

# Jenkins Exercise

## Managing Jenkins

Configure Jenkins

- Install all the suggested plugins when you first login as admin user.

## User/Role Mgmt.

User Management	Manage Jenkins -> Manage Users
Manage & Assign Roles	<ul style="list-style-type: none"><li>• Install Role-based authorization Strategy Plugin</li><li>• Configure Global Security -&gt; Authorization -&gt; Role based strategy</li></ul>
	Manage Jenkins -> Manage Roles <ul style="list-style-type: none"><li>• Create Global and Project level roles.</li><li>• Assign required permissions.</li></ul>
	Manage Jenkins -> Assign Roles <ul style="list-style-type: none"><li>• Assign roles to the users</li></ul>
Configure Global Security	Go through the options available.

## Integration with GIT

SCM	Create a Git repository in Github Dashboard -> Click on existing project (or create a new one) Project -> Configure -> Source Code Management; Select Git Give the Github repository URL Apply and save the settings Build the project
Polls SCM	Project -> Configure -> Build Triggers Check "Poll SCM" Enter * * * * * in the Schedule box (to poll Github every one minute to see if any changes are there)

## Setting up Jobs

Build Options	Project -> Configure -> Build Add build step -> Execute Shell Give any shell command (e.g. pwd, ls etc.,)
	Invoke top-level Maven targets (ensure Maven is installed in jenkins server box) Goals: Give any maven command (e.g. test to run all the Junit tests)
○ Post build options	Understand the available options. We have seen the below options Delete workspace when build is done Build other projects Deploy war/ear to container (this requires Deploy to Container plugin)

- Trigger it remotely
  - Project -> Configure -> Build Triggers
  - Give any authentication token
  - Open the URL in another browser, pass the authentication token in the place of TOKEN\_NAME
- Chain of execution
  - Create multiple projects
  - Set dependencies on each other as we have seen in the class room

## CATLIGHT - Status notifier

- Download Catlight for windows from <https://catlight.io/>
- Give the jenkins URL, username & password

Automated Deployment	<p>Deploy Plugin is required; Deploy to container plugin from Manage Jenkins -&gt; Manage Plugins</p> <ul style="list-style-type: none"> <li>• Create a new freestyle project by the name "FirstWebApp"</li> <li>• <b>Configure</b> -&gt; Source Code Management -&gt; Git</li> <li>• <b>Repository URL:</b> <a href="https://github.com/raghupss/WebAppSample.git">https://github.com/raghupss/WebAppSample.git</a></li> <li>• <b>Configure</b> -&gt; Build -&gt; Invoke top-level Maven targets</li> <li>• <b>Goals</b> -&gt; package</li> <li>• <b>Post build actions</b> -&gt; Add post -build action -&gt; Deploy WAR/EAR to a container</li> <li>• <b>WAR/EAR files:</b> Provide the WAR name (e.g. target/firstwebapp.war)</li> <li>• <b>Context Path:</b> firstwebapp</li> <li>• <b>Containers:</b> Tomcat 7.x (NOTE: works for Tomcat 8.x also) <ul style="list-style-type: none"> <li>○ <b>Manager user name:</b> Give the one that we have configured in the /var/lib/tomcat8/conf/tomcat-users.xml (e.g. deployer)</li> <li>○ <b>Password:</b> give the password (e.g. deployer)</li> <li>○ <b>Tomcat URL:</b> e.g. <a href="http://13.126.110.18:8081">http://13.126.110.18:8081</a> (replace it with your server's public IP)</li> <li>○ Apply and Save the build settings</li> </ul> </li> <li>• Build the project with "Build Now" option</li> <li>• Go to "Console Output" and ensure build and deploy steps succeeded.</li> </ul> <p><b>RUN THE PROJECT:</b> go to browser and give the tomcat URL along with the Tomcat's deployment folder name and the file name (e.g <a href="http://13.126.110.18:8081/firstwebapp/first">http://13.126.110.18:8081/firstwebapp/first</a></p>
----------------------	--

# Github Jenkins Webhook

	Goto Github	
	Create a new project	Test
	Goto "Settings" section Click on "webhooks" Click on "Add webhook"	
	Payload URL	<Jenkins URL>/github-webhook/
	Content Type	Application/json
	Which events would you like to trigger this webhook? Select " Just the push event. "	
Jenkins		
	Install "github integration plugin"	
	Create a freestyle item	Test1
	General -> Check "GitHub Project" Paste your Github project URL	
	Got to "Source Code Management" section Select "Git" Paste the Repository URL	<a href="https://github.com/raghupss/proj1.git">https://github.com/raghupss/proj1.git</a>
	Goto "Build Triggers" section Check "GitHub hook trigger for GITScm polling" option	
Github	Add some text to "ReadMe.md" Commit the change	
Jenkins	Watch the build getting queued up as the result of a change in the connected Github project	