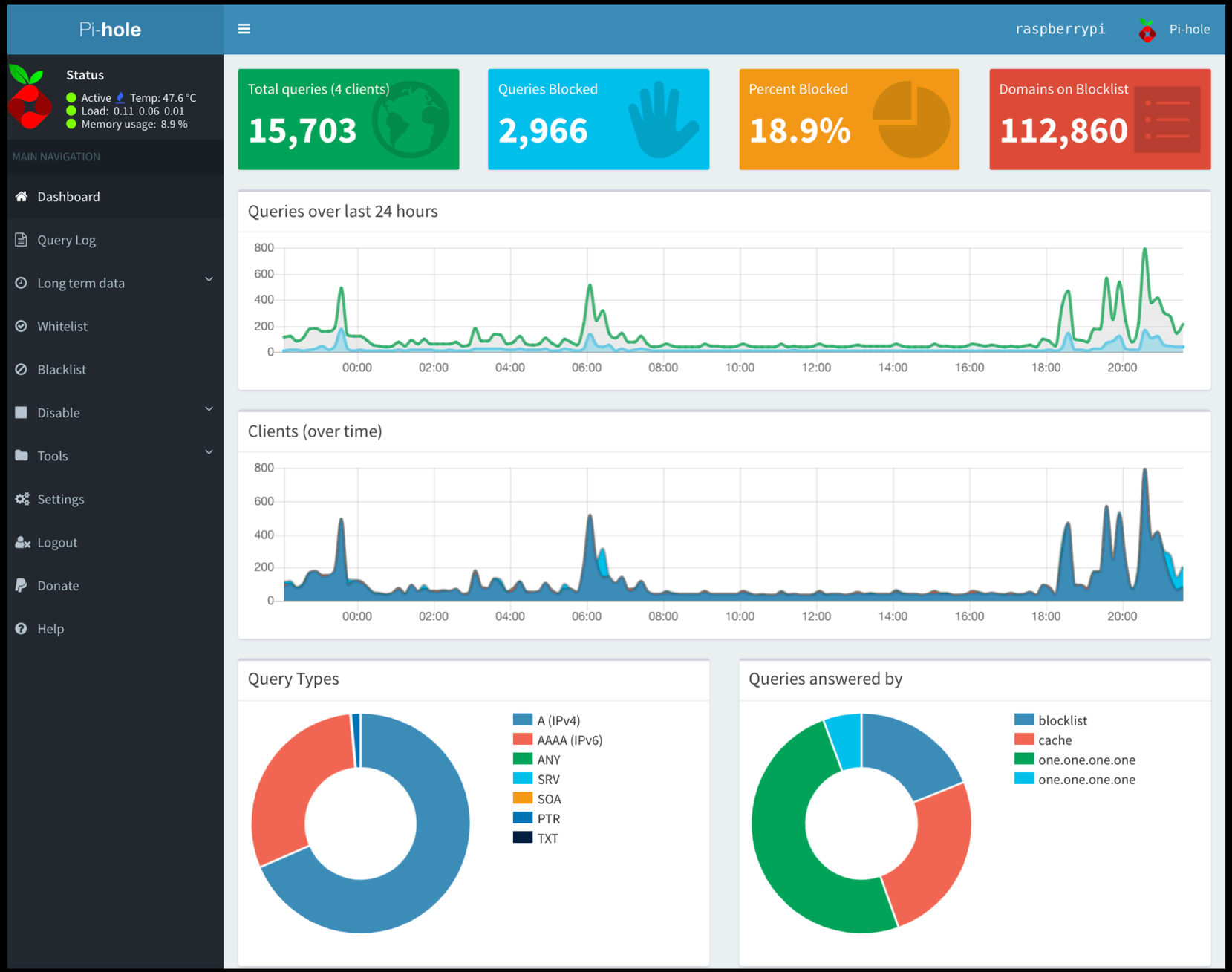
Monitor Everything/*

20%


Pi-hole
raspberrypi
Pi-hole
Status
Active
Temp: 47.6 °C
Load: 0.11 0.06 0.01
Memory usage: 8.9 %
MAIN NAVIGATION
Dashboard
Query Log
Long term data
Whitelist
Blacklist
Disable
Tools
Settings
Logout
Donate
Help
Total queries (4 clients)
15,703
Queries Blocked
2,966
Percent Blocked
18.9%
Domains on Blocklist
112,860
Queries over last 24 hours
800
600
400
200
0
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
Clients (over time)
800
600
400
200
0
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
Query Types
A (IPv4)
AAAA (IPv6)
ANY
SRV
SOA
PTR
TXT
Queries answered by
blocklist
cache
one.one.one.one
one.one.one.one

Don’t Limit Monitoring to Infrastructure

- [ ] Introduction
- [ ] Operations Overview
- [ ] Failure Models
- [ ] Demo
- [ ] Best Practices & Recap

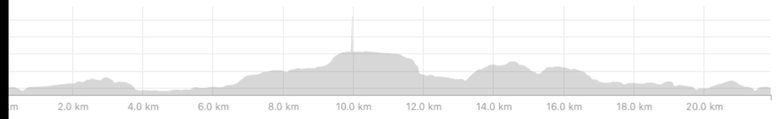
docker run who-is-brian

Brian Christner

- Co-Founder 56K.Cloud / SRE
- Docker Captain
- Passionate about Monitoring and anything with .io domains


56K.
CLOUD

Zugerberg, Zug, Switzerland



ents

Learn more about segments

	Name	Time	Speed	Power	VAM	HR
	Schluechthof In 0.79 km 11 m 0 %	2:10	22.0 km/h	218 W		153 bpm
	Bann 0.99 km 38 m 4 %	3:51	15.4 km/h	285 W		156 bpm
	Hasenjagd 0.34 km 30 m 9 %	1:58	10.6 km/h	343 W		165 bpm
☆	Fast and loose 0.39 km 46 m -12 %	42 s	33.7 km/h	96 W		150 bpm
	Aufstieg im Wald 0.76 km 45 m 6 %	3:43	12.4 km/h	301 W		165 bpm
	We Turn Red 0.21 km 5 m 2 %	34 s	22.6 km/h	352 W		148 bpm
	dark red 0.96 km 24 m 2 %	3:06	18.7 km/h	278 W		152 bpm
2	Stumpen Climb 0.51 km 17 m 3 %	1:44	17.8 km/h	306 W		158 bpm

