Jenkins Class notes by Sunil sir.

Q: What is Jenkins?

Ans: Its a Java based application. Because of this java its a platform independent. No matter what operating system we are running it will support. windows, mac, linux, unix.

Q: why we will use Jenkins ?

Ans: used for continuous Integration and continuous delivery.

Q: Why we run Jenkins in standalone server Or Jenkins on Tomcat ?

Ans: The best things of install Jenkins inside of tomcat is that Tomcat is a single server where end number web applications are install and the best things is both Jenkins and Tomcat are written in Java.

Always go with Tomcat version 5 or above. and java with 7 or above.

Try to do something's extra go to

Q: What is the use of Maven and jenkins ?

Ans: Maven is a build tool. In short a successor of Ant. it helps in build and version control and

However Jenkikns is a Continuous integration system,

where in Maven is used for build. Jenkins can be used to automate the deployment process.

[bile@node1 ~]$ whereis java

[bile@node1 ~]$ rpm -qa java

[bile@node1 ~]$ sudo yum install java

Dependencies Resolved

================================================================================

Package Arch Version Repository

Size

================================================================================

Installing:

java-1.8.0-openjdk x86\_64 1:1.8.0.171-7.b10.el7 updates 246 k

Transaction Summary

================================================================================

Install 1 Package (+30 Dependent packages)

Upgrade ( 7 Dependent packages)

Total download size: 38 M

Is this ok [y/d/N]: y

Total 811 kB/s | 38 MB 00:48

Installed:

java-1.8.0-openjdk.x86\_64 1:1.8.0.171-7.b10.el7

[bile@node1 ~]$ whereis java

[bile@node1 ~]$ java -version

openjdk version "1.8.0\_171"

OpenJDK Runtime Environment (build 1.8.0\_171-b10)

OpenJDK 64-Bit Server VM (build 25.171-b10, mixed mode)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Devops Class notes by Sunil sir.

What is a deployment ?

The .war file should get uploaded into the webapps folder.

1. Server method:

2. Client Method:

3. Manual Method:

4. Script Method:

cd

pwd -----------> print working directory.

ls -l

gitbucket.war

ls -lh gitbucket.war

sudo cp gitbucket.war /var/lib/tomcat/webapps ----> copy gitbucket.war file to /var/lib/tomcat/webapps

Once we copy , apply will start automatically. tomcat is watching the apply. once it gets the the new .war file, it will deploy the apply.

ip a ------> get the ip address in linux machine

go to the browser

http://servcer ip:8080/manager/login

right click gitbucker.war

because tomcat will start these applications automatically on reboot, infact all apply.

In UI(user interface) sing in user as developer and password : redhat. and create a remote repository. and also copy the url of the repository.

sudo systemctl enable tomcat.service

l s -lat

git init

git config --list

ls -al

git status

git add .

git status

git comitt -m " first cut"

ls -lat

vi .gitignore

target

:wq

git add .

git commit -m "first cut"

git remote

git remote -v

git push -u origin master

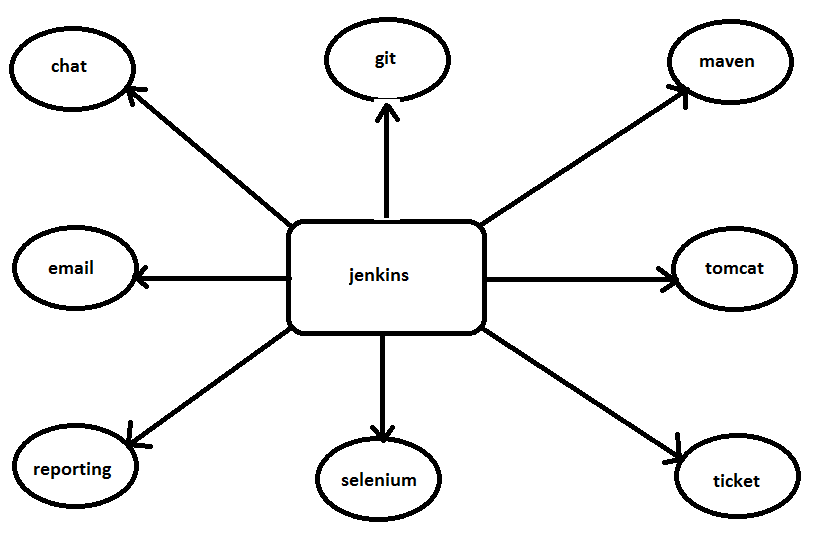
go to the browser (UI) and find your .war file.

