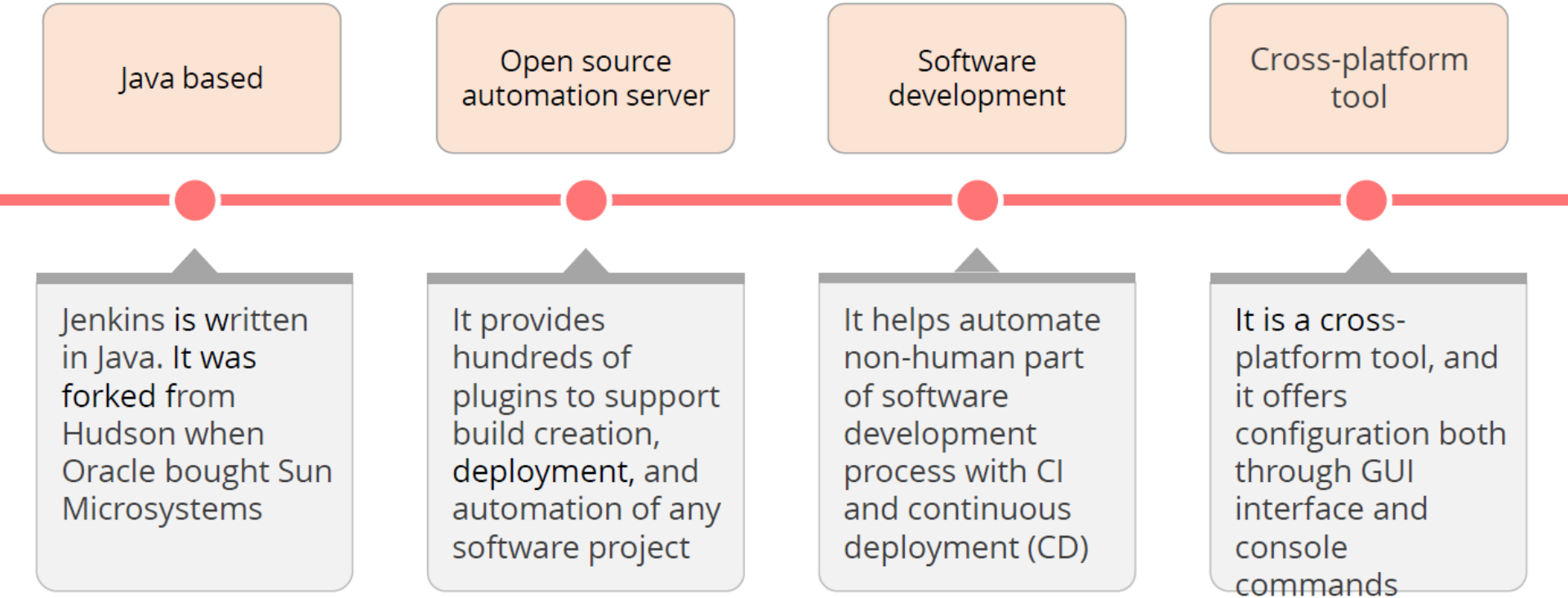
Chapter 3 – CI-CD with Jenkins

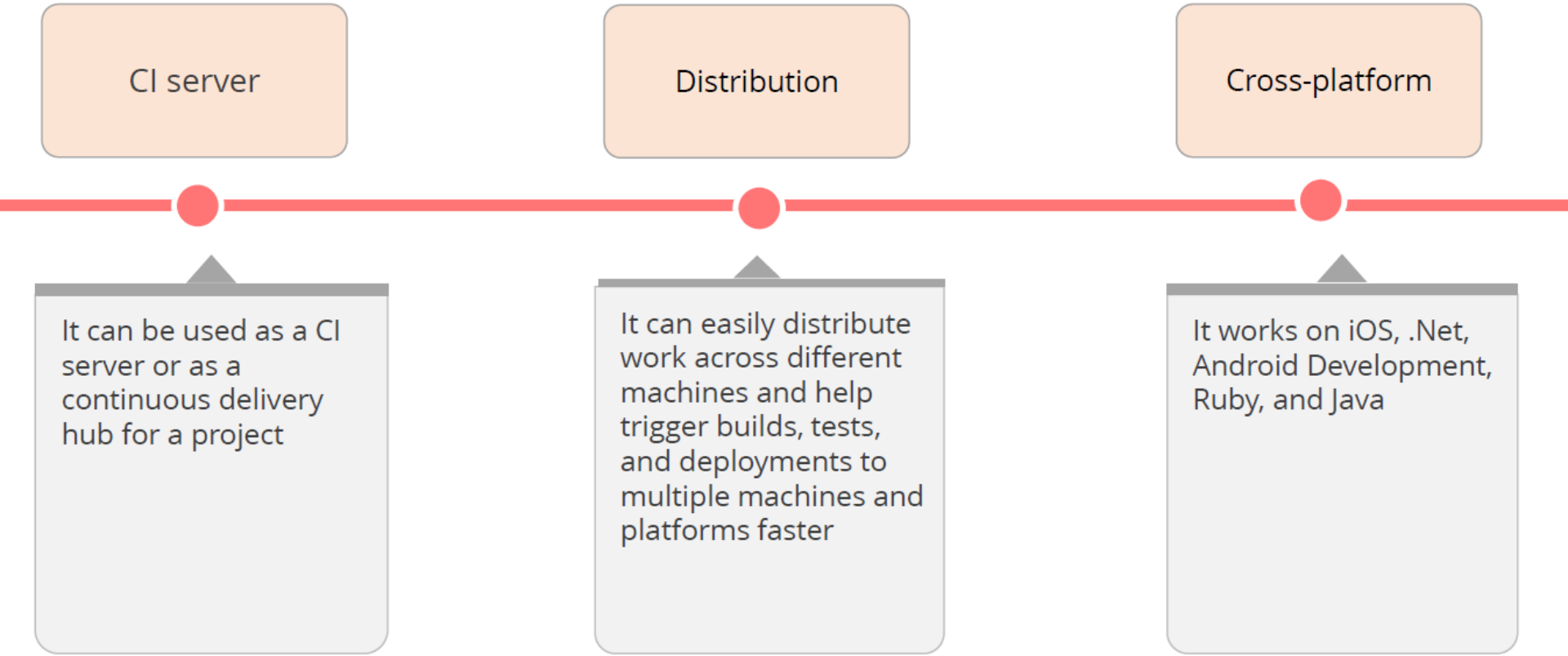
Learning Topics

- Overview of Jenkins
- Jenkins Jobs
- Overview of Continuous Integration
- CI with Jenkins
- CI-CD with Jenkins

Jenkins


Java based
Open source automation server
Software development
Cross-platform tool
Jenkins is written in Java. It was forked from Hudson when Oracle bought Sun Microsystems
It provides hundreds of plugins to support build creation, deployment, and automation of any software project
It helps automate non-human part of software development process with CI and continuous deployment (CD)
It is a cross-platform tool, and it offers configuration both through GUI interface and console commands

Jenkins Cont.



Jenkins Lab – Pre-requisites

- Create Users in Jenkins
- Login to Jenkins Console
- Please Don’t Delete Anything
- Please Don’t Modify System Configurations