STRAVA Dashboard Training Explore Challenges Upgrade

Fast and loose / **Effort Comparison**

Effort Comparison

Share

Fast and loose

0.3km -12% Grade 46m

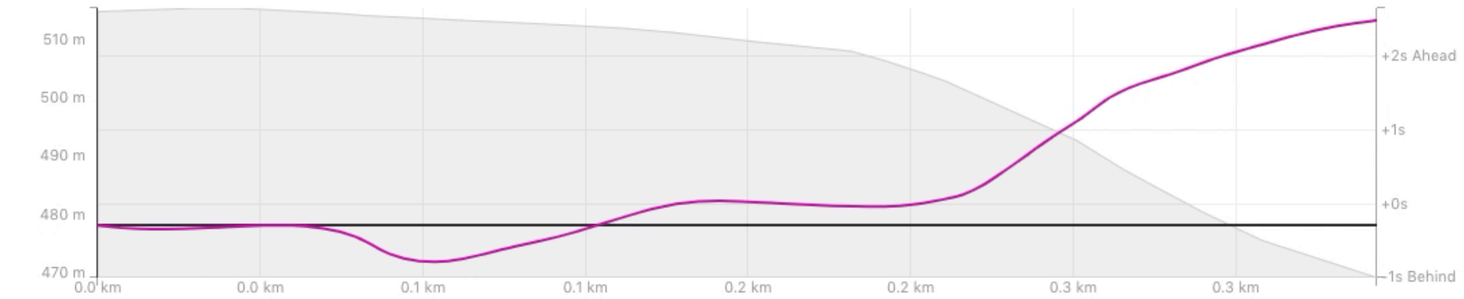
Your PR 42s
Brian Christner

KOM 39s
Dani Mendler

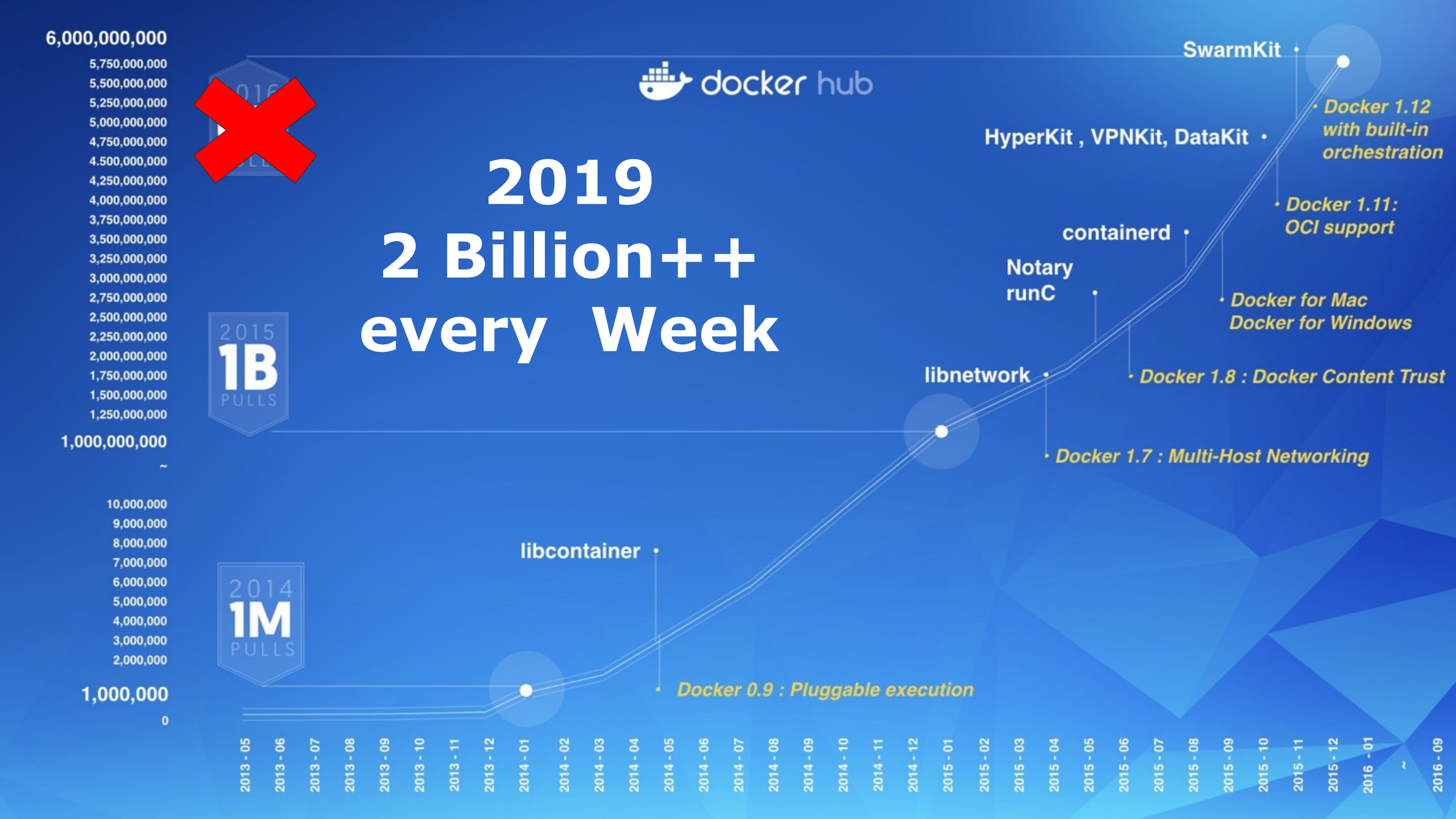
Athletes	Time	Speed
You 10/20/18 - PR	0s	36.0 km/h
Dani Mendler 11/16/18 - KOM	0s	+4.0 km/h

Customize and compare up to five efforts and see where the race was won.

