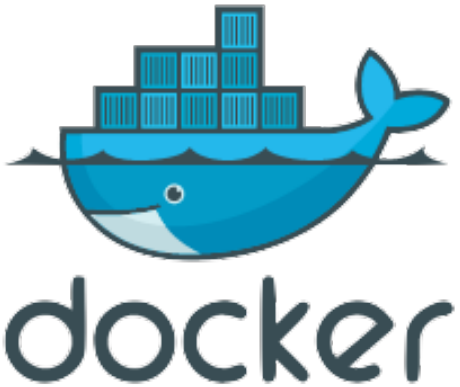
# Docker from Scratch

@pablovilla83



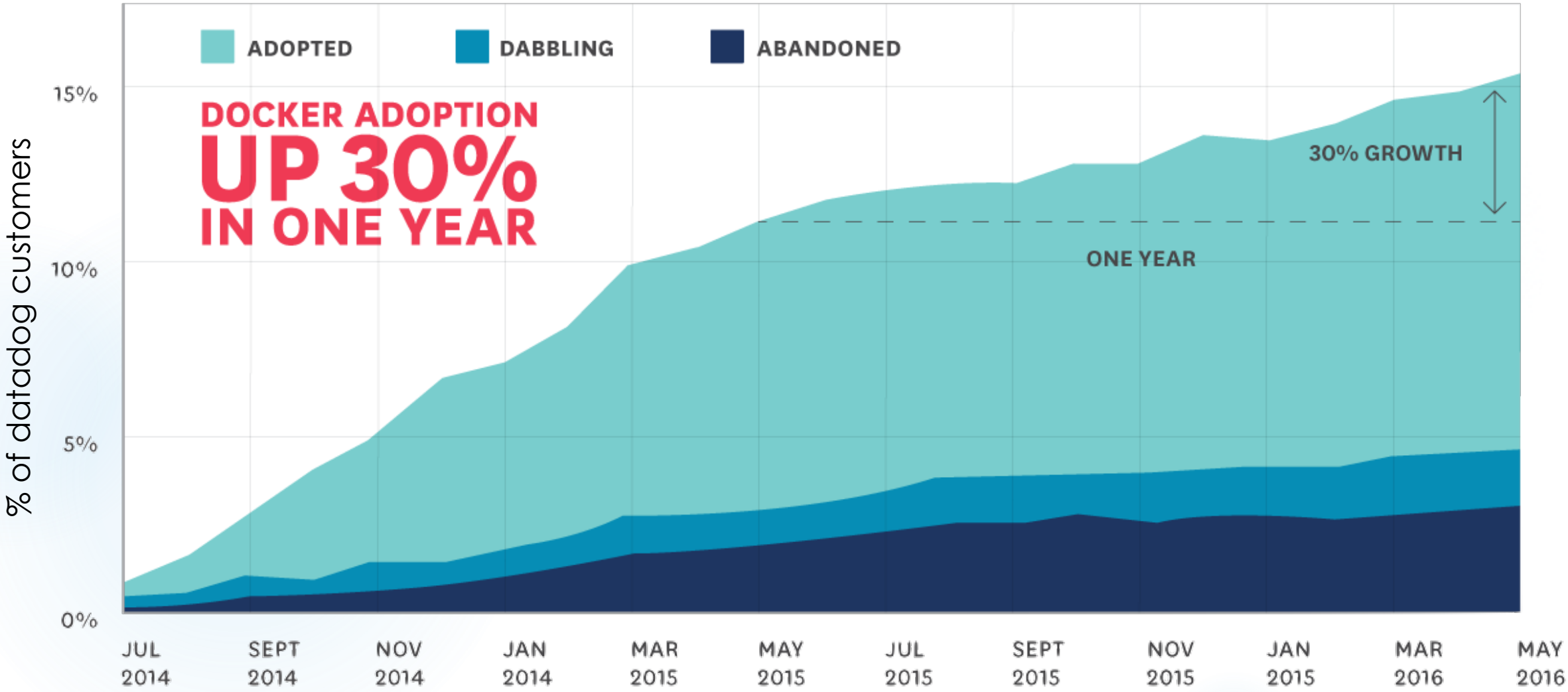
# What is Docker?

