.out.println("New Test Class Object created");
229     }
230 }
231 --- Modify TestMain.java
232 package com.ameya;
233
234 public class TestMain {
235
236     public static void main(String[] args) {
237         System.out.println(" Hello Wolrd");
238         NewTestClass nt=new NewTestClass();
239     }
240
241 }
242 --- git add .
243 --- git commit -m "Added NewTestClass.java and Instantiated in TestMain.java"
244 --- git push origin master
245
246 --- Goto Jenkins and see in browser a new Build will be triggered
247 --- Click on the build you will see all the projects being built
248
249 -----
250
251 --- Install delivery pipeline plugin
252 --- Manage jenkins --- manage plugins --- available --- delivery pipeline ---
    Dwnlod & install after restart
253 --- restart jenkins --- login again as username/password
254 --- Configure Delivery pipeline view
255 --- Click on the + on the dashboard just next to All above the jobs view
256 --- View name : Project1DeliveryPipeline
257 --- Select Delivery Pipeline View Radio button
258 --- Click OK
259 --- In the configurations view scroll down to pipelines sections
260 --- Add Component---Component Name Project1DevDeliveryPipeline ---Initial Job
    Project1Dev
261 ---Go back to Dashboard
262 --- In Eclipse Add code To TestMain.java
263 --- git add .
264 --- git commit -m "New Functionality Added"
265 --- git push origin master
266 ---Observe the pipeline in Jenkins
267 ---You can also trigger a build by clicking the button on top with a circle and triangle
    icon

```

268 --- Create a Build Pipeline  
269 --- Manage Jenkins --- Manage plugins --- available --- Build pipeline ---- Dwnlod  
& install after restart  
270 --- restart jenkins  
271 --- Click on the + on the dashboard just next to All above the jobs view  
272 --- View name : Project1BuildPipeline  
273 --- Select Build Pipeline View Radio button  
274 --- Click OK  
275 --- On the configuration page --- scroll down  
276 select InitialJob : Project1Dev  
277 --- Click Apply and OK  
278