

In this post, we have covered the five most asked Selenium questions and their answers in great detail. Many of our readers had requested us to write a dedicated post to address these important concepts.

Since these Selenium questions are quite important from the interview perspective, so we adopted a practical approach while answering each of them. We tried to bring every minute detail that is important for you and could be crucial during the interview.

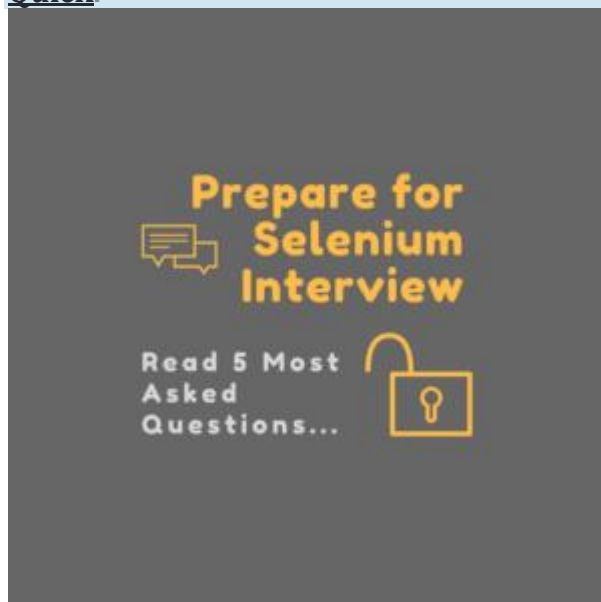
You can read these five questions by clicking on the hyperlinks given below.

Top Five Selenium Questions For Interview.

- [What types of data have you handled in Selenium for automating web applications?](#)
- [What are various build and deploy tools apart from Maven used in the industry?](#)
- [What is WebDriverIO? Which technology does it use? Why should QA use it for UI automation?](#)
- [What are the different network protocols that Selenium supports? Also, how will you handle them in Selenium?](#)
- [What is Continuous Integration \(CI\), what are its benefits? What are different CI tools available in the Industry?](#)

Also, if you like to revise your core Selenium knowledge and want to prepare thoroughly, then you should consider reading the below tutorial.

→ [**Recommended - 35 Selenium Interview Questions for Beginners to Start Quick.**](#)

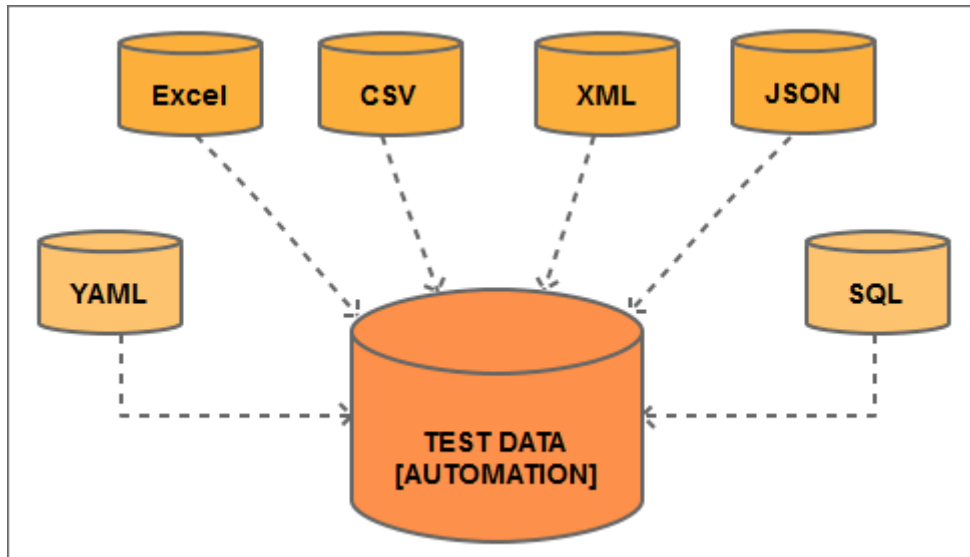


1) What Types Of Data Have You Handled In Selenium For Automating Web Applications?

Answer.

Data is probably a very critical and important piece of information in ensuring that Automation can run successfully. And it can be in a variety of form factors to serve different purposes. For example, in automation, we use input data that includes test cases, and it could be test data that feeds into test cases as parameters.