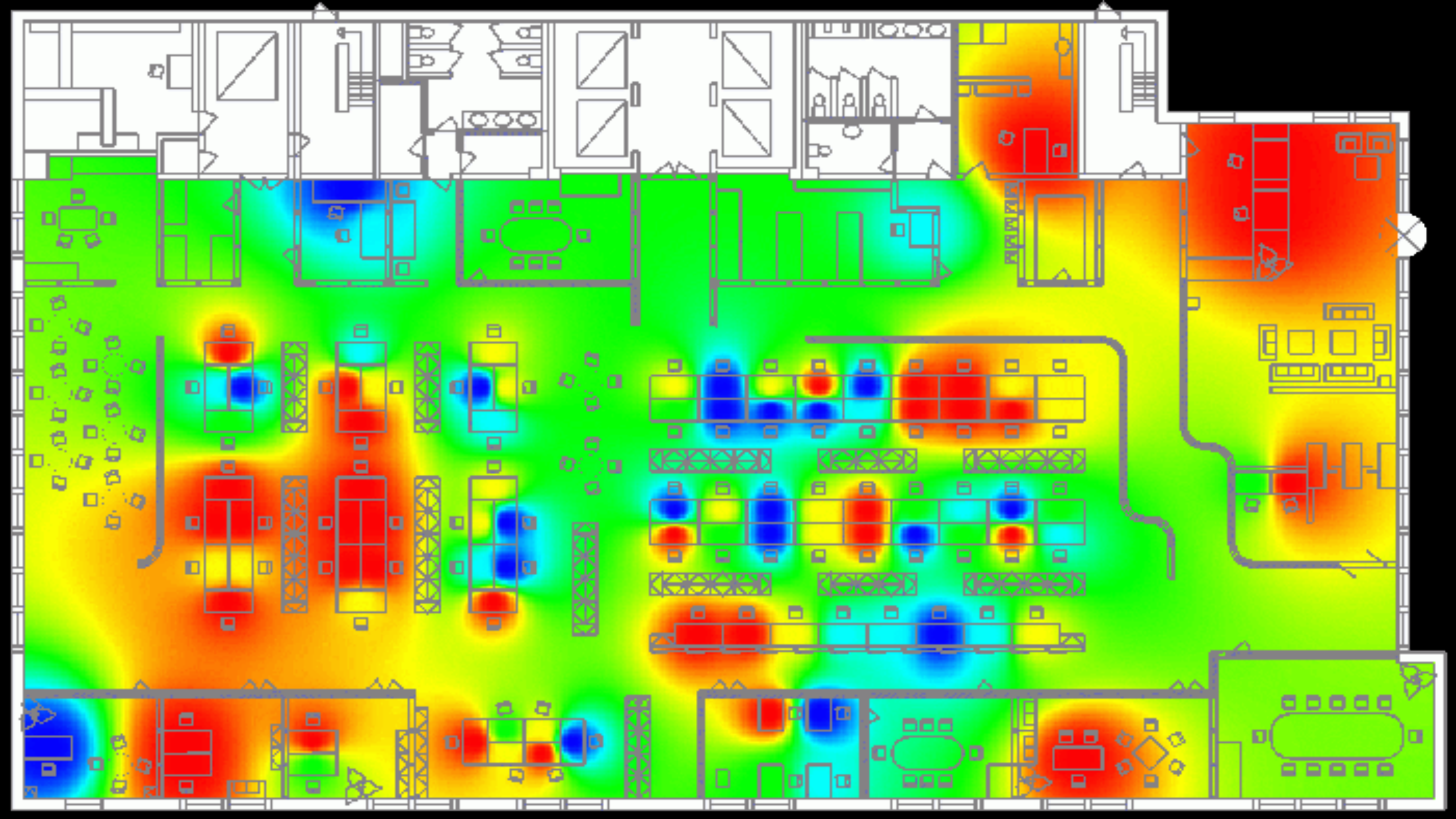
Get the Training Pack



OK, give me a business example


6,000,000,000
5,750,000,000
5,500,000,000
5,250,000,000
5,000,000,000
4,750,000,000
4.500,000,000
4,250,000,000
4,000,000,000
3,750,000,000
3,500,000,000
3,250,000,000
3,000,000,000
2,750,000,000
2,500,000,000
2,250,000,000
2,000,000,000
1,750,000,000
1,500,000,000
1,250,000,000
1,000,000,000
~
10,000,000
9,000,000
8,000,000
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
docker hub
2015
1B
PULLS
2014
1M
PULLS
2019
2 Billion++
every Week
SwarmKit
Docker 1.12
with built-in
orchestration
HyperKit , VPNKit, DataKit
Docker 1.11:
OCI support
containerd
Notary
runC
Docker for Mac
Docker for Windows
libnetwork
Docker 1.8 : Docker Content Trust
Docker 1.7 : Multi-Host Networking
libcontainer
Docker 0.9 : Pluggable execution
2013 - 05
2013 - 06
2013 - 07
2013 - 08
2013 - 09
2013 - 10
2013 - 11
2013 - 12
2014 - 01
2014 - 02
2014 - 03
2014 - 04
2014 - 05
2014 - 06
2014 - 07
2014 - 08
2014 - 09
2014 - 10
2014 - 11
2014 - 12
2015 - 01
2015 - 02
2015 - 03
2015 - 04
2015 - 05
2015 - 06
2015 - 07
2015 - 08
2015 - 09
2015 - 10
2015 - 11
2015 - 12
2016 - 01
~
2016 - 09

Casinos

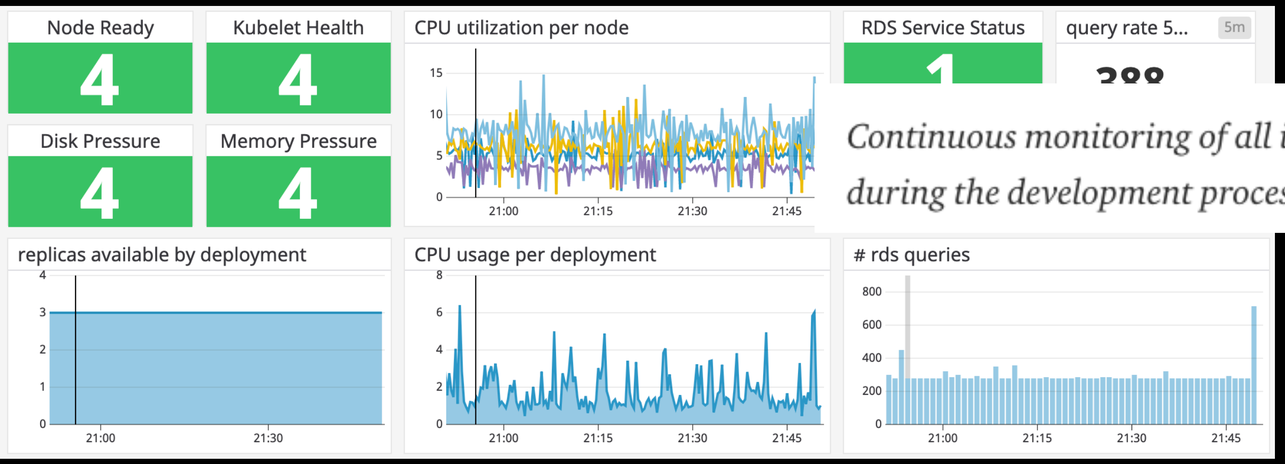


Industrial machines

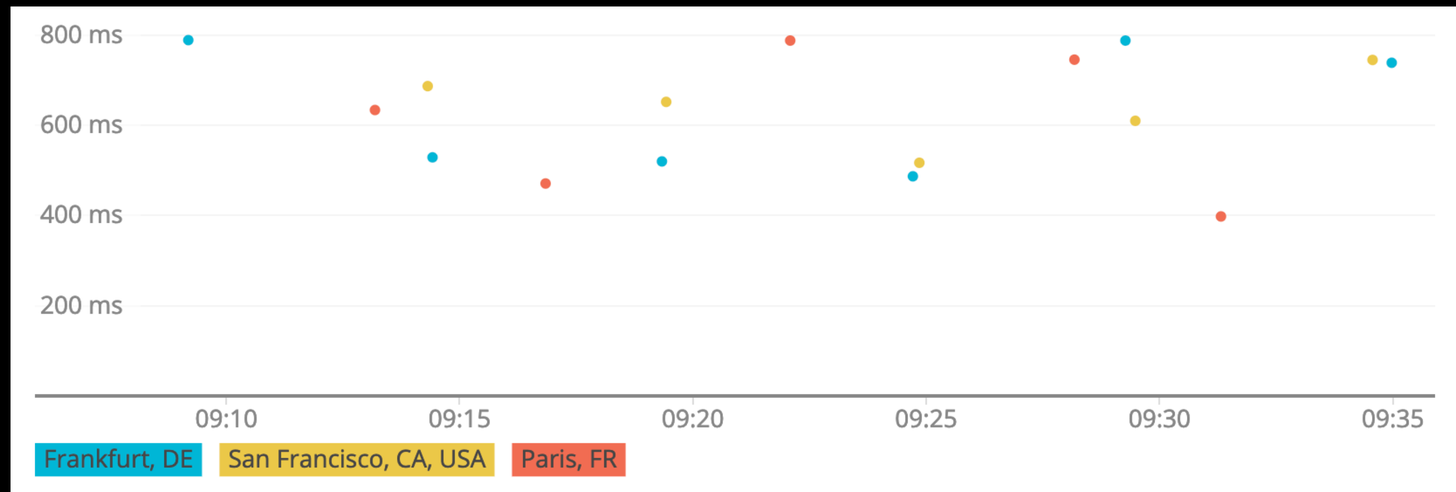
Monitor Early

www.your-now.com

A joint collaboration between BMW & Daimler AG



Continuous monitoring of all infrastructure and application components already during the development process is key to a successful launch.



- [X] Introduction

- [ ] Operations Overview

- [ ] What to Monitor

- [ ] Demo

- [ ] Best Practices & Recap

Ops Paradise

- Everything is Automated
- Reduce Costs
- No support calls / support tickets

Ops Firefighting

Operational Models

Manual	• User initiated • Interactive, command-line tools, simple scripts • Checklist and process driven
Reactive	• Hardware-centric data collection • Simple metric and log collection • Siloed tools and information • Manual analysis and remediation
Proactive	• Application-centric data collection • End-to-end **observability** • **Key metrics** and thresholds well understood • Semi-automated analysis and remediation

