**Jenkins pipeline**

**Pre-requesties**

1.Launch the Jenkins installer package , run the Jenkins installer

Run this url : <http://localhost:8080/> By default it will run in this url

2.Create Ant Project ,here the sample ant project



3.Create the master git repository ,copy the source code and clone it



**Pipeline**

Before building a pipeline we have to create various project steps and configure it into pipeline

In this pipeline we are going to configure many project steps

0.Checkout source from the GIT

1.Compile the code

2.Static code analysis-Sonar

3.War Generation

4.Push the war file into Nexus Repository

5.Download the war from Nexus Repository

6.Deploy war into the Server

**Step 0: Checkout source from the GIT**

**Plugins :**

Jenkins -> Manage Jenkins -> Manage Plugins ->advanced

Install **Git Plugin** and dependency plugins to install software successfully

From this link <https://wiki.jenkins-ci.org/display/JENKINS/Git+Plugin>

**Dependency Plugin**

[workflow-scm-step](https://wiki.jenkins-ci.org/display/JENKINS/Pipeline+SCM+Step+Plugin) (version:1.14.2)   
[credentials](https://wiki.jenkins-ci.org/display/JENKINS/Credentials+Plugin) (version:2.1.8)   
[git-client](https://wiki.jenkins-ci.org/display/JENKINS/Git+Client+Plugin) (version:2.1.0)   
[mailer](https://wiki.jenkins-ci.org/display/JENKINS/Mailer) (version:1.18)   
[matrix-project](https://wiki.jenkins-ci.org/display/JENKINS/Matrix+Project+Plugin) (version:1.7.1)   
[promoted-builds](https://wiki.jenkins-ci.org/display/JENKINS/Promoted+Builds+Plugin) (version:2.27, optional)   
[scm-api](https://wiki.jenkins-ci.org/display/JENKINS/SCM+API+Plugin) (version:1.3)   
[ssh-credentials](https://wiki.jenkins-ci.org/display/JENKINS/SSH+Credentials+Plugin) (version:1.12)   
[token-macro](https://wiki.jenkins-ci.org/display/JENKINS/Token+Macro+Plugin) (version:1.12.1, optional)   
[parameterized-trigger](https://wiki.jenkins-ci.org/display/JENKINS/Parameterized+Trigger+Plugin) (version:2.4, optional)

