Jenkins:
========

Introduction to jenkins
Installation
Archetecture
master and salves configurations
what is job
creating jobs
scheduling jobs
labels
dependencys
Continuous Integration
maven project
Continuous Deployment
Email Notifications
sonarcube
jfrog
creating users
creating roles
backups
list and nested views
best-pratices
Blue Occean
CatLight Notifier

Without Continuous Integration:- usually when all the developers commits the code to repository(git), at the end of the day QA collects Integrate all the code and start testing (also called nightly builds), if any bugs found they've to wait up to next day morning to report to developer.

With CI:- whenever dev commits, instantly it integrate the changes and test(unit) it (hardly it takes 10-15minites to report to dev)

Benefits of CI:- @ immediate bug deduction
@ Minimal workflow:- integrating the latest changes and testing, if we want to continue like deployment and other steps...like pkg, deploy…etc Go head
@ we can deploy at any point of time
@ record the build history for tracking

List of CI/CD Tools:
----

*Hudson (Enterprice Licenced Tool)
*Jenkins
*Buildforge,
*Travis CI,
*Go CD,
*Continum,
*Anthi pro,
*Circle ci,
*Code fresh,
*Cruse control

*bamboo
*teamcity

# Hudson and Jenkins Both are same

-jenkins derived from hudson
-koshuke kawagachi(open source community developer)   (eclipse=> sun microsystems (oracle))

Why jenkins?

continuous integration
continuous deployment
jenkins has thousands of plugins which is used to connect to other tools also
jenkins is a frame work(you chose what process you want and ask jenkins to do)
jenkins Acts as cron(jobs) server replacement

Prerequisites:

Java 7 and above should be installed

Installation:

----

* Goto jenkins official website: https://jenkins.io/
* Click on Downloads
* Click on Windows under Long-term Support(LTS) session.
* Unzip the folder(jenkins-2.138.2) and install
    Note:- while installing jenkins redirect to any of D:/E:/F:/G:/ drive instated of installing in default C:/ drive
* Now open any of your web browser and type http://localhost:8080
* It will ask for Unlock Jenkins by giving Administrative Passward, in my case i installed in D:/ Drive (D:\jenkins 2.138.2\secrets\initialAdminPssword)
* Click on Continue and select Install Suggested Plugins
* Create First Admin User
* Click on Start using Jenkins

(OR)

* Install Jenkins through CLI(command line interface)
* Navigate to jenkins.war file in (D:/jenkins 2.138.2)
* And give this command   jave -jar jenkins.war
    Note:- If you want to change port for jenkins,then run jenkins on another port(9090) by giving this command
    jave -jar jenkins.war --httpPort=9090 in jenkins.war file navigation.

(OR)

* apache tomcat server
  jenkins.war

(OR)

1. wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
2. rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
3. yum install jenkins -y
4. systemctl / service start jenkins
5. systemctl enable jenkins

Archetecture:

10 projects
2hrs each to build
only we have 7 hrs to build

#code integration:

project m1, m2, m3...m100
no of issures will arise = 10000
we cannot predection of integration issures

```
|--|--|--|--|-------------|
 m1 m2 m3
m1=10issues, m2=10issues, m3 10issues  (1 hr_integrate + 7 hr_dev)
.
.
.
```

finally only 10 issures will be there

Configurations:

Global Tool Configuration
 >>Tools
 >>Environmental Variables
Job Configuration
 >>where to run
 >>when to run
 >>what exactly to run
Node Configuration
Master Configurations
Plugin Configuration

# Jenkins is client and server architecture, but no need to install Jenkins in both sides, only one i.e. Jenkins/master server.

Note:- if we not configure(specify) any node, then the jobs will run by default in jenkins master server where Jenkins installed.

Global Configuration(Go to Manage jenkins ---> Configure system)
  - global master settings giving to node
@ System Message:- is a Banner/User message who ever login as jenkins user
@ # of executions:- How many jobs a node can take and it depends on the hardware
      and nodes on which you running. (cpu and mem utilization).
@ label:- group of servers
@ Usage:- if we not configure(specify) any node, then the jobs will run by

default in jenkins master server where Jenkins installed.
E.g. if we want to get minimal things like time, mem usage…etc

@ Quiet Period:- before executing particular task Jenkins put job on hold(wait)
for 5 sec e.g. any network issues…etc

@ SCM checkout retry count:- if jenkins is not connecting to any SCM Tools then it
retries to connect.

