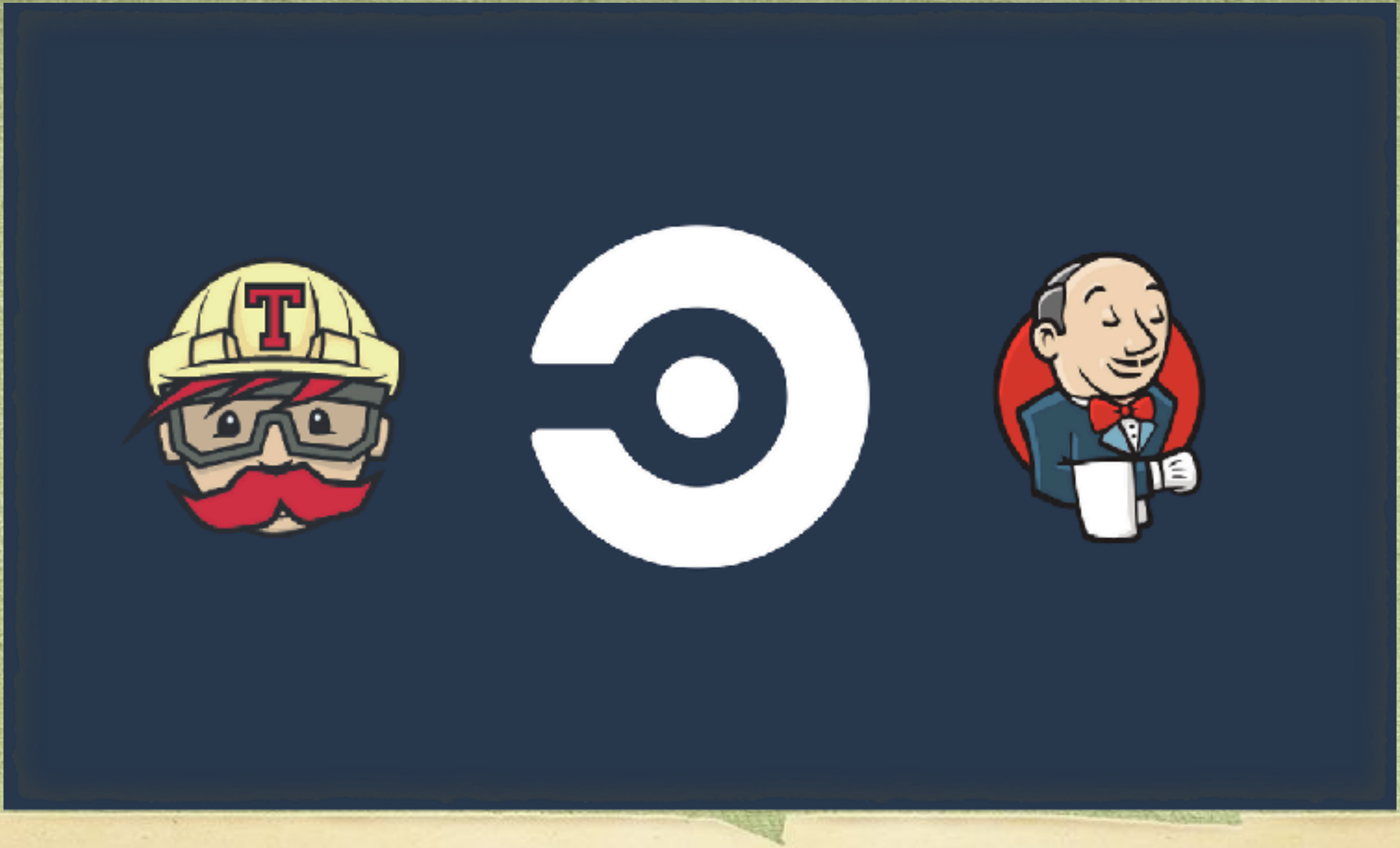