- Open source project – 38+k stars, 11+k forks, 180 pull requests, 1,945 issues
- Tool designed to make it easier to create, deploy, and run applications
- Envelope for software delivery
- Pushing code to the server shouldn't be too hard. Docker helps to fix the problem of "will it run in Production?"
- The application stack has increased in complexity (Apache server vs Node JS project with tons of dependencies). Making sure code behaves the same way across all devices is nearly impossible.

# What is Docker? Contd.

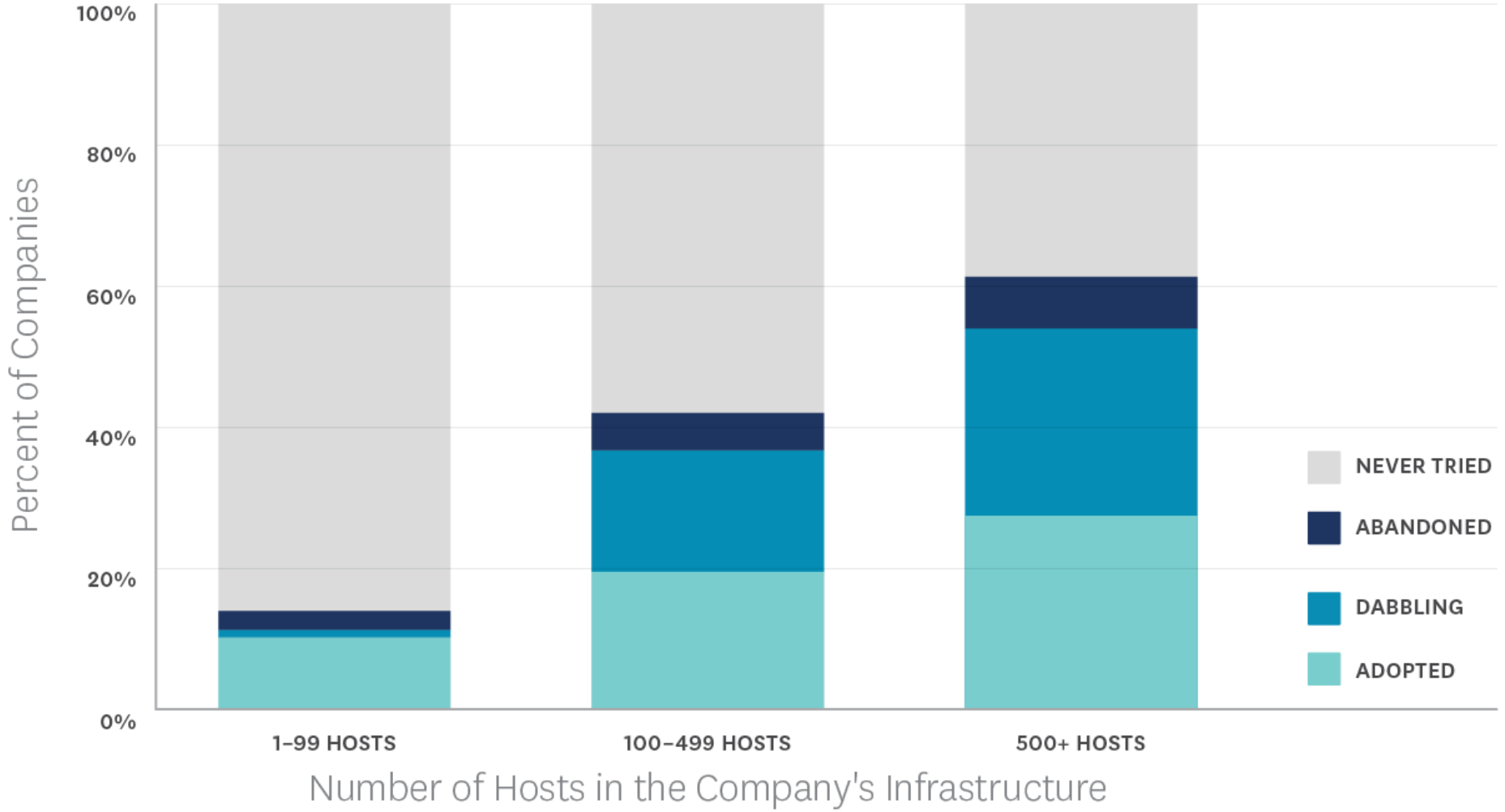
- The transportation industry has solved this problem
- Achieved separation of concerns. I focus on my load and I can handle it to a wide variety of infrastructure providers
- Developer worries about the inside of the box. DevOps worries about the outside

