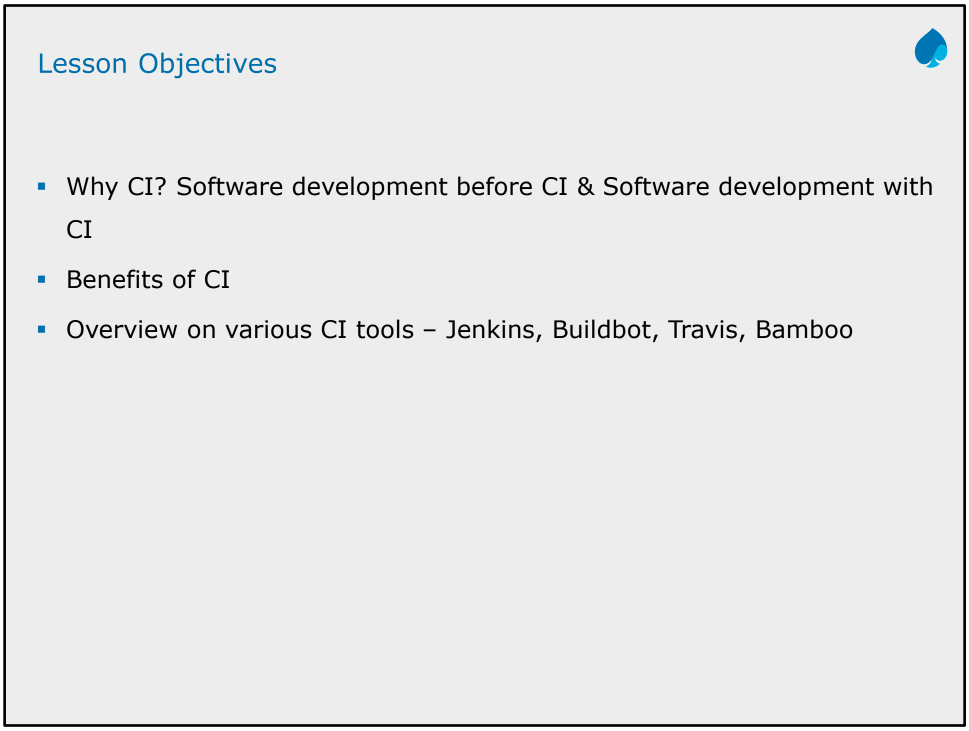
Git & Jenkin

Lesson 1: Introduction to CI

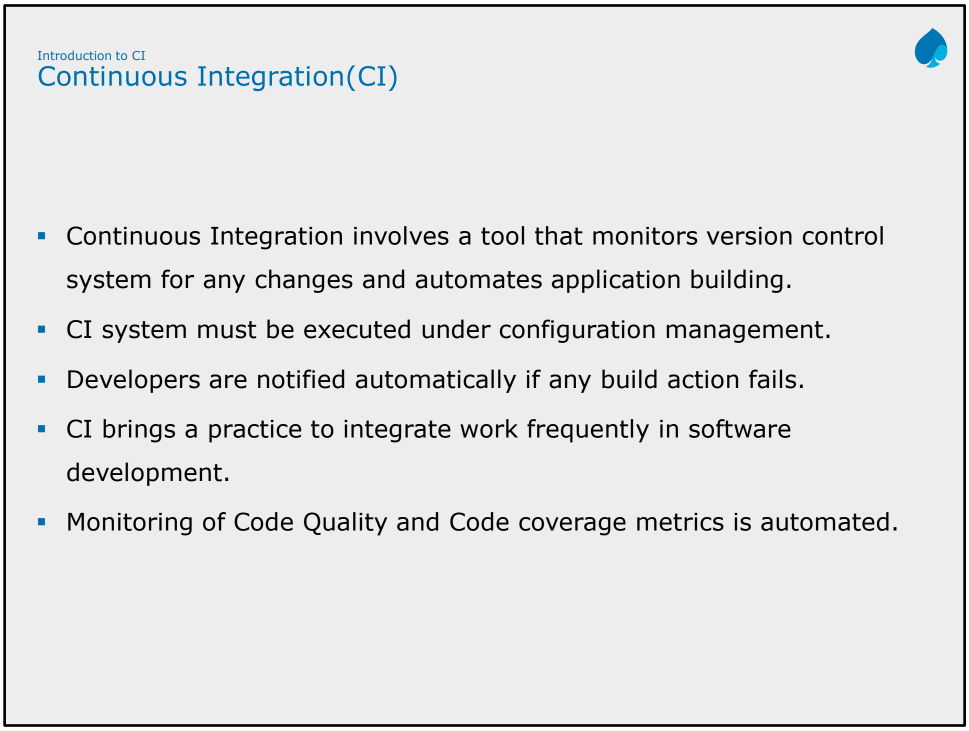
Add instructor notes here.



Lesson Objectives

- Why CI? Software development before CI & Software development with CI
- Benefits of CI
- Overview on various CI tools – Jenkins, Buildbot, Travis, Bamboo

Add instructor notes here.



Introduction to CI

Continuous Integration(CI)

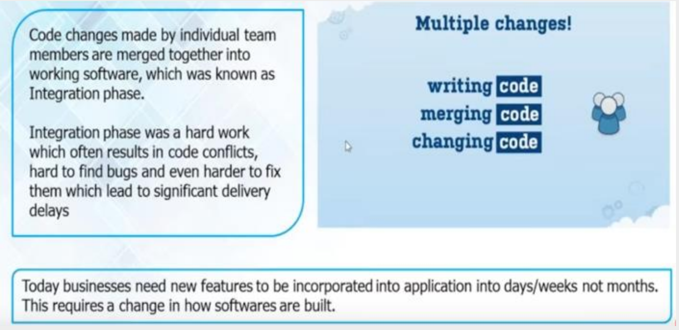
- Continuous Integration involves a tool that monitors version control system for any changes and automates application building.
- CI system must be executed under configuration management.
- Developers are notified automatically if any build action fails.
- CI brings a practice to integrate work frequently in software development.
- Monitoring of Code Quality and Code coverage metrics is automated.

Add instructor notes here.

Introduction to CI

Why CI?

Software Development Before CI



Code changes made by individual team members are merged together into working software, which was known as Integration phase.

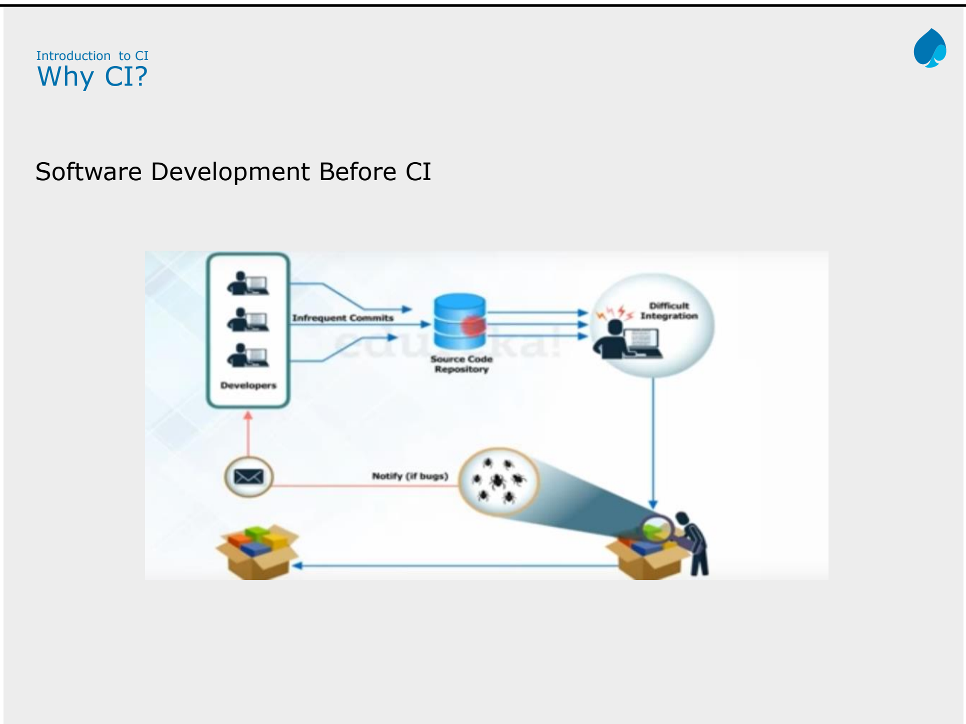
Integration phase was a hard work which often results in code conflicts, hard to find bugs and even harder to fix them which lead to significant delivery delays

Multiple changes!

writing code
merging code
changing code

Today businesses need new features to be incorporated into application into days/weeks not months. This requires a change in how softwares are built.

Add instructor notes here.

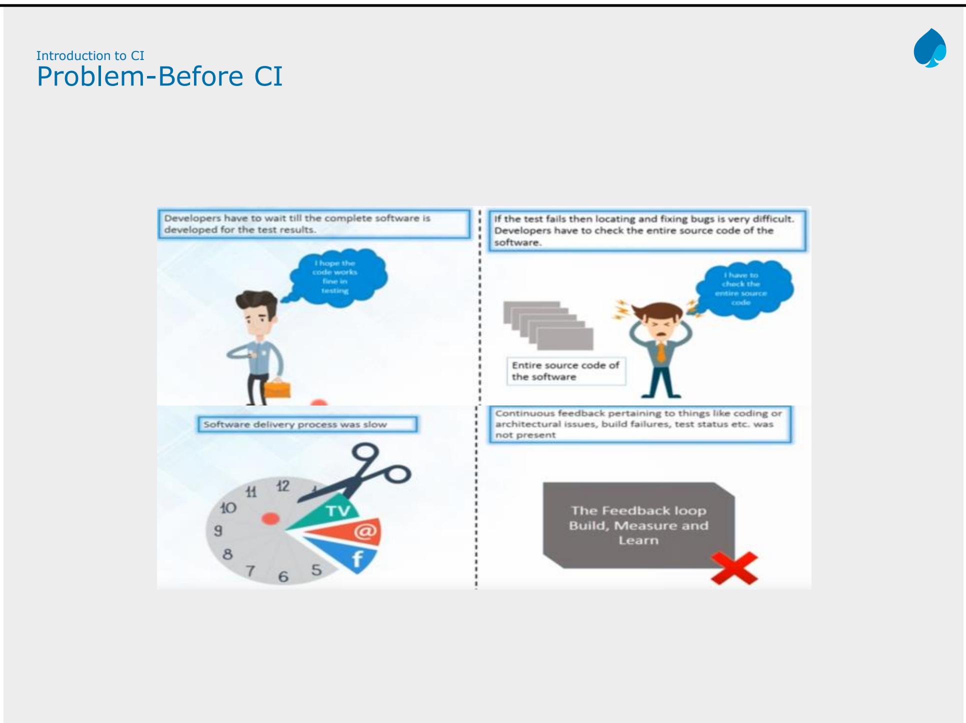


Introduction to CI

Why CI?

Software Development Before CI

Add instructor notes here.



Introduction to CI

Problem-Before CI

Developers have to wait till the complete software is developed for the test results.

I hope the code works fine in testing

If the test fails then locating and fixing bugs is very difficult. Developers have to check the entire source code of the software.

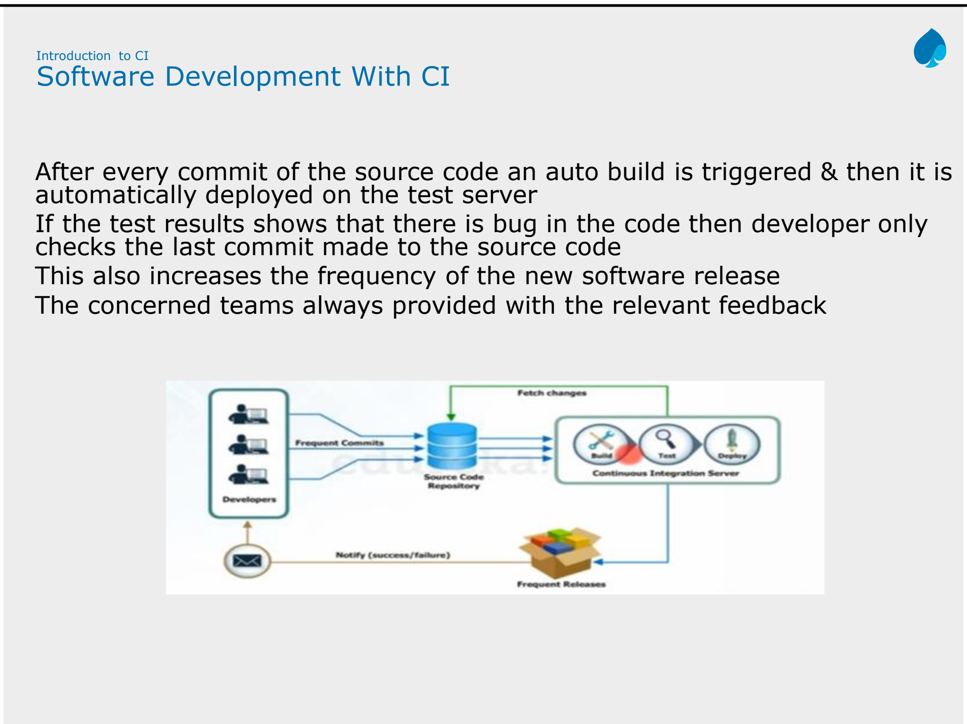
Entire source code of the software
I have to check the entire source code

Software delivery process was slow

12
11
10
9
8
7
6
5
TV
@
f

Continuous feedback pertaining to things like coding or architectural issues, build failures, test status etc. was not present

Add instructor notes here.



Introduction to CI

Software Development With CI

After every commit of the source code an auto build is triggered & then it is automatically deployed on the test server

If the test results shows that there is bug in the code then developer only checks the last commit made to the source code

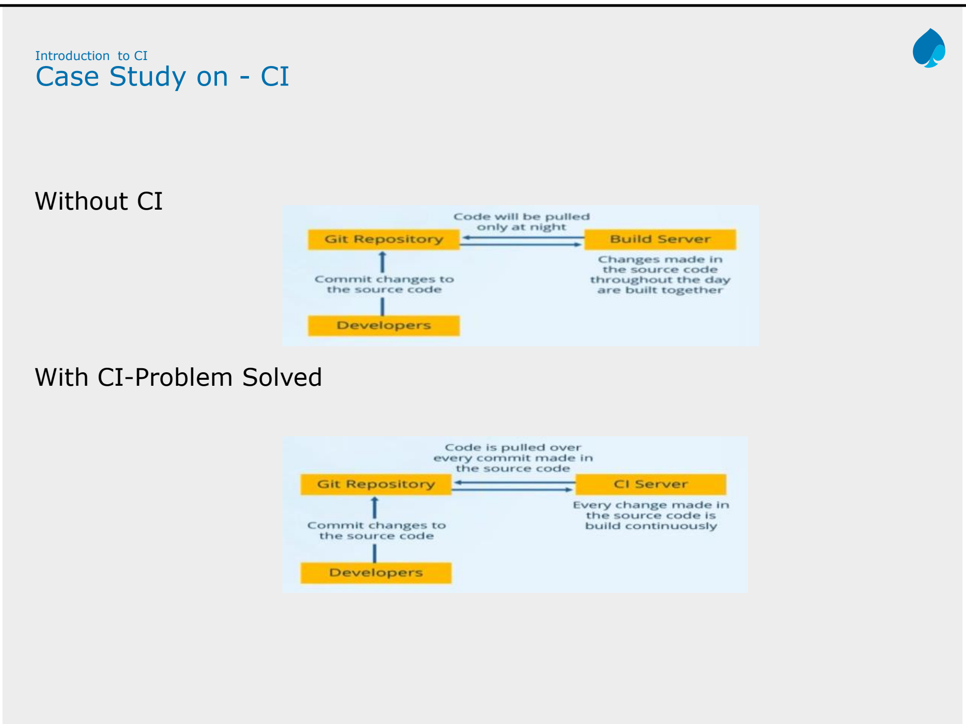
This also increases the frequency of the new software release

The concerned teams always provided with the relevant feedback

1: Application must run under source control management
2: Daily code commits to SVN will be baseline.
3: CI polls for any code changes in SVN and triggers build actions if any.
4: Automated build, testing and deployment of an application will be performed by CI.
5: After build action, developer will be accessible with latest code and build
6: Developers will be notified with any build errors and automated test results.

Need of CI in software Development

Helps to locate code based defects in a centralized location.
Tools can be used to automate deployment.
Minimizes integration errors in SVN during build process(Errors are uncovered during Manual Build) by invoking automation.
Increase amount of quality code and improve development standards.



Introduction to CI

Case Study on - CI

Without CI

Code will be pulled only at night

Git Repository

Build Server

Commit changes to the source code

Changes made in the source code throughout the day are built together

Developers

With CI-Problem Solved

Code is pulled over every commit made in the source code

Git Repository

CI Server

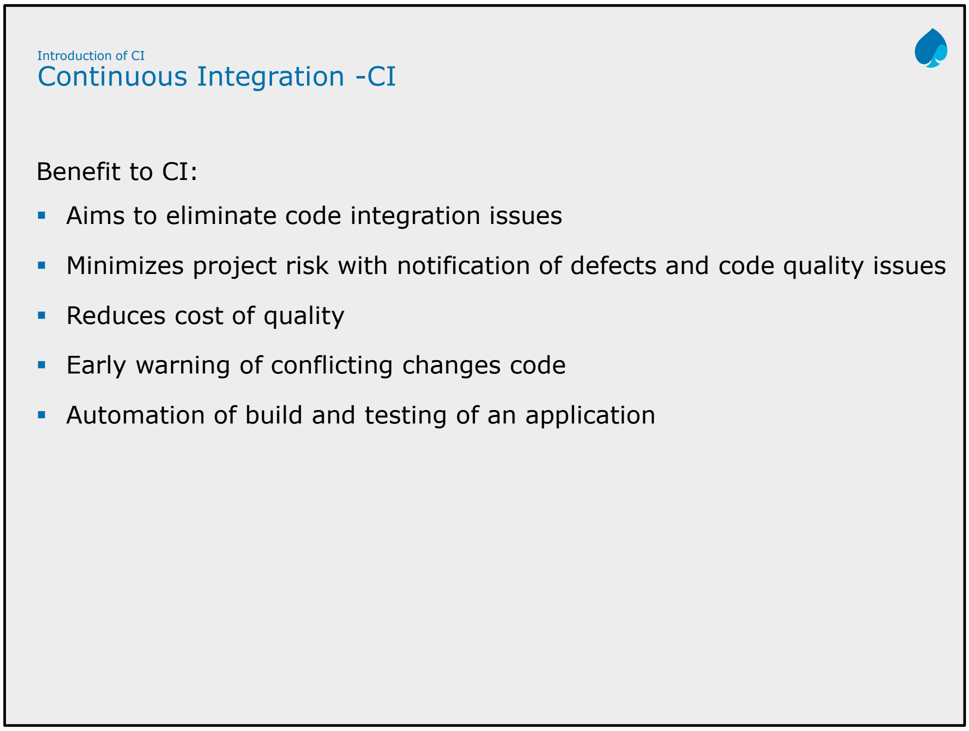
Commit changes to the source code

Every change made in the source code is build continuously

Developers

Tell about scenario if user is doing with or without CI .Give example of Nokia uploading

Add instructor notes here.



Introduction of CI

Continuous Integration -CI

Benefit to CI:

- Aims to eliminate code integration issues
- Minimizes project risk with notification of defects and code quality issues
- Reduces cost of quality
- Early warning of conflicting changes code
- Automation of build and testing of an application

Reduce risks
By integrating many times a day, you can reduce risks on your project. Doing so facilitates the detection of defects, the measurement of software health and a reduction of assumptions.

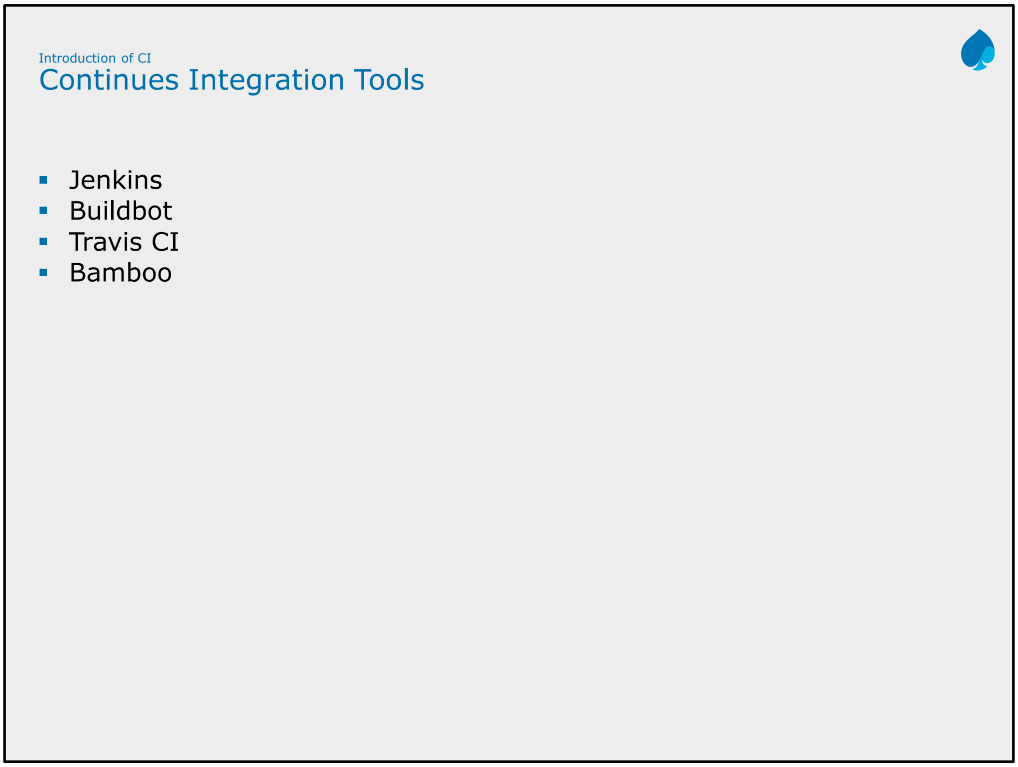
Defects are detected and fixed sooner: Because CI integrates and runs tests and inspections several times a day, there is a greater chance that defects are discovered *when they are introduced* (i.e., when the code is checked into the version-control repository) instead of during late-cycle testing.

Health of software is measurable: By incorporating continuous testing and inspection into the automated integration process, the software product's health attributes, such as complexity, can be tracked over time.

Reduce assumptions: By rebuilding and testing software in a clean environment using the same process and scripts on a continual basis, you can reduce assumptions (e.g., whether you are accounting for third-party libraries or environment variables).

CI provides a safety net to reduce the risk that defects will be introduced into the code base. The following are some of the risks that CI helps to mitigate. We discuss these and other risks in the next chapter.

Lack of cohesive, deployable software
Late defect discovery
Low-quality software
Lack of project visibility

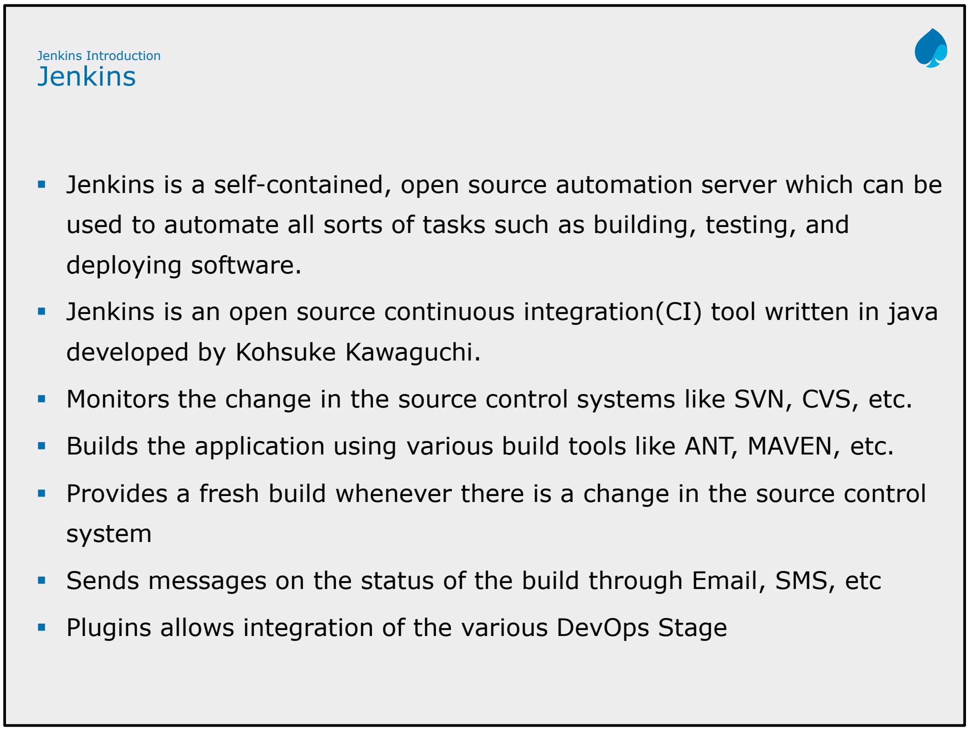


Introduction of CI

Continues Integration Tools

- Jenkins
- Buildbot
- Travis CI
- Bamboo

Add instructor notes here.

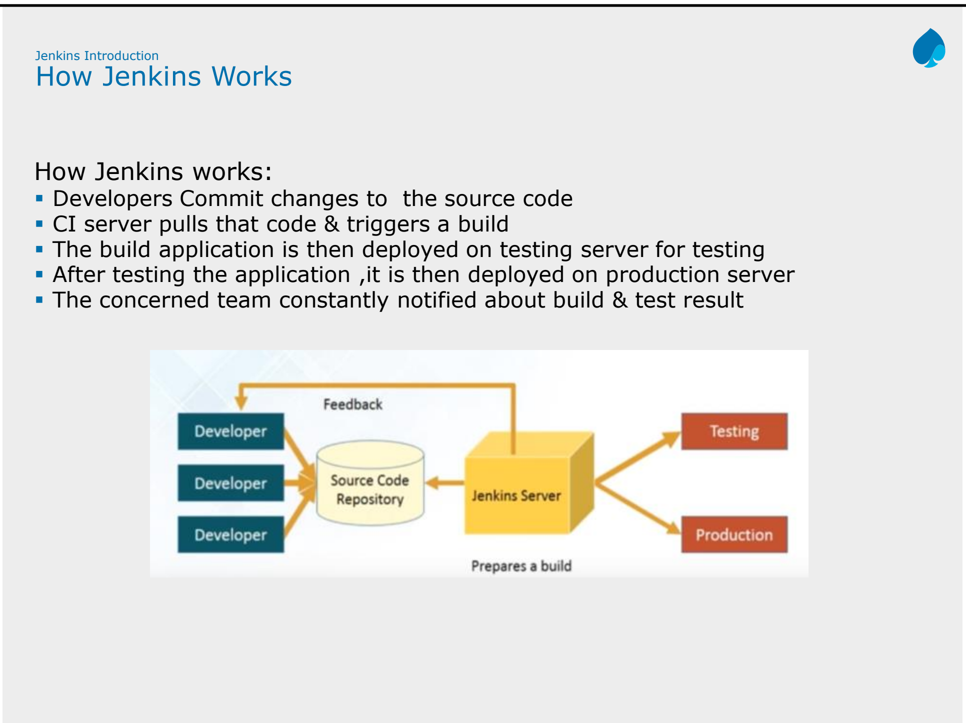


Jenkins Introduction

Jenkins

- Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software.
- Jenkins is an open source continuous integration(CI) tool written in java developed by Kohsuke Kawaguchi.
- Monitors the change in the source control systems like SVN, CVS, etc.
- Builds the application using various build tools like ANT, MAVEN, etc.
- Provides a fresh build whenever there is a change in the source control system
- Sends messages on the status of the build through Email, SMS, etc
- Plugins allows integration of the various DevOps Stage

Add instructor notes here.



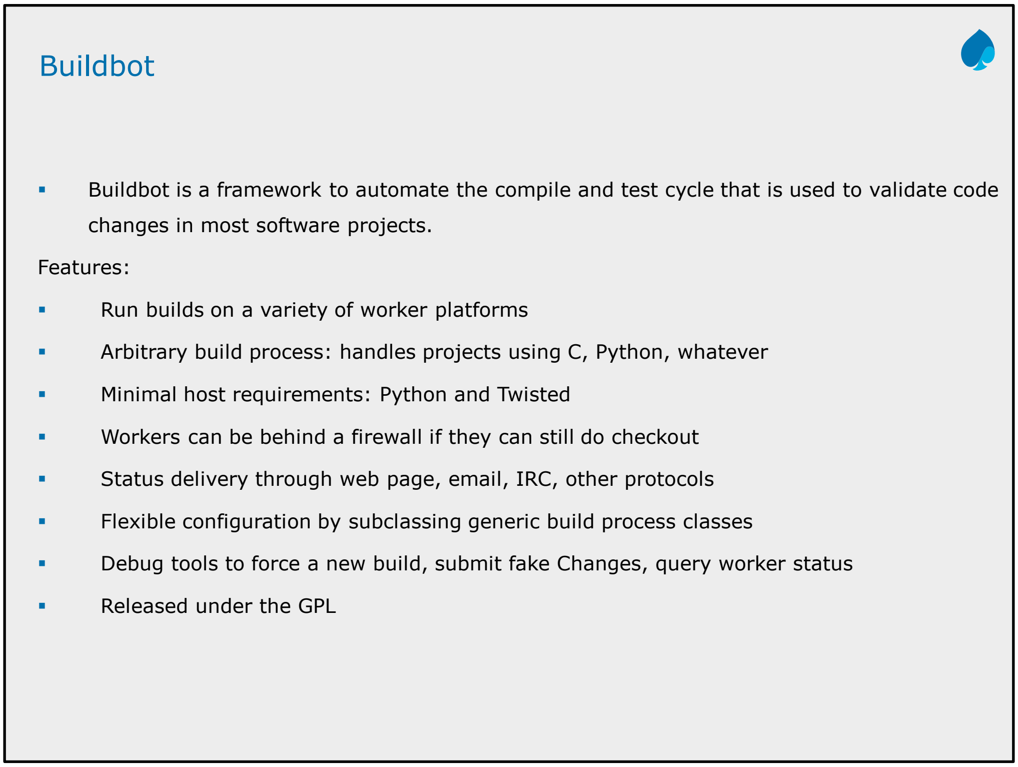
Jenkins Introduction

How Jenkins Works

How Jenkins works:

- Developers Commit changes to the source code
- CI server pulls that code & triggers a build
- The build application is then deployed on testing server for testing
- After testing the application ,it is then deployed on production server
- The concerned team constantly notified about build & test result

Feedback
Developer
Developer
Developer
Source Code Repository
Jenkins Server
Testing
Production
Prepares a build

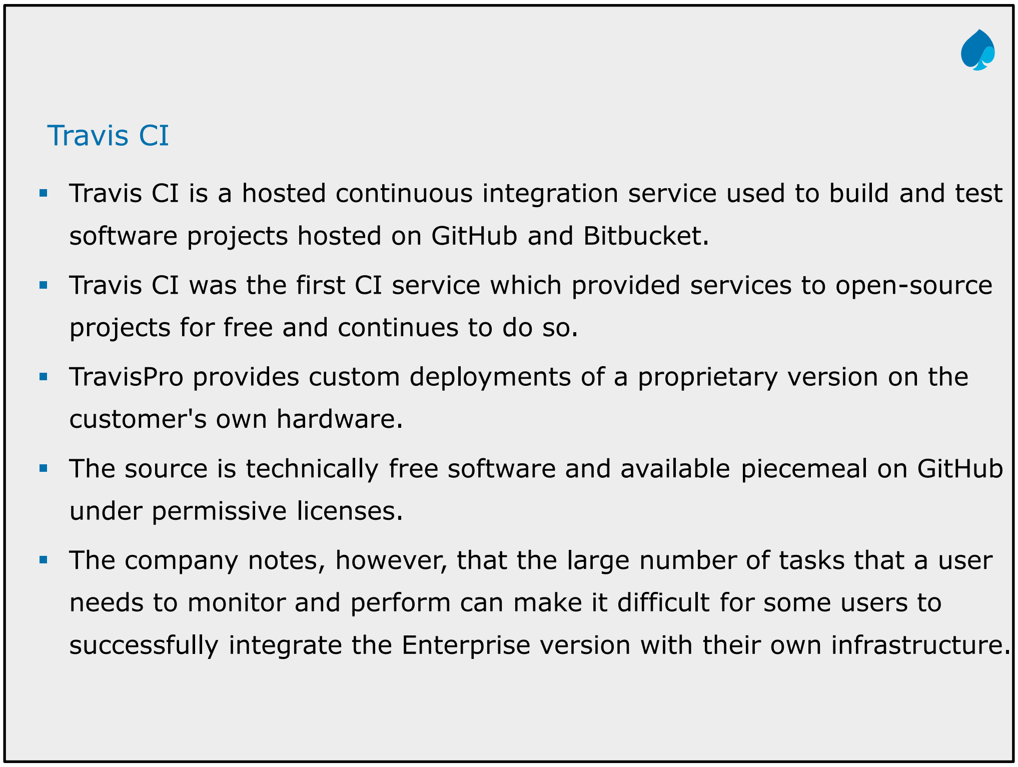


Buildbot

- Buildbot is a framework to automate the compile and test cycle that is used to validate code changes in most software projects.

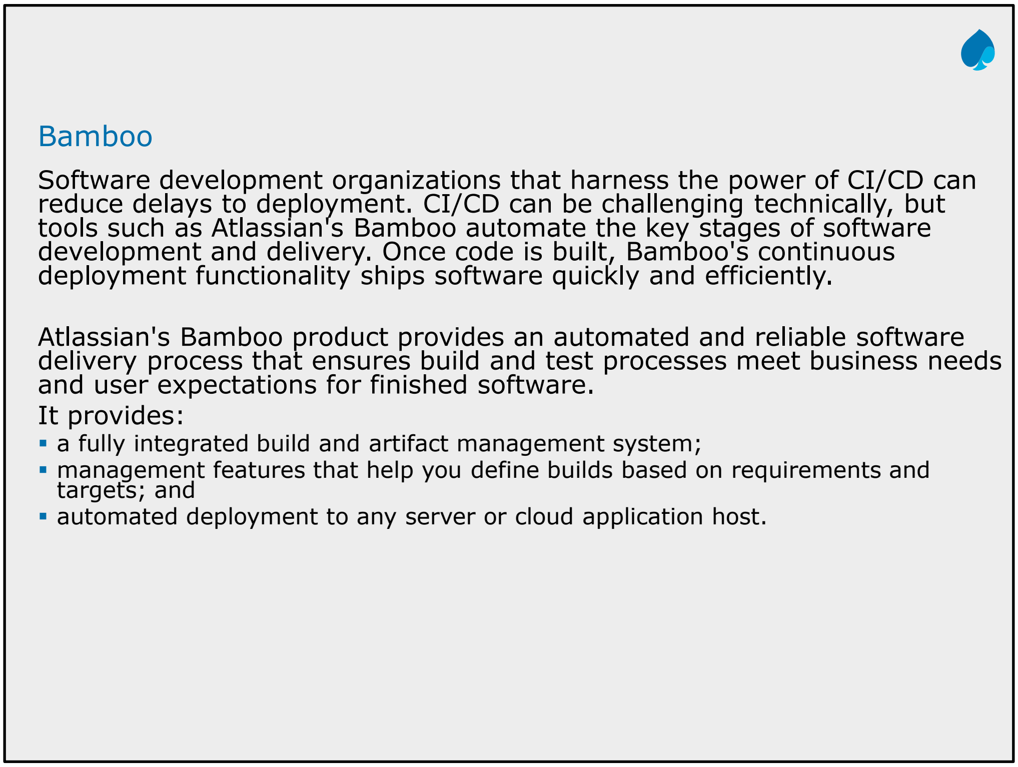
Features:

- Run builds on a variety of worker platforms
- Arbitrary build process: handles projects using C, Python, whatever
- Minimal host requirements: Python and Twisted
- Workers can be behind a firewall if they can still do checkout
- Status delivery through web page, email, IRC, other protocols
- Flexible configuration by subclassing generic build process classes
- Debug tools to force a new build, submit fake Changes, query worker status
- Released under the GPL



Travis CI

- Travis CI is a hosted continuous integration service used to build and test software projects hosted on GitHub and Bitbucket.
- Travis CI was the first CI service which provided services to open-source projects for free and continues to do so.
- TravisPro provides custom deployments of a proprietary version on the customer's own hardware.
- The source is technically free software and available piecemeal on GitHub under permissive licenses.
- The company notes, however, that the large number of tasks that a user needs to monitor and perform can make it difficult for some users to successfully integrate the Enterprise version with their own infrastructure.



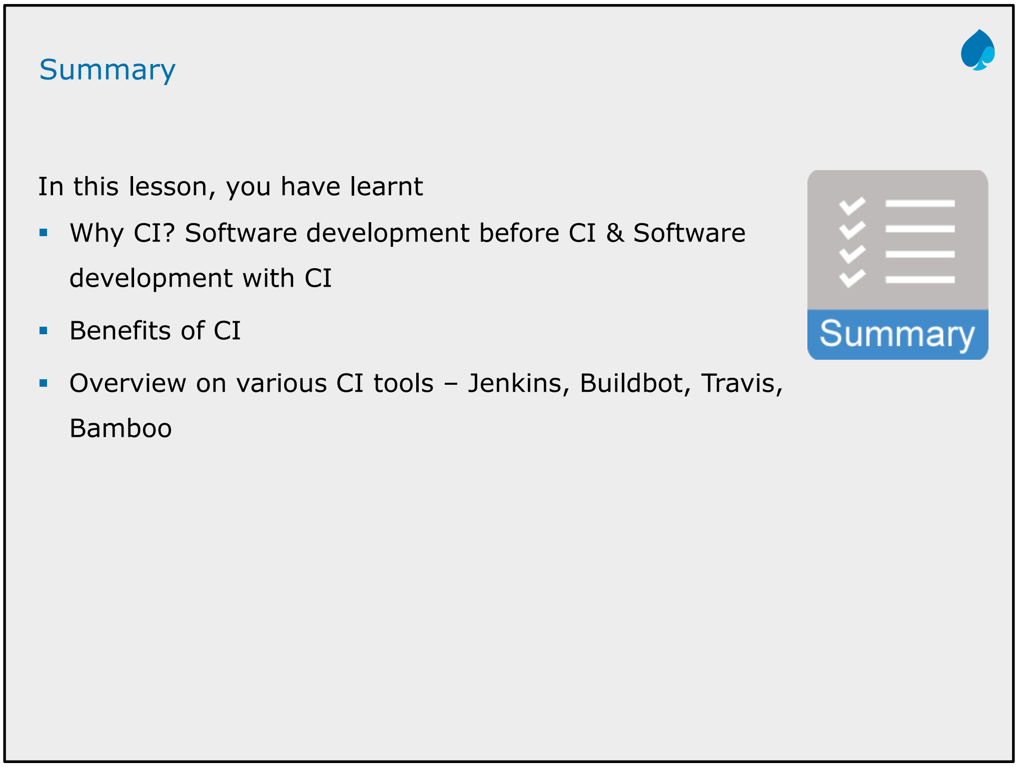
Bamboo

Software development organizations that harness the power of CI/CD can reduce delays to deployment. CI/CD can be challenging technically, but tools such as Atlassian's Bamboo automate the key stages of software development and delivery. Once code is built, Bamboo's continuous deployment functionality ships software quickly and efficiently.

Atlassian's Bamboo product provides an automated and reliable software delivery process that ensures build and test processes meet business needs and user expectations for finished software.

It provides:

- a fully integrated build and artifact management system;
- management features that help you define builds based on requirements and targets; and
- automated deployment to any server or cloud application host.



Summary

In this lesson, you have learnt

- Why CI? Software development before CI & Software development with CI
- Benefits of CI
- Overview on various CI tools – Jenkins, Buildbot, Travis, Bamboo

Add the notes here.

Review Question

Question 1:

What steps are in Continuous Integration?Increase test coverage?

- Compilation
- Unit Tests
- Code Quality Gates
- All of these

What are the benefits of Continuous Integration?

- Errors detected late
- BIncreases bug accumulation
- Setting the stage for Continuous Delivery
- All of the above