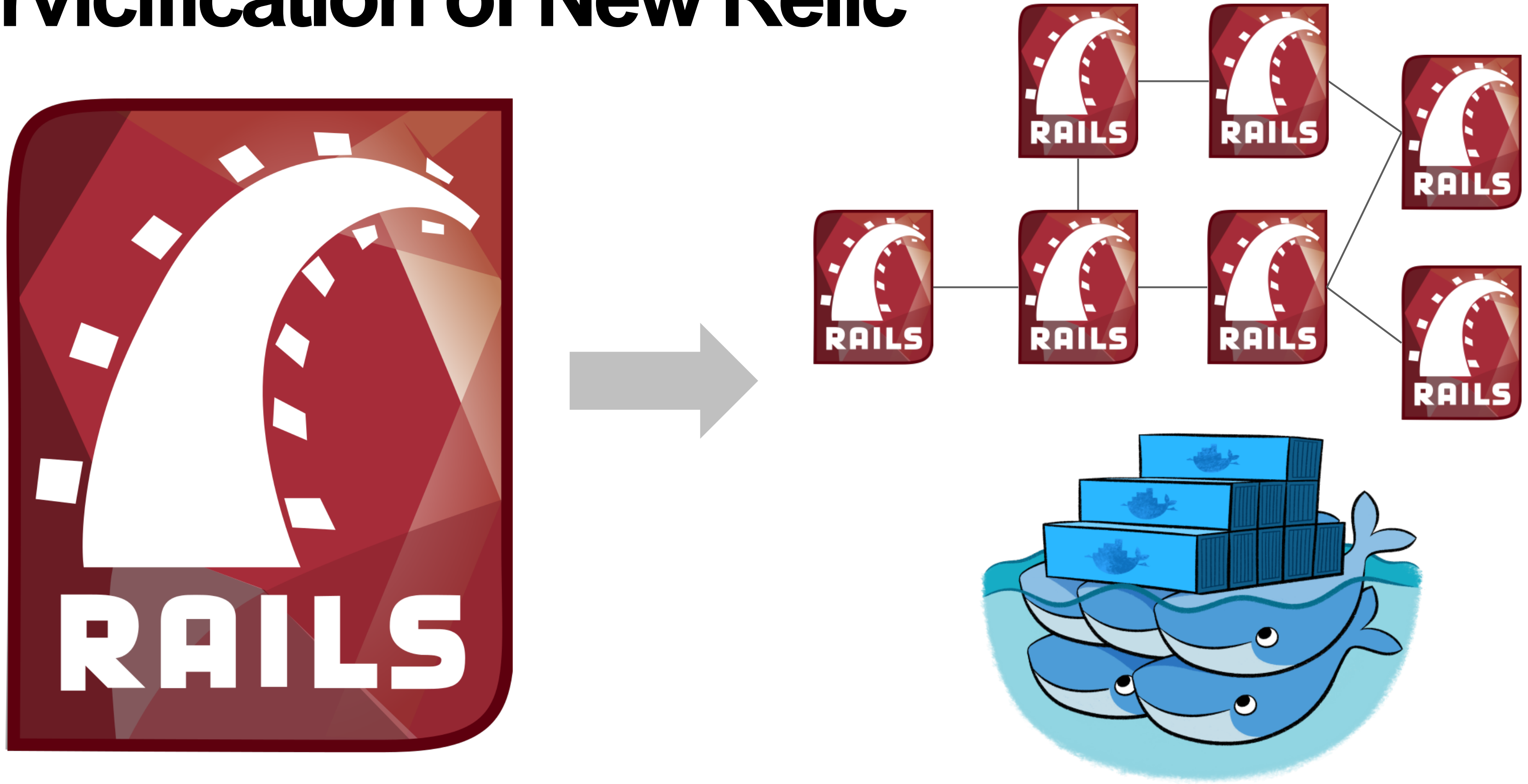
The Truth about Docker Container Lifecycles

KEVIN MCGUIRE

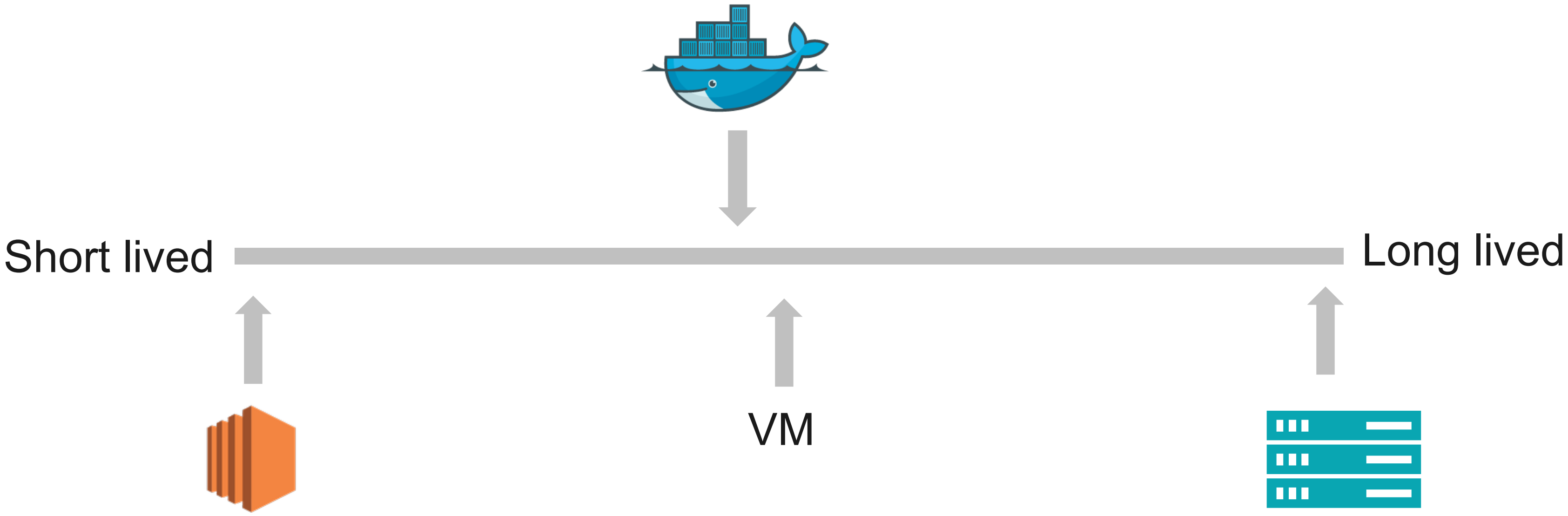
AUGUST 17/2015

A story begins…

Servicification of New Relic



Service container life cycle



Short lived

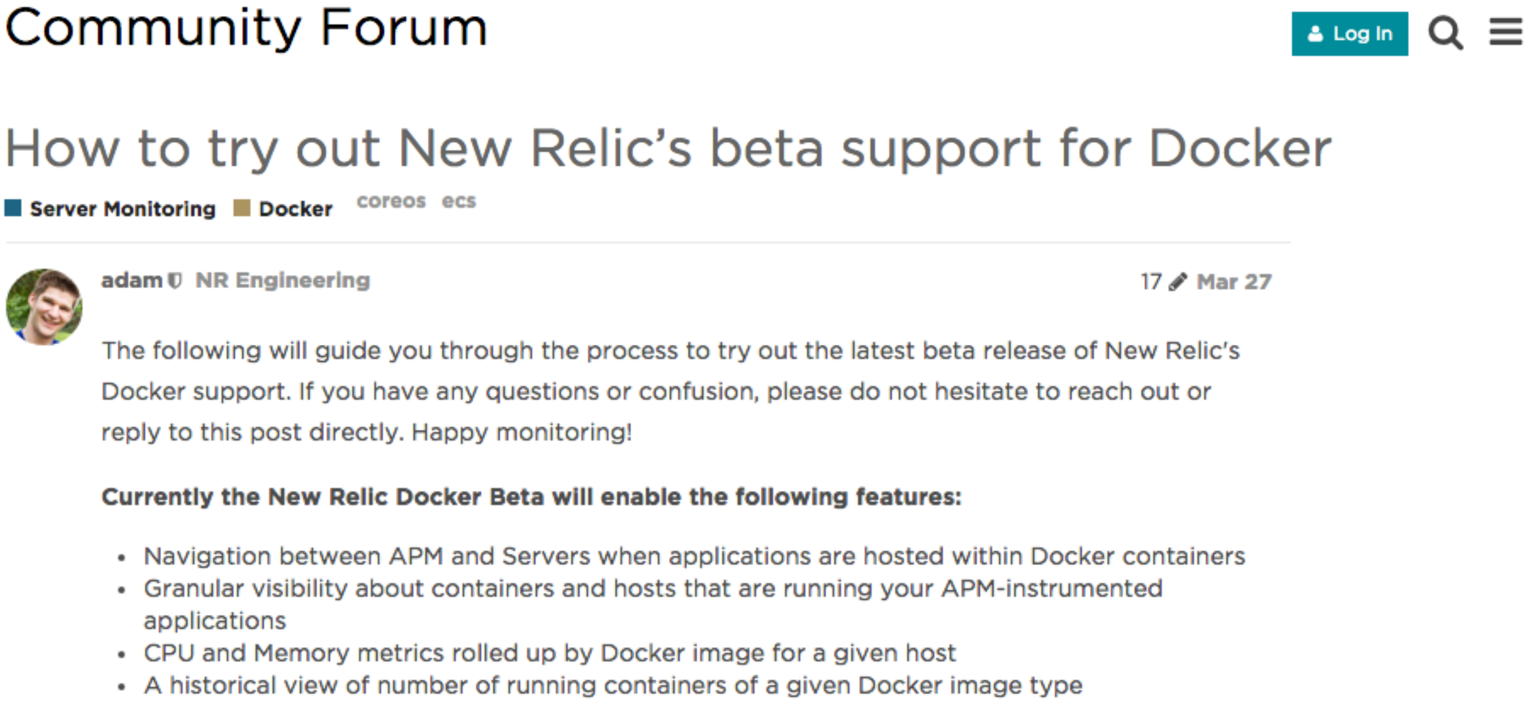
Long lived

VM

Theory

Docker is a lightweight VM

Let’s monitor it!