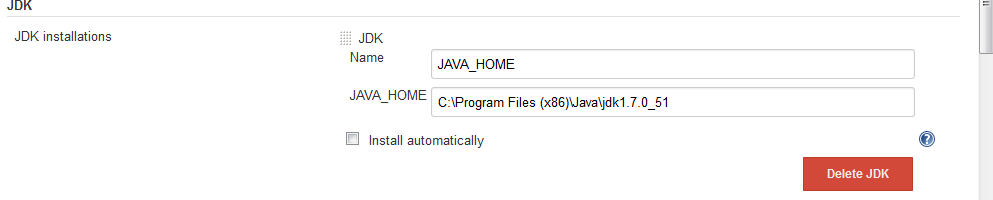
**Global Configuration**

We can made the global configurations here

Jenkins -> Manage Jenkins ->Configure System

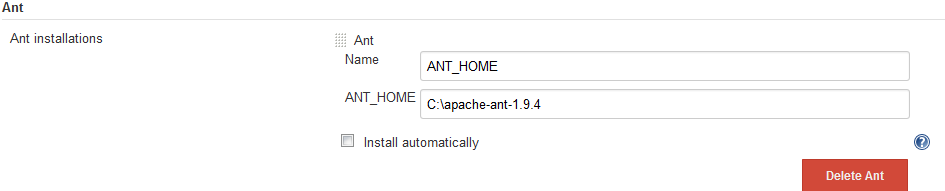
**JDK Installation**

Set Java\_Home



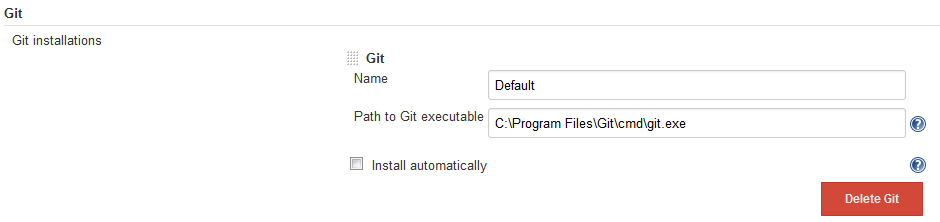
**Ant Installation**

Set Ant\_home



**Git**

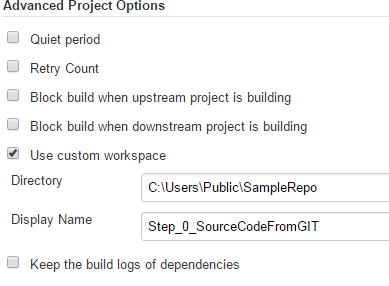
Set Git\_home



**Local Configuration**

Create New item->freestyle project

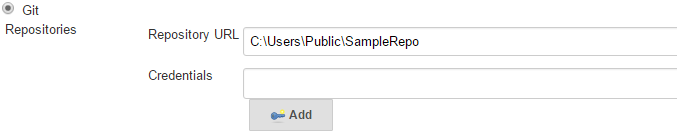
1.Under **Advanced Project Options**



Directory : Mention the custom workspace directory

DisplayName :Display name of the project

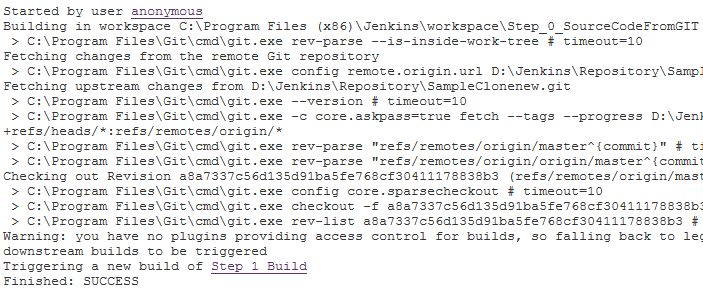
2.Under source code management -> git



Repository URL: Cloned git repository URL

->Save->BuildNow

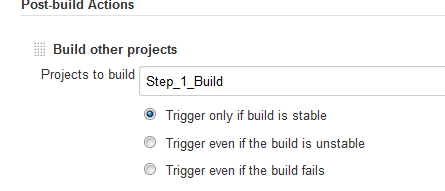
In Console



Can able to check out from the cloned repository

Build Successful!!!

3.In Post build action->build other project



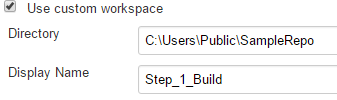
Give name of the other project which has to be run in sequential steps

**1. Compile the code**

**Local Configuration**

Create New item->freestyle project

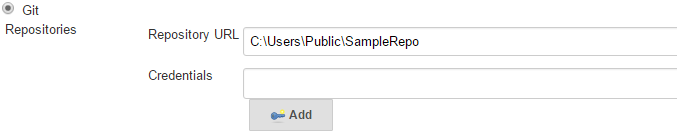
1.Under **Advanced Project Options**



Directory : Mention the custom workspace directory

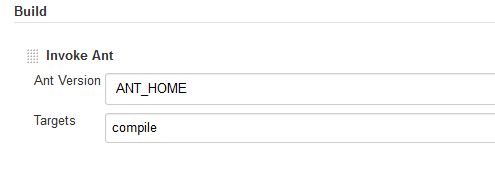
DisplayName :Display name of the project

2.Under source code management -> git



Repository URL: Cloned git repository URL

3. Build



Select->invoke ant

Ant version : select the ant home in the dropdown

Targets :mention the target which has to be takes place(Target name as same as <target> specified in the build.xml)

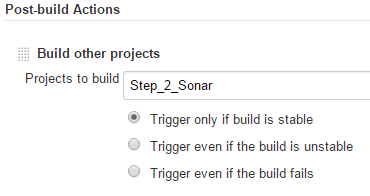
->Save->BuildNow

In Console



Build Successful!!!

