**1. Which tool have you used for implement CI/CD?**

ANS: Jenkins

**2. Any alternate tool do you know for CI/CD?**

**ANS:** There are many other CI tools, and the most prominent ones are –

* TeamCity
* Bamboo
* Perforce
* Circle CI
* Go
* ThoughtWorks
* Integrity
* Travis CI.

3**. What is Continuous Integration?**

ANS: get the source code management ---compile--test--build--static code analysis--nexus--build failure send notification

Continuous integration is a step in which all code is merged as developers complete code in order to run automated builds and tests

**4. What type of jobs have you configured in Jenkins?**

ANS:

* Freestyle project.
* Maven project.
* Pipeline.
* Multibranch pipeline.
* External Job.
* Multi-configuration project.
* GitHub organization.

**5. What are the types of jobs are available in Jenkins?**

**ANS**

* Freestyle project.
* Maven project.
* Pipeline.
* Multibranch pipeline.
* External Job.
* Multi-configuration project.
* GitHub organization

**6. What is difference b/w freestyle and pipeline?**

**ANS:** We can integrate any type of files. In freestyle

In pipeline Sequence of jobs, one job is connected to another job.

**7. What is pipeline?**

**ANS:** Jenkins Pipeline (or simply "Pipeline") is a number of things forming a series or set. of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins. A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers

**8. What is declarative pipeline?**

Declarative pipeline is a relatively new feature that supports the pipeline as code concept. It makes the pipeline code easier to read and write. This code is written in a Jenkins file which can be checked into a source control management system such as Git.

**9. What is script in Jenkins pipeline?**

The script has the elements “pipeline”, “agent” and “steps” which are specific to Declarative Pipeline syntax; “stage” is common to both Declarative and Scripted; and finally, node” is specific for the Scripted one. “Pipeline” defines the block that will contain all the script content.

**10. What is difference between declarative pipeline and script based pipeline?**

**Scripted Pipeline Job**

node('SlaveSCMConnected') {

stage('SCM operation'){

checkout scm

sh 'mvn clean install'

}

}

**Declarative Pipeline JOB**

pipeline {

agent none

options {

skipDefaultCheckout()

}

**11. Write the pipeline syntax?**

pipeline {

agent any

tools {

maven 'apache-maven-3.0.1'

}

stages {

stage('Example') {

steps {

sh 'mvn --version'

}

}

}

}

**12. What is master/slave architecture? What is the use of master/slave?**

Master/slave is a model of asymmetric communication or control where one device or process (the "master") controls one or more other devices or processes (the "slaves") and serves as their communication hub.

Jenkins supports the master-slave architecture, i.e. many slaves work for a master. It is also known as Jenkins Distributed Builds. It also allows you to run jobs on different environments like Linux, Windows, MacOS, etc.

**13.** **How many we can connect the slaves?**

You have a 7-bit address. That means you can address up to 127 slaves."

Jenkins uses a Master-Slave architecture to manage distributed builds. In this architecture, Master and Slave communicate through TCP/IP protocol. Your main Jenkins server is the Master

**14. How many ways we can provide security for your Jenkins server?**

#### ANS: There are 3 ways –

* The default way is to store user data and credentials in an internal database.
* Configure Jenkins to use the authentication mechanism defined by the application server on which it is deployed.
* Configure Jenkins to authenticate against LDAP server.

**15. What is sonarqube? Have you configure? How you configure?**

**16. What type of artifactory repository tool have you used? nexus,jfrog/s3**

**17. What is the use of artifactory tools?**

**18. How you declare a variables in pipeline?**

**ANS:** Variables in a Jenkinsfile can be defined by using the def keyword.

Such variables should be defined before the pipeline block starts.

When variable is defined, it can be called from the Jenkins declarative pipeline using ${...} syntax.

**19. What is DSL language? Domain specific language?**

**ANS:** Let me introduce **Jenkins** Job **DSL** / Plugin, a project made up of two parts: the **Domain Specific Language** that allows users to describe jobs using Groovy-based **language**, and a **Jenkins** plugin which manages the scripts and the updating of **Jenkins** jobs which are created and maintained as a result

**19. What is upstream/downstream projects? What is the use of it? Which scenario you configure?**

**ANS:** A **downstream** job is a **configured project** that is triggered as part of a execution of pipeline. **Upstream** and **downstream** jobs help **you** to **configure** the sequence of execution for different operations and hence **you** can orchestrate the flow of execution.

