

What's the Deal with Docker? Some Stats from Datadog on Docker Adoption



There's a running joke where every tweet "needs more #docker". It's because of the nearly meteoric rise in popularity of the idea of containers as triggered by the Docker containers format.

As some may know, [Docker](#) came from dotCloud and was a containerization technology to enable them to work on a PaaS infrastructure. Much like the [accidental discovery of products like Post-It Notes](#), the container technology actually became the product.

The (un)necessary Disclaimer: Docker isn't for Everyone...yet

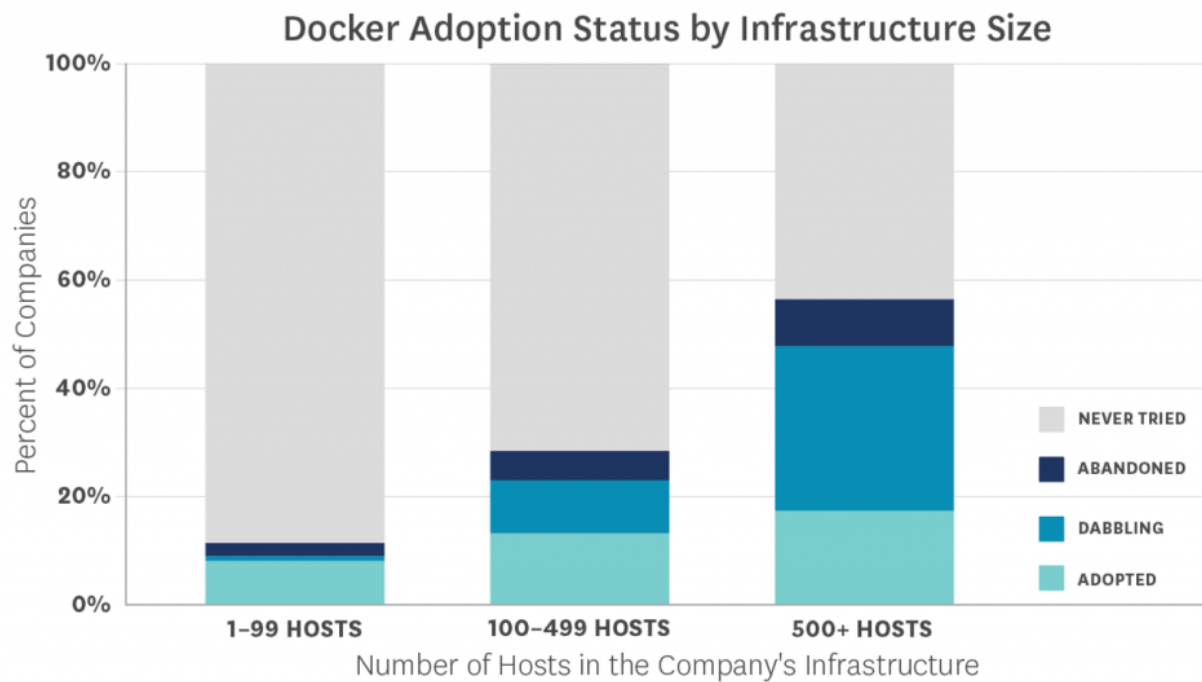
Before the container hate begins, let's just remember that containers aren't necessarily a strategy for everyone. Let's also remember that the same was stated about virtualization in the early 2000s. Including Docker containers in your IT portfolio requires that you evaluate the use-cases, and test the technology in your environment. In my opinion, it would be silly to discard it as a option without proving it out in practice.

Will SMB (Small to Medium Business) embrace Docker as a container strategy? Probably not first hand. They may consume a service that in turn uses Docker containers, but the cloud-native apps and microservices architectures are still more likely to be adopted by agile startups, and forward leaning enterprises who are looking towards gaining efficiencies with better development methodologies.

Docker stats from the field: a Datadog report

There was a recent blog put out by the team at Datadog which opened the proverbial kimono on their data pulled from environments under management by their product.

Some of the highlights that I was pleased to see included amount of increase, size of environments, and also the list of images that are the top for building.



The numbers in the chart illustrate that even with a relatively high untried rate, the adoption versus abandonment shows that Docker containers are sticking around in these environments. I particularly like that we see representation from environments of all sizes.

Container Methodology, VM Thinking

Number four in the list of most used images is Ubuntu. While this isn't necessarily a surprise given the broad adoption of Ubuntu around the world today in VM and bare-metal form, it did tell me one thing: people are building containers like VMs to start.

The reason that this is funny (funny peculiar, not funny haha) is that we are seeing the same practice with containers that we saw with early virtualization. There were much larger VMs used that carried a lot more overhead and resembled their predecessors, the good old physical machines. As a lot of newcomers to containers bring Docker into the lab, they will probably tend to draw on their existing build documents and processes for a while until they refactor and adapt to the more agile, thin container strategy that will drive up the density.

Density of 4:1 for containers as an average in the survey results also showed that there are tentative users who still are concerned about overprovisioning using containers in the way that we became used to when virtualization became mainstream and saw ratios of up to 40:1 for guest to hosts. Early virtualization was more like 4:1, so it isn't difficult to predict where it will go next.

Know the Audience

It is important for us to know who the target audience is for the survey, and clearly Datadog has given us the details. It is from their existing customer who will tend to be application focused.

One data point stood out for sure. As you can see from the median age of containers was 3 days. This shows that the portable, immutable, stateless nature of containers gave way to a naturally volatile deployment. This is precisely how they are meant to be used, however, this is just one data point.

When compared to the median VM lifetime of 12 days, we can see that this audience may already be embracing stateless application architecture and rapid, elastic deployments. When the traditional enterprise audience gets further down the road with Docker and container adoption, the numbers will be much higher on both sides in my opinion.

Thanks to the Datadog team for sharing this information, and I hope that you find it useful. As someone who is leaning in heavily with containers and the next generation of application architecture, I can tell you that this is the beginning of some exciting times!