- environment variables
- Build tools info
- scm tools

2. Node (servers) Configuration
3. Job---> Group of tasks
   - What
   - How
   - When

Note:- By default Jenkins run continuous integration

Creating Jobs:-
This project is parameterized: passing parameters as values
Discard old builds : logs
Restrict where this project can be run : in which slave the job should run

Labels
------
if jobs are running in one server, all of sudden it went down then the jobs in that server will not run.so, to overcome this
in jenkins by grouping the servers and label them with a name and assign jobs.
Ex:-

```
           Server x
           job 1            MY_SERVERS
           job 2            centos
           job 3            febora
                                       Redhat
                                       ubuntu
           Server y
           job 1
           job 2
           job 3
```

CI:
--

```
Jan|---------|-------|--------------------|------|--------|July
   RA        DE      feb    devphase      may    QA     prod
```

# devops guy is responsible to perform a CI in a project
  -- Repository(scm/svn) : Poll Scm
  -- integrating the changes
  -- build(Incremental build)

>> resource
>> compile
>> test
>> package
>> install
>> deploy
>> code coverage
>> Static code analysis

-- report to developers
-- action
-- fixing by developers
CI is routine task (integrate & report)

Scheduling:

- On Demand (Build Now)
- time based (Build Periodically)
- poll based (Poll SCM)

Poll SCM : it proceeds for first time and sinks, then 2nd time it will not proceed unless and until new changes happened. (CI job)
Build Periodically: normal job

Dependences:
===========

(post build) : if the build get failed, now I need to send a email……

Up stream: before it proceeds to execute a job, it will check for dependences job and execute

Downstream: once we complete the job at the end it calls another job

Note:- Jenkins Servlet Containers link
https://wiki.jenkins.io/display/JENKINS/Tomcat
Environment Variables:-
User variables: specify path including bin directory
System Variables: it is HOME where software installed.

Automated Deployment:
====================

DEV--->BUILD-->TEST-->RELEASE

start your jenkins
install deploy plugin
create a build job in jenkins
add post build actions war/ear containers
keep war file in jenkins home workspace
add a user,role in tomcat-user.xml file
run and validate

Servlet Containers:- these are containers where Jenkins can install and run, tomcat is very popular.
Glassfish

Tomcat
JBoss
Jetty/Winstone: default servlet container for Jenkins
WebLogic
IBM WebSphere

Tomcat:

Download and unzip tomcat and place it in one folder
And start tomcat by startup.sh file in bin folder(bash startup.sh)
Start tomcat : ./startup.sh   or bash startup.sh(in git)
Verufy if tomcat started by going browser – http://localhost:8080
Verify if Jenkins is running on tomcat : http://localhost:8080/jenkins

By default Jenkins also run on port 8080. So, now run Jenkins in standalone.
--> If i want to run Jenkins in standalone then go to Jenkins folder and type below
      Java -jar Jenkins. War –httpPort=9090. Note: - both will not run on same port 8080.

Note:- Here in Port “P” is capital
Note:- to change port in tomcat got to config-->server XML files
Note:- to chang port for jenkins go to D:\jenkins_home-2.73.2\   jave -jar jenkins.war --httpPort=9090
Note:- to change user,role, and password got to tomcat-users xml files, and give below details

