**JENKINS**

It is very popular built and release tool

Jenkins help with CI (Continuous Integration) and CD (continuous development)

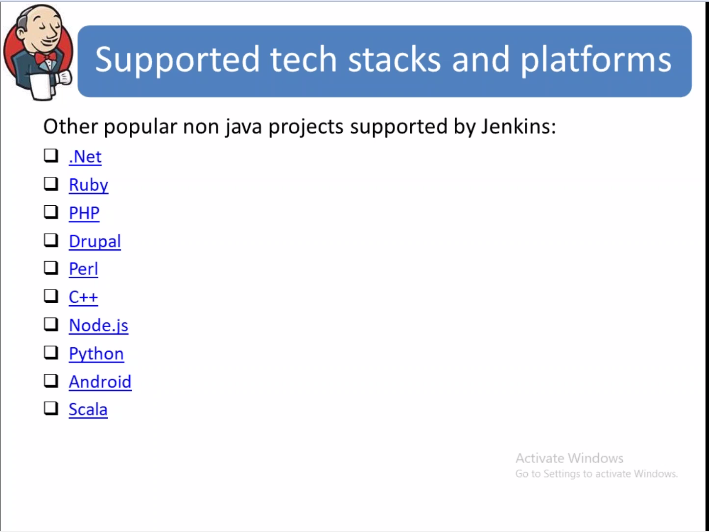
Entire Jenkins functionality is broken into plugins

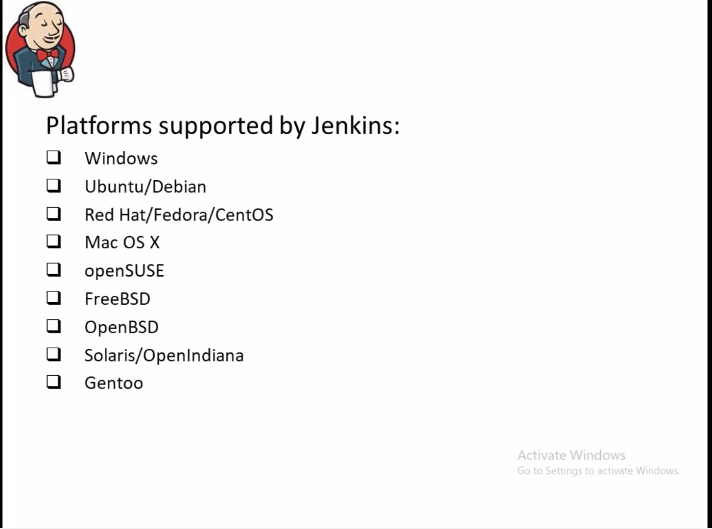
Jenkins is a JAVA based APP

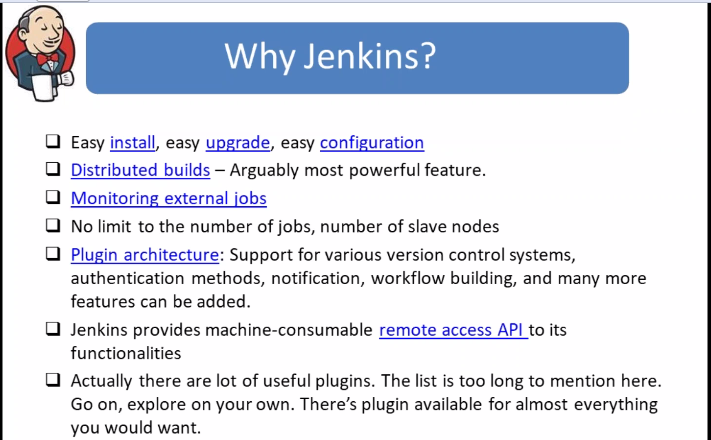
It can be installed any devices like (Windows any unix platform)

We have Jenkins plugin repository where lot of plugins are available even possible to build our own plugin if required