Here are some of the commonly used mechanisms to handle data in automation.



1. Excel Data.

It is the most popular data format used widely by automation testers. We can not only use it to store test cases but to keep test data as well.

This format is very flexible and supports CRUD (Create /Delete /Update /Delete) operations also. And almost all languages that integrate with Selenium like Java, Python, Ruby have ready-made libraries available to access and manipulate Excel data.

2. CSV Data.

CSV stands for comma-separated values. It is the most simplistic way of organizing test data in tabular format. It represents a sequence of rows and columns (delimited by commas) stored in a text file with CSV as the file extension. Each row corresponds to a record whereas every column maps to a field.

It is trivial to access CSV data as most programming languages like Java and Python come with standard libraries to read/write text files. However, Excel does support this type of data and can handle it the same way as it does its own.

3. XML Data.

XML is an acronym for Extensible Markup Language. It generates documents in a format which is readable by both human and machine. XML is a web standard and tends to be faster than Excel when it comes to read/write data. And there are either built-in or third-party XML parsers available in Java, Python, and others to manage the data.

```
<testsuite name="TestSuite">
  <test id="1" desc="Create a new document" >
    <step value="New" onclick="CreateNew()" />
    <step value="Open" onclick="Open()" />
    <step value="Close" onclick="Close()" />
  </test>
</testsuite>
```

Sometimes both Excel and XML get used together in Selenium automation projects. For example, we can write the test cases in Excel and create a macro to convert them in XML format. The automation suite then reads the test cases from XML and executes them.

Also, XML comes with another unique feature to search specific elements called as XPath. It is a known Selenium term which we usually use to traverse the DOM of a web page. However, it works the same with any XML data or document.

4. JSON Data.

JSON is an acronym for JavaScript Object Notation. It supports a light-weight data interchange format. Since JSON data is compact and fast, it has become a preferred choice for supplying input in web services.

And we can also use it to represent the test data replacing Excel and XML formats. If the size of data is invariably large, then JSON would surely deliver better performance.

```
{
  "testsuite": {
    "test": {
      "step": [
        { "_value": "New", "_onclick": "CreateNew()" },
        { "_value": "Open", "_onclick": "Open()" },
        { "_value": "Close", "_onclick": "Close()" }
      ],
      "_id": "1",
      "_desc": "Create a new document"
    },
    "_name": "TestSuite"
  }
```

```
}  
}
```

5. YAML Data.

YAML is another human-readable language for data serialization. It stands for YAML Ain't Markup Language. This type of data format is ideal for holding configurations. However, it could be useful for various other purposes. It is a superset of JSON language and can store both JSON and XML data within itself.

Unlike other data formats, it allows referencing other elements inside YAML using anchors.

```
testsuite:  
  test:  
    step:  
      -  
        _value: New  
        _onclick: "CreateNew() "  
      -  
        _value: Open  
        _onclick: "Open() "  
      -  
        _value: Close  
        _onclick: "Close() "  
    _id: 1  
    _desc: "Create a new document"  
    _name: TestSuite
```

6. SQL Data.

Another type of data which doesn't get used often is SQL data. It is highly customizable and enables fast data access. It can store both the test cases and any configuration or settings as required.

