Bamboo

CI/CD with Bamboo helps in building CI/CD pipelines, especially when the application involves cross-connectivity with different APIs and various third-party applications. This also provides immunity to the risk of a failed production deployment. GitLab helps in version controlling the code, whereas CI/CD with Bamboo helps to pipeline the source code for different types of testing (i.e., cross-browser testing, white-box testing, unit testing, etc.). The best part is that the advantages offered by CI/CD with Bamboo are independent of the development language. CI/CD with Bamboo is used for PHP projects for ensuring that the code going into the production pipeline is of high quality.

Till now, we touched upon the importance of pipelines in the automated process of building software applications and deploying them to the target production environment. With the stage all set, let’s look at how to set up the CI/CD pipeline with Bamboo for the PHP project.

In case you are curious to know how Bamboo stacks up against other CI/CD tools, make sure to check out the [Comparison of top CI/CD tools](https://www.lambdatest.com/learning-hub/cicd) in the LambdaTest Learning Hub.

**Table of content**

* [Pre-requisites for Bamboo Installation](https://www.lambdatest.com/blog/how-to-setup-cicd-pipeline-with-bamboo-for-php-projects/#Prerequisites)
* [Why use Bamboo Pipeline for PHP Project](https://www.lambdatest.com/blog/how-to-setup-cicd-pipeline-with-bamboo-for-php-projects/#BambooPipeline)
* [How to install and run Bamboo CI](https://www.lambdatest.com/blog/how-to-setup-cicd-pipeline-with-bamboo-for-php-projects/#install)
* [How to Configure Bamboo CI](https://www.lambdatest.com/blog/how-to-setup-cicd-pipeline-with-bamboo-for-php-projects/#Configure)
* [How to integrate Bamboo Pipeline with Cloud Selenium Grid](https://www.lambdatest.com/blog/how-to-setup-cicd-pipeline-with-bamboo-for-php-projects/#integrate)
* [Using Bamboo CI/CD to run tests on LambdaTest](https://www.lambdatest.com/blog/how-to-setup-cicd-pipeline-with-bamboo-for-php-projects/#CICD)

Bamboo is a popular Continuous Integration (CI) tool developed by Atlassian in 2007. Bamboo is a commercial product that lets you perform automated builds, tests, and releases. All of this in a single workflow!

**Pre-requisites for Bamboo Installation**

Before proceeding with the Bamboo CI/CD pipeline for PHP projects, you need to ensure that prerequisites for Bamboo are met. Here the prerequisites for installing and configuring Bamboo on the requisite (i.e. Windows/Ubuntu) server.

* **Installation of Atlassian Bamboo**

Atlassian Bamboo is a Continuous Integration and Continuous Deployment (CI/CD) server. Bamboo provides assistance to software development teams with automated building and testing of software source code. It can also work as a reporting tool for statistical analysis. The steps to install Bamboo are provided ahead in the article.

* **Installation of PHP**

PHP is a server-side scripting language that is used to develop static websites, dynamic websites, or web applications. It stands for Hypertext Preprocessor, which was earlier known as personal home pages. The installation of PHP and PHPUnit (i.e. unit testing framework in PHP) is necessary since we would be demonstrating CI/CD with Bamboo for the PHP project. Please check out our [detailed Selenium PHP tutorial](https://www.lambdatest.com/blog/selenium-php-tutorial/) for steps on installing PHP, PHPUnit, Composer, and more.

* **Installation of Node.js**

In the integration of LambdaTest with Bamboo, we will install Node.js as a second task. Node.js is a JavaScript runtime that is used to build scalable network applications. Node.js has no thread networking and thus is free from deadlocks. You can download the zip version or the windows installer version from [here](https://nodejs.org/en/download/). You can further read [Selenium NodeJs tutorial](https://www.lambdatest.com/blog/how-to-run-automation-tests-using-selenium-and-nodejs/) to gather details on automation testing with NodeJs.

* **Git repo of a PHP Project**

Git repository is used for version control of source code. It is also used to clone the repository and create a build of the application. The PHP project in the [Git repo](https://github.com/rachnaagrawal/php-selenium-sample.git" \t "_blank) will be used for the demonstration of CI/CD with Bamboo with a PHP project.

For the demonstration of Bamboo integration with LambaTest integration, we will use the repository with Javascript project and Protractor tests. The link to the repository is <https://github.com/LambdaTest/protractor-selenium-sample-gitlab.git>

* **New Project in Bamboo**

The new project in Bamboo is to create a new pipeline for the build (or deployment) of the application. Since we will be using the LambdaTest Selenium Grid, we first create an account on LamdaTest. Please make a note of the username and access-key from the [profile section on LambdaTest](https://accounts.lambdatest.com/detail/profile). We will be integrating LambdaTest with the Bamboo pipeline.

Atlassian Bamboo can only run with the JDK (JAVA Development Kit). Thus, a supported version of Java is required. JAVA\_HOME is a system environment variable that provides the path to Java’s home directory. If a path is not returned you’ll need to set the JAVA\_HOME environment variable manually before installing Bamboo.

**Why use Bamboo Pipeline for PHP Project**

Now that we have covered the prerequisites for CI/CD with Bamboo, let’s look at the essentials of the Bamboo pipeline. The Bamboo pipeline lets you build and run unit & integration tests by integrating the source code.

Bamboo Pipeline also provides reliable, secure, and well-managed benefits over other tools like Jenkins. Some of the prominent advantages of using a Bamboo pipeline for PHP projects are below:

* **Automated build and test process**

The source code when pulled out of a repository to a Bamboo pipeline gets compiled and focuses on creating a build. After compilation, the unit tests and integration tests are run. The pipeline gives different ways to manage builds that have different requirements or targets.

* **Build and deployment process independent of the local environment**

When a large team assembles to create a secure software product, working in a local environment might not suffice the long-term requirements of the project. Version controlling using tools like GitHub, GitLab, etc. is an absolute must.

Bamboo pipeline fetches the code from the relevant repository, triggers a build, and integrates the tests. This pipeline makes the whole process automated and triggers as soon as a developer commits code. This is one of the major advantages of using CI/CD with Bamboo or in general using CI/CD along with automation testing.

* **Automatic trigger of unit tests and integration tests**

The Bamboo pipeline helps in optimizing the build performance through parallelism.  
This pipeline leverages elastic resources and can perform different tests in different environments.

* **Integrate code repositories from version control systems**

The Bamboo pipeline can connect to a lot of version control systems via the relevant plugins. Build can also be tested against different target environments. The build and test process can be expedited with parallel test execution.

* **Automatic builds and deployments are performed to a server**

The Bamboo pipeline starts the build, just like it is done by MSBuild in Visual Studio. Once the project is built, “Artifacts” are created. “Artifacts” are the build results like an executable app, config files, property files, etc.

* **Plugins and Interfaces for performing tasks**

Bamboo has a rich plugin ecosystem with plugins available for doing tasks such as cloning a source code repository, performing user acceptance testing (UAT), and more. Some plugins also enable Sonarqube integration that helps in performing code quality checks. The Integration builds from the pipeline provide early ‘fail fast’ feedback on the quality of new changes.

* **Detailed reports statistics**

Bamboo uses the concept of a ‘plan’ with ‘jobs’ & ‘tasks’ to configure and order the actions in the workflow. A Bamboo pipeline can have more than one plan and can provide reporting across all plans. The reporting can be done via the wallboard which is a dashboard of the pipeline. It provides detailed statistics like the number of builds, deployment environments, etc. The pipeline also provides links to the other applications and allows setting up different permissions for all the plans.

