**Evaluation Criteria**

Ease of Installation:

* How easy is it to install each CI

Travis was very easy as you don’t even need download it to your local machine. It is also hosted on [www.travis-ci-com](http://www.travis-ci-com)

How Intuitive is it:

* Is it GUI Based or Terminal based?

Travis features both a GUI to manage and view your builds in the form of [www.travis-ci-com](http://www.travis-ci-com) and a feature complete command line interface

* If it’s GUI based:
  + How difficult is it to navigate

The GUI is very easy to use, your homepage immediately presents you with your previous and running builds. It allows your to view the builds on your current branch as well as your other branches.

A link to the documentation is also immediately available on the homepage.

It has no sub menus on the GUI.

It’s very hard to make a mistake in travis CI, if you mess up your .travis.yml or build.xml files you can easily just re-open them, fix the issue in the file and then commit the change to your GitHub.

The GUI is simple and presents just enough information at any one time that you don’t feel overwhelmed just trying to use the GUI.

Of the three CIs Travis takes the cake for having the best look and feel, unlike the other two it actually feels like a professional website. It manages to be aesthetically pleasing whilst not being too flashy.

* If it’s Terminal based

As mentioned previously Travis includes a fully feature CLI which has its documentation available here: <https://github.com/travis-ci/travis.rb#readme>

The issue with Travis’s CLI is that it is not available by default due to the main features of Travis being a web app as a result the CLI must be downloaded from the above linked repo.

All of the commands are documented in the ReadMe available on the GitHub. Each command and its description is listed in the ReadMe with a link available to further documentation for that specific command for those who require more information on each individual command.

The commands are very understandable with each command accurately reflecting the feature it is used for.

Eg. enable – This command enables a project

encrypt – This command encrypts the values of your yml file

Is it well documented:

The entirety of the documentation for the main web app is available on <https://docs.travis-ci.com/> with documentation on the CLI available on <https://github.com/travis-ci/travis.rb#readme>

On the documentation homepage you have the option to follow a beginners tutorial which runs you through the basics of installing Travis CI in your git repo and connecting your Travis project to your GitHub repo. It also includes detailed tutorials on each of the basic tasks outside of installation.

However if you aren’t interested in following step by step tutorials it also offers general documentation for each function include deployment guides and language guides.

While the documentation is comprehensive each one for the most part reads like a step by step tutorial, which for some might be great. But for those that already know what they are doing for the most part it can slow them down when they just want to find relevant information.

How quick is it to build and test?

Travis is quite long to build and test with it averaging at 1:10, compared the the average of 5s with Jenkins this is quite poor.

Is it open or closed source?

Travis CI is closed source

What are the system requirements:

As it is hosted on a remote server Travis requires no memory on the local machine, it also requires no disk space and will support any operating system provided it is capable of supporting a web browser.

However one downside to it being hosted online is that it cannot be ran locally.

How well does it scale:

Travis is ideal for both large scale and small scale projects, the reason I say this is that it is quick to set up, only requiring you add a build.xml and a .travis.yml to the branch in your repo you wish to add and then connecting your repo to your travis project on [www.travis-ci.com](http://www.travis-ci.com)