COMPARISONS ABOUT BUILD & DEPLOY SERVICE

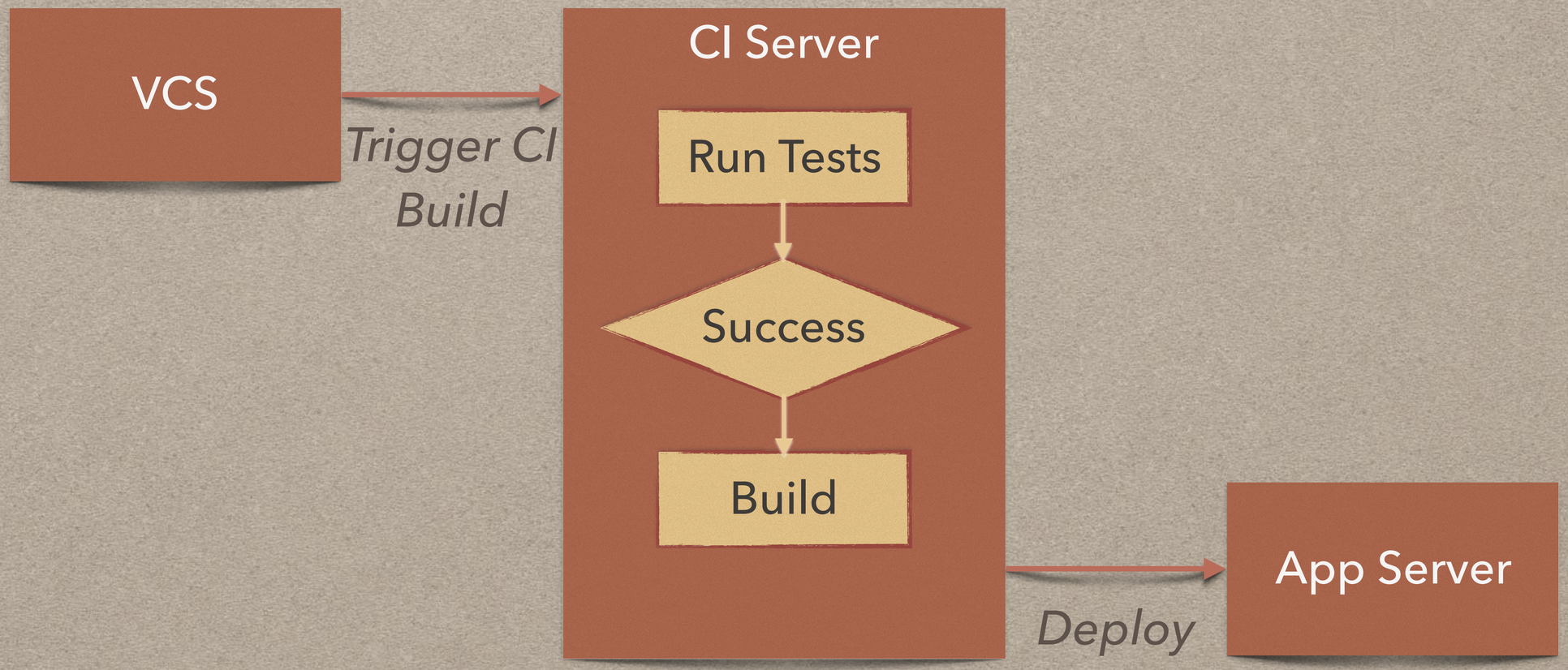
TRAVIS VS CIRCLE VS JENKINS

CI DEFINITION & ITS MAIN GOAL

Continuous Integration (CI) is a software development practice that is based on a frequent integration of the code into a shared repository. Each check-in is then verified by an automated build.

The main goal of continuous integration is to identify the problems that may occur during the development process earlier and more easily. If you integrate regularly – there is much less to check while looking for errors. That results in less time spent for debugging and more time for adding features. There is also an option to set up inspection of the code style, cyclomatic complexity (low complexity makes the testing process more simple) and other checks. That helps to minimize the efforts of the person responsible for the code review, saves time, and improves the quality of the code.

HOW IT WORKS


VCS
Trigger CI Build
CI Server
Run Tests
Success
Build
Deploy
App Server

HOW IT WORKS

- Developers check the code locally on their computers
- When completed – they commit changes to the repository
- Repository sends a request (webhook) to CI system
- CI server runs job (tests, coverage, check syntax and others)
- CI server releases saved artifacts for testing
- If the build or tests fail, the CI server alerts the team
- The team fixes the issue

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 has a free plan even for a 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

COMPATIBILITY

- 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

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

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 тт 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 FEATURES

- Almost similar to Circle CI
- Both of them:
  - Have YAML file as a config
  - Are cloud-based
  - Have support of Docker to run tests

Travis CI DIFFERENCE

- Travis CI offers while Circle CI doesn’t:
  - Option to run tests on Linux & Mac OS X at the 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

Travis CI BUILD MATRIX

- Build matrix is a tool that gives an opportunity to run tests with different versions of language and packages. You may customize it in different ways. For example, fails of some environments can trigger notifications but don’t fail all the build (that’s helpful for development versions of packages)

```
language: python
python:
  - "2.7"
  - "3.4"
  - "3.5"
env:
  - DJANGO='django>=1.8,<1.9'
  - DJANGO='django>=1.9,<1.10'
  - DJANGO='django>=1.10,<1.11'
  - DJANGO='https://github.com/
django/django/archive/
master.tar.gz'
matrix:
  allow_failures:
    - env: DJANGO='https://
github.com/django/django/archive/
master.tar.gz'
```

Travis CI TOX

- In case you prefer any other CI platform – there is always an option to create a Build Matrix by using Tox.
- Tox is a generic virtualenv management and test command line tool. You may install it by using **pip install tox** or **easy_install tox** command.

```
[tox]
envlist = py{27,34,35}-
django{18,19,110,master}
[testenv]
deps =
    py{27,34,35}: -rrequirements/
test.txt
    django18: Django>=1.8,<1.9
    django19: Django>=1.9,<1.10
    django110: Django>=1.10,<1.11
    djangomaster: https://
github.com/django/django/archive/
master.tar.gz
commands = ./runtests.py
[testenv:py27-djangomaster]
ignore_outcome = True
```

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

- 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

Jenkins FEATURES

- 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

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.