* **Scheduling and coordinating sequential stages**

Another major advantage of CI/CD with Bamboo is sequential execution of different stages is possible with the Bamboo pipeline. Jobs in different stages can be executed sequentially and/or parallelly on a need basis. Stages process jobs on multiple agents and multiple environments. Sequential stages use artifacts from the previous stage in the upcoming stage. The end result is a more structured and well-organized pipeline workflow.

If you are a PHP expert, you can acquire a benchmark industry certification specializing in core PHP programming skills and expand your opportunities to advance your career in PHP automation testing.

**How to install and run Bamboo CI**

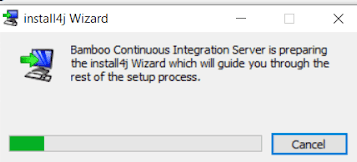
Now that we have covered the advantages of CI/CD with Bamboo, let’s look at how to run Bamboo for continuous testing:

Follow the below-mentioned steps for installing Bamboo on Windows:

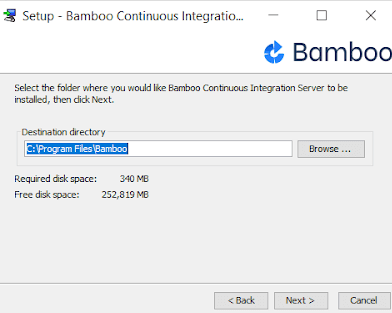
1. Check for the supported operating systems, hardware platforms, databases, and browsers for the version of Bamboo that is to be installed (more details [here](https://confluence.atlassian.com/bamboo/supported-platforms-289276764.html)).
2. Java installation and JAVA\_HOME environment variable should be set correctly. set correctly.
3. Download the [latest Bamboo](https://confluence.atlassian.com/bamboo/installing-bamboo-on-windows-289276813.html) executable for installation.

**Installing Bamboo using Windows Installer**

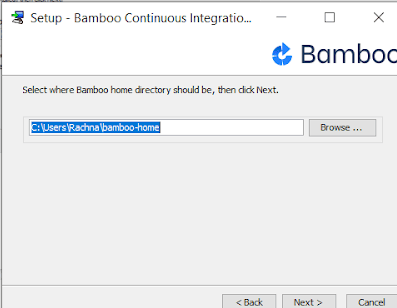
The installer is a setup wizard known as “install4j wizard” that sets up the Home directory for Bamboo and saves all the required bat files.



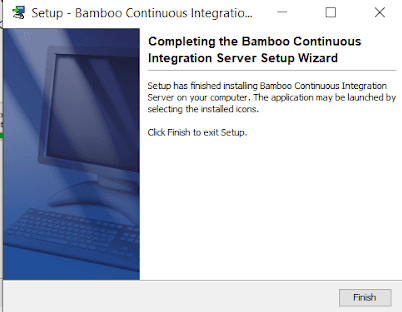
Select the folder where you want to install Bamboo and click Next



Set the home directory as C:\Users\<current-user>\bamboo-home and then click Next.



Click on Finish to complete the installation

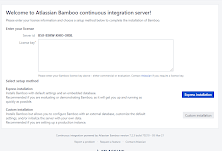


Installing Bamboo on Windows using the Zip Archive

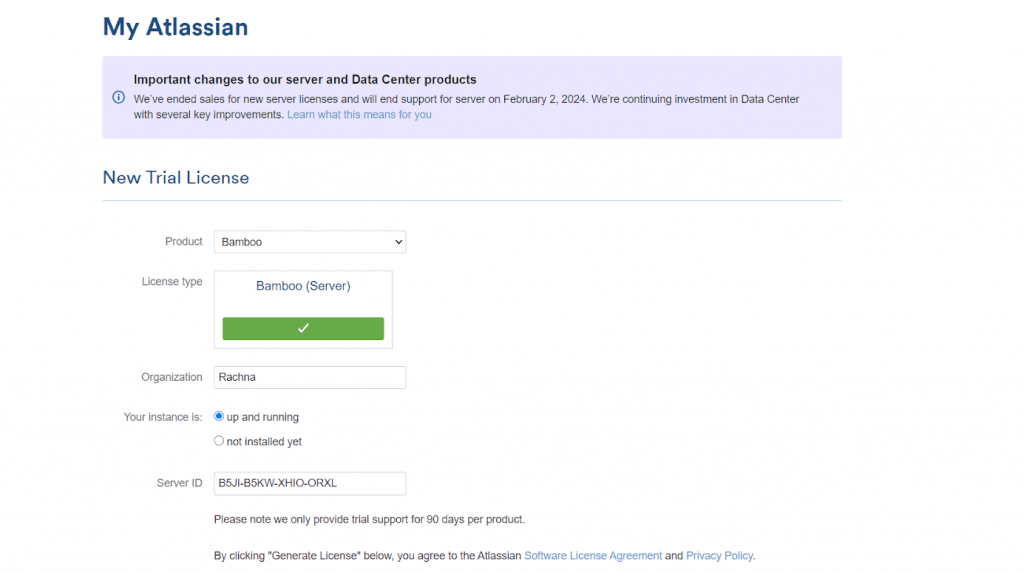
1. Download the ZIP folder for the [latest version of Bamboo](https://www.atlassian.com/software/bamboo/download)
2. Extract the contents of this ZIP Archive folder into a particular location (e.g. C:\Program Files\Bamboo)
3. Set up the Bamboo home directory to C:\Users\<current-user>\Bamboo-home.
4. With this, the installation of Bamboo is completed
5. Start the Bamboo server from the location: <Bamboo installation directory>\bin\start-bamboo.bat
6. Go to the web browser and access the running Bamboo instance by typing http://localhost:8085/ in the address bar.

**How to Configure Bamboo CI**

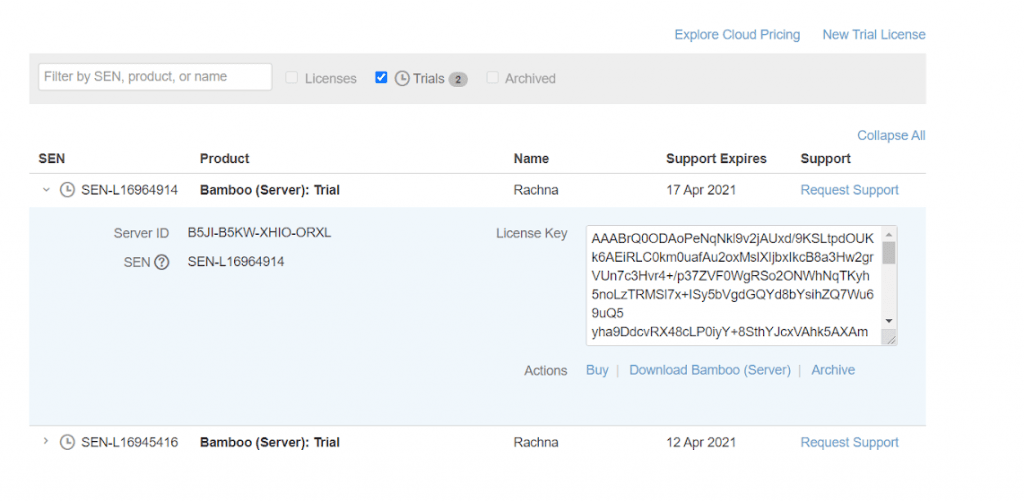
1. When you open the URL http://localhost:8085/ for the first time, you will be redirected to the setup wizard page
2. In order to use Bamboo, you must have a valid license key (evaluation or commercial). The evaluation license can be generated from the MyAtlassian self-service account.



