DevOps

Caltech | **Center for Technology & Management Education**

Post Graduate Program in DevOps

simplilearn

Continuous Integration, Continuous Deployment, and Build Tools

Learning Objectives

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

- ◉ Describe the importance of continuous integration and continuous deployment

- ◉ List the features of Jenkins and demonstrate their uses

- ◉ List the features of TeamCity and demonstrate their uses

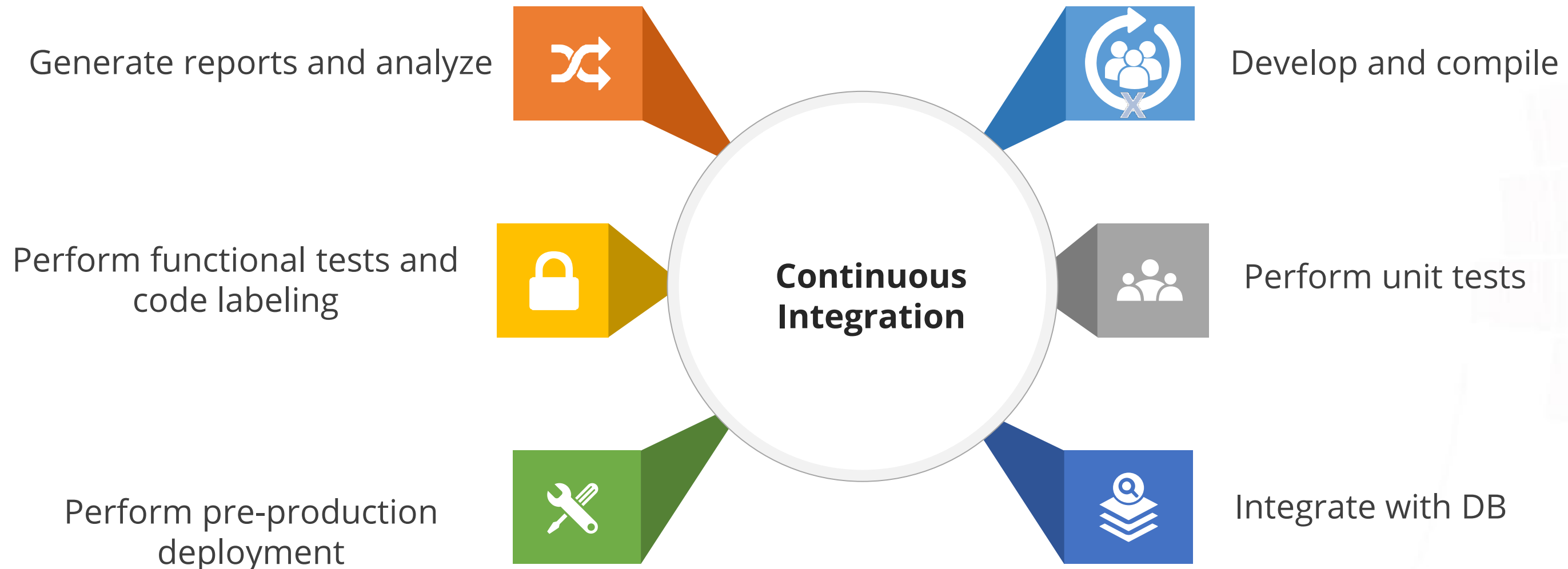- ◉ Select a suitable build tool for your organization

Overview and Importance of Continuous Integration and Continuous Deployment
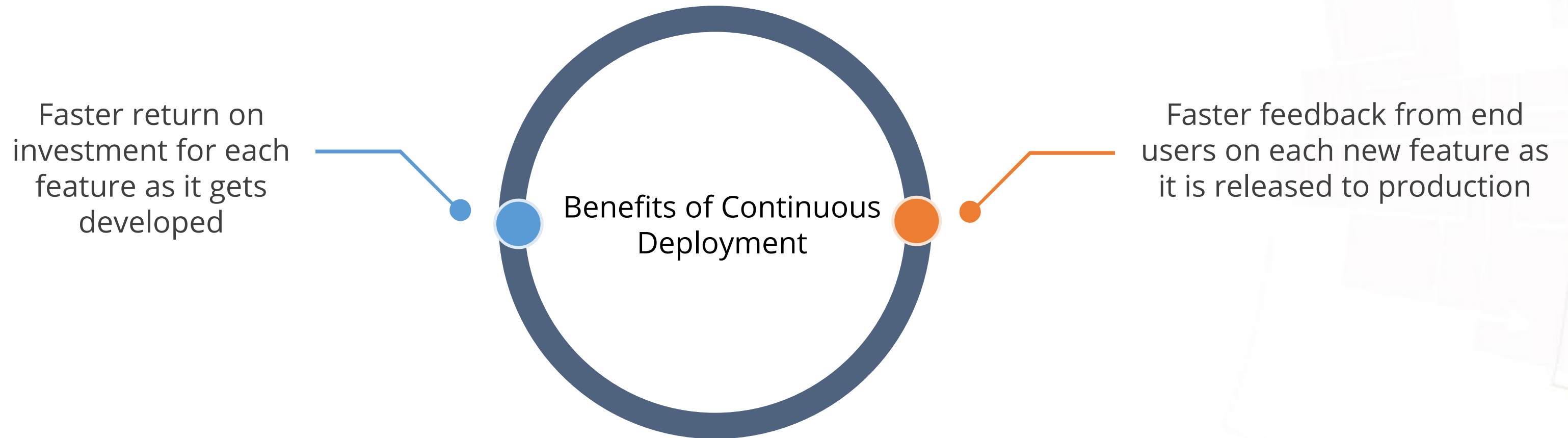
Overview of Continuous Integration

Continuous integration is a development practice of code integration into a shared repository.

Each integration is verified by an automated build and automated tests.

