CNCF Overview

Dan Kohn, Executive Director, @dankohn1

This presentation is available at:
https://github.com/cncf/presentations

CLOUD NATIVE
COMPUTING FOUNDATION

Cloud Native Computing Foundation

- Non-profit, part of the Linux Foundation; founded Dec 2015

Graduated

Orchestration

Incubating

Monitoring

Distributed Tracing API

Logging

Remote Procedure Call

Container Runtime

Container Runtime

Networking API

Service Mesh

Distributed Tracing

Software Update Spec

Security

Storage

Service Discovery

Messaging

Service Mesh

Sandbox

Storage

Identity Spec

Identity

Policy

Serverless

Tooling

- Platinum members:

Today the Linux Foundation is much more than Linux

Security

We are helping global privacy and security through a program to encrypt the entire internet.

Networking

We are creating ecosystems around networking to improve agility in the evolving software-defined datacenter.

Cloud

We are creating a portability layer for the cloud, driving de facto standards and developing the orchestration layer for all clouds.

Automotive

We are creating the platform for infotainment in the auto industry that can be expanded into instrument clusters and telematics systems.

Blockchain

We are creating a permanent, secure distributed ledger that makes it easier to create cost-efficient, decentralized business networks.

Web

We are providing the application development framework for next generation web, mobile, serverless, and IoT applications.

Let's Encrypt

LF NETWORKING

CLOUD NATIVE COMPUTING FOUNDATION

AUTOMOTIVE GRADE LINUX

HYPERLEDGER

node JS

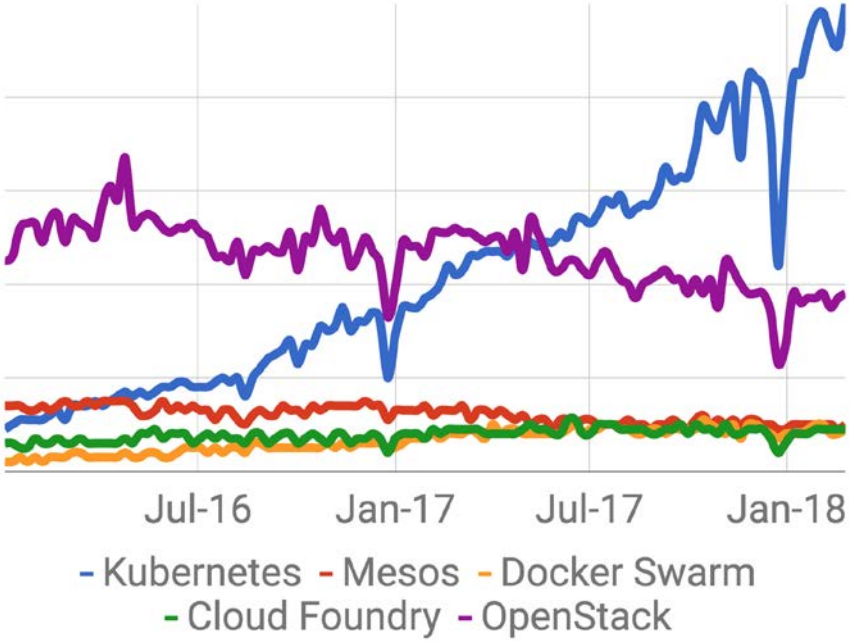
We are regularly adding projects; for the most up-to-date listing of all projects visit tlfprojects.org