**20. If a have 10 repositories in GitHub how many jobs you can configure?**

**21. Dou you have experience to install Jenkins?**

**Yes**

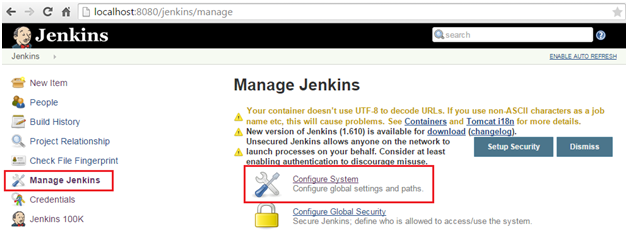
**22. How you configure jdk, maven, gradle...etc?**

**23. In my environment i have different version for java implementation projects is there? How you configure multiple jdk's?**

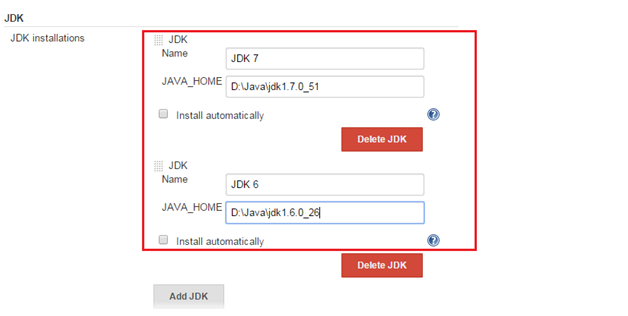
**Configuring JDK in Jenkins**

Below are the steps to be followed in order to configure the JDK in Jenkins:

* Deploy the Jenkins.war into the tomcat server and start the server.
* From the home page, click on the link **Manage Jenkins** and click on **Configure System** as shown below :

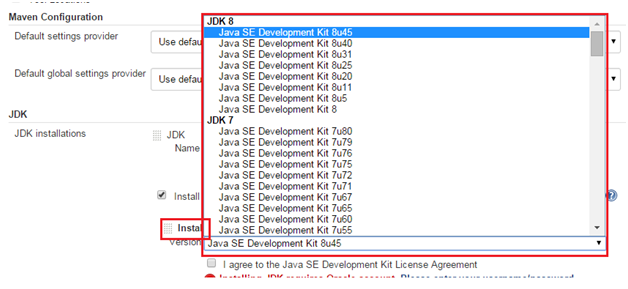


1. From the next page, click on "Add JDK". The easiest way to configure JDK is to provide a name and path of the java installation directory. (Consider the same path used for JAVA\_HOME). In the below screenshot JDK 6 and JDK 7 has been configured,



**NOTE:** The Jenkins will validate to check whether the installation directory exists for the specified JDK versions.

1. JDK can also be installed by intimating Jenkins to do the task. The check box **install automatically** needs to be checked in this case. Jenkins will download and installs the specified version of the JDK into the tools directory of the Jenkins home directory.



1. After all the settings made to configure the JDK, just click the "Save" at the bottom of the page to save the configuration.

**24. What are the plugins have you used in your project?**

**25. How to take backup my Jenkins? Thin backup plugin?**

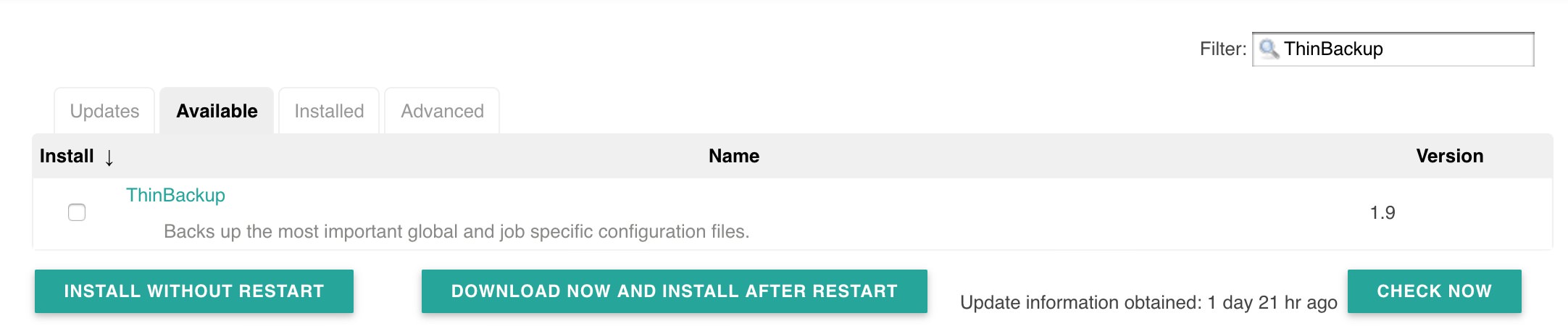
## **Jenkins Backup Using Thin Backup Plugin**