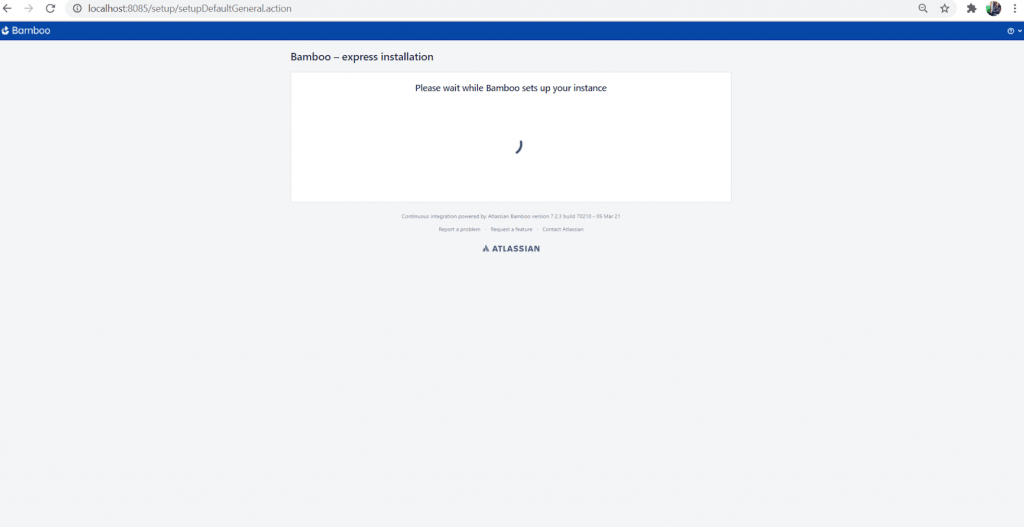
1. To get the valid license key, click on the link given down the box(on keyword ‘Atlassian’). You will be redirected to a new page for generating the license key. The trial version is valid for 90 days per project.



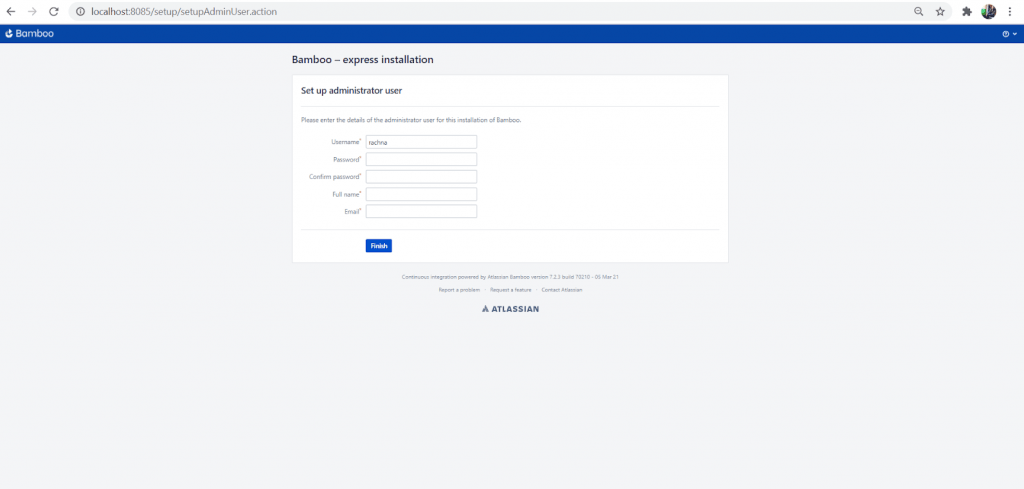
1. Check the Server ID and then click on “**Generate License**”.
2. You will be redirected to a new page, where the license key will be available in a box next to Server ID.



1. Copy the valid License key and paste it in the required box below the Server ID. You should also select **Express Installation**.
2. Alternately, you can also select Custom Installation. After selecting Custom Installation, it will redirect you to a new page. If you want to change the URL (or any system paths or directory), you can change it in this step. Post that, click on the ‘Continue’ button.
3. After a couple of minutes, the Bamboo CI instance will be ready for usage.



1. Set the administrator user. Go ahead and provide details like Username, Password, Full Name, and Email Id to register as the admin user.



1. After creating the admin user, you can log in to the Bamboo Application and create a new project or plan.

**How to integrate Bamboo Pipeline With Cloud Selenium Grid**

With the completion of Bamboo CI/CD installation, we are all set to perform Selenium test automation with PHP using Bamboo CI/CD. Here the major steps for using CI CD with Bamboo and Selenium PHP on a cloud-based Selenium Grid:

Cloud-based Selenium Grid specializes in running serial (and parallel) tests across multiple browsers, operating systems, and devices. LambdaTest provides a cloud-based Selenium Grid that lets you perform Selenium test automation on 2000+ different browsers and operating systems online. Check out our detailed tutorial on [cloud testing](https://www.lambdatest.com/blog/cloud-testing-tutorial/) to get an in-depth understanding of the benefits of cloud-based Selenium Grid.

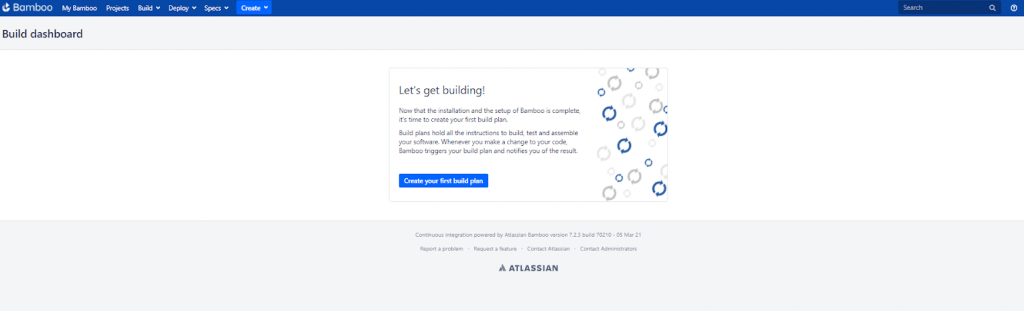
We will be using Bamboo CI/CD for the PHP project, with the execution on LambdaTest’s cloud-based Selenium Grid. Here are the steps for integrating the Bamboo pipeline with LambdaTest Grid:

1. Check for the installation of Bamboo, PHP, PHPUnit, and Composer with Selenium dependencies.

2. Once the setup is installed, we will check out the sample code in a [Github repository](https://github.com/rachnaagrawal/php-selenium-sample.git" \t "_blank). The code in the repository is a simple PHP and Selenium web automation script that searches for LambdaTest on Google.

