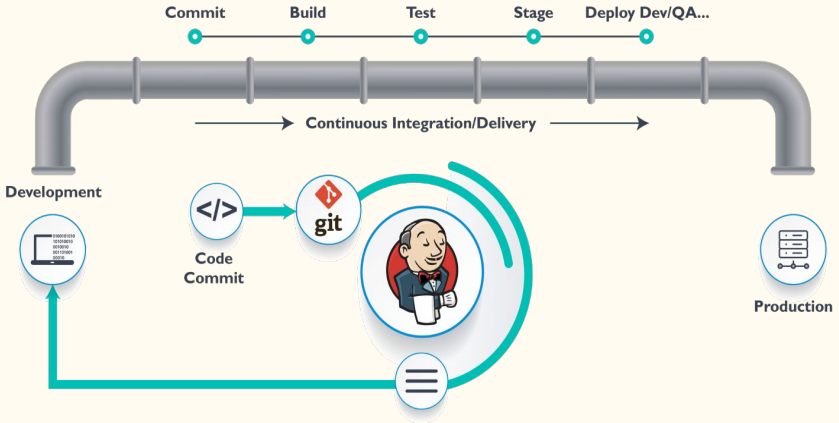
JENKINS

WHAT IS JENKINS

Jenkins is an open-source Continuous Integration server written in Java for orchestrating a chain of actions to achieve the Continuous Integration process in an automated fashion. Jenkins supports the complete development life cycle of software from building, testing, documenting the software, deploying, and other stages of the software development life cycle.

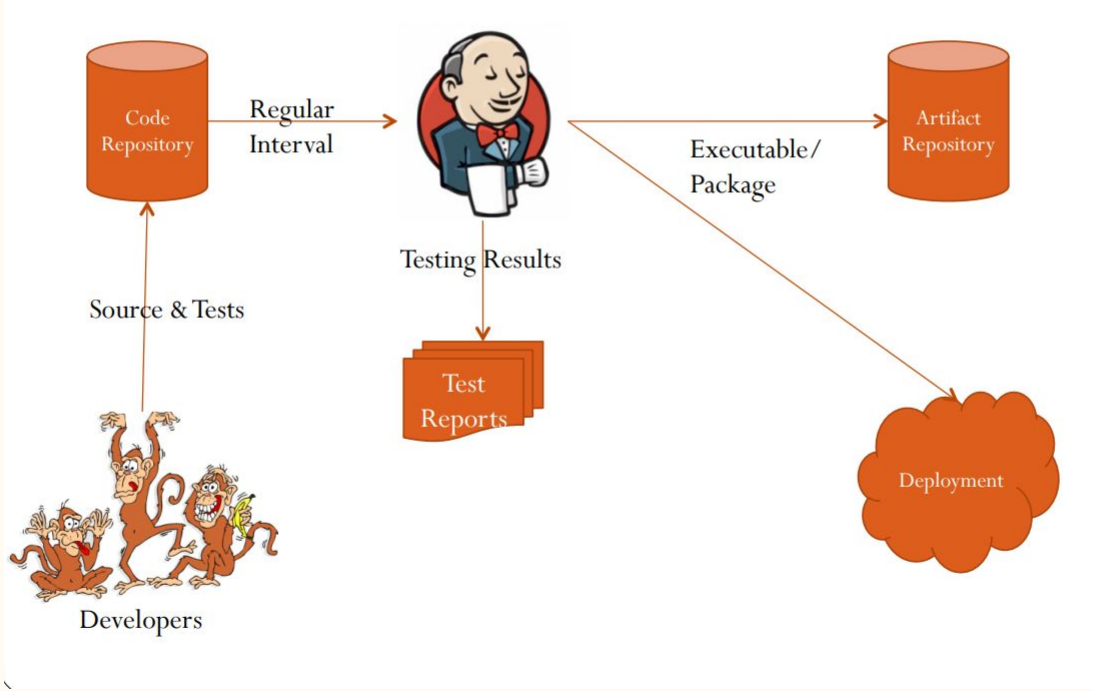
CI – What does it really mean?



At a regular frequency (ideally at every commit), the system is:

- **Integrated** - All changes up until that point are combined into the project.
- **Built** - The code is compiled into an executable or package.
- **Tested** - Automated test suites are run.
- **Archived** - Versioned and stored so it can be distributed as is, if desired.
- **Deployed** - Loaded onto a system where the developers can interact with it.

Where the jenkins fits in ?



Why Jenkins? Flexibility!

- Jenkins is a highly configurable system by itself
- The additional community developed plugins provide even more flexibility
- By combining Jenkins with Ant, Gradle, or other Build
- Automation tools, the possibilities are limitless

What can Jenkins do?

- Generate test reports
- Integrate with many different Version Control Systems
- Push to various artifact repositories
- Deploys directly to production or test environments
- Notify stakeholders of build status

How Jenkins works ?

When setting up a project in Jenkins, out of the box you have the following general options:

- Associating with a version control server
- Triggering builds
- Polling, Periodic, Building based on other projects
- Execution of shell scripts, bash scripts, Ant targets, and Maven targets
- Artifact archival
- Publish JUnit test results and Javadocs
- Email notifications
- As stated earlier, plugins expand the functionality even further

How Jenkins works ?

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

- Building
  - Jenkins executes the build in an executor

    By default, Jenkins gives one executor per core on the build server
- Jenkins also has the concept of slave build servers

  Useful for building on different architectures

  Distribution of load

How Jenkins works - Reporting

- Jenkins comes with basic reporting features
- Keeping track of build status

  Last success and failure

  “Weather” – Build trend

- These can be greatly enhanced with the use of pre-build plugins

  Unit test coverage

  Test result trending

  Findbugs, Checkstyle, PMD

Conclusion

- Continuous integration is a necessity on complex projects due to the benefits it provides regarding early detection of Problems.

- A good continuous build system should be flexible enough to fit into pre-existing development environments and provide all the features a team expects from such a system

- Jenkins, a continuous build system, can be an integral part of any continuous integration system due to it’s core feature set and extensibility through a plugin system