# Docker Adoption Behavior

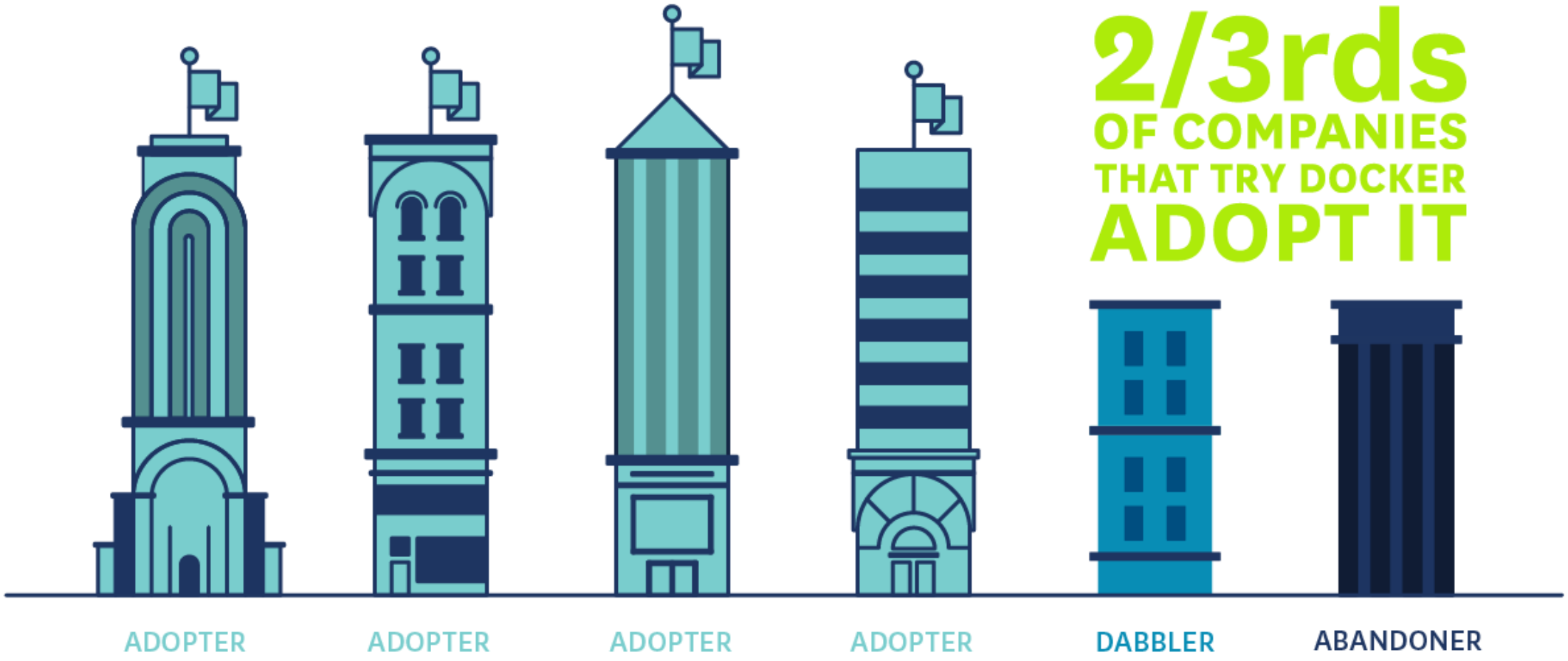



*Source: Datadog*

## Docker Adoption Status by Infrastructure Size




*Source: Datadog*



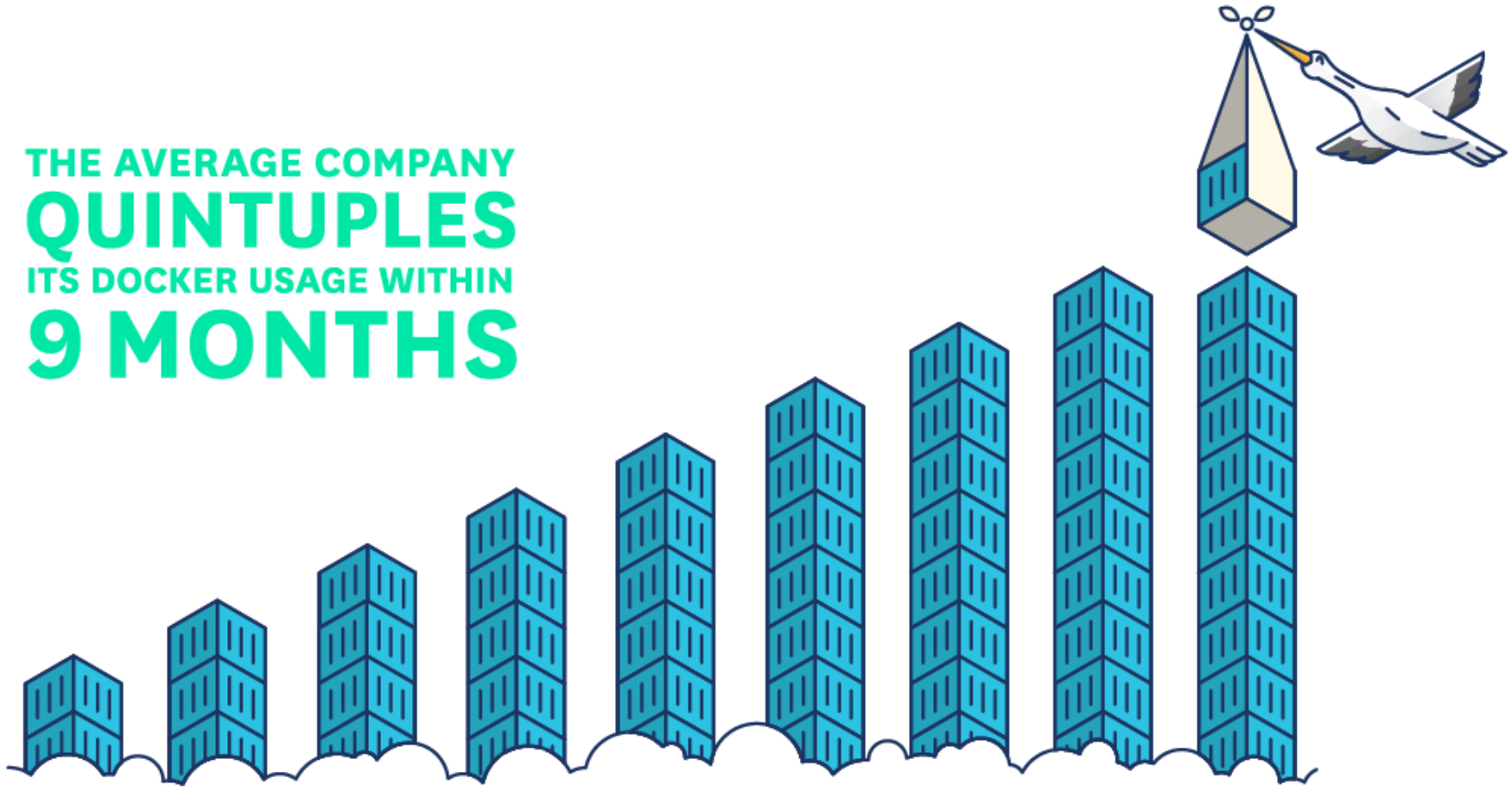

*Source: Datadog*

THE AVERAGE COMPANY

# QUINTUPLES

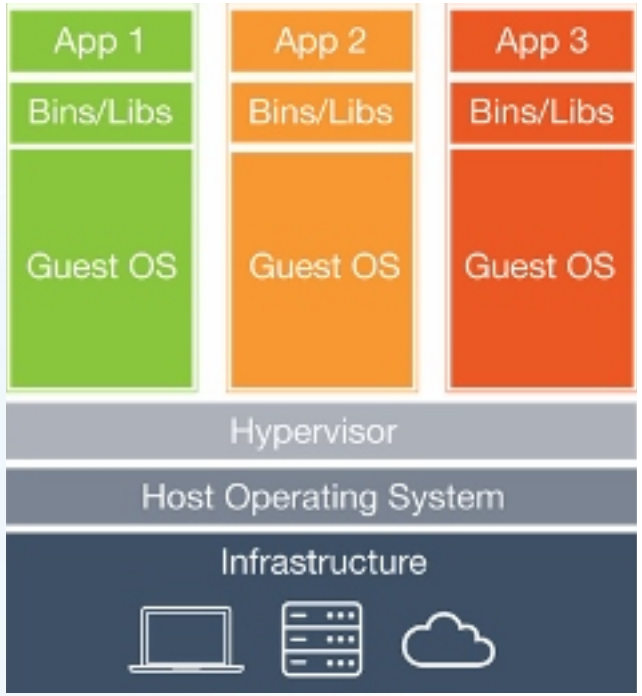
ITS DOCKER USAGE WITHIN

# 9 MONTHS

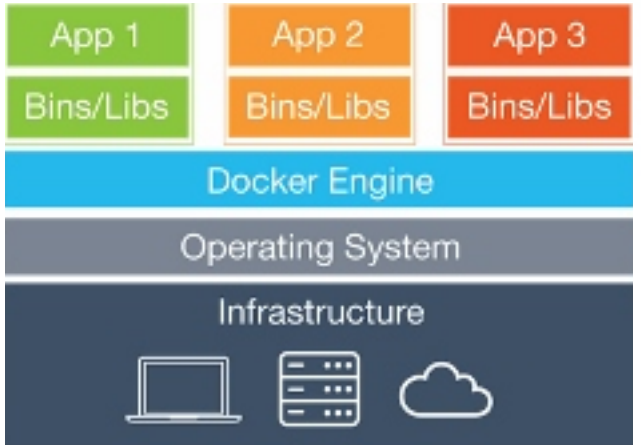


*Source: Datadog*

# Differences between VM's and Containers




Virtual Machines




Containers

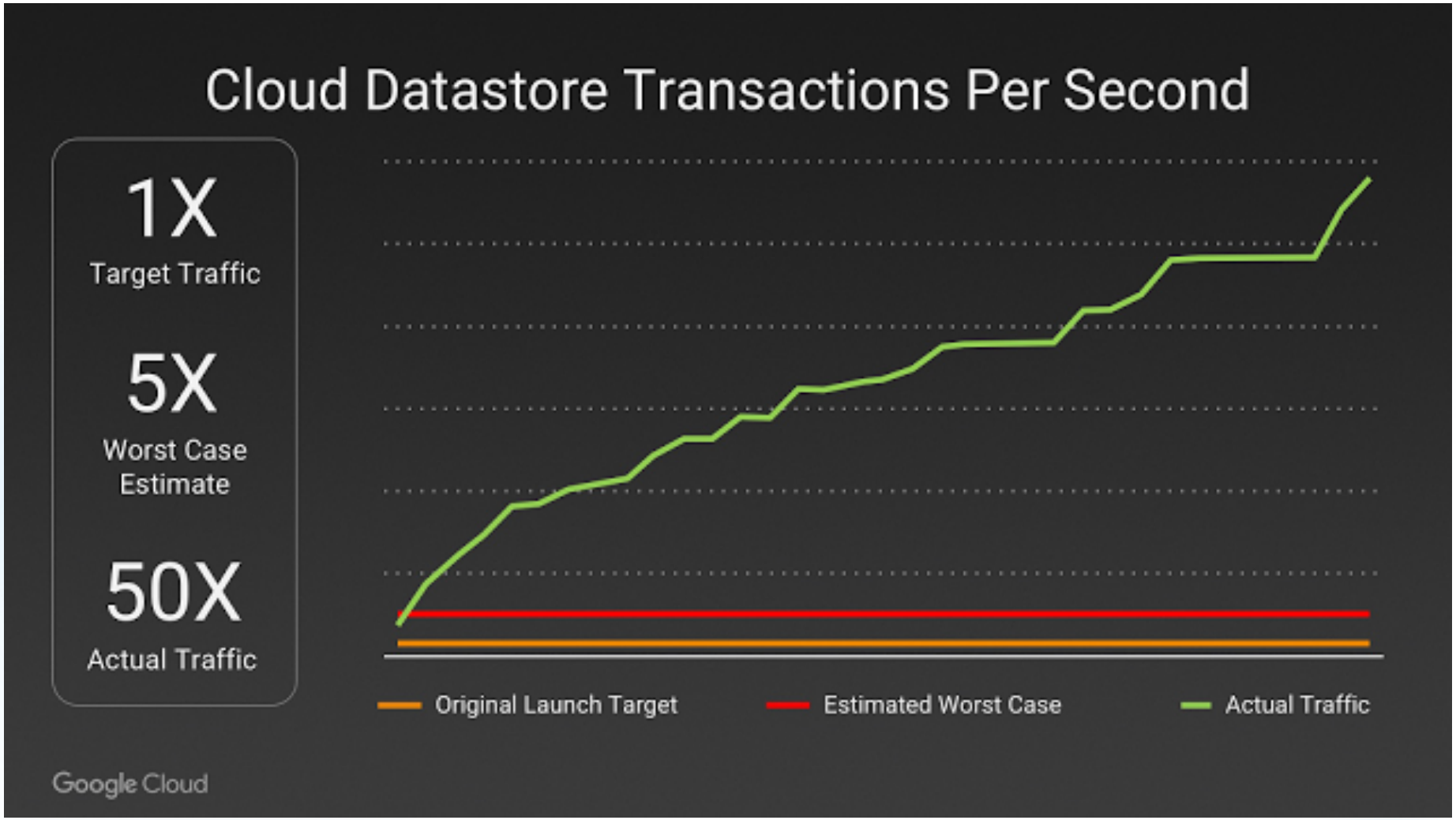
# Advantages of Containers

- Containers include the minimal application requirements to run, making them lightweight. I avoid dealing with stuff I don't need.
