Continuous Integration

Jenkins

Features

- Easy to install (good for new employees)
- Open source (free!)
- 1500+ plugins (versatile)
- Easy Distribution (Easy to work with multiple people/ teams)

Getting Started

It's easy with a simple windows installer with the Installation setup wizard. Set the destination, decide a port number, and you're good to go! Then you go to what port you set up to unlock Jenkins. The password will be in the filepath of your download. You can then customize Jenkins with different plugins either suggested by Jenkins or you can select which plugins you'd like to use. And then you'd be good to go to start using Jenkins. They have documentation to show specifics on how to use the program with projects. They also have a "Guided Tour" to help you create your first pipeline as a tutorial!

Jenkins was first released February 2, 2011, and there seem to be a fair amount of people uploading projects with Jenkins on Github. With a quick google search you can see Jenkins is one of the more popular CI tools out there.

Real Time Error Monitoring

Raygun

Features

- Complete visibility of your tech stack
- Code-level diagnostics
- Prioritize errors
- Able to unify all your monitoring from Customer experience and deployment

Getting Started

There is a guide to help with installation with each programming language, and how to integrate it into your code. For example, with react you start by including the script. Then you configure Raygun, and then set up tracking route changes. Then there are instructions on how to use react router alongside browserrouter to automatically track changes. And they also show optional additions you can add to track even more.

Raygun was founded in 2007, They service large companies such as Coca-cola, Domino's Pizza, Microsoft, and Samsung. They have an impressive clientele that speaks to their reliability. There are quite a few repositories on Github using Raygun, and has a good popularity in the coding community.

Runtime Analysis

tinyArray
Append: 90.6 µs
Insert: 34.3 µs

smallArray
Append: 103.8 µs
Insert: 47.9 µs

mediumArray
Append: 135.9 µs
Insert: 188.9 µs

largeArray
Append: 650.5 µs
Insert: 9.3784 ms

extraLargeArray
Append: 2.7731 ms
Insert: 893.993 ms

When observed you can see that inserting (.unshift) is quicker with smaller arrays, but relatively quickly appending (.push) becomes quicker than inserting. So appending is linear and inserting is quadratic. So it is smarter/ better practice to append, because the larger the data you're working with more effective it will be over inserting. This is because when you're inserting you're altering data for each index, rather than when you append you just add on indexes rather than altering.