[Jenkins Thin Backup](https://wiki.jenkins.io/display/JENKINS/thinBackup) is a popular plugin for backing up Jenkins. It backs up all the data based on your schedule and it handles the backup retention as well.

To get started, first install the plugin.

1. Go to Manage Jenkins –> Manage Plugins

2. Click the Available tab and search for “Thin backup”

[](https://devopscube.com/wp-content/uploads/2017/07/Screen-Shot-2017-07-31-at-7.02.30-PM_mini.jpg)

3. Install the plugin and restart Jenkins.

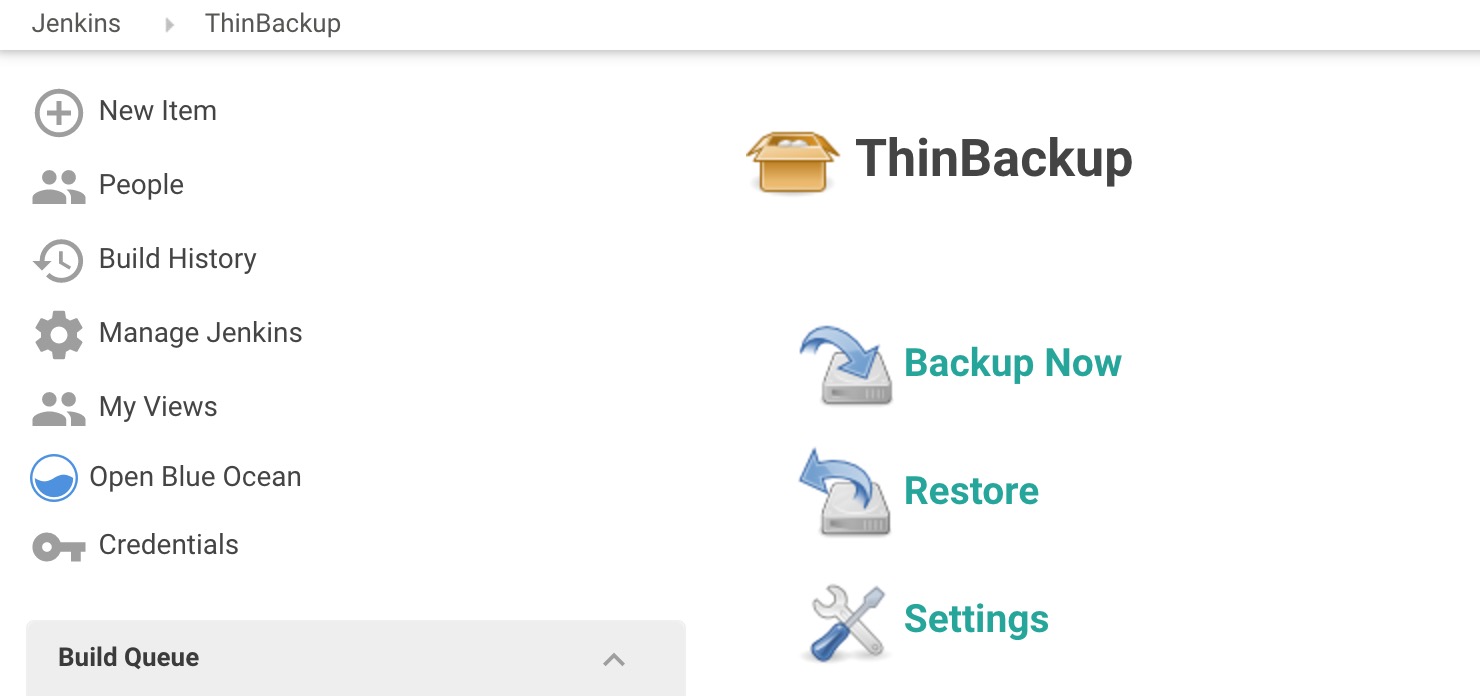
[irp posts=”1289″ name=”How to Setup Custom UI Theme For Jenkins”]