It is free and open source



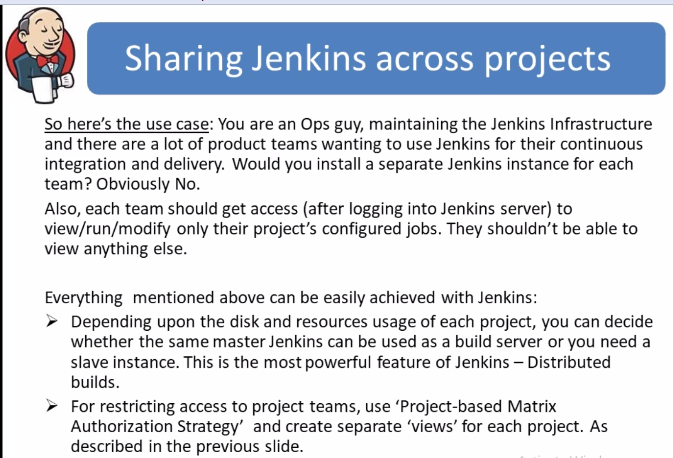


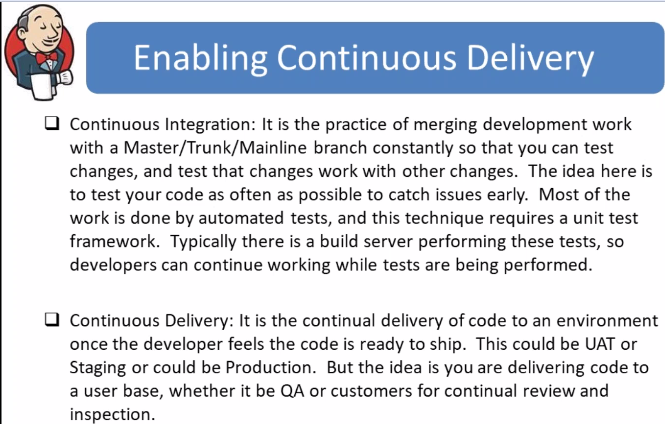


Jenkins have both GUI and CLI

Jenkins to run multiple builds it has cluster environment like master will monitor slaves where slaves will do actual work

Jenkins give granular level and flexible security controls it supports unix /etc/passwd authentication like if you are running Jenkins on unix machine create user in /etc/passwd and use it for Jenkins login or Jenkins has it own DB or it can be integrated with ldap for centralised authentication or it has project level authentication

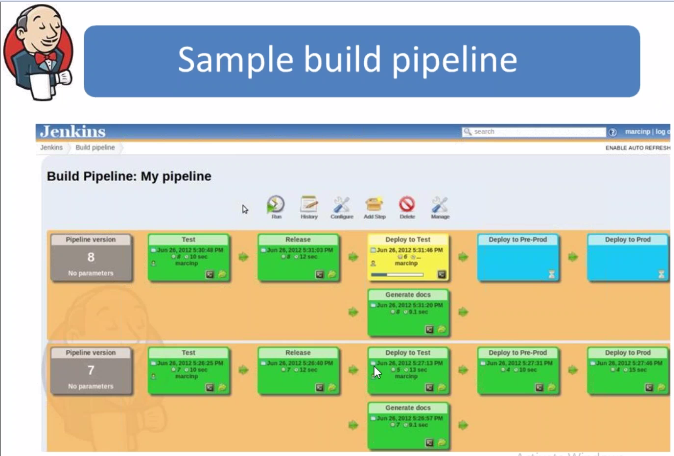




Deployment may require down time it depends on the environment if it has only 1 server

Other way is rolling fashion deployment where code is deployed on 1server and kept for observation if everything works fine then deployed on other nodes

Latest is blue green deployment where we will not even touch the existing environment like, we will build new VM servers and deploy all the latest codes and keep it for observation if everything is fine will add new server ip to dns or load balancer and remove the old ip and shutdown the old server if any issues we will start the old server and add this ips again in DN or load balancer removing new server ip’s how ever building a server and code deployment becames minutes to hours of work with latest technology



Pipeline of Jenkins 🡪 Different stages of Jenkins it can be different form APP to APP