Jenkins Lab 1 – First Job – Free Style

Jenkins Lab 2 – Repeatable Job

Jenkins Lab 3 – Parameterized Job

Jenkins Lab 4 – Email Notifications

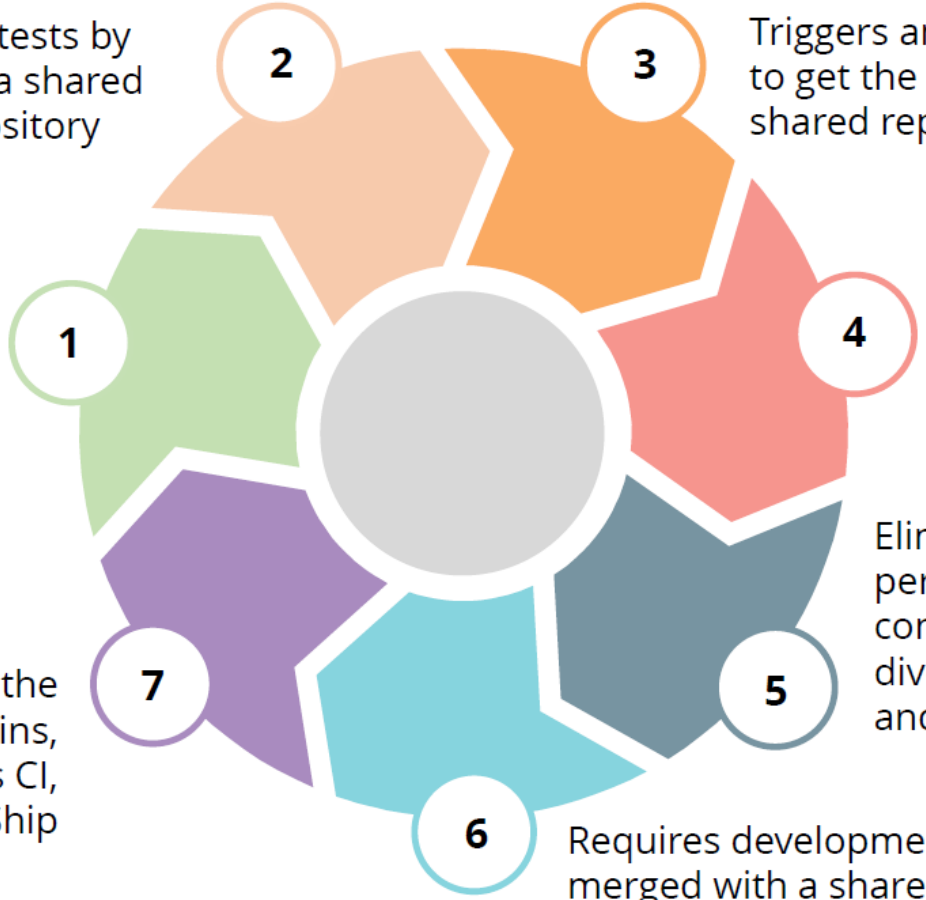
Integration Hell

- Developers takes a copy of the code base to work on
- Other developers submit changes when work is being done
- The repository can become very different to the developer's code base
- It can require more work to integrate changes than to do the development!
  - This is known as integration hell
- Worst case is that the developer has to redo work on latest code base

Continuous Integration (CI)

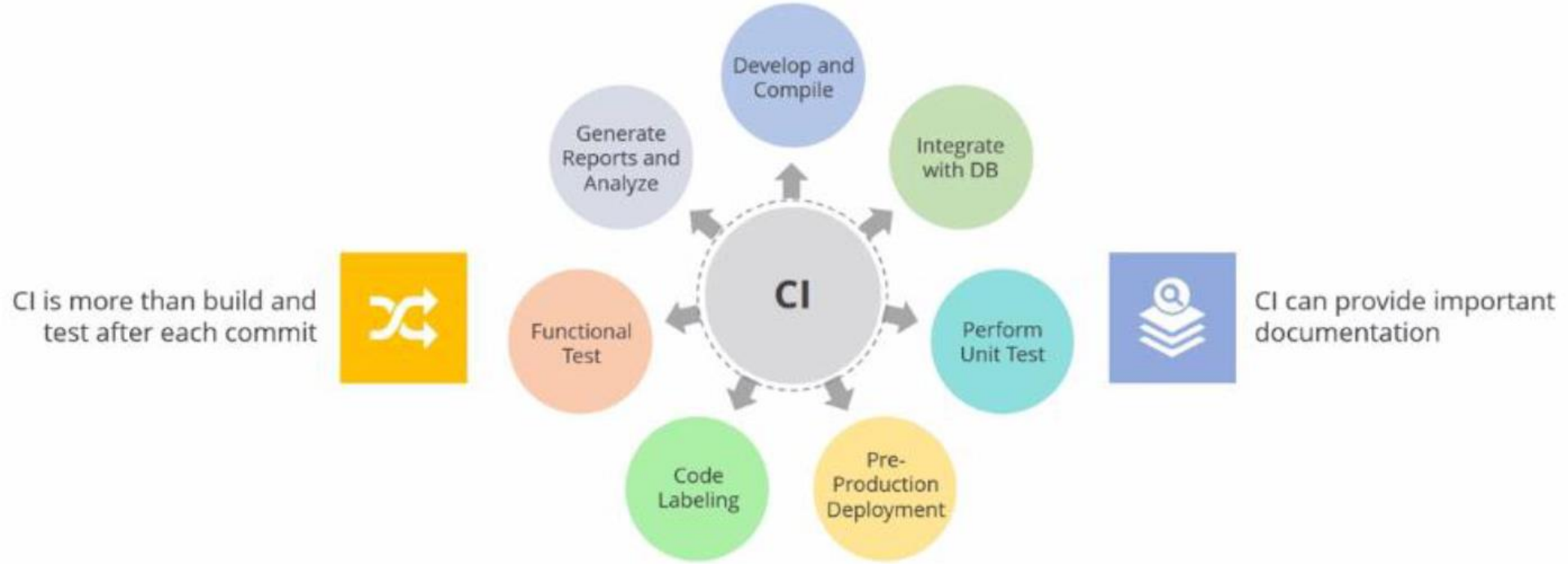
- Continuous integration (CI) solves these issues
- Integrate changes early and often
- Need to detect and correct errors quickly
- Automation is the best practice

