JENKINS

**Plugins in Jenkins**

* A plugin in jenkin is an add-on that extends the functionality of the Jenkins software.
* Git
* Maven
* Docker
* Aws authentication
* Blue ocean
* Credentials login

Plugin configuration in Jenkins

User sign in with jenkins website login

Jenkins aws plugin aws

**Variables in Jenkins**

Variables are used to store data that can be reuse4d throughout the build process. These variables can hold values such as paths, filenames or configuration settings making it easier to manage and maintain build scripts.

**User defined variables:**

**Local variables :** works only with in a item /job.

**System level variables:** we can add at Jenkins system level config and use across all the jobs.

**Jenkins variables:** built in variables , no need to configure anything, anywhere.

**Access location:** <http://jenkins-url:8080> / env- vars.html./

**Generating job with git cloning in Jenkins**

1. Create one instance called tomcat while creating that give some modifications in the network settings click on edit and go to the subnet give the same availability zone as which is allocated for Jenkins.
2. Make sure that both Jenkins and tomcat are in running state.
3. Now connect the tomcat to the linux server and enter sudo su.
4. Now install the java using command **sudo dnf install java-17-amazon-corretto-devel.**
5. Also make **sure** that maven is installed correctly **sudo yum install maven -y.**
6. Now open the chrome browser and type tomcat webserver and in the page left side we can see tomcat -q click on it and in that binary distributions🡪core🡪tar.gz🡪here right click on it and copy the address.
7. Now in gitbash use the command like **sudo wget <url>**
8. By giving ls we can see the zip file is downloaded.
9. Now extract the zip file using **tar xvf apache-tomcat-9.0.102.tar.gz**
10. Now giving ls we can see the file is unzipped.
11. Now change the directory to the **cd apache-tomcat-9.0.102.**
12. Now give the command like **chmod 777 webapps.**
13. Now change the directory to the **cd webapps**.
14. Noe open the nano editor using **nano webapps/manager/METRA-INF/context.xml**  when it is opening the nano editor it will display some content in that delete the lines which is start with <value, allow>.
15. Now change the directory to the cd .. [apache tomcat]
16. Now give the command **nano conf/tomcat.user.xml.**
17. In this editor copy and paste the lines

**<role rolename="manager-gui"/>**

**<role rolename="manager-script"/>**

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

1. Now we need to change the directory to cd bin.
2. Now give the command like ./startup.sh
3. Now copy the ip4 address and paste it in the google with the port number 8080 there we can see the tomcat server asking the username and password bother are tomcat.
4. Now click on Jenkins instance and copy the ip4 address and paste it int the goggle with the port number 8080 then the Jenkins page will be open and login into the dashboard.
5. In the Jenkins dashboard we can see **new item** which is used to create a new project/job and giving a sample name and freestyle project.
6. After this step we will enter into the configure page.
7. Now we are doing this project by suing git so that we should select one repository for example myweb so by clicking the git icon copy the url of https and paste it there.
8. No need to select any triggers.
9. In the environment we can select the first field like optional.
10. Next in the buildsteps🡪add build step🡪invoke top-level maven target🡪goals🡪clean install.
11. Next go to the post-build actions🡪war/ear files🡪target/\*.war🡪context.path🡪/myweb.
12. In the containers click on the tomcat 9.x remote and add user and password which is used in tomcat login page🡪tomcat url🡪click on save.

Note : in this project we made some changes in the pom.xml file.

1. Next click build now🡪the project will be executed successfully it shows even it fails.

This is how we build the entire project.