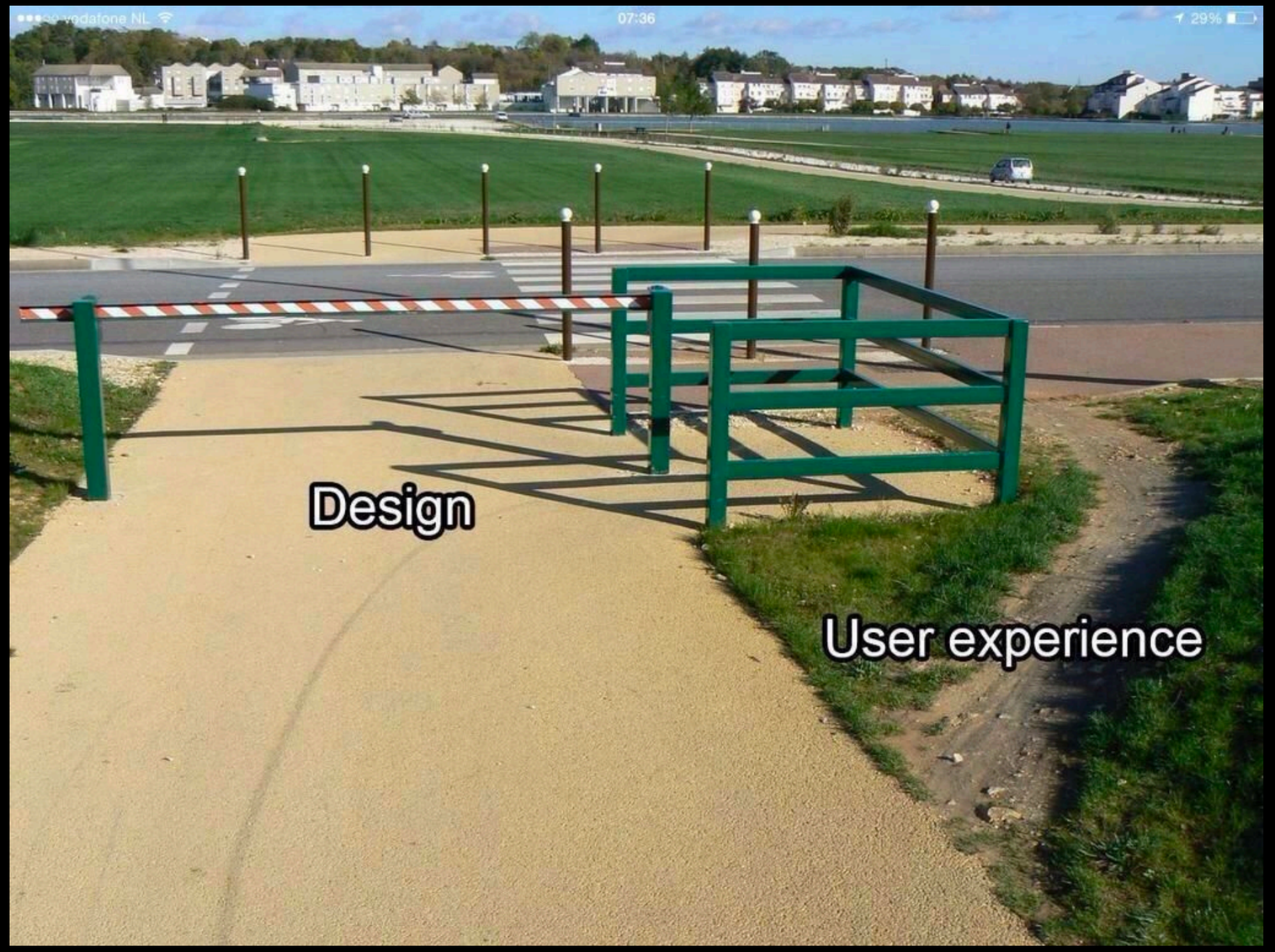
Users Care about 3 Things

- **Availability** - Is my System Online Yes/No
- **Latency** - Does it take a long time to access applications x, y, z
- **Reliability** - Can the user rely on using the application

Brain Based Tools

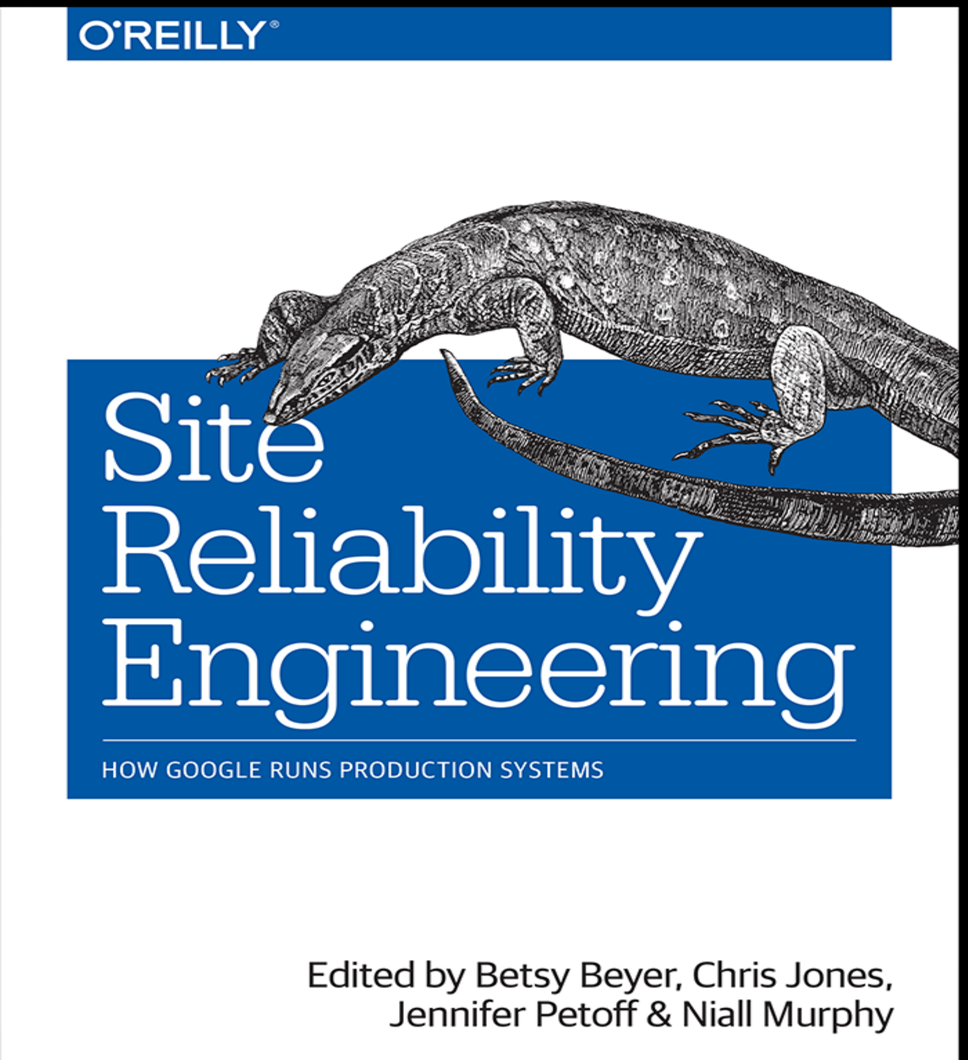
- We can track 8 objects on average
- 4 Moving Objects
- Build Dashboards & Tools accordingly

Design vs Experience



SRE

SRE (Site Reliability Engineering)