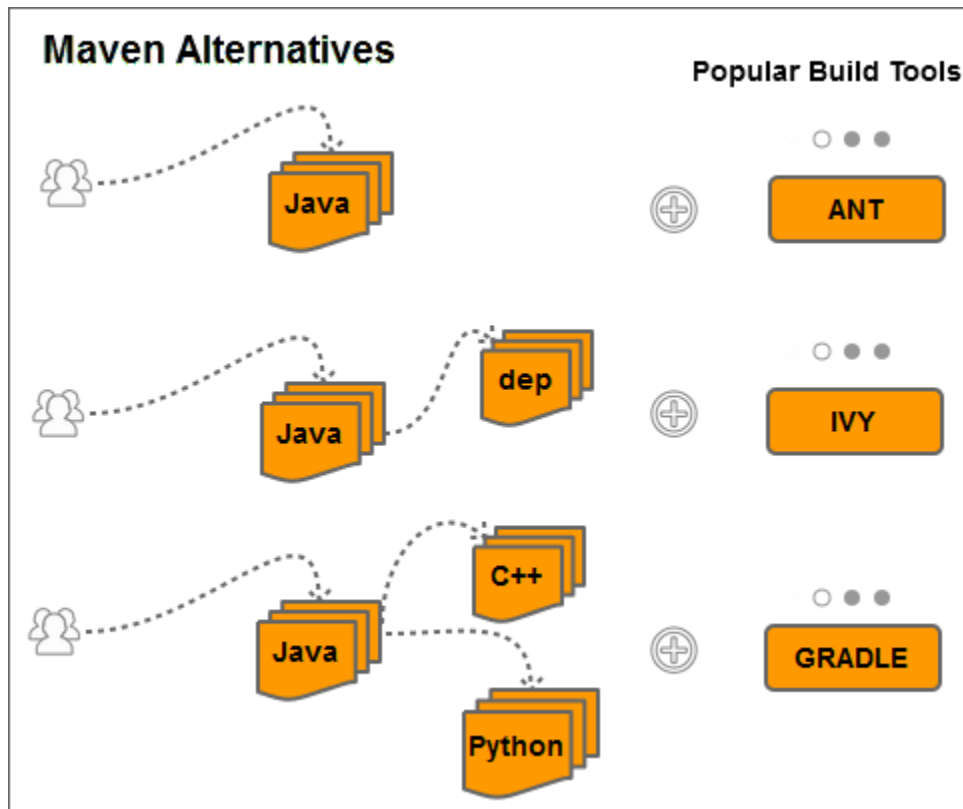
However, there is a separate database software like MySQL/MongoDB/SQL/Oracle needed to make it work. It is an additional overhead for the automation engineers to install the database software to prepare a new setup.

2) What Are Various Build And Deploy Tools Apart From Maven Used In The Industry?

Answer.

Use of Maven is wide-spread not just because it builds code or manages dependencies on the fly but also due to its ability for supporting deployments. So it eases up a lot of post development tasks. However, we can't ignore the challenges that show up while working with complex commands and XML configuration.

Hence, here are a bunch of other tools that are worth trying.



Ant.

ANT is an acronym for Another Neat Tool. It is a command-line tool developed by James Duncan Davidson and written in Java. It makes use of XML configuration to build modules, define targets, and specify dependencies.

```
<?xml version="1.0"?>
<project name="Demo" default="compile">
  <target name="clean" description="remove class files">
    <delete dir="classes"/>
  </target>
  <target name="cleanup" depends="clean"
description="perform cleanup">
    <delete file="hello.jar"/>
  </target>
  <target name="compile" description="build a java module">
    <mkdir dir="classes"/>
```

```

        <javac srcdir="." destdir="classes"/>
    </target>
    <target name="jar" depends="compile" description="build
jar">
        <jar destfile="demo.jar">
            <fileset dir="classes" includes="**/*.class"/>
            <manifest>
                <attribute name="Main-Class" value="Demo"/>
            </manifest>
        </jar>
    </target>
</project>

```

IVY.

Ivy is another build tool from Apache and brought as an extension to Ant. It helps in managing the external dependencies during the development, provides a way of adding them to the classpath, and packaging them with the application.

It is a perfect light-weight alternative of Maven.

Gradle.