Generate reports and analyze

Perform functional tests and code labeling

Perform pre-production deployment

Continuous Integration

Develop and compile

Perform unit tests

Integrate with DB

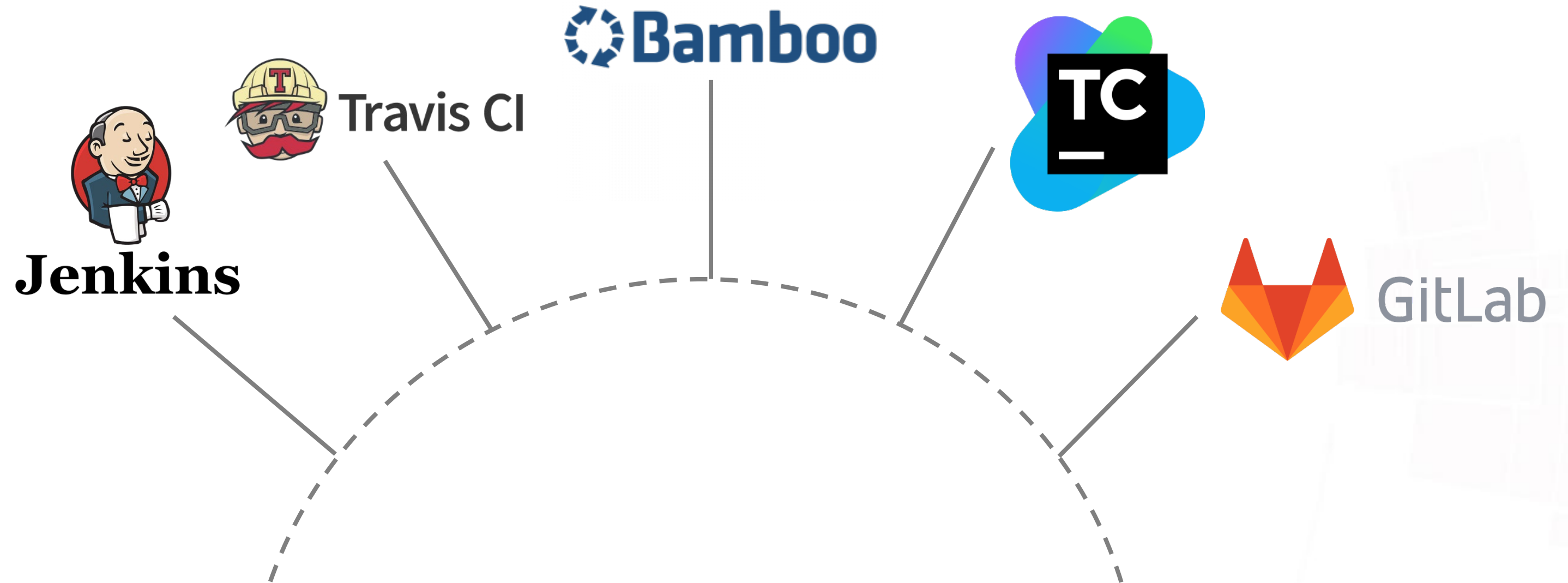Caltech | Center for Technology & Management Education   simplilearn
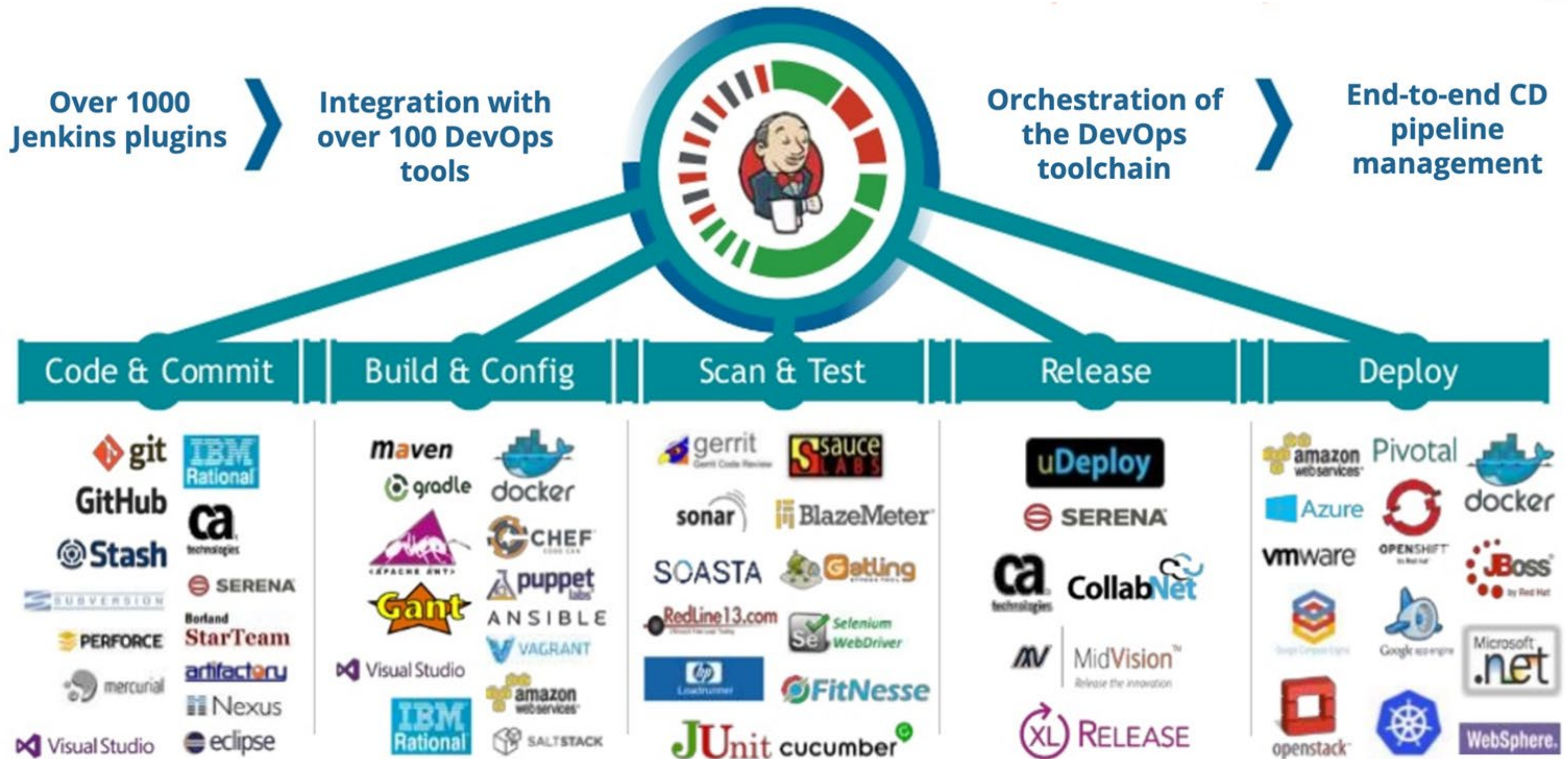
Overview of Continuous Deployment

Continuous Deployment is an extension of continuous integration. It targets to reduce the time between development team writing one new line of code and using it in production.

Faster return on investment for each feature as it gets developed

Benefits of Continuous Deployment

Faster feedback from end users on each new feature as it is released to production

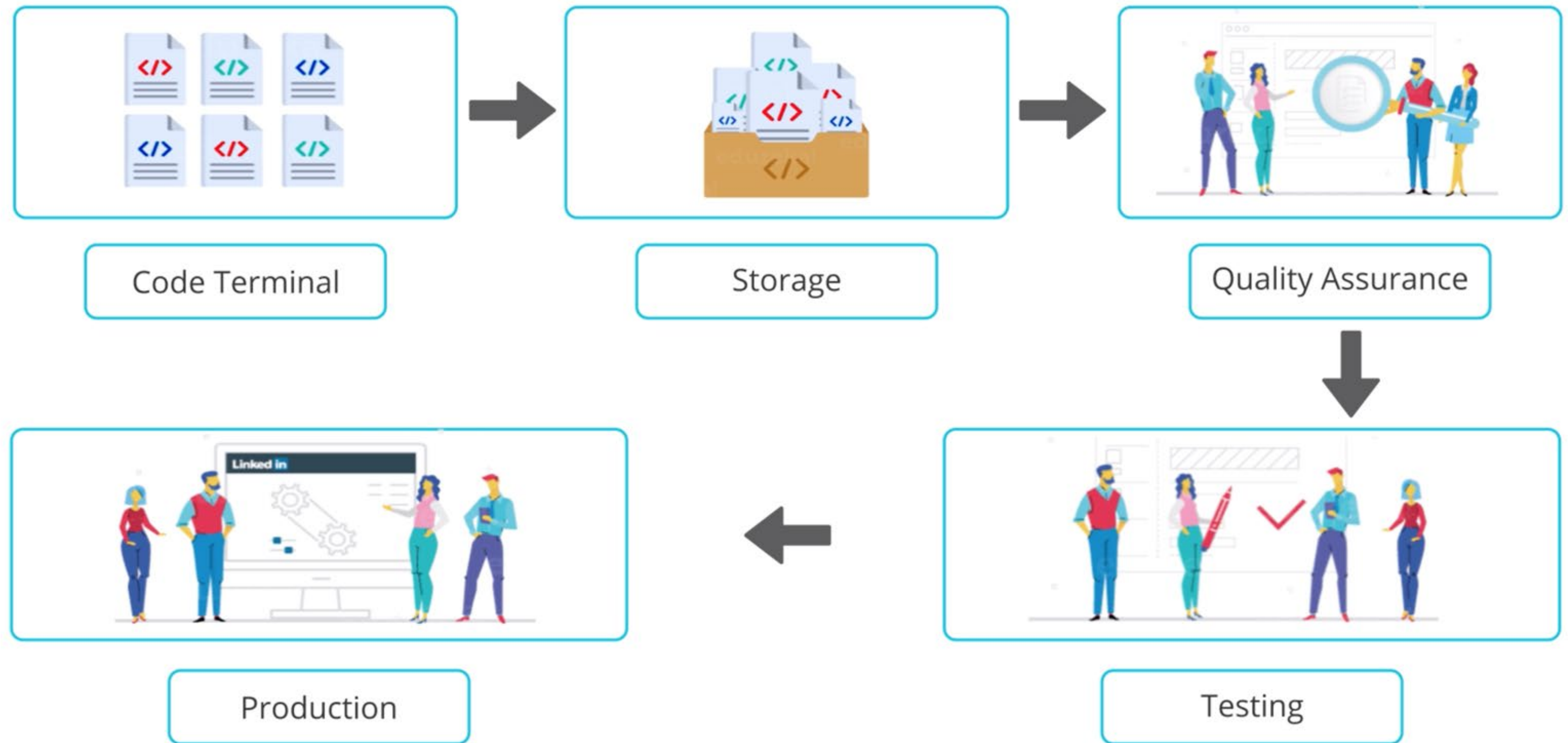Caltech | Center for Technology & Management Education