52 Companies in the End User Community

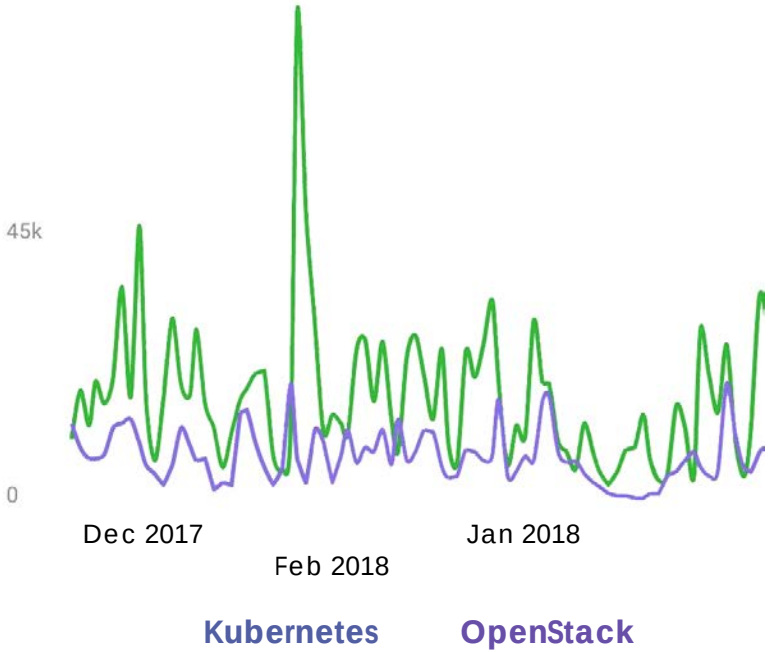
Plus 4 non-public members

Kubernetes in Search Trends

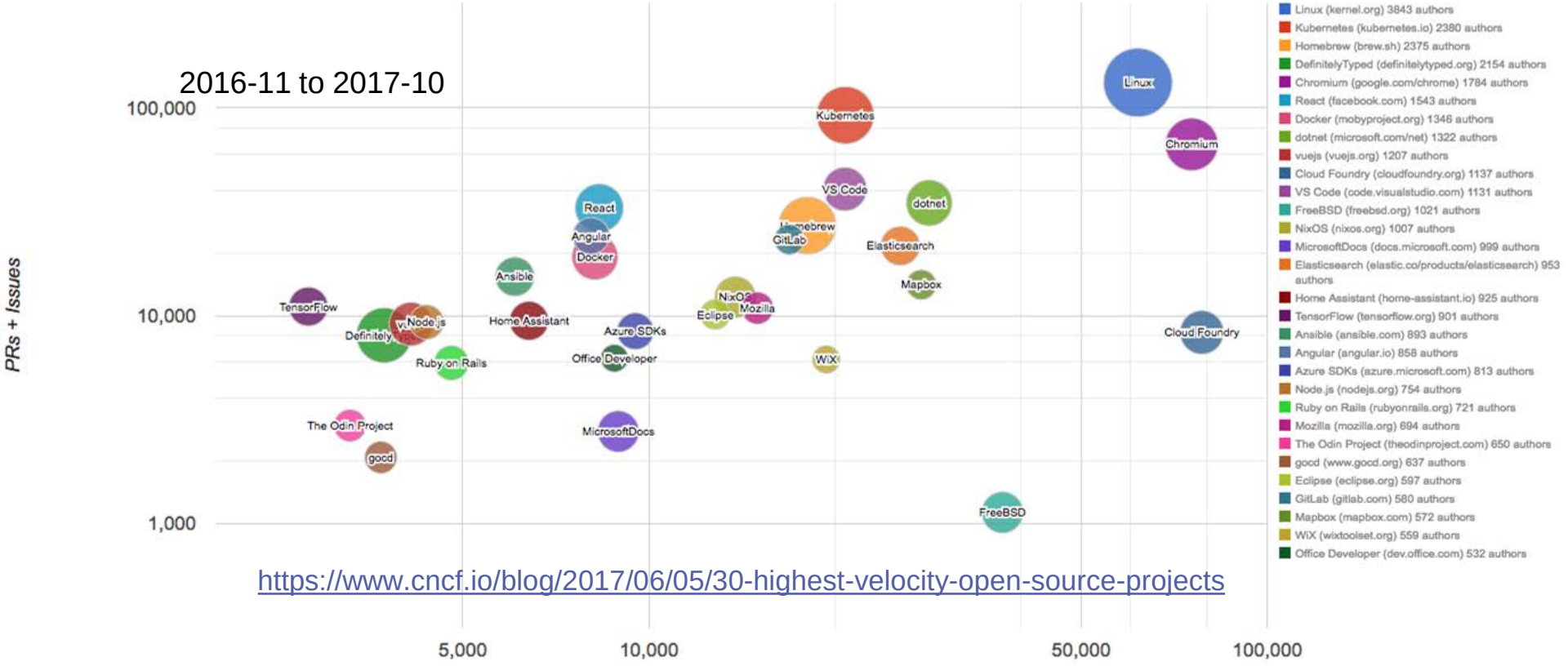