Community Forum

How to try out New Relic’s beta support for Docker

■ Server Monitoring ■ Docker coreos ecs

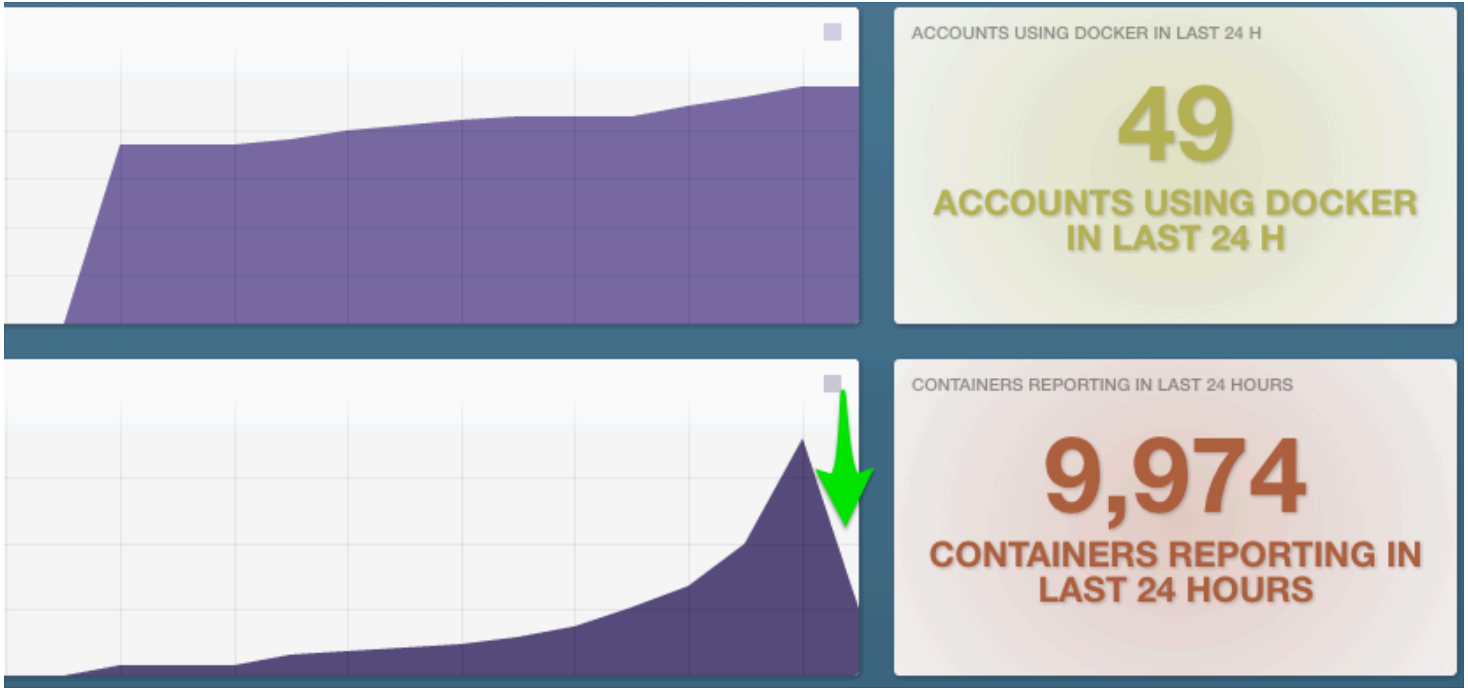
adam NR Engineering 17 Mar 27

The following will guide you through the process to try out the latest beta release of New Relic's Docker support. If you have any questions or confusion, please do not hesitate to reach out or reply to this post directly. Happy monitoring!

Currently the New Relic Docker Beta will enable the following features:

- Navigation between APM and Servers when applications are hosted within Docker containers
- Granular visibility about containers and hosts that are running your APM-instrumented applications
- CPU and Memory metrics rolled up by Docker image for a given host
- A historical view of number of running containers of a given Docker image type

Well that was surprising



ACCOUNTS USING DOCKER IN LAST 24 H

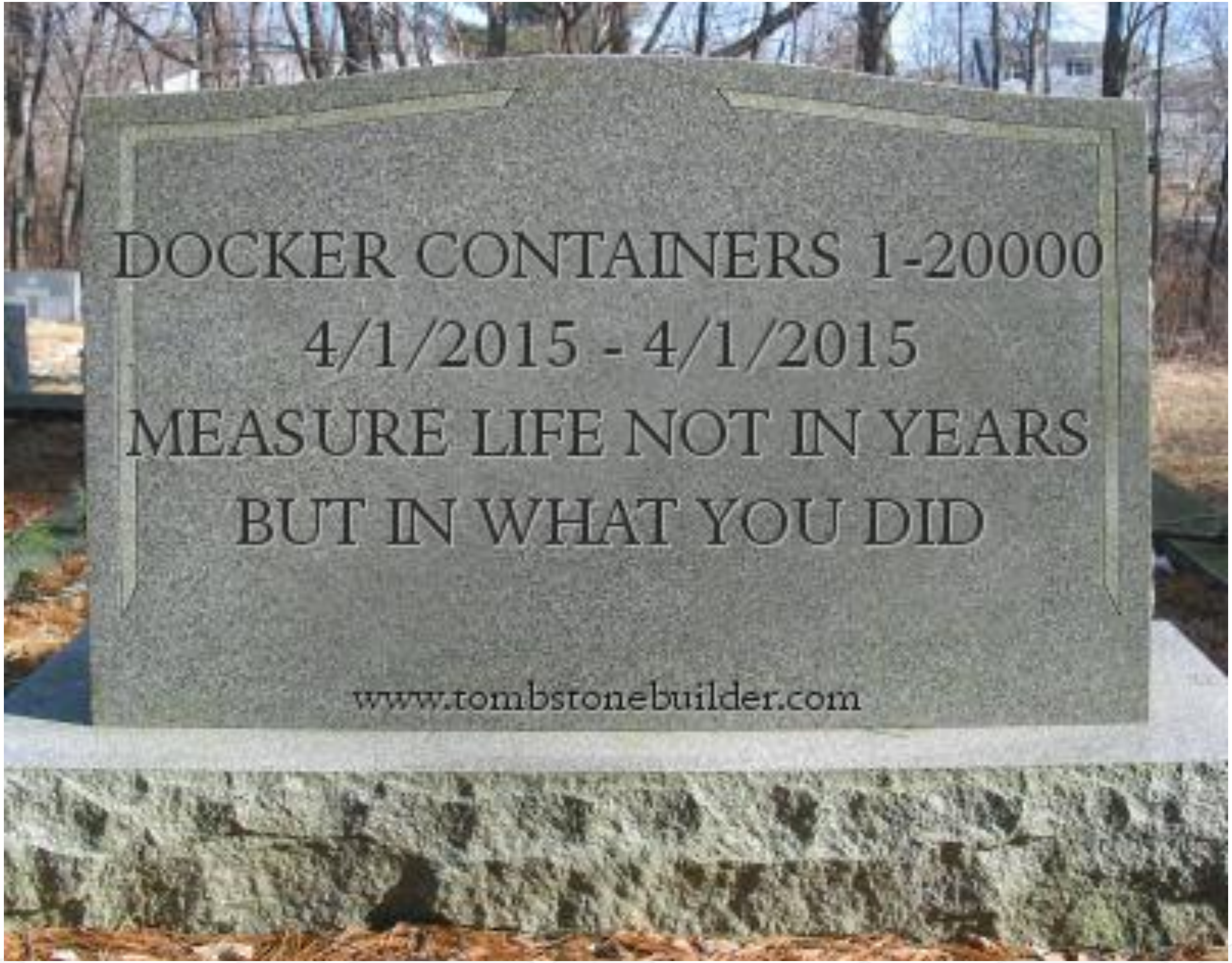
49

ACCOUNTS USING DOCKER IN LAST 24 H

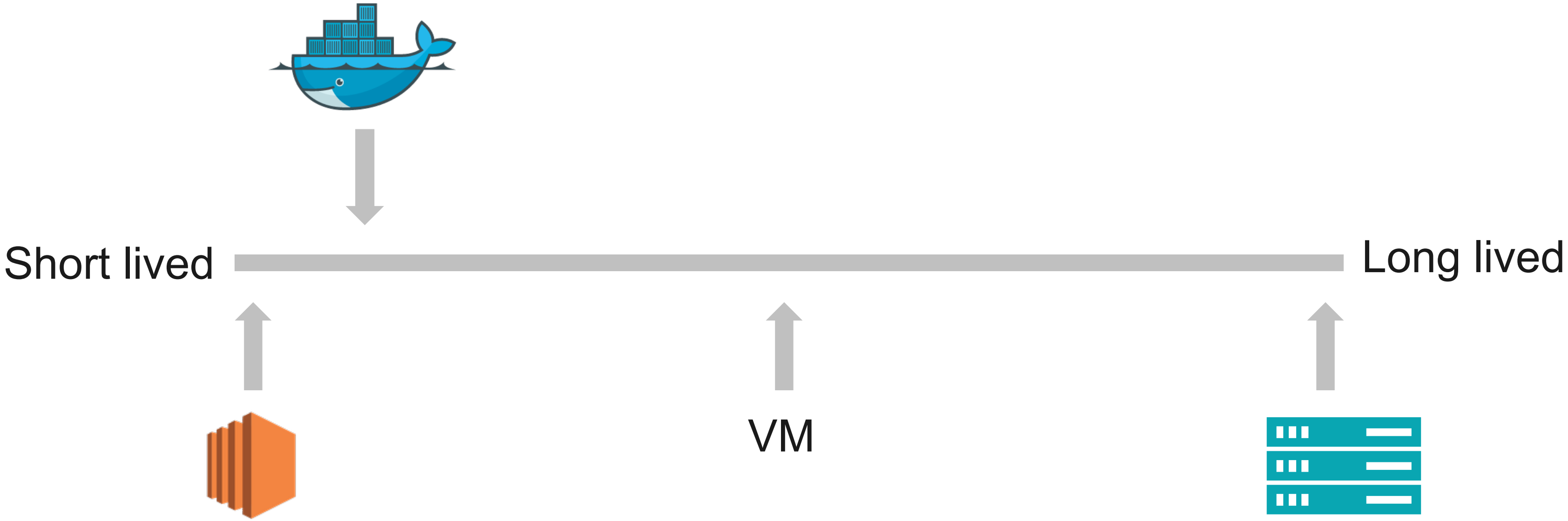
CONTAINERS REPORTING IN LAST 24 HOURS

9,974

CONTAINERS REPORTING IN LAST 24 HOURS


DOCKER CONTAINERS 1-20000
4/1/2015 - 4/1/2015
MEASURE LIFE NOT IN YEARS
BUT IN WHAT YOU DID
www.tombstonebuilder.com

