



Jenkins Administration (DO-JNK)

Keywords

DevOps, Continuous Integration,
Continuous Delivery, Continuous Deployment,
Jenkins, Plugins, Pipeline, Jenkinsfile

References



Jenkins User Documentation

<https://jenkins.io/doc/>



DevOps

DevOps

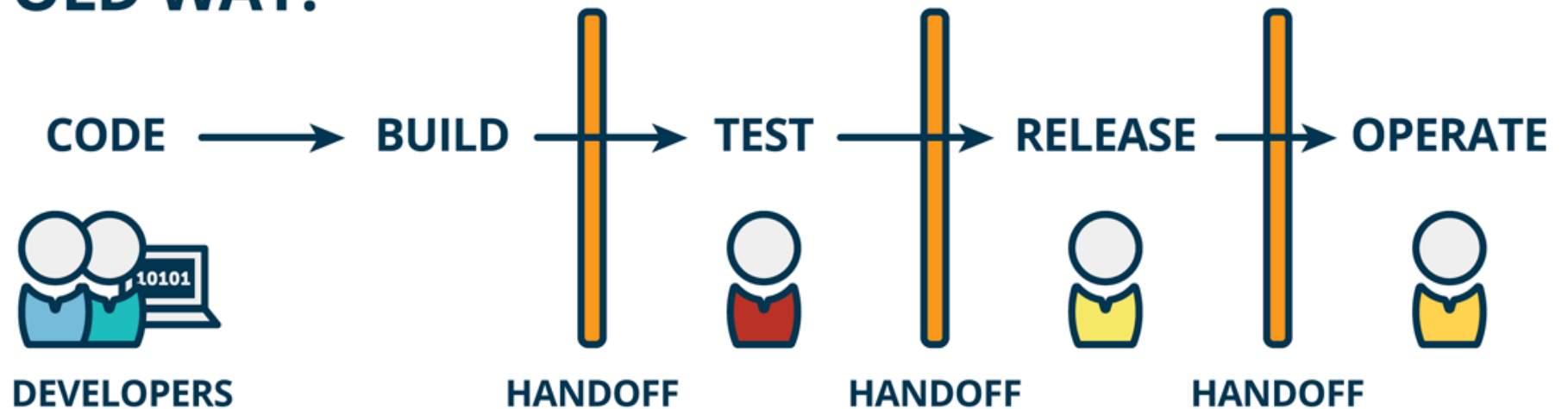
- The word 'DevOps' comes from the combination of the words 'development' and 'operations'.
- DevOps is a culture that promotes collaboration between developers and other technology professionals, often called operations or just ops.
- Specifically, communication and collaboration during the software delivery and deployment process, with the goal of releasing better quality software more quickly and more reliably.




CI/CD

Software Development

OLD WAY:

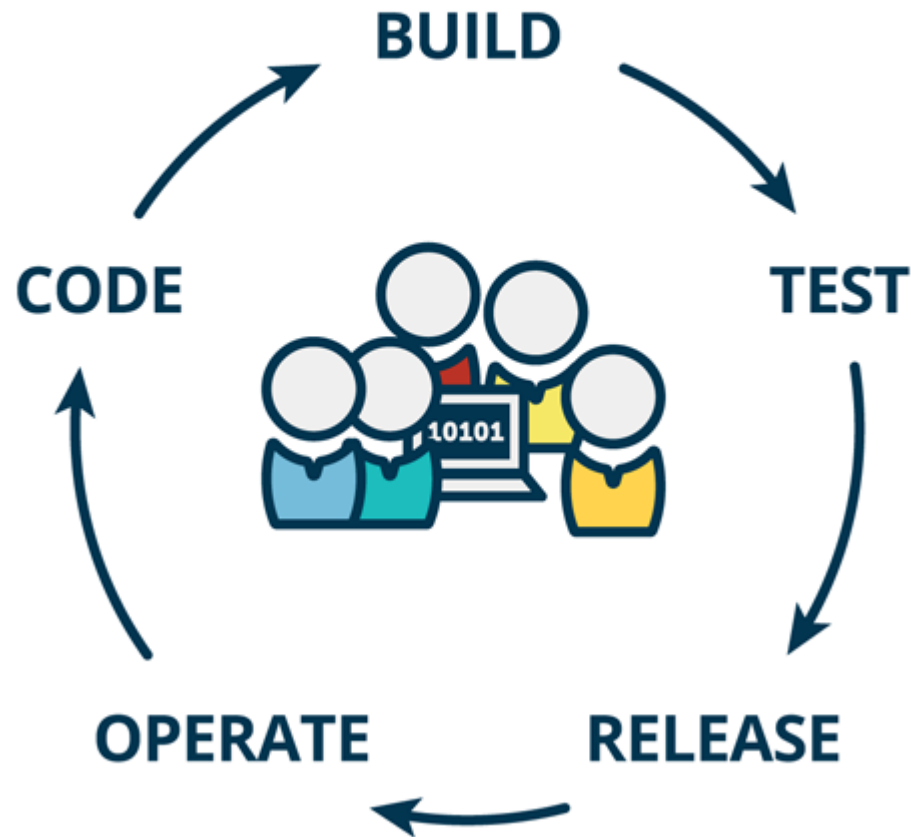


Old Ways of Software Development

- Code changes made by individual team members are merged together into working software, which was known as Integration phase
 - Integration phase was a hard work which often result in code conflicts, hard to find bugs and even harder to fix them which lead to significant delivery delays
- 

Type Software Development (2)

NEW WAY:



New Way of Software Development

- **Continuous Integration**

is a development practice that requires developers to integrate code into a shared repository several times a day.