What is the jenkins ?

It is a java based tool. it is used for CI/CD and Continius Integration and Continius Development/Deployment.

It support 1000+ Plugins(Add-on). based on architecture.

It supports project like java, .net, .pl, .rb, .py, .php



Plug-ins it's like a device driver connectivity both jenkins and others application

go to this url

https://jenkins.io/download

LTS ----Long term support

Generic java package (.war) <---------right click --------> copy link location

go to the terminal ------> wget <paste the link>

sudo cp jenkins .war /var/lib/tomcat/webapps -------> automatically tomcat call start jenkins.war

right click on the jenkins.war open a new tab. by default jenkins application is locked.

http://server ip:8080/jenkins

http://server ip:8080/gitbucket

http://server ip:8080/dxcwebapps

How to unlock jenkins

/usr/share/tomcat/.jenkins/secrets/initialAdminPassword ----->encrypted password of admin user of jenkins.

sudo cat /usr/share/tomcat/.jenkins/secrets/initialAdminPassword ----> copy the encrypted password

got to the browser tab where we open jenkins page and paste the password an unlock this admin user account.

It will force u to create user for acces jenkins. (first user is always admin)

Now move to the interface of the jenkins ----> select 2nd options------>

1) Install suggested plugins ------go with this option

2) user selected plugin

Dashboard of jenkins overview:

New items

Build History

New View

Credentials

Build Queue

Build Execution

1.

2.

fill in the blanks with username -------> continue.

What is a job ?

It's can be any task given by Jenkins to perform.

\* To run shell scripts in periodic manner.

\* To compile the code

\* To create a desired packages

What are the different types of jobs ?

1. freestyle :- support java, python, php, ruby, perl, .net, .sh

2. Maven :- Only Java

3. pipeline: cascaded sequence of jobs. i.e: job1(java), Job2(php), Job3(shell) .....

After finishing first job it(jenkins) will start another job automatically.

4. Multi configuration pipeline

J1 ----------> J2 -------------J3 Android

Ja -----------> Jb -----------> Jc IOS(Apple)

JA ----------> JB ----------> JC Go (Google)

go to UI (Jenkins) Interface

Create a new job (design-to success-job) ------>freestyle -------> directory----> check legend

Credentials ----------->to interact with 3 rd party application.

Check legend

Credential -----> To install with 3rd party application.

whenever we start our system it will automatically start tomcat.

1) gitbucket

2) jenkins

go to the browser:

http://server-ip:8080/jenkins

go to the terminal using putty.

Plug-in Manage Jenkins

What is plugin ?

It is piece of a software which will Jenkins to connect with 3rd party tools.

Method to install :-

1) Automatic

2) Manual

1) Automatic Method:-

Login Jenkins with admin user --------> manage jenkins ------> manage plugins ------->Update/Available/Installed/Advance

Integration with gitbucket

Download and install at restart (real-time)

go to browser http://server-ip:8080/jenkins

Login with last created admin user

manage Jenkins -----> manage Plugins ------->Available ---------> search for gitbucket and check

Download now & install after restart

http://server-ip:8080/jenkins/restart

Only admin will able to see this restart but not normal user for security. YES/NO

2) Manual Method: No readymade plugins will be available here.

1. search for plugin --------In searching area

2. Download & install the plugin.

extension:- .hpi (hudson plugin), .jpi (jenkins plugin)

ticketing system tools:- mantis/ jira/ otrs/ remedi/ bugzilla

here we are discussing about mantisbt

mantis: Its bug tracking tool(open source)

any bug created, jenkins will create a ticket using mantisbt

3 way of communication:-

1. chat 2. email 3. ticket

why we are go with ticketing system ?

we will go with ticketing system because of data (log reports) will stay for a years.