Continuous Integration



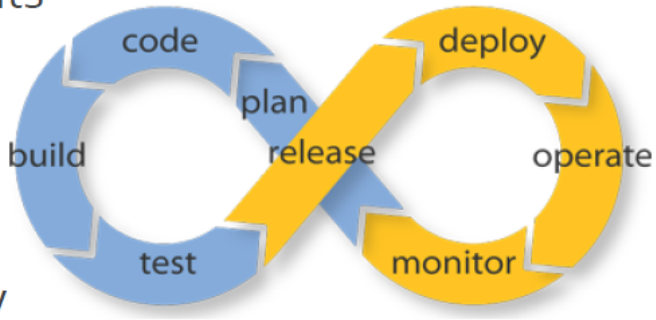
1. Process of automating the build and testing of code
2. Shares code and unit tests by merging changes into a shared version control repository
3. Triggers an automated build system to get the latest code from the shared repository
4. Integrates the changes made by software developers with the team's code base
5. Eliminates long waiting periods to minimize merge conflicts, hard to fix bugs, diverging code strategies, and duplicated efforts
6. Requires development team's code to be merged with a shared version control branch continuously to avoid problems
7. Popular CI tools in the market are Jenkins, TeamCity, Travis CI, Bamboo, and CodeShip

Continuous Integration Cont.