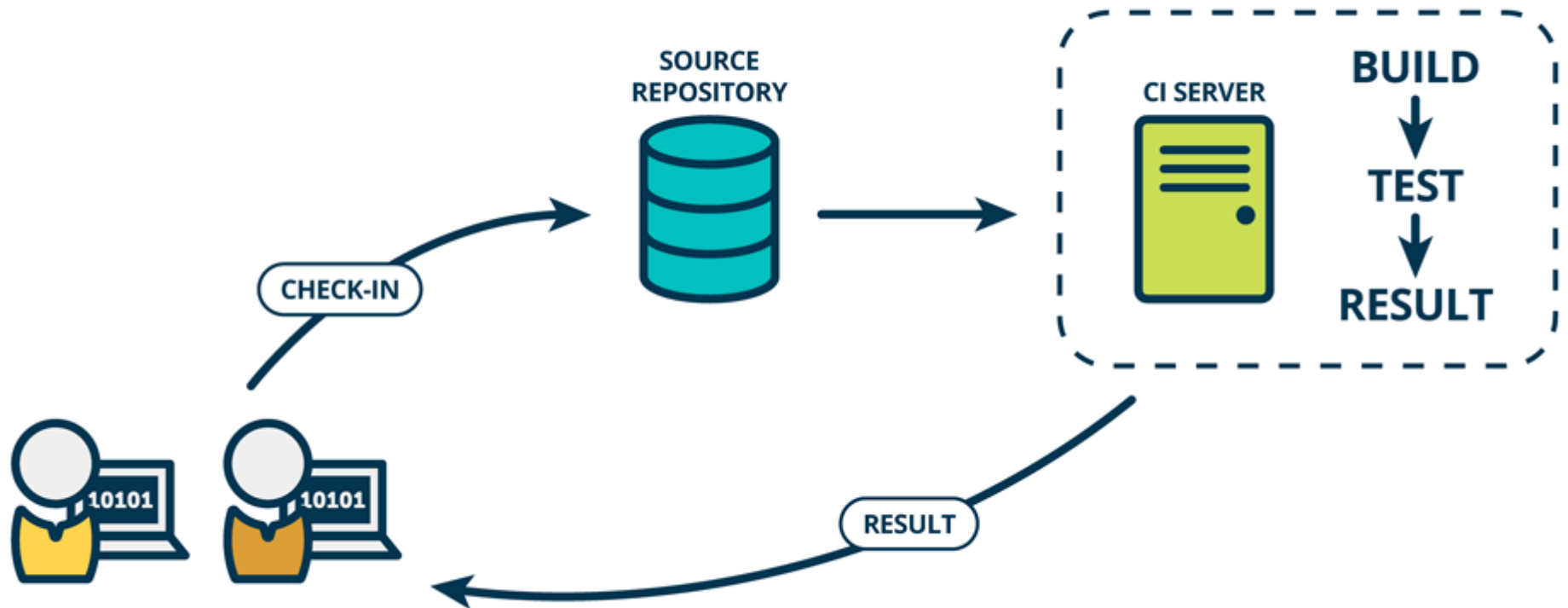
- **Continuous Delivery**

Continual delivery of code to an environment once the code is ready to ship. This could be staging or production. The idea is the product is delivered to a user base, which can be QAs or customers for review and inspection.