Search for jenkins mantis plugin info

ttps://wiki.jenkins.io/display/JENKINS/Mantis+Plugin

on the plugin site or download

https://plugins.jenkins.io/mantis

check the compatibility of versions -------> archives ------>0.20 --------> .hpi -----> install this now.

manage jenkins :-

manage jenkins ------>Advanced ---->upload plugin ------> browes your mantis pligin ------> it will checkm dependencies -------->wait & "restart Jenkins" options

restart the compulsory after installation of plugins

Global tool integration

example: git, jdk, maven, ant, gradil, docker

the tools which are helping you to create a package.

Integrate maven :-

Maven: Install plugin -----maven-3.5.4 (manual)

In real time we needed different versions of build tools. as we have to tell jenkins which version we want..

maven-3.5.0 (Automatic)

"for different project different version of build tools required.

anything we want to configure in Jenkins we have to go in manage Jenkins.

manage Jenkins -----> global tool configuration ------> maven section ------> add maven (because jenkins is not familiar to know maven has installed or not.

If already have any maven in past then no need to install automaticlly. just mention the address of the previous version. MVN\_HOME=/opt/apache-maven-3.5.4.

given name:- mvn 3.5.4 ----> Add maven but this automatic to method way & save.

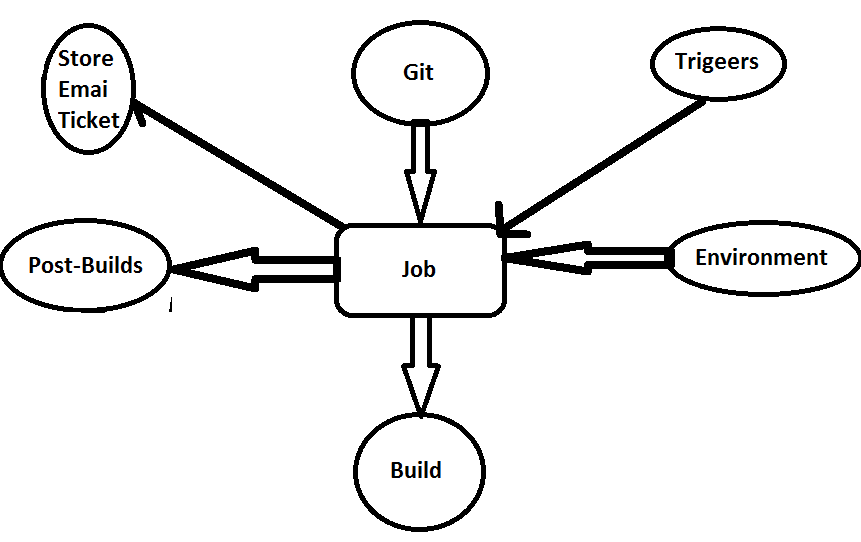
Now let's create job

what is a job ?

It is a task which can be perform, complie, test, packaging, operations, basically its a build operation and many more activities.

for example: we have a job i/p is taking from git and triggers to start the job and environment, then build ( It is a center of a job)

Build: compile, testing, packaging, & then post-build. extra activity after build is completes.



input: raw code through git

output: package as build

create job ------> entries a new--------> design-to-success[item name] ----->freestyle job ------> ok

bottm side ------> Build -----> Execute shell -----> command [date, pwd ] -----> save

Run this job -----> schedule the job & run the job ------>Build Scheduled. # -------> Console.

The output will be in the Console. Every job will have its own Console ------> here is the output.

Whenever we create a job, it will automatically create a directory with the same name as job.

Q: Where all my details are saved under Jenkins ?

Ans: JENKINS\_HOME= /usr/share/tomcat/.Jenkins

Note: Whenever we take a backup, we take a backup of this directory contents: jobs, usr, & plug-in, builds any change we will do in the Jenkins, it will store them in this directory. we can change this directory to others directory , we can migrate.

Create a new Job ---> New Item -----> design-to-fail-job ----->freestyle -----> Build -----> Shell Scripts -----> command [clearrrrrrrrrrr ] ----->save ----->run ----->Build Schedule.

Note:-[ To get maven project option, you have to install maven-plug-in. when we install mantisbt. it will come automatically.]