Google Trends



WeChat

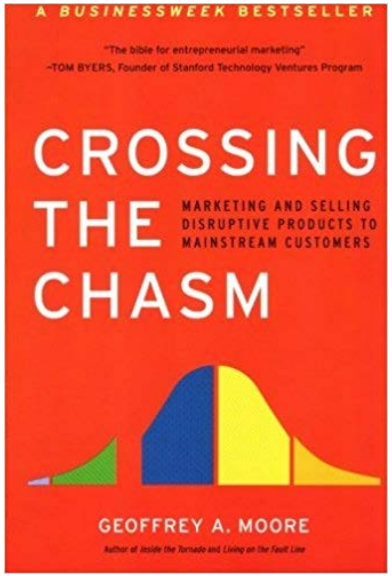


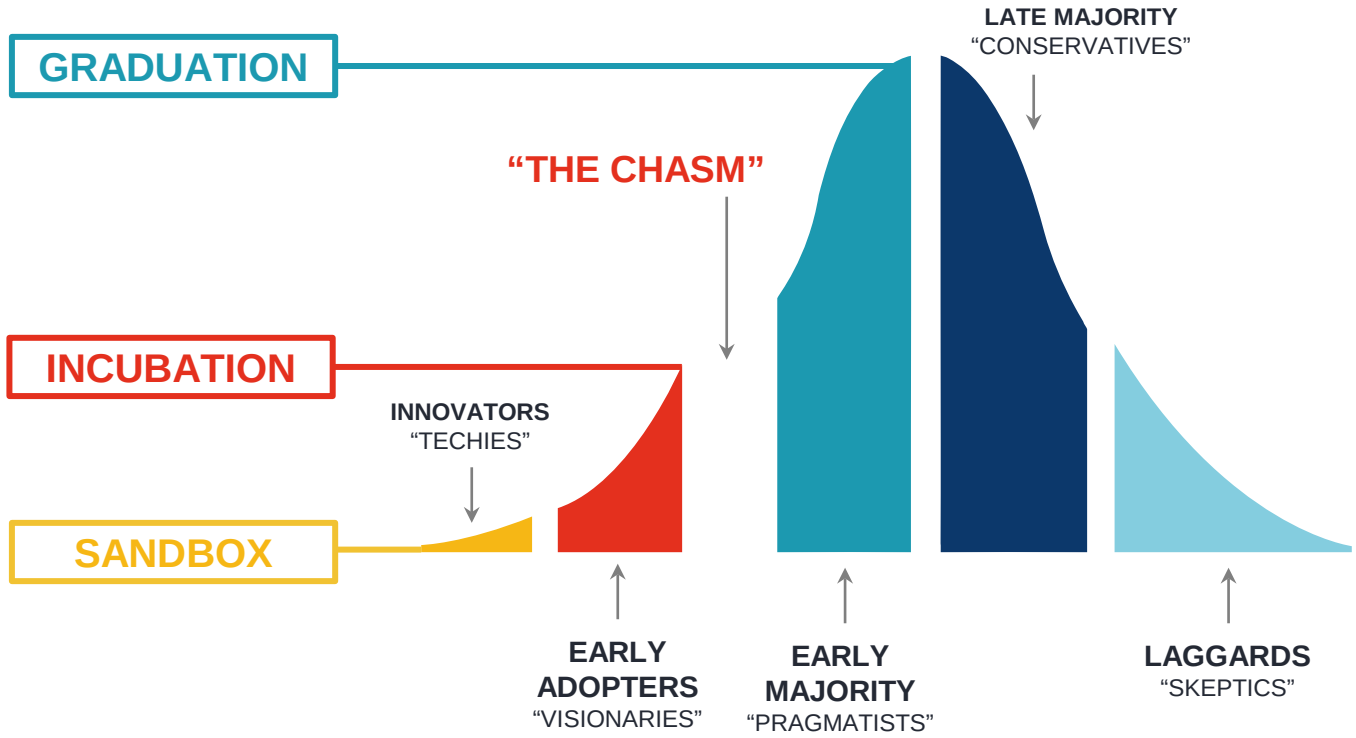
30 Highest Velocity Open Source Projects



