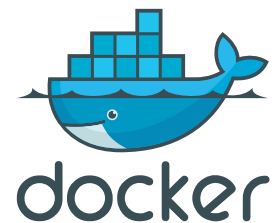


# Evolution of the Modern Software Supply Chain

The Docker Survey, 2016



# Overview

**Containers are the base unit of the modern software supply chain.** The results of the 'Evolution of the Modern Software Supply Chain' 2016 Survey demonstrate that the Docker platform sits at the nexus, and is the driving force, of three aspects of modern application development strategy: the shift to (hybrid) cloud, the transition to microservices architecture, and the transformation of app development through DevOps practices.

**Docker provides the necessary agility, control and portability that the software supply chain demands.**

As every business — whether the disruptive or the disrupted — has embraced a digital transformation, they have they been driven to deliver apps and experiences at ever increasing rates. The efficiency of the software supply chain is a key business performance characteristic. These characteristics are agility, control and portability for the teams and products they create, all of which can be derived by offering 'Containers as a Service' to app and IT ops teams with the Docker platform. It is no surprise that Docker has emerged as the enabler of this latest evolution — something Redmonk describes as '[The Docker Pattern](#)'.

**Docker is delivering quantifiable improvements to application delivery processes through improved DevOps practices.**

The transformation of application development is being driven by a desire by organizations to shake off the inertia of legacy software: long release cycles, ballooning monolithic code bases and an ever-growing rift between developers and IT operations teams. The search for speed paved the way for Agile development and DevOps practices to gain momentum not only in new web companies but is also within traditional enterprises.

**Docker is central to hybrid cloud strategies, enabling freedom of choice of on-premises, private and public environments.**

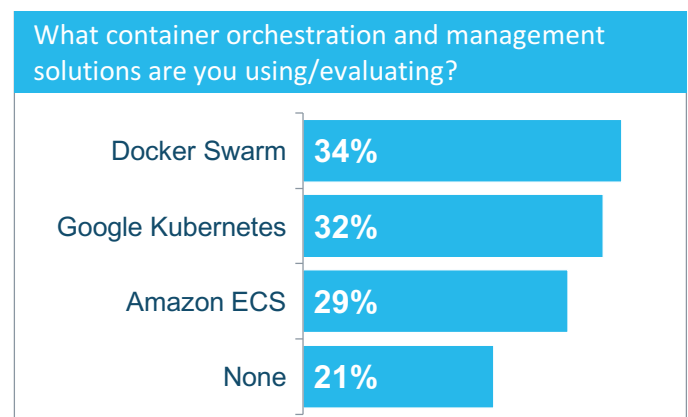
Apps using containers are fully portable across any infrastructure offering a higher level of control as business pursue a hybrid or full cloud strategy across multiple vendors while avoiding lock-in at a level similar to that of the old tech behemoths of the 2000s.

**Docker enables microservice architecture delivery and the modernization of legacy monolithic apps.**

Another effect of the explosion of app development options, and enormous growth of web services offering developer-focused APIs is the shift toward microservice architectures, and the modernization of legacy monolithic apps. Apps are now assembled with potentially dozens of individual services offering flexibility and agility in app design, along with complexity in app management. Docker provides the control plane necessary for management of microservice apps: from composition and networking through to orchestration and scheduling.

**Docker offers portability as unique value amongst container management vendors.**

With respect to the orchestration and management ecosystem, respondents overwhelmingly reported using Docker Swarm, Google Kubernetes and Amazon EC2 Container Service. Out of the top three, only Docker Swarm is the container orchestrator that is cloud agnostic.



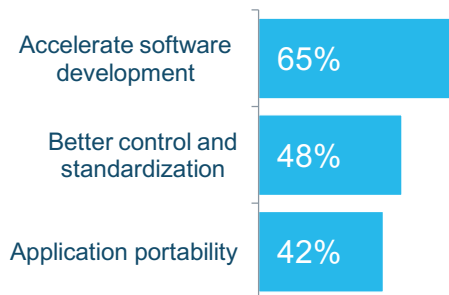
# Key Findings

Changes in DevOps practices, the shift to the cloud and the continuing need to modernize legacy apps demand increased agility, control and portability for development and operations teams who are increasingly turning to Docker to provide these capabilities for their app environments.

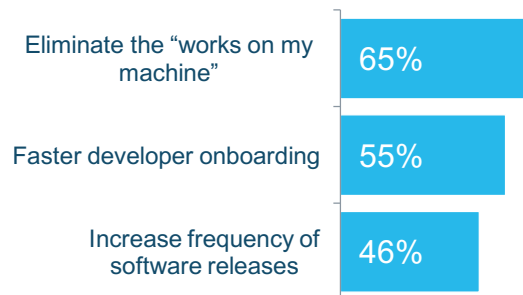
**1** Docker provides the software supply chain with agility, control and portability for app development.

