**DataDog:**

**Getting Started:**

**The Getting started section is easy to find. Button in top right of screen.**

**Easily Visible. Click on the button and bring’s a small window. Easy form to fill out or just sign up with your Google Account. Second step of Getting started brings you to a page to Describe your Stack. Lots of options and easily choose those you may be using. Giving you options to describe why you are going to use the tool, as well as describe maybe some of the stack options that aren’t there. Last page of the Getting started has easy step by step instructions to get your first agent installed on a variety of different platforms. Easy to see it’s flexible where you want to use this program!**

**After it’s installed, it brings you to a homepage, where I was able to set dark mode easy which is awesome on the eyes. From this page, you can locate just about any tool you need. Create your own Dashboard, link stacks, add new things to your stacks, Invite people to help you, install integration and so much more! This is an amazing tool!**

**DataDog History:**

**DataDog was founded in 2010 and has been climbing the charts ever since. It was originally Python Based but then was re-written to now being a Go based agent. With over 350 integrations possible it is very versatile and very useful on many scales and may needs in the tech industry! How Popular is it? Well at over 2,500 employees and in 2021 Nasdaq is reporting 802 million in profit I would say its popular enough to give it a good try and thoroughly test what it could do for our company!**

**HoneyComb:**

**Getting Started:**

Upon entering the site, the Getting Started button is easy to find in the top right hand. This takes you to a new page to fill out the form to sign up or Easy sign up with Google. After Selecting your Google Account it takes you to a Create Team Page. You can join a previous team or create a unique team id so that you can invite others directly to help out or see your work! The Setup then takes you to the less than easy setup page. In order to get the whole process rolling, you have to Choose an app of yours that you would like to report data and observe that data, go into in the development field(example: through the command line) and then Using the commands they give, install the correct packages and add ons to your application. After you have taken the time to connect everything, start your app, and start making requests and then HoneyComb should start picking up those requests and report your data and also alert you of any errors that you may have missed.

**HoneyComb.io Histoy:**

Honeycomb was founded in 2016 and is based in the bay area of San Francisco. They have multiple rounds of funding and received plenty to continue funding their service. It is still rising in popularity and they are adding functionality constantly as well as smoothing their current operations out. Honeycomb is there to help you monitor data and errors for any walk of life. The reporting is clean and you can choose how to display your reports. Whether in graph form or just data lines they work to get you your data in fast, easy to read ways! May not be the most popular company, but with the presentation and actual resources they bring to the table, I think this would be a great tool for our DevOps teams! Maybe we could even head out to San Francisco for a tour! Beautiful town!

***Runtime Analysis***

|  |  |  |
| --- | --- | --- |
| Array Name: | Append | Insert |
| tinyArray | 148.6 us | 69.8 us |
| smallArray | 75.7 us | 183 us |
| mediumArray | 293.9 us | 376.3 us |
| largeArray | 1.3019 ms | 14.6963 ms |
| extraLargeArray | 5.0136 ms | 1.258 s |

**The runtime was an interesting bit. I saw that the .unshift() method is typically running faster than the .push() method. As the arrays get bigger, with more inputs and bigger numbers then the methods are taking longer to run and process the information. As they should, the results are actually crazy that almost all of the inputs are all processed and edited under a second with the XL being the only that ran over a second. The scaling is as I said before, as the inputs grow so does the processing time for the amount of inputs to be added to the array and put in the right spot of that array.**