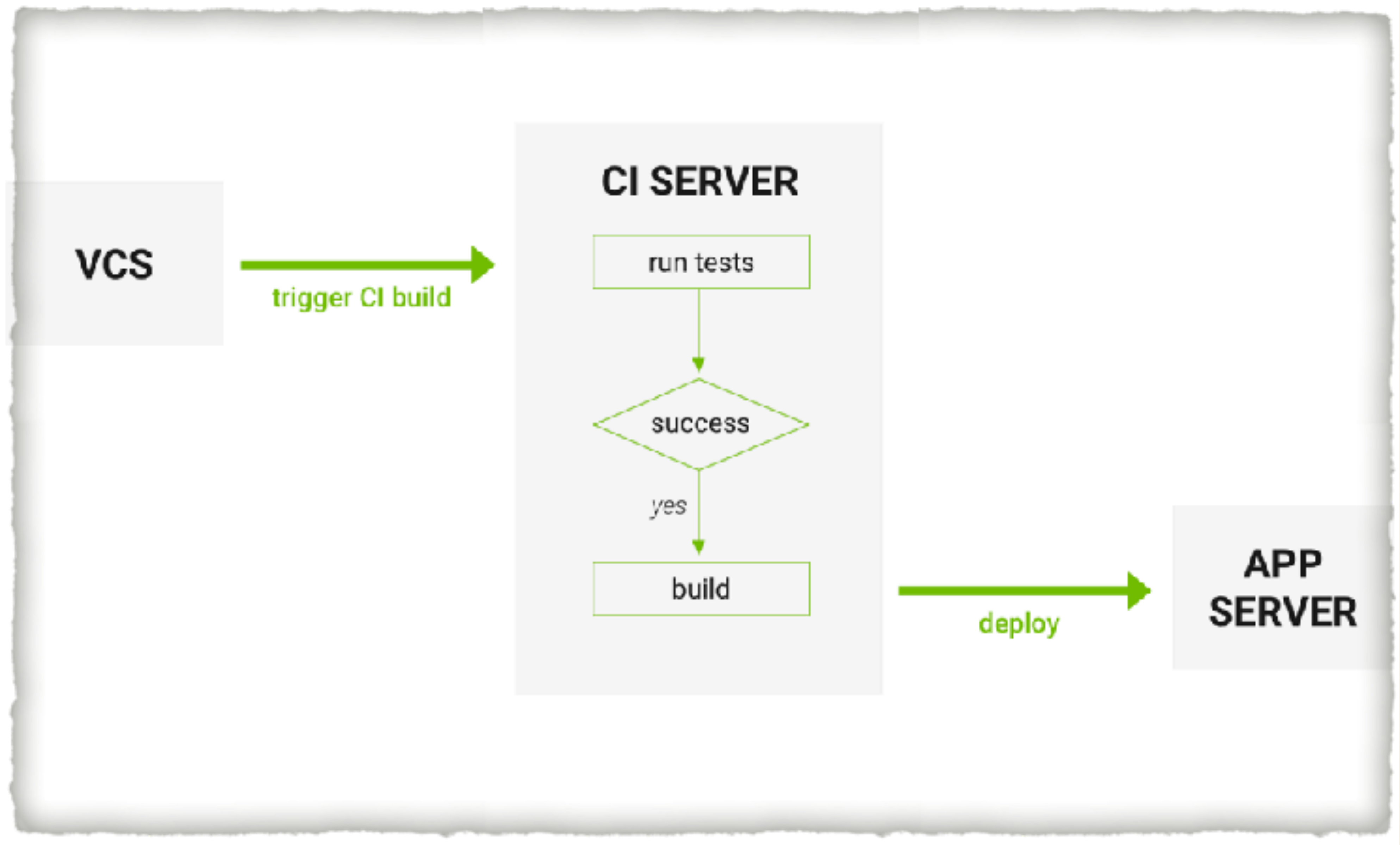