### **Backup Configuration**

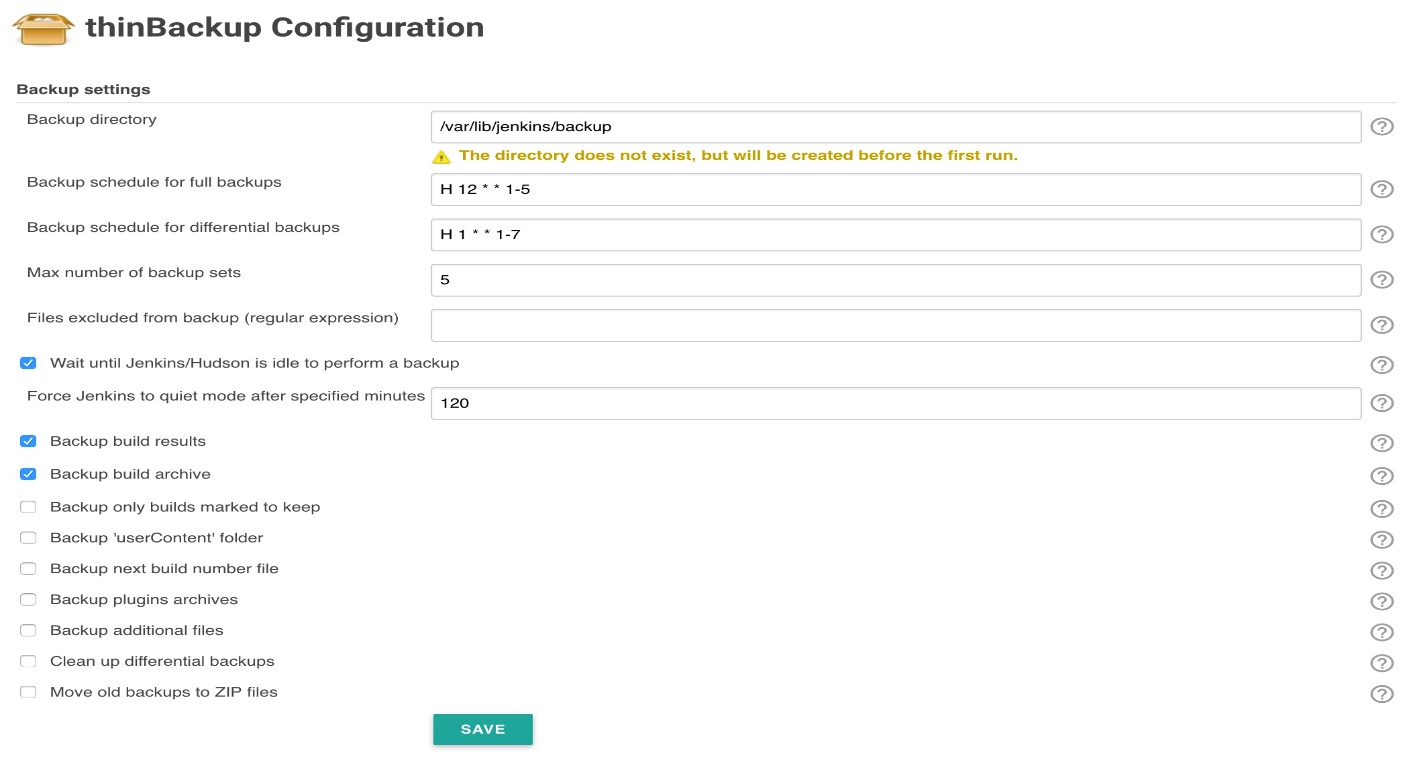
Once installed, follow the steps given below for configuring backup settings.