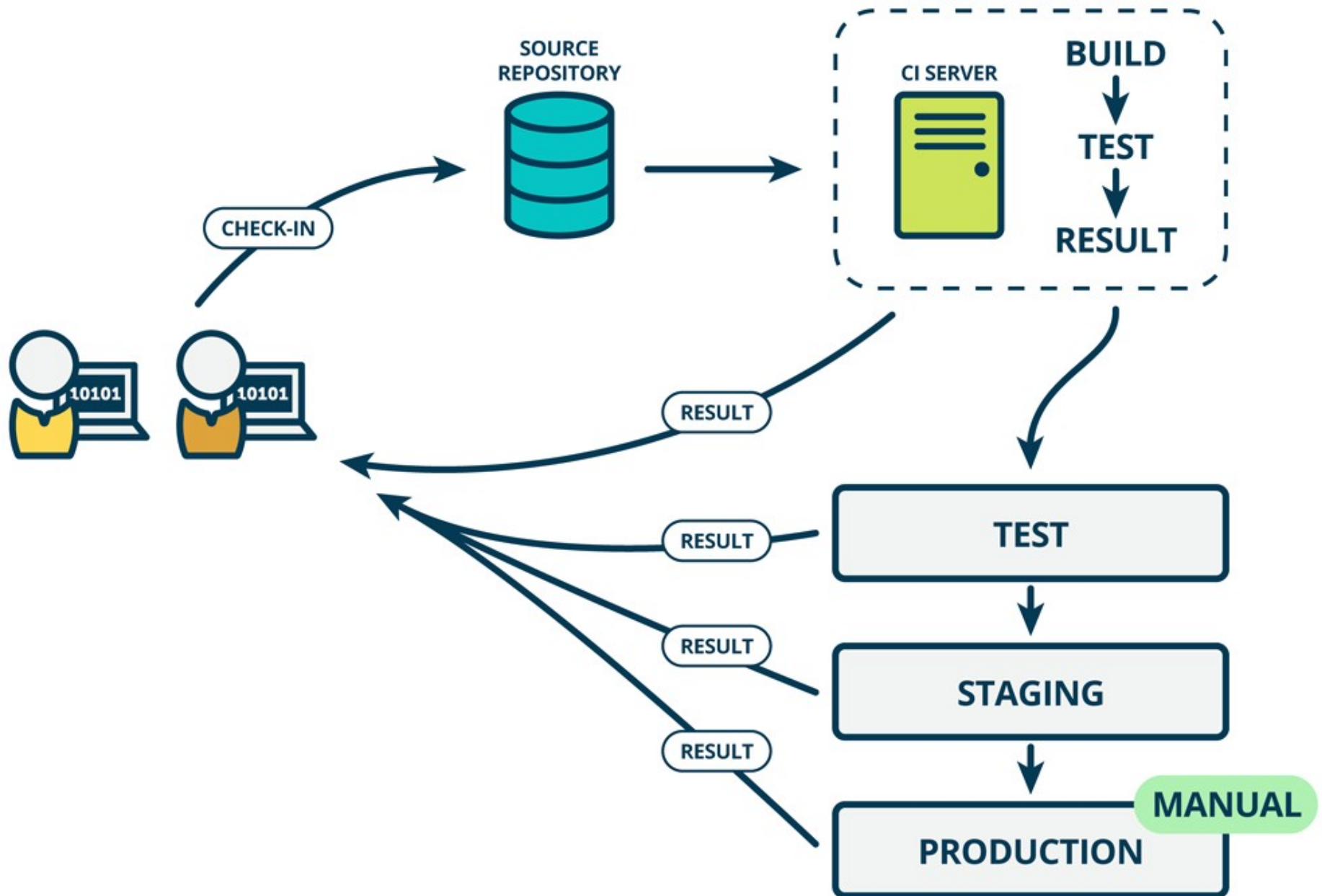
- **Continuous Deployment**

The deployment or release of code to production as soon as it is ready

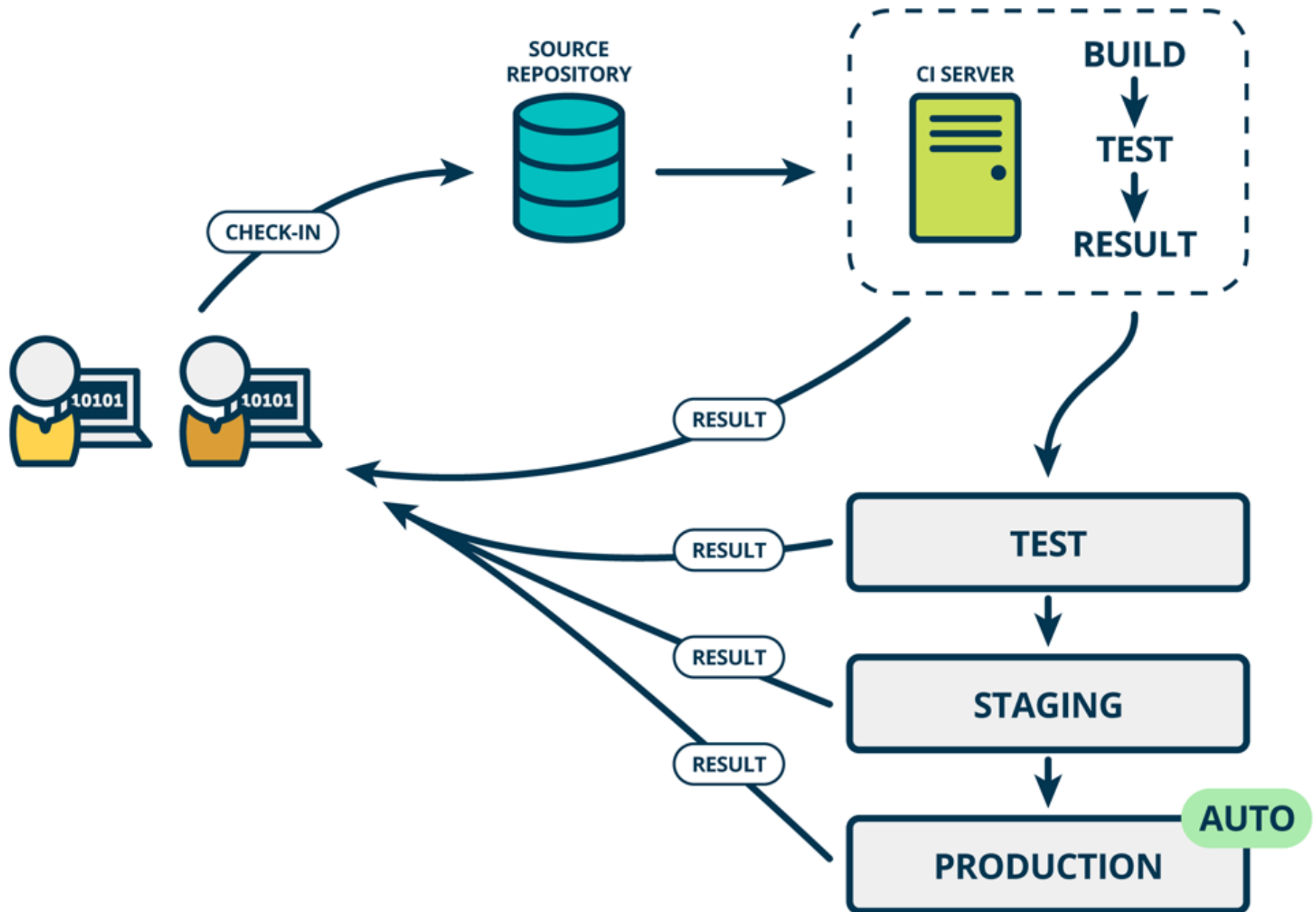
Continuous Integration (CI)



Continuous Delivery (CD)



Continuous Deployment (CD)






Jenkins

Jenkins Overview – The Ultimate CI Tool

- Jenkins is a continuous integration and build server.
- It is used to manually, periodically, or automatically build software development projects.
- It is an open source Continuous Integration tool written in Java.
- Jenkins is used by teams of all different sizes, for projects with various languages.