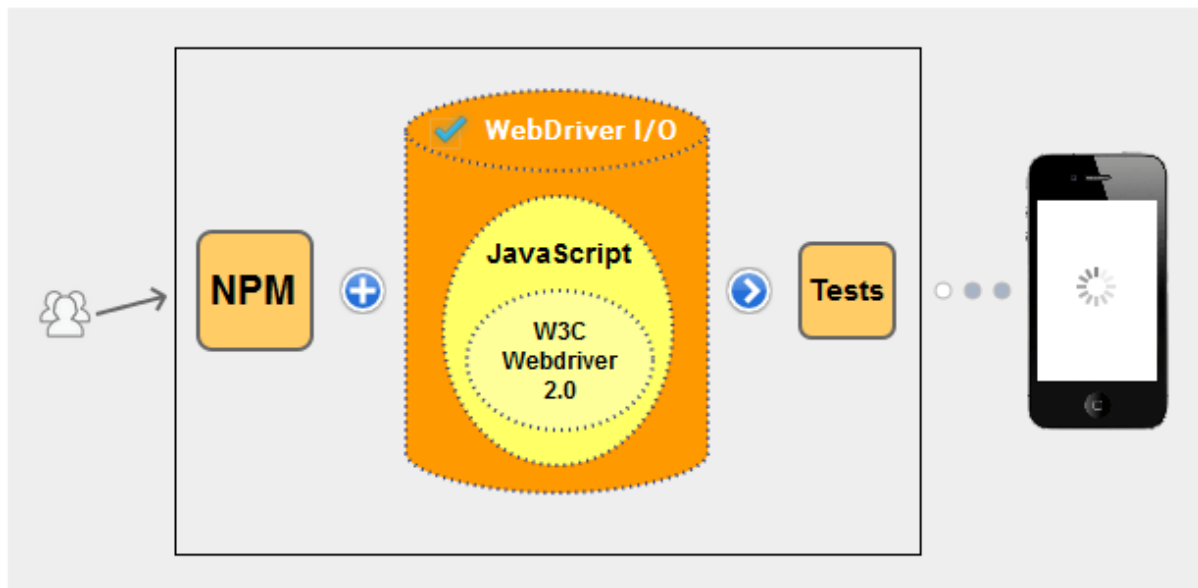
Gradle is an open source build tool. It automates the build process taking over Ant and Maven tools. Unlike its predecessors, Gradle makes use of a domain-specific language (DSL) instead of the XML templates for defining the project configuration. This DSL follows the groovy syntax.

It proposes a polyglot build system which integrates projects with different technologies and programming languages. It makes the dependency version management easier than Maven with the help of dynamic versioning.

3) What Is WebDriverIO? Which Technology Does It Use? Why Should QA Use It For UI Automation?

Answer.

WebDriverIO is a powerful framework for **testing mobile and web applications**. It uses JavaScript to implement the Selenium WebDriver API and the wire protocol as per the W3C standards. It is available for installation as NPM package and runs on Node.js. Hence, it is also famous with the name as “Selenium 2.0 bindings for NodeJS”.



Here, we are listing down a few of the silent features of WebdriverIO.

WebdriverIO Features.

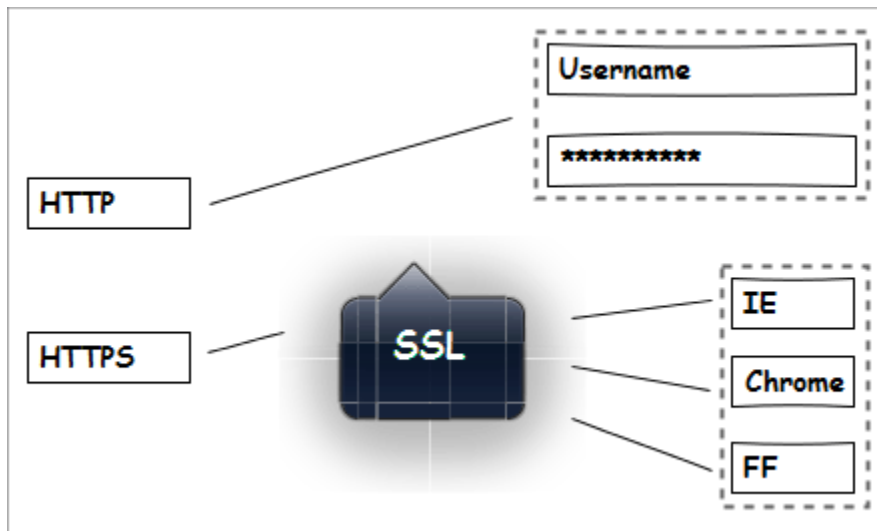
- It is a multipurpose automation tool which can fit in to test both the web applications and the native mobile Apps.
- Command execution takes place asynchronously in WebdriverIO. However, we can write them in a synchronous way so to avoid any race conditions that may occur while handling a promise.
- WebdriverIO provides a set of simple JavaScript functions to write tests which are lean, fast and easy to understand.
- It comes with an intelligent Test Runner which optimizes test execution to achieve maximum concurrency. Ideally, it can run all your tests in parallel if adequate CPU/memory resources are present.
- Test runner also adds a feature to register hooks for handling errors and altering test flow based on previous test result. For example, capturing a screenshot when any error occurs or ignoring a set of test cases if a dependent test gets failed.
- It integrates seamlessly with CI tools like Jenkins and utilizes JUnit reporter to debug issues and tracking test execution.
- Services like Sauce Labs and BrowserStack can plugin with WebdriverIO to run tests through the secure tunnel of respective environments.
- Another service for Appium enables testing for mobile devices and eliminates the need to start it manually for test execution.

