



# 2017 Docker Usage Report

---

1 April 2017



Many Docker usage reports are based on user surveys. These are incredibly useful to understand intent and objectives, but can sometimes be inconsistent with how people are actually adopting Docker within their environments right now.

The main question we wanted to answer was, *“How are people using Docker in their application environments right now?”* As the premier container monitoring solution, Sysdig is in a fantastic position to answer this question with actual Docker usage data across hundreds of customers.

The data you see here represents a snapshot of our customer behavior in early spring 2017. We have aggregated the data across all environments to provide unique insight into how users are adopting containers right now. This is not meant to be a broad statement on the market as a whole - we can only speak to what we can report on from our own data.<sup>1</sup>

<sup>1</sup> Potential biases include: We are a commercial monitoring service, which may appeal to a segment of customers; We are reporting off our data from our cloud service, which also may appeal to a segment of customers; While we have had Docker container integration from the beginning, our Kubernetes integration is more mature than that of other orchestrators.

## Sample size of the Docker usage survey

There are lots of containers out there. A lot. We took a point-in-time snapshot of a representative sample of the containers we monitor via our Cloud Service for the purpose of this report. We also provide an on-premises container monitoring solution but that data is not included within the container usage survey results.

445,000

## Container density as a measure of adoption

One of the initial advantages that many teams see in containers is their ability to provide “lightweight virtualization” as compared to hypervisors. The lightweight nature implies that they can run at a high container density, make better use of underlying physical resources, and thereby save money.



We wanted to see if users were actually seeing this benefit. We asked the question, “Of customers running containers in their environments, what is the median number of containers being run?” Note this median is derived from the average number of containers per host per customer, which felt like a more accurate way to understand where customers are with regards to density. We also saw a very broad range of deployment densities.

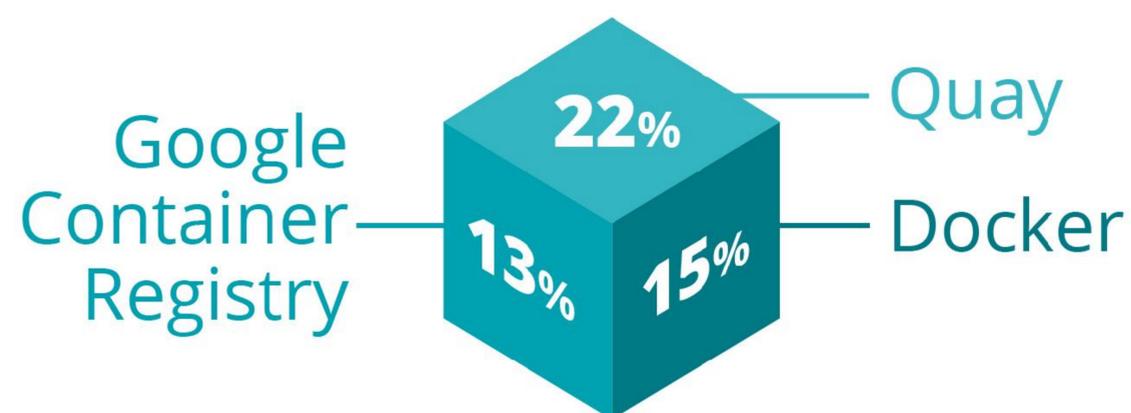
**We saw some customers running at 95 containers per host**, and others who are running at a single container per host. The latter situation, while it may sound unusual, has underlying logic. Talking to those customers, we have learned that their core software uses all the resources on the machine. The benefit for containerization to them is not providing container density. Rather, it’s the ability to develop, deploy, and scale software more quickly where Docker adds value.

## What Orchestrators do people use with Docker containers?



The popularity of Kubernetes is not surprising. In this stage of Docker adoption we saw Kubernetes take an early lead against other orchestrators. However, with the newer version of Swarm, and the increasing efforts behind Open DC/OS, we have seen an increased variety of orchestrator platforms in use. These platforms have not caught up to Kubernetes within our customer base yet, but we expect to see continued competition here as these platforms mature.

