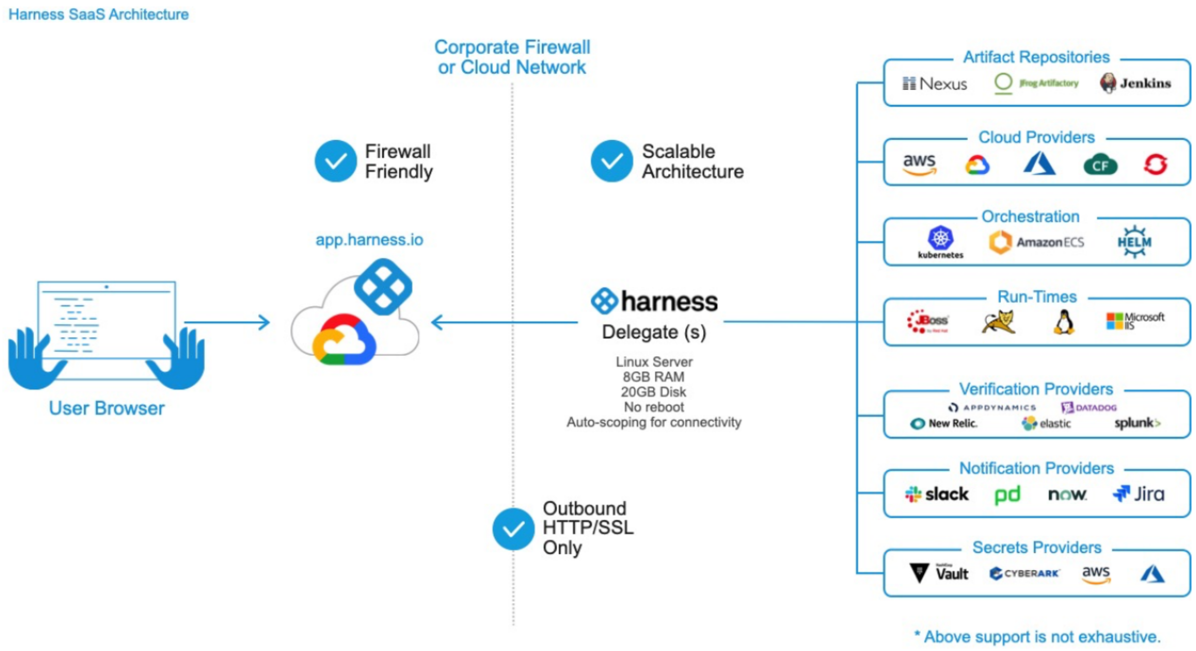
Harness

- Was designed with developers in mind and is set up in a layout that is very visual and easy to see your pipeline at all times.
- Everything is container based and language agnostic
- Has a YAML tab that uses an IDE type set up so you can work on YAML right in the platform without jumping back and forth to VS code
- Getting started
    - Docs are laid out with different topic headers and a table of contents for each header to make it easy to jump around different subjects in the docs.
    - The docs have a built in video where you can set up harness within “5 mins”



-
    - Docs are filled with a lot of visual diagrams to explain the topics being covered
- Harness is a relatively new start up company that launched back in 2019. While they haven’t been around for a long time yet they were rated as LinkedIn top startups in 2020. They’re already a Series C company with almost 500 employees

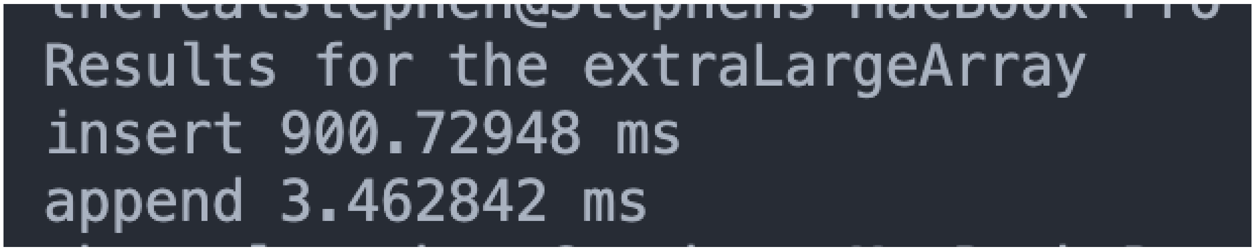
AppVeyor

- Extremely fast onboarding and being able to start use of the product
- Able to integrate with any source control
- Multi stage deployments.
- Getting started
    - First thing you see in the docs is a quick video giving you an intro into how everything works with steps to quickly add your first project.
    - Docs are easy to navigate and read overall, but there is a lot of content in them so it would take quite a bit of time to learn the whole platform