**2** Docker is delivering quantifiable improvements to application delivery through changing DevOps practices.

## Desired Outcome of Docker Initiative



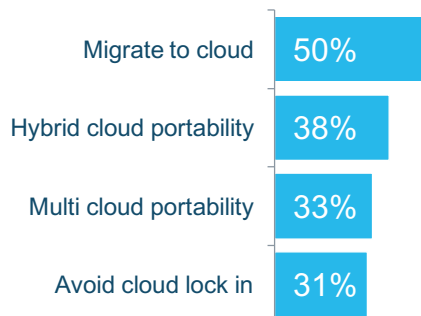
## Docker Impact for Developers



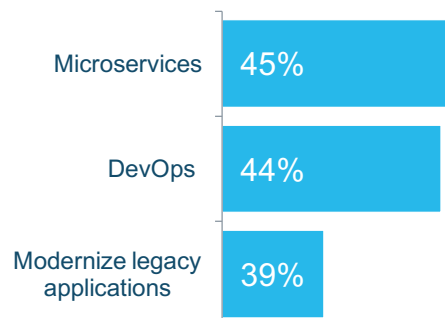
**3** Docker is central to hybrid and multi cloud strategies, enabling freedom of choice across on-premises, private and public environments.

**4** Docker enables microservice architecture delivery and the modernization of legacy monolithic apps.

## How is Docker part of your Cloud strategy?



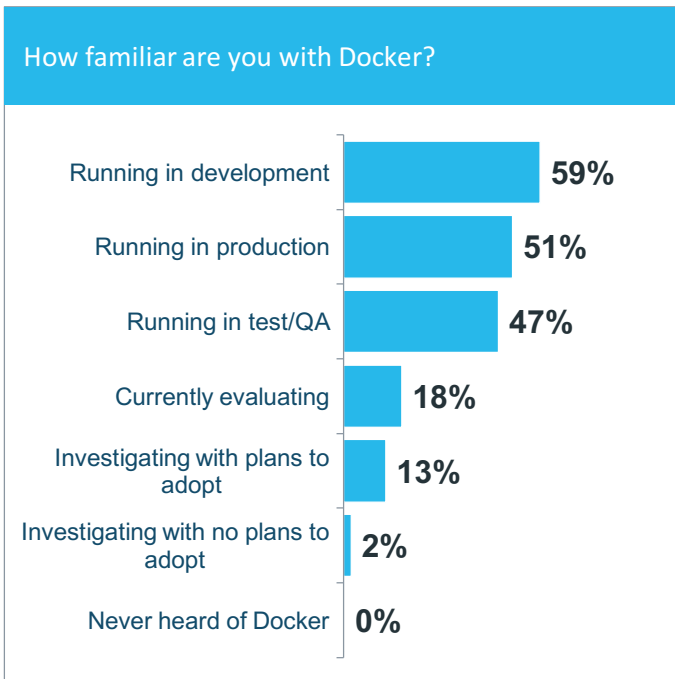
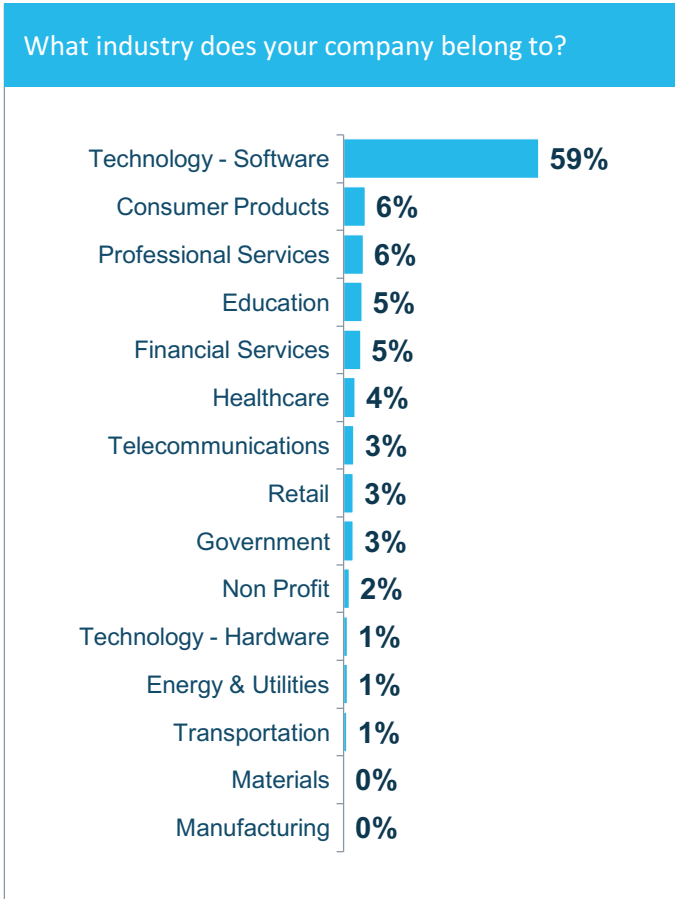
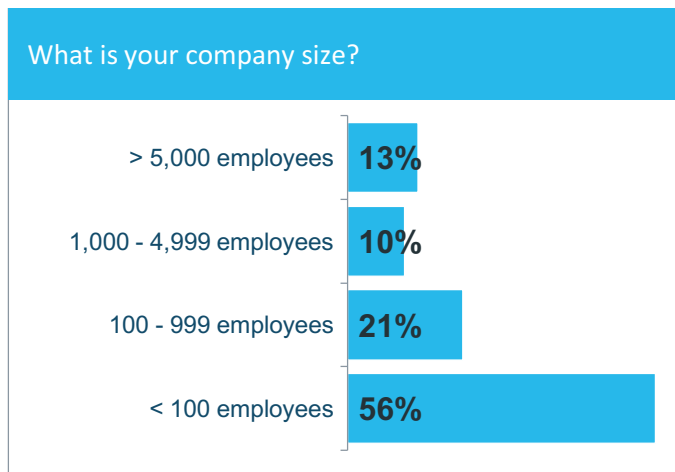
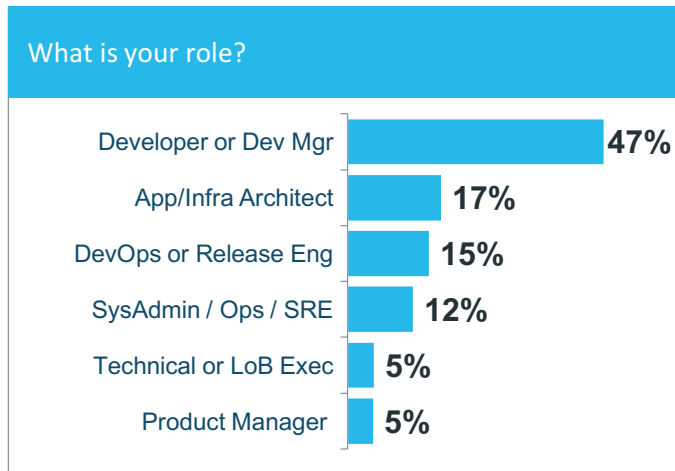
## What are your top initiatives in 2016?