4.In Post build action->build other project



Give name of the other project which has to be run in sequential steps

**2.Static code analysis-Sonar**

**Plugins :**

Jenkins -> Manage Jenkins -> Manage Plugins ->advanced

Install **SonarQube Plugin** and dependency plugins to install software successfully

From this link <https://wiki.jenkins-ci.org/display/JENKINS/SonarQube+plugin>

**Dependency Plugin**

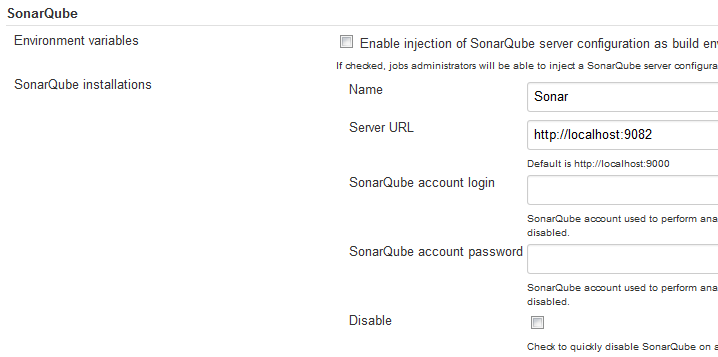
[configurationslicing](https://wiki.jenkins-ci.org/display/JENKINS/Configuration+Slicing+Plugin) (version:1.40, optional)   
[maven-plugin](https://wiki.jenkins-ci.org/display/JENKINS/Maven+Project+Plugin) (version:2.12.1, optional)   
[jquery](https://wiki.jenkins-ci.org/display/JENKINS/jQuery+Plugin) (version:1.11.2-0)

**Global Configuration**

We can made the global configurations here

Jenkins -> Manage Jenkins ->Configure System

1.SonarQube

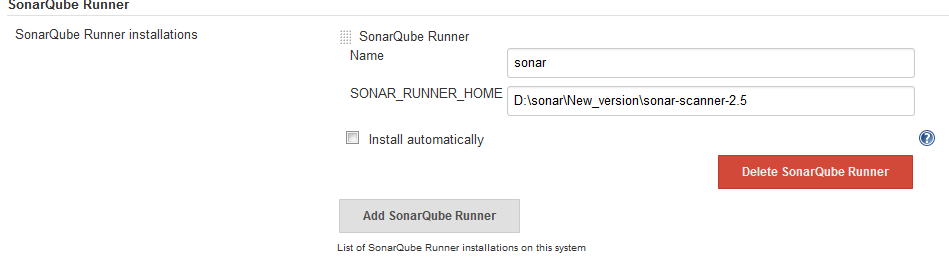


Name: name of our choice

Server URL: server url in which sonarqube server is running

(note :server should be in running condition)

2.Sonarqube runner



SonarQube runner name: name of the sonarqube runner

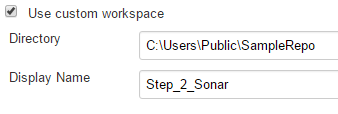
Sonar\_runner\_home:we have to mention the sonarqube scanner url in which ,scanner is running

(note :server should be in running condition)

**Local Configuration**

Create New item->freestyle project

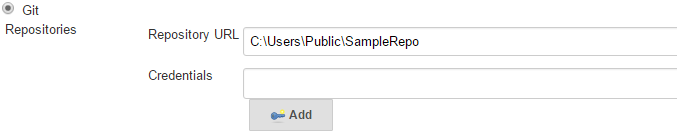
1.Under **Advanced Project Options**



Directory : Mention the custom workspace directory

DisplayName :Display name of the project

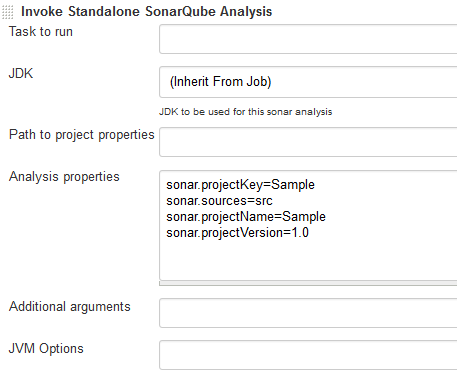
2.Under source code management -> git



Repository URL: Cloned git repository URL

3. Build

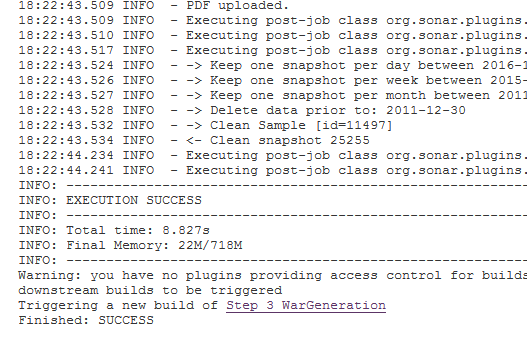
Select->Invoke standalone sonarqube analysis



