

CONTINUOUS INTEGRATION

TravisCI vs CircleCI vs Jenkins



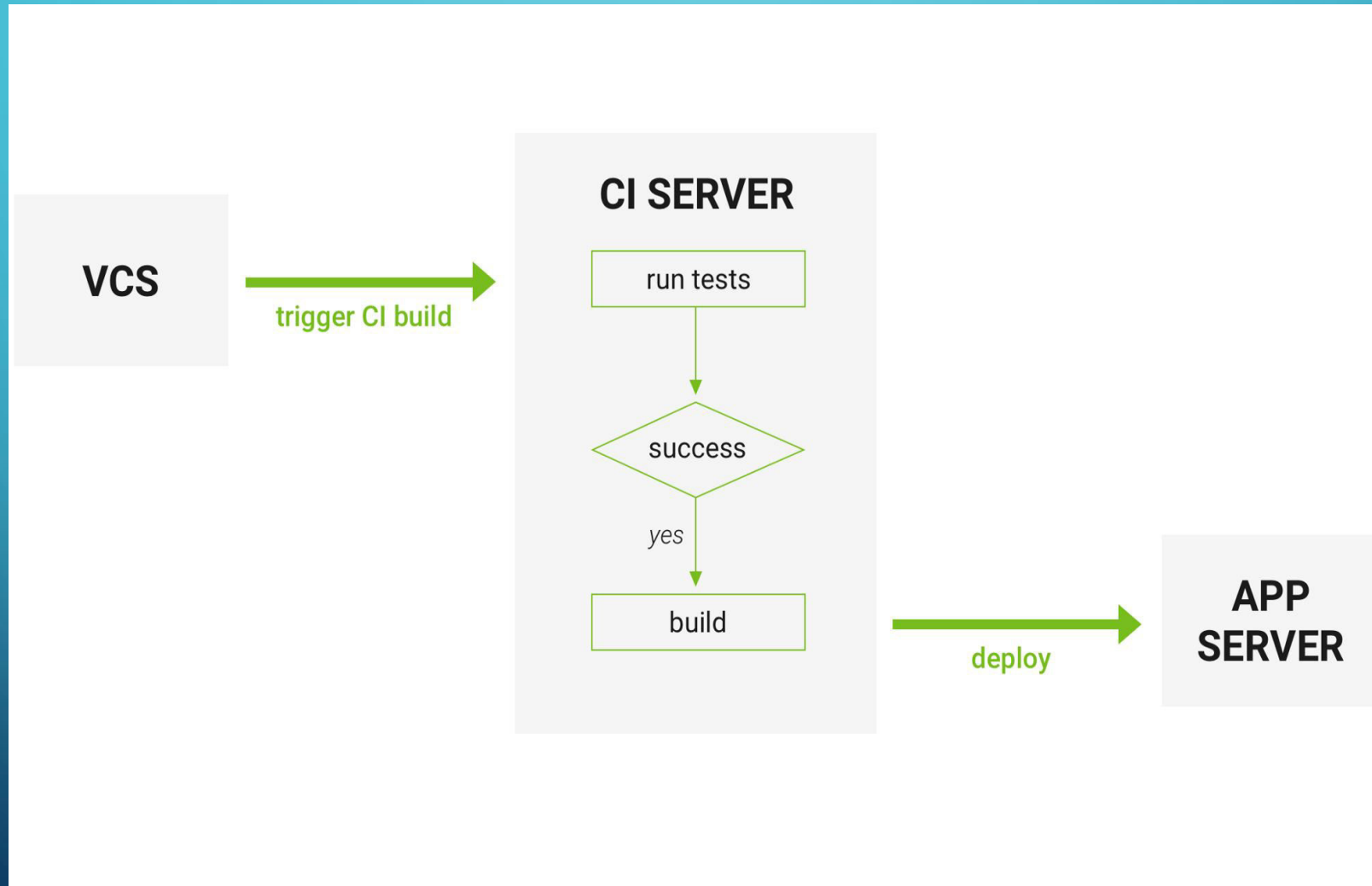
Travis CI



Jenkins

Rohit Chandra
Danish Siddiqui
Shivani Bhalchandra
Ahmet Bahcivan

CONTINUOUS INTEGRATION





Features :