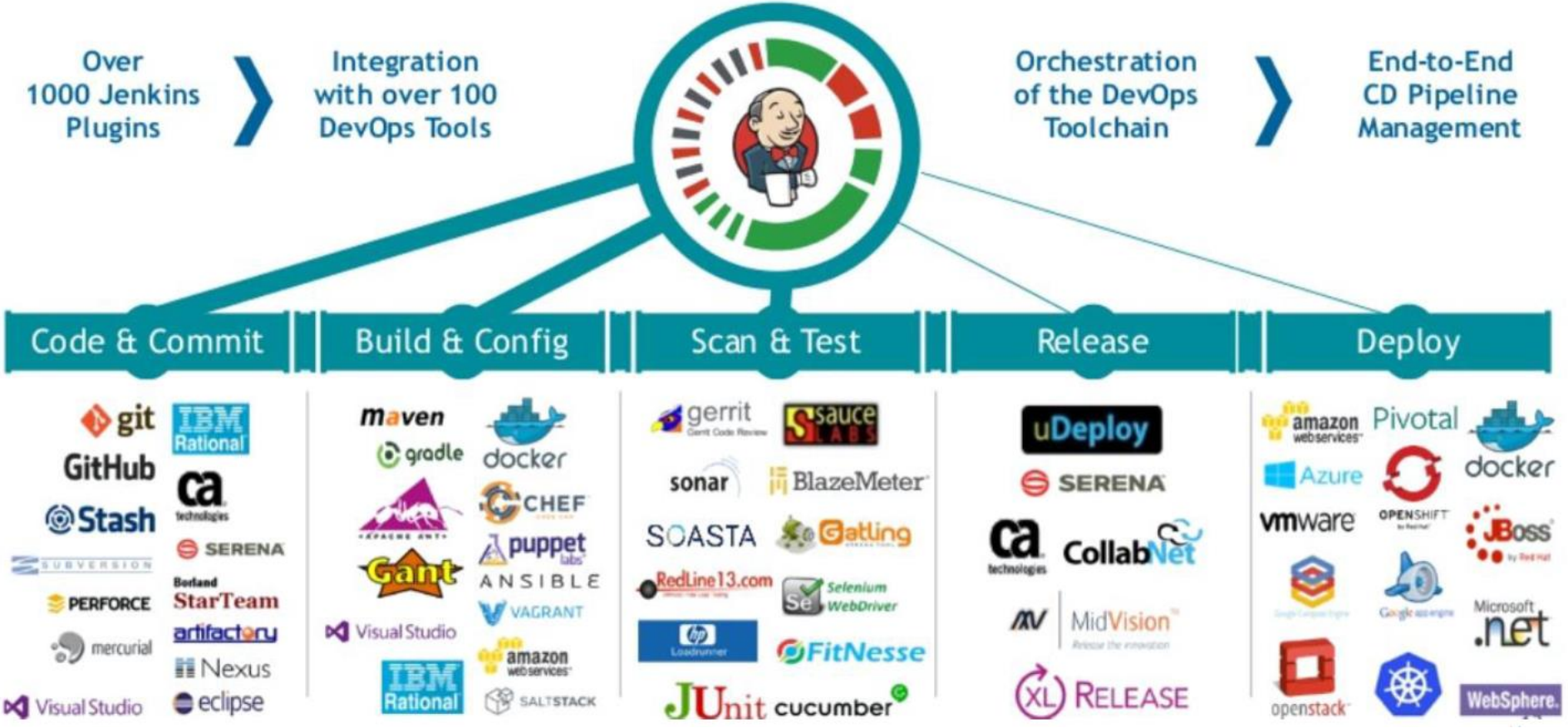


Continuous Integration Tools

- Build the code from the repository soon after every commit
- The version control system must support atomic commits
- All tests should be run after a successful build
- Build and tests need to run quickly
- Results of build and test must be available to everybody
- It should be easy to detect problems and find who committed the change
  - A Web interface
  - Sends emails to interested parties



CI Using Jenkins



CI Using Jenkins

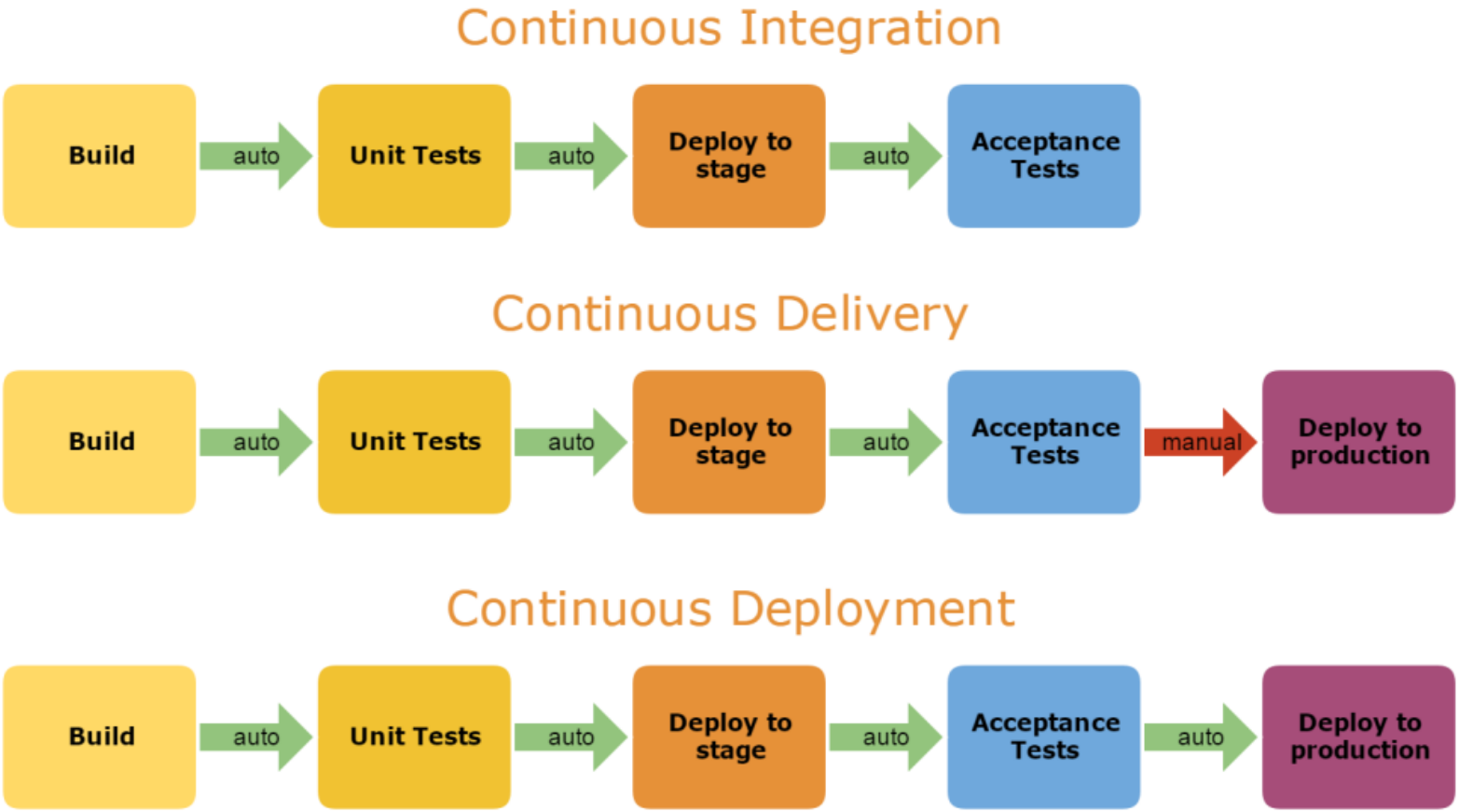
- Builds can be triggered from a commit in version control system or scheduling a cron job or by other builds in the queue
- Support version control systems like CVS, Subversion, Git, Perforce, Clearcase
- Can be integrated with bug tracking databases Jira, Bugzilla, Sonar Quality Gate
- Integrates with testing tools like Nunit, Junit, TestLink, Celenium Capability Axis, qTest, QMetry for Jira, Sonar
- Integrates with build tools like NAnt, EasyAnt, Ansible, Ant, Maven, Gradle, Visual Studio Code Metrics, SaltStack, Python, Ruby, Shell and Windows commands
- Integrates with config tools like Chef, Puppet, Ansible, Vagrant, IBM Rational, SaltStack
- Has plugins for Puppet Enterprise Pipeline, Ansible, OctopusDeploy, Docker Pipeline, Google Deployment Manager, Amazon Web Services, VMWare, Azure, Microsoft .Net, OpenStack