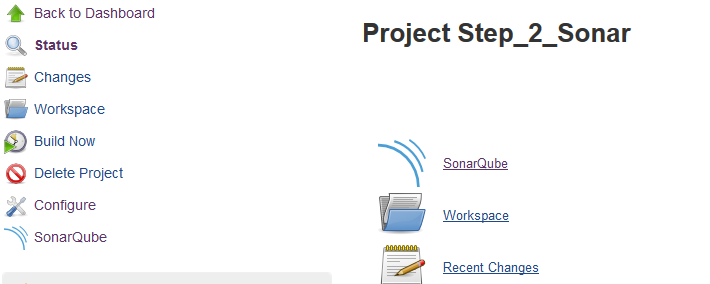
We have to mention the analysis properties (things mentioned in the sonar\_runner.properties file)

->Save->BuildNow

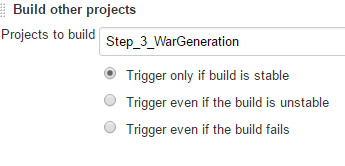
Console



We can able to see the report by clicking sonarqube



4.In Post build action->build other project



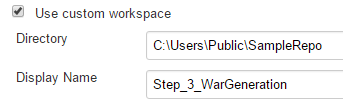
Give name of the other project which has to be run in sequential steps

**3. War Generation**

**Local Configuration**

Create New item->freestyle project

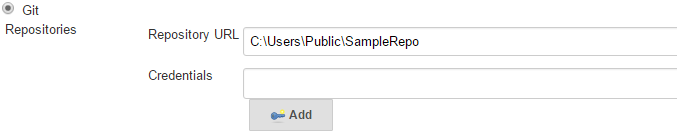
1.Under **Advanced Project Options**



Directory : Mention the custom workspace directory

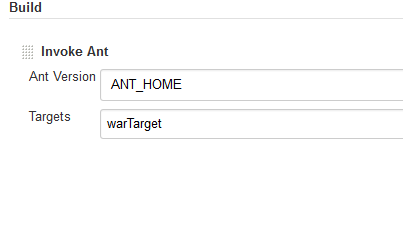
DisplayName :Display name of the project

2.Under source code management -> git



Repository URL: Cloned git repository URL

2. Build



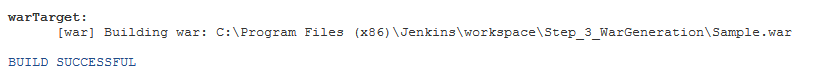
Select->invoke ant

Ant version : select the ant home in the dropdown

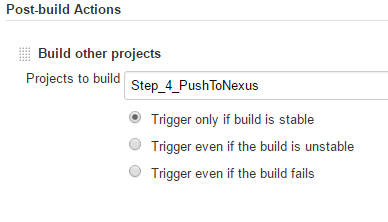
Targets : mention the target which has to be takes place(Target name as same as <target> specified in the build.xml)

->Save->BuildNow

In Console



4.In Post build action->build other project



Give name of the other project which has to be run in sequential steps

**4.** **Push the war file into Nexus Repository**

**Plugins :**

Jenkins -> Manage Jenkins -> Manage Plugins ->advanced

Install **Sonatype Nexus Plugin** and dependency plugins to install software successfully

From this link <http://download.sonatype.com/nexus/ci/latest.hpi>

**Dependency Plugin**

workflow-step-api-1.15.hpi

plain-credentials-1.1.hpi

environment-script.hpi

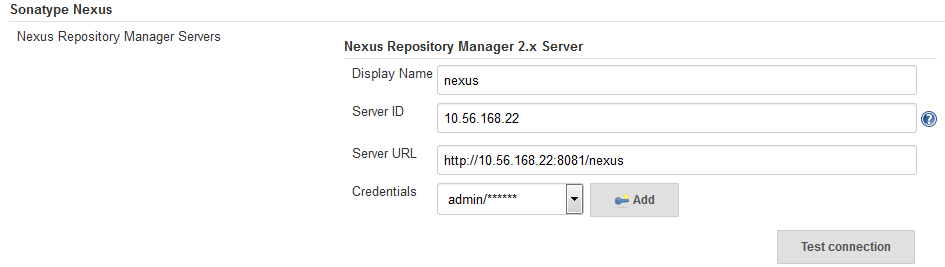
**Global Configuration**

We can made the global configurations here

Jenkins -> Manage Jenkins ->Configure System

**Sonatype Nexus -** to connect with Nexus Repository

Select -> **Nexus Repository 2.x Server**



Under **Sonatype nexus**

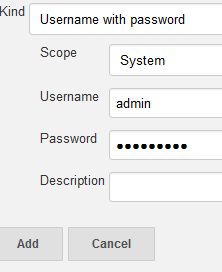
1.Display Name : Name to display for nexus