Apparent customer usage



Short lived

Long lived

VM

Theory

~~Docker is a lightweight VM~~
Docker is a cloud compute container

Pets vs Cattle

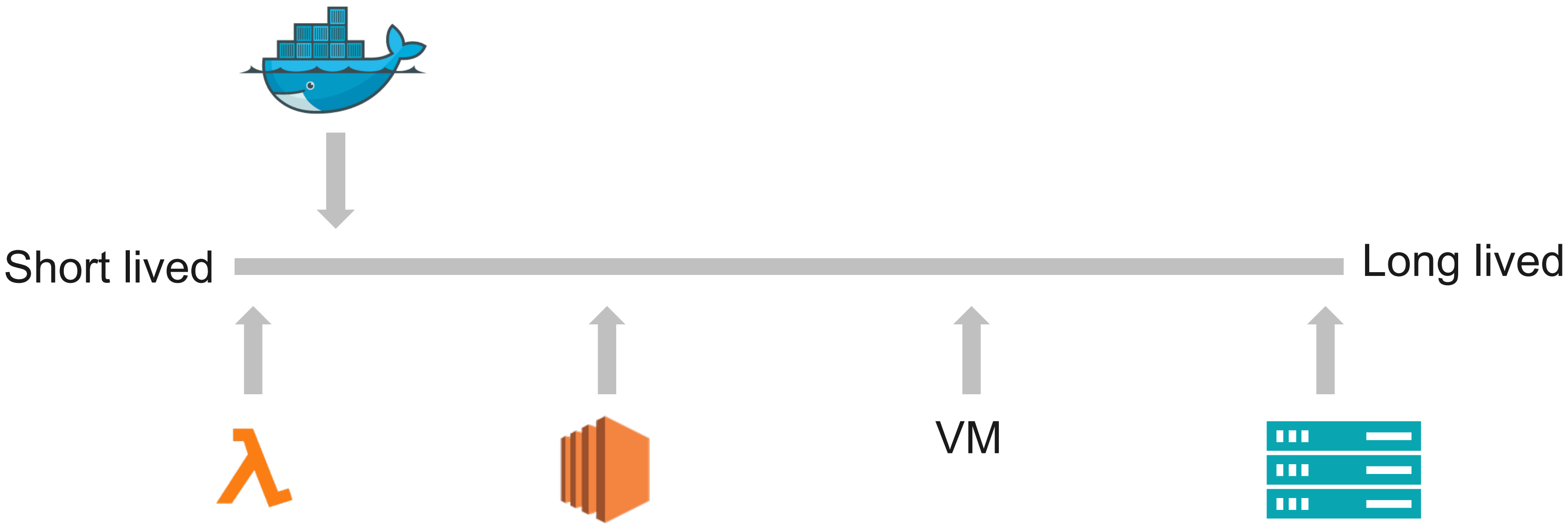
Along came New Relic Synthetics…

- User authored selenium scripts run in our data center
- Each run in its own container for security isolation
- Mostly run for under a minute

Along came New Relic Synthetics…

- Test external availability and performance
- User authored selenium scripts run in our data center
- Each run in its own container for security isolation
- Most run for under a minute

A lighter weight usage



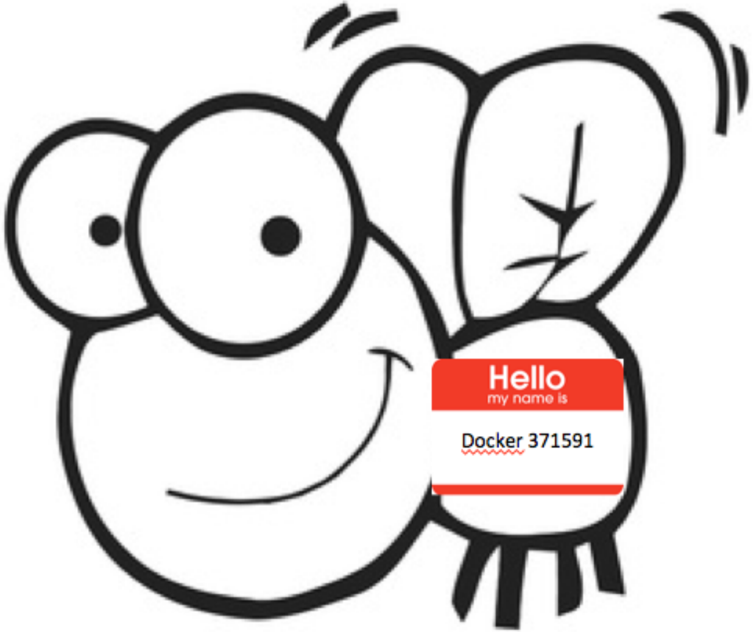
Short lived

Long lived

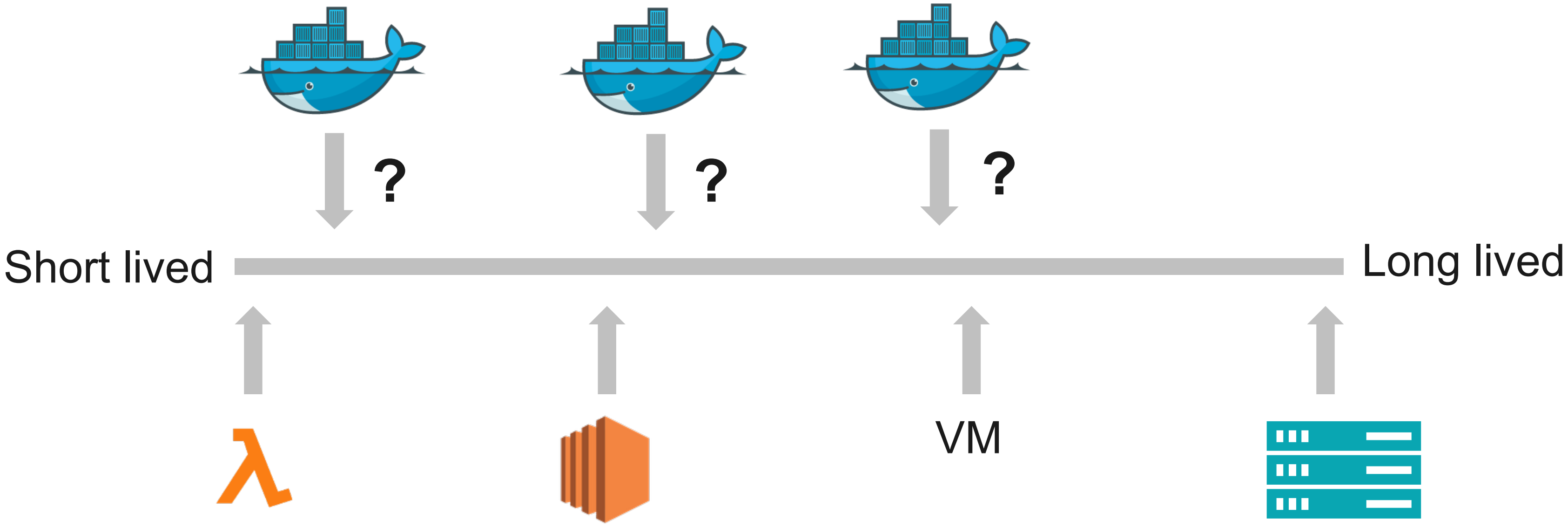
VM

Theory

~~Docker is a lightweight VM~~
~~Docker is a cloud compute container~~
Docker is a short lived compute engine

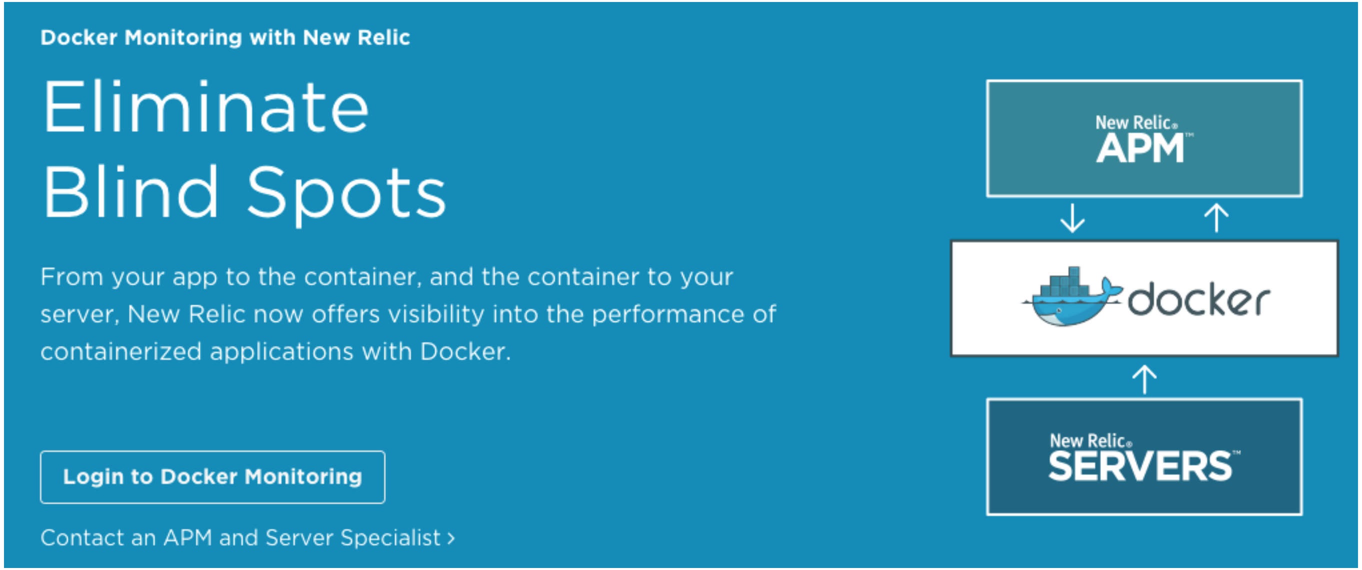


What the heck’s going on?



New Relic.

1: Hey we're monitoring it!



