DevOps

Caltech | Center for Technology & Management Education

Post Graduate Program in DevOps

simplilearn

CI/CD Pipeline with Jenkins

Learning Objectives

By the end of this lesson, you will be able to:

- Set up Jenkins
- Integrate Jenkins with Git, Maven, and Ant
- Illustrate Jenkins builds
- Execute remote jobs using Jenkins

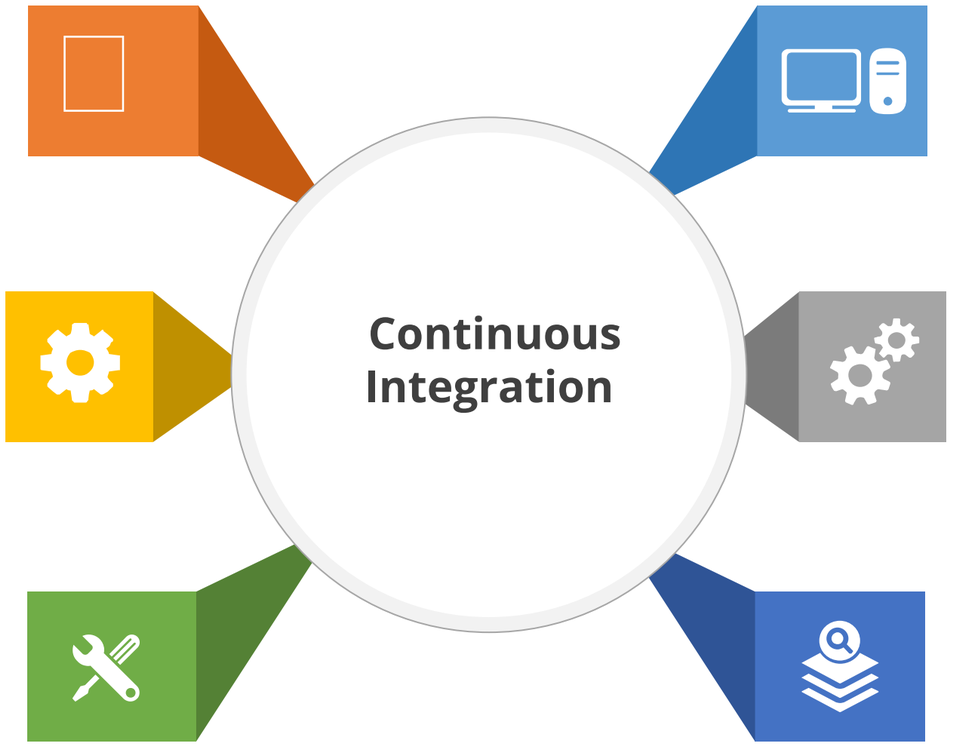
Continuous Integration, Continuous Delivery, and Continuous Deployment

Continuous Integration

Continuous integration (CI) is a development practice in which all development work is integrated as early as possible.

- The developers need to commit the changes to the source code in a shared repository at regular intervals. Every commit made in the repository is then built.
- It requires the developers to have regular builds. The general practice is that whenever a code commit occurs, a build should be triggered.
- This process helps to identify the errors in the early stages of a project.

Tasks Involved in Continuous Integration



Generating reports and analyzing

Developing and compiling

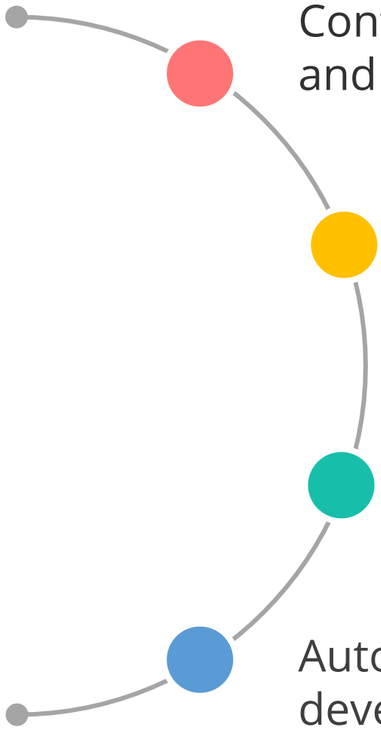
Performing functional tests and labeling code

Performing unit tests

Performing preproduction deployment

Integrating with DB

Advantages of Continuous Integration



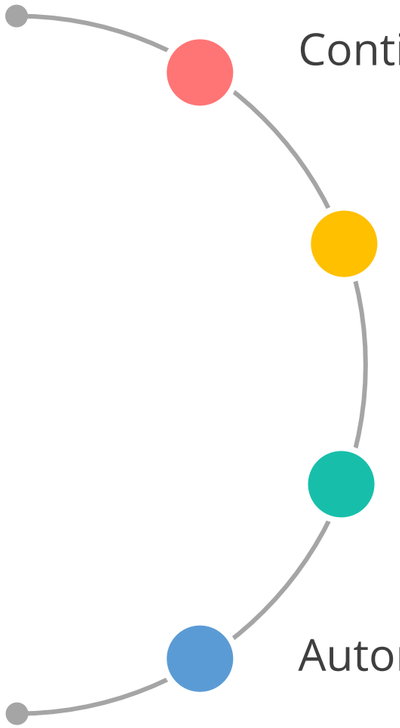
Continuous integration automatically monitors the health of your codebase, code quality, and code coverage metrics.

