**2.2 What can Jenkins do?**

• Generate test reports

• Integrate with many different version control systems

• Push to various artefact repositories

• Deploys directly to production or test environments

• Notify stakeholders of build status

**2.3 How Jenkins work - Setup**

• When setting up a project in Jenkins, there are following general options – o Associating with version control server

o Triggering builds Polling, Periodic, building based on other projects

o Execution of shell scripts, bash scripts, Ant and Maven targets

o Artifact archival

o Publish test results

o Email notifications

**2.4 How Jenkins work - Building**

• Once project is successfully created in Jenkins, all future builds are automatic

• Building o Jenkins executes the build in an executer By default, Jenkins gives one executer per core on the build server

o Jenkins also has a concept of slave build servers Useful for building on different architectures

Distribution of load

JENKINS->Poll for changes in GIT->Builds Artifacts(exe,website,app)->Run tests(Nunit,MSTest,Gradle,StyleCop)->Deploy (Publish output to live server)->Report back

Start jenkins: Java -jar jenkins.war

Changing jenkins home directory-> create new folder->copy and paste all files from old dir->change env variable-> JENKINS\_HOME and set to new dir

Windows->My computer->prperties->advanced system settings-> environment varaibles-> edit-> valuse and path

Linux-> terminal->export JENKINS\_HOME~/ "new path"

restart jenkins

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

Basic configurations:

Manage Jenkins-> Configure system, Configure global security,Manage plugins, manage users,jenkisn cli, script console, manage & assign roles

Manage jenkins->configure system :

Home directory, system message, # of executers- parallel jobs,labels- nodes,Quiet period, SCM Checkout retry count-if fails

Restrict project naming-> Naming startegy->Default, Pattern-> Name pattern, Role based strategy, Global properties-> Environment variables

Jenkins location -> Jenkins URL :http://localhost:8080/, system admin email address, ssh server: SSHD Port, shell->shell executable : executable path (default)

manage jenkins->Configuring global security-authorization->role based authorization strategy-> save , restart jenkins

manage jenkins->creating users-> Manage users-> create user, configuring users

manage jenkins-> manage and assign roles-> manage roles, assign roles & role based strategy macros

manage roles->Global roles--employee , project roles--roles to add: developer, tester, Pattern : DEV, TEST

Assign roles-->Global roles-- Add users, Project roles-assign developer & tester to users

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

Getting started with JOBS:

Jenkins-> New Item->Enter job name->Freestyle/multi conf

Configure project:

General->project name,restrict where project can be run

source code management->Git,

Build triggers->trigger build remotely, build after other projects are built,Build periodicaly : cron job

Build->add build step : Execute windows batch command, execute shell, Invoketop-level Maven targets,

Post build actions-> Add build action : Aggregate downstream report, Archive the artifacts, build other projects, Publish Junit test result report

Project name->status, changes, build now, Delete project, configure, build history-> select build-> console output, delete build, previous build

S-> Last status build, W-Last five runs of particular project

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

How to trigger job remotely: project->configure->build trigers->Trigger build remotely->copy the URL and give authentication token name

How to chain job executions : proj1 (build step-shell command), proj2 (build step-shell command), proj3 (build step-shell command)

select proj1-> configure-> build triggers->build after other projects are built-> Projects to watch-> proj2 name and then

Post build actions->build other projects->projects to build->proj3 name

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

Jenkins Intergration with GIT:

1. create java program or file in git Project

2. Adding project in git to github:

Goto project location in git folder-> initilize the repository (git init) and then git status

add untracked files with git add . & git commit -m "added java program"

--Goto github account and create project-> create repository (hello world) and copy the location (URL) of repository ssh or https--

git remote add origin and paste the url of the repository

git push -u origin master

3.Create jenkins job to run java program->configure->source code management->Git(Install git plugin)-> paste the repository URL and give github credentials

4. Build triggers->POLL SCM->Schedule->cron job (\*\*\*\*\*), apply and save.

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

Automated deployments: It is the process of Automating the deployment process in a Continuos delivery system

Main stages is Continous delivery and deployment pipeline are Build->Deploy->Test->Release

Developers--commit -> VCS--Poll for changes/checkout&build -> Build job--war/ear artifacts -> Deploy--Functional test env & Performance test env -> Deploy--Production env

How to do Automated deployments with Jenkins:

1. start jenkins : java -jar jenkins.war --httpPort 9090

2. Install Deploy Plugin : deploy to container plugin

3. Create a build job in jenkins like AutomatedDeploymentTest

4. Get war/ear file to deploy like sample.war

5. Add post build action-> Deploy war/ear to container->WAR/EAR Files, Context path (Jenkins home-workspace-AutomatedDeploymentTest-war file) :sample.war, Containers:Tomcat 7

6. In tomacat-users.xml add users for deployment : <user=username"deployer"password="deployer"roles="manager-script"/>

7. Start tomacat : cd /users/raghu/Desktop/tools/tomcat/bin/ | .startup.sh

8. Run and validate : goto job AutomatedDeploymentTest -> select build now and select console output

--On first job run, log will say -Doing a fresh deloyment, on subsequent run logs will say - Redeploying

9. war file will available in Tomcat>webapps folder

10. goto tomcat browser (localhost:8080/sample), it will displays the application

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

How to send email jenkins notifications:

Manage jenkins-> Configure system-> Email notifications : SMTP Server, SMTP Authentication, test configuration--test email recipient

Goto any jobs-> Configure->Post build actions-> E-mail notification-> Recipients

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

Jenkins Pipeline: what is pipeline in Jenkins - Pipeline is a workflow with group of events or jobs that are chained and integrated with each other in sequence

Every job in a pipeline has some dependency on one or more other jobs.