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-jmx"/>
<role rolename="manager-status"/>
<role rolename="admin"/>
<user username="admin" password="admin" roles="admin,manager-gui,manager-script"/>
```

delivery pipeline:

create some jobs
and link one job to other
install delivery plugin
start plugin

Email Notification:

Configuration: Go to jenkins manage >> config system >> Email Notification(bottom of the page)

SMTP : smtp.gmail.com (for other mial servers go to https://www.arclab.com/en/kb/email/list-of-smtp-and-pop3-servers-mailserver-list.html)
Default e-mail suffix : @gmail.com
Use SMTP Authentication : username

: passwd
Use SSL
SMTP Port : 465

Errors: 1) Login to your gmail account.

2) Go to https://www.google.com/settings/security/lesssecureapps and Turn On this feature.

3) Go to https://accounts.google.com/DisplayUnlockCaptcha and click Continue.

(OR)

Stack Exchange Link :- https://serverfault.com/questions/635139/how-to-fix-send-mail-authorization-failed-534-5-7-14

JSON and XML format notification plugins, below link
https://wiki.jenkins.io/display/JENKINS/Notification+Plugin --->Notification plugins

for HTML format notification plugins,below link
https://wiki.jenkins.io/display/JENKINS/Extreme+Notification+Plugin

for advanced email notifications,below link
https://wiki.jenkins.io/display/JENKINS/Email-ext+plugin

Note: install email extension plugin for normal email notifications
Note: by default when build get fails it will send emails and once the same build is success, then
this time it sends success mail also.

jenkins Views:

In case if you have hundreds of jobs in your jenkins dashboard,the how to view specific catagory jobs.
1.list view
2.nested view (install nested view plugin)

Maven Project:
----

maven project setup in jenkins, go to--> global tool configuration
maven path
JDK path

Changing Jenkins Home Dir:

Why: To move from home dir to location where enough space.
project requirement also.
-->C:\Users\vinodh\.jenkins--> All confs, plugins,Logs, secrets...etc/profile
i moved .Jenkins folder files to new folder, And give env variables in system var

Restart:

--> control + c
--> java -jar jenkins.war   (OR) http:/localhost:8080/restart/
Note:- http:/localhost:8080/systemInfo --> you will get all sys info("I" Capital)

Command line interface (CLI):- it is very easy, faster, memory management, continues integration
----

Go to manage jenkins-->Configure global security-->enable security
http://localhost:8080/cli/
-->download cli.jar and test

Users:

Create new users
Configure users
Create and manage user roles
Roles strategy plugin – download – restart Jenkin

CatLight:

>> status notifier for developers
>> catlight will notify your when builds,bugs and tasks need your attention.
>> it is very handy and useful when you have to manage multiple job

1. Choose things to track
2. See status in tray
3. Get notified about the changes

https://catlight.io/

Blueoccean: Look and feel of jenkins pipelines,jobs,nodes...etc
----

continuous delivery
----

cd is a step that we will do on the top of ci on which we deploy the application on production like systems(pre-production) and ferform some automation tests.

note:- services.msc

```
<build>
<plugins>
  <plugin>
    <groupId>org.jacoco</groupId>
    <artifactId>jacoco-maven-plugin</artifactId>
    <version>0.5.5.201112152213</version>
       <configuration>
          <destFile>${basedir}/target/coverage-reports/jacoco-unit.exec</destFile>
          <dataFile>${basedir}/target/coverage-reports/jacoco-unit.exec</dataFile>
       </configuration>
          <executions>
             <execution>
                <id>jacoco-initialize</id>
                   <goals>
                       <goal>prepare-agent</goal>
                   </goals>
             </execution>
             <execution>
               <id>jacoco-site</id>
                   <phase>package</phase>
                   <goals>
                      <goal>report</goal>
                   </goals>
            </execution>
       </executions>
  </plugin>
</plugins>
 </build>
```

SonarQube: static code analysis
----

https://www.sonarqube.org/downloads/
http://localhost:9000
Manage Jenkins --> Configure System -> SonarQube Servers -> Add SonarQube
Download - https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner
Manage Jenkins --> Global Tool Configurations -> SonarQube Scanners -> Path of scanner home directory
Goto jenkins job -> Add new ->Build ->Add build step ->Execute SonaQube Scanner ->Analysis properties

```
 # Metadata
   sonar.projectkey=VINU
   sonar.projectName=SonarDemo
   sonar.projectVersion=1.0
 # Give path to src directory of maven project
   sonar.sources=src
   sonar.jacoco.reportPath=target\\coverage-reports\\jacoco-unit.exec
```

jfrog/Nexus

