4.1 Learning Objectives

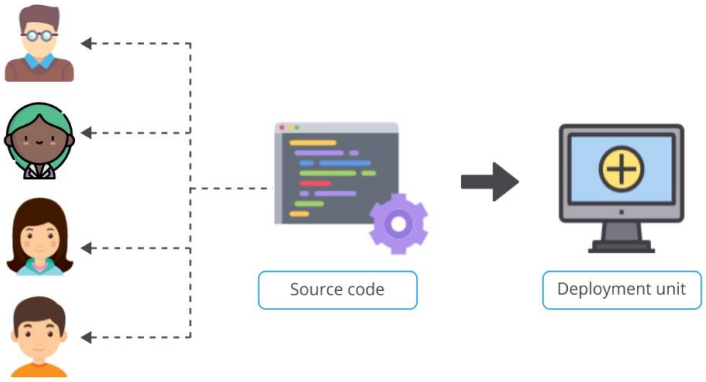
By the end of this lesson, you will be able to:

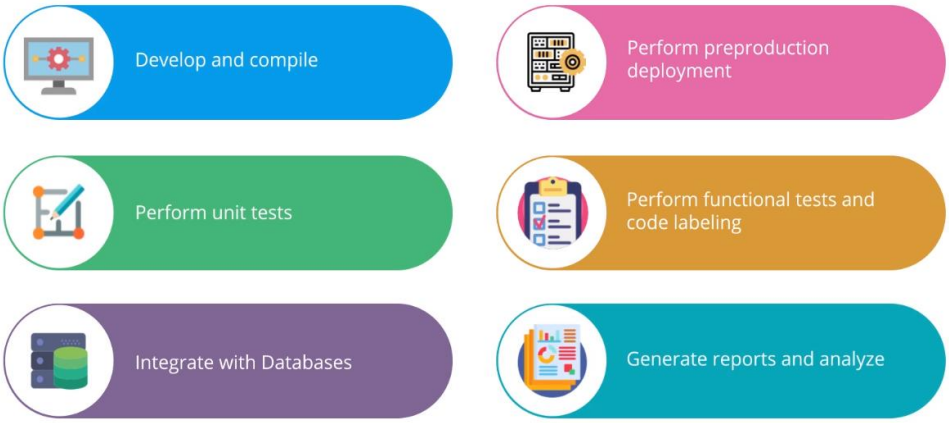
- Describe the importance of continuous integration and continuous deployment
- List the features of Jenkins and demonstrate their uses
- List the features of TeamCity and demonstrate their uses
- Select a suitable build tool for your organization

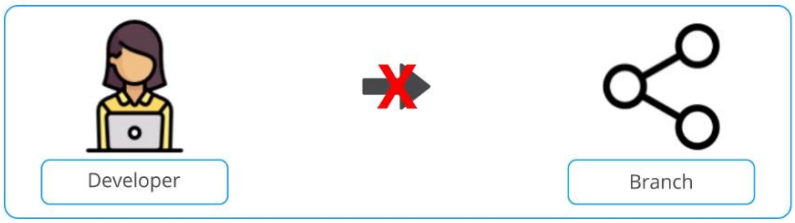
4.2 Overview and Importance of CI and CD

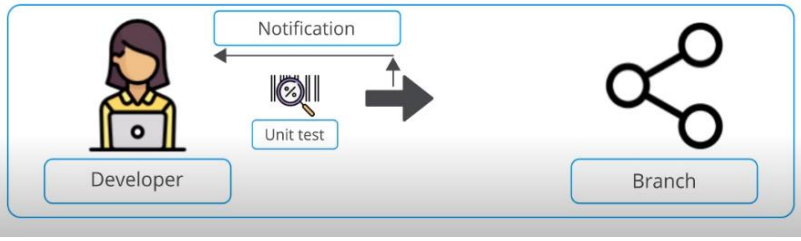
Overview of Continiuous Integration:

Continuous integration is a development practice of code integration into a shared repository. Each integration is verified by an automated build and automated tests.

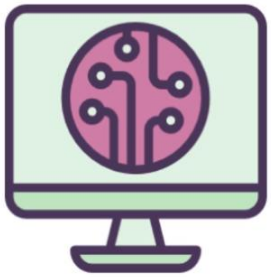




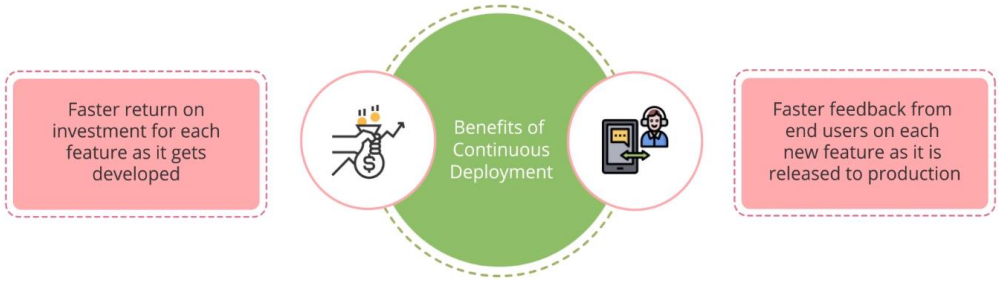


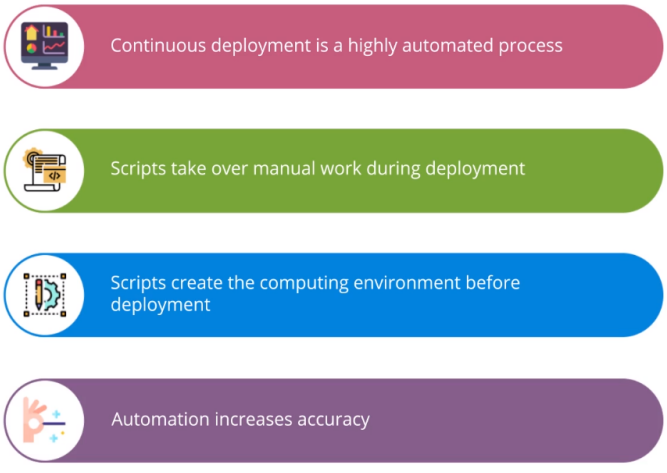


Overview of Continiuous Deployment:

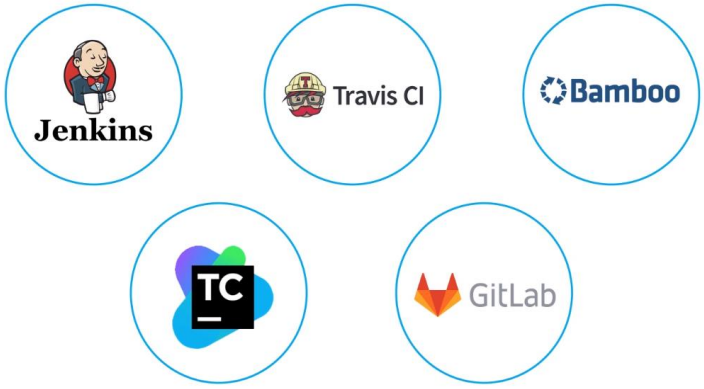


Continuous deployment is an extension of continuous integration. It aims to reduce the time the development team takes between writing one new line of code and using it in production.

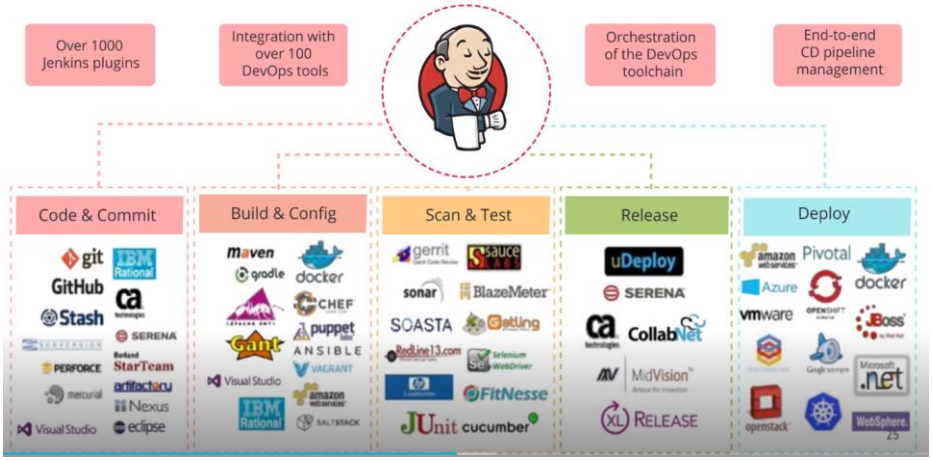




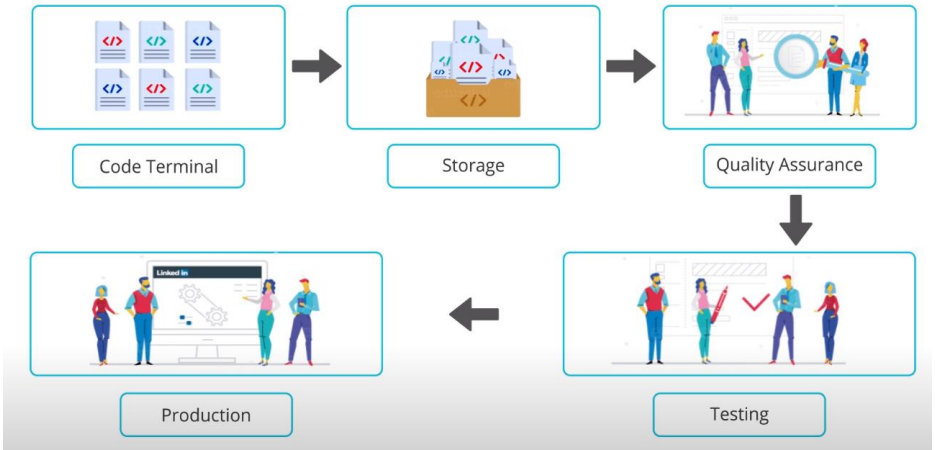
Popular Tools in Continuous Integration and Continuous Deployment:



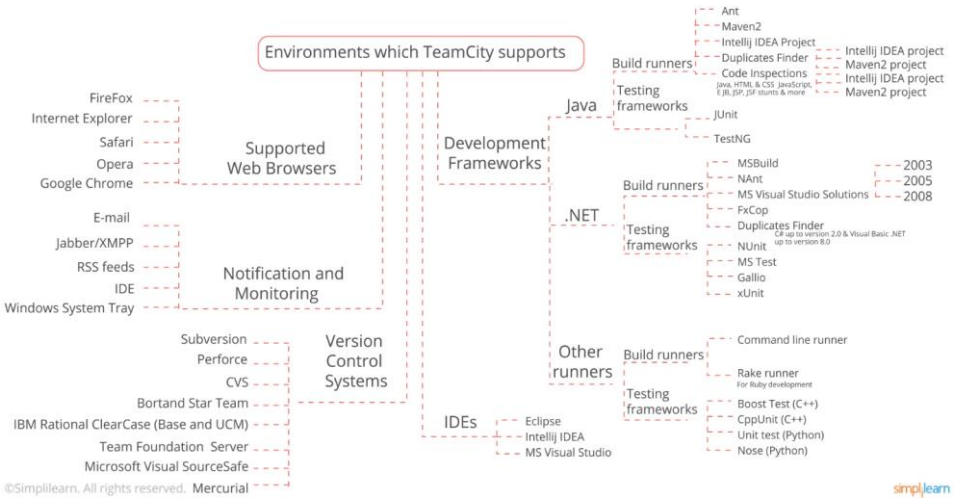
Continuous Integration with Jenkins:



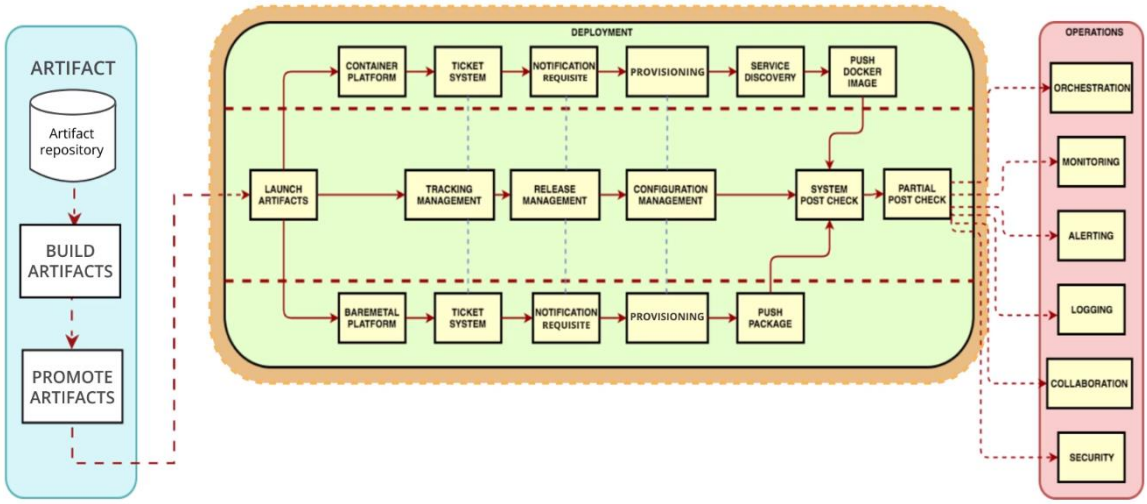
Continuous Deployment with Jenkins:



Continuous Integration with TeamCity:



Continuous Deployment with TeamCity:



4.3 Overview and Features of Jenkins

Jenkins as a Continuous Integration Tool:

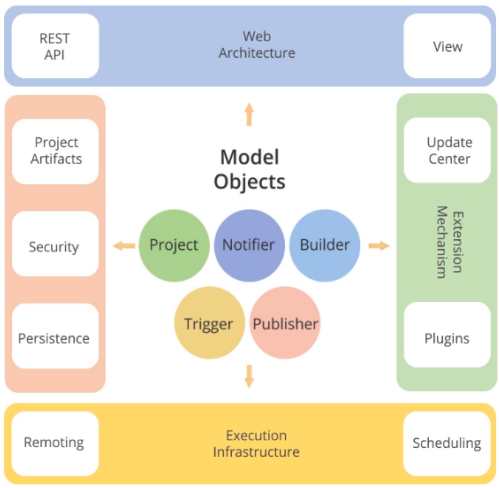
Jenkins

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.

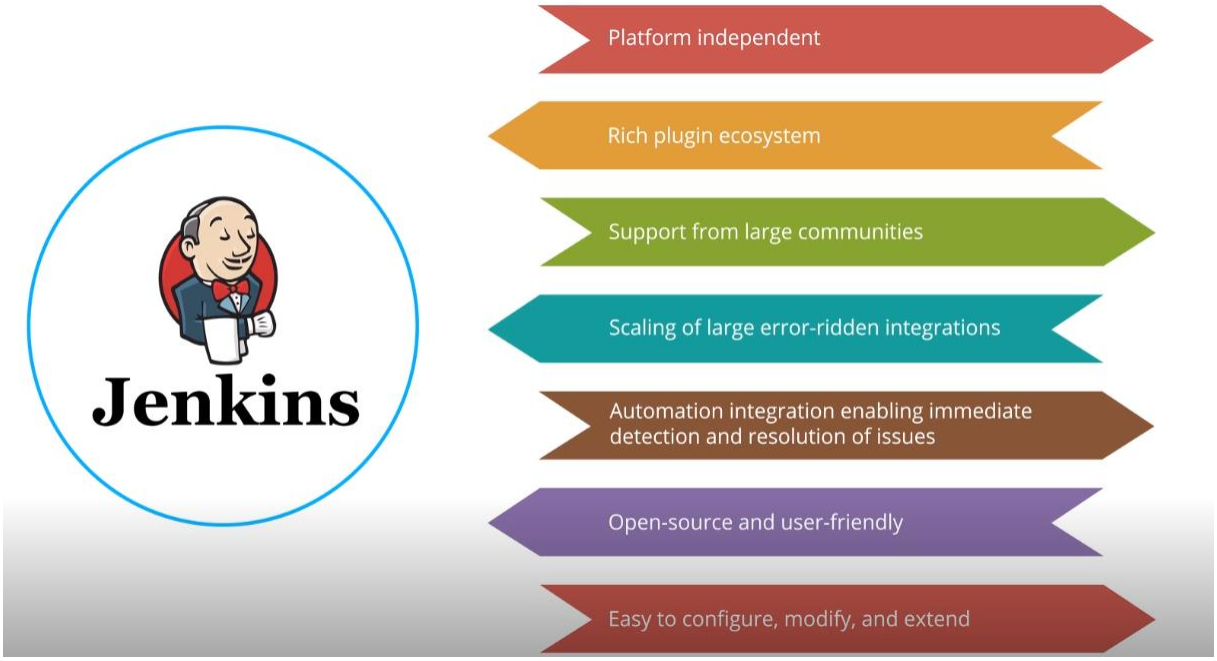
CI Server	Distribution	Cross-platform
It can be used as a CI server or as a continuous delivery hub for a project	It can easily distribute work across different machines and help trigger builds, tests, and delployments to multiple machines and platforms faster	It works on iOS, .Net, Android Devlopment, 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:



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

4.4 Set up Jenkins

Praxisbeispiel

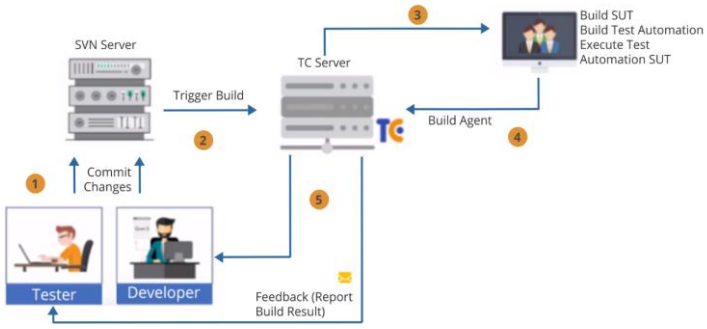
- Login to your Ubuntu Lab provided with the course.
- Open the terminal and add the key to the system.
- Edit the sources.list file, add the command to the file, and save it.
- Update the apt-get package.
- Install JDK 8+ version.
- Install Jenkins via apt-get package.
- Navigate to x.x.x.x:8080 in the browser of your virtual Machine.
- Get the password, and enter it in the Jenkins window.
- Create a new role/job in Jenkins.

4.5 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.

TeamCity Workflow:



Popular Features of TeamCity:

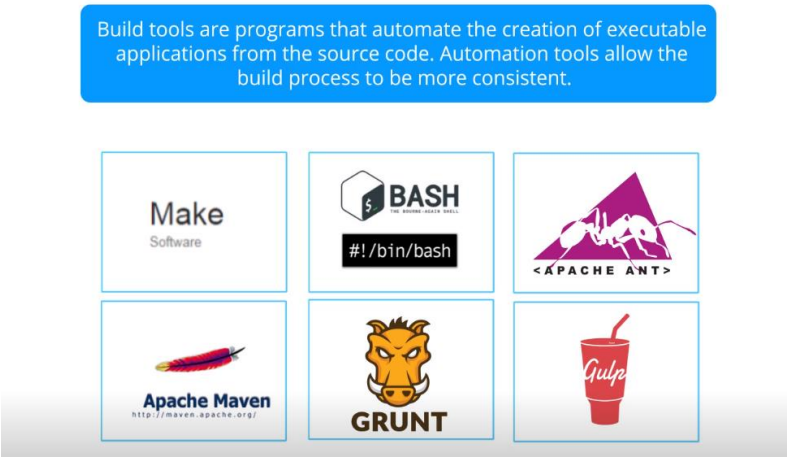
4.6. Set up TeamCity

Praxisbeispiel:

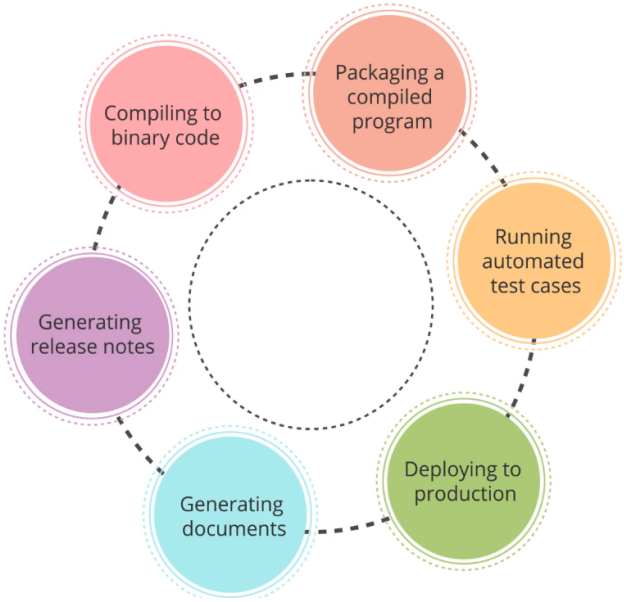
- Login to your Ubuntu Lab provided with the course.
- Download TeamCity from the official site.
- Unzip the folder, and install TeamCity.
- Provide the read, write, and execute mode access to TeamCity.
- Run TeamCity at x.x.x.x:8111 where x.x.x.x is your IP address.
- Create an account in TeamCity, and add the basic details to complete the setup process.
- Explore options such as **Projects, Changes, Agents**, and **Build Queue.**

4.7 Build Tools and Their Uses

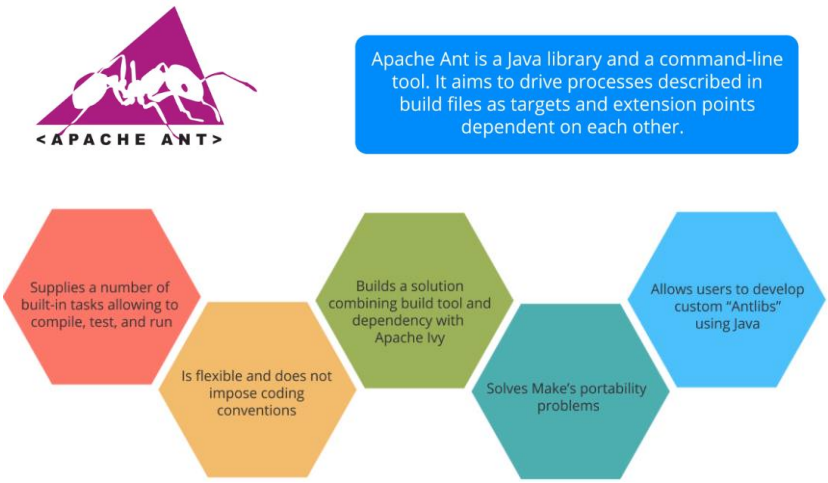
Build Tools:



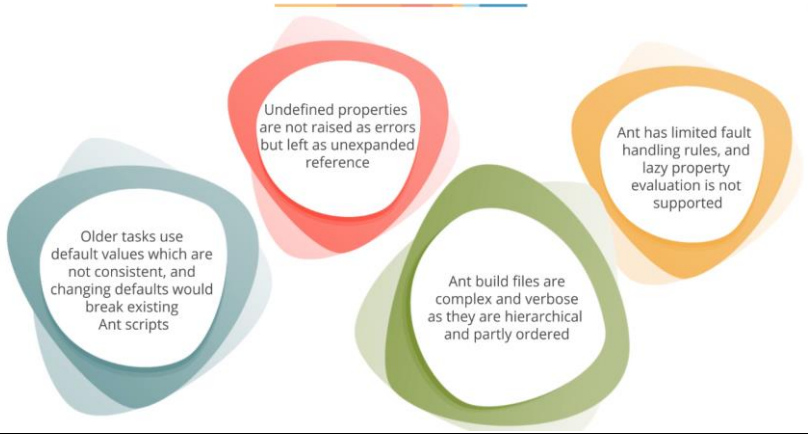
Popular Features of Build Tools:



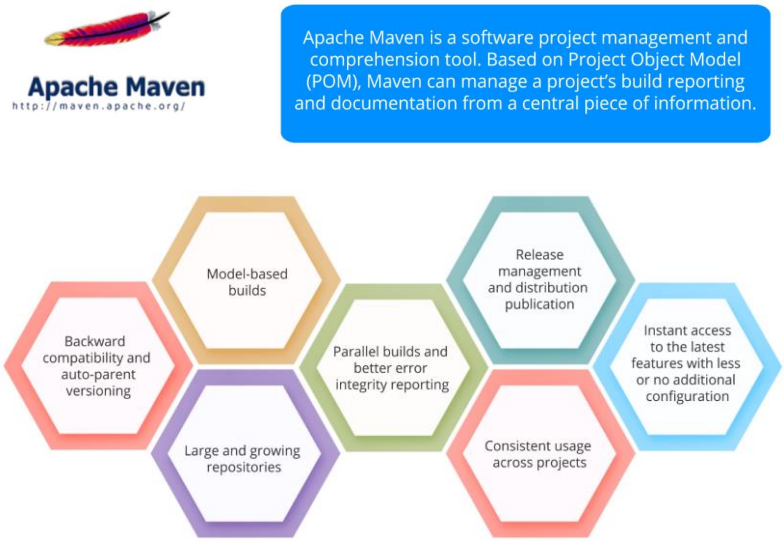
Overview of Apache Ant:



Limitations of Apache Ant:



Overview of Maven:



Drawbacks of Maven:

Maven over Ant:

Project Object Model (POM):

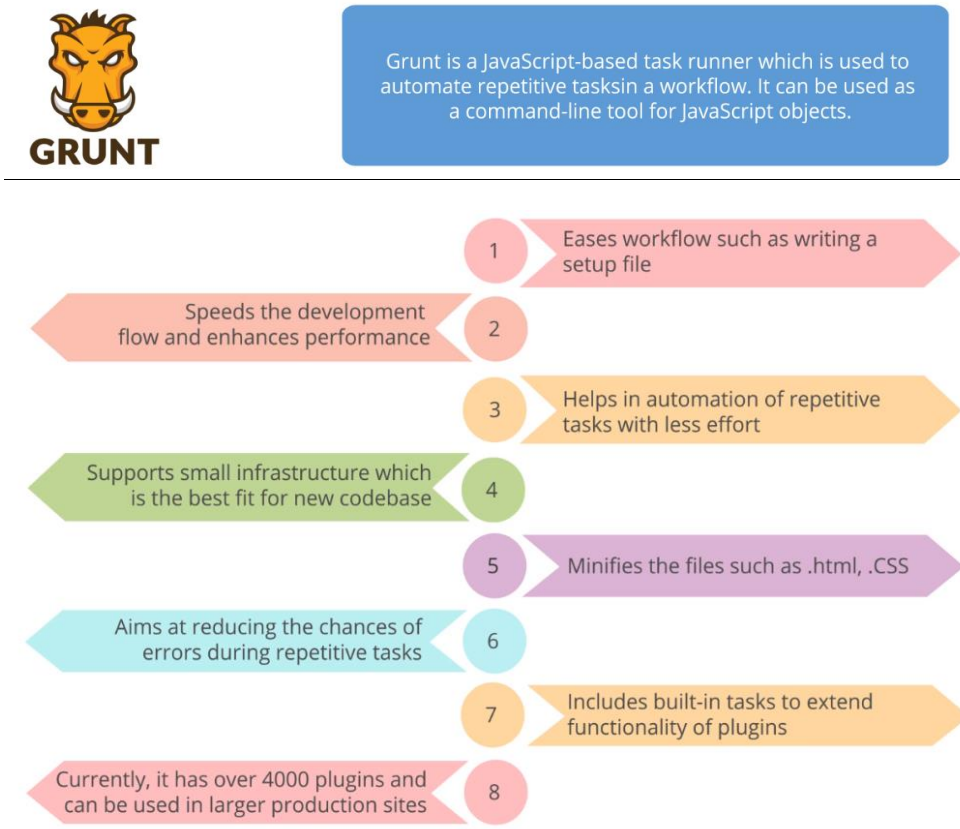
Project Object Model is an XML representation of a Maven project which provides general configuration such as a project's name, its owner, and its dependencies on other projects.

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

```
<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>
```

Overview of Grunt:



Overview of Gulp:

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

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

4.8 CI with Jenkins and Maven

Praxisbeispiel

- Login to your Ubuntu Lab provided with the course.
- Login to Jenkins and create the first Jenkins job.
- Install and configure Maven.
- Configure Jenkins with Java, Git, and Maven.
- Create a Jenkins job for your Maven build project, and run the project.
- Poll Git for commits and automatically trigger the build.
- Build the trigger using Push mechanism instead of Pull.
- Repeat steps 6 and 7 multiple times to observe the results at console output section.

4.9 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