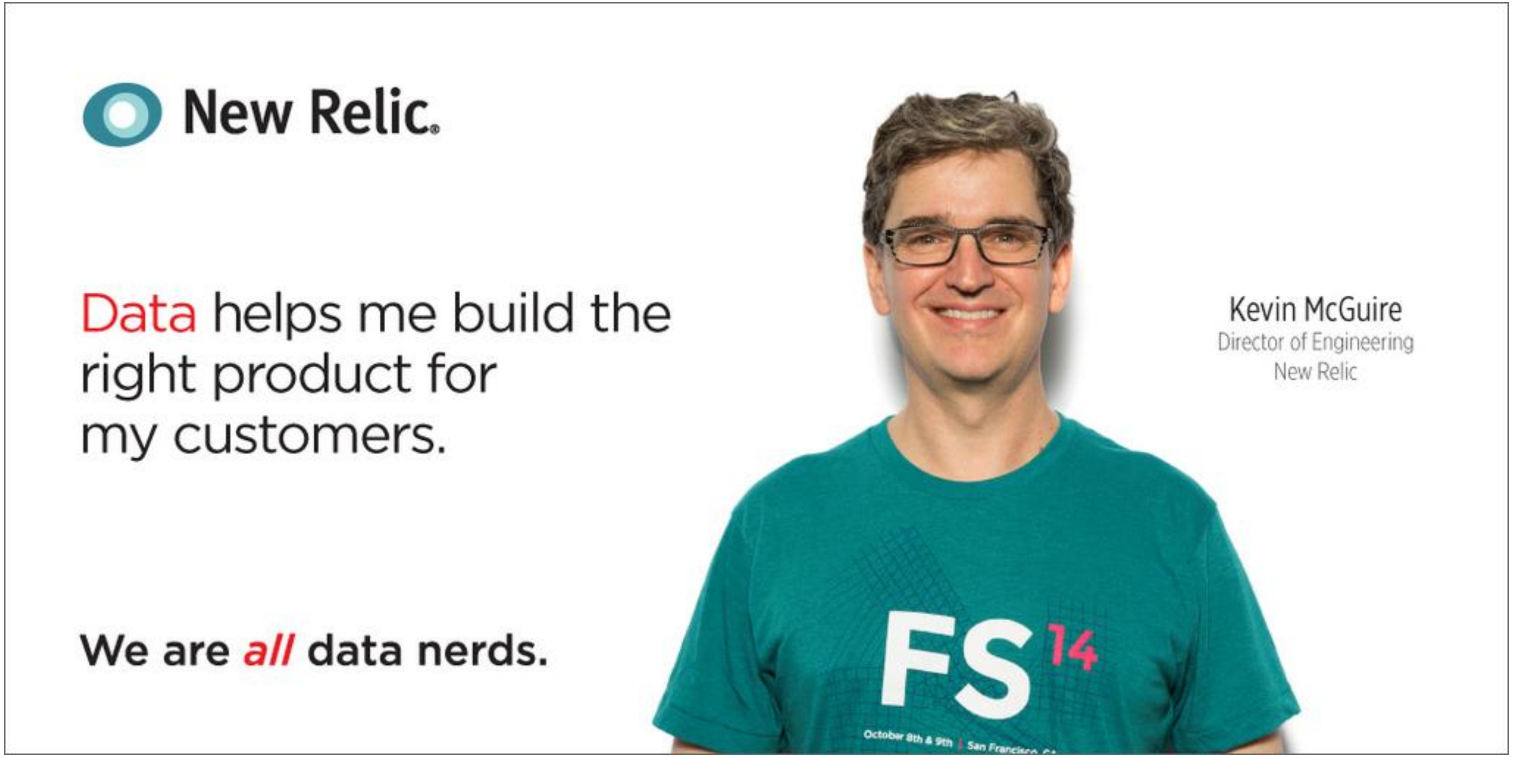
Docker Monitoring with New Relic

Eliminate Blind Spots

From your app to the container, and the container to your server, New Relic now offers visibility into the performance of containerized applications with Docker.

Login to Docker Monitoring
Contact an APM and Server Specialist ›
New Relic APM
docker
New Relic SERVERS

2: We’re Data Nerds!



New Relic.

Data helps me build the right product for my customers.

Kevin McGuire
Director of Engineering
New Relic

We are *all* data nerds.