```
jfrog is a place where we store binarys
Download https://jfrog.com/open-source/
login http://localhost:8081/artifactory/
--> install artifactory plugin form manage plugins
--> manage jenkins ->configure system ->artifactory
     >> uncheck Enable push to bintray option
     >> pass Server ID
     >> URL (Jfrog artifactory url like http://localhost:8081/artifactory)
     >> UserName |
                 |------>>JFrog Artifactory Credentials
     >> Password |
```

>> And click on test connection

>> Goto job ->configure ->Build Environment -> Maven3-Artifactory Integration ->refresh for default

```
>> Goto Build -> Invoke Artifactory Maven 3 and
              -> Execute SonarQube Scanner
```

>> finally run the job

note:- while running jfrog aritfactory batch file, if it shows "could not reserve enoughf space for object heap" then
open artifactory batch file in notepad++ and replace "Xms2g" to "Xms1024m" in rem defaults session.

Job Customization(View):

default view will be All

Maintanance:

http://localhost:8080/jenkins/exit -->> to shutdown jenkins
http://localhost:8080/jenkins/restart --->> to restart jenkins
http://localhost:8080/jenkins/reload --->> to reload the jenkins configuration

Security:

Configure Global Security

1. Security Realm :- who to login into jenkins
   Single sign on(SSO)
   here jenkins has it's own database
   Lightweight Directory Access Protocol (LDAP)
2. Authorization :- Once login into jenkins, what are the permissions required to user..
   Matrix-based security

Creating Users,Manage And assign Roles:

Prerequisites: 1) install Role-based Authorized plugin
2) And enable Role-Based Strategy in Authorization section of Configure Global security option in "manage jenkins"

* Create users by going to manage jenkins >> manage users >> create user
* configure user by going--> "vinodh" right top corner of jenkins page
* assign and manage roles to users
* validate by creating sample jobs

Jenkins Backup Home:

# Jenkins home directory :- where we store all the information about jobs,builds,nodes,logs,plugins,config....etc

# if we want to take the entire jenkins backup we need to copy the jenkins home dir like "A:\jenkins" and place in other system.
# (or) we have a plugin called "backup" plugin.

Jenkins Best Pratices:

$ Secure Jenkins always (Configure global security)
$ Not to run many jobs in jenkins master server. only run on critical situations like backups...
$ Backup jenkins home directory regurarly.
$ Setup your jenkins on partition which has more free disk space.
$ archive unused jobs before removing
$ not to schedule all the jobs at the same time(jenkins performance issue)

Pipeline:
========

## Pipeline is workflow with group of events/jobs that are chained and integrated with each other in sequence.
## and every job in pipeline has some dependencys on one/more jobs

Build Pipeline Plugin:- rather then following the jenkins standards, we can create our own workflow/process.

>> Step:- Specify what step we want to do first,2nd, 3rd....etc
eg:- after build i do not want to connet to SCM
after completing of one task i want to connect to scm

>> Stage:- collection of steps
and we can keep group of steps in one stage and specify where you want to run

>> Groovy Script:- Advanced script which is only designed for jenkins.

Declaritive

```
pipeline {
    agent  {
       node  {
          label "mygroup"
          customworkspace "A:\sam/PIPE"
       }
   }
   triggers {
      cron('* * * * *')
   }
   stages {
       stage('stage1') {
           steps {
              echo 'hello welcome to groovy script stage I'
           }
       }
       stage('stage2') {
           steps {
              echo 'hello welcome to groovy script stage II'
           }
       }

   }
   post {
           always {
               echo 'End of groovy script pipeline'
           }
       }
```

Defect/Issue/Bug Tracking Tools:
================================

ZIRA, BugZilla, Mantis BT, HP-ALM, IBM Rational Clearquest

What is Bug:- Any kind of problem/enhancement which requires some change is called bug

workflow:- detcting the bug
> when we testing
> while doing any enhancements
> customer identifies...

plugins
----

extra futures and functionaltys which is not getting by default jenkins installation.

Delivery pipeline
Build pipeline
Deploy
Role based Authorization strategy
Nested plugin
Jacoco (code coverage)
SonarQube Scanner (Stack Code Analysis)
findbugs
artifactory plugin
BlueOccean
Parameterized Plugin
Pre SCM buildstep plugin
Job configuration history plugin
Build failure analyzer
Matrix Authorization Strategy
Build timeout
Throttle Concurrent builds