# About Our Respondents

The Docker survey resulted in 511 respondents - all technology professionals covering an array of development and operations focused roles:

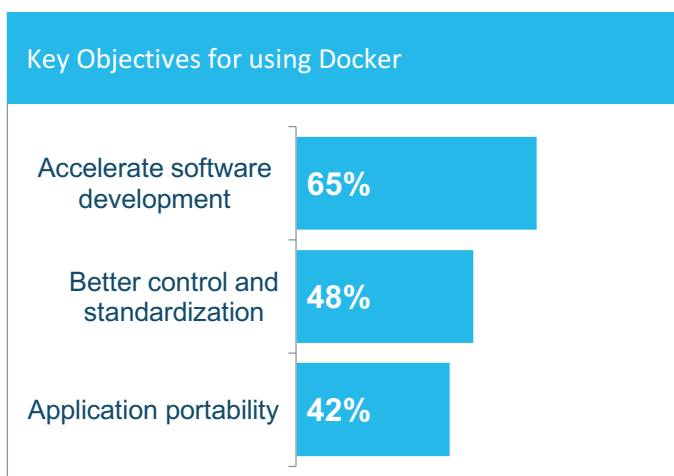
- 47% are Developers, or Development Managers.
- 56% are from companies with less than 100 employees.
- 59% are from software companies, with the remainder are spread across a broad array of industries.
- 51% are currently running Docker in production.



## Key Finding 1: The Docker pattern provides agility, portability and control.

The first step to building a more efficient software supply chain requires teams to lower development hurdles. By reducing dependencies and increasing efficiencies, organizations can attain a more productive software development process. The promise of agility (speed and flexibility) has drawn developers to Docker and has enabled them to own what goes on inside the container and understand how to accelerate the software development process. In fact, more than 65% were using Docker because of the potential productivity gain and streamlined development cycle.

Developers are also drawn to the portability benefits of Docker and the ability to run Dockerized applications across any infrastructure, from on-premises datacenters to public clouds, across a vast array of network and storage providers. In fact, more than 42% of all respondents surveyed indicated that application portability was a key factor in their decision to use Docker.



Portability and agility enables developers to make the Continuous Integration (CI) process more efficient but CI only realizes half of the potential

value of Docker because the app pipeline just becomes a “waterfall” deployment. To provide this much needed management and control, some organizations have tried to use Platform-as-a-Service (PaaS) solutions and other bundled solutions but end up sacrificing portability and agility for the user. Others chose not to adopt these solutions and instead built it themselves with glue code and bash scripts because they didn’t want to forego the agility and portability that drew them to Docker in the first place.