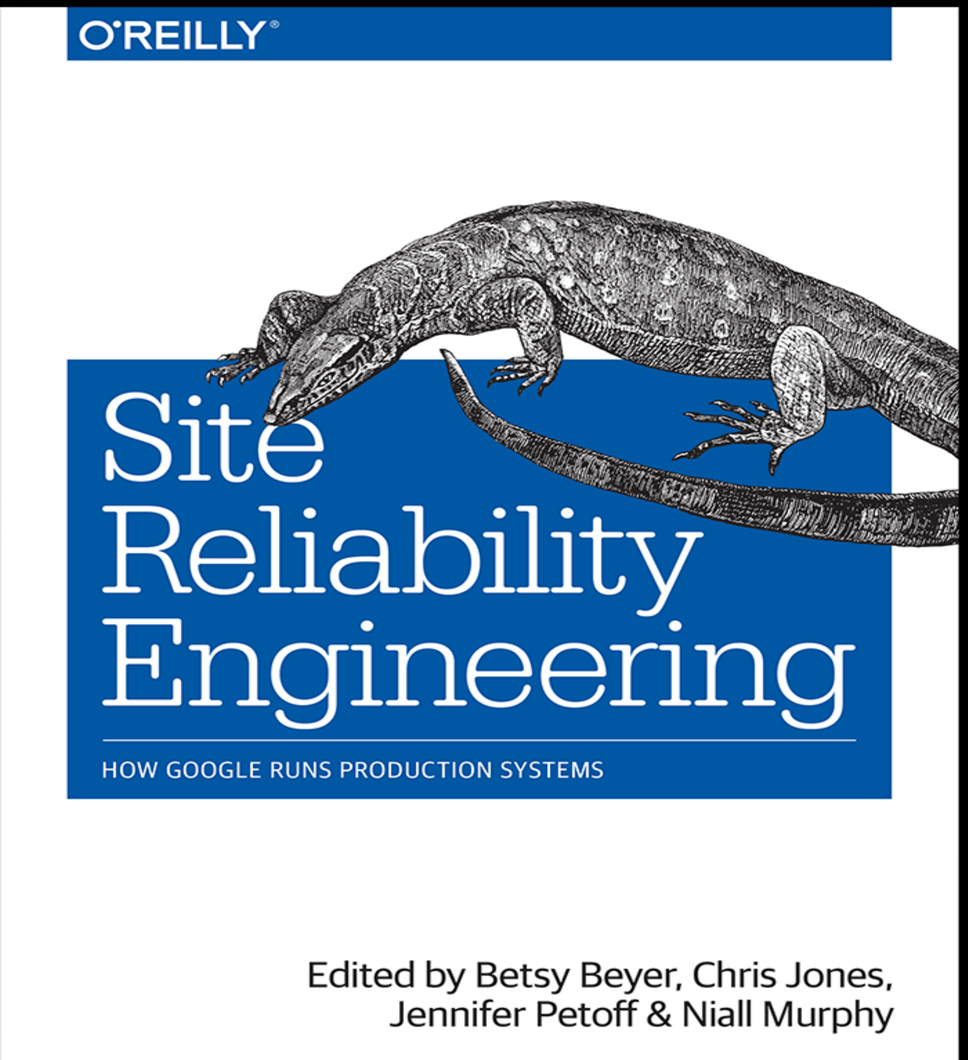


SRE is treats Operations as if it were a Software Problem

"Hope is not a strategy."
Traditional SRE saying

www.google.com/sre

TL;DR - SRE



4 Golden Signals


Latency


Traffic


Errors


Saturation

R.E.D (Microservice Level)

(Request) Rate: the number of requests, per second, you services are serving.

(Request) Errors: the number of failed requests per second. Utilization: the average time that the resource was busy servicing work

(Request) Duration: distributions of the amount of time each request takes.

U.S.E (Low Level / Infrastructure)

For every resource, check Utilization, Saturation, and Errors

Resource: **all physical server functional components (CPUs, disks, busses, ...)**

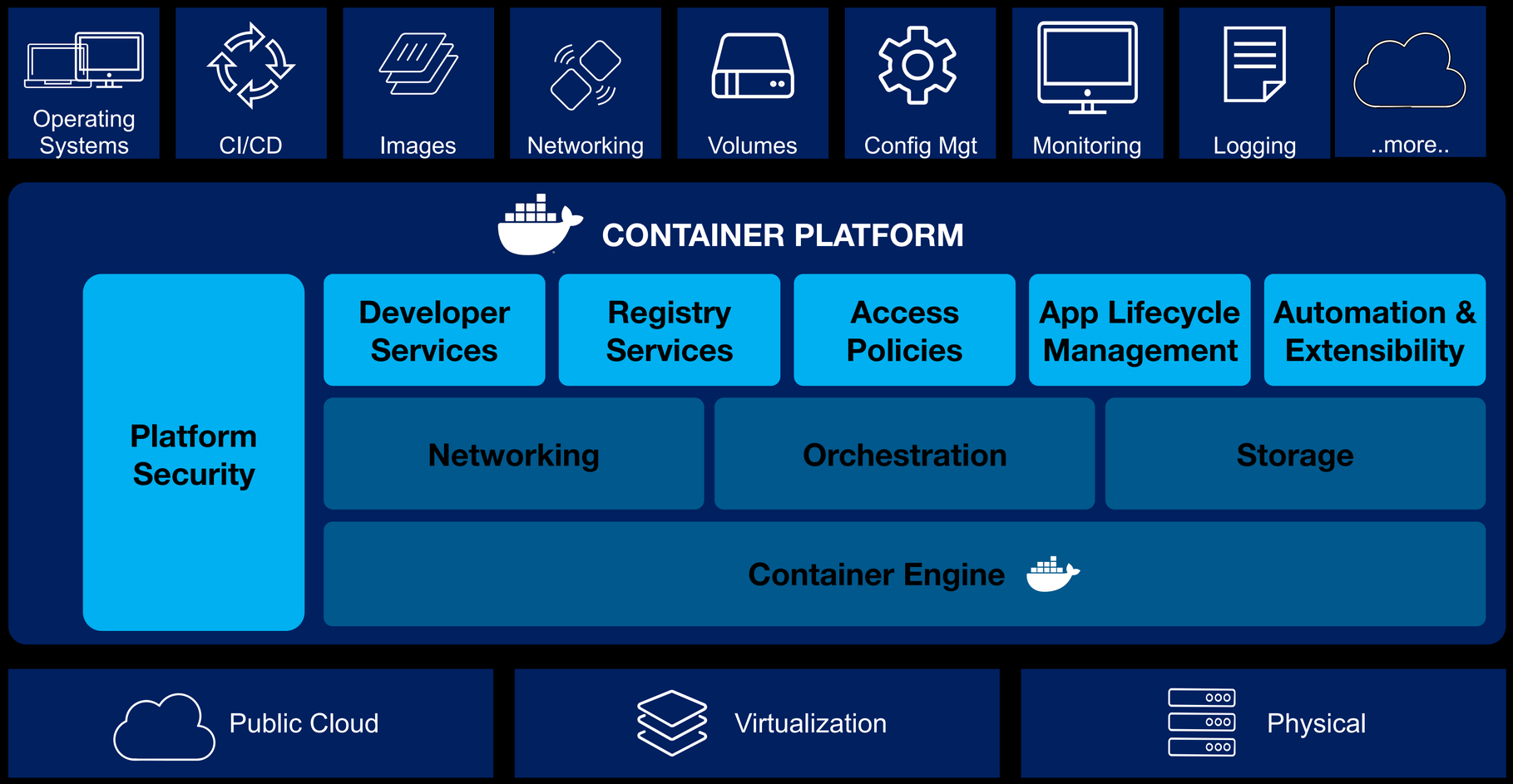
Utilization: the average time that the resource was busy servicing work

Saturation: the degree to which the resource has extra work which it can't service, often queued