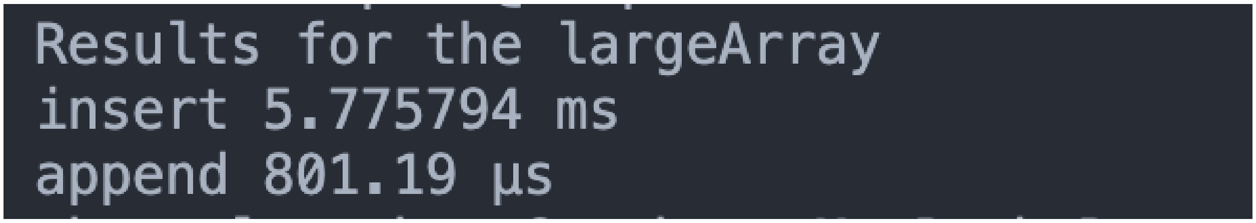
- AppVeyor appears to be a premium brand in this space and has landed tech giant customers like Google, Netflix, Facebook, Microsoft, and GitHub.
- They’ve deployed over 2.5 million Applications.

Raygun

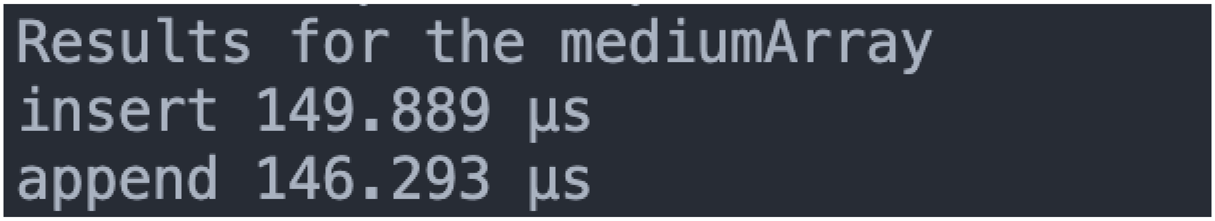
- Offers a free trial with no credit card required to be able to test it out.
- Supports all major programing languages
- Claims to allow you to set up in mins with just a couple lines of code
- Simple pricing model with charges of just $4/per user
- Getting started
  - Step by step instruction on getting Raygun set up
  - Full support team to help for those who have issues with self service
  - Documentation offers key word search to find specific content
- Several enterprise customers (Domino’s, HBO, Microsoft)



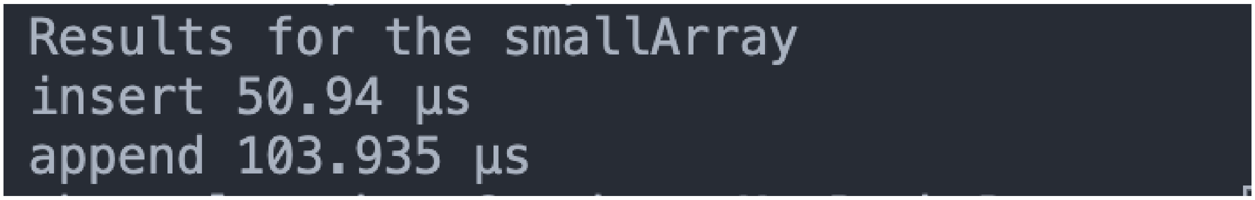
```
therealstephen@Stephens-MacBook-Pro
Results for the extraLargeArray
insert 900.72948 ms
append 3.462842 ms
```



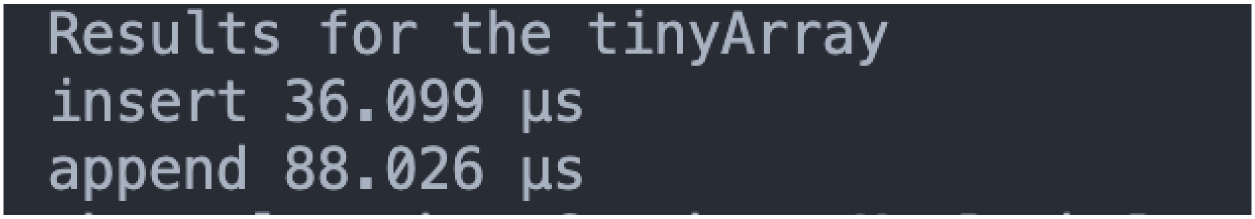
```
Results for the largeArray
insert 5.775794 ms
append 801.19 μs
```



```
Results for the mediumArray
insert 149.889 μs
append 146.293 μs
```



```
Results for the smallArray
insert 50.94 μs
append 103.935 μs
```



```
Results for the tinyArray
insert 36.099 μs
append 88.026 μs
```

Both functions perform pretty similar with the tinier arrays (with insert even being a little bit faster). However, as far as scaling goes append becomes much more efficient as we work with more data and bigger arrays.

The biggest reason for this I believe is that append uses .push() vs .unshift(). By using .unshift() it's adding the new value at the beginning of the array and then has to manually shift each value over one to make space for it. This causes it to loop through the array a second time and slows things down. On the other hand .push() just adds it to the end of the array and doesn't have to loop through the array again to shift all of the indexes.