* Set Global Tool Setting for different Tools eg Java,Git,Maven,Docker
* Use Role Authorization plugin to create different roles
* Create users and assign them to different roles and test
* Create a Simple Free Style Project with Windows Shell executable to print date and time
* Create a project with Cron Job with Windows Shell executable to print date and time
* Create a job with Source Code connected to GitHub and Poll SCM
* Create a maven repo on GitHub and install maven, tomcat on the Jenkins server

For tomcat in conf/server.xml change connector port to 8081 as Jenkins uses 8080

<Connector port="8081" protocol="HTTP/1.1"

Also create user for Jenkins to come and deploy conf/tomcat-user.xml

<role rolename="manager-script"/>

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

In the bin go to command prompt and do startup.bat

In Jenkins create a new freestyle project add Git as source code and the mvnwebapp project to it. For build step add windows shell script mvn package.

One the mvnwebapp.war file is created successfully deploy it on tomcat server.

For this add plugin “Deploy to container” then in the same project post build step use deploy war/ear to container

|  |  |  |
| --- | --- | --- |
| WAR/EAR files |  | [Help for feature: WAR/EAR files](http://localhost:8080/job/Mvn_WebApp/configure) |
|  | |  |  |
|  |  | |  |
|  | Context path |  | [Help for feature: Context path](http://localhost:8080/job/Mvn_WebApp/configure) |
|  | |  |  |
|  |  | |  |
|  | Containers | |  |  |  |  | | --- | --- | --- | --- | | **Tomcat 8.x Remote** | | |  | |  | Credentials | Add |  | |  | |  |  | |  | Tomcat URL |  | [Help for feature: Tomcat URL](http://localhost:8080/job/Mvn_WebApp/configure) | |  | |  |  | |  |  | |  | |  |  | Advanced... |  | | Delete | | | |   Add Container |  |

The credentials will be the tomcat credentials.

After the job runs successfully check the output on : <http://localhost:8081/mvnwebapp/>

* For Master-Slave config can use JNLP – Java Network Launch Protocol to launch Slaves.

For this enable JNLP from Global Security Settings

To Launch new nodes

Config Node > New Node

Create a Job which run only on Slave

* To Create Jenkins Backup install Backup plugin and create backup and restore from it
* Pipeline: Create 3 jobs: 1 upstream and 1 downstream
* Plugin to view Pipeline Structure: Delivery Pipeline Plugin
* Post Jenkins 2.0 now we can write pipelines in Groovy language

Eg

pipeline {

agent any

stages {

stage('Compile') {

steps {

echo 'Compiled Successfully'

}

}

stage('JUnit') {

steps {

echo 'JUnit Passed Successfully'

}

}

stage('Quality Gate') {

steps {

echo 'Quality Gate Passed Successfully'

}

}

stage('Deploy') {

steps {

echo 'Deployment Successfully'

}

}

}

post{

always{

echo 'This will always run'

}

success{

echo 'This will run only if successful'

}

failure{

echo 'This build has failed'

}

unstable{

echo 'Unstable Build'

}

changed{

echo 'This will only run if the state of the build has changed'

echo 'for eg if previous build had failed and this is successful'

}

}

}

With Git Integration and on slave

pipeline {

agent {

label 'Slave'

}

stages {

stage('Git Checkout') {

steps {

echo 'Checkout from Git Repo'

git 'https://github.com/BrijKaur/Pipeline\_Script.git'

}

}

stage('Build') {

steps {

echo 'Building the checkout project'

bat 'Build.bat'

}

}

stage('JUnit') {

steps {

echo 'JUnit Passed Successfully'

bat 'Unit.bat'

}

}

stage('Quality Gate') {

steps {

echo 'Quality Gate Passed Successfully'

bat 'Quality.bat'

}

}

stage('Deploy') {

steps {

echo 'Deployment Successfully'

bat 'Deploy.bat'

}

}

}

post{

always{

echo 'This will always run'

}

success{

echo 'This will run only if successful'

}

failure{

echo 'This build has failed'

}

unstable{

echo 'Unstable Build'

}

changed{

echo 'This will only run if the state of the build has changed'

echo 'for eg if previous build had failed and this is successful'

}

}

}

Parallelly and non parallelly with script pulled out from git

The script should be committed on Git by the name JenkinsFile

Pipeline script on SCM