Technical debts are kept down, and maintenance costs are low.

Publicly-visible code quality metrics encourage developers to improve their code quality.

Automated end-to-end acceptance tests provide a clear picture of the current state of development efforts.

Advantages of Continuous Integration



Continuous integration reduces risk by providing faster feedback.

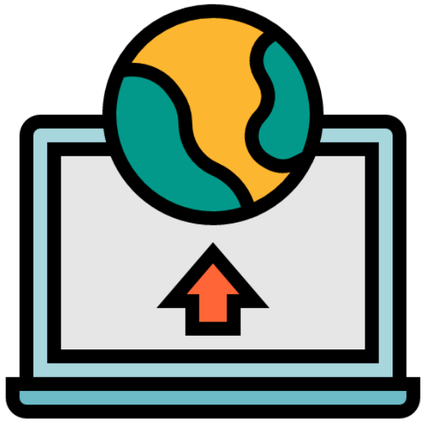
CI tools are designed to help identify and fix integration and regression issues faster, resulting in fewer bugs and quicker delivery.

CI helps simplify and accelerate delivery by automating the deployment process.

Automating the deployment process allows testers and end users quick access to the software.

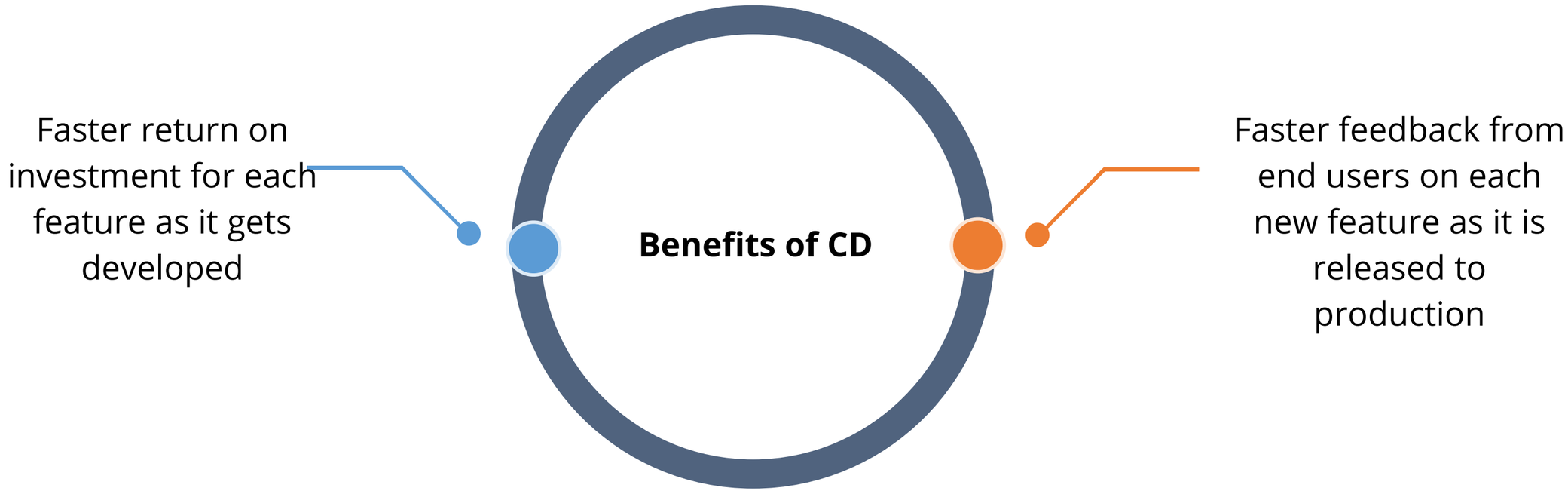
Continuous Delivery

- Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.
- With continuous delivery, any successful build that has passed all the relevant automated tests and quality gates can potentially be deployed into production and be in the hands of the end user within minutes.
- But this process is not automatic.
- It is the business and not the IT that decides the best time to deliver the latest changes.