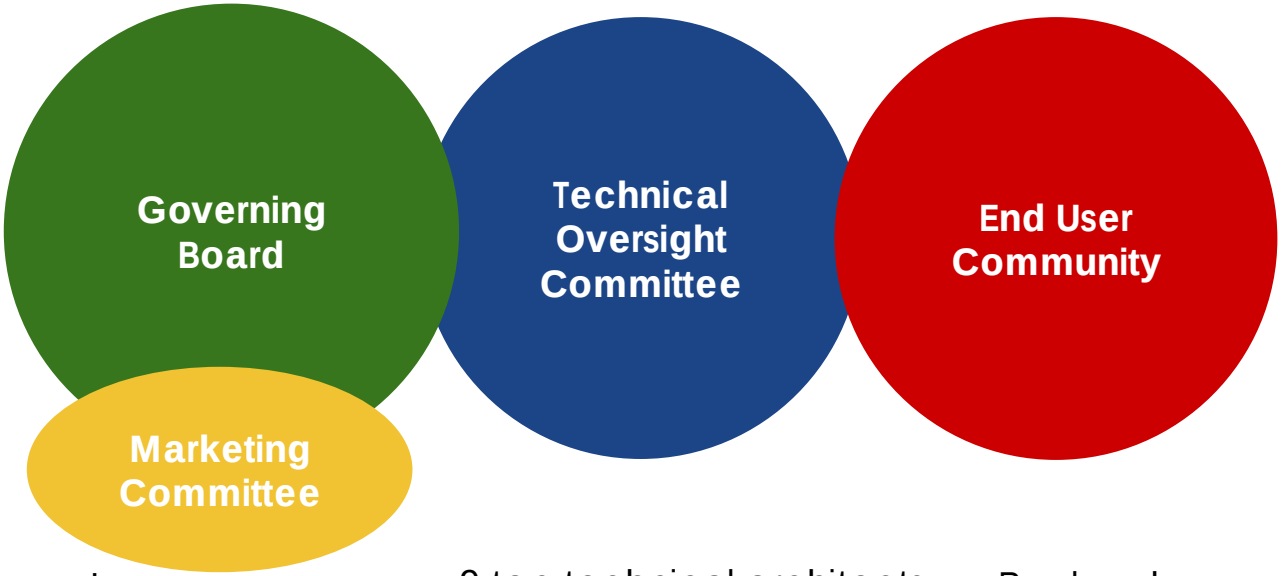
https://www.cncf.io/blog/2017/06/05/30-highest-velocity-open-source-projects

CNCF Project Maturities





CNCF Structure



- Mainly vendors
- Fund the organization
- Marketing and strategic direction

- 9 top technical architects
- Admit new projects
- Acts as a resource to projects

- Real end users of these technologies
- Communicate back requirements
- And good and bad experiences

220+ Members and Growing

Platinum Members

Gold Members

End User Members

End User Supporters

Academic/Nonprofit

220+ Members and Growing (Silver 1)

220+ Members and Growing (Silver 2)

Migrating Legacy Monoliths to Cloud Native Microservices Architectures on Kubernetes

