**DevOps-CI**

**Jenkins-Workflow**

1. Building the maven project

2. Storing those artifact into NEXUS Repository

3. Retrieving (Downloading) those artifact to the local workspace

4. Deploy that war file into tomcat server

Steps:

**Plugins to be added**

Various plugins to be added

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

Can be able to upload from our local

To connect with nexus repository following plugins has to be added



To retrieve artifact from nexus, following plugin has to be added



To Deploy the war into tomcat ,following plugin has to be added



Once we add all these plugins then we have to made following changes in global configuration

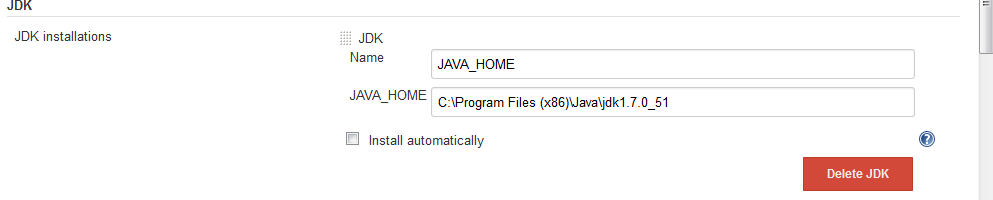
**Global Configuration**

We can made the global configurations here

Jenkins -> Manage Jenkins ->Configure System

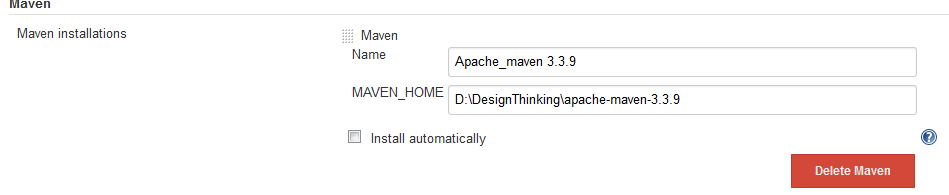
**JDK Installation**

Set Java\_Home



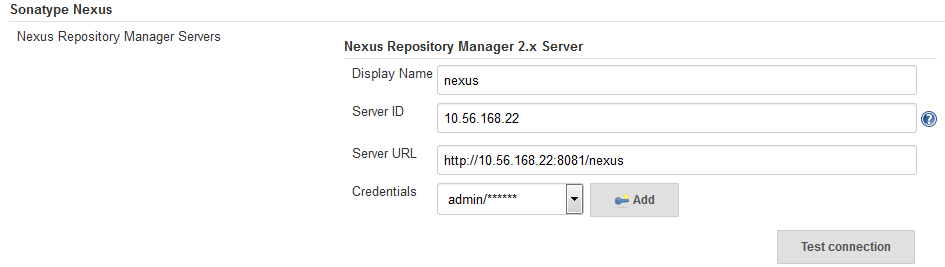
**Maven**

Set Maven\_Home



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

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



Under **Sonatype nexus**

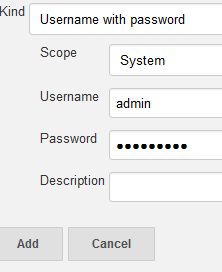
1.Display Name : Name to display for nexus