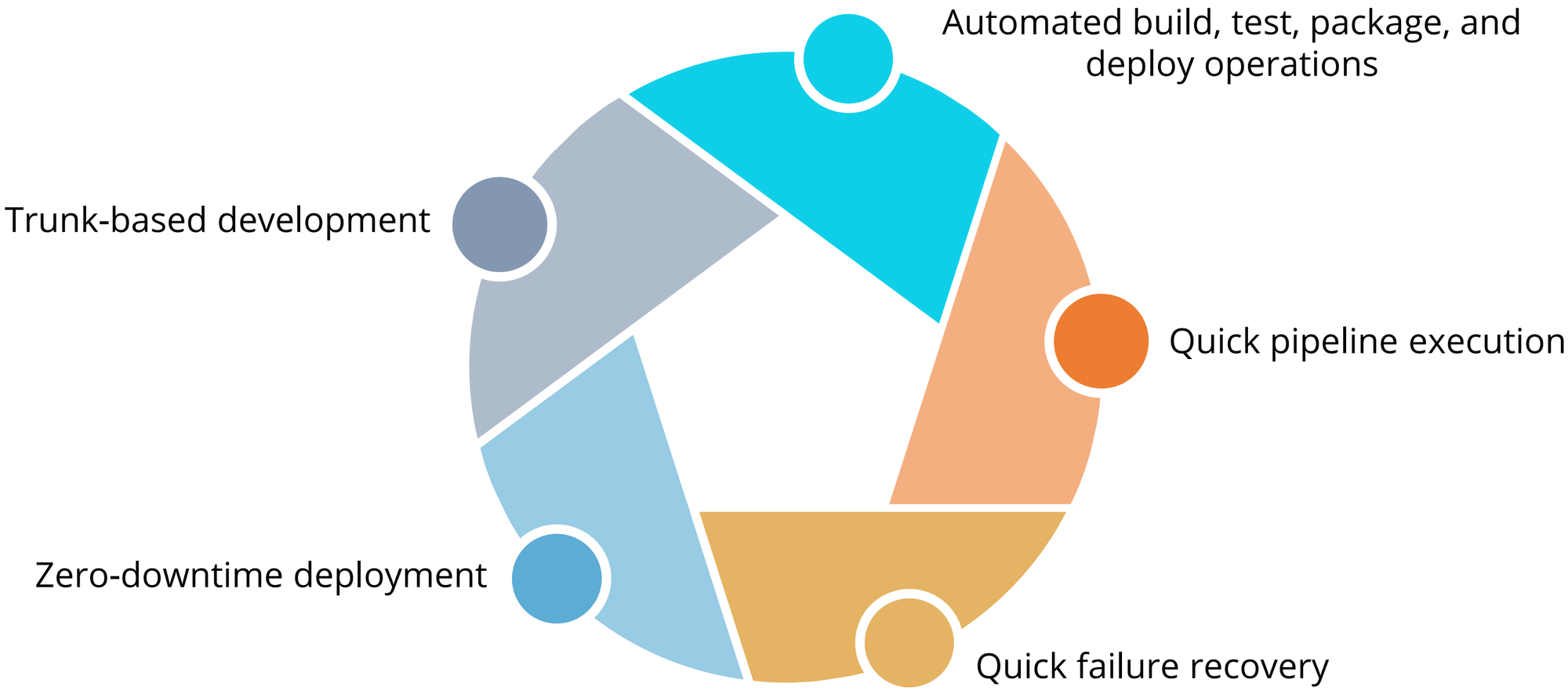
Continuous Deployment

Continuous deployment is a step up from Continuous Delivery in which every change in the source code is deployed to production automatically, without explicit approval from a developer.



Prerequisites for CI/CD

Here are a few technical prerequisites for adopting the CI/CD process:

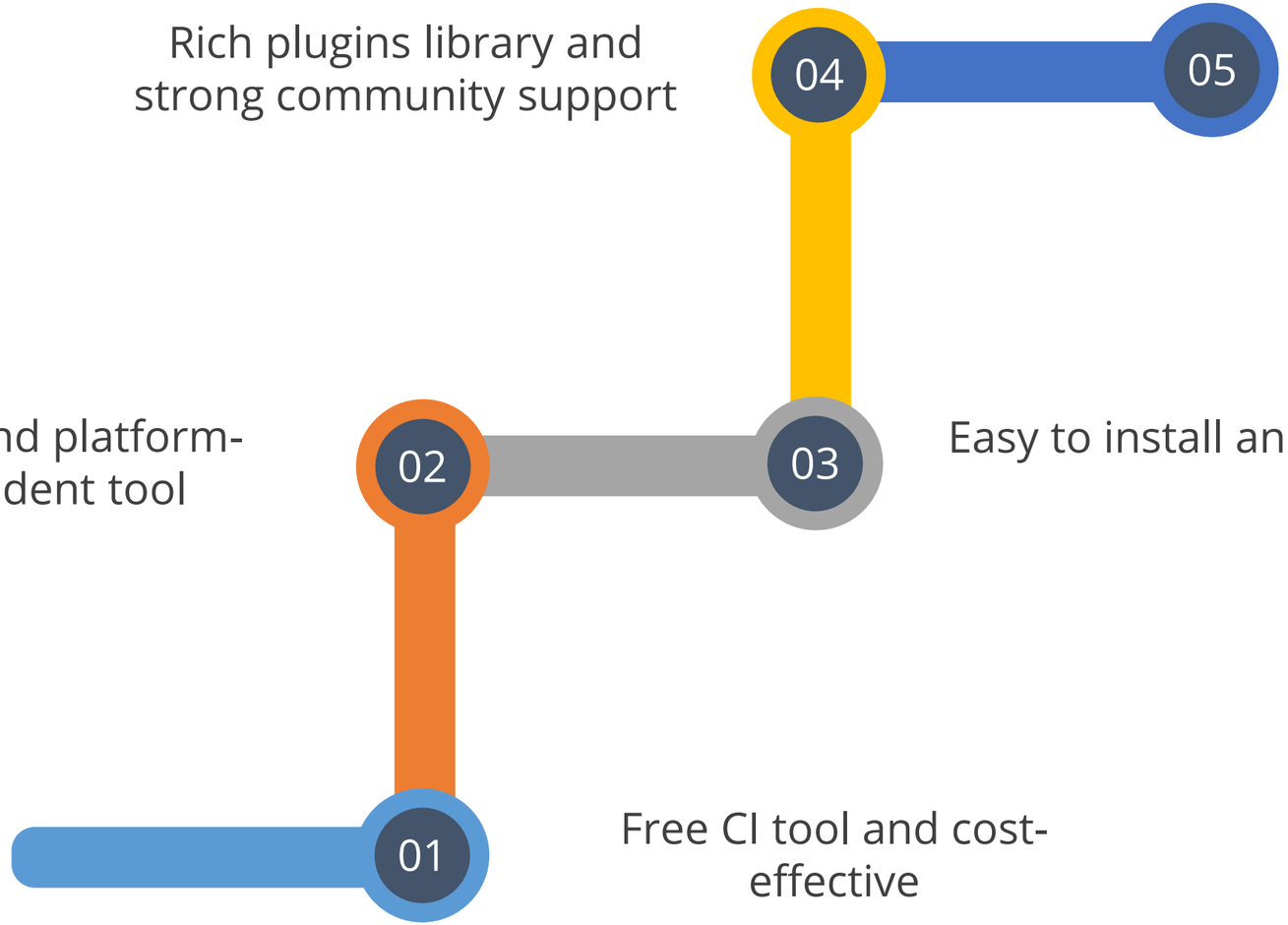


Introduction to Jenkins

What Is Jenkins?

- Jenkins is a powerful automation tool written in Java.
- It allows continuous integration and continuous delivery of projects.
- It allows to continuously deliver the software by integrating with number of testing and deployment technologies.
- It supports Windows, MacOS, and Unix-like operating systems.