Data set analyzed

TOTAL CONTAINERS

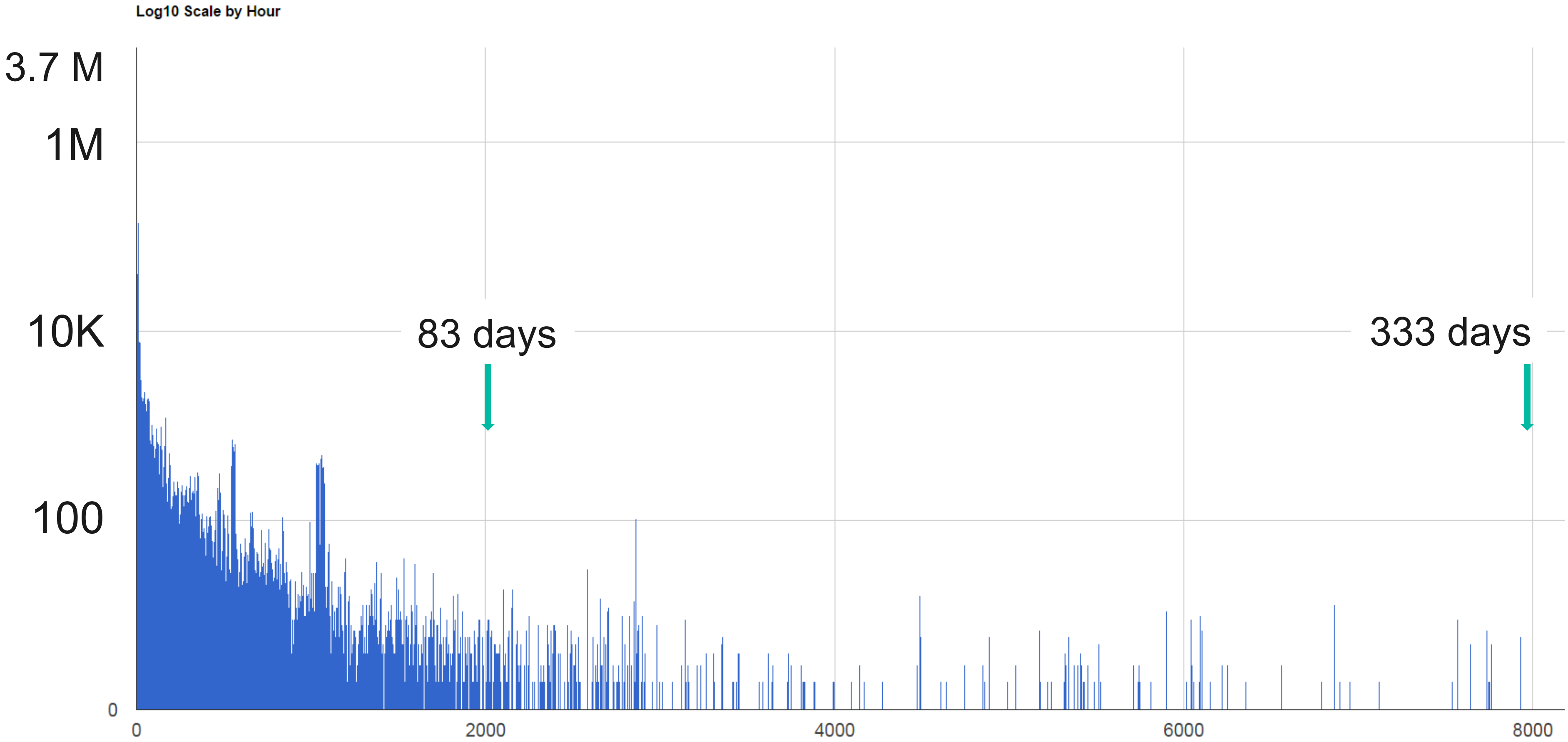
8+ million

CUSTOMERS

Approx. 1000

AVERAGE 24 HR CONTAINERS

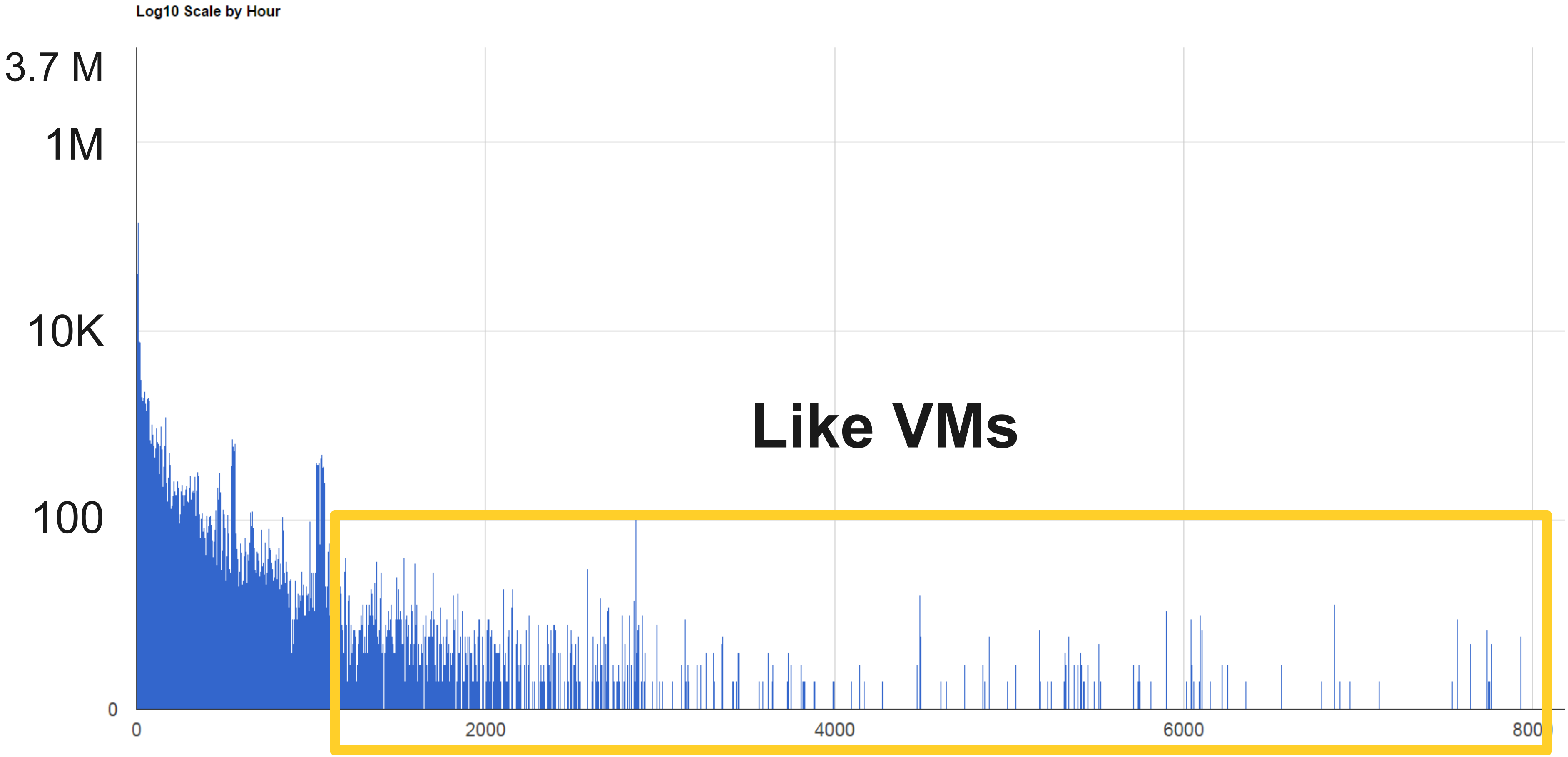
300,000+



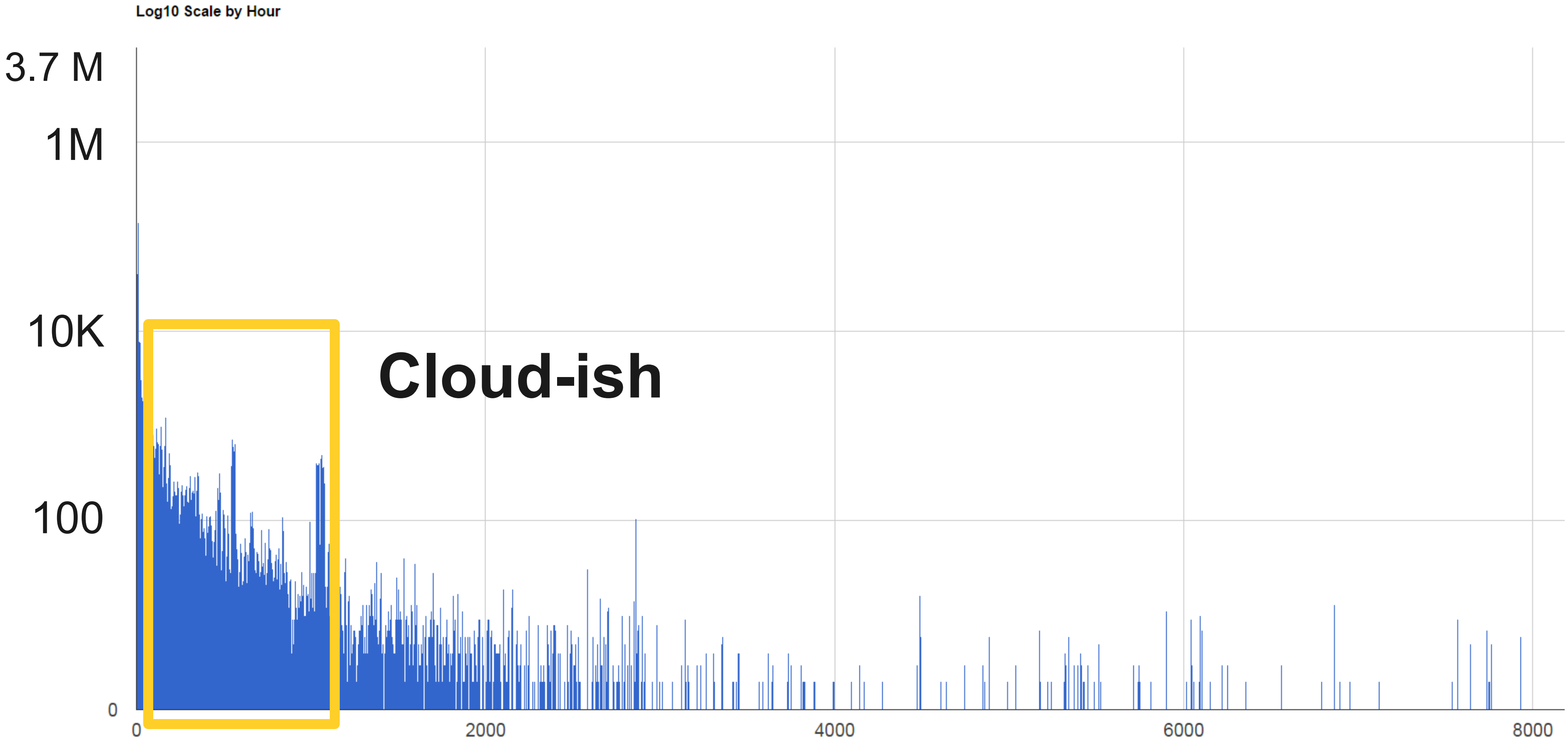
Log10 Scale by Hour

83 days

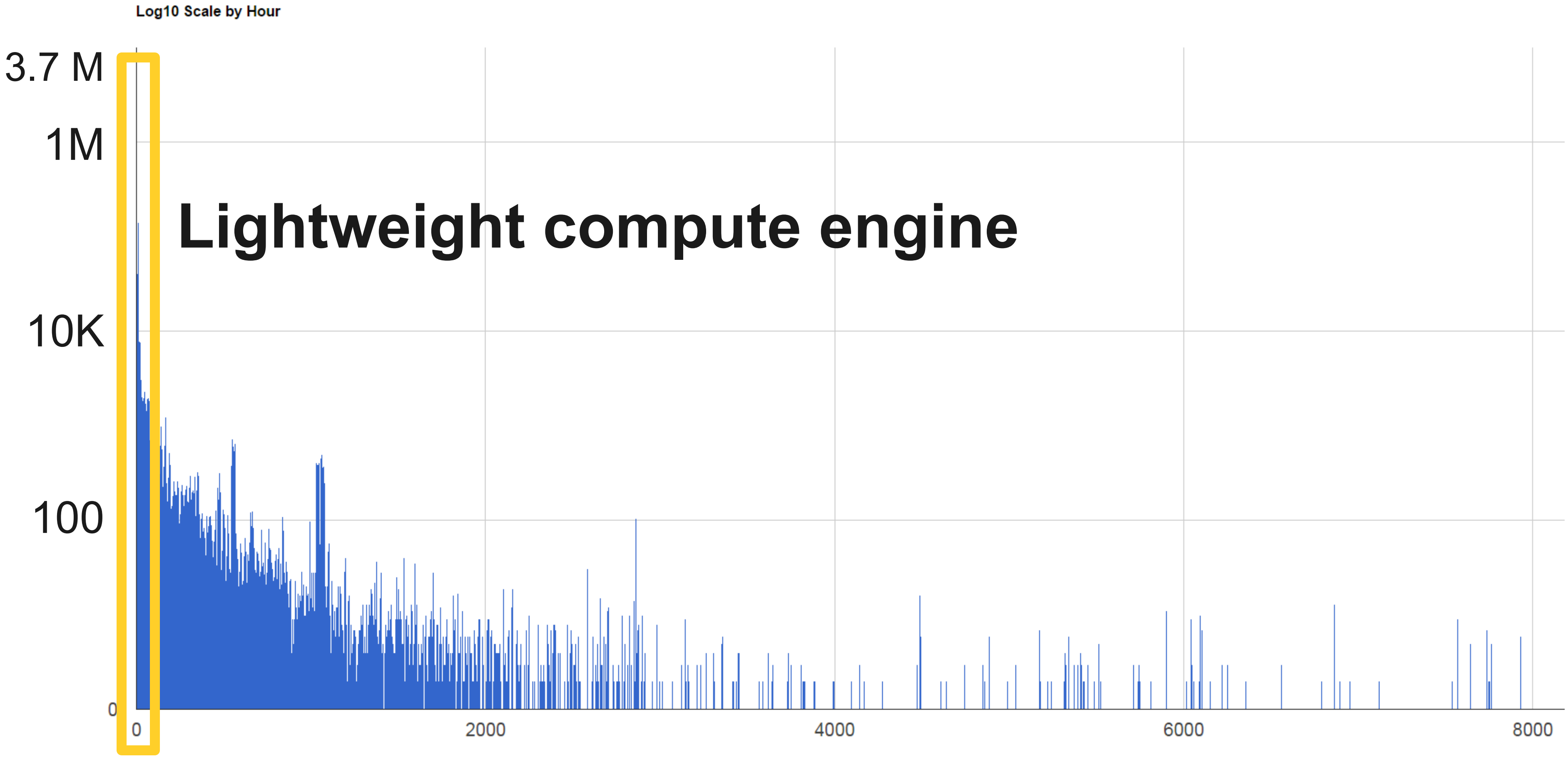
333 days