- CircleCI is a cloud-based system—no dedicated server required, and you do not need to administrate it. However, it also offers an **on-prem** solution that allows you to run it in your private cloud or data center.
- It is free even for business account
- Rest API—you have an access to projects, build and artifacts The result of the build is going to be an artifact or the group of artifacts. Artifacts could be a compiled application or executable files (e.g. android APK) or metadata (e.g. information about the tests`success)
- CircleCI caches requirements installation. It checks 3rd party dependencies instead of constant installations of the environments needed
- You can trigger SSH mode to access container and make your own investigation (in case of any problems appear)
- That's a complete out of a box solution that needs minimal configuration\adjustments

CircleCI is compatible with:

- Python, Node.js, Ruby, Java, Go, etc
- Ubuntu (12.04, 14.04), Mac OS X (paid accounts)
- Github, Bitbucket
- AWS, Azure, Heroku, Docker, dedicated server
- Jira, HipChat, Slack



• CircleCI Pros:

- Fast start
- CircleCI has a free plan for enterprise projects
- It's easy and fast to start
- Lightweight, easily readable YAML config
- You do not need any dedicated server to run CircleCI

CircleCI Cons:

- CircleCI supports only 2 versions of Ubuntu for free (12.04 и 14.04) and MacOS as a paid part
- Despite the fact CircleCI do work with and run on all languages it supports only the following programming languages “out of the box”: Go (Golang), Haskell, Java, PHP, Python, Ruby/Rails, Scala
- Some problems may appear in case you would like to make customizations: you may need some 3rd party software to make those adjustments
- Also, while being a cloud-based system is a plus from one side, it can also stop supporting any software, and you won't be able to prevent that



Travis CI

Travis CI and CircleCI are almost the same

Both of them:

- Have YAML file as a config
- Are cloud-based
- Have support of Docker to run tests

What does TravisCI offer that CircleCI doesn't?

- Option to run tests on Linux and Mac OS X at same time
- Supports more languages out of the box:

Android, C, C#, C++, Clojure, Crystal, D, Dart, Erlang, Elixir, F#, Go, Groovy, Haskell, Haxe, Java, JavaScript (with Node.js), Julia, Objective-C, Perl, Perl6, PHP, Python, R, Ruby, Rust, Scala, Smalltalk, Visual Basic

- Support of build matrix / other CI's have TOX option to be able to build matrix



Travis CI

Travis CI Pros:

- Build matrix out of the box
- Fast start
- Lightweight YAML config
- Free plan for open-sourced projects
- No dedicated server required

Travis CI Cons:

- Price is higher compared to CircleCI, no free enterprise plan
- Customization (for some stuff you'll need 3rd parties)



Jenkins

Features:

- Jenkins is a self-contained Java-based program, ready to run out-of-the-box, with packages for Windows, Mac OS X and other Unix-like operating systems
- With hundreds of plugins in the Update Center, Jenkins integrates with practically every tool in the continuous integration and continuous delivery toolchain
- Jenkins can be extended via its plugin architecture, providing nearly infinite possibilities for what Jenkins can do
- Various job modes: Freestyle project, Pipeline, External Job, Multi-configuration project, Folder, GitHub Organization, Multibranch Pipeline
- Jenkins Pipeline. That's a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins. Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code" via the Pipeline DSL
- Allows you to launch builds with various conditions.
- You can run Jenkins with Libvirt, Kubernetes, Docker, and others.
- Rest API—have access to Controlling the amount of data you fetch, Fetch/Update config.xml, Delete a job, Retrieving all builds, Fetch/Update job description, Perform a build, Disable/Enable a job



Jenkins

Jenkins Pros:

- Price (it's free)
- Customization
- Plugins system
- Full control of the system

Jenkins Cons:

- Dedicated server (or several servers) are required. That results in additional expenses. For the server itself, DevOps, etc...
- Time needed for configuration / customization

CONCLUSION

- What CI system to chose? That depends on your needs and the way you are planning to use it.
- CircleCI is recommended for small projects, where the main goal is to start the integration as fast as possible.
- Travis CI is recommended for cases when you are working on the open-source projects, that should be tested in different environments.
- Jenkins is recommended for the big projects, where you need a lot of customizations that can be done by usage of various plugins. You may change almost everything here, still this process may take a while. If you are planning the quickest start with the CI system Jenkins might not be your choice.

RECOURSES

- <https://djangostars.com/blog/continuous-integration-circleci-vs-travisci-vs-jenkins/>