Why Jenkins is popular

- **Easy to use**
 - **Great extensibility**
 - Support different version control systems
 - Code quality metrics
 - Build notifiers
 - UI customization
- 

The Story of Jenkins

Hudson



Jenkins

- Hudson was started in 2004 at Sun by Kohsuke Kawaguchi as a hobby project.
- First release in 2005.
- Kohsuke worked on Hudson full time in early 2008.
- Became the leading Continuous Integration solution with a market share of over 70% in 2010.
- Renamed to Jenkins in 2011.

Jenkins Release

- The Jenkins project produces two release lines, LTS and weekly. Depending on your organization's needs, one may be preferred over the other.
- LTS (Long-Term Support) releases are chosen every 12 weeks from the stream of regular releases as the stable release for that time period.
- Weekly A new release is produced weekly to deliver bug fixes and features to users and plugin developers.
- Both release lines are distributed as .war files, native packages, installers, and Docker containers. Packages with the gear icon are maintained by third parties.

Companies Supporting Jenkins Project



Microsoft



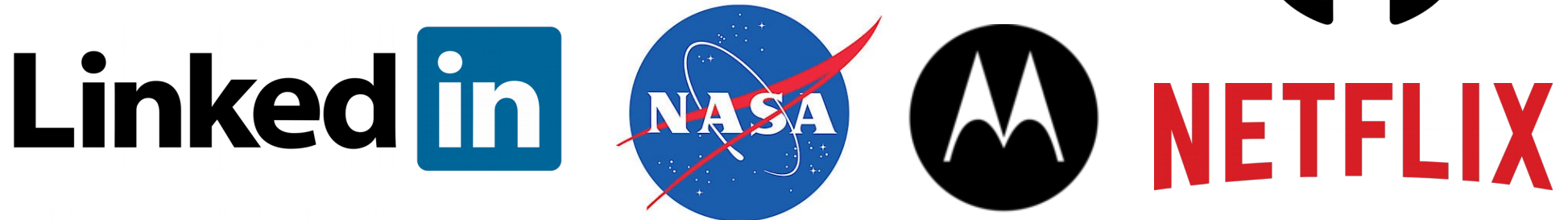
cloudbees.



redhat.

Companies Using Jenkins

Alcatel-Lucent  cloudbees  cloudera



<https://wiki.jenkins.io/pages/viewpage.action?pageId=58001258>

Prerequisites

- Minimum hardware requirements:
 - 256 MB of RAM
 - 1 GB of drive space(although 10 GB is a recommended minimum if running Jenkins as a Docker container)
- Recommended hardware configuration for a small team:
 - 1 GB+ of RAM
 - 50 GB+ of drive space
- Software requirements:
 - Java 8 - either a Java Runtime Environment (JRE) or a Java Development Kit (JDK) is fine

Jenkins Plugins



Plugins Index

Discover the 1000+ community contributed Jenkins plugins to support building, deploying and automating any project.

Browse ▾

Find plugins...



