Tools Integration in Jenkins: GIT & MAVEN

To change the URL of Jenkins Server:
Dashboard → Manage Jenkins → System Configuration → click on System → Jenkins URL → make changes to URL here → Apply + Save

Integration of GIT:
First of all, install GIT on our master server (on Ec2 machine where Jenkins is installed). When you want to integrate any tool and installing plugin of that tool in Jenkins that tool has to be first installed on master Server, plugin installation doesn't mean tool has been installed on master server.

To clone a public GIT Repo:
New Item → add job name → Freestyle Project → OK → select Git →Add Repository URL → add branches to build → Apply + Save → Build Now
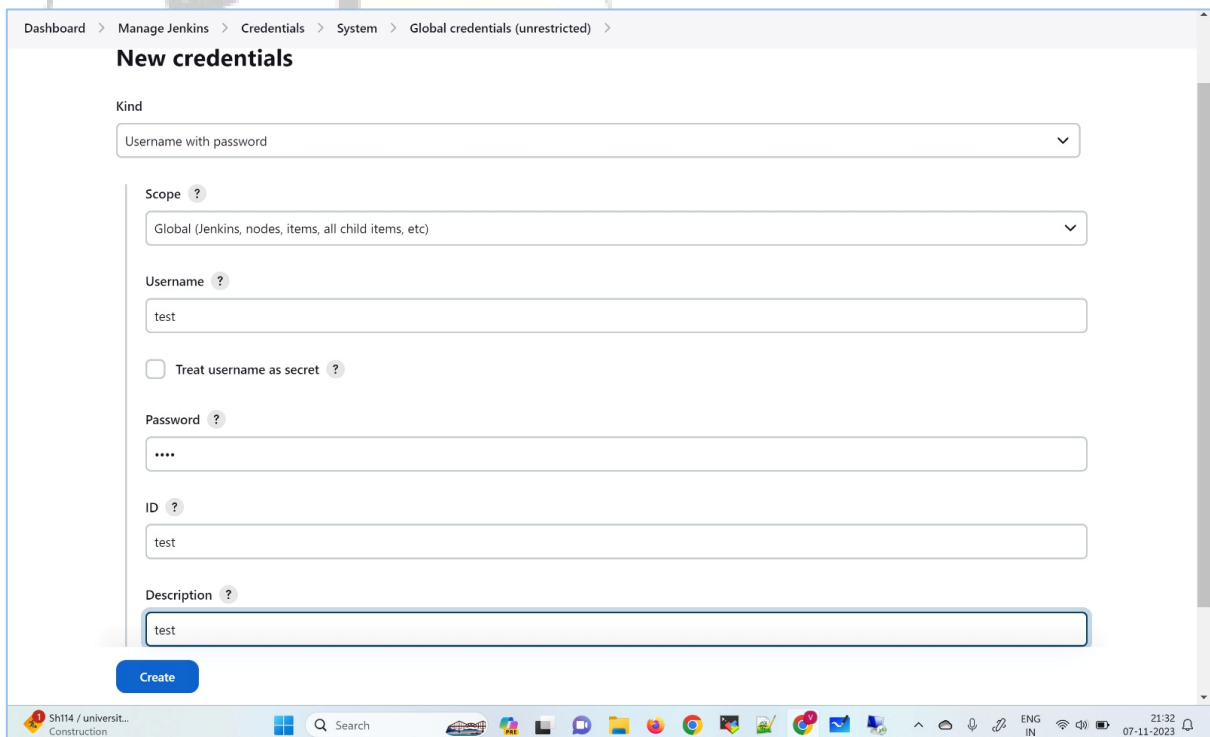If you make any changes to git repository and build again those changes you can be seen in Build with commit id.

To clone a Private GIT Repo:
We can set our gits personal access tokens in Jenkins by using credentials.

First create credentials:
Dashboard → manage Jenkins → Security→ Credentials → Click on Global (You can use this for all jobs) → Add Credentials → Select kind - Username with password → Add username → add git hub token for password → add ID → add description → click on Create (You can select - Treat username as a secret - so username will not be visible, you cannot see password again, you can change the password but original password cannot be viewed again)

Create a new Job:
New Item → give name → Freestyle Project → OK → select Git Plugin → Add Repository URL → add Credentials → add branches to build→ Apply + Save → Build Now

Integration of maven:

Install maven on master server first,
Another way of installing maven is through zip:

```
cd /mnt
Mkdir build_tools
Cd build_tools/
Wget https://dlcdn.apache.org/maven/maven-3/3.9.5/binaries/apache-maven-
3.9.5-bin.zip
unzip apache-maven-3.9.5-bin.zip
rm -rf apache-maven-3.9.5-bin.zip
cd apache-maven-3.9.5
cd bin
./mvn
```

To set environment variables
```
Cd
Ls -ltra
vi .bash_profile
```

.bash_profile

# Get the aliases and functions
if [-f ~/.bashrc]; then
     . ~/.bashrc
fi

User specific environment and startup programs

PATH=$PATH:$HOME/bin

export MAVEN_HOME=/mnt/build_tools/apache-maven-3.9.5

export PATH=$MAVEN_HOME/bin:$PATH

export PATH

```
ESC + :x
```

Now logout and login to the root user again and we can see mvn command can run from any location

```
logout
sudo su
```

Now restart Jenkins by restarting Apache tomcat to reflect the changes:

```
Cd /mnt/servers/apache-tomacat-9.0.82/bin
./shutdown.sh
./startup.sh
```

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export MAVEN_HOME=/mnt/build_tools/apache-maven-3.9.5

export PATH=$MAVEN_HOME/bin:$PATH

export PATH
~
~
~
~
~
~
~
~
~
~
~
~
~
~
".bash_profile" 16L, 266B                               14,1         All
```

Integrate maven in Jenkins:

On Jenkins dashboard → Tools → Maven installations → Add maven → add name → add Maven home path -
Eg: `/mnt/build_tools/apache_maven-3.9.5` → Apply +save

Dashboard → New Item → give name → freestyle project → discard old builds → Advanced → use custom
workspace → enter directory path `/mnt/vel-app` → GIT → add git repository URL
(https://github.com/jidnyasactc/loginwebapp.git) → select master branch → select delete workspace before the
build start → add build steps → execute shell →

```
rm -rf /root/.m2/repository
```

→ add build step → invoke top level maven targets → select maven version → Goals as `clean install` → add build step → Execute Shell →

```
cp -r target/LoginWebApp.war /mnt/servers/apache-tomcat-9.0.82/webapps
```

→ Apply + Save

Click on Build Now

Hit URL in browser:
public _ip:8080/LoginWebApp
Eg: http://3.6.88.146:8080/LoginWebApp/

How to add Plugins in Jenkins:
Dashboard – Manage Jenkins – System Configuration – Plugins – Available plugins – search the name of the plugins – select plugin – click on install

Plugins path on server:
```
cd /root/.jenkins/plugins/
```

Official Website:
plugins.jenkins.io