## Most popular container registry



A container registry is a place to store and distribute Docker images. You can think of a registry as a gas station - no matter where you go the gas is really the same, but one station might have decent coffee, another throws in a free car wash, and yet another is attached to the Safeway so you can get your groceries and gas at once. Anyone who is seriously using Docker containers will likely be using a registry, either as a service or as on premise software (aka a private registry.)

As you can see from the numbers above, even the top three combined don't constitute a majority of the Docker containers in use within the sample. That's indicative of how fragmented the registry offerings are. There are two footnotes to this data: (1) these percentages are based on container count, not based on registry per customer; (2) For a portion of our sample, the data did not allow us to accurately identify which container registry is in use. We assume that these are evenly distributed - nothing to our knowledge would indicate otherwise.

## The top 12 application components running in containers

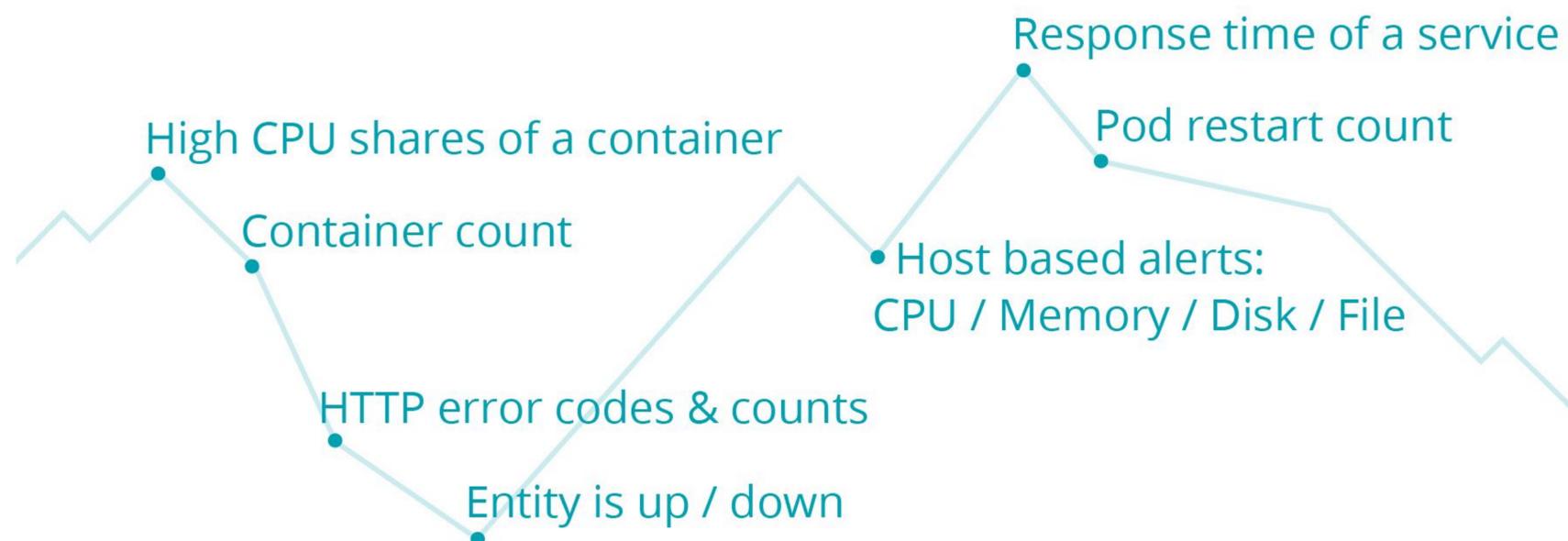
What are people actually running inside their containers? Sysdig's auto-discovery mechanism gives us a method to understand which open source components users are adopting inside their containers, without any explicit configuration or management by users. Below is a list of the twelve most common application checks we see running.