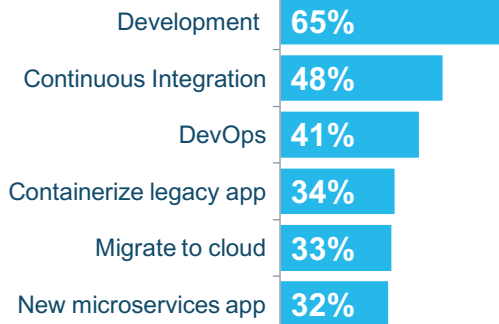
The Docker Container as a Service (CaaS) platform addresses the requirements for organizations that want to manage the application lifecycle of Dockerized applications from development through production, essentially achieving better control and standardization. In fact, more than 48% of respondents indicated their intention to use Docker for better management of the software supply chain.

The Docker CaaS platform provides an IT managed and secured application environment where developers can, in a self-service manner, build and deploy applications. Some of the world’s most security-conscious organizations, have bought and deployed Docker Datacenter to deliver a CaaS that provides agility for development teams, control for operations teams and portability of applications across any infrastructure, from on-premises datacenters to public clouds, across a vast array of network and storage providers.

### Organizations are using Docker to move applications into production

As RedMonk recently reported in its blog on the [Rise of the Docker Pattern](#): “ the path is now set clear for Docker to become an industry standard production platform.” In this article, Red Monk comments on the growing popularity of using Docker for development but also indicates that very quickly, Docker will become the standard with both developer and IT ops. In this survey, more than 90% of respondents are initially using Docker to containerize existing apps or create new microservices.

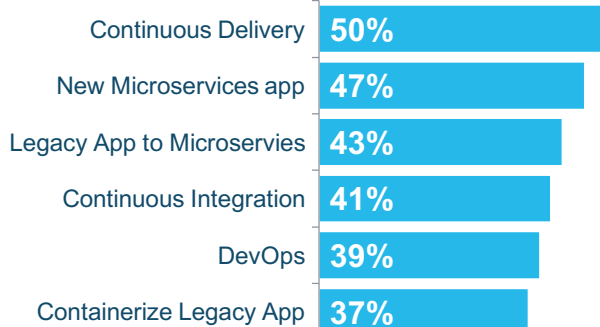
### Docker Use Cases Already Deployed



focus on securing, deploying and managing apps at scale. Docker provides organizations with a solid foundation for their container environment, enabling teams to spend more time developing and shipping the applications that bring value to their customers.

But then as Dockerized applications begin to take root throughout the organization, IT operations needs to find a way to manage these applications. This could be why more than 58% indicated that they have some application in production with respondents indicating that more than 80% would have Dockerized applications in production by the end of 2016. A comparison of existing deployments to planned 2016 initiatives indicate a greater emphasis on continuously delivering microservices applications to production systems.

### Docker Use Cases Planned for 2016

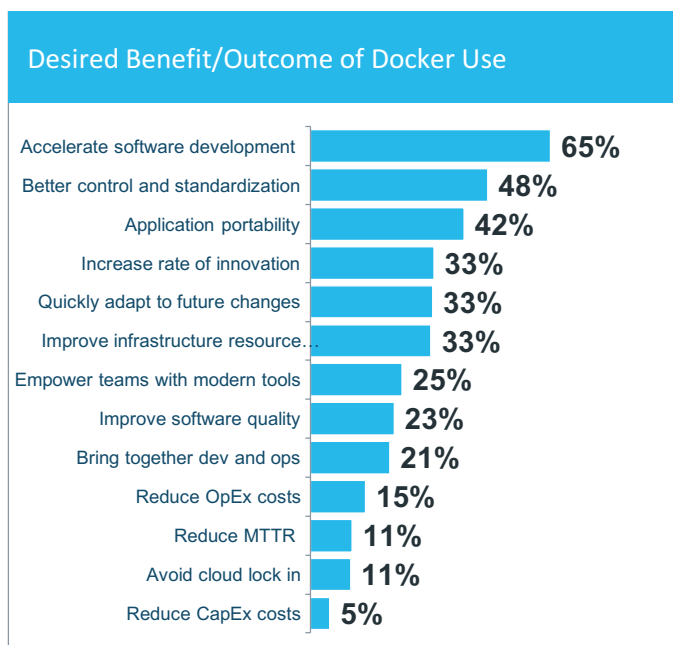


The developer begins with Docker because it is easy to use and makes them faster. Then by the very nature of the Docker technology, they are able to easily share their software and dependencies with others without conflicts between applications or environments. Meanwhile, IT ops teams can

## Key Finding 2: Docker is delivering quantifiable improvements to application delivery by accelerating DevOps practices.