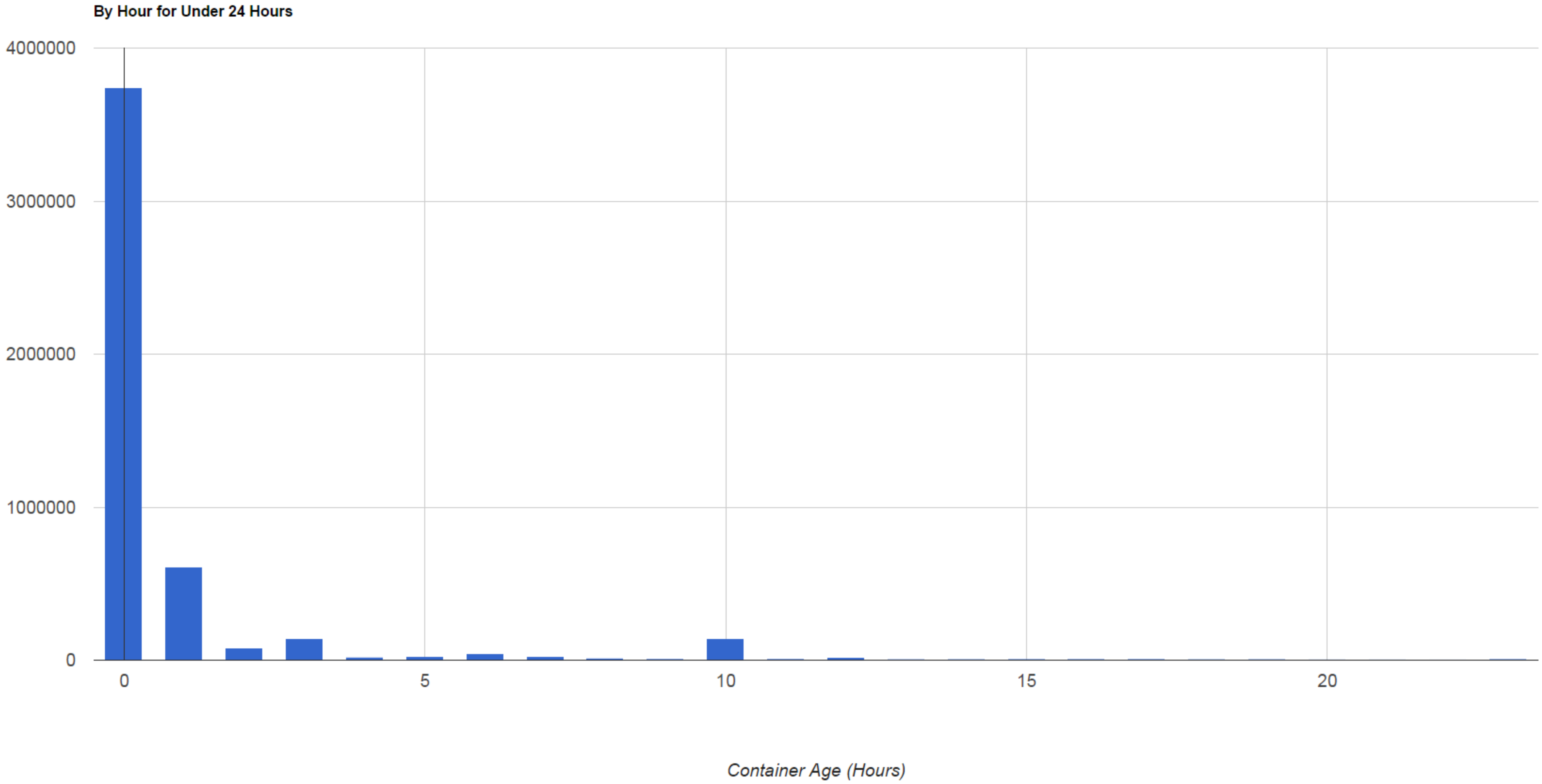
Like VMs

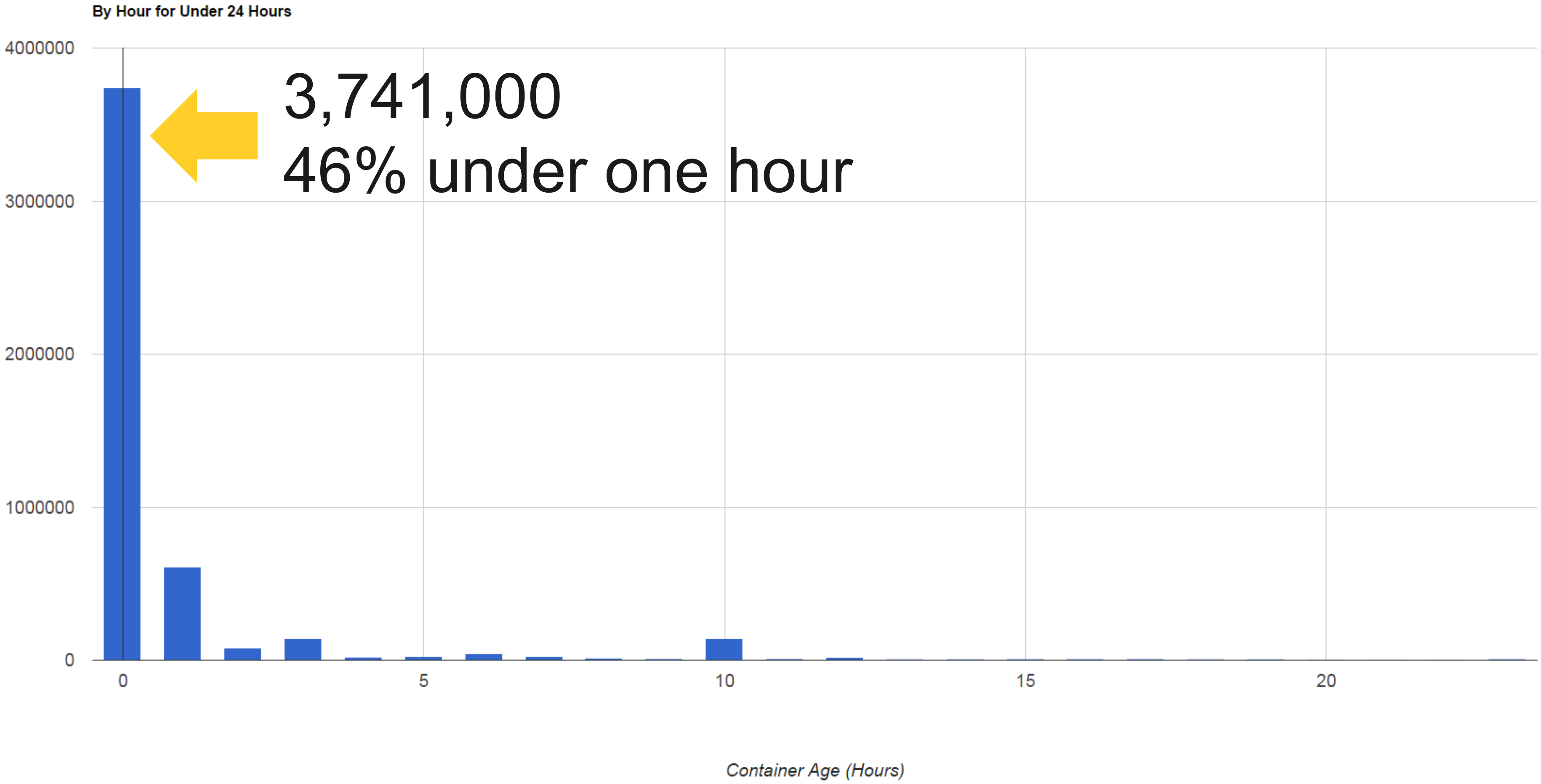

Log10 Scale by Hour
3.7 M
1M
10K
100
0
0
2000
4000
6000
8000
Cloud-ish

New Relic.



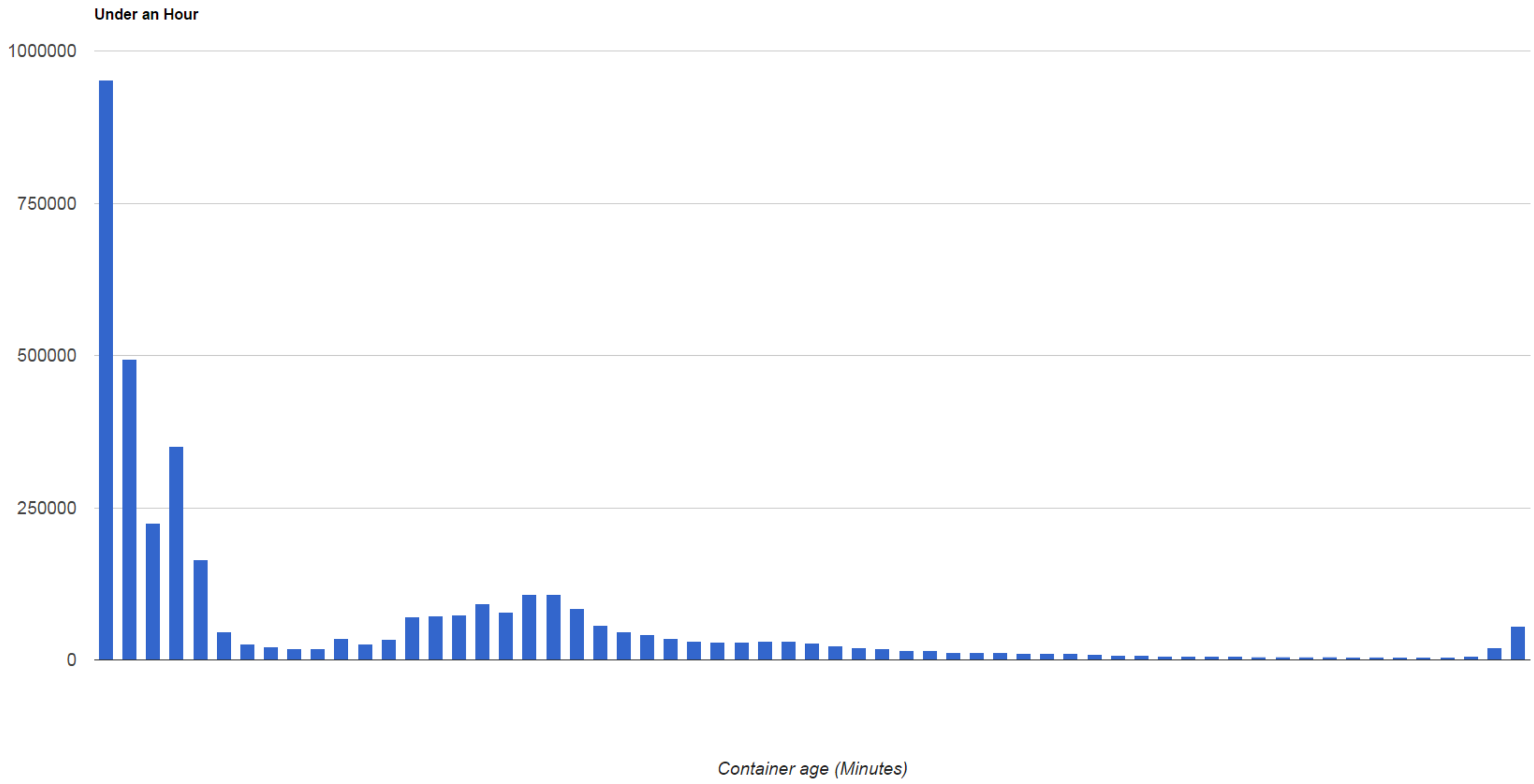
Lightweight compute engine


By Hour for Under 24 Hours
4000000
3000000
2000000
1000000
0
0
5
10
15
20
Container Age (Hours)

