2. Buddy - Continuous Integration tool totes one of the most seamless integrations with github and bitbucket. Connecting our repos to our continuous integration tool will not need to be second guessed. Buddy is a main competitor to CircleCi and with their empowered uploads and lower deployment failure rate, Buddy has a huge advantage of allowing developers to deploy 46 times faster than CircleCi and other classic CI/CD tools. Their deployment failure rate is 5 times less than traditional tools, and not only is it easily configured they also offer a scripted and SCRIPTLESS configuration method. Imagine that, scriptless.

Buddy offers dependency caching. Dependencies are bulky and take time to load, this explains their fast loading rates. Traditional CI/CD tools are limited in testing by having to read text in logs and error messages, Buddy offers this approach of course but also adds visual testing, job conditions, and free technical support. Integrating from one tool to another can be stressful, especially learning how to get started quickly and effectively. Buddy curbs set up time, and with free technical support, implementation should have minimal downtime.

Raygun offers error monitoring and crash reporting. With minimal setup, Rayun can be configured for our entire stack. We no longer rely solely on log files or support tickets with incomplete information. Raygun dives deep into eros and crashes which narrows down the exact line of code while tracing the entire environment along the way. Their integrations with git are simple and seamless. Alerts can be issued through slack to notify team members immediately and errors can be searched by customers even.

Using graphs and visuals are a huge part of Rayguns toolkit as well, we can track trends in our crash monitoring, get insights into server-side crashes, errors, and performance issues. These visuals help us find bottlenecks, errors , and slow-running queries easily with detail into our APM.

3. Getting started with buddy is insanely easy https://buddy.works/docs/quickstart/javascript Following this link gives in depth instructions with images to help visualize the process. Simply you select your repo, enable On Push delivery pipelines which will tell Buddy to fetch your code once it is pushed to the repo. Select the actions Buddy should take once the code is fetched, there are dozens of options ranging from Slack alerts to building and testing the app. Thats all it takes to get started with buddy. Notice there is no YAML config here, however they do offer that type of configuration method as well.

Getting started with Raygun is relatively easily as well, the getting started guide also has visuals and text instructions to make it easy and visual. Simply add script tags to you head tag of HTML (or you can npm install it), iud we do use the CDN then we add additional script tags in the HTML head. One more real user monitoring, then a tag for crash reporting, then user tracking and handling user logout (these last two steps are optional). Raygun also supports SPA user monitoring which is nice for libraries such as React and Vue.js, and that is it! Youre ready to go with Raygun!
https://raygun.com/documentation/language-guides/javascript/real-user-monitoring/installation/

4. Buddy seems to have its first github repo initial commit 6 years ago and Raygun shows its first around 5 years ago. While both are relatively young, they are touted as newer and more

modern tools. Link to Buddys customer stories: https://buddy.works/about. Link to Rayguns customer stories: https://raygun.com/customer-stories.