2.ServerID : IP Address for nexus repository

3.Server URL: Url to connect with your nexus repository

4.Credentials:

We have to select **username with password** option in the dropdown and enter the credentials



And click on add

If we click on **Test Connection** ->

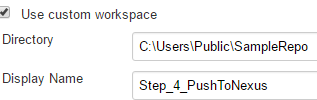
If connection established it will show like

**Nexus Repository Manager 2.x connection succeeded (2 hosted release Maven 2 repositories)**

**Local Configuration**

Create New item->freestyle project

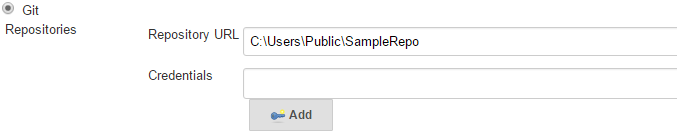
1.Under **Advanced Project Options**



Directory : Mention the custom workspace directory

DisplayName :Display name of the project

2.Under source code management -> git



Repository URL: Cloned git repository URL

**3.Nexus Repository Manager Publisher** - to publish artifacts into repository

**Add build step->** **Nexus Repository Manager Publisher**

1. Nexus Instance: select the instance name from dropdown which we have created in global configuration section

2. Nexus Repository: select the repository to which we have permission to access

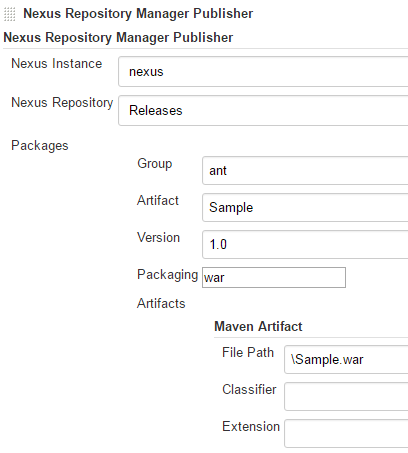
3. **Package**: select the package

Under while fill the details about the artifact which we are going to upload into nexus repository

Group, artifact , version and packaging(ex.war)

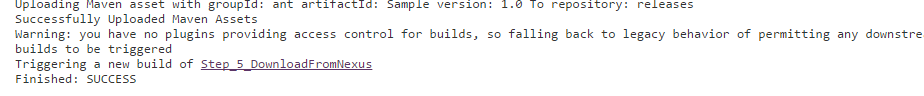
Click on **Artifact-Add artifact path-Maven artifact**

And mention the file path where the war or jar file stored

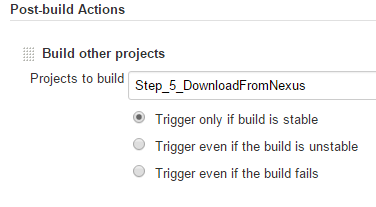


->Save->BuildNow

In Console



4.In Post build action->build other project



Give name of the other project which has to be run in sequential steps

**5.** **Download the war from Nexus Repository**

**Plugins :**

Jenkins -> Manage Jenkins -> Manage Plugins ->advanced

Install **Repository Connector plugin** and dependency plugins to install software successfully

From this link <https://wiki.jenkins-ci.org/display/JENKINS/Repository+Connector+Plugin>

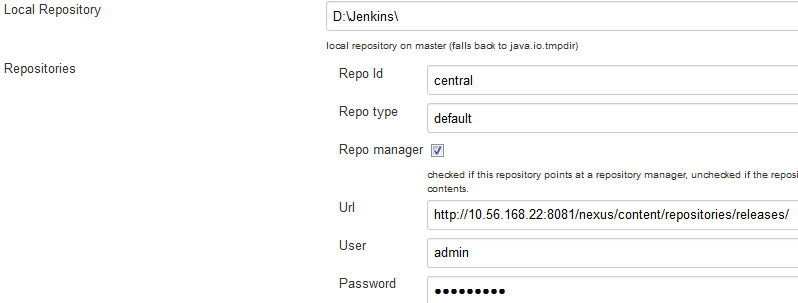
**Global Configuration**

We can made the global configurations here

Jenkins -> Manage Jenkins ->Configure System

**Artifact Resolver :**This Configuration has to be done once after the Artifacts stored into nexus then only we can retrieve .