Travis CI vs Circle CI vs Jenkins

Ruixin Li

How CI works?


VCS
trigger CI build
CI SERVER
run tests
success
yes
build
deploy
APP SERVER

How CI works?

1. Developers check the code locally on their computers
2. When completed — they commit changes to the repository
3. Repository sends a request to CI system
4. CI server runs job (tests, coverage, check syntax and others)
5. CI server releases saved artifacts for testing
6. If the build or tests fail, the CI server alerts the team
7. The team fixes the issue

Overview

CircleCI: 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. Free plan even for enterprise accounts.

TravisCI: Travis CI is one of the more common names in the CI/CD ecosystem, created for open source projects and then expanded to closed source projects over the years. It's focused on the CI level, improving the performance of the build process with automated testing and an alert system.Price is higher compared to CircleCI, no free enterprise plan.

Jenkins: Just like a CI tool promises, with Jenkins you can automate your build, test and deploy tasks. The tool supports Windows, Mac OSX and various Unix systems, and can be installed using native system packages, as well as Docker or installed as a standalone on any machine with a Java Runtime Environment (JRE) installed. It's free for individual and enterprise.

Similarities:

——CircleCI and Jenkins

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. 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)

Similarities:

— —CircleCI and Travis CI

1. Have YAML file as a config

2. Are cloud-based

3. Have support of Docker to run tests

Differences

—system model

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.

Travis CI is basically same with CircleCI.

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

Differences

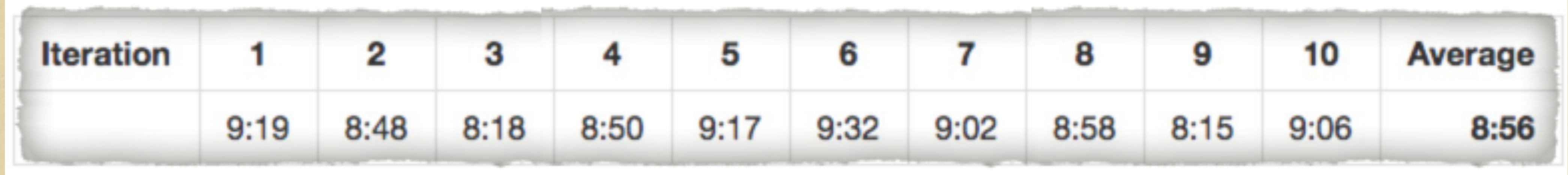
— —Initial setup

- CircleCI:Sign up requires only an authorization of CircleCI with Github. You will then see a list of repositories, so you can create the project with one click. The system looks for a file called circle.yml to determine what steps conform the build. The documentationis very helpful, providing many examples for the most common languages and features. Setting the node version only requires a statement.
- TravisCI:The first thing to learn about TravisCI is that there are two sites: Travis-CI.com for enterprise clients and Travis-CI.org for open source projects. Immediately, I discovered a big delay, as I found out that Travis comes with a really old C++ compiler that doesn't support `C++11`. This is important because `nodejs` and many other languages compile libraries/gems using C++, so you will get strange errors that don't seem to relate to your code. Finally, you must add a file called .travis.yml to your repo and Travis will start triggering builds when code is committed to Github. TravisCI does have good documentation and community base.
- Starting from scratch with Jenkins may be the worst option for a development group without sysadmin experience. However, at the same time, it can be a totally viable and desirable option for a group with sysadmin experience. This is due to the ability to customize the service, integrate it with slack or other services, and get better performance over the hosted offerings. It is worth to mention that Jenkins 2.x comes with a wizard that facilitates the setup.

Differences

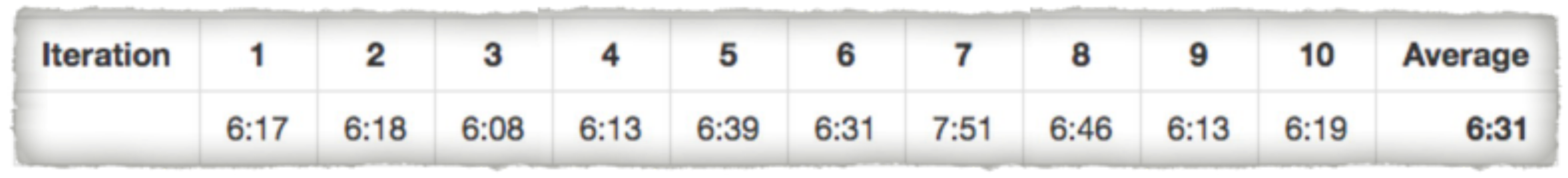
——Performance

CircleCI:



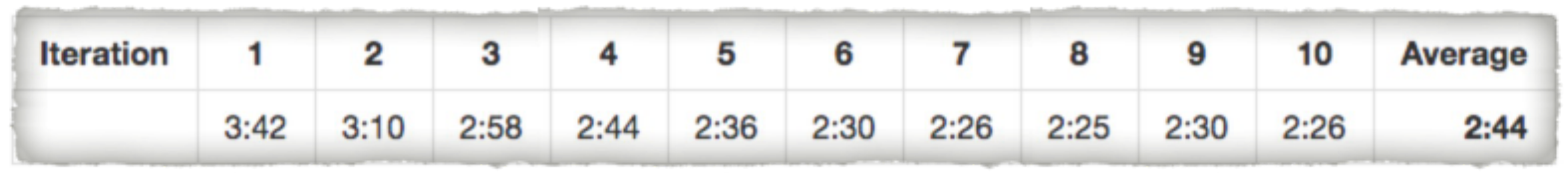
Iteration	1	2	3	4	5	6	7	8	9	10	Average
	9:19	8:48	8:18	8:50	9:17	9:32	9:02	8:58	8:15	9:06	**8:56**