Once the code is uploaded to VCS we will download that follow the jenkin stages

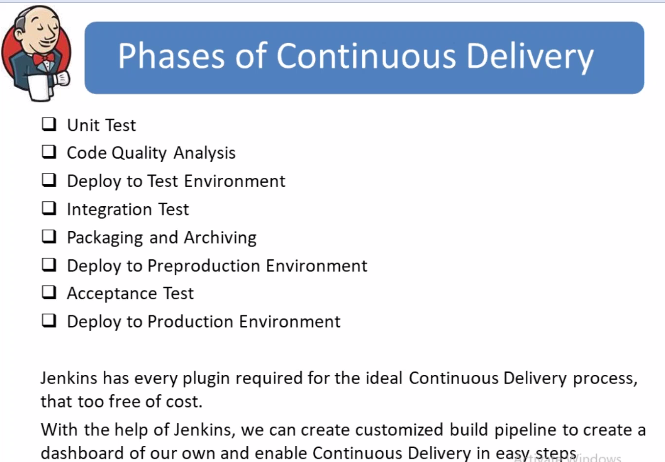
Test it is will be compiling stage

Release 🡪 is like testing code quality

Deploy to test 🡪 check in Test environment

Pre prod 🡪 Testing in Low env

Prod 🡪 Available to clients



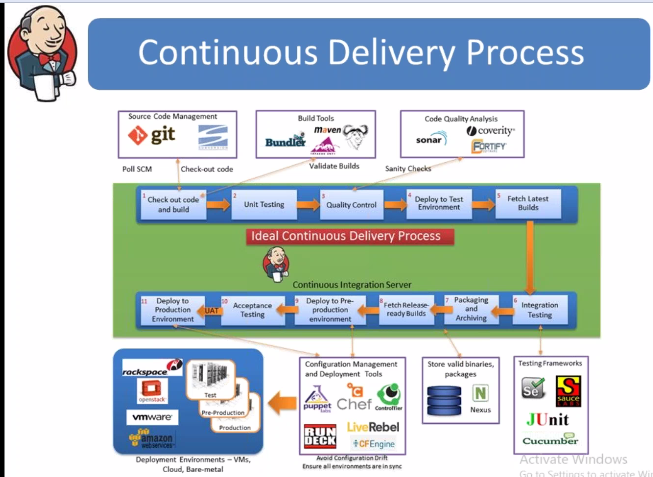
Unit test : Compiling phase

Code Quality Analysis : Companies will have some standards on following codes like Developers are restricted to use some libraries which are easily hackable or has some security issues and they need to follow documentation and many others

To test this code on policies there are tools like sonarcube it is used to run code quality test case and others.

In ansible we have ansilbe lint to test our roles or playbooks to check if we are following best practise

Remaining steps from above screenshot are normal



To practise Jenkins we need to have at least 2GB RAM

To install Jenkins pre-requisites are JAVA as the Jenkins are developed in JAVA

Jenkins Installation steps are below

yum install -y java

wget -O /etc/yum.repos.d/jenkins.repo <http://pkg.jenkins-ci.org/redhat/jenkins.repo>

rpm --import <https://jenkins-ci.org/redhat/jenkins-ci.org.key>

yum install -y jenkins

start the Jenkins service

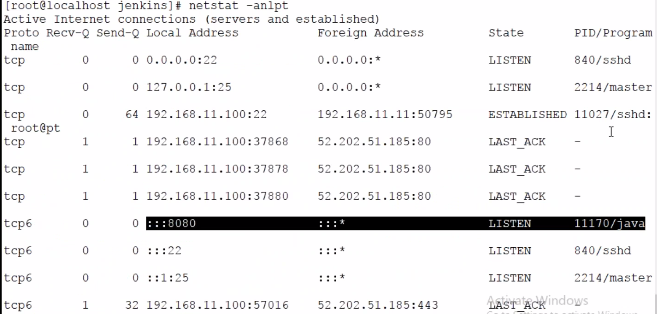
#systemctl start Jenkins

#ps –ef |grep –i Jenkins

From the above command we could see home directory and default port as below

Default path for Jenkins is and port is 8080

We can verify the same by command #netstat –anlpt



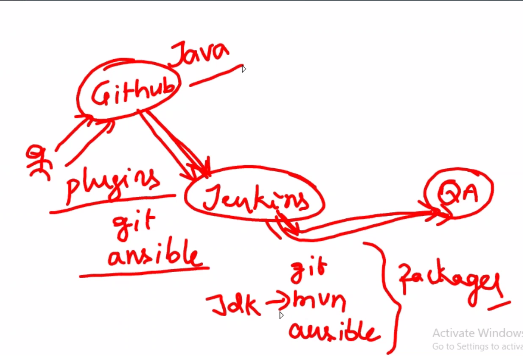
#cd /var/lib/Jenkins 🡪 All Jenkin configuration files present here

After installation to check Jenkins open browser and paste IP:8080

For the first time it will ask the password that is generated in the server path is

#cat /var/lib/Jenkins/secret/InitailAdminpassword

**Test case#**



To install plugins go to **manage plugins**

In Advanced we can create own plugin

For git plugin select as github plugin,Ansible plugin

Most of the time we need to restart Jenkins after installation of the plugins

From the above we have selected only 2 plugins but on the screen we could see many sub plugins are installed it is like yum in linux dependencies are installed automatically