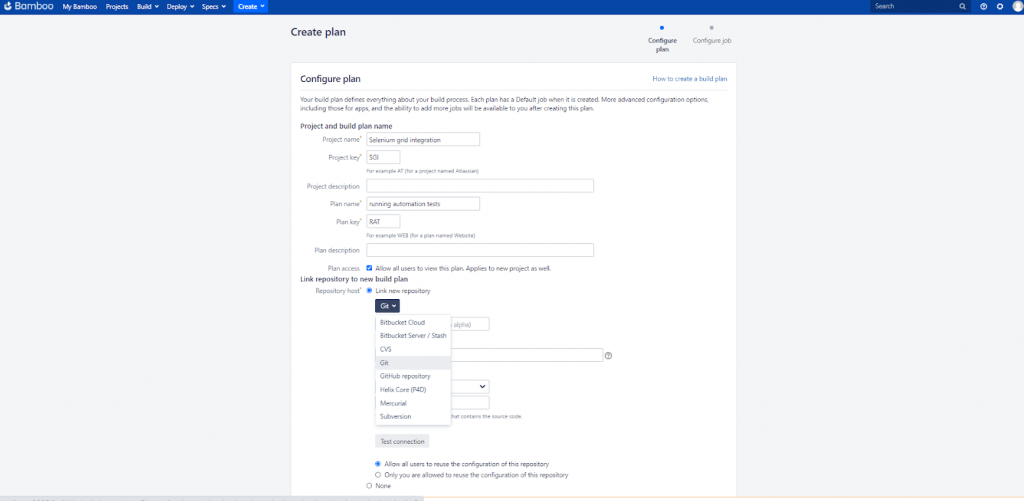
|  |  |
| --- | --- |
|  | <?php |
|  |  |
|  | require\_once('vendor/autoload.php'); |
|  |  |
|  | use Facebook\WebDriver\Remote\RemoteWebDriver; |
|  | use Facebook\WebDriver\WebDriverBy; |
|  | use PHPUnit\Framework\Assert; |
|  |  |
|  | /\* |
|  | LambdaTest selenium automation sample example |
|  | Configuration |
|  | ---------- |
|  | username: Username can be found at automation dashboard |
|  | accessToken: AccessToken can be generated from automation dashboard or profile section |
|  | Result |
|  | ------- |
|  | Execute PHP Automation Tests on LambdaTest Distributed Selenium Grid |
|  | \*/ |
|  |  |
|  | # username: Username can be found at automation dashboard |
|  | $LT\_USERNAME = getenv("LT\_USERNAME"); |
|  |  |
|  | # accessKey: AccessKey can be generated from automation dashboard or profile section |
|  | $LT\_ACCESS\_KEY = getenv("LT\_ACCESS\_KEY"); |
|  |  |
|  | $host= "http://". $LT\_USERNAME .":" . $LT\_ACCESS\_KEY ."@hub.lambdatest.com/wd/hub"; |
|  |  |
|  | $result="passed"; |
|  |  |
|  | $capabilities = array( |
|  | "build" => "Sample PHP Build", |
|  | "name" => "Sample PHP Test", |
|  | "platform" => "Windows 10", |
|  | "browserName" => "Chrome", |
|  | "version" => "88.0" |
|  | ); |
|  |  |
|  | try{ |
|  | $driver = RemoteWebDriver::create($host, $capabilities); |
|  |  |
|  | $driver->get("https://www.google.com/ncr"); |
|  |  |
|  | $element = $driver->findElement(WebDriverBy::name("q")); |
|  |  |
|  | if($element) |
|  | { |
|  | $element->sendKeys("LambdaTest"); |
|  | $element->submit(); |
|  | Assert::assertEquals($driver->getTitle(),'LambdaTest - Google Search'); |
|  | } |
|  | } catch(Exception $e) { |
|  | $result="failed"; |
|  | print "Test failed with reason ".$e->getMessage(); |
|  | } |
|  | finally{ |
|  | // finally quit the driver |
|  | $driver->executeScript("lambda-status=".($result == "passed" ? "passed":"failed")); |
|  | $driver->quit(); |
|  | } |
|  | ?> |

3. With the completion of the basic requisites for Bamboo CI/CD with PHP project, let’s log in to the Bamboo dashboard. Click on the Create your first build plan to construct a new Bamboo project.

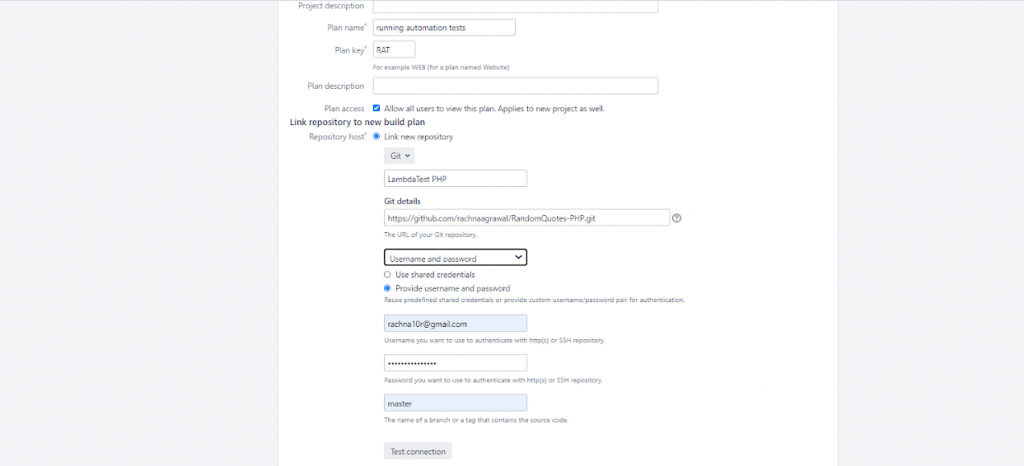


4. Enter the Bamboo project name as ‘Selenium grid integration’ and plan its name as ‘running automation tests’. The next step is to link the GitHub repository containing the source code and tests.

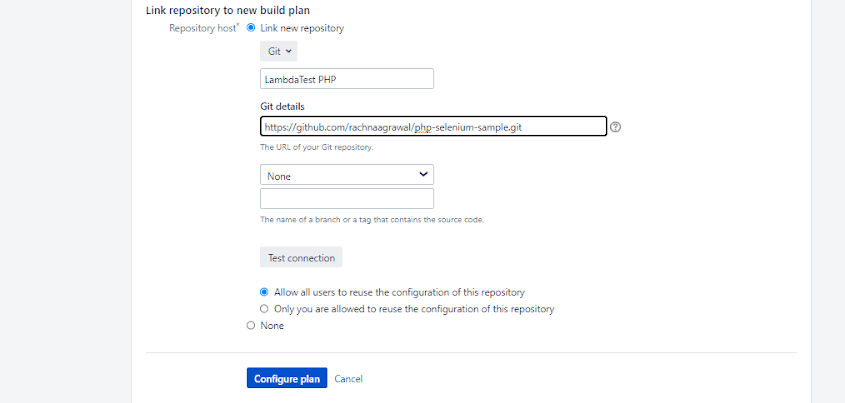
5. For linking the repo, click on Link new repository besides Repository host and from the dropdown ‘Select repository host’. Select Git as the preferred version control system.



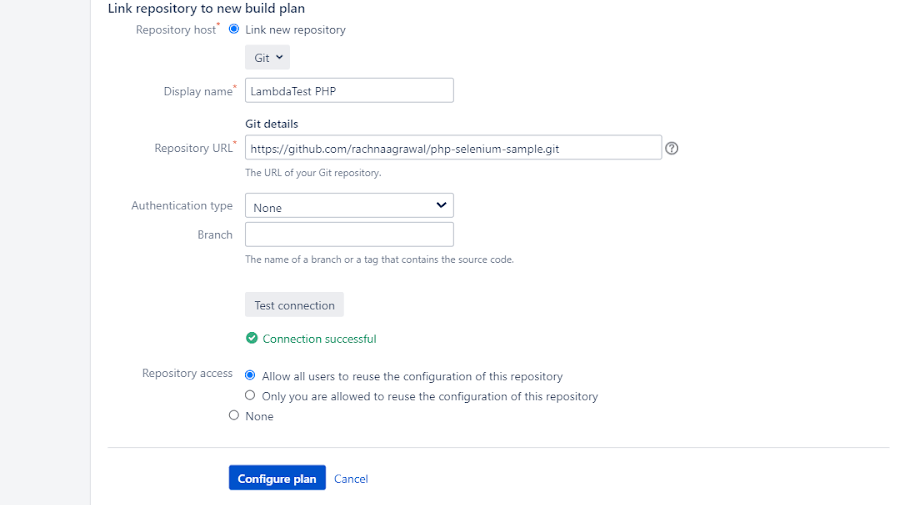
6. Enter the Git repo name (in this case it’s LambdaTest PHP) below the dropdown and then provide the repo URL beneath the Git details. If the repository is private, you would need either Username & Password or SSH Keys to authorize the repository.