- Portability across machines of an entire environment. As long as the Host OS runs Docker, I'm all set!
- Version control for an application's runtime - DockerHub
- Increased control of server's resources. I don't assign static resources to a VM, rather give the container exactly what it needs.
- Boot speed.

# Advantages of Containers Cont'd

- A container stops when the main process finishes. Shorter lifetime.
- This means that Applications can be deconstructed into much smaller components (i.e. micro-service architecture) – Docker changes the way in which we build software
- Makes the management in production easier
- Much better horizontal scaling and less overhead. Deploying a new container takes a few seconds at most.

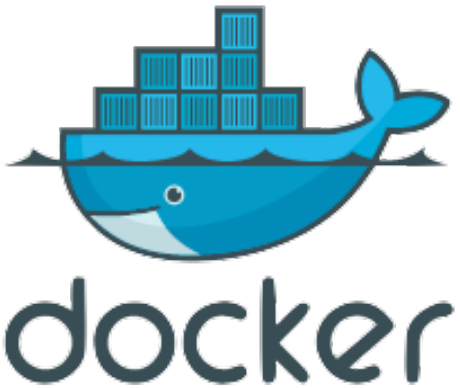
# Example in the Wild: Pokémon GO



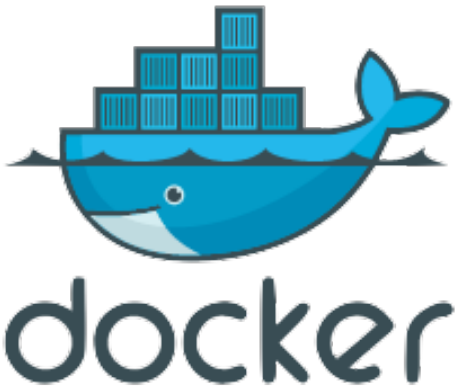
- The application logic for the game runs on a Container Engine (GKE) powered by the open source Kubernetes project.
- Containers can be scaled at planetary scale
- The developers were free to develop live changes for their players
- Pokémon GO was the largest Kubernetes deployment on GCP ever.
- To support Pokémon GO's massive player base, many tens of thousands of cores were provisioned for Niantic's Container Engine cluster.

# Hands on

https://github.com/pablovilla83/do-cdmx-nginx

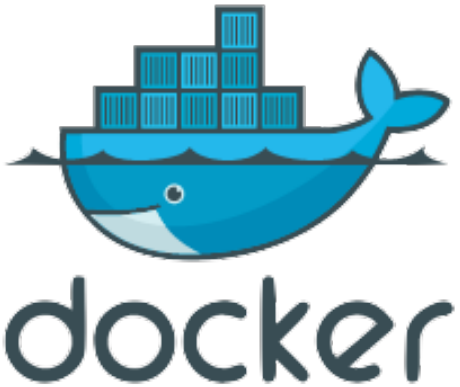


# Questions?



# Thank you!

@pablovilla83



# Sources and Extra Material

- Docker for the virtualization admin
- https://goto.docker.com/docker-for-the-virtualization-admin.html

- 8 surprising facts about Docker adoption
- https://www.datadoghq.com/docker-adoption/

- Bringing Pokémon Go to life in GCP
- https://cloudplatform.googleblog.com/2016/09/bringing-Pokemon-GO-to-life-on-Google-Cloud.html

- Docker commands
- https://docs.docker.com/engine/reference/commandline/