

Step 1.

1. **Jenkins** is one of the most popular free open-source CI solutions. It is used in software engineering. It is written in Java and requires a web server to operate. Some of the highlights of Jenkins is that it is completely free, easy to install thanks to the pre-installed OS X, Unix and Windows packages, as many features and plugins, and is made by developers for developers. The reputation is enough for me to try it. I think the team would be able to benefit using Jenkins. It's free and has many features so that we won't waste money by buying a program for the team in case it doesn't work out.
2. **Raygun** is a tool for real time error monitoring. Key features include but not limited to seeing the exact line of code that caused an error, support for all major languages and frameworks, easy setup using light weight SDKs, groups errors by affected users, and filters though your errors by date, time, version, tag, host, OS, browser, custom tags, and more. The software/development teams would benefit from Raygun as its cheap 4\$ charged every 10,000 events if you get a year subscription or 6\$ every month. It is supported in a lot of languages and frameworks we use on our day to day. The list of integrations is all big with the likes of GitHub, Jira Software, Slack and others. With the ability to see the exact line where the errors might occur it will cut down on the time spent hunting the bugs in our code.
3. Installing Raygun, sign up, link the application you've made. Send data to crash reporting.
<https://raygun.com/documentation/product-guides/crash-reporting/introduction/>
Installing Jenkins. Download the package based on your OS (windows). Configuration is next then add any plugins you think you need for integration and continuous delivery. <https://www.jenkins.io/>
4. Jenkins was made on February 2nd 2011. As of February 2018 Jenkins has over 1.65 million active users around the world. They have an active github repo. Raygun is trusted by microsoft and coca cola to name the popular ones. It has 100,000 plus in over 120 countries. They also have a github.

Step 2.

The insert function took about 1 second (844.3469 ms) while the append took only (3.0334 ms).

Results for append function:

tiny: 7.7 µs

small: 7.9 µs

medium: 56.8 µs

large: 532.1 µs

Results for the unshift method

tiny: 11 µs

small: 14.3 µs

medium: 148.6 µs

large: 8.7087 ms

In the beginning the difference in run times is not noticeable to the human eye. Once you get to the bigger functions the unshift function gets increasingly slower. The difference is so enormous that one is measured still in milliseconds while the unshift took almost a full second to run. Even in the large array the push method is in the microseconds while unshift is into the milliseconds range.

The unshift method is significantly slower because of the number of steps the computer has to take on the backend just to make one new insertion into the array.