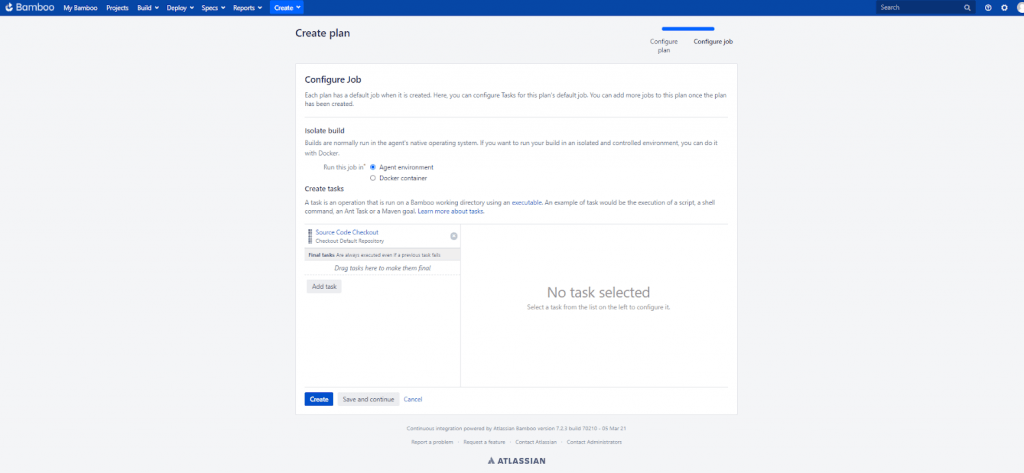
7. For public repositories, simply provide the URL



8. Use Test Connection to see if Bamboo is able to perform Source code Checkout appropriately or not. After clicking on Test Connection, you should get ‘green tick’ with indicating ‘Connection successful’.

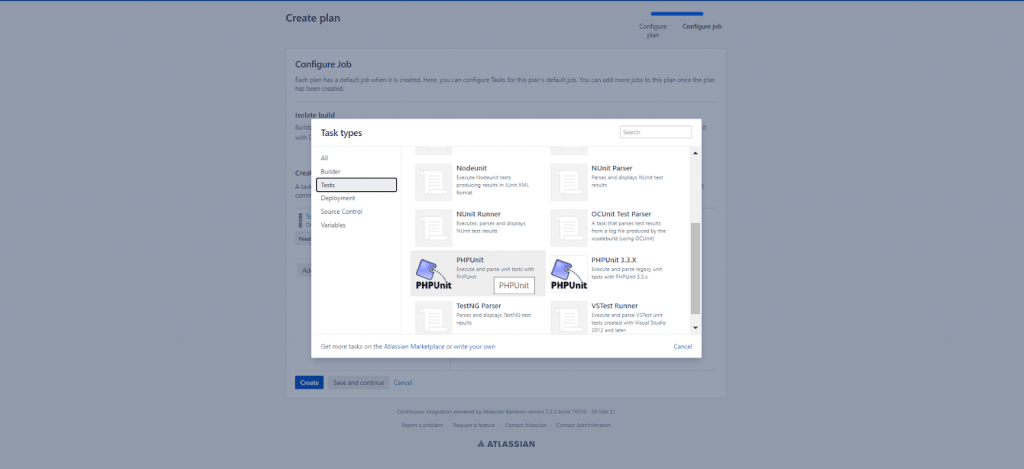


9. Click on the Configure Plan button. With this, the Bamboo CI/CD for PHP project is ready for use! A new page will appear to configure more tasks ahead.

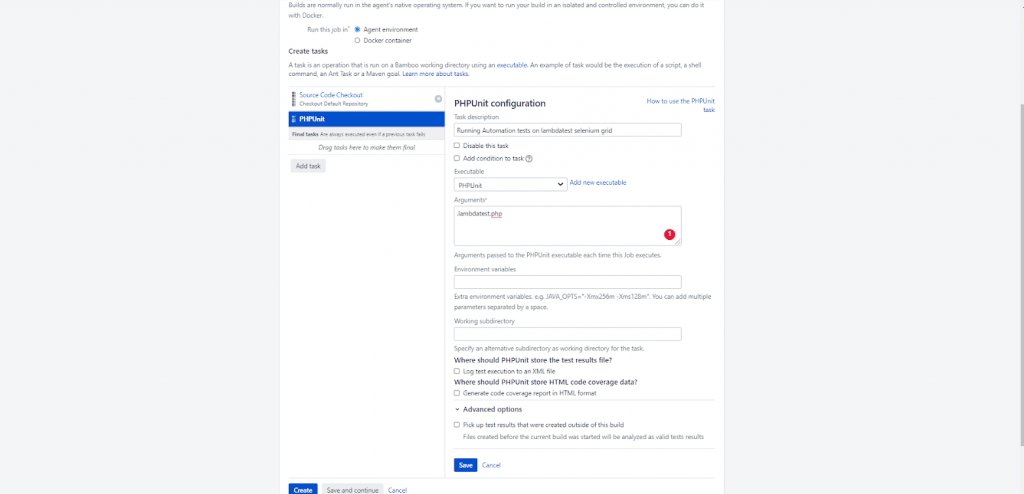


10. The first task is ‘Source Code Checkout’ which is built by default as we connect the Git repository to the Bamboo project. You can add more tasks by clicking on ‘Add task’. The build plan will run on the agent environment, but you can always provide the Docker image after selecting ‘Run this job in’ as ‘Docker container’. Check out our detailed blog on [how to run Selenium tests with Docker](https://www.lambdatest.com/blog/run-selenium-tests-in-docker/).

11. Click on ‘Add task’ to create a task for running Selenium web automation tests on the LambdaTest Grid via PHPUnit. Select ‘test’ from the left menu of the popup box and choose PHPUnit.

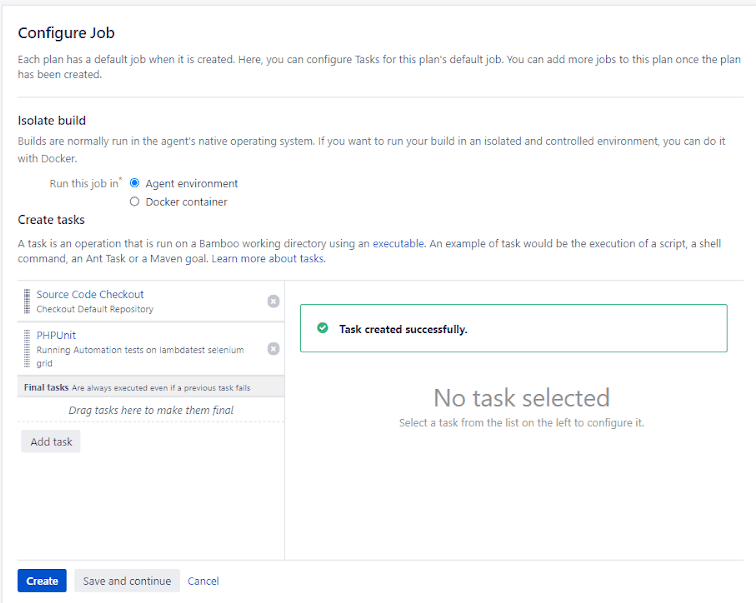


12. A new dialog box opens up in the right column of the page. Here, you will provide the required Task description, executable, arguments, environment variables etc. The details are given below.

