**Project Activities**

* **Fetch Code & Compile**
* **Perform Unit Testing**
  + **Report Generation**
* **Perform Code Coverage**
  + **Generate Coverage Report**
* **Perform Sonar Analysis**
* **Generate Documentation**
* **Package & Distribute Artifact To Nexus**
* **Deploy to Tomcat**

**Git**

* sudo yum -y install git
* git --version

**Java**

* sudo yum -y install java-1.8.0
* sudo yum -y install java-1.8.0-devel
* java -version

**Maven**

* sudo yum -y install maven
* mvn --version

**Nexus**

* wget <nexus-2-link.tar>
* cd nexus/nexus-2.x/bin
* ./nexus start
* netstat -ntpl | grep 8081
* [http://ip-addr:8081/nexus](http://ip-addr:8081/nexus   %20   %20)

**# login and give default credentials admin & admin123**

**Sonarqube    [Requires java 1.8+]**

* **Download Sonarqube**
  + wget <sonarqube-link.zip>
  + unzip ~/sonarqube-6.7.7.zip
  + Start Sonarqube $ ~/sonarqube-6.7.7/bin/linux-x86-64/sonar.sh start
  + <http://serverAddress:9000>

**Jenkins**

* sudo wget -O /etc/yum.repos.d/jenkins.repo<https://pkg.jenkins.io/redhat-stable/jenkins.repo>
* sudo rpm --import<https://pkg.jenkins.io/redhat-stable/jenkins.io.key>
* sudo yum -y install Jenkins
* sudo systemctl start Jenkins
* sudo systemctl enable Jenkins
* allow port 8080 through firewall
* [http://ServerAddress:8080](http://serveraddress:8080)

**Tomcat**

**# Download tar of tomcat7 and extract → cd tomcat/bin → ./startup.sh**

**# sudo vi tomcat-users.xml    { gui tomcat manager }**

Copy this in **<tomcat-users>** tags

*<user username="tomcat" password="tomcat" roles="manager-gui"/>*

*<user username="tomcat1" password="tomcat1" roles="manager-script"/>*

* **change the port number of tomcat from 8080 to 8090**

**# sudo vi server.xml    { change port in** Connector **}**

    Allow port through **firewall**

* **sudo systemctl start tomcat**
* <http://ServerAddress:8090>

**Starting the Project**

**JOB - 1**

Freestyle Job [ **01 - Validate and Compile** ]

→ Discard Old builds

    Max no of builds 3

→ Source code management

    GitHub url

→ Build

**Invoke top-level Maven targets**

**Goals : validate compile**

**JOB - 2**

Freestyle Job [ **02 - Unit Testing** ]

→ Discard Old builds

    Max no of builds 3

→ Source code management

    GitHub url  
→ Build

**Invoke top-level Maven targets**

**Goals : test  
  
But we can’t see how many test cases ran and how many were failed/succeeded**

**→** Go to workspace → target → surefire-reports

**→ At the end Post-build Actions  
    Publish JUnit test result report**

**Locations : \*\*/target/surefire-reports/\*.xml**

→ Run the build again for 2-3 times and you can see the test graph  
→ Click on **Latest Test Result** ⇒ On left side click on **History**

**→ Refresh the Job page to view updated changes**

**JOB - 3**

Freestyle Job [ **03 - Cobertura Code Coverage** ]

→ Discard Old builds

    Max no of builds 3

→ Source code management

    GitHub url  
→ Build

**Invoke top-level Maven targets**

**Goals : cobertura:cobertura -Dcobertura.report.format=xml**

**→ Now if you see console output :  
    → Cobertura: Loaded information on 10 classes.  
    → Cobertura Report generation was successful.**

**→ But we can’t see any reports, so let’s go and work with plugin now  
    → Even if we see in Post-build actions nothing is found  
    → Install Cobertura plugin**

**→ At the end Post-build Actions  
    Publish Cobertura Coverage Report**

**Locations : \*\*/target/site/cobertura/coverage.xml**

**→ Refresh the Job page to view updated changes**

**JOB - 4**

Freestyle Job [ **05 - Sonar Analysis - Code Analysis** ]

→ Discard Old builds

    Max no of builds 3

→ **Source code management**

    GitHub url

→ **Build**

**Invoke top-level Maven targets**

**Goals : clean compile sonar:sonar**

**JOB - 5**

Freestyle Job [ **04 - Documentation** ]

→ Discard Old builds

    Max no of builds 3

→ Source code management

    GitHub url  
→ Build

**Invoke top-level Maven targets**

**Goals : javadoc:javadoc**

**→ Now if you see console output :  
     → All the documentation is being generated in target/site/apidocs**

**→ Even if we see in Post-build actions nothing is found  
    → Install Javadoc Plugin**

**→ At the end Post-build Actions**

**Publish Javadoc**

**Locations : target/site/apidocs**

**→ Refresh the Job page to view updated changes**

**JOB - 6**

Freestyle Job [ **06 - Packaging and Nexus Release** ]

→ **Source code management**

    GitHub url

→ **Build**

**Invoke top-level Maven targets  → Advanced**

**Goals : clean package deploy**

→ **Settings File  
          File path - settings.xml**

**JOB - 7**

Freestyle Job [ **07 - Tomcat Deployment** ]

→ Discard Old builds

    Max no of builds 3

→ Source code management

    GitHub url

→ Build

**Invoke top-level Maven targets**

**Goals : clean package**

**→ Install Deploy to container plugin**

**→ At the end Post-build Actions**

**Deploy war/ear to a container**

WAR/EAR files - \*\*/\*.war

Context path    - /SpringProject

Containers

Tomcat 7.x

Credentials tomcat1/\*\*\*\*\*\* (Tomcat) Add

Tomcat URL<http://104.197.89.223:7070>

**CI/CD Pipeline View**

**Install Build Pipeline Plugin → New View → CI - CD and check Build Pipeline View**

* Click on Build pipeline view and sort the jobs accordingly from job-1 to job-7
* From now, if you check the build process of one job it will automatically builds the process of all jobs in pipeline.