Organizations of all sizes are driven by a desire for speed and to shake off the inertia of legacy software; freedom from long release cycles, ballooning monolithic code bases and an ever expanding rift between developers and IT teams. This rift is exacerbated by having to find ways to absorb the exponential growth in languages, frameworks and APIs for development (itself driven by the open source movement). This added diversity stresses systems already unable to reconcile the differences in dependencies and packages in development, QA and production.

Roughly half of respondents have already traveled the path of CI and DevOps and are looking to advance those practices to continuous delivery into production. The remaining respondents are looking to catch up with DevOps/CI in 2016.

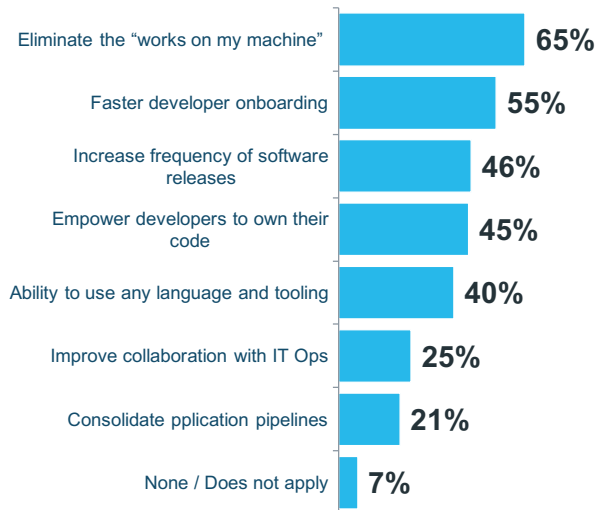


The search for speed in the software supply chain is fueled by agile methodologies and DevOps practices to drive more frequent changes, improve software quality and reduce mean time to resolution. Docker is uniquely able to enable this transformation. The Docker container is the essential unit of control for both developer and IT ops, allowing for a separation of concerns, standardization and a streamlining of the development to production pipeline. Docker provides the ability for development teams to leverage this diverse set of languages and tools in a manner that standardizes the application environment. Making changes simpler and easier. The Ops side of this equation requires control to prevent the development agility from becoming chaos and defeating the value it promises with tooling to manage and secure the software supply chain without restricting the developer workflow or tools/languages.

**Shipping more software faster is not strictly a developer agenda but also impacts IT operations and the business.**

Developers feel more empowered to freely create software and effortlessly ship the same code from their machine to test and production. Through that, survey respondents reported on average a 13X increase in frequency of software releases.

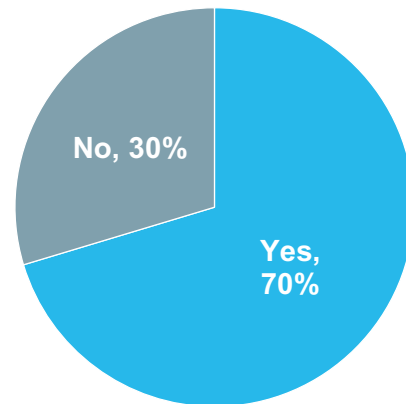
## Docker Impact for Developers



The biggest issue solved by Docker that accelerates the "development to production" pipeline of DevOps is eliminating the "works on my machine" problem. A problem born from the dependency matrix and realized when you ship software to a new server. The new server may not have the same packages installed that the developer's machine did, thus causing the application to not work properly. Docker isolates the software and the required dependencies and libraries together in the container - now the exact thing you built will be tested in QA and then run in production.

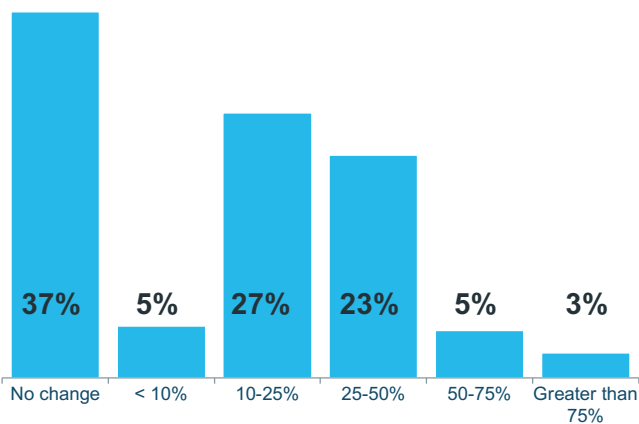
Because Docker makes it simple and easy to push software out, isolate issues and roll back, over 63% of organizations report a reduction in their MTTR which impacts overall software quality and customer satisfaction.

## Has Docker transformed how dev and IT ops teams collaborate to build and operate applications



70% of survey respondents agree that Docker helps developers and IT ops teams collaborate better to create and operate applications. The Docker architecture allows for a true separation of concerns between the application and infrastructure so each respective team can adjust what they need without breaking the other's.