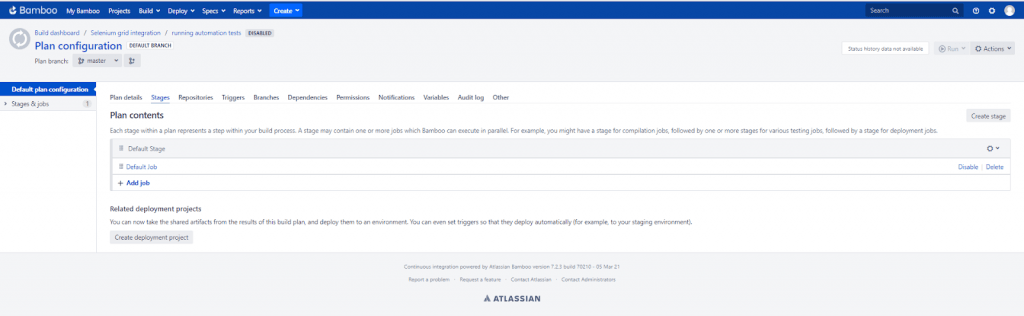


13. If the executable has nothing in the dropdown, you need to specify the address of the PHPUnit executable installed on your machine. Click on ‘Add new executable’ and a new popup dialog box appears. Here, provide the ‘Executable label’ as ‘PHPUnit’ and set the path as ‘C:\PHP7\vendor\bin\phpunit’( i.e. path where PHPUnit is installed). Select Add.

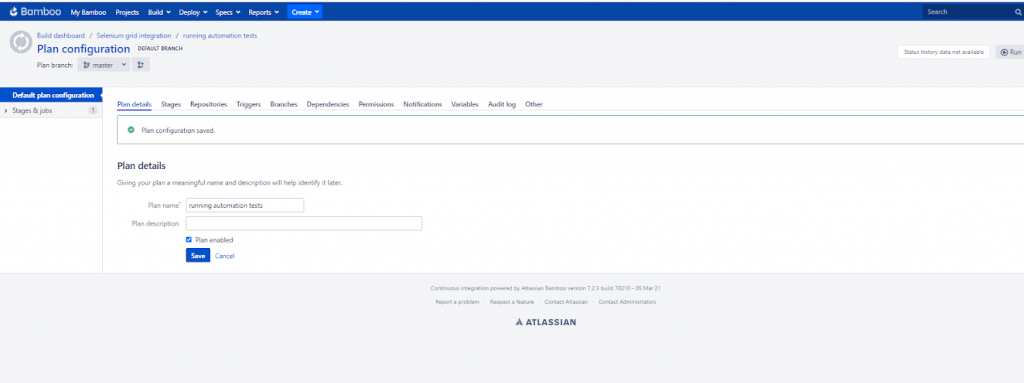
14. In this demo of Bamboo CI/CD for PHP projects, we run the tests implemented in lambdatest.php on the LambdaTest Grid. Provide lamdatest.php in arguments and click on Save. This will add a new task successfully.



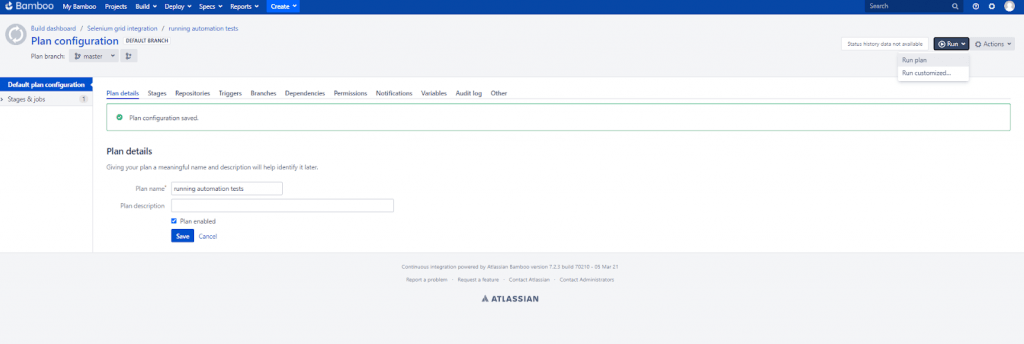
15. Click on the ‘Save and continue’ button, to save the plan configuration. A new page will open with all the plan details and configuration.



16. Click on the Default plan configuration and below the Plan details tab, check the Plan enabled checkbox. Click on Save to save the plan configuration.



17. Now, to run the plan and two tasks sequentially, just click on the Run > Run Plan button on the topmost right side of the page.



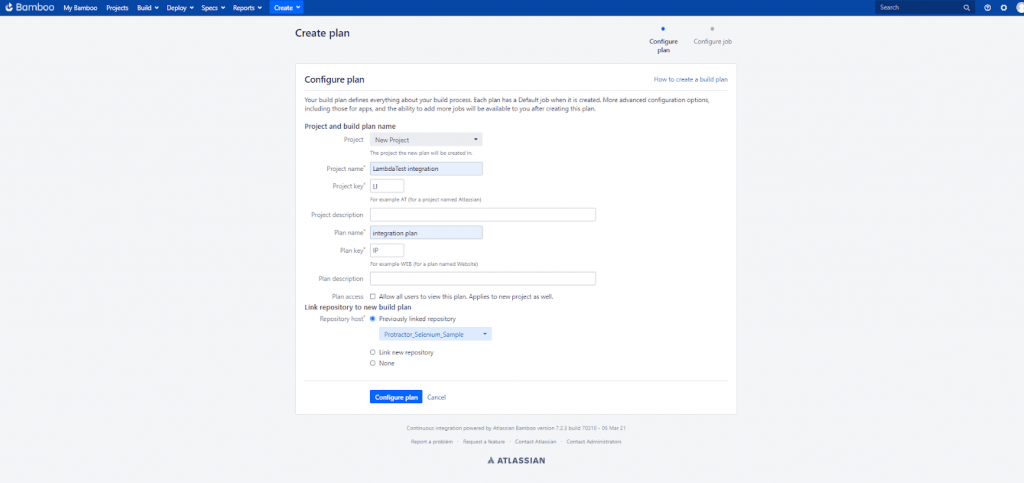
18. This will initiate the job and logs will be reflected under the logs tab.

**Using Bamboo CI/CD to run tests on LambdaTest**

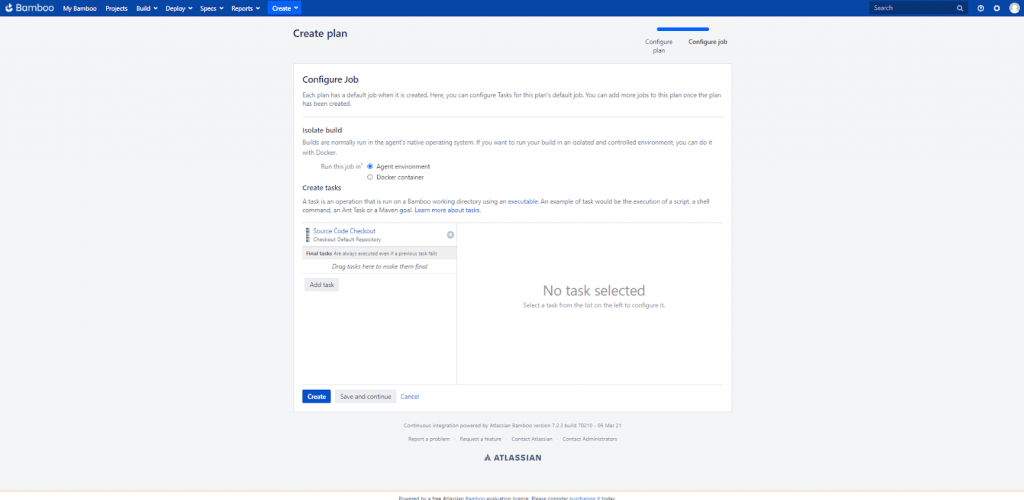
With LambdaTest integration with Bamboo CI/CD, you can perform cross-browser testing 2000+ browsers and operating systems online. The tests could be performed directly from the Bamboo server (e.g. browser compatibility testing can be performed as soon as the changes are committed to the build). Integration of LambdaTest with Bamboo CI/CD server can help in reducing release cycles and shipping the product faster.