Sort relevance

- ☒ Relevance
- ☐ Most installed
- ☐ Trending
- ☐ Title
- ☐ Release date

Categories

☐ Platforms

☐ iOS development

☐ .NET

☐ Android development

☐ Ruby development

☐ User interface

☐ User Interface

☐ List view column plugins

☐ Administration

☐ Agent controllers

☐ Page decorators

☐ Users and security

☐ Cluster management

☐ CLI extensions

☐ Source code management

☐ SCM connections

☐ SCM related

☐ Build management

☐ Build triggers

☐ Build wrappers

☐ Build notifiers

☐ Deployment plugins

☐ Build parameters

☐ Clean-up actions

☐ Build tools

☐ Build reports

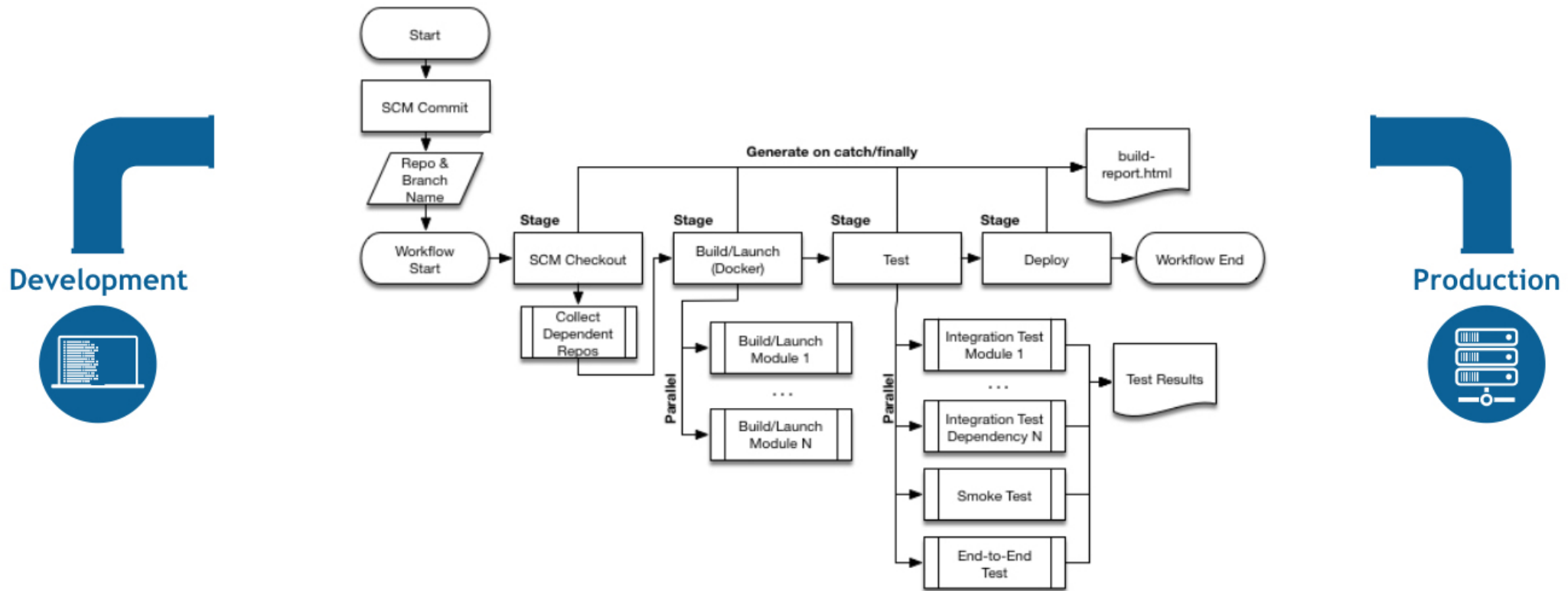
☐ Artifact uploaders



Lab I

Pipeline

Jenkins Pipeline (or simply "Pipeline" with a capital "P") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.





Pipeline (2)

- Series of steps which orchestrates work required to Build, Test & Deploy applications
- They are defined in the file called Jenkinsfile which is stored in the root of the project repository
- **Requirements**
 - 1) Jenkins 2.0 & Above
 - 2) Maven, git installed in Jenkins global tools configuration section

Pipeline as a Code

```
1 pipeline {  
2   agent any  
3   tools {  
4     maven 'maven'  
5     jdk 'jdk'  
6   }  
7   stages {  
8     stage('Build') {  
9       steps {  
10         sh 'mvn clean package'  
11       }  
12     }  
13     stage('Test') {  
14       steps {  
15         sh 'mvn test'  
16       }  
17       post {  
18         always {  
19           junit 'target/surefire-reports/*.xml'  
20         }  
21       }  
22     }  
23   }  
24 }  
25 }
```



Lab I



www.btech.id