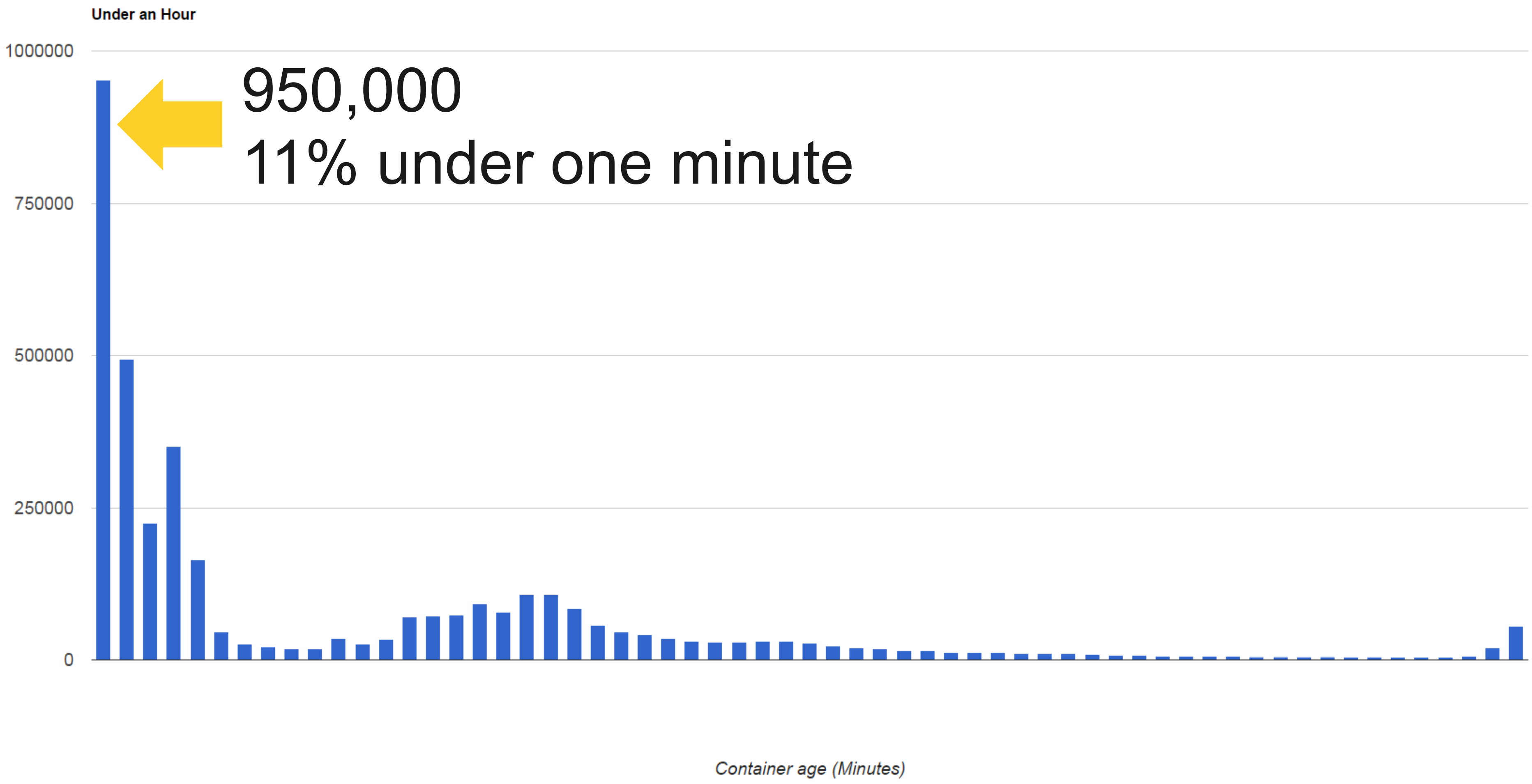


3,741,000
46% under one hour

New Relic

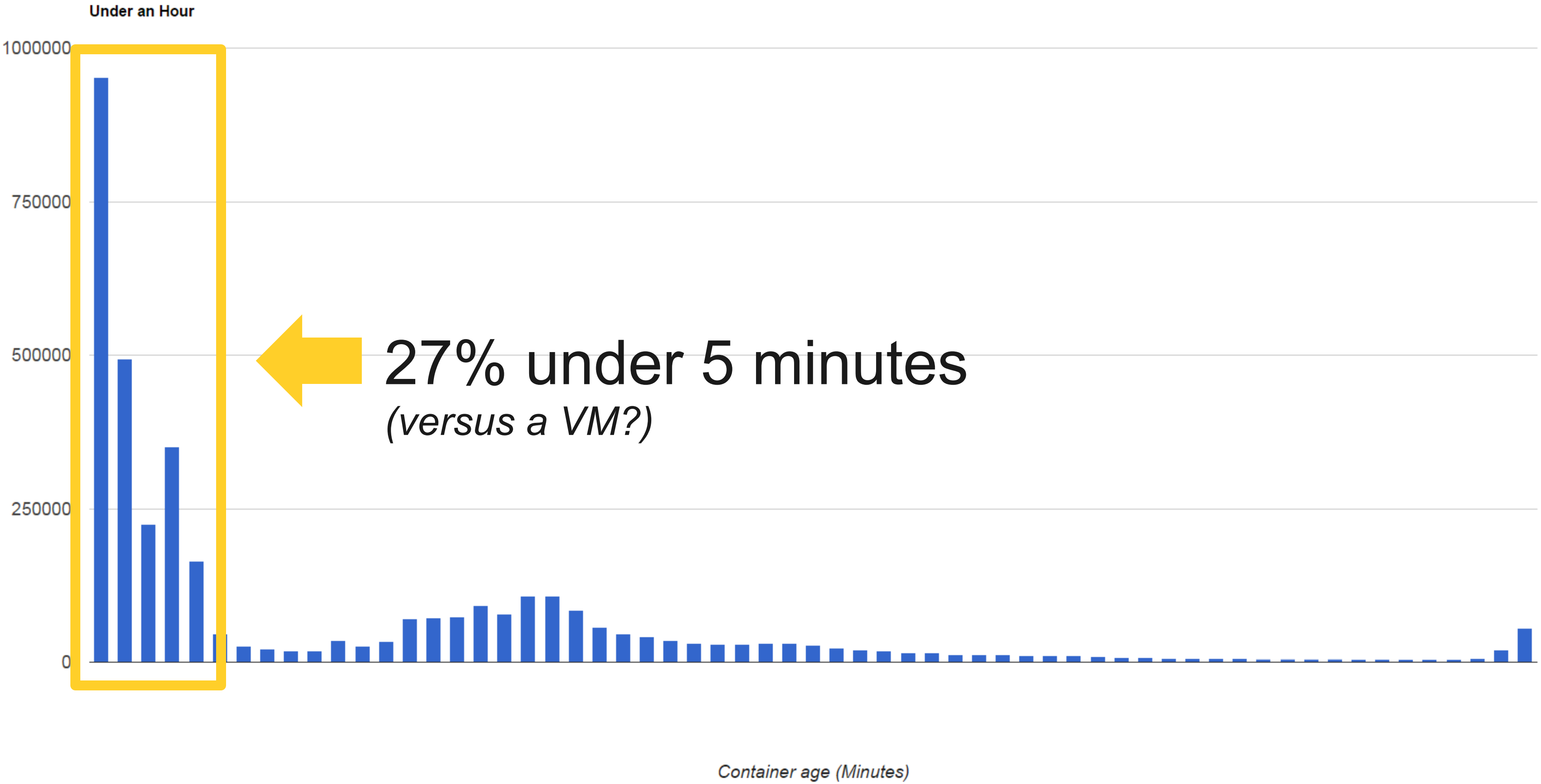

Under an Hour
1000000
750000
500000
250000
0
Container age (Minutes)

New Relic.