Here are the steps for integrating LambdaTest with Bamboo CI/CD:

1. After installing the pre-requisites, login to your Bamboo server as [Administrator](https://www.lambdatest.com/software-testing-questions/what-is-an-admin-administrator)
2. Create a new Project named ‘LambdaTest integration’ and add a new plan named ‘integration plan’.
3. After creating a new plan, create 3 tasks. To know how to create a new plan and task, just refer to the previous steps discussed earlier.
4. The link to the new Git repository with Javascript project and Protractor tests is <https://github.com/LambdaTest/protractor-selenium-sample-gitlab.git>
5. Provide the repository name at the time of construction of the new plan
6. Fork this repository via your Git Account and replace <username> and <Access\_Key> in \*.conf.js

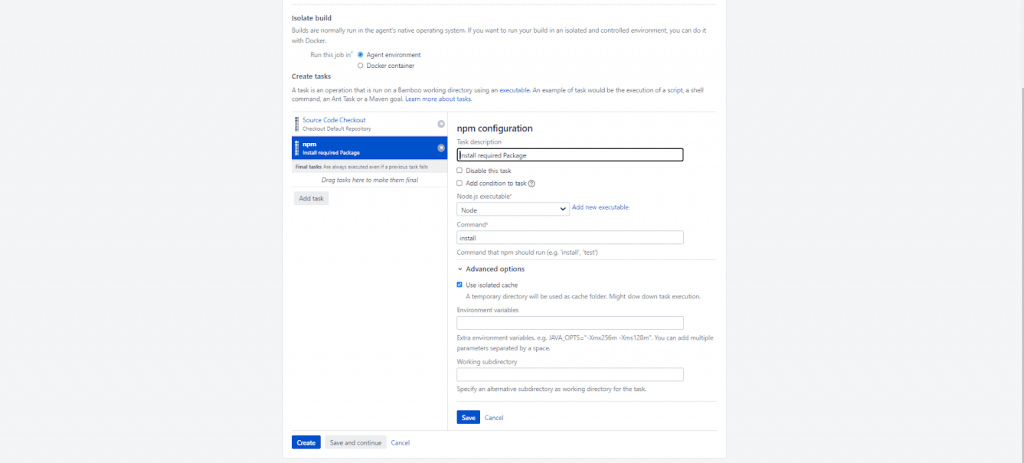


1. Click on the Configure Plan button. Our first task will be source code checkout which is created by linking to the repository. The plan will be created with the default job and stage.



1. Install required Package is the second task that needs to be added here. We will install Node.js. Click on ‘Add Task’ and in the popup dialog box, select npm. Specify the details as:

|  |
| --- |
| Task description: Install required Package |
| Node.js executable: <Add the path to the location, where Node.js is installed in your computer> |
| Command: install |
| Advanced Options : tick on the ‘Use isolated cache’ Checkbox |



1. In the third task, we will attempt Protractor testing which is an automation and end-to-end BDD testing framework that plays an important role in testing AngularJS Applications.

You can install Protractor via npm install command:

npm install -g protractor

You can also create a configuration file known as conf.js, for the local machine. The code to be put in conf.js is:

|  |
| --- |
| exports.config = { |
| seleniumAddress: 'http://localhost:4444/wd/hub', |
| spec: ‘single.js’ |
| }; |

To configure the protractor test for LambdaTest, you would need to use the below script:

|  |
| --- |
| echo "start testing" |
| //Add access credentials in plan variable |
| set LT\_USERNAME=<Your\_LambdaTest\_Username> |
| set LT\_ACCESS\_KEY=<Your\_LambdaTest\_Access\_Key> |
|  |
| wget https://downloads.lambdatest.com/tunnel/v3/windows/64bit/LT\_Windows.zip |
| unzip -o LT\_Windows.zip |
|  |
| //To start Tunnel |
| LT --user %bamboo.LT\_USERNAME% --key %bamboo.LT\_ACCESS\_KEY% |
|  |
| node node\_modules/protractor/bin/protractor single.conf.js |

1. Along with replacing bamboo.LT\_USERNAME and bamboo.LT\_ACCESS\_KEY with your LambdaTest Account Username and Access Key in the Script, you will also need to configure the LambdaTest Tunnel.

To configure the Tunnel in your local machine, login to your LambdaTest account. The Dashboard will appear, click on Automation from the left-side column on the dashboard. On the topmost right side, you will find a button named ‘Configure Tunnel’.

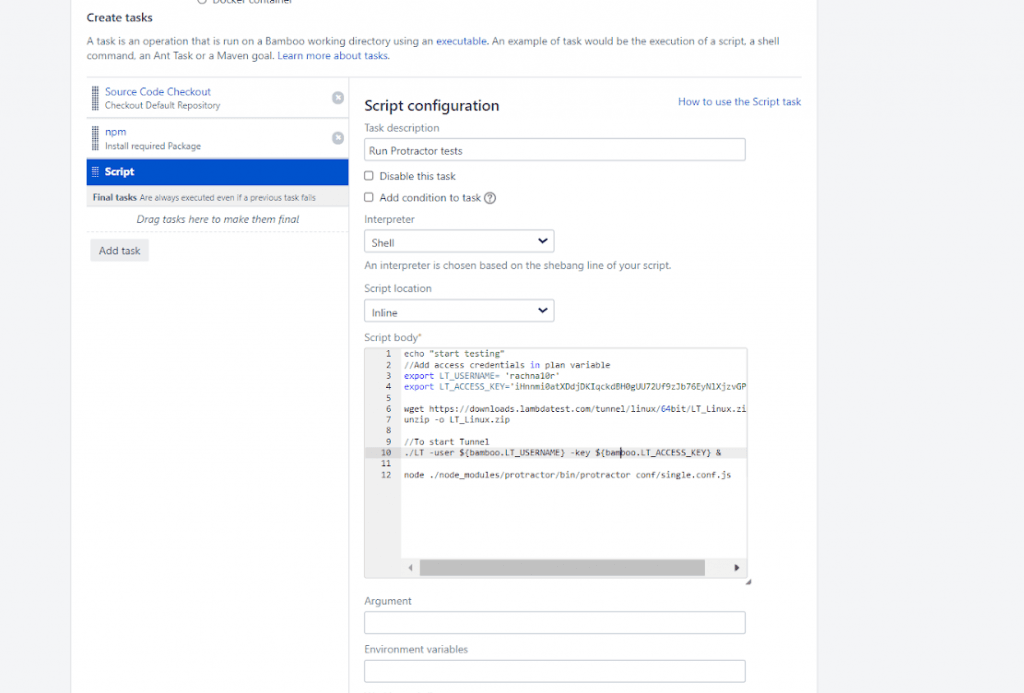
On click, you will find the WinrarZIP Archive with a binary file. Extract this zip to a new folder, and execute the command given above to start the Tunnel after navigating to that folder.