simplilearn

Popular Tools in Continuous Integration and Continuous Deployment

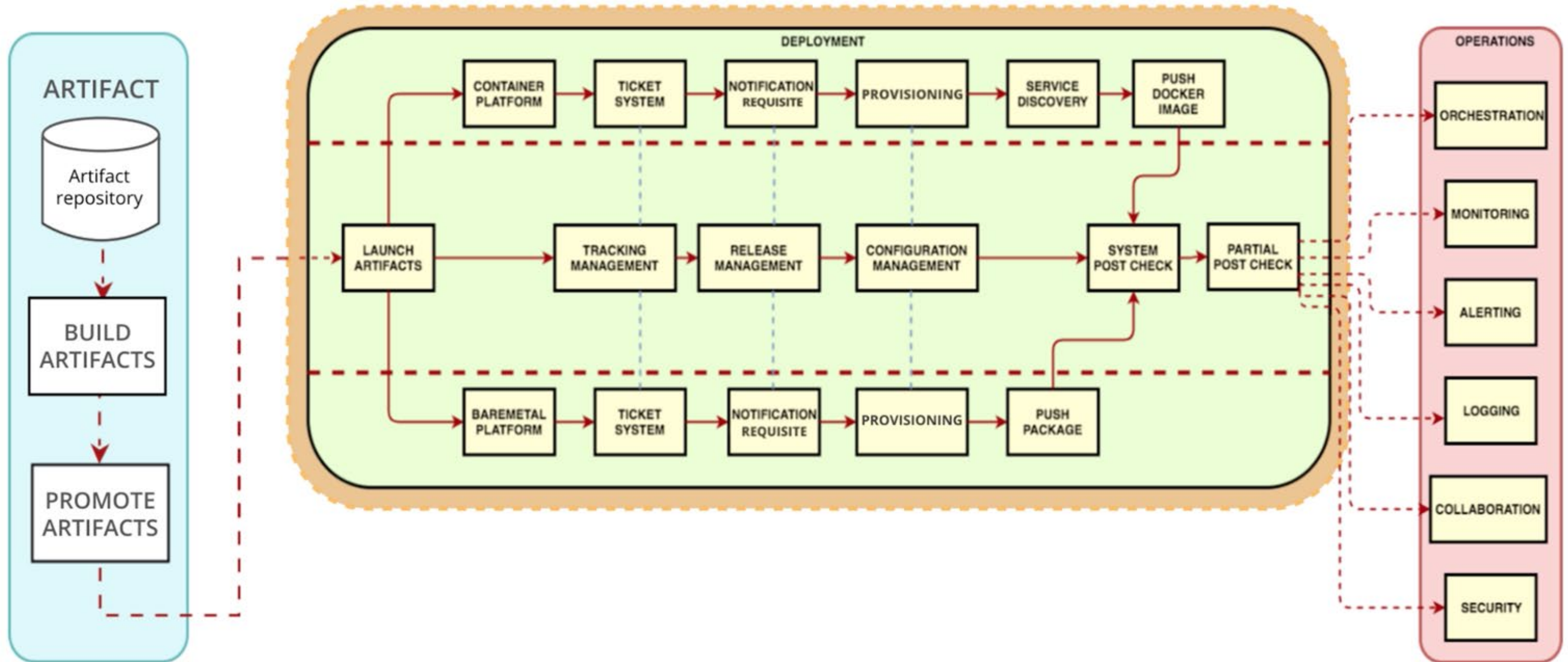# Continuous Integration with Jenkins

Continuous Deployment with Jenkins

Code Terminal

Storage

Quality Assurance

Production

Testing

A case study by LinkedIn

Continuous Integration with TeamCity

Environments which TeamCity supports

Supported Web Browsers
- FireFox
- Internet Explorer
- Safari
- Opera
- Google Chrome

Notification and Monitoring
- E-mail
- Jabber/XMPP
- RSS feeds
- IDE
- Windows System Tray

Version Control Systems
- Subversion
- Perforce
- CVS
- Bortand Star Team
- IBM Rational ClearCase (Base and UCM)
- Team Foundation Server
- Microsoft Visual SourceSafe
- Mercurial

Development Frameworks

Java
- Build runners
  - Ant
  - Maven2
  - Intellij IDEA Project
  - Duplicates Finder — Intellij IDEA project, Maven2 project
  - Code Inspections (Java, HTML & CSS JavaScript, E JB, JSP, JSF stunts & more) — Intellij IDEA project, Maven2 project
- Testing frameworks
  - JUnit
  - TestNG

.NET
- Build runners
  - MSBuild
  - NAnt
  - MS Visual Studio Solutions — 2003, 2005, 2008
  - FxCop
  - Duplicates Finder (C# up to version 2.0 & Visual Basic .NET up to version 8.0)
- Testing frameworks
  - NUnit
  - MS Test
  - Gallio
  - xUnit

Other runners
- Build runners
  - Command line runner
  - Rake runner (For Ruby development)
- Testing frameworks
  - Boost Test (C++)
  - CppUnit (C++)
  - Unit test (Python)
  - Nose (Python)

IDEs
- Eclipse
- Intellij IDEA
- MS Visual Studio

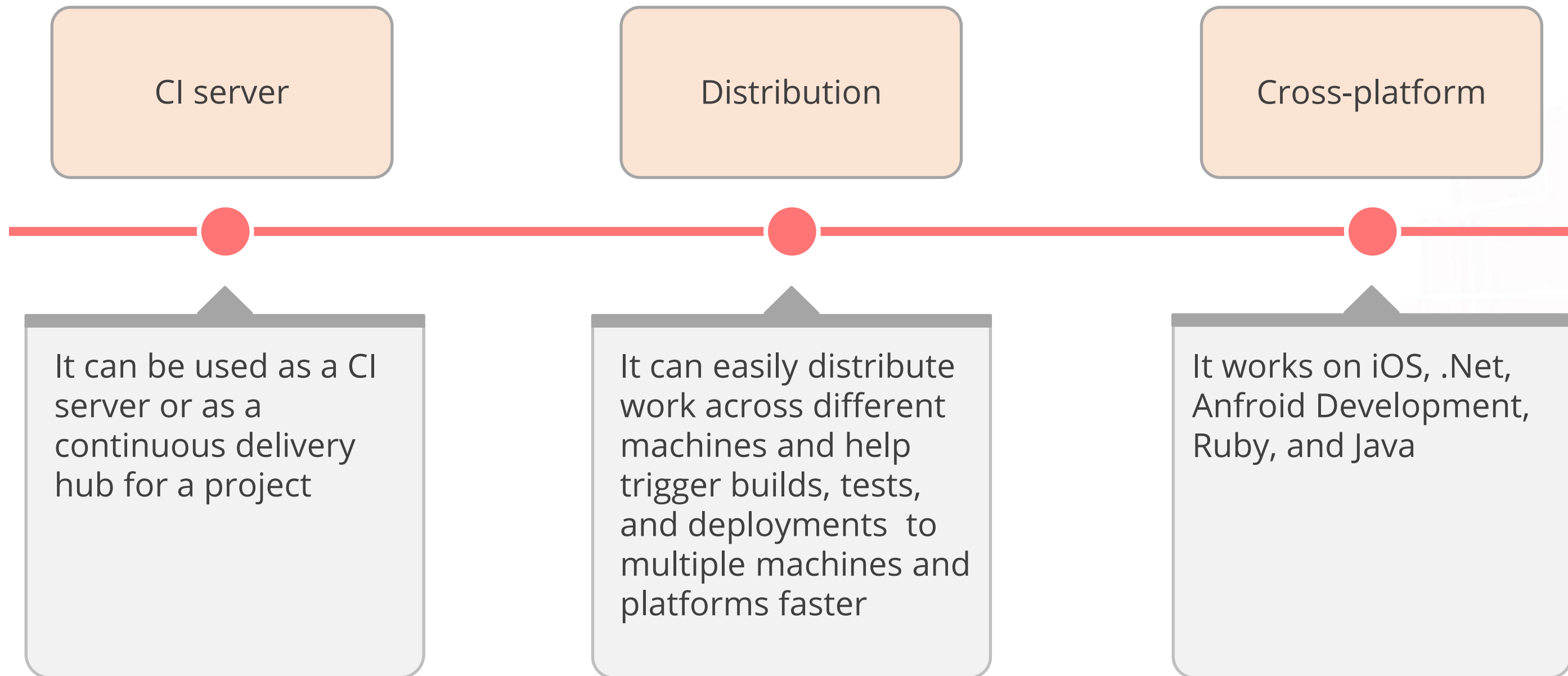Caltech | Center for Technology & Management Education