TravisCI:



Iteration	1	2	3	4	5	6	7	8	9	10	Average
	6:17	6:18	6:08	6:13	6:39	6:31	7:51	6:46	6:13	6:19	**6:31**

Jenkins:



Iteration	1	2	3	4	5	6	7	8	9	10	Average
	3:42	3:10	2:58	2:44	2:36	2:30	2:26	2:25	2:30	2:26	**2:44**

Differences

—Service

CircleCI: CircleCI provides many services out of the box and it's bounded to localhost. The list includes: Postgres, MySQL, MongoDB, Cassandra, redis, elasticsearch, rabbit-mq, etc, so for this application nothing is required.

Travis CI: Services in Travis work similar to CircleCI, with the exception that in Travis these are not started by default. So, you need to specify which services you want load from this list: MySQL, Postgres, MariaDB, MongoDB, CouchDB, RabbitMQ, memcached, Redis, Cassandra, Neo4j, ElasticSearch, RethinkDB, etc. For each of these Travis supports a number of versions.

Jenkins: You have to manually install services you want to use. This has to be considered when setting the specs of your server, because Jenkins requires memory and disk depending on your use, and that adds up with the memory, disk and cpu required by the different services.

Some comparisons between TravisCI and CircleCI —Building Box

- CircleCI: Circle by default allows for SSH connections while the build is running provided you have SSH keys installed as part of your build. Additionally, the build can be re-run with ssh enabled allowing for 30 minutes of connection time after a build is complete. However, this utilizes a build slot while active preventing other builds from progressing if there's no other capacity.
- Travis CI: Travis used to have the ability to download vagrant files of the build box. This was amazing in that you could download the box and spin it up locally, and if you made changes you could destroy it and recreate the box. Unfortunately, it appears they no longer offer this feature which leaves a hole in the ability to debug issues They do offer a debug VM that they can make available for 24 hours. But you have to manually request it and wait for their support cycle.

Some comparisons between TravisCI and CircleCI —Build artifacts

- CircleCI: CircleCI supports automatic uploading of build artifacts to a built-in repository, and/or alternate locations. Easily accessible in the UI for that build via an "Artifacts" link as well as the CircleCI API. The artifacts have an indefinite retention period, and the feature is included in the base cost.
- Travis CI: Travis supports automatically uploading of build artifacts, but requires an S3 bucket to do so. While configuring this feature is simple enough, takes seconds to create an S3 bucket and to configure the artifacts themselves, there is an added cost of $0.03/GB/month. Plus the overhead of having to support another system.

Some comparisons between TravisCI and CircleCI

—Deployment Support

- CircleCI: CircleCI provides out of the box support for deployments to S3 and CodeDeploy. CodeDeploy may better fit our needs further down the road, but at the time of writing this it’s not part of our deployment stack. The S3 support appears to be somewhat limited and will require more effort than Travis to ensure it is properly used.
- Travis CI: Travis provides out of the box support for deployments to S3 and Elastic Beanstalk. While Elastic Beanstalk is the direction we’re planning on going, the complexity of our needs around this preclude our using this functionality. The bigger story here is the S3 support as it is simple and straightforward.

Pros

	Jenkins	TravisCI	CircleCI
Pros	• Hosted internally • Free open source • Great to build, deploy or launch anything async • Tons of integrations • Rich set of plugins with good documentation • Has support for build pipelines • Open source and tons of integrations • It is open-source • Easy setup • Workflow plugin •	• Github integration • Free for open source • Easy to get started • Nice interface • Automatic deployment • Tutorials for each programming language • Friendly folks • Osx support • Support for multiple ruby versions • Easy handling of secret keys •	• Github integration • Easy setup • Fast builds • Competitively priced • Slack integration • Docker support • Awesome UI • Great customer support • Ios support • Hipchat integration

Cons

	Jenkins	TravisCI	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 tt supports only the following programming languages “out of the box”:	- Price is higher compared to CircleCI, no free enterprise plan - Customization (for some stuff you’ll need 3rd parties) -	- Dedicated server (or several servers) are required. That results in additional expenses. For the server itself, DevOps, etc… - Time needed for configuration / customization

Thank you for listening!