1. Use the below command to connect the application to LambdaTest Selenium web automation platform:

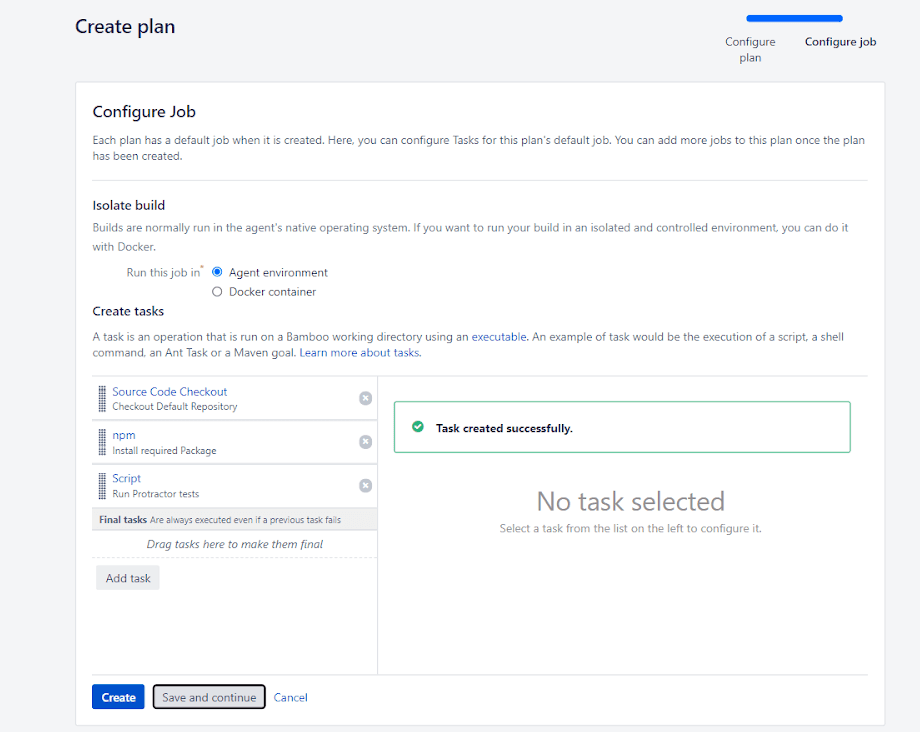
LT --user %bamboo.LT\_USERNAME% --key %bamboo.LT\_ACCESS\_KEY%

1. The last step in Bamboo CI/CD with the PHP project is the creation of the script. Click on ‘Add Task’ and select ‘Script’ from the popup dialog box. As soon as it is selected, a new task will be added below npm.

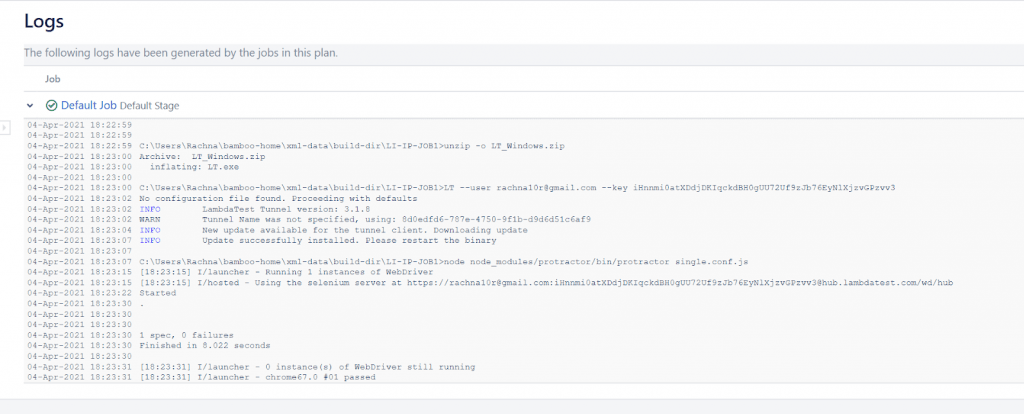
|  |
| --- |
| Task description : Run Protractor Tests |
| Interpreter : Shell |
| Script location : Inline |
| Script body : (This consist of script given above) |



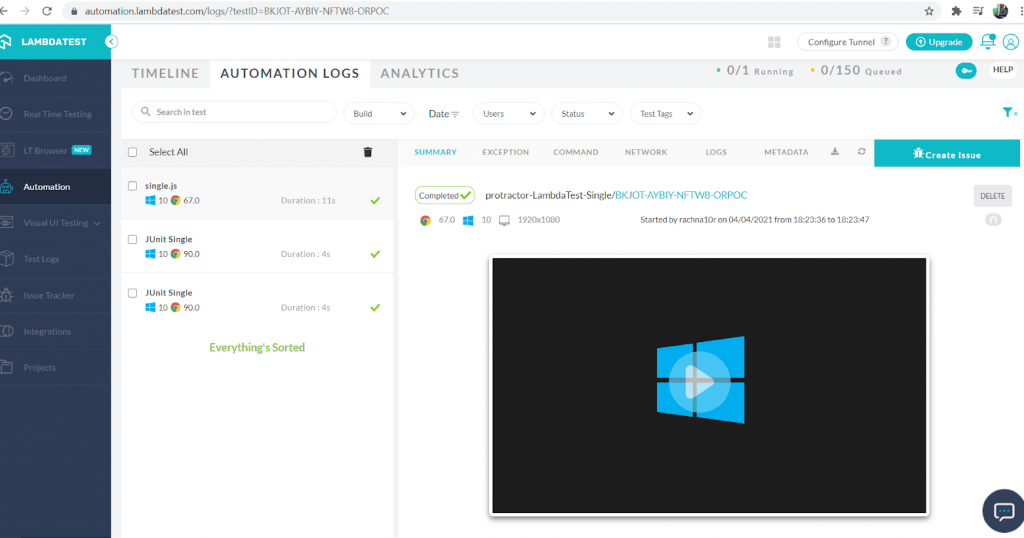
1. Click on Save and then select the Save and Continue button.



1. Enable the Plan and Run it by clicking the top-most ‘Run>Run Plan’ button. You will get all the details related to the execution of the Selenium web automation test in the tunnel.



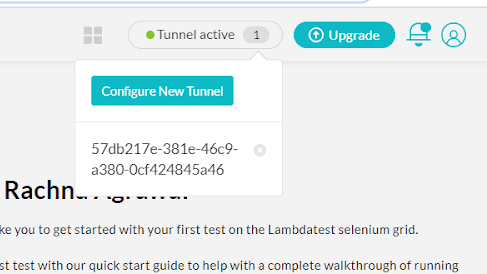
1. You can check the status of the test using the Automation Dashboard on LambdaTest.



You can also run multiple tests in parallel, which is the most demanding feature of the Selenium Grid. To run tests in parallel on LambdaTest, change the script body specified in the last step:

|  |
| --- |
| echo "start testing" |
| set LT\_USERNAME=<Your\_LambdaTest\_Username> |
| set LT\_ACCESS\_KEY=<Your\_LambdaTest\_Access\_Key> |
| node node\_modules/protractor/bin/protractor parallel.conf.js |

On the top right corner of LambdaTest Dashboard, you will see a Tunnel Active option, click on the same to monitor the test status.



**Wrapping Up !**

Kudos! Now, you know how to work with the most popular Continuous Integration Server – Bamboo CI/CD. You have also seen how it makes testing super easy and integrates perfectly with cloud-based Selenium Grid like LambdaTest. Bamboo CI/CD can be used with PHP projects to run Selenium automation tests as scale.

CI/CD with Bamboo helps enterprises utilize the power of automation via cloud-based Selenium Grid. Though we demonstrated CI/CD with Bamboo for the PHP project, the popular CI tool is usable with projects spanning other popular programming languages.

Be sure to check the [best CI/CD tools](https://www.lambdatest.com/blog/31-best-ci-cd-tools/) that help you build, test, and deploy at an accelerated pace. That’s all folks. I hope this tutorial would be a knowledge – packed bundle for all of you. If you find this useful, share it with your friends and on social media platforms too.