## MODULE-IV

Continuous Integration with 3enkins

Continuous Integration (CI) is a DevOps software development practice that enables the developers to merge their code changes in the central repository to run automated builds and tests. It refers to the process of automating the integration of code changes coming from several sources.

## Features of Continuous Integration

Following are some of the main features or practices for Continuous Integration.

1. Maintain a single source repository — All source code is maintained in a single repository. This avoids having source code being scattered across multiple locations. Tools such as Subversion and Git are the most popular tools for maintaining source code.
2. Automate the build — The build of the software should be carried out in such a way that it can be automated. If there are multiple steps that need to be carried out, then the build tool needs to be capable of doing this. For .Net, MSBuild is the default build tool and for Java based applications you have tools such as Maven and Grunt.
3. Make your build self-testing — The build should be testable. Directly after the build occurs, test cases should be run to ensure that testing can be carried out for the various functionality of the software.
4. Every commit should build on an integration machine — The integration machine is the build server and it should be ensured that the build runs on this machine. This means that all dependent components should exist on the Continuous Integration server.
5. Keep the build fast — The build should happen in minutes. The build should not take hours to happen, because this would mean the build steps are not properly configured.
6. Everyone can see what is happening — The entire process of build and testing

and deployment should be visible to all.

1. Automate deployment — Continuous Integration leads to Continuous deployment. It is absolutely necessary to ensure that the build should be easy to deploy onto the production environment.

What Does Build Mean?

The term build may refer to the process by which source code is converted into a stand- alone form that can be run on a computer or to the form itself. One of the most important steps of a software build is the compilation process, where source code files are converted into executable code. The process of building software is usually managed by a build tool. Builds are created when a certain point in development has been reached or the code has been deemed ready for implementation, either for testing or outright release.

A build is also known as a software build or code build.

Build automation is the process of automating the retrieval of source code, compiling it into binary code, executing automated tests, and publishing it into a shared, centralized repository.

# Need /Importance of Continuous Integration

### Reduces Risk

The frequent testing and deployment of code reduce the project's risk level, as now the code defects and bugs can be detected earlier. This states that these bugs and errors can be easily fixed and take less time, making the overall process cheaper. The general working speeds up the feedback mechanism that makes the communication smoother and effective.

### Better Communication

The Continuous Integration process collaborates with the Continuous Delivery workflow that makes code sharing easy and regularized. This makes the process more transparent and collaborative among team members. In the long term, this makes the communication speed more efficient and makes sure that everyone in the organization is on the same page.

### Higher Product Quality

Continuous Integration provides features like Code review and Code quality detection, making the identification of errors easy. If the code does not match the standard level or a mistake, it will be alerted with emails or SMS messages. Code review helps the developers to improve their programming skills continually.

### Reduced Waiting Time

The time between the application development, integration, testing, and deployment is considerably reduced. When this time is reduced, it, in turn, reduces the waiting time that may occur in the middle. CI makes sure that all these processes continue to happen no matter what.

# What is 3enkins

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes and used to build and test software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows to continuously deliver software by integrating with a large number of testing and deployment technologies.

# History of 3enkins

Kohsuke Kawaguchi, who is a Java developer, working at SUN Microsystems, was tired of building the code and fixing errors repetitively. In 2004, he created an automation server called Hudson that automates build and test task.

In 2011, Oracle who owned Sun Microsystems had a dispute with Hudson open source

community, so they forked Hudson and renamed it as Jenkins.

Both Hudson and Jenkins continued to operate independently. But in short span of time, Jenkins acquired a lot of contributors and projects while Hudson remained with only 32 projects. Then with time, Jenkins became more popular, and Hudson is not maintained anymore.



Only two pages were converted.

Please **Sign Up** to convert the full document.

[**www.freepdfconvert.com/membership**](https://www.freepdfconvert.com/membership)