Dan Kohn, Executive Director

This presentation is available at:

https://github.com/cncf/presentations

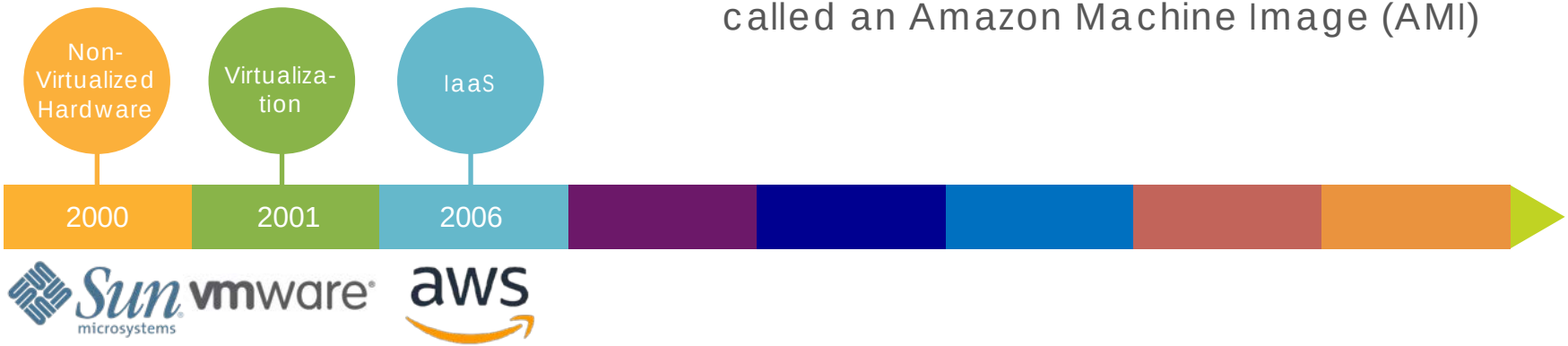
Non-Virtualized Servers: Sun (2000)

- Launching a new application? Buy a new server; or a rack of them!
- Building block of your application is physical servers

Virtualization: VMWare (2001)

- Releases for server market in 2001
- Popularizes virtual machines (VMs)
- Run many VMs on one physical machine, meaning you can buy fewer servers!
- Architectural building block becomes a VM

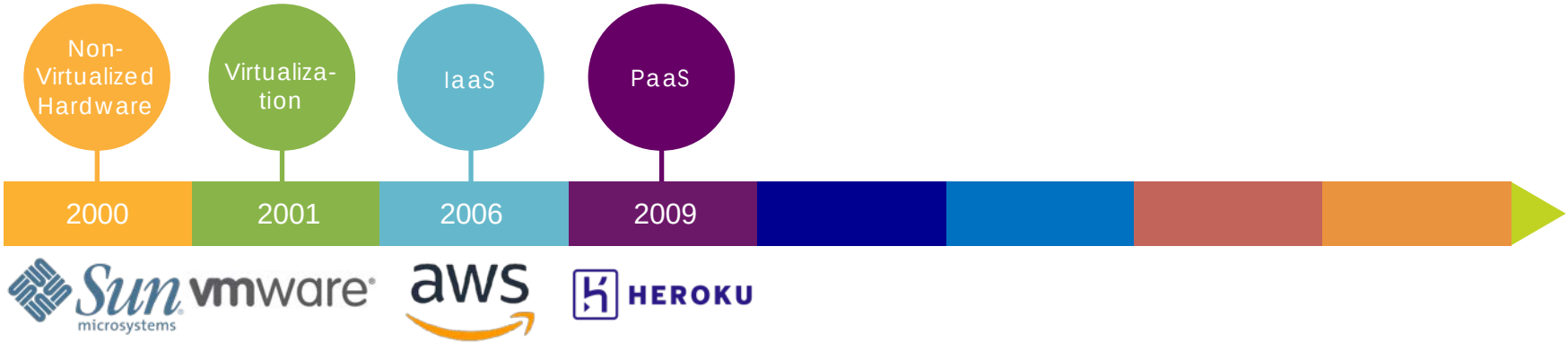
IaaS: AWS (2006)



- Amazon Web Services (AWS) creates the Infrastructure-as-a-Service market by launching Elastic Compute Cloud (EC2) in 2006
- Rent servers by the hour
- Convert CapEx to OpEx
- Architectural building block is also a VM, called an Amazon Machine Image (AMI)