1. Go to Manage Jenkins — > Thin Backup

2. Click settings option.

[](https://devopscube.com/wp-content/uploads/2017/07/Screen-Shot-2017-07-31-at-7.16.30-PM.jpg)

3. Enter the backup options as shown below and save it. All the options are self-explanatory. The backup directory you specify should be writable by the user which is running the Jenkins service. All the Jenkins backup will be saved to the backup directory you specify.

[](https://devopscube.com/wp-content/uploads/2017/08/Screen-Shot-2017-08-20-at-10.51.43-AM.jpg)

4. Now, you can test if the backup is working by clicking the Backup Now option. It will create a backup of Jenkins data in the backup directory you specified in the settings.

[](https://devopscube.com/wp-content/uploads/2017/08/Screen-Shot-2017-08-20-at-11.07.44-AM.jpg)

1. If you check the backup directory in the server, you can see the backup created. For every new backup, it will attach the timestamp to the folder name and keeps the old backup based on the retention policy you mentioned in the settings.

An example is shown below.

[devopscube@jenkins backup]$ pwd

/var/lib/jenkins/backup

[devopscube@jenkins backup]$ ls

FULL-2017-08-20\_05-42 FULL-2017-08-20\_05-43 FULL-2017-08-20\_05-44

[devopscube@jenkins backup]$

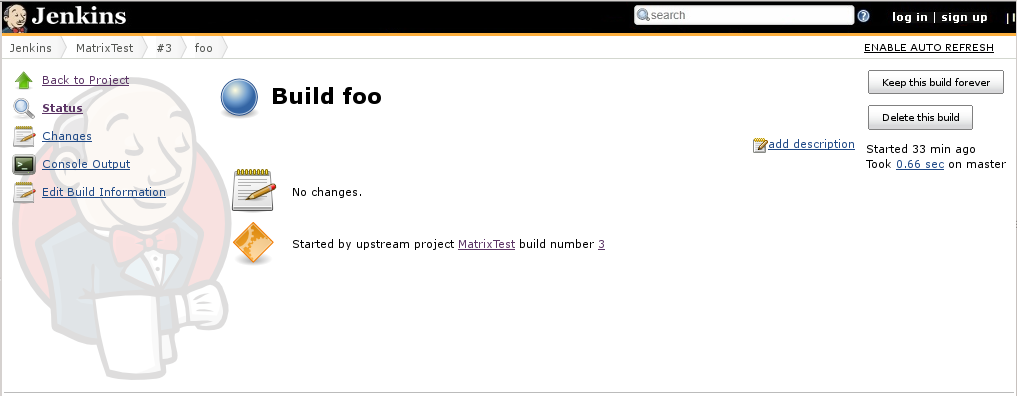
**26. What is Jenkins Home directory? Under the usr .jenkins**

**ANS:** All the settings, logs and configurations are stored in the JENKINS\_HOME directory.

27. **How to delete old builds automatically**?

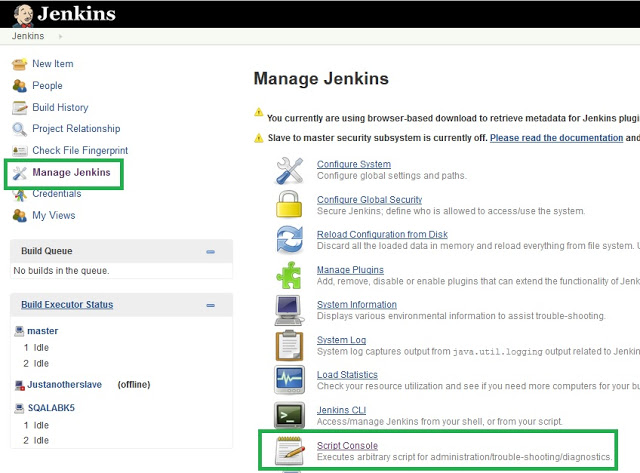
* Delete a Jenkins build via GUI.

Go into the build you want to delete, and click the **Delete this build**   button in the upper right corner.

[](https://i.stack.imgur.com/s8NQ2.png)

* If you need to clean the Jenkins build history, and reset the **build number** back to 1, you can run a simple script in Jenkins Script Console. [source](https://codeketchup.blogspot.com/2016/05/how-to-clean-and-reset-jenkins-build.html)
  1. Go to Jenkins *Script Console*.