Of interest to us was that Nginx was the most popular. It is frequently serving at the endpoint of a service, or functioning as a traditional loadbalancer. Also somewhat surprising to us was that consul was running so closely to etcd - we expected more distance between the two.

Note this data does not focus on programming languages, though PHP and JVM are present. Most users are writing at least some custom application code in various languages and we see those as "custom apps".

## Popular alert conditions when using Docker containers



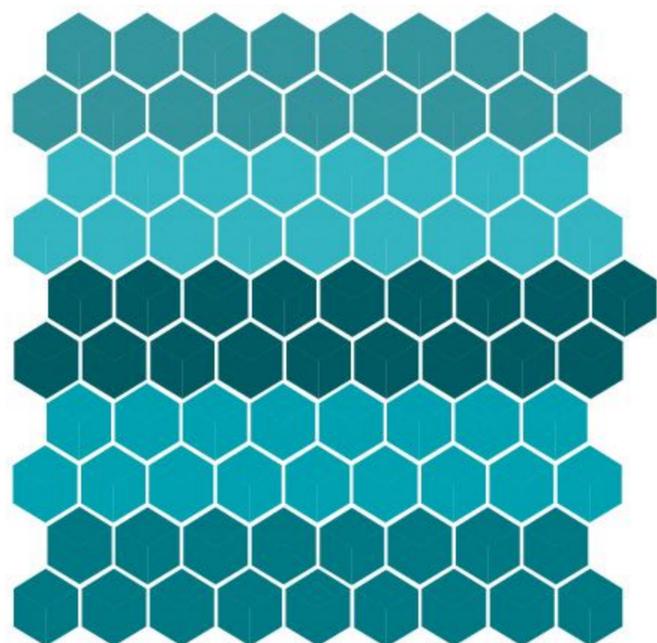
We wanted to better understand if docker adoption had changed the way people architect and operate their software. While alerts are not a perfect way to understand such a deep question, they do provide a glimpse into how people are thinking about their software in production.

From the data above, we see that a shift is in process, but older alerting conditions are still in use. For example, tried-and-true alerts such as High CPU/Memory/Disk are still very much in use. At the same time there are newer alert conditions which better relate to today's infrastructure. A good example here is "Pod Restart

Count." We also see a number of alerts that adapted for the modern age: "Entity is Down" now is used across containers in addition to hosts, and "High CPU shares" is a CPU alert tuned for containers.

Finally, we see a number of alerts that haven't changed nor should they. Alerts on the response time of a service and alerts on HTTP Error Codes continue to be popular because they represent the state of the application, not just the state of the container infrastructure. The biggest change with these alerts tends to be the scope across which they trigger, which we cover next.

## Popular alert scopes



Deployment Name

Lower level orchestrator constructs  
(eg, pod, ReplicaSet etc)

Role of Host

Cloud provider tags

Container name

We see a few things when examining Alert Scopes in relation to how users have adopted Docker containers. First is the concept of tagging. This allows The most common number of tags we see used to scope out an alert

Many of the alert scopes are actually operating across orchestrator constructs, such as a Kubernetes Deployment, Pod, or something similar. We also see, however, that many people are using the container name, which implies the user has a more personal understanding of what is running where inside their infrastructure.

Finally, we do see some important tagging related to physical infrastructure as well. Cloud provider tags typically represent some physical aspect of deployment, such as the host, region or availability zone. The “Role of the Host” is typically applied at the Sysdig agent. Given users only need to deploy a single Sysdig agent per host - regardless of the number of containers or pods - implies they are using the tag in relation to physical infrastructure.

## Conclusion

### Docker usage continues to advance

Our experience in the Docker ecosystem is that users continue to advance in both the scale and sophistication of their Docker usage. This report gives the reader a good sense of where users are today, and provides a complement to survey-based reports that provide more insight on intention or objectives.

*We expect to repeat this report in about a year, and imagine that we will see significant changes in that time.*