How to set up Delivery pipeline in Jenkins :

1. Chain required jobs in sequence -> Add upstream (a job to be executed before current job) & downstream job (a job to be executed after current job)

2. Create sample jobs : SampleBuildJob (build step-shell command), SampleDeployJob (build step-shell command),SampleTestJob (build step-shell command)

3. select SampleDeployJob-> configure-> build triggers->build after other projects are built-> Projects to watch-> SampleBuildJob name and then

SampleTestJob-> configure-> build triggers->build after other projects are built-> Projects to watch-> SampleDeployJob.

4. Install Delivery pipeline plugin : Manage Jenkins-> Manage Plugins-> Available/Installed -> Delivery pipeline plugin

5. Add delivery pipeline view -> Configure the view -> Goto jenkins dashboard and click on + icon, name the view as "TestDeliveryPipeline" select delivery pipeline view.

Goto pipeline section -> Component name and give the initial job name "SampleBuildJob" Apply and save. It shows Delivery pipeline view

6. Run and validate

--Options: Enable start burtton for new piplin

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

How to setup Build pipeline in Jenkins :

1. Chain required jobs in sequence : same process above in step 2 & 3

2. Install Build Pipeline plugin : Manage Jenkins-> Manage Plugins-> Available/Installed -> Build Pipeline

3. Add Build Pipeline View -> Configure the view -> Goto jenkins dashboard and click on + icon, name the view as "BuildPipelineTest" and select the Build Pipeline View and ok

It opens configuration window. In this, select Pipeline flow -> Select initial job : SampleBuildJob Apply and save. It shows Delivery pipeline view

Configure-> Display options -> No of dispalyed builds : 5

4. Run and validate

--Options : Run - To trigger the build pipeline jobs, Link for job - To navigate to job window, Configure-> Console output link style: New window - view console output

**2.5 How Jenkins work – Reporting**

• Jenkins comes with basic reporting features o Keeping track of build status Last success and failure

‘Weather’ – build trend

• These can be greatly enhanced with the use of pre-build plugins o Unit test coverage

o Test result trending

**2.6 Enhancing Jenkins**

• Jenkins plugin system can enable a wide range of features including (but certainly not limited to) o SCMGit, Subversion

o Testing Selenium, TestLink

o Notifications Twitter, Jabber

o Reporting Findbugs, Doxygen

o Artifact Saving Amazon S3, SCP

o Triggers Jabber, Directory watchers

o External integration GitHub, Bugzilla, JIRA

**2.7 Running Jenkins yourself**

• Jenkins is packaged as WAR, so it can be dropped to whichever servlet container we prefer to use

• Native/supported package exist for o Windows

o Ubuntu/Debian

o Redhat/Fedora/CentOS

o Mac OSX

o openSUSE

o FreeBSD

o openBSD

o Solaris

o Gentoo

#############JENKINS INSTALLATION STEPS################################

wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins

sudo cat /var/lib/jenkins/secrets/initialAdminPassword 0f31b7a349464af78811224add08b79a

Sudo nano /etc/default/jenkins #to change the port

Deploy Plugin is required; Deploy to container plugin from Manage Jenkins -> Manage Plugins

• Create a new freestyle project by the name "FirstWebApp"

• Configure -> Source Code Management -> Git

• Repository URL: https://github.com/raghupss/WebAppSample.git

• Configure -> Build -> Invoke top-level Maven targets

• Goals -> package

• Post build actions -> Add post -build action -> Deploy WAR/EAR to a container

• WAR/EAR files: Provide the WAR name (e.g. target/firstwebapp.war)

• Context Path: firstwebapp

• Containers: Tomcat 7.x (NOTE: works for Tomcat 8.x also)

? Manager user name: Give the one that we have configured in the /var/lib/tomcat8/conf/tomcat-users.xml (e.g. deployer)

? Password: give the password (e.g. deployer)

? Tomcat URL: e.g. http://13.126.110.18:8081 (replace it with your server's public IP)

? Apply and Save the build settings

• Build the project with "Build Now" option

• Go to "Console Output" and ensure build and deploy steps succeeded.

RUN THE PROJECT: go to browser and give the tomcat URL along with the Tomcat's deployment folder name and the file name (e.g <http://13.126.110.18:8081/firstwebapp/first>

Build-Package your code into executable files

Continuos testing - Junit abd selenium