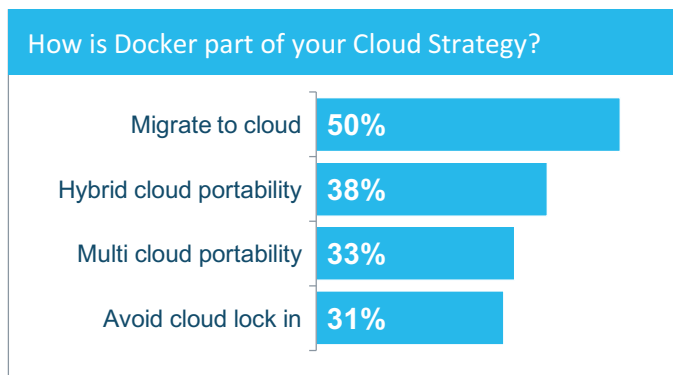
## Average MTTR Reduction with Docker



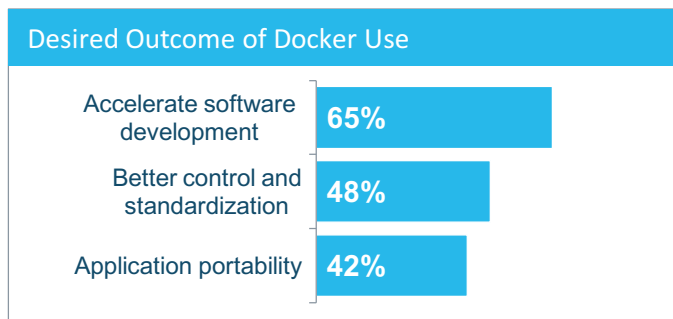


## Key Finding 3: Docker is central to hybrid cloud strategies as it enables freedom of choice of on-premises, private and public environments.

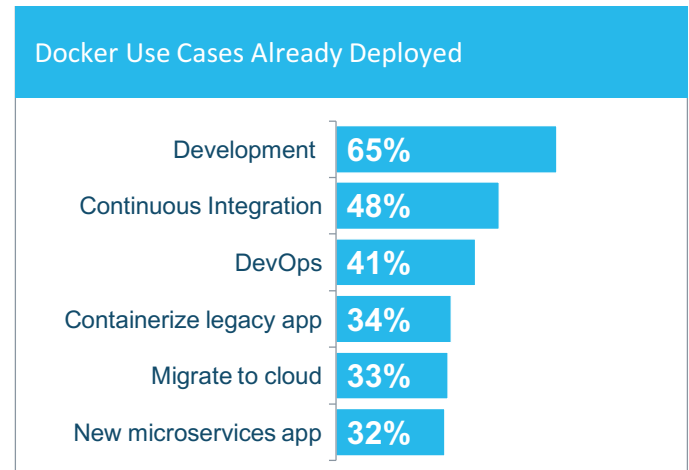
80% of organizations using Docker have described it as central to their cloud strategy. Organizations view Docker as the infrastructure agnostic platform to connect across any combination of vendors supplying their public and private compute infrastructure.



At the heart of that interest is leveraging Dockerized distributed applications which are fully portable across any infrastructure offering a higher level of control as business pursue a hybrid- or full-cloud strategy across multiple vendors while avoiding lock-in at a level similar to that of the old tech behemoths of the 2000s.



This portability is defined at not just the compute layer but also the ability to move across networks and storage to maintain a fully-functioning distributed application across environments without changing the app code. Portability thus increases the speed at which the app is able to move through the pipeline to deployment.

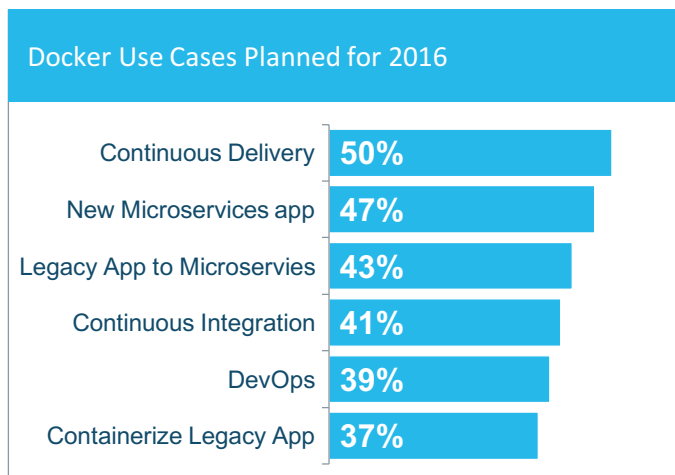


An essential part an enterprise cloud strategy is to look at existing app workloads. As much as Docker is central to microservices architecture initiatives, it is also being leveraged to take existing legacy applications and make them more portable across any infrastructure. In that process, legacy applications not just shadow IT applications can benefit from the public cloud. Additionally, as workloads move to the cloud enterprises want to be assured that applications remain portable and that they have full leverage to determine in which cloud their applications reside at any moment in time. Having that flexibility ensures that the operational tooling and the experience that is developed is optimized first around the applications and is specifically not tethered to a specific cloud provider.