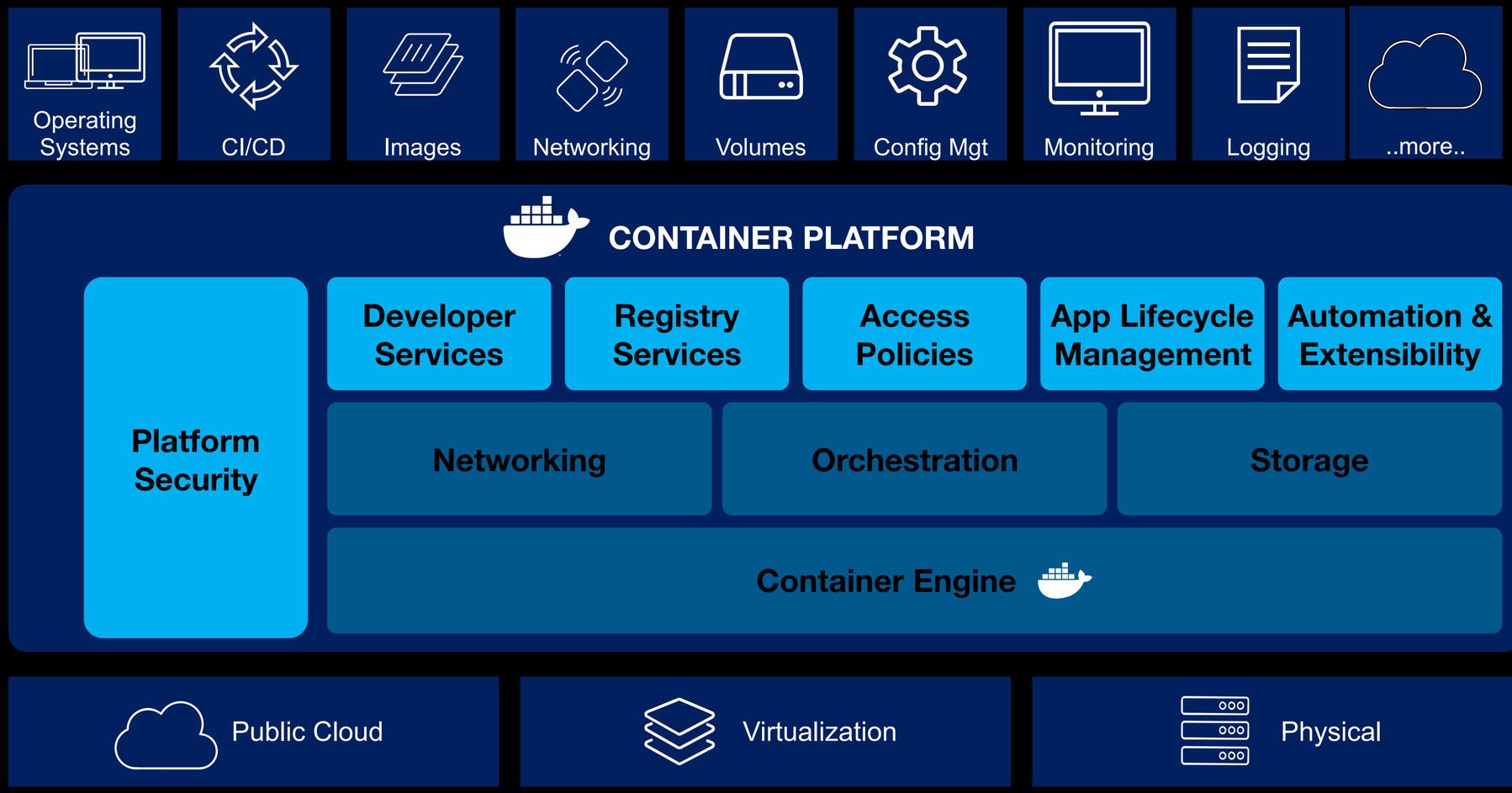
Errors: **the count of error events**

- [X] Introduction
- [X] Operations Overview
- [ ] What to Monitor
- [ ] Demo
- [ ] Best Practices & Recap