Go to your Jenkins home page -> *Manage Jenkins* -> *Script Console*.

[](https://i.stack.imgur.com/jGU0L.jpg)

* 1. Run this script to clean and reset.

Copy and paste this script to your Console Script text area and change the "copy\_folder" to the project name that you need to clean the history. Then click the *Run* button.

def jobName = "copy\_folder"

def job = Jenkins.instance.getItem(jobName)

job.getBuilds().each { it.delete() }

job.nextBuildNumber = 1

job.save()

If you're using multibranch pipeline or folders, the second line of the above script has to be modified like this:

def jobName = "path/to/your/jenkins/pipeline/master"

def job = Jenkins.getInstance().getItemByFullName(jobName, Job.class)

**Another way to remove old builds or jobs is from the command-line.**

Login to Jenkins command-line.

To remove an entire job and build history remove the job folder, e.g.:

rm -rf jobs/jobname

To look at build history:

**$** ls jobs/jobname/builds

1409 1411 1413 1415 1417 1419 1421 1423 1425

923 963 974 985 lastFailedBuild lastSuccessfulBuild

lastUnsuccessfulBuild 1191 1334 1348 1379 1381 1383

1406 1408 1410 1412 1414 1416 1418 1420 1422 1424

913 962 973 978 987 lastStableBuild lastUnstableBuild

legacyIds

To remove selected builds then remove the build number directory, e.g.:

rm -rf jobs/jobname/builds/97\* jobs/jobname/builds/13

28. How to configure multiple environment deployment?

### **29. What are the types of pipelines in Jenkins?**

There are 3 types –

1. CI CD pipeline (Continuous Integration Continuous Delivery)
2. Scripted pipeline
3. Declarative pipeline

**30. How does CI CD work?**

**ANS:** A **CI**/**CD** Pipeline implementation, or Continuous Integration/Continuous Deployment, is the backbone of the modern DevOps environment. It bridges the gap between development and operations teams by automating the building, testing, and deployment of applications

### **31. What are the two components that you can integrate Jenkins with?**

**ANS:** According to me, the integration of Jenkins is possible with the following:

* Version Control system like GIT, SVN.
* Build tools like Apache Maven.

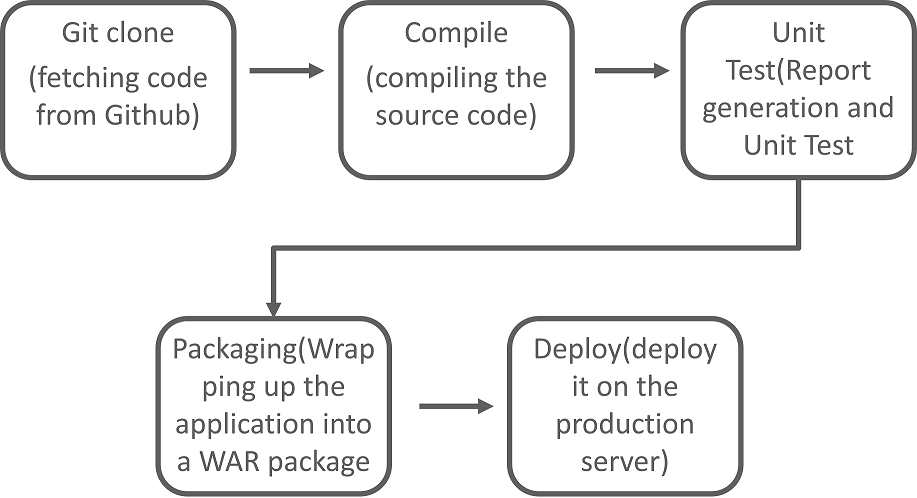
If you have anything else in your mind then mention that as well but make sure you include the above two components in your answer.

### **31. What is Maven? What is the benefit of integrating Maven with Jenkins?**

**ANS:** [Maven](https://maven.apache.org/) is a build management tool. It uses a simple pom.xml to configure all the dependencies needed to build, test and run the code. Maven manages the full lifecycle of a test project. Once integrated with Jenkins, the maven Web driver will build the project and execute all tests efficiently.

### **Q51. Can you define a Continuous Delivery Workflow?**

The flowchart below shows the Continuous Delivery Workflow. Hope it will be much easier to understand with visuals.

****