## Key Finding 4: Docker enables microservices architecture delivery and the modernization of legacy monolithic apps.

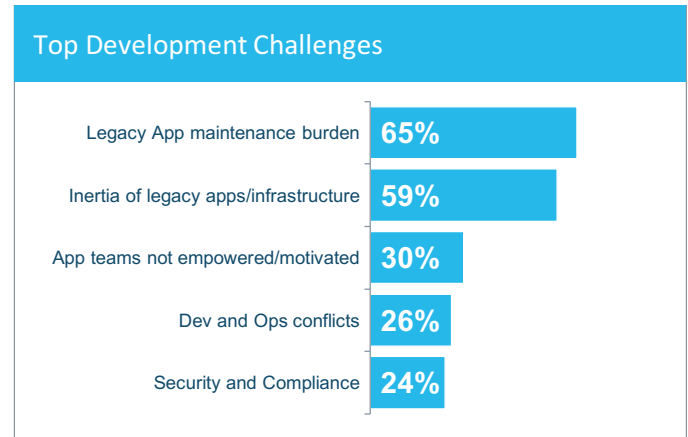
Another effect of the explosion of app development options, and enormous growth of web services offering developer-focused APIs is the shift toward microservice architectures, and the modernization of legacy monolithic apps. Apps are now assembled with potentially dozens of individual services offering flexibility and agility in app design, along with complexity in app management.

Delivering on microservice architectures is seen as a key benefit of Docker and is being actively planned by 47% of respondents as an initiative for 2016.

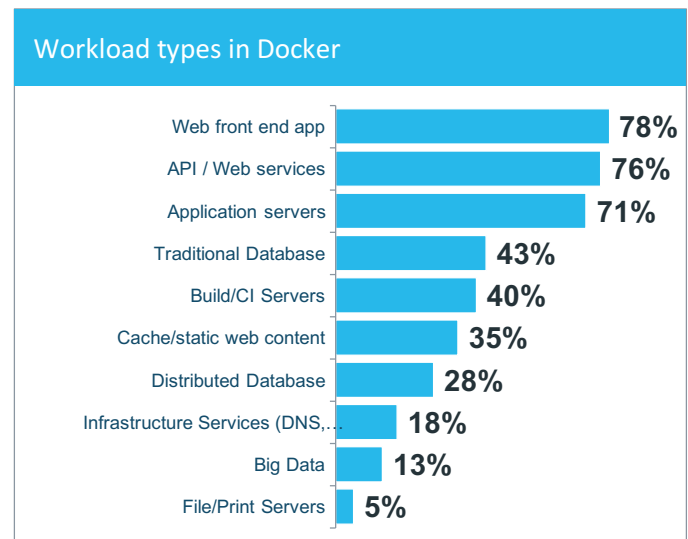


While Docker continues to enable the growth of microservice architectures, it is also enabling the shift of traditional workloads into container environments to assist with modernization and portability of any apps, with 43% of respondents planning to do that.

This addresses the burden of maintenance of legacy apps and the inertia of those environments, the top two challenges faced by development teams. App teams are doubly burdened by long and infrequent release cycles as a result of their legacy environments in a world where apps are fundamentally driving every customer engagement.



Perhaps unsurprisingly, Docker is being used to run Web Apps by 78% of users, with APIs and associated Web Services following up with 75%. However, 70% of users are running Application Servers, and databases - both traditional and distributed - are being run by 42% and 27% of users respectively. Big Data workloads makes an appearance with 13% of users running Big Data processing (such as Hadoop) with Docker.