Benefits of Jenkins



Free CI tool and cost-effective

Portable and platform-independent tool

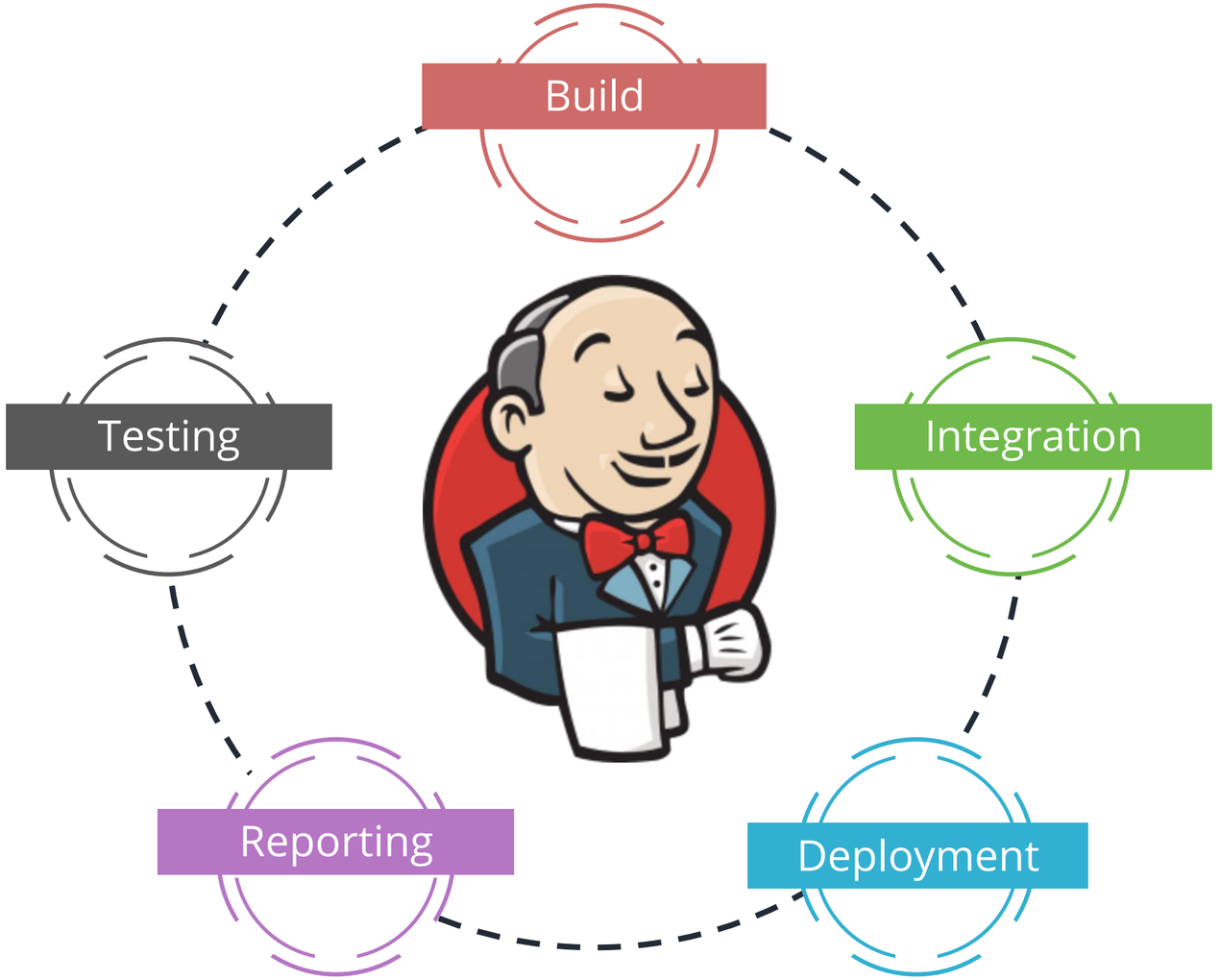
Easy to install and configure

Rich plugins library and strong community support

Limitless integrations

Role of Jenkins in DevOps

Jenkins is used to perform different roles:



Unassisted Practice

Setting Up Jenkins

Duration: 20 Min.

Problem Statement:

Set up and run Jenkins.

Unassisted Practice: Guidelines

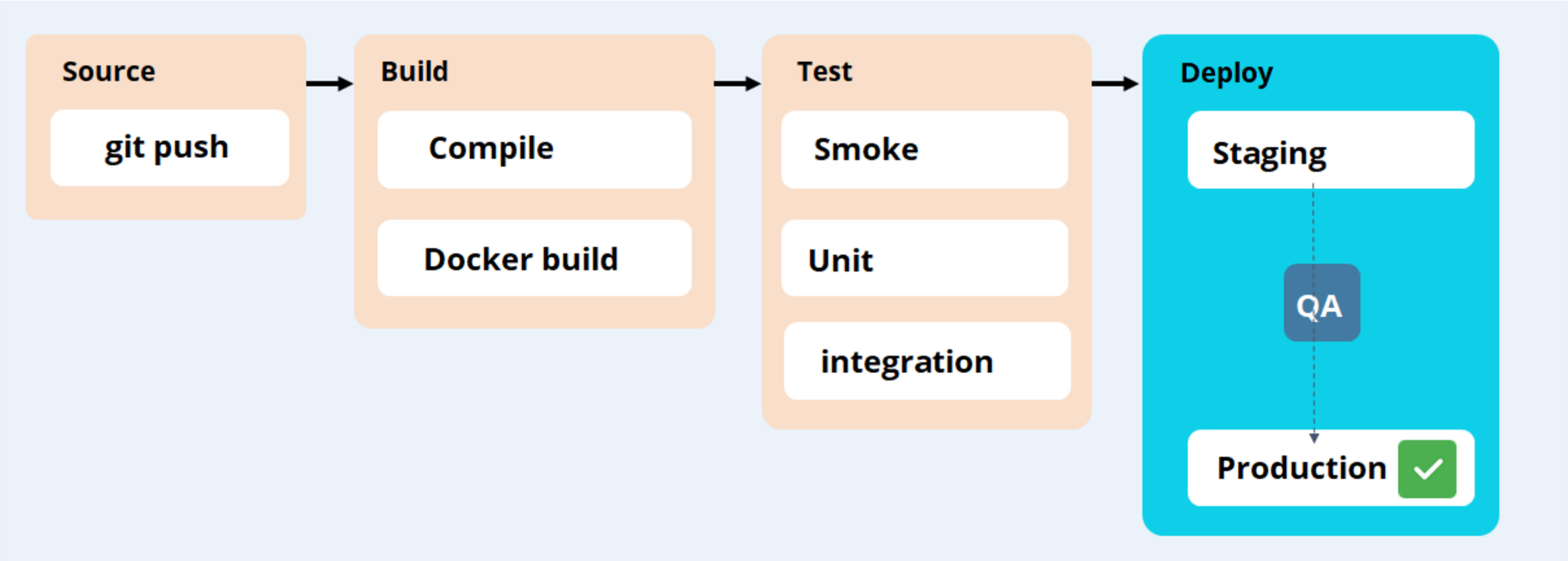
Steps to set up Jenkins:

1. Download the Java Runtime Environment.
2. Download and install the Jenkins app.

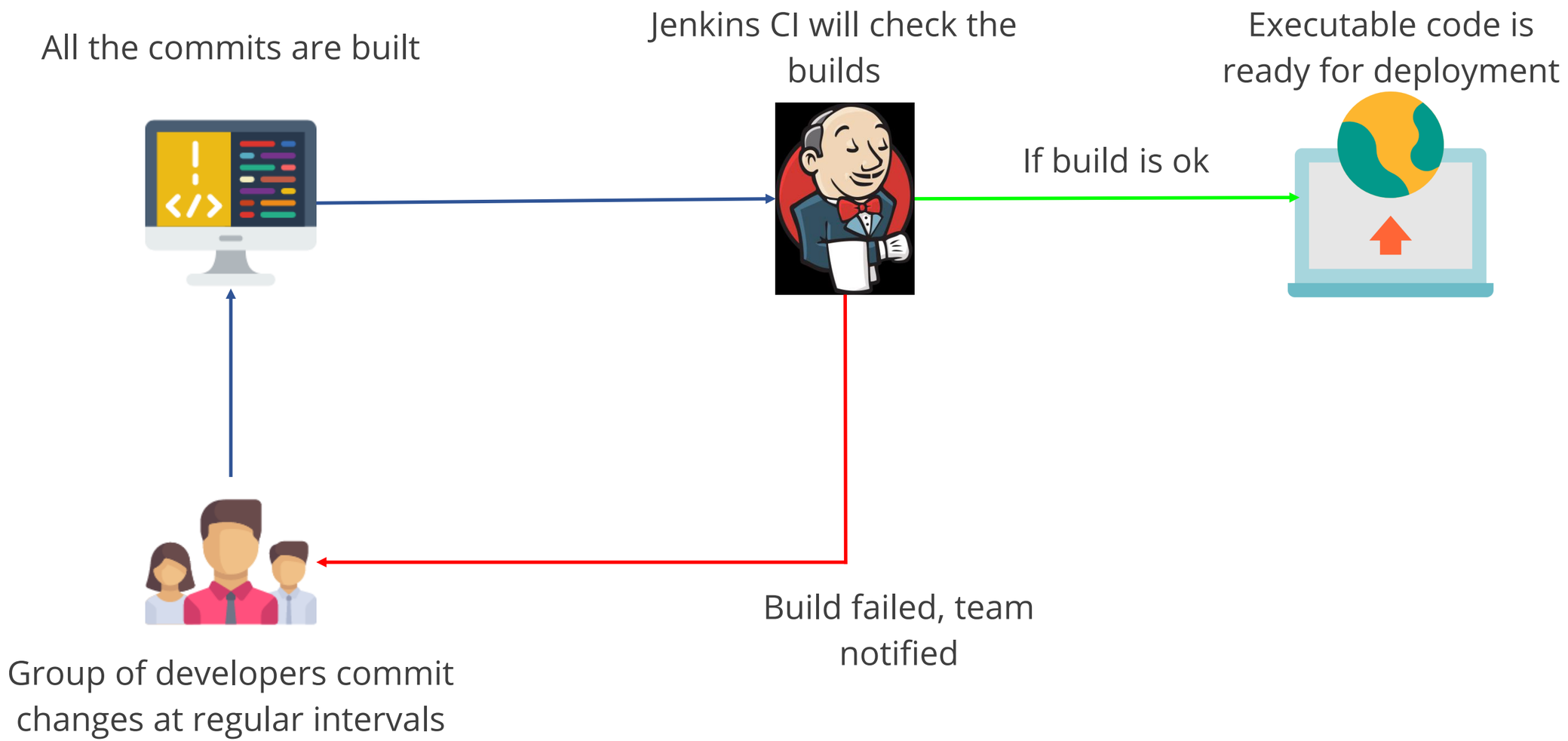
Jenkins as a CI Tool

Stages of a CI/CD Pipeline

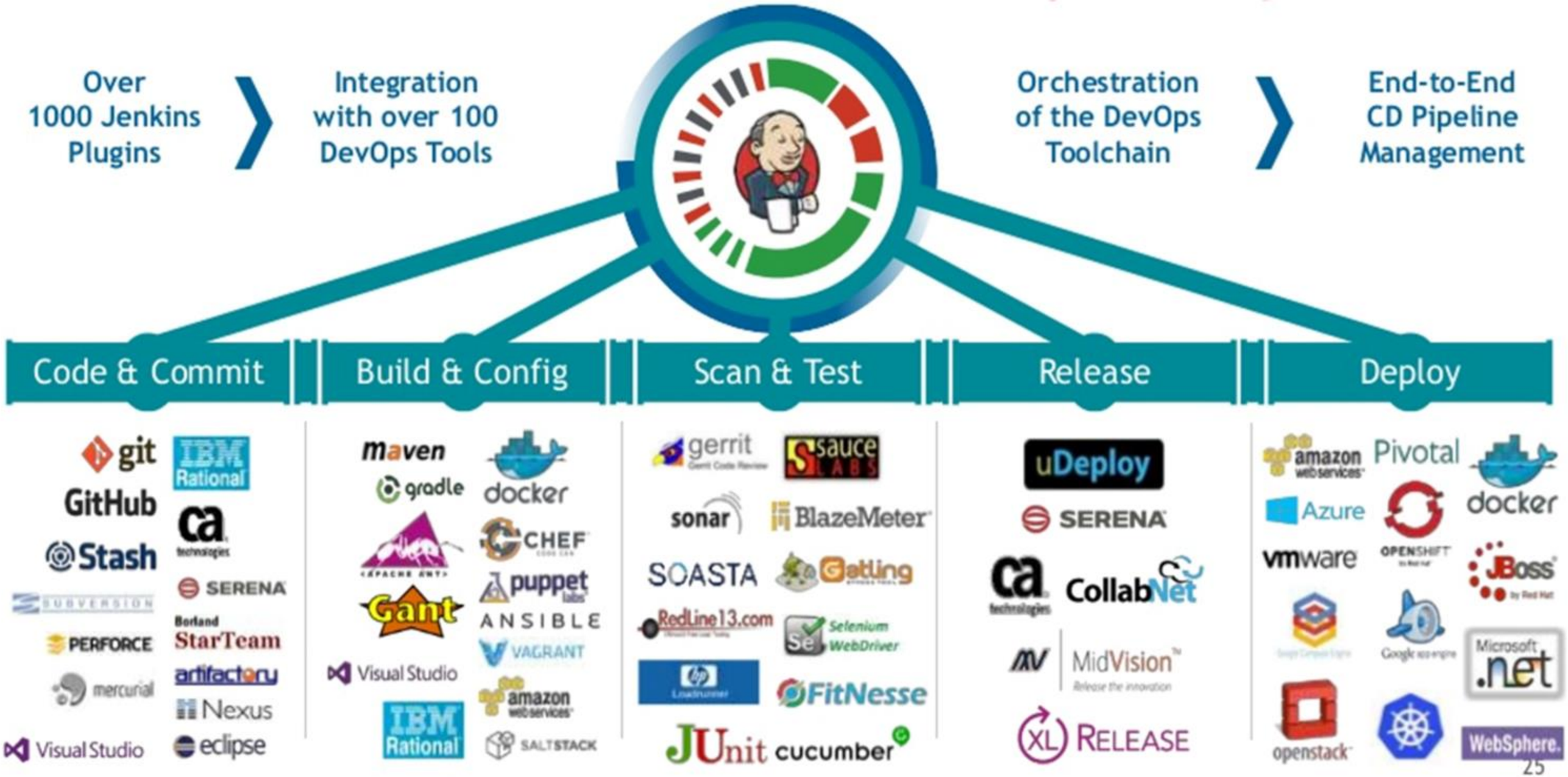
A CI/CD pipeline is essentially a runnable specification of the steps that need to be performed in order to deliver a new version of a software product. A CI/CD pipeline usually has the following stages:



How to Achieve CI with Jenkins?



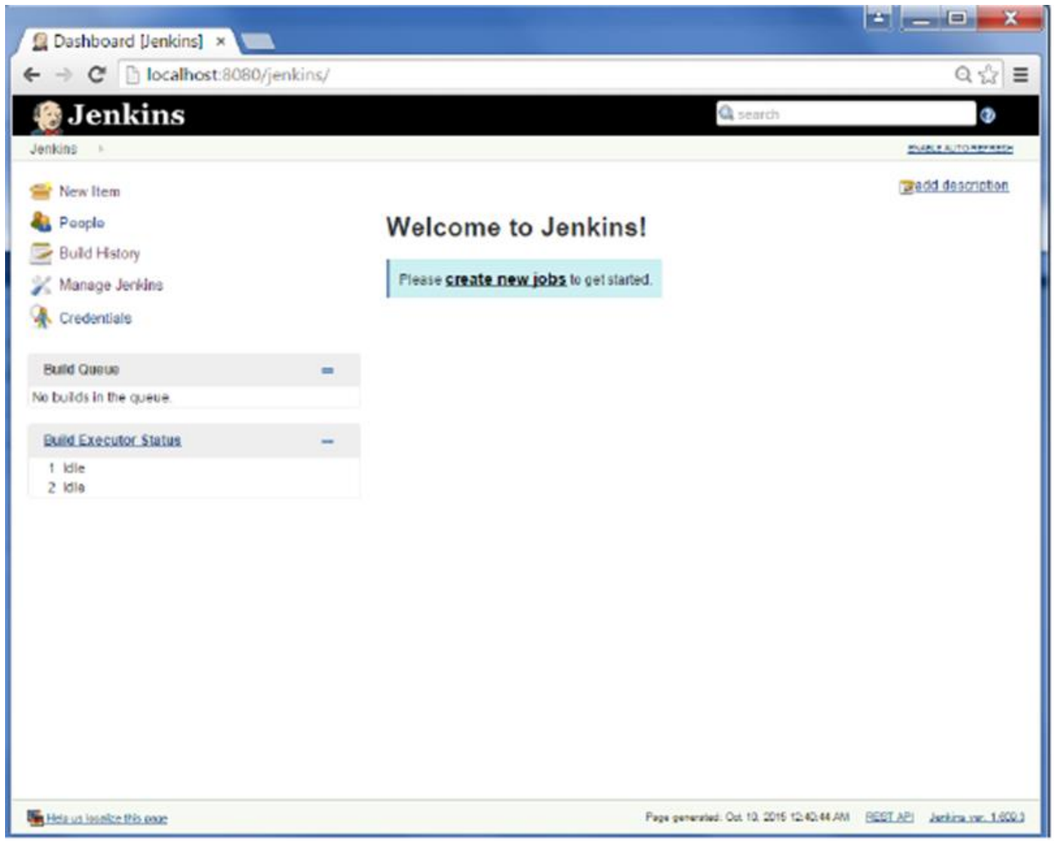
CI with Jenkins



Understanding Jenkins Build Jobs

Jenkins Build Jobs

You can create new build jobs by clicking on the *New Item* button shown in the UI below. This will give you the list of build jobs available in Jenkins to choose from.



