Alternative Tools Research

Continuous Integration alternative: AWS CodePipeline

AWS CodePipeline
AWS CodePipeline is a fully managed continuous integration and delivery service that helps you automate release pipelines for fast and reliable application updates. This software automates the build, test and deploy process every time a new piece of code is pushed to the repository. This program can easily be integrated with third party sources like GitHub. If you are deploying an application with AWS, this is the CI/CD program to use.

Getting Started - AWS CodePipeline
The AWS CodePipeline documentation can be found here. Although the process for setting up a pipeline (or as AWS calls it an "S3 bucket") is very long, it is also very straightforward. The process of setting up the software is very well documented and makes for a very user friendly setup process.

Maturity of AWS CodePipeline
This CI/CD program was introduced in 2015. As with all Amazon products, the program is tested and is widely used throughout the software development community. AWS CodePipeline is the preferred method of continuous integration and deployment for companies across the globe.

Real Time Error Monitoring Alternative: Raygun

Raygun
Raygun is an error monitoring software program that allows you to diagnose problems in your codebase. Once Raygun is added to your code, you will be able to monitor every error in the project as soon as they happen. Additionally, you are able to sort through errors to find those that have the biggest impact on the user's experience. Once an error is located, you are able to see the full stack trace, environment, browser, version, OS and identify the commit that introduced the issue.

Getting Started - Raygun
The Raygun documentation can be found here. Raygun offers integration for every major language and OS. Once you click on the "Language Guides" section of the documentation, just click the language you intend to use for your project. Once you specify the language, Raygun outlines what exactly you need to install for that specific language to integrate the error handling software.

Maturity of Raygun
Raygun has over 100,000 users and is trusted by the world's leading software teams. They have 50 billion+ errors processed for their customers, 100,000+ developers using Raygun across 120 countries.

Part 2: Runtime.js

extraLargeArray results:

The function “doublerAppend” is much faster than “doublerInsert” because it uses the push method instead of unshift. When we push an element, we add it to the end of the array thus leaving the elements before it unaffected. Alternatively, the unshift method adds an element to the beginning of the array which then means that all of the elements after it have to be reassigned an index value. This results in the insert function taking over 1s to finish as opposed to the append function which only takes 7ms.

DoublerAppend()	DoublerInsert
tinyArray - 79.25 us	tinyArray - 30.59 us
smallArray - 98.92 us	smallArray - 45.04 us
mediumArray - 146.417	mediumArray - 211.75 us
largeArray - 153.63 us	largeArray - 231.2 us
extraLargeArray - 7.18 ms	extraLargeArray - 1.07 s

Results Reflection

The append function (using the push method) is slower for the tiny and small array, but is faster for the medium, large and extra large arrays. The insert function was consistently becoming slower than the append function based on how many elements were in the array. This is happening because the unshift method adds elements to the beginning of the array which forces all elements after that to be reassigned an index value. This is different from the push method because adding an element to the end of the array does not affect the element’s index position at the beginning. Therefore, the push method scales more efficiently than the unshift method.