simplilearn

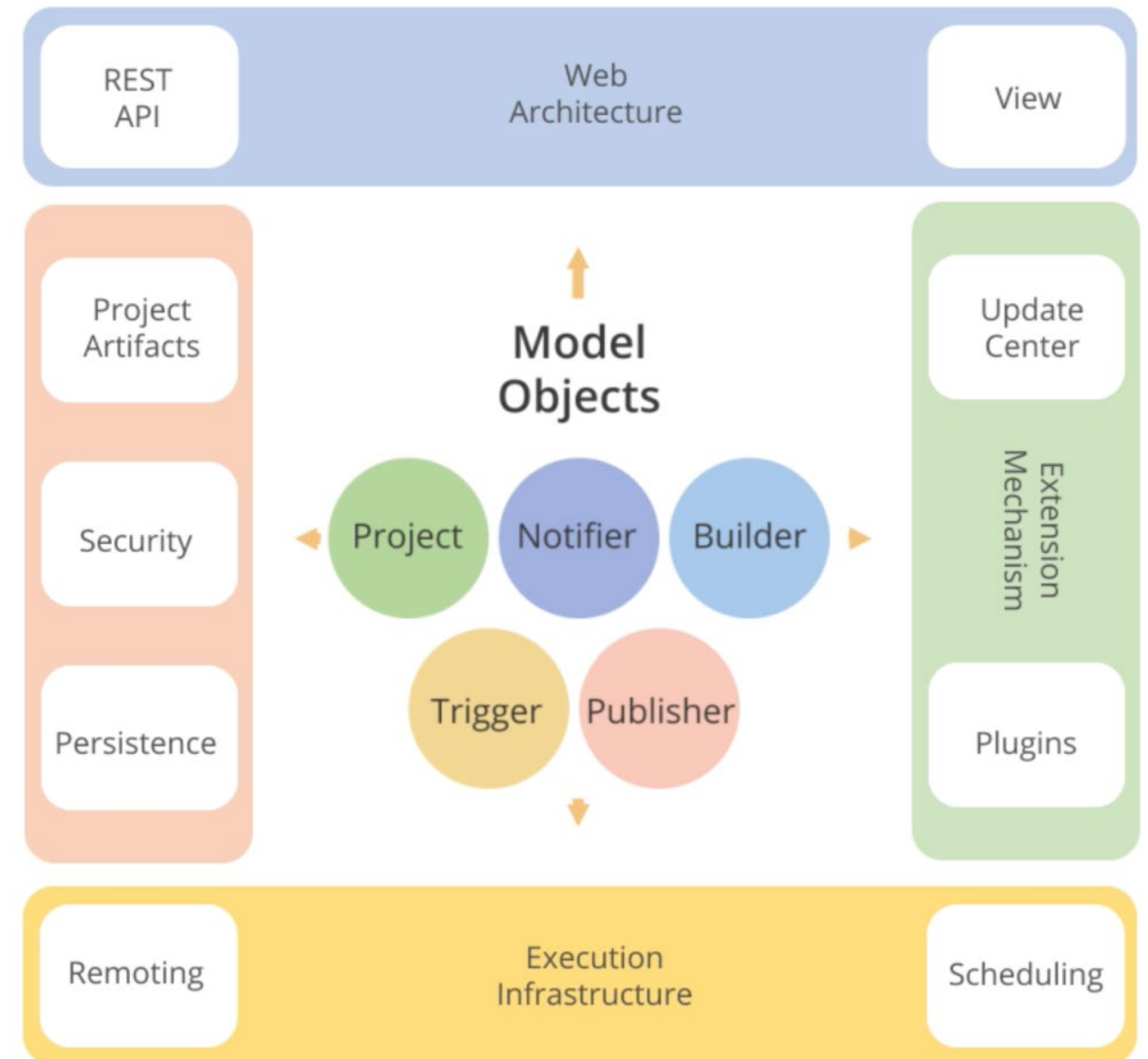# Continuous Deployment with TeamCity

Overview and Features of Jenkins

Jenkins as a Continuous Integration Tool

Jenkins is a Java-based, open source automation tool. It functions as a server and is a software development and cross-platform tool used for continuous integration and continuous deployment.
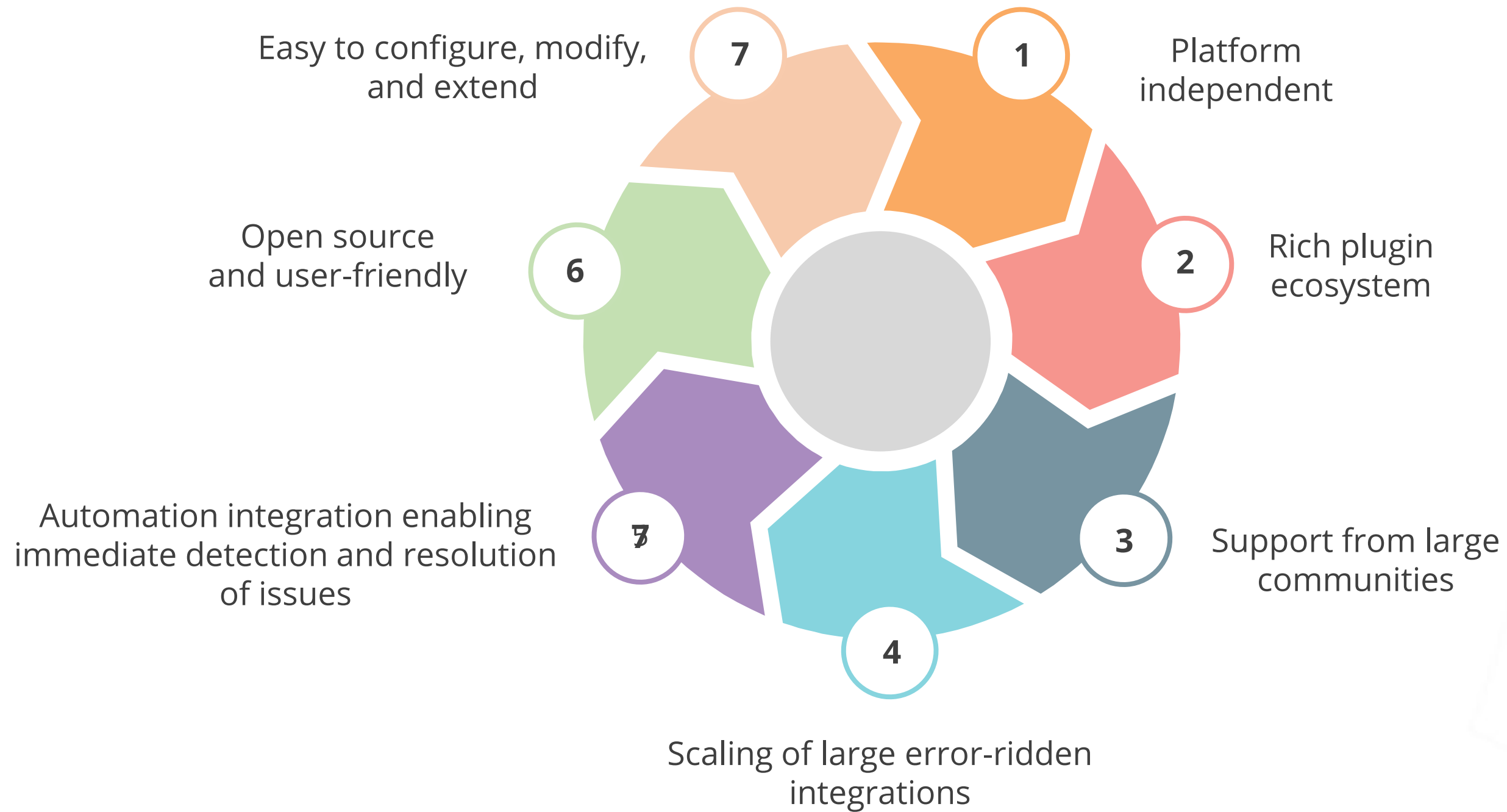
CI server	Distribution	Cross-platform
It can be used as a CI server or as a continuous delivery hub for a project	It can easily distribute work across different machines and help trigger builds, tests, and deployments to multiple machines and platforms faster	It works on iOS, .Net, Anfroid Development, Ruby, and Java

