

Battle of the Automation Servers: Jenkins vs. Bamboo vs. TeamCity



In many product development workflows, there are three main concerns: building, testing, and deployment. In these scenarios, every change that is made to the code means something could accidentally go wrong, so to lessen the likelihood of this happening, developers assume many strategies to reduce incidents and bugs. One strategy is to adopt *continuous integration* (CI) tools, which are used together with a source version software to verify if something has gone wrong for every update. CI is a procedure in which every commit on the repository triggers a custom workflow that can do all sorts of testing. It can also code lint to detect bad programming and take care of code style and report generation.

A step ahead of CI is *Continuous Deployment* (CD). With CD, not only will all the code be tested for every change, but it will also be ready to be deployed. There are a lot of complexities at this stage. Database dependent systems, for example, should apply database migrations and data transformation at every deployment that needs it. In the same case, it is common practice to prepare rollback scripts if the deployment fails.

Automation servers are a crucial aspect of modern software development workflows. It is important to detect problems before they go to production as soon as possible, and CI workflows try to detect them after every developer change. This way, potential big code reworks are avoided.

In this article, three popular automation servers, as well as their testing, deployment, and delivering solutions, will be compared: Jenkins, Bamboo, and TeamCity.

Automation and the Role of Continuous Deployment

Using CD, every successful code change is ready to be deployed, and, in most cases, is automatically deployed to production. In this scenario, the *fail fast* strategy is essential when deploying to production, as even the best automated tests could not cover all edge cases. Those special cases will be quickly discovered in production, and the deployment may be rolled back.

As a job executor, the automation tool is also used to automate repetitive jobs (such as backup/restore databases), turn on and off machines, collect statistics about a service, and more. The automation server can be used for tasks other than CI or CD: for example, the automation of tasks such as cleaning up the server, logging, archiving, and others.

Comparing Jenkins, Bamboo, and TeamCity

In this section, we will review three of the most popular automation tools for continuous integration.

Jenkins

Jenkins is a very popular self-contained, open-source, Java-based CI server that is used to automate tasks related to building, testing, deploying, and delivering software.

With Jenkins, developers can redirect their attention to work that matters because most of the integration and testing is managed by the tool. For this reason, problems are detected and solved right away, which maintains the software in a safe-release stage. It makes development faster, saving time and money, and providing the client with high-quality products.

Jenkins is a good option for a CI tool because it is free, straightforward to install, has more than 1,000 plugins and add-ons, and is written in Java. Hence, it is portable to all major platforms and has a very vibrant community.

Bamboo

Bamboo is a CI and CD server that is used to automate the release management for a software application, providing a continuous delivery pipeline. The tool schedules and coordinates the work involved in building and testing.

Bamboo is an Atlassian product, and can be integrated with many of the company's tools, such as Jira Software, Fisheye, and Bitbucket. It can also be customized with tons of features and

add-ons that can be found at the Atlassian Marketplace (for example, it is possible to get a plugin for Slack notifications).

TeamCity

TeamCity is a Java-based build management and CI server for developers and build engineers. TeamCity is made by JetBrains, the producer of IntelliJ Idea, Pycharm, and Android Studio. It has some platform-specific features, such as running code coverage and duplicates finder for Java and NET, in addition to being able to customize statistics on build duration, success rate, code quality, and custom metrics.

Comparison

In this section, we will evaluate some key, carefully selected categories.

Installation

Jenkins has many solutions for installation; it's possible to install it through package managers like APT, by downloading the artifact, or by running it through a Docker machine. On Windows, the operator will need to perform a manual operation because there is no installation wizard. In order to update, the user just needs to change the artifact, but it can be tricky in some versions. Backup and restore are manual processes that can be tricky too.

Bamboo provides the same Jenkins installation features and includes a Windows installation wizard. Upgrades may be complicated; if you have a very different version, you may need to upgrade several times to achieve the desired effect. Backup and restore are easily managed.

TeamCity has no package manager for the Linux environment (you have to deploy the artifact), but has one for Windows. New versions are automatically detected and upgraded, and backup and restore are easily handled.

Features

All tools can run distributed; each task can run it through a different machine without affecting the graphical user interface. Only Jenkins can use the same instance it runs the graphical user interface on to execute tasks. Although tool deployment is easy, there are some performance issues.

Jenkins has an extensive integration platform, with many official and unofficial plugins. Bamboo has excellent integration with other Atlassian products, like Jira or Confluence. TeamCity, on the other hand, has excellent integration with other JetBrains products, like IntelliJ or Pycharm.