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

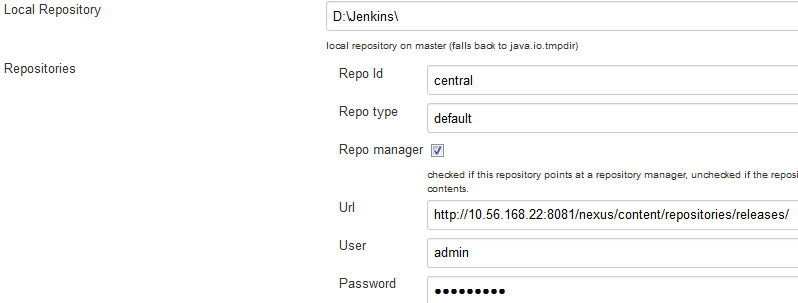
If we click on **Test Connection** ->

If connection established it will show like

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

**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

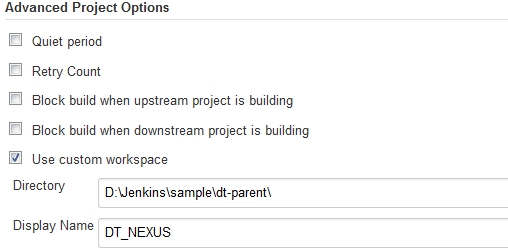
**Local Configuration**

Create new project in Jenkins

Jenkins -> New Item ->Give Project name ->free style project

Under **Advanced Project Options**

Give the workspace where the project is located



**Source Code Management**

Set to – None



**Build – to build the source**

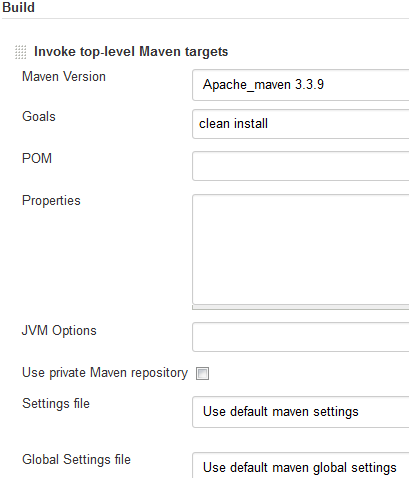
**Add build step->Invoke top level maven target**

1.MavenVersion – Select the maven version

2.Goals - clean install

3. Settings file – use default maven setting

4. Global Settings file - use default maven global setting



**Nexus Repository Manager Publisher** - to publish artifact 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

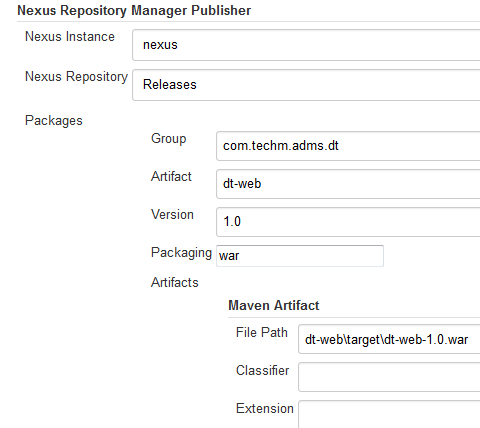
3. **Package**: select the package

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

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

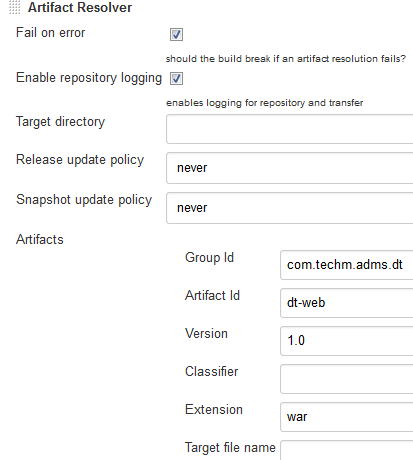
Click on **Artifact-Add artifact path-Maven artifact**

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



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

**Add build step -> Artifact Resolver**



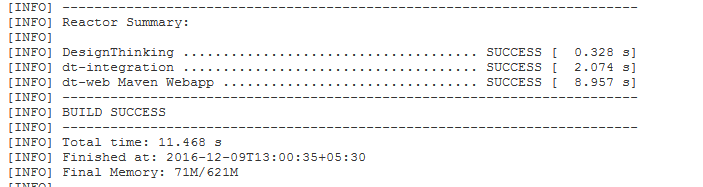
We have to mention the details about the artifact which we are going to download

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

If we save and click on-> build now



Result for building the maven project is successful!!!



Artifacts stored into the nexus repository successfully!!!



Artifacts downloaded from the nexus Repository and downloaded file is copied into the target folder…

**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)

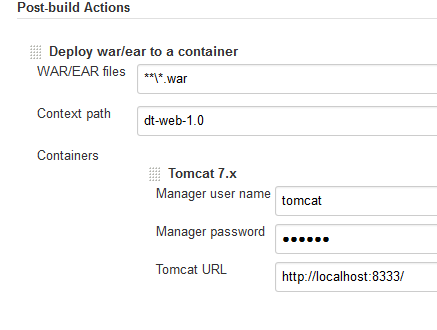
2.Context path- The context path that the container should use to publish the WAR/EAR

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

a. manager username-username of the tomcat server

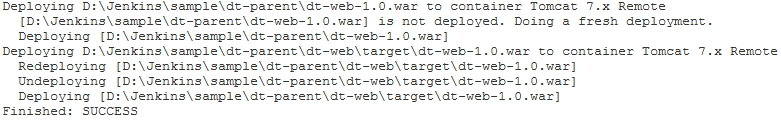
b. manager password-password of the tomcat server

c. tomcat url- URL in which tomcat is running



Click on save.

Click ->Build Now



War deployed into the tomcat server successfully!!!