Architecture of Jenkins

- Jenkins has classes like project and build.

- It uses Jelly as the view technology.

- It uses file system to store its data. Directories are

  created inside $JENKINS_HOME.

- It supports plugins, which can plug into those

  extension points and extend the capabilities of Jenkins.

Popular Features of Jenkins

1. Platform independent
2. Rich plugin ecosystem
3. Support from large communities
4. Scaling of large error-ridden integrations
5. Automation integration enabling immediate detection and resolution of issues
6. Open source and user-friendly
7. Easy to configure, modify, and extend

Caltech | Center for Technology & Management Education

simplilearn

Build Status and Job Health

Status of the build	Description
🔴	Failed
🟡	Unstable
🔵	Success
⚪	Pending
⚪	Disabled
⚪	Aborted

Figure a: Build status

Job health	Description
☀️	No recent builds failed
🌤️	20-40% of recent builds failed
☁️	40-60% of recent builds failed
🌧️	60-80% of recent builds failed
⛈️	All recent builds failed
	Unknown status

Figure b: Weather reports

Assisted Practice
Set up Jenkins

Problem Statement: You are given a project to install and configure Jenkins on your Ubuntu operating system.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

Assisted Practice: Guidelines to Install and Configure Jenkins

1. Login to your Ubuntu lab provided with the course.

2. Open the terminal and execute the command available in the lab document 3.1 to add the key to the system.

3. Edit the sources.list file, add the command to the file, and save it.

4. Update the apt-get package.

5. Install JDK 8+ version.

6. Install Jenkins via apt-get package.

7. Navigate to x.x.x.x:8080 in the browser of your virtual Machine.

8. Get the password and enter it in the Jenkins window.

9. Create a new role/job in Jenkins.

10. Explore the free style project section and build section.

Overview and the Features of TeamCity

TeamCity as a Continuous Integration Tool

TeamCity is a Java-based, management and continuous integration server. It is a licensed commercial software used for continuous integration and continuous deployment.
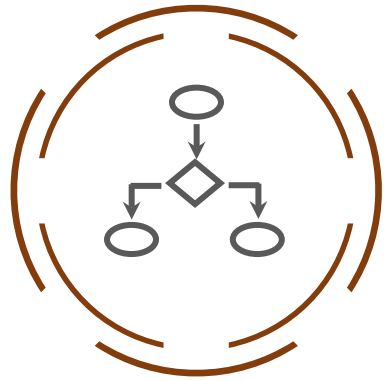
Gated commits

Build grid

Integrated code

Integration with IDEs

Cross-platform support

Caltech | Center for Technology & Management Education

simplilearn

TeamCity Workflow

SVN Server

TC Server

Build SUT
Build Test Automation
Execute Test
Automation SUT

Trigger Build

3

2

Build Agent

4

Commit
Changes

1

Tester

Developer

Caltech | Center for Technology & Management Education

simplilearn

Popular Features of TeamCity

Pre-tested commit

User management and roles

Instant notifications

Configurable test

TeamCity

Code coverage and monitoring

Easy to verify code

Comprehensive build infrastructure

Enhanced VCS

Assisted Practice
Set up Teamcity

Problem Statement: You are given a project to install and configure TeamCity on your Ubuntu operating system.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

Assisted Practice: Guidelines to Install and Configure TeamCity

1. Login to your Ubuntu lab provided with the course.

2. Download TeamCity from the official site.

3. Unzip the folder and install TeamCity.

4. Provide the read, write, and execute mode access to TeamCity.

5. Run TeamCity at x.x.x.x:8111 where x.x.x.x is your IP address.

6. Create an account in TeamCity and add the basic details to complete the set up process.

7. Explore options such as **Projects**, **Changes**, **Agents**, and **Build Queue**.

Build Tools and Their Uses

Build Tools

Build tools are programs that automate the creation of executable applications from the source code.

Automation tools allow the build process to be more consistent.

Make
Software

<APACHE ANT>

GRUNT

BASH
THE BOURNE-AGAIN SHELL

#!/bin/bash

Apache Maven
http://maven.apache.org/

Gulp

Popular Features of Build Tools

- Compiling to binary code
- Packaging a compiled program
- Running automated test cases
- Deploying to production
- Generating documents
- Generating release notes