950,000

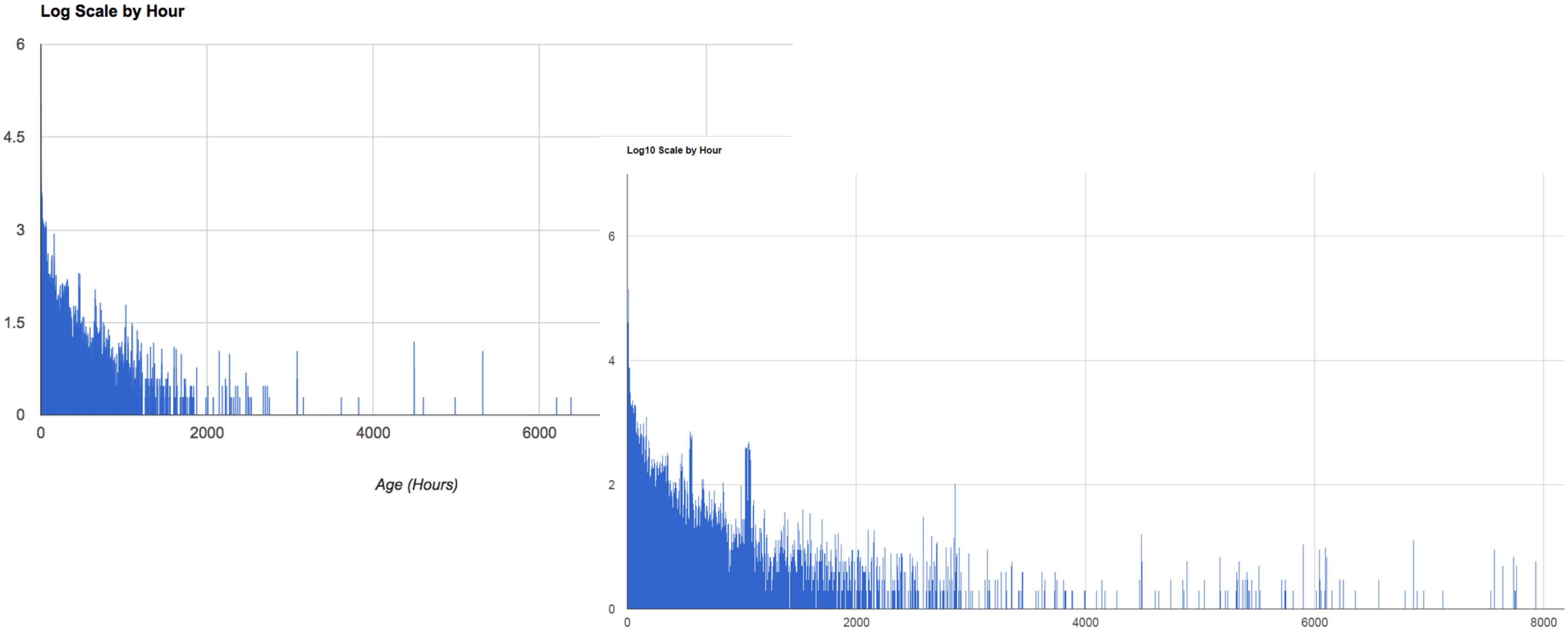
11% under one minute



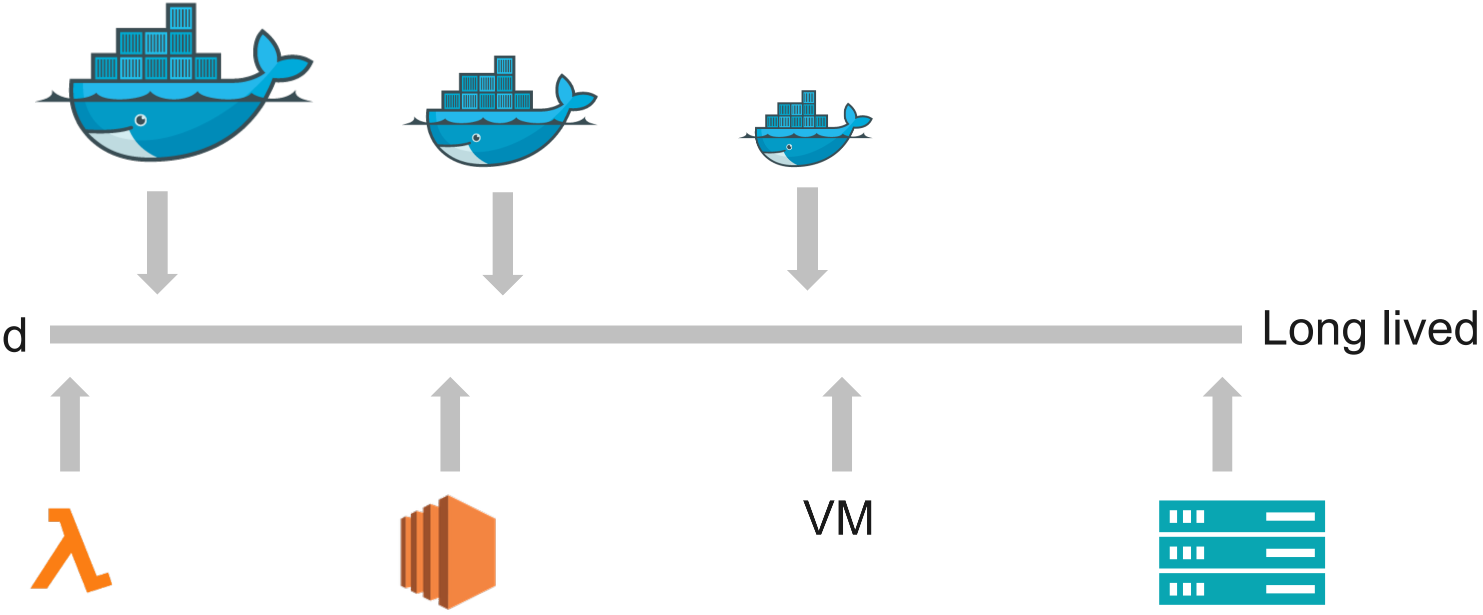
27% under 5 minutes

(versus a VM?)

June versus now: 5x data, same shape



A surprising result



New Relic.

Why this matters

1. Usage is evolving in (fascinating) unexpected ways

2. Single technology that can span such wide usage is a game changer

3. Monitoring tools need to fit the lifecycle

Takeaways

- Same technology spans a *huge* usage scope
- What explains all this?
  - Batch jobs?
  - “Microservices”?
- The missing metric: computational work
  - There will be a lot more short lived *anything*

The evolution of computation as a service

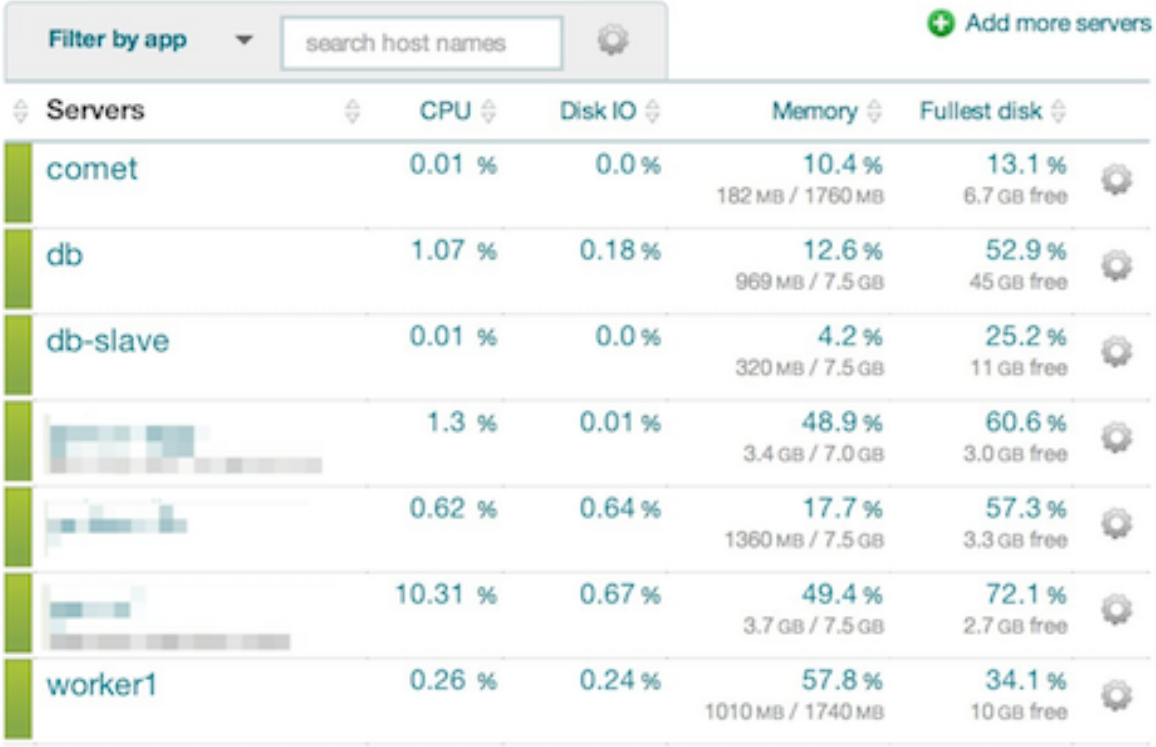
- Short startup time (orders mag.) allows very short lived computing
  - Containers are created
  - Do their work
  - Go away

- Containers only exist, and only for as long, as they provide value. Full stop.

Implications of computation as a service?

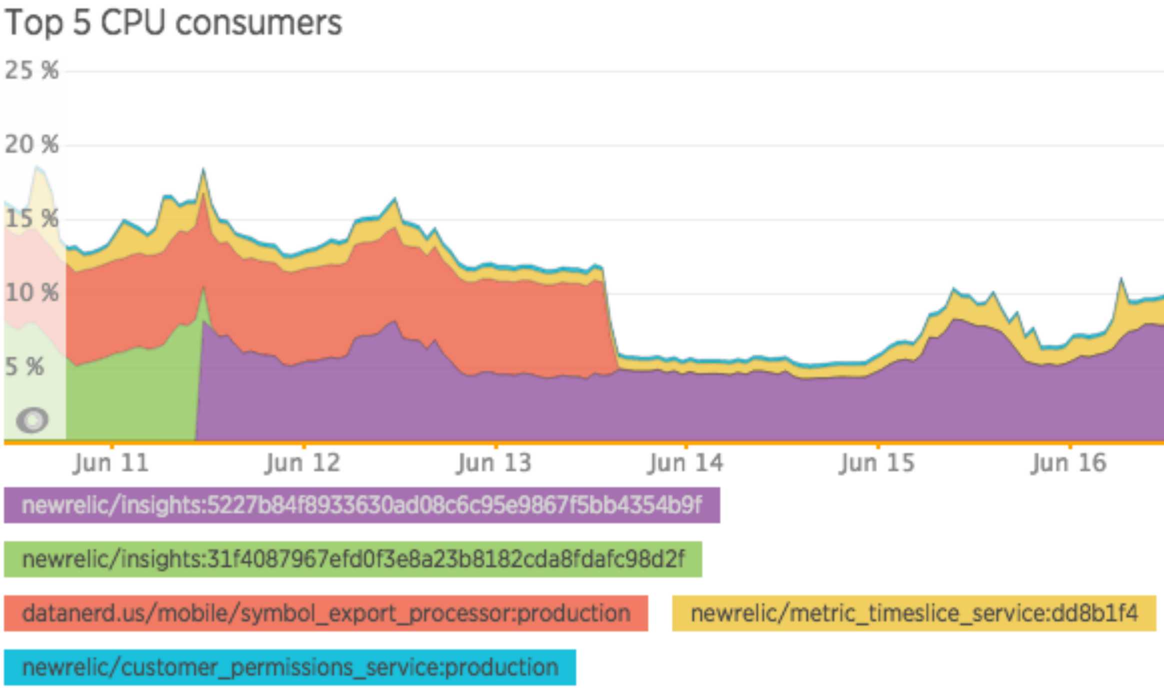
- What does it mean to network them together?
- What does it mean for orchestration of work?
- What does it mean for CI? Does it increase agility?
- What does it mean for provisioning, load balance, availability?
- How do we know what they're doing?
  (And, what is "what"?)

Monitoring servers



Servers	CPU	Disk IO	Memory	Fullest disk
comet	0.01 %	0.0 %	10.4 % 182 MB / 1760 MB	13.1 % 6.7 GB free
db	1.07 %	0.18 %	12.6 % 969 MB / 7.5 GB	52.9 % 45 GB free
db-slave	0.01 %	0.0 %	4.2 % 320 MB / 7.5 GB	25.2 % 11 GB free
[redacted]	1.3 %	0.01 %	48.9 % 3.4 GB / 7.0 GB	60.6 % 3.0 GB free
[redacted]	0.62 %	0.64 %	17.7 % 1360 MB / 7.5 GB	57.3 % 3.3 GB free
[redacted]	10.31 %	0.67 %	49.4 % 3.7 GB / 7.5 GB	72.1 % 2.7 GB free
worker1	0.26 %	0.24 %	57.8 % 1010 MB / 1740 MB	34.1 % 10 GB free

Monitoring computation



Thank you

Kevin McGuire
kevin@newrelic.com
@KevinMcGUI