**Jenkins**

**Q #1) What is Jenkins?**

**Answer:**Jenkins is a free open-source Continuous Integration tool and automation server to monitor continuous integration and delivery. It is written in Java.

It is known as an automated Continuous Delivery tool that helps to build and test the software system with easy integration of changes to the system. Jenkins follows Groovy Scripting.

Also, it enables developers to continuously check in their code and analyse the post-build actions. The automation testers can use to run their tests as soon as the new code is added, or code is modified.

**Q #2) What are the features of Jenkins?**

**Answer: Following features of Jenkins:**

1. Free open source.
2. Easy installation on various operating systems.
3. Build Pipeline Support.
4. Workflow Plugin.
5. Test harness built around JUnit.
6. Easy upgrades.
7. Rapid release cycle.
8. Easy configuration setup.
9. Extensible with the use of third-party plugins

**Q #3) What are the advantages of Jenkins? Why we use Jenkins?**

**Answer:**Jenkins is used to continuously monitor the large code base in real-time. It enables developers to find bugs in their code and fix them. Email notifications are made to the developers regarding their check-ins as a post-build action.

**Advantages of Jenkins are as follows:**

* Build failures are cached during the integration stage.
* Notifies the developers about build report status using LDAP (Lightweight Directory Access Protocol) mail server.
* Maven release project is automated with simple steps.
* Easy bug tracking.
* Automatic changes get updated in the build report with notification.
* Supports Continuous Integration in agile development and test-driven development.

**Q #4) Mention some of the important plugins in Jenkins?**

**Answer: Plugins in Jenkins includes:**

* Gits
* Maven 2 Project
* HTML Publisher
* Copy Artcraft
* Join
* Green Balls
* Amazon EC2

**Q #5) What is Continuous Integration in Jenkins?**

**Answer:**Continuous integration is the process of continuously checking-in the developer’s code into a version control system and triggering the build to check and identify bugs in the written code.

**This is a very quick process and also gives them a chance to fix the bugs. Jenkins is one such continuous integration tool.**

In software development, multiple developers work on different software modules. While performing integration testing all the modules are being integrated together. It is considered as the development practice to integrate the code into the source repository

Whenever the programmer/developer makes any change to the current code, then it automatically  
gets integrated with the system running on the tester’s machine and makes the testing task easy and speedy for the system testers.

**Continuous Integration comprises of:**

* Development and Compilation
* Database Integration
* Unit Testing
* Production Deployment
* Code Labeling
* Functional Testing
* Generating and Analyzing Reports

**Q #6) What is the difference between Hudson and Jenkins?**

**Answer:**There is no difference between Hudson and Jenkins. Hudson was the former name of Jenkins, after going through several issues the name was changed to Jenkins.

**Q #7) What is Groovy in Jenkins?**

**Answer:**Groovy is the default scripting language that is being used in the development of JMeter Version 3.1.

Currently Apache Groovy is the dynamic object-oriented programming language that is used as a scripting language for the Java platform. Apache Groovy comes with some useful features such as Java Compatibility and Development Support.

**Q #8) Which command is used to start Jenkins?**

**Answer: You can follow the below-mentioned steps to start Jenkins:**

1. Open Command Prompt
2. From the Command Prompt browse the directory where Jenkins. war resides
3. Run the command given below:

D:\>Java –jar Jenkins.war

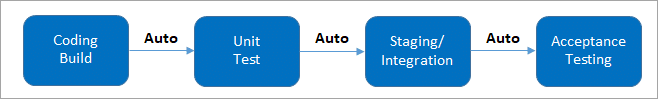
**Q #9) What is Jenkinsfile?**

**Answer:**The text file where all the definitions of pipelines are defined is called Jenkinsfile. It is being checked in the source control repository.

**Q #10) What is the difference between Continuous Integration, Continuous Delivery, and Continuous Deployment?**

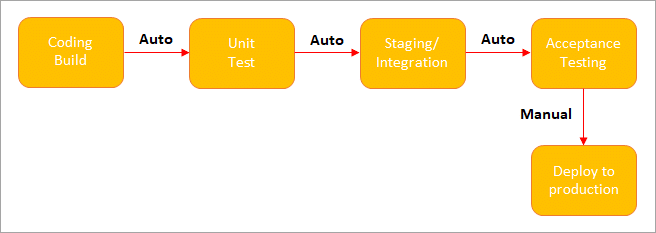
**Answer:**Differences between Continuous Integration, Continuous Delivery and Continuous Deployment