Understanding Failure Models



Host / Hardware



CPU

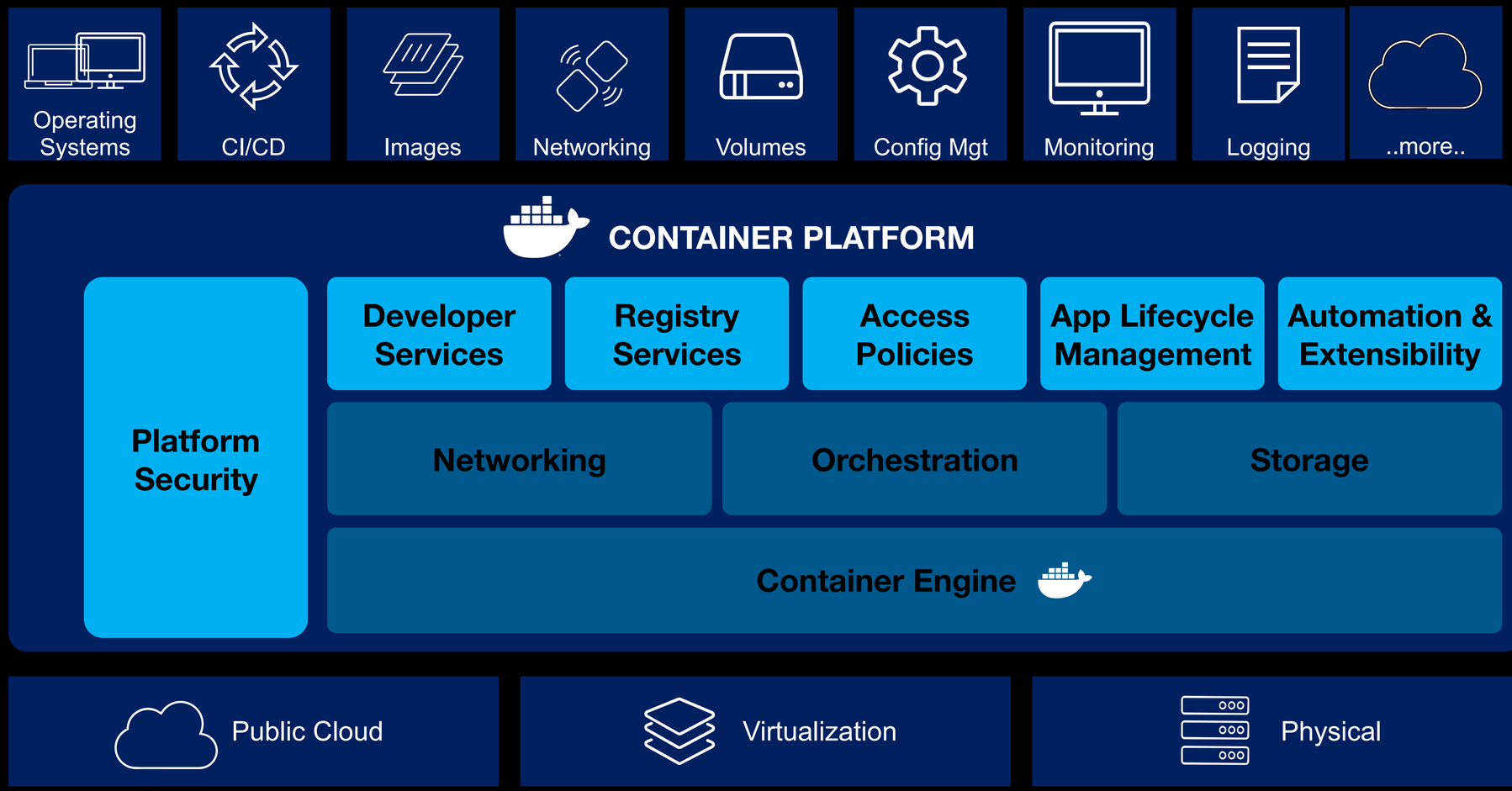
Memory

Liveness

File Descriptors

Storage Capacity

Networking



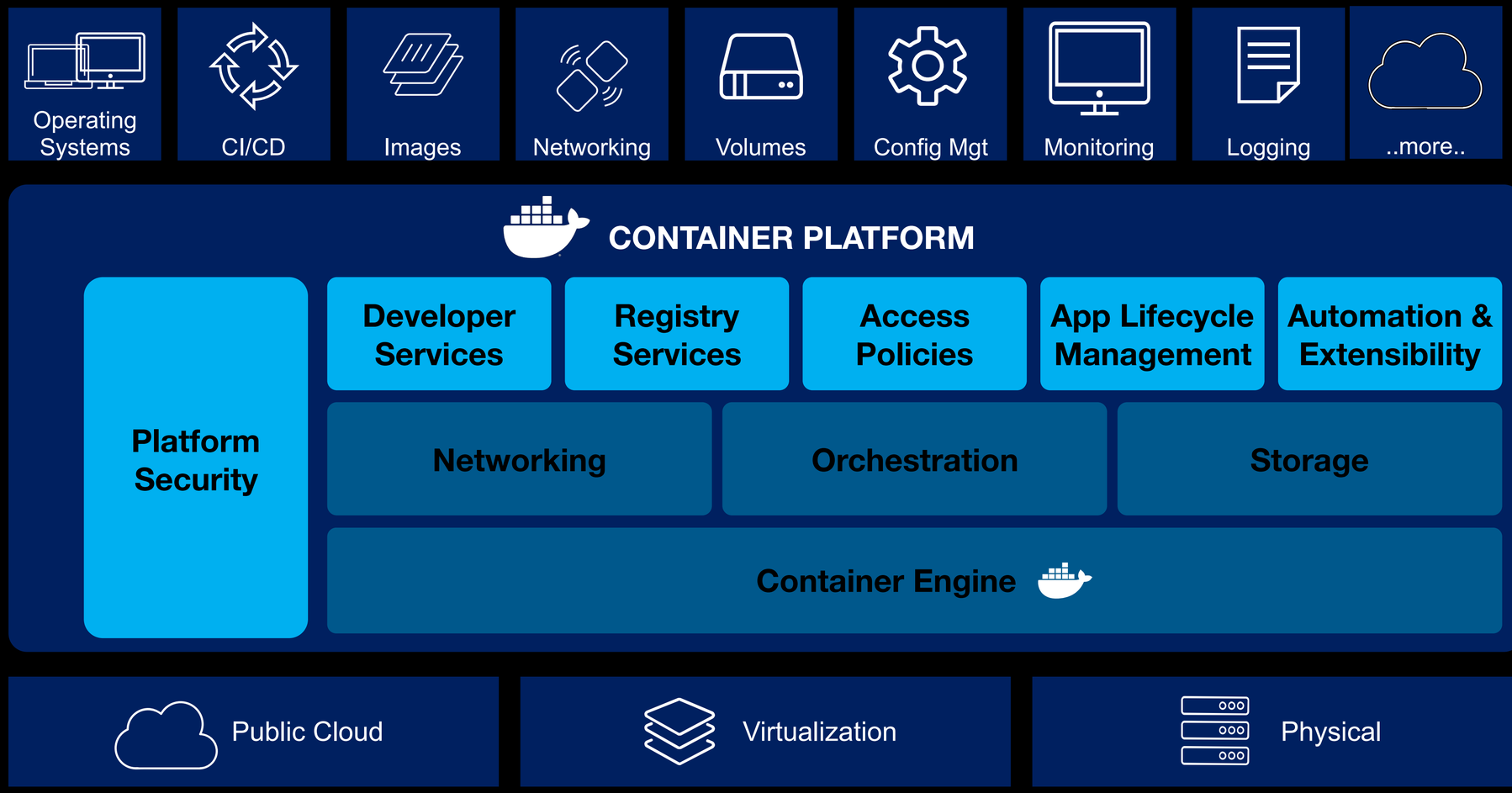
Reachability

Link Utilization

File Descriptors

Storage Capacity

Orchestration



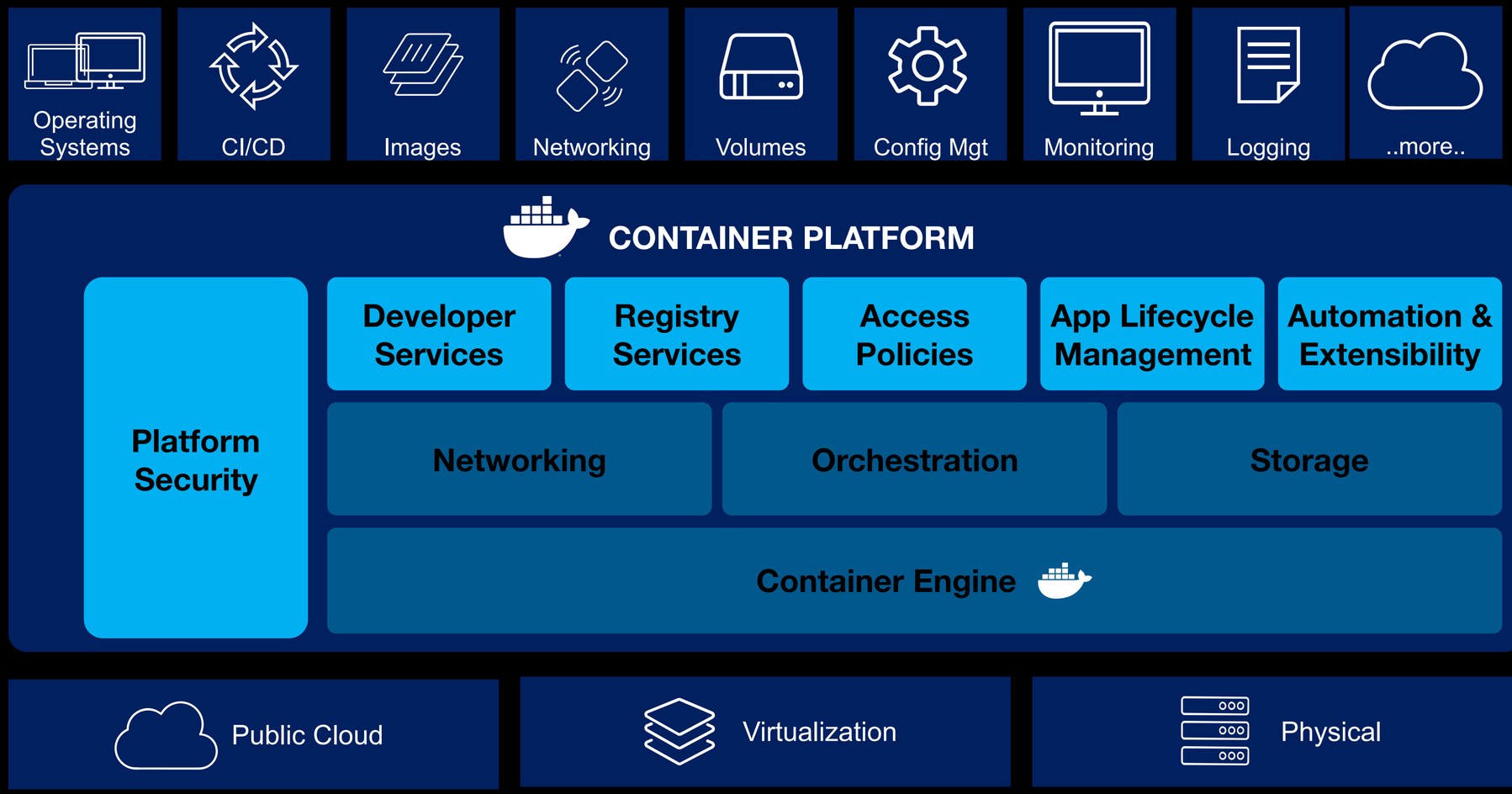
State

Deployment Rates

Capacity

Scheduling Events

Applications



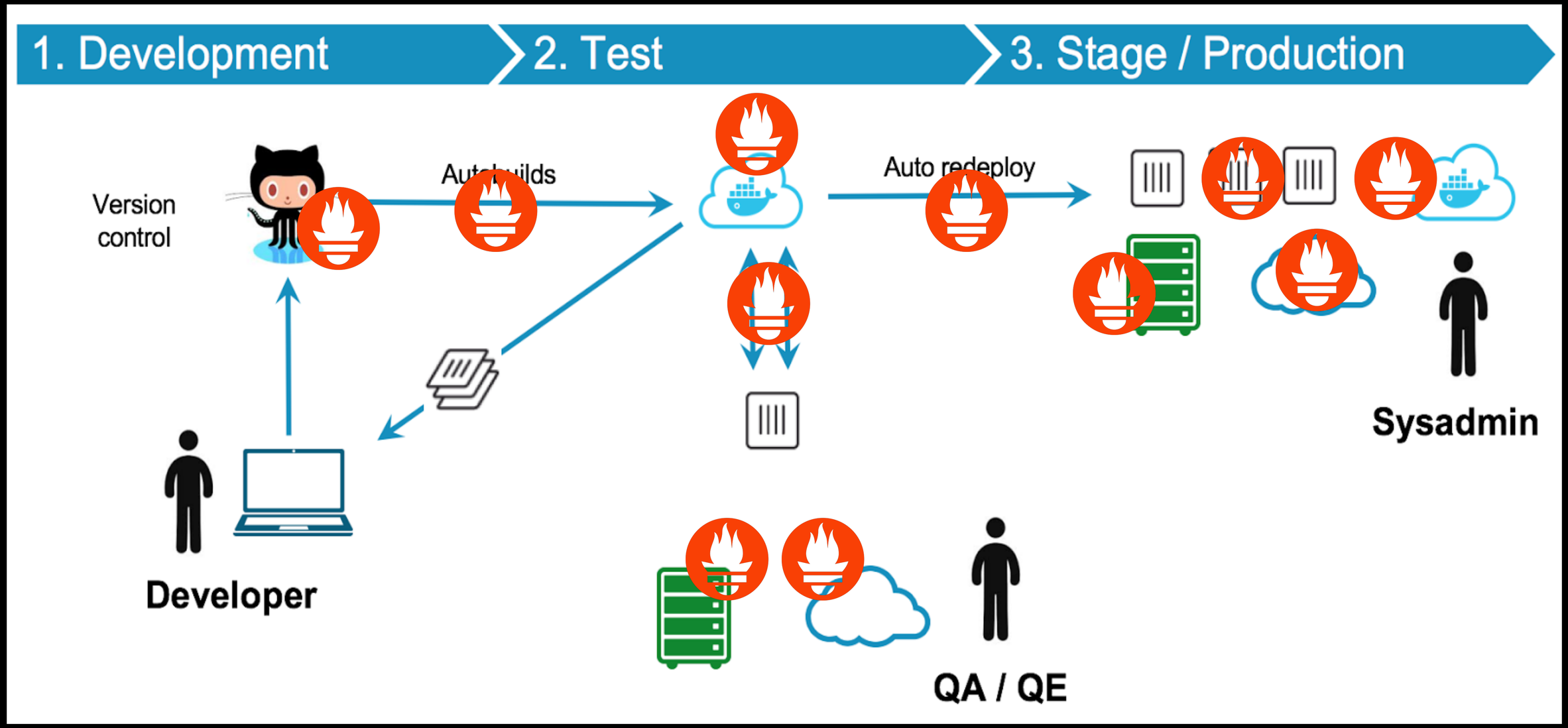
CPU

Memory

Liveness

File Descriptors

Storage Capacity


1. Development
2. Test
3. Stage / Production
Version control
Auto builds
Auto redeploy
Developer
Sysadmin
QA / QE

Website Down?

- Total Downtime: Just under 4 minutes
- 502 error messages total: 12 000
- People affected by the 502 error who did not get their bargain: 400

3 Easy Steps - Code

https://about.gitlab.com

- [X] Introduction

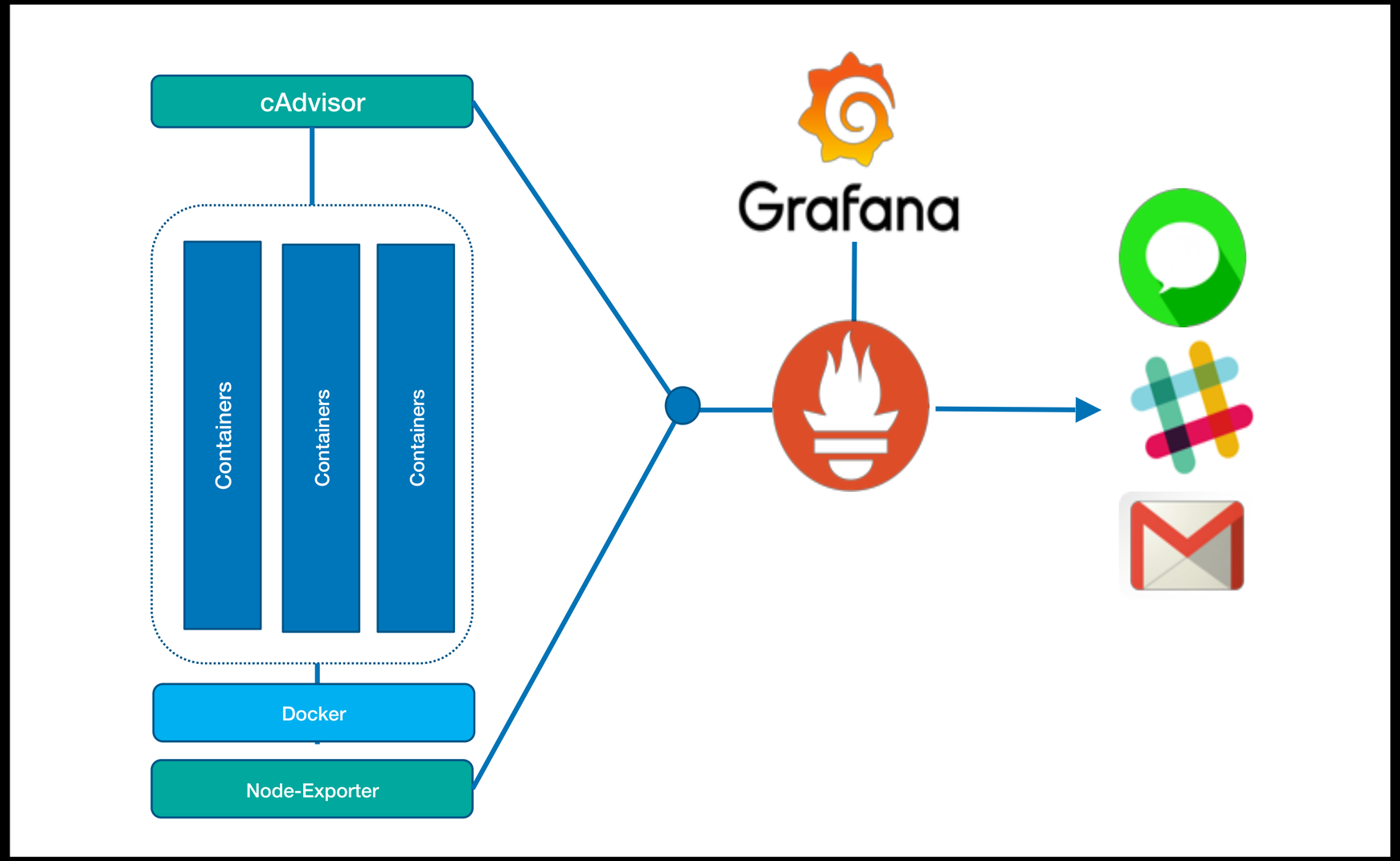
- [X] Operations Overview

- [X] What to Monitor

- [ ] Demo

- [ ] Best Practices & Recap

DEMO


cAdvisor
Containers
Containers
Containers
Docker
Node-Exporter
Grafana

- [X] Introduction

- [X] Operations Overview

- [X] What to Monitor

- [X] Demo

- [ ] Best Practices & Recap

Improve Incrementally

Best Practices

- Start small & increment
- Don’t Overlert yourself
- Set Resource Limits
- Aim for actionable Information
- Run separate from Workload
- Test for Failures
- Know your Failure Models

Kelsey Hightower

@kelseyhightower

Following

The thing you're trying to build should be more exciting than the tools used to build it.

4:56 PM - 11 Mar 2019

499 Retweets **1,841** Likes

36 499 1.8K

Resources

- 56K.Cloud - https://56K.Cloud
- Prometheus - https://github.com/vegasbrianc/prometheus
- Monitoring Labs – github.com/56kcloud/Training/
- Docker Resource Link - https://awesome-docker.netlify.com
- GitLab Dashboards - https://monitor.gitlab.net

Thank You

Brian Christner
brian@56K.cloud
@idomyowntricks