At the bottom tick Restart Jenkins when installation completed and no jobs are running

Note the Jenkin version we are currently using (2.93 in video) because features will be different for version to version

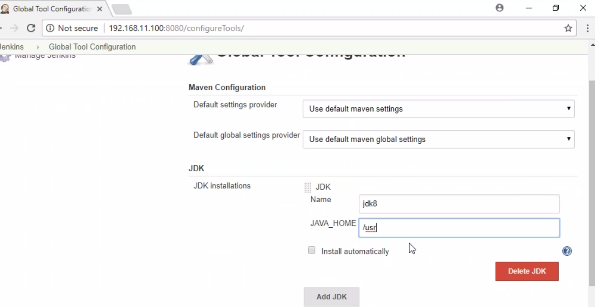
Some times while installing plugins it may take long time and it may even fail try to re install again

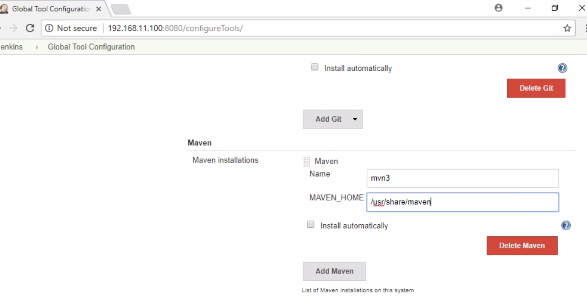
From the above screen shot we are doing below

1. On Jenkins configure Git and Ansible plugin
2. On serer we are installing packages Git,MVN,ansible and JDK
3. Java version and installation present in wiki -- http://wiki.polyglotit.com/
4. To check if java is installed successfully
5. Which javac (java compiler)
6. We can use different versions of jdk and maven on single server in Jenkins we need to provide the location of respective Jdk nd maven of respective projects

In Jenkins go to manage Jenkins and Global tool configuration we can configure JDK and Maven

Screen shot below and in wiki we have default home path for Jdk and Maven





Now select create new job in Jenkins

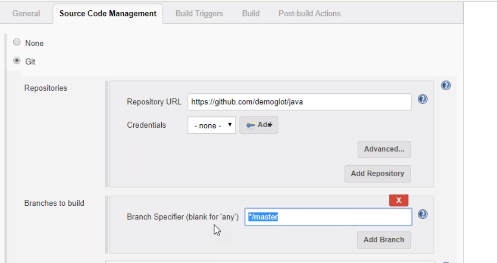
Enter an item name – any name randomly

We will select free style project

In source code management it will show only git as we installed that if needed any other we can install that plugin

In source code management select git and at repository URL : <https://github.com/demoglot/java>

Above provided URL is public accessible to no credentials are required if to access any private URL need to provide credentials before providing Public URL there will be red highlighted once we provide public URL and select no credentials Jenkins will validate

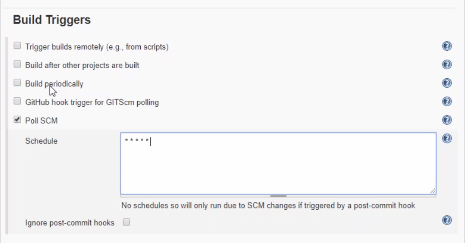


**Build Triggers**

It means how build should happen and when at the first time it will download code and it will build the code or compile the code

Next time when the git hub is updated or developed pushes latest code it should be updated but how does Jenkins know that, it has 2 ways

1. From the Github we can send trigger to Jenkins, but if git is in public domain and Jenkins is in Private domain it is not possible
2. Other way is that we will use pull technique and inform Jenkins to check git repository for every minute using crontab technique screenshot below



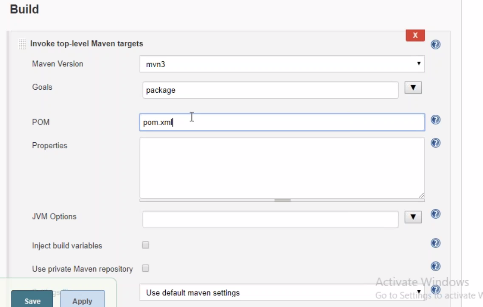
In Build

Select Invoke top-level Maven targets

Maven version will be different as per the project which we have provided

Goals will be provided by the developed they are different using “install”

POM is pam.xml file which we have provided the link if it is in subdirectory we need to provide those details



From the above steps compilation is done next by using ansible we need to push this to QA server

Save the code and check the project status