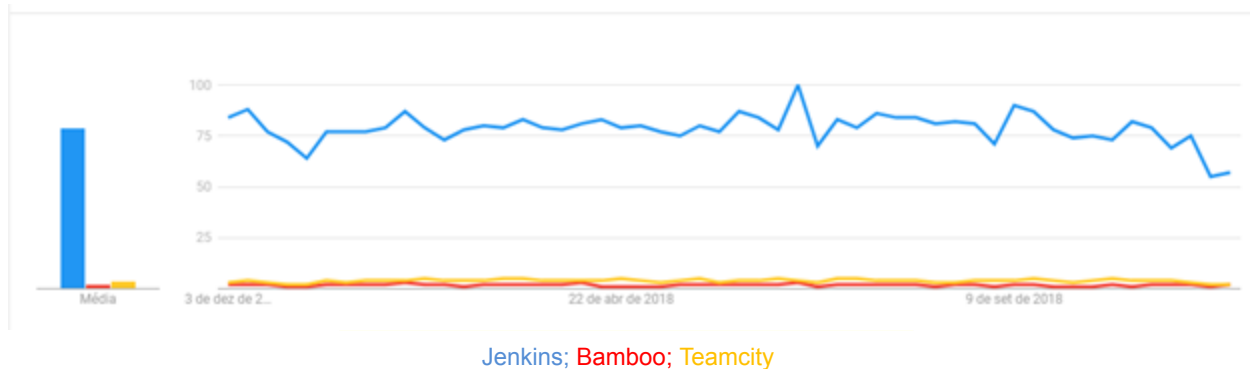
Architecture

All three products are client-server tools that can be quickly deployed on the cloud although there aren't PaaS solutions (as they need several integrations with the environment that are not feasible on a PaaS).

They are heavily modified with plugins, but there is one highlight for Jenkins: it is entirely plugin-driven, and its large community helps maintain its popularity.

All three tools have separate agents to run tasks on diverse processes (or machines), with different ways to auto-scale the number of agents running on demand. This is a manual operation in most cases, with Jenkins being more difficult.

Popularity



As an open-source product, Jenkins shines because of its popularity. It's also one of the more established products available in this area.

On StackOverflow, at the time of this article, we have the following numbers:

Product	Number of questions	Jobs on StackOverflow
Jenkins	32,647	110
Bamboo	1,253	15
TeamCity	5,258	21

The greatest difference in the number of questions asked can be explained by the fact that Bamboo and TeamCity are paid products, and as such, have dedicated commercial support. The number of jobs indicates the clear winner in terms of popularity.

Price

- Jenkins: Free and open source.
- Bamboo: Trial version for free. Price starts at 10 USD for up to 10 jobs, then increases to 1,100 USD for unlimited jobs and one remote agent (you will not be able to run jobs in parallel).
- TeamCity: The professional server is free for up to 100 build configurations and 3 build agents. Each new build agent plus 10 additional build configurations adds 299 USD to the bill. The Enterprise version, with unlimited build configurations and priority support, starts at 1,999 USD.

Conclusion

The table below summarizes our comparison of the three tools:

	Installation	Features	Architecture	Popularity	Price
Jenkins	Easy install. Easy upgrade. Tricky backup.	Extensive integration platform. Distributed runners.	No PaaS. Plugin-driven. Dynamic scale possible.	Most popular	Free and open source
Bamboo	Easy install. Tricky upgrade. Easy backup.	Extensive integration platform. Distributed runners. Integration with Atlassian products.	No PaaS. Plugin-driven. Dynamic scale possible. Integration-based.	Third in popularity	Commercial

TeamCity	Easy install. Easy upgrade. Easy backup.	Extensive integration platform. Distributed runners. Integration with JetBrains products.	No PaaS. Plugin-driven. Dynamic scale possible.	Second in popularity, far behind Jenkins	Commercial
-----------------	--	---	---	--	------------

Integration solutions are essential tools for reaching delivery reliability with a large team. As every developer commits its changes to a shared repository, each commit triggers an automated process for testing and building. If the process fails, it's possible to repair what is wrong quickly and safely. When it's likely to find problems and solve them quickly, we release higher-quality products and more stable experiences for the client.

There are lots of options on the market to help developers' workflow. As we analyzed in this article, there are many excellent and reliable solutions for you to choose from. While Jenkins' highlights are its community and many plugins, Bamboo's and TeamCity's bright spots are the product integrations such as Jira and IntelliJ.

In order to make a good decision, a solid approach would be to freely evaluate the tools and how they integrate with your software development products. To properly evaluate each tool, we recommend using Docker repositories to deploy a sandbox environment easily:

- [Jenkins \(agent\)](#)
- [Bamboo \(agent\)](#)
- [Teamcity \(agent\)](#)

Automation servers are an essential part of the majority of development processes worldwide. They are constructed so that a team can quickly identify an issue—either by automated tests or by rapid response for production issues (which involves an automated deployment). In this article, we've evaluated three popular automation tools and compared their features, installation processes, and prices so that you can decide which one is best for your company.

Note: This blog was originally published on Logz.io - <https://logz.io/blog/automation-servers/>