**Continuous Integration:** Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early



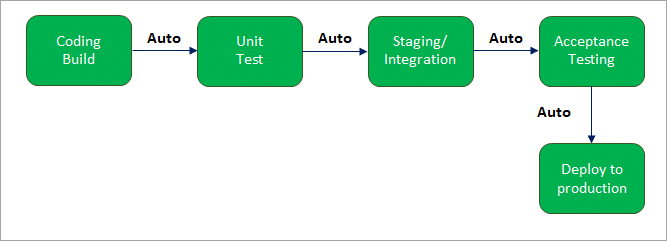
(It involves keeping the latest copy of the source code at a commonly shared hub where all the developers can check to fetch out the latest change in order to avoid conflict.)

**Continuous Delivery:** Continuous delivery (CD) is a software development practice where code changes are automatically prepared for a release to production. ... Continuous delivery lets developers automate testing beyond just unit tests so they can verify application updates across multiple dimensions before deploying to customers.



(**Manual Deployment to Production.**It does not involve every change to be deployed.)

**Continuous Deployment:** Continuous Deployment (CD) is a software release process that uses automated testing to validate if changes to a codebase are correct and stable for immediate autonomous deployment to a production environment. The software release cycle has evolved over time.



(**Automated Deployment to Production.**Involves every change to be deployed automatically.)

**Q #11) What are the differences between Continuous Integration, Continuous Delivery, and Continuous Deployment?**

|  |  |  |
| --- | --- | --- |
| **Continuous Integration** | **Continuous Delivery** | **Continuous Deployment** |
| Continuous Integration (CI) is a DevOps software development practice that permits developers to combine/merge the changes to their code in the central repository to run automated builds and tests. | Continuous Delivery (CD) refers to the building, testing, and delivering improvements to the software code. The most critical part of the CD is that the code is always in a deployable state | Continuous Deployment (CD) is the ultimate stage in the DevOps pipeline. It refers to automatic release of any developer changes from the repository to the production stage. |

**Q #12) What is Jenkins Pipeline?**

**Answer:**The pipeline can be defined as the suite of plugins supporting the implementation and integration

of continuous delivery pipelines in Jenkins. Continuous integration or continuous delivery pipeline consists of build, deploy, test, release pipeline.

The pipeline feature saves a lot of time and error in maintaining the builds. Basically, a pipeline is a group of build jobs that are chained and integrated in sequence.

The pipeline represents the continuous delivery and continuous integration of all the jobs in the SDLC and DevOps life-cycle.

**Q#13) Name the three different types of pipelines in Jenkins?**

**Answer: T**he three different types of pipelines in Jenkins are:

* CI/CD pipeline
* Scripted pipeline
* Declarative pipeline

**Q #14) What is a CI CD pipeline?**

**Answer:** Continuous integration or continuous delivery pipeline consists of build, deploy, test, release pipeline. The pipeline feature saves a lot of time and error in maintaining the builds. Basically, a pipeline is a group of build jobs that are chained and integrated in sequence.

**Q #15) What are Scripted Pipelines in Jenkins?**

**Answer: Scripted Pipeline follows Groovy Syntax as given below:**

Node {

           }

In the above syntax, the**node** is a part of the Jenkins distributed mode architecture, where there are two types of node, **Master** which handle all the tasks in the development environment and the**Agent**is being used to handle multiple tasks individually.

**Q #16) What are Declarative Pipelines in Jenkins?**

**Answer:**Declarative Pipelines are the newest additions to Jenkins that simplify the groovy syntax of Jenkins pipelines (top-level pipeline) with some exc

eptions, such as:

No semicolon to be used as a statement separator. The top-level pipeline should be enclosed within block viz;

**The common syntax is:**

pipeline {

/\* Declarative Pipeline \*/

}

Blocks must contain Sections, Directives, steps or assignments.

pipeline {

agent any

stages {

stage(‘Build’) {

steps {

// Statements…

}

}

stage (‘Test’) {

steps {

// Statements…

}

}

}

}

**The above code has 3 major elements**

* **Pipeline:**The block of script contents.
* **Agent:** Defines where the pipeline will start running from.
* **Stage:** The pipelines contain several steps enclosed in the block called Stage.

Table

Description automatically generated

**Continuous** Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day

which manual command is used to restart jenkins without waiting for build to complete

(Jenkins\_url)/restart

You can use any one of the following commands to start Jenkins manually: **(Jenkins\_url)/restart**:

Forces a restart without waiting for builds to complete. (Jenkin\_url)/safeRestart: Allows all running builds to complete.