Part One

Data Dog:
The data dog client won't open. After installing and spending 30 minutes trying to get it to work, it just won't open. Super frustrating. Will not be used.
Octopus:
After trying for nearly 75 minutes to get octopus to work, it continually keeps crashing and doesn't work. Will be trying something else.
AWS:
Aws was the easiest and finally got it to work compared to the other 3 that I tried. There were several guides that I found that worked. AWS was launched in 2002 and is the largest cloud based services. It generate 46 billion a year in 2020.

Realized that I did part one wrong because I didn't read through the description carefully enough…

Amazon Web Services:

Amazon Web Services has a large list of features. Computing, storage, networking, database, analytics, application services, deployment, management, machine learning, mobile, developer tools. Amazon elastic compute cloud, amazon simple storage service. And pipeline tools like CD/CI. I think it would be easy to convince someone to use AWS considering 60% of the internet is hosted on AWS.

There is an easy to find getting started page with sources to help. The benefits and features are rapid delivery, improved quality, configurable workflow, and fast setup(indeed!). There is a nice 3 minute youtube video shown that is an introduction to AWS codepipeline.

AWS has been around since 2002. Aws generated 46 billion dollars in 2020. There are over 1 million active users that use AWS. AWS has constantly grown by between 30-50% for the last 5 years. Its in 190 countries.

DataDog

DataDog is a monitoring service for cloud scale applications providing monitoring of servers, databases, tools, and services, through a SaaS(software as a service) based data analytics platform.

DataDog was founded in 2010. The features of datadog are the following: SaaS and cloud providing, automation tools, monitoring and instrumentation, source control and bug tracking, databases and common server components.

In 2021 datadog has 15000 active customers.

Part Two:

Results for extraLargeArray: insert 1 s, append 2.8468ms.
Results for tinyArray: Insert took 31.1us, append took 76.4us.
Result for smallArry: Insert 43.6, append 86.2us
Results for mediumArry: Insert 162.6us, append 128us
Results for largeArray : Insert 7.9279 ms, append 500.5us

Each array is getting 10 times longer with each array, so it's obviously taking a bit longer for each sized array. But the insert seems to be taking longer than the append method. Therefore the insert method would be less efficient. The append method scales better.