**Artifact Resolver –** to retrieve artifact from Nexus



Under **Artifact Resolver**

1.Local Repository - we can able to give local directory path so that downloaded artifacts will be stored here

2.Repository

a)Repo Id –Central( by default)

b)Repo type- Default(by default)

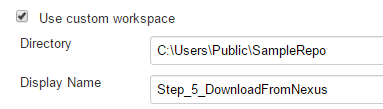
c)Url-path where your files stores inside the repository

d)username and password - which is used when connecting the nexus repository

**Local Configuration**

Create New item->freestyle project

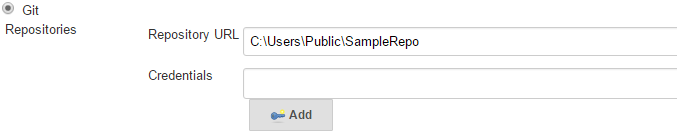
1.Under **Advanced Project Options**



Directory : Mention the custom workspace directory

DisplayName :Display name of the project

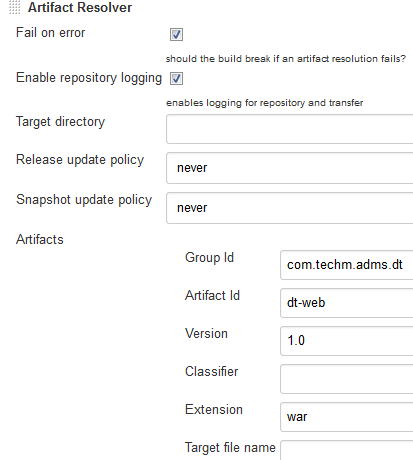
2.Under source code management -> git



Repository URL: Cloned git repository URL

**3. Artifact Resolver –** to retrieve (download) artifact from nexus

**Add build step -> Artifact Resolver**



We have to mention the details about the artifact which we are going to download

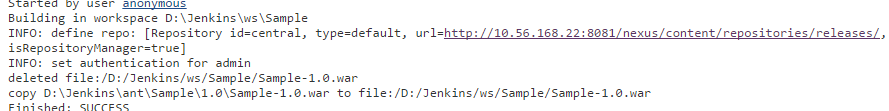
1.Target directory-the space where the downloaded file will be copied

2.Release update policy and snapshot update policy we can select ad per our wish

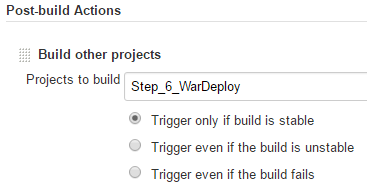
3.Artifact-have to mention the artifact details which we are going to download from the nexus repository

->Save->BuildNow

**In Console**



4.In Post build action->build other project



Give name of the other project which has to be run in sequential steps

**6.** **Deploy war into the server**

**Plugins :**

Jenkins -> Manage Jenkins -> Manage Plugins ->advanced

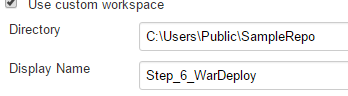
Install **Deploy Plugin** and dependency plugins to install software successfully

From this link <https://wiki.jenkins-ci.org/display/JENKINS/Deploy+Plugin>

**Local Configuration**

Create New item->freestyle project

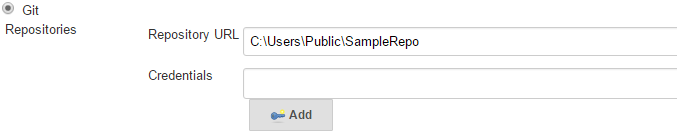
1.Under **Advanced Project Options**



Directory : Mention the custom workspace directory

DisplayName :Display name of the project

2.Under source code management -> git



Repository URL: Cloned git repository URL

**3. Post-build action**

In **Post-build action->deploy war/ear to a container**

1.War/ear file- \*\*\\*.war(by default it will take the copied war from the workspace which we have mentioned)

2.Context path- The context path that the container should use to publish the WAR/EAR