Types of build job in Jenkins:

- Freestyle project
- Maven project
- External job
- Multi-configuration project
- Pipeline
- Folder
- GitHub organization
- Multibranch pipeline

Poll SCM, Webhook, and Build Periodically

Poll SCM

Polling involves checking the version control server at regular intervals for any changes that have been committed.

Webhook

Webhooks provide your application a way of consuming new event data from an endpoint and instead of sending repeated requests for new events, you provide the endpoint with a URL,which your application monitors.

Build Periodically

- Build periodically builds the project periodically even if nothing has changed.
- It might tests that should be run periodically (every morning for example) or a DB clean up Jenkins job or any other Jenkins job.

Scheduling Jenkins Jobs

Scheduled builds simply trigger your build job at regular intervals.

This strategy can be used for regular nightly builds.

Scheduled builds are not the same as continuous integration and could be done with something as simple as a Unix cronjob.

Scheduling Jenkins Jobs

Continuous integration requires faster feedback than scheduled builds.

Scheduled builds can be used for long running build jobs, where quick feedback is less critical.

Scheduling Jenkins Jobs

Jenkins uses cron-style syntax for scheduling the jobs.

Fields

- The meaning of these fields are as follows:
- MINUTE: Minutes within the hour (0–59)
- HOUR: The hour of the day (0–23) DOM
- DOM: The day of the month (1–31)
- MONTH: The month (1–12)
- DOW: The day of the week (0–7), where 0 and 7 are Sunday

Scheduling Jenkins Jobs

Here are the shortcut syntaxes used for scheduling build jobs:

Notation	Meaning	Example
*	Represents all possible values for a field	* * * * * means once a minute
M–N	Defines ranges	1-5 in the DOW field would mean Monday to Friday
/	Defines skips through a range	*/5 in the MINUTE field would mean every five minutes
Comma-separated list	Indicates a list of valid values	15,45 in the MINUTE field would mean at 15 and 45 minutes past every hour
Shorthand values		@yearly, @annually, @monthly, @weekly, @daily, @midnight, and @hourly

Assisted Practice

Integrate Git with Jenkins

Duration: 30 Min.

Problem Statement:

You are asked to integrate Git with Jenkins.

Assisted Practice: Guidelines

Steps to set up Jenkins:

1. Install Git plugin.

Assisted Practice

Creating a Freestyle Build Job

Duration: 20 Min.

Problem Statement:

You have been asked to create a freestyle build job using Jenkins to compile and run a simple Java program.

Assisted Practice: Guidelines

Steps to create a freestyle build job:

1. Create a GitHub repository.
2. Add a Java program to the repository.
3. Create a freestyle build job in Jenkins.
4. Build the Java program with Jenkins.

Assisted Practice

Integrate Maven with Jenkins

Duration: 40 Min.

Problem Statement:

You are asked to integrate Maven with Jenkins.

Assisted Practice: Guidelines

Steps to integrate Maven with Jenkins:

1. Install Maven plugin.
2. Set the Global Tool Configuration.
3. Fork a sample repository.
4. Integrate Maven with Jenkins.

Assisted Practice

Integrate Ant with Jenkins

Duration: 30 Min.

Problem Statement:

You are asked to integrate Ant with Jenkins.

Assisted Practice: Guidelines

Steps to integrate Ant with Jenkins:

1. Create a GitHub repository.
2. Add build.xml file to the repository.
3. Set the Global Tool Configuration.
4. Integrate Ant with Jenkins.

Builds

Parameterized Builds

Jenkins simply provides an interface to enter values for the parameters. It is the job of the build script to analyze and process the parameter values correctly.

The parameterized build plugin lets you configure parameters for a build job that are either entered by the user when the build job is triggered or from another build job.

When running a build job involving web tests, you can specify the browser to run the Selenium or WebDriver tests in. You can also upload files to be used by build jobs.

For a deployment build job, you can either choose the target environment from a drop-down list or specify the version of the application.

Assisted Practice

Remote Triggering of a Parameterized Build

Duration: 30 Min.

Problem Statement:

You are asked to execute the remote triggering of a parameterized build.

Assisted Practice: Guidelines

Steps to execute the remote triggering of a parameterized build:

1. Configure a parameterized build in Jenkins.
2. Trigger a parameterized build remotely.

Key Takeaways

- Continuous integration is a development practice of integrating code into a shared repository.
- Jenkins is a powerful automation tool written in Java which allows continuous integration and continuous delivery of projects.
- A CI/CD pipeline usually has four stages: source, build, test, and deploy.

Lesson-End Project

Building a Maven Project with Jenkins

Project Agenda: Use Jenkins to create a Maven build job which will compile and run a local Maven project.

Description: You are a DevOps engineer, and you are asked to create a sample maven project, add a sample java code in it, and deploy the same on Jenkins.

Thank You