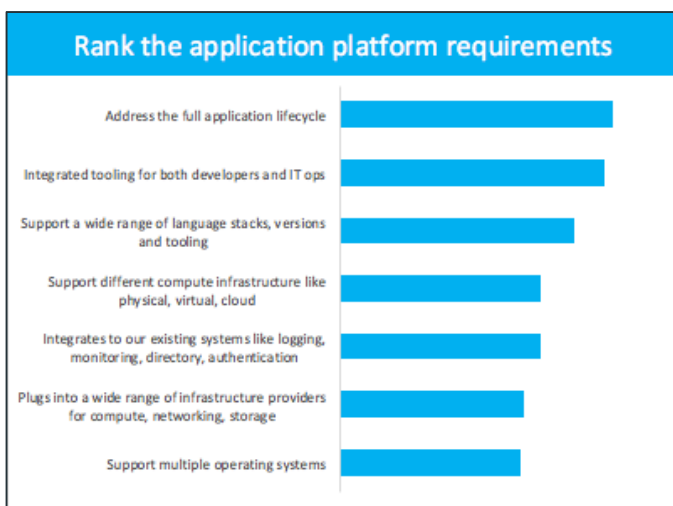


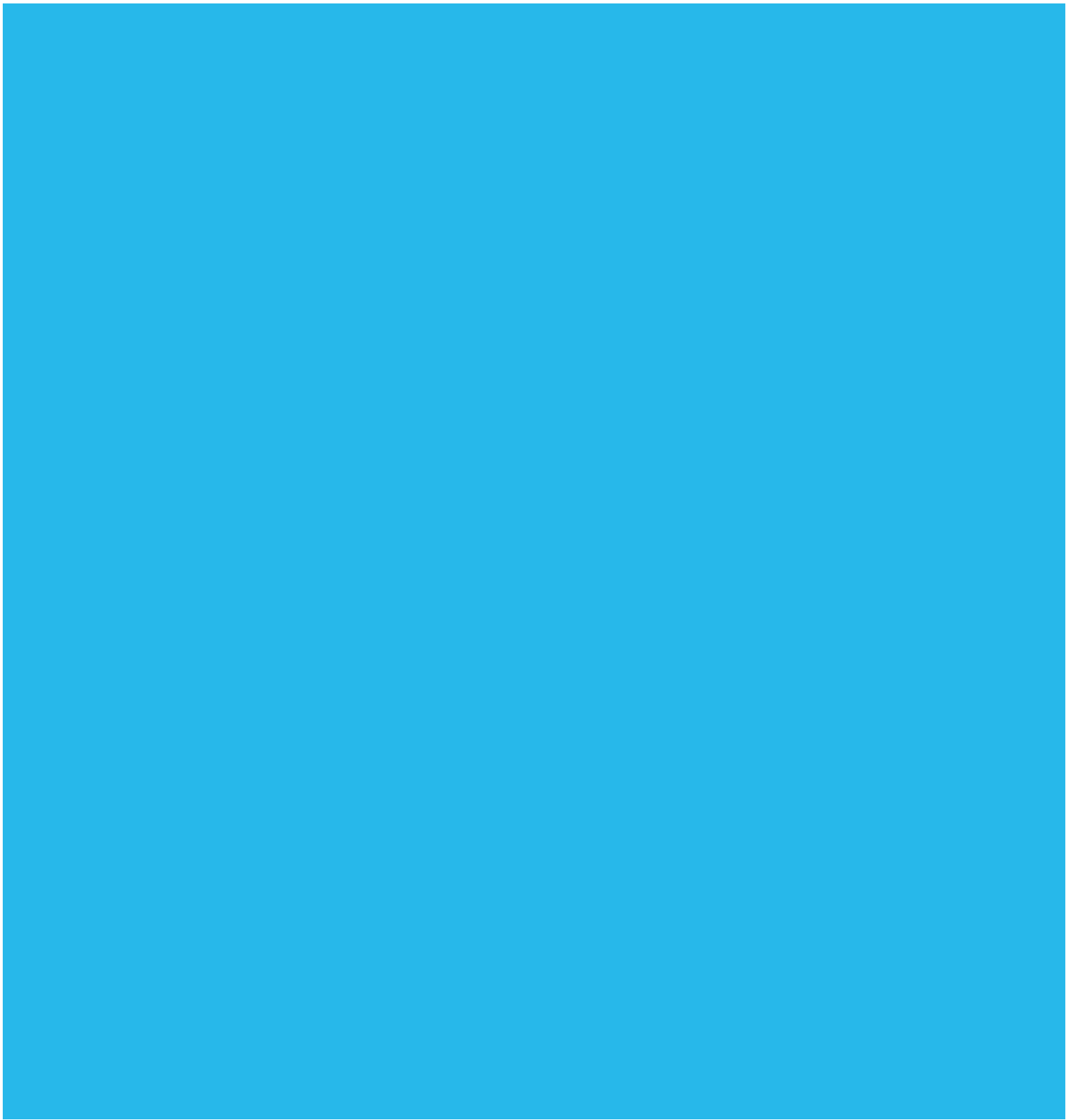
# Docker enables the transition to a modern app development pipeline

As organizations begin to move toward a more modern application delivery lifecycle, it is clear that companies are seeking the most efficient solution that also takes into account and adjusts for the trappings of previous platform generations. In search of agility, portability and control for both developers and IT ops teams, organizations are looking to make cloud computing, DevOps practices and microservices architectures a reality.

To achieve this new modern app platform, organizations are looking at capabilities and characteristics that address a world that is increasingly heterogeneous with respect to application stacks and infrastructure. This new platform must support the need for flexibility and freedom along with the DevOps practices by providing integrated tooling for both development and IT ops teams across the full app lifecycle. Teams are no longer interested in silo-ed tools that separate and isolate workflows, teams and applications from each other.

This survey indicates that the unique properties of Docker to provide a Containers as a Service platform to accelerate these initiatives in a manner that provides both standardization and customization. Applications and tooling are standardized on the container format while an open and pluggable platform allows organizations to integrate to existing systems or custom workflows designed for their business.





[www.docker.com](http://www.docker.com)