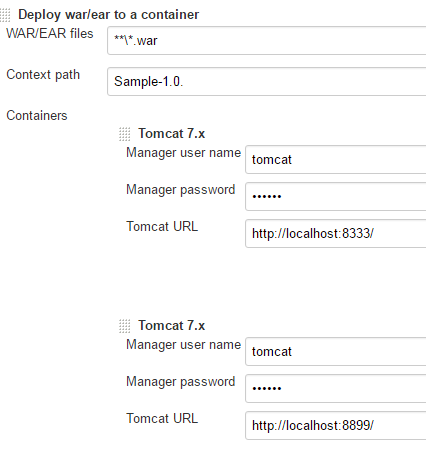
3.Container-Add container->can able to select the server container which is used for deployment

a. manager username-username of the tomcat server

b. manager password-password of the tomcat server

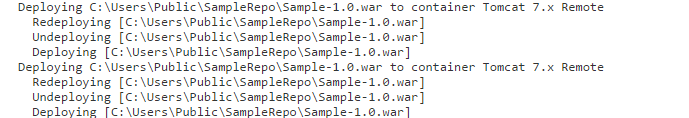
c. tomcat url- URL in which tomcat is running

we can also mention n number of servers



->Save->BuildNow

**In Console**



Project Steps completed Successfully!!!

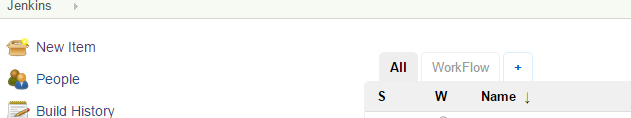
Pipeline:

To create a build pipeline add **Pipeline Plugin**

From this link <https://wiki.jenkins-ci.org/display/JENKINS/Build+Pipeline+Plugin>

In Jenkins UI

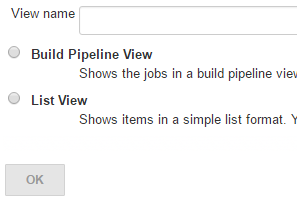
Click on ”+” sign in order to add pipelin



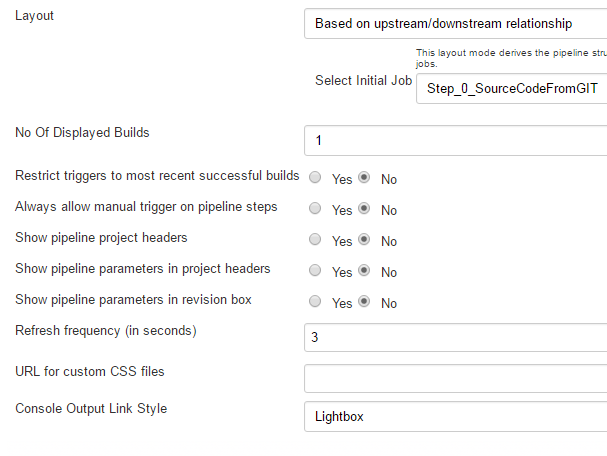
New pipeline

View name: name of the pipeline

click on build pipeline view



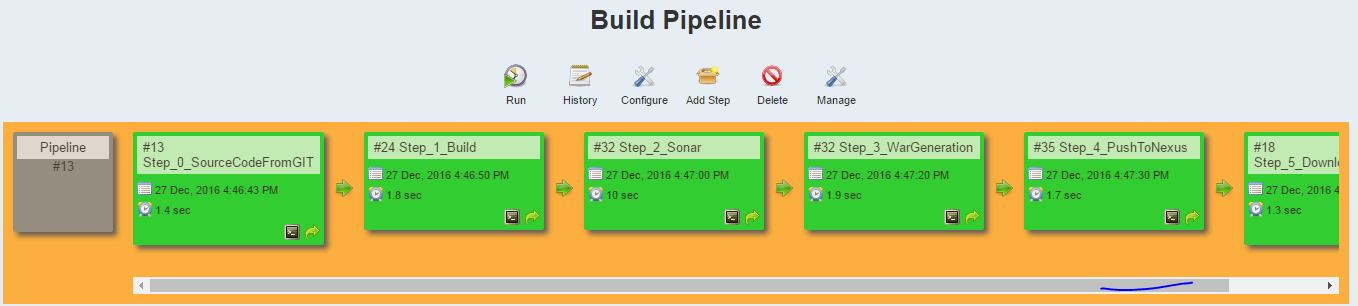
Click on OK



Select initial job: select the first project which has to be execute first in the pipeline

And click OK

Pipeline will appear ->click ->run



Pipeline executed successfully!!!