Jenkins Lab 5 – CI & Webhook

Group Assignment – CI with GitHub

1. Team of 4-5
2. Team Leader and Team Members
3. Team Leader to create repo
4. Add Team Members as Collaborators
5. Team Members to clone repo
6. Create their own branch (use your name)
7. Create New File
8. Push to GitHub
9. CI with Jenkins
10. Email Notifications for All

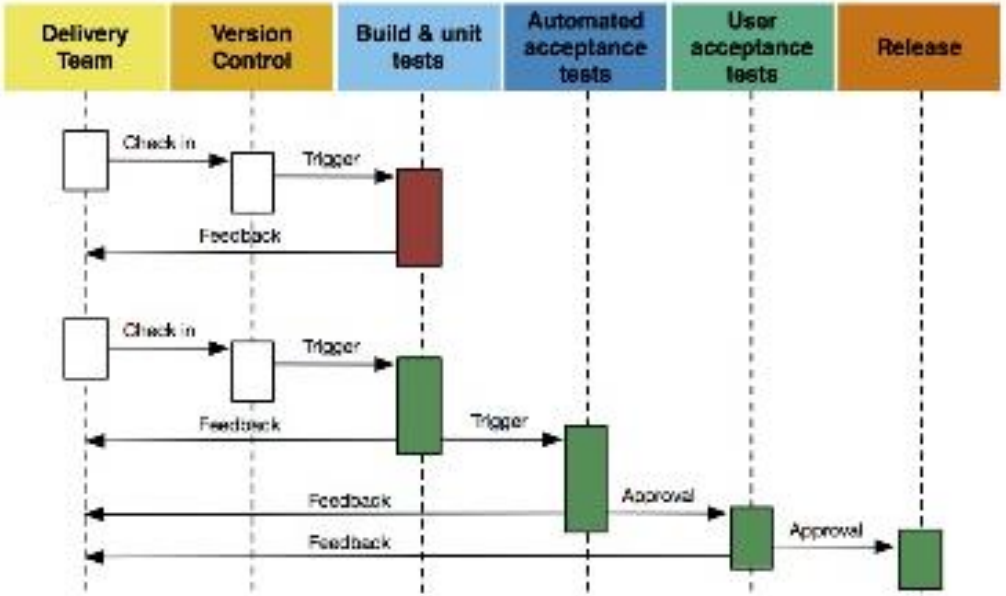
CI-CD-Continuous Deployment



Jenkins Lab 6 – Continuous Deployment

Jenkins Pipelines

- Every check-in triggers pipeline execution
- Feedback to the team in every stage
  - *"Bring the pain forward"*
  - *"Fail fast, fail often"*
- Minimize execution time
- Always aware of latest *stable* release



Jenkins Lab 7 – Pipelines

Individual Assignment

1. Create pipeline for all your jobs (5 jobs)
2. Trigger first job from GitHub

This concludes Chapter 3

Let us move to Chapter 4

Containerization with Docker