Caltech | Center for Technology & Management Education

simplilearn
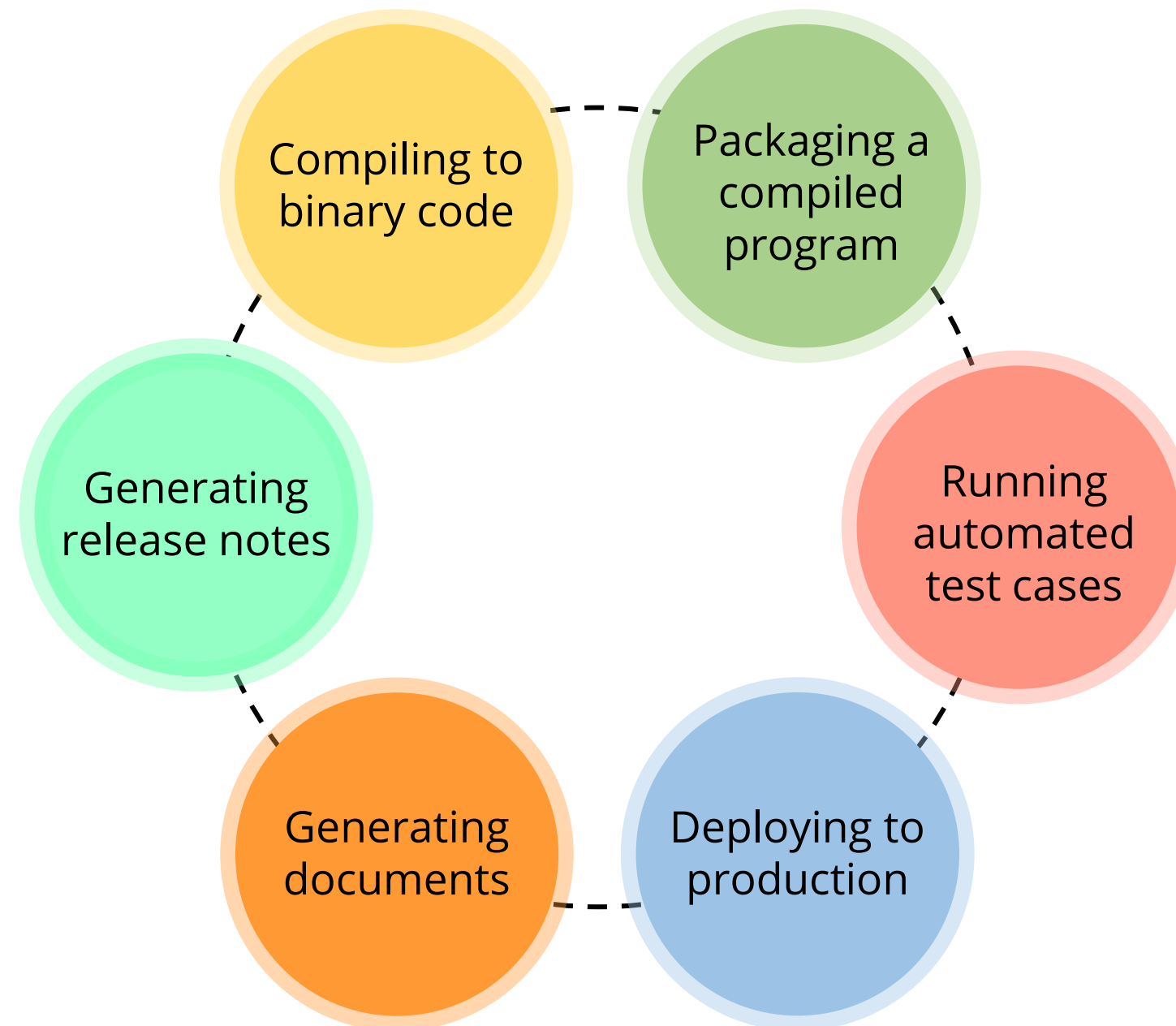
Overview of Apache Ant

Apache Ant is a Java library and a command-line tool. It aims to drive processes described in build files as targets and extension points dependent on each other.

Supplies a number of built-in tasks allowing to compile, test, and run

Flexible and does not impose coding conventions

Features of

Pilot any type of process

Build solution combining build tool and dependency with Apache Ivy

<APACHE ANT>

Users can develop custom "Antlibs" using Java

Solves Make's portability problems

Caltech | Center for Technology & Management Education
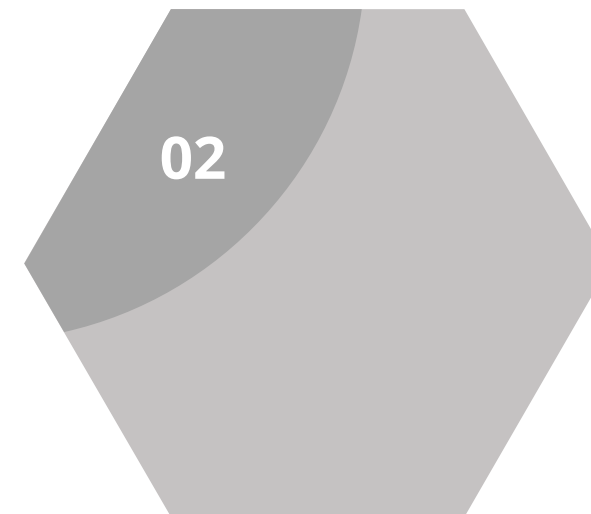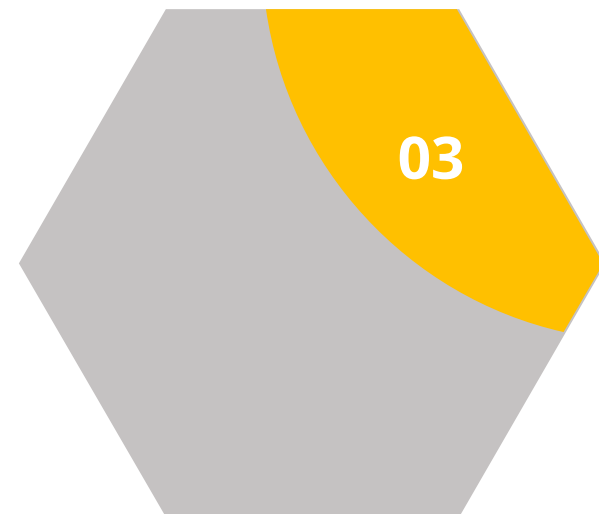
simplilearn

Limitations of Apache Ant

Ant build files are complex and verbose as they are hierarchical and partly ordered

04

01
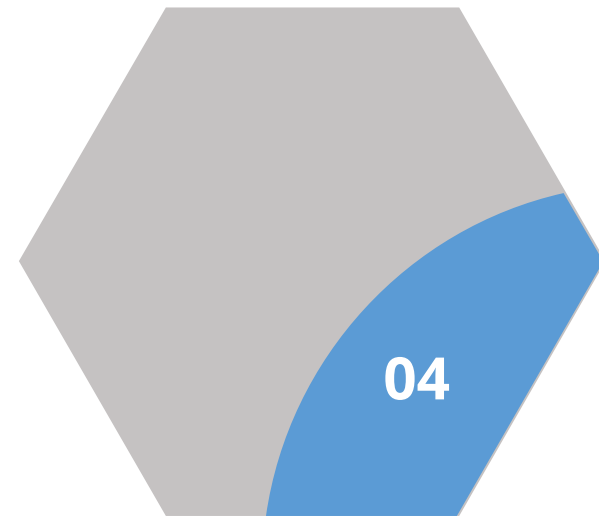
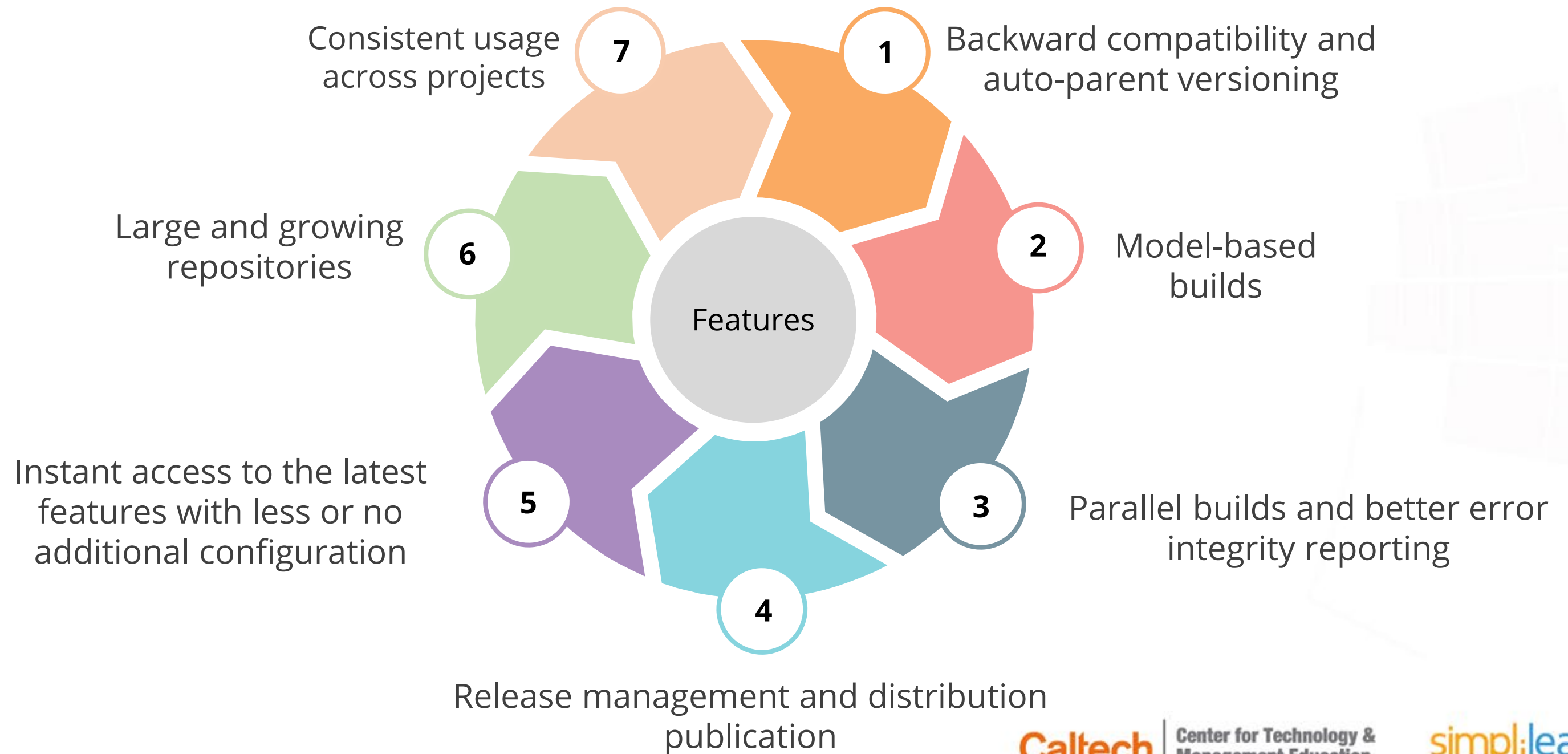Undefined properties are not raised as errors but left as unexpanded reference

Older tasks use default value, which are not consistent, and changing defaults would break existing Ant scripts

03

02

Ant has limited fault handling rules and lazy property evaluation is not supported

Caltech | Center for Technology & Management Education

simplilearn

Overview of Maven

Apache Maven is a software project management and comprehension tool. Based on Project Object Model (POM), Maven can manage a project's build reporting and documentation from a central piece of information.

Features

7 — Consistent usage across projects

1 — Backward compatibility and auto-parent versioning

2 — Model-based builds

3 — Parallel builds and better error integrity reporting

4 — Release management and distribution publication

5 — Instant access to the latest features with less or no additional configuration

6 — Large and growing repositories

Drawbacks of Maven

Unreliable with Eclipse

Unable to depend on the outcome status

Slow and a partial black box

Verbose and complex

Maven over Ant

Better collaboration and debugging

More powerful builds

More componentized builds and reduced duplication

Apache Maven
http://maven.apache.org/

Better dependency management

More consistent project structure

Caltech | Center for Technology & Management Education

simplilearn

Project Object Model (POM)

Project Object Model is an XML representation of a Maven project, which provides general configurations such as project's name, its owner, and its dependencies on other projects.

- The POM needs to define the Group ID Artifact ID, and Version
- The Packaging should also be declared – the default is jar

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                        http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>
    <groupId>eu.totaleclipse</groupId>
    <artifactId>calculator</artifactId>
    <version>1.0</version>
    <packaging>jar</packaging>
</project>
```

Caltech | Center for Technology & Management Education

simplilearn

Overview of Grunt

Grunt is a JavaScript-based task runner, which is used to automate repetitive tasks in a workflow.

It can be used as a command-line tool for JavaScript objects.

Eases workflow such as writing a set up file

Speeds the development flow and enhances performance

Helps in automation of repetitive task with less effort

Supports small infrastructure which is a best fit for new codebase

Minifies the files such as .html, .CSS

Aims at reducing the chances of errors during repetitive tasks

Includes built-in tasks to extend functionality of plugins

Currently, it has over 4000 plugins and can be used in larger production sites

Caltech | Center for Technology & Management Education

simplilearn

Overview of Gulp

Gulp is an open-source JavaScript toolkit used as a streaming build system in front-end web development. It automates time-consuming and repetitive tasks involved in development.

Features

- Code minification and concatenation
- Usage of pure JavaScript code
- Converts LESS or SASS to CSS compilation
- Manages file manipulation in the memory

Advantages

- Easy to code
- Easy to test the web apps
- Plugins are simple to use

Disadvantages

- More number of dependencies
- Multiple tasks cannot be performed
- Configuration is tedious

Assisted Practice
Continuous Integration with Jenkins and Maven

Problem Statement: You are given a project to configure Jenkins, poll Git commits and build the project code using Maven on your Ubuntu operating system.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

Assisted Practice: Guidelines to Configure Jenkins and Maven

1. Login to your Ubuntu lab provided with the course.

2. Login to Jenkins and create the first Jenkins job.

3. Install and configure Maven.

4. Configure Jenkins with Java, Git, and Maven.

5. Create a Jenkins job for your maven build project and run the project.

6. Poll Git for commits and automatically trigger the build.

7. Build the trigger using Push mechanism instead of Pull.

8. Repeat steps 6 and 7 multiple times to observe the results at **console output** section.
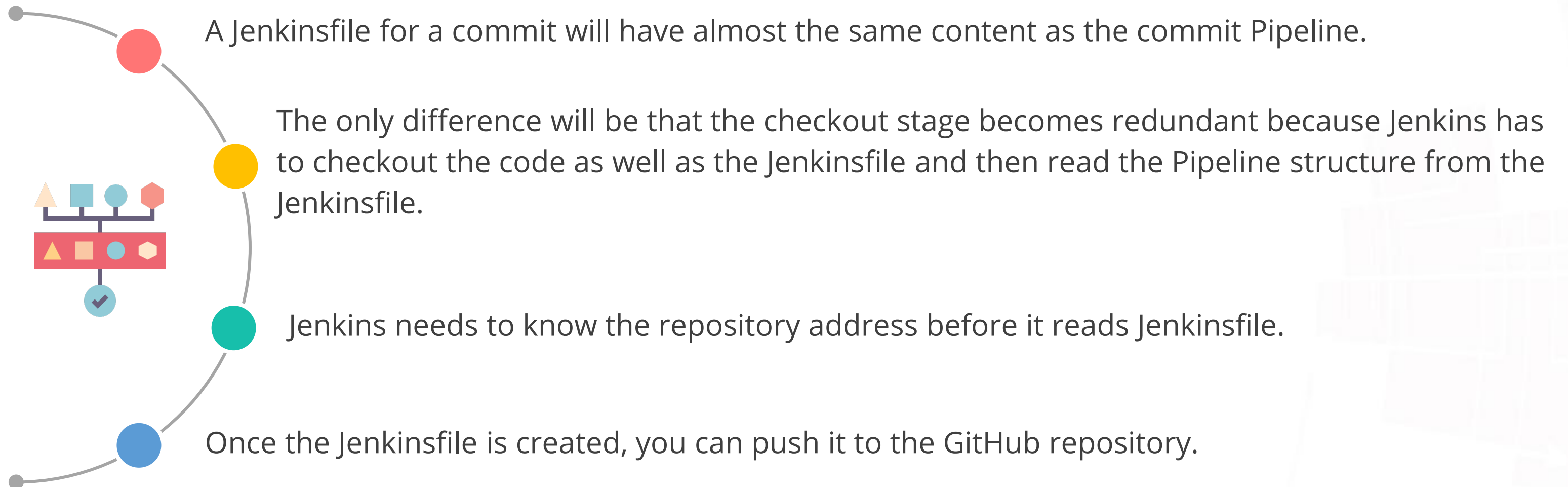
Working with Jenkins File

Jenkinsfile

- The Pipeline definition in the **Jenkinsfile** can be committed to the repository along with the source code.

- This method is more consistent as the Pipeline structure becomes strongly tailored to the project.

For example: if you don't need the code compilation because your programming language is interpreted, then you won't have the Compile stage.

Creating Jenkinsfile

A Jenkinsfile for a commit will have almost the same content as the commit Pipeline.

The only difference will be that the checkout stage becomes redundant because Jenkins has to checkout the code as well as the Jenkinsfile and then read the Pipeline structure from the Jenkinsfile.

Jenkins needs to know the repository address before it reads Jenkinsfile.

Once the Jenkinsfile is created, you can push it to the GitHub repository.

Caltech | Center for Technology & Management Education

simplilearn

Creating Jenkinsfile

Here is a sample Jenkinsfile for a commit Pipeline:

```
Pipeline {
    agent any stages {
        stage("Compile") {
            steps {
                sh "./gradlew compileJava"
            }
        }
        stage("Unit test") {
            steps {
                sh "./gradlew test"
            }
        }
    }
}
```

Running Jenkinsfile

After the Jenkinsfile is pushed, we can run the Pipeline as given below:

Open the Pipeline configuration.

Change *Definition* from *Pipeline script* to *Pipeline script from SCM*.

Select *Git* in SCM.
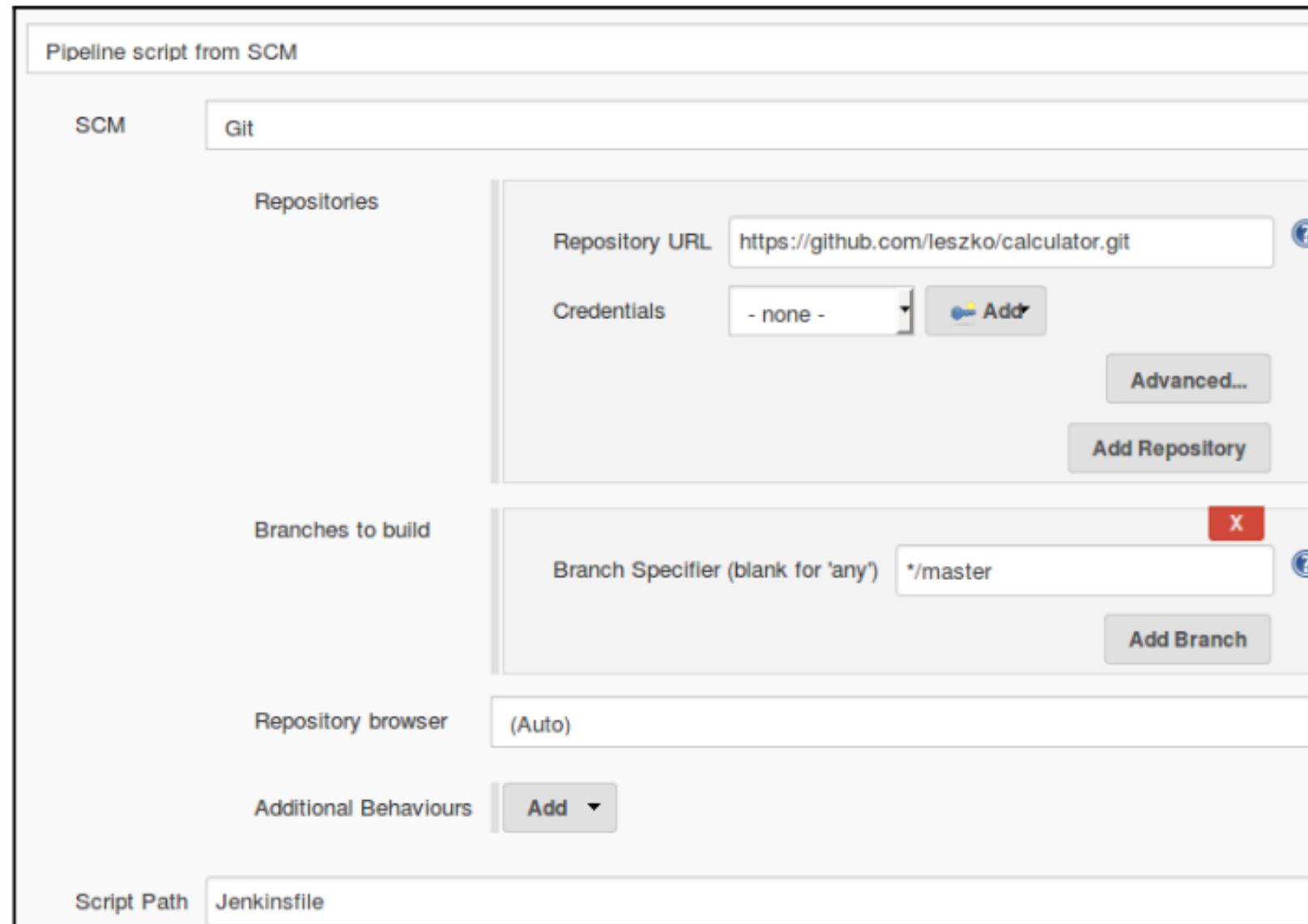
Enter Repository URL.

Save the Pipeline.

After saving, the build will always run from the current version of Jenkinsfile in the repository.

Caltech | Center for Technology & Management Education

simplilearn

Running Jenkinsfile

The screenshot below shows how to run a Pipeline from a Jenkinsfile:

Build with Jenkinsfile

The Build stage of the Pipeline will be where source code is assembled, compiled, or packaged.

The Jenkinsfile is not a replacement for an existing build tool such as GNU/Make, Maven, Gradle, etc.

It can be considered a glue layer to bind the multiple phases of a project's development lifecycle like build, test, and deploy together.

Jenkins has plugins for invoking practically any build tool.

Caltech | Center for Technology & Management Education

simplilearn

Test with Jenkinsfile

- Running automated tests is a crucial component of any successful continuous delivery process.

- Jenkins has a number of test recording, reporting, and visualization facilities provided by a number of plugins.

- The example below uses the JUnit step, provided by the JUnit plugin. If tests fail, the Pipeline is marked *unstable*.

```
Jenkinsfile (Declarative Pipeline)
pipeline {
    agent any

    stages {
        stage('Test') {
            steps {
                /* `make check` returns non-zero on test failures,
                * using `true` to allow the Pipeline to continue nonetheless
                */
                sh 'make check || true'
                junit '**/target/*.xml'
            }
        }
    }
}
```

Test with Jenkinsfile

- Deployment can be anything from publishing built artifacts to an Artifactory server, to pushing code to a production system.

- The example below shows a Deploy Pipeline. At this stage of the example Pipeline, both the *Build* and *Test* stages have been successfully executed.

```
Jenkinsfile (Declarative Pipeline)
pipeline {
        agent any

        stages {
        stage('Deploy') {
        when {
        expression {
        currentBuild.result == null || currentBuild.result == 'SUCCESS'
        }
        }
        steps {
        sh 'make publish'
        }
        }
        }
}
```

Key Takeaways

You are now able to:

- Describe the importance of continuous integration and continuous deployment

- List the features of Jenkins and demonstrate their uses

- List the features of TeamCity and demonstrate their uses

- Select a suitable build tool for your organization

Knowledge Check

What advantage does Maven have over Ant?

A.   There isn't one

B.   Ant only compiles code

C.   Maven is easier to configure

D.   It resolves dependencies

Knowledge Check

1

What advantage does Maven have over Ant?

A.   There isn't one

B.   Ant only compiles code

C.   Maven is easier to configure

D.   It resolves dependencies

The correct answer is **A**

It resolves dependencies on external jar files.

Knowledge Check

2

What advantage does continuous integration provide?

A.   It simplifies the build process

B.   It stops developers checking in bad code

C.   Build errors are quickly detected and reported

D.   There are no real advantages

Caltech | Center for Technology & Management Education

simplilearn

Knowledge Check

2

What advantage does continuous integration provide?

A. It simplifies the build process

B. It stops developers checking in bad code

C. Build errors are quickly detected and reported

D. There are no real advantages

The correct answer is **C**

Continuous integration can also perform testing and can generate documentation.

Knowledge Check

3

What does POM stand for?

A. Project Object Model

B. Project Oriented Model

C. Project Operational Model

D. Purpose Only Manufacturing

Knowledge Check

3

What does POM stand for?

A. Project Object Model

B. Project Oriented Model

C. Project Operational Model

D. Purpose Only Manufacturing

The correct answer is **A**

POM stands for Project Object Model.

Knowledge Check

4

What is continuous deployment?

A.   A deployment server

B.   Deployment tool

C.   Open source deployment server for containers

D.   Minimizing the time elapsed between writing new code and using new code in production

Knowledge Check

4

What is continuous deployment?

A.   A deployment server

B.   Deployment tool

C.   Open source deployment server for containers

D.   Minimizing the time elapsed between writing new code and using new code in production

The correct answer is   **D**

Continuous deployment refers to automated, faster, and quicker deployments of code into production.

Knowledge Check

5

Which of the following is NOT a continuous deployment tool?

A.   Microsoft Visual Studio

B.   GitHub

C.   ElectricFlow

D.   Bamboo

Knowledge Check

5

Which of the following is NOT a continuous deployment tool?

A.   Microsoft Visual Studio

B.   GitHub

C.   ElectricFlow

D.   Bamboo

The correct answer is   **B**

GitHub is not a continuous deployment tool as it is a source code repository.

Lesson-End Project

Continuous Integration with Jenkins, Git, and Maven

Problem Statement:

Create a FreeStyle project in Jenkins and complete the following:

- Install "Email Extension plugin" in Jenkins.
- Configure Gmail in Jenkins.
- Receive an email when build fails and succeeds.

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.