Any job we want to check, we have to go to the console it will show error or line 2 something like that..... because Jenkins has automatically added first line (shell line).

New item -----> dxcwebapps.compile -----> Maven Project ------> Ok.

Go to the gitbucket ------> http://server-ip:8080/gitbucket -----> login -----> copy the url of the project (which one we want to add in Jenkins) -----> paste the url in Jenkins and scroll down & in src code management.

.git -----> paste the url again. -----> Add credentials of the user created in gitbucket -----> once we select the new field. Jenkins go to that project present in gitbucket.

Build Trigger:- Don't select anythings

Build Environment:- Delete workspace before build start.

JENKINS\_HOME /workspace/dxcwebapps-compile -------> it will remove all unwanted files from that.

Build :- maven version (go with latest)

Goal:- compile

save directly.

go to dashboard: run the job.

Build executer status will show running job ------.> go to the console.

Note:- [ How many times you run the build, it will keep on incrementing the number (if we run again) ]

login system via putty

We have already created dxcwebapp-compile job. Now it's time to create a new job. which is dxcwebapp-test job.

New Item ------> dxcwebapp-test ------->copy from dxcwebapp-compile ----->Ok.

Build triggers ------>Build after other projects are build ----->project to which dxcwebapp-compile

Trigger only if build is stable

i.e: If our dxcweapp-compile job is successfully run then only start our dxcwebapp-test job.

Build ------> goals & options --test

Create a new item ------>package ----->copy from dxcwebapp-test -------> ok

Build triggers

Build after other projects are build

Projects to watch -dxcwebapp-test ------> trigger only if build is stable.

Build -----> goals & options - package

start dxcwenapp-compile -----> enable auto refresh page.

if our previous job fails then the further jobs wouldn't execute.

Disable auto refresh page. since jobs are executed.

New View (or + ):- non-project jobs -----> list view .my view ------> ok

select specific jobs in filters & give ok.

Lets add third view type.

Q: How to enable pipeline view ?

Ans: Install Build pipeline plugin (set the build pipeline view)

Manage Jenkins --------> manage plugins ------> Available ------> go to search bar. and type "Build pipeline" ----->download and install after restart. ------> wait for to complete download plugins

Login again http://server-ip:8080/Jenkins

New view -----> dxcwebapp-pipeline ------>Build pipeline view ----->ok

pipeline from -------> enable auto refresh (Jenkins web page)

Select initial job dxcwebapp-compile ----->ok.

everything shown in green that means is completed and if yellow means job is in running states. and if blue that means yet to run. so that we can know where is our job is fail and that stage.

Disable auto refresh. (Jenkins web page).

Q. How to setup auto deployment from Jenkins to tomcat web server ?

Ans:1) Make system ready from both sides.

2) Jenkins: (Install plugin) ---->Deploy to containers plugin

3). Tomcat ------->

create user , define the password

useradd:deploy

passwd: deploy

role: manager-script

4) restart both tomcat and Jenkins

5) modify the package job to talk with tomcat.

Manage Jenkins ------>manage pluf-in ------> Available ------>deploy to containers ------> download and install after restart.

login from putty

cd /etc/tomcat

sudo cp tomcat-users.xml tomcat.backup just take a backup

vi tomcat-users.xml

<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

<role rolename="tomcat-gui"/>

<user name="managermgr" password="tomcatmgr" roles="manager-gui" />

<role rolename="manager-script" />

<user name="deploy" password="deploy" roles="manager-script" />

</tomcat-users>

:wq

sudo systemctl stop tomcat.service

sudo netstat -ntpl | grep 8080 --------it will show the status of port no

sudo systemctl start tomcat.service

sudo netstat -ntpl | grep 8080

go to UI

http://server-ip:8080/Jenkins

go to the dash board

dxcwebapp -----> package ------> configure -----> Build setting -----> at last ------> post build

Deploy war/ear to a container.

war/ear files : \*\*/\*.war

containers /Add containers ----->tomcat 7.0

tomcat 7.0

tomcat url http://server-ip:8080

credentials : Add ------>Jenkins

username: deploy

password: deploy

ID: deploy deploy

description: deploy on tomcat

add:

select deploy users n tomcat

context path: /dxcwebapp ------>save.