The primary objective of WebdriverIO is to automate the end to end testing on a large scale. Its test runner helps in building a reliable Test Suite which is easy to understand and manage. It resolves many issues that occur with standard automation libraries. With it, we can organize the tests to run in split mode to enable concurrency. Moreover, it does provide session management and a lot of features to help in debugging runtime issues and find errors in the test cases.

4) What Are The Different Network Protocols That Selenium Supports? Also, How Will You Handle Them In Selenium?

Answer.

HTTP and HTTPS are the two most commonly used network protocols. And in Selenium, there are multiple ways to handle both of these protocols.



Handle HTTP Protocol.

Website Not Using Authentication.

If a website is only using the HTTP and does not require any authentication, then it's very easy to handle in Selenium.

```
WebDriver driver = new ChromeDriver();  
driver.get("http://destination-url/");
```

Web Site Using Basic Authentication.

There are many websites which apply basic authentication scheme before allowing access to their home page. Here are three most prominent ways to do it in Selenium.

Below code is using WebDriverWait and Alert classes to implement basic authentication for HTTP.

```
WebDriverWait testwait = new WebDriverWait(driver, 10);  
Alert testalert =  
testwait.until(ExpectedConditions.alertIsPresent());
```



```
testalert.authenticateUsing(new UserAndPassword(**user**,  
**pass**));
```

Alternatively, we can also pass the user/pass pair within the HTTP URL as a parameter to the Webdriver's Get method.

String target = "http://user:pass@host";

```
public void login(String username, String password) {  
    WebDriver driver = new ChromeDriver();  
    String URL = "http://" + username + ":" + password + "@" +  
"website link";  
    driver.get(URL);  
    driver.manage().window().maximize();  
}
```

We can also perform basic authentication by setting up browser preferences using the Selenium's profile classes.

```
FirefoxProfile FF = new FirefoxProfile();  
FF.setPreference("network.http.phishy-userpass-length", 255);  
driver = new FirefoxDriver(FF);  
driver.get("http://user:pass@www.targetsite.com/");
```

Handle HTTPS Protocol.

HTTPS is the secure version of HTTP. And it encrypts all the communication exchanged between the web server and the browser. It makes use of SSL certificates which get downloaded into the browser upon initiating the first request. The certificate contains a public key for encrypting the data flowing from client to the server. Similarly, the server reserves a private key corresponding to the certificate and use it to decrypt the data received. These public/private keys are none other than the famous RSA key-pair and available in different key lengths such as 512, 1024, 2048, and 4096 bits.

So in reality, we need to handle the SSL certificates via Selenium. Also, we've to follow different approaches for managing the SSL in Firefox, Chrome, and IE browsers.

Handle SSL In Firefox.

In Firefox, we can enable SSL via Selenium Profile APIs.

First of all, we load the profile of running browser instance.

```
ProfilesIni pf = new ProfilesIni();  
FirefoxProfile ff = pf.getProfile ("myProfile");
```

Consequently, we have to set the following two properties to avoid any security exception that might occur while opening the HTTPS site.

```
ff.setAcceptUntrustedCertificates(true);  
ff.setAssumeUntrustedCertificateIssuer(false);
```

Finally, we can create the Firefox driver instance by using the profile object.

```
WebDriver driver = new FirefoxDriver (ffProfile);
```

Handle SSL In Chrome.

Unlike Firefox, Chrome requires setting up the SSL options via desired capabilities of Selenium Webdriver. Here is the code to suppress all SSL errors and accept all the SSL certificates in Chrome.

First of all, we'll prepare DesiredCapabilities instance in our code.

```
DesiredCapabilities set_ssl = DesiredCapabilities.chrome();  
set_ssl.setCapability (CapabilityType.ACCEPT_SSL_CERTS, true);  
WebDriver driver = new ChromeDriver (set_ssl);
```

Handle SSL In IE.

Managing SSL in IE is a little different than the other two browsers. Again, it is possible with following two methods.

With Native JavaScript Code.

```
driver.navigate ().to  
("javascript:document.getElementById('overridelink').click()")  
;
```

With DesiredCapabilities Class.

```
DesiredCapabilities ssl = new DesiredCapabilities();  
ssl.setCapability(CapabilityType.ACCEPT_SSL_CERTS, true);  
System.setProperty("webdriver.ie.driver", "IEDriverServer.exe")  
;  
WebDriver driver = new InternetExplorerDriver(ssl);
```

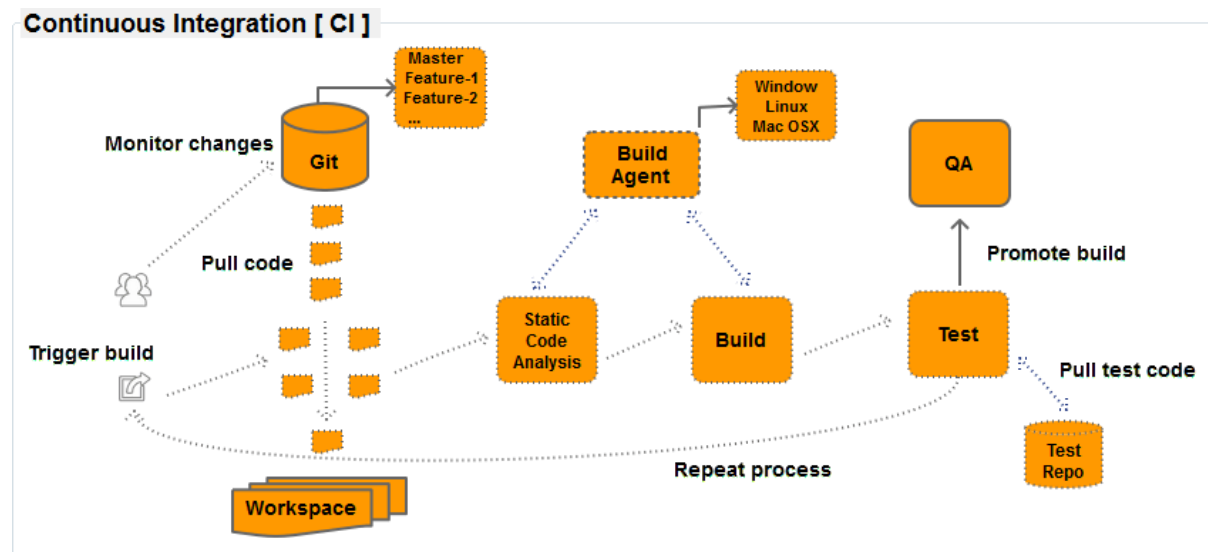
5) What Is Continuous Integration (CI), What Are Its Benefits? What Are Different CI Tools Available In The Industry?

Answer.

Continuous Integration is a software engineering technique which regulates the development process by automating the build and testing of code for every commit made to the version control.

It greatly helps in making sure that new changes didn't have any adverse effect on the build and existing functionality is intact. Even in a failed scenario, everyone

including the one responsible for the changes would know about the failure without any delay. With such a timely reporting, the developer not only finds fair time to fix the issue but also gets the prompt feedback.



It promotes best practices amongst the developers by making them merge their changes with other's code before checking into a shared repository. Timely code integration eliminates chances of conflicts, duplication of functionality, and diverging design tactics.

CI proposes to use a version control system like Git which supports Git workflows for intelligent branching. It suggests having a main branch which keeps the stable code and can be used to deliver a build at any point in time. On the other hand, the developer may create transient features branches to begin his work without interrupting others. He then needs to sync his code with the Main Branch at regular intervals. Once the feature is complete, he should merge the code into the main after due verification. The feature branch is subject to deletion when no longer it is required.

There are many CI tools available which can assist in implementing the continuous integration process. Here we are listing a few of the popular ones.

1. Jenkins.

Jenkins is undoubtedly the most popular CI tool used across the IT industry. It is open source, written in Java and created by Kohsuke Kawaguchi. It offers easy management via commands and GUI. It supports a rich set of plugins to customize both build and automation jobs.

2. Bamboo.

Bamboo is another great CI tool but a paid one. It's from the Atlassian, the company which provides helping tools like JIRA, BitBucket for software development. Unlike

Jenkins which uses Build/Automation jobs terminology, Bamboo has introduced the concepts of projects and plans. It targets to bring more user-friendly features so that the user can implement CI with ease.

Some other **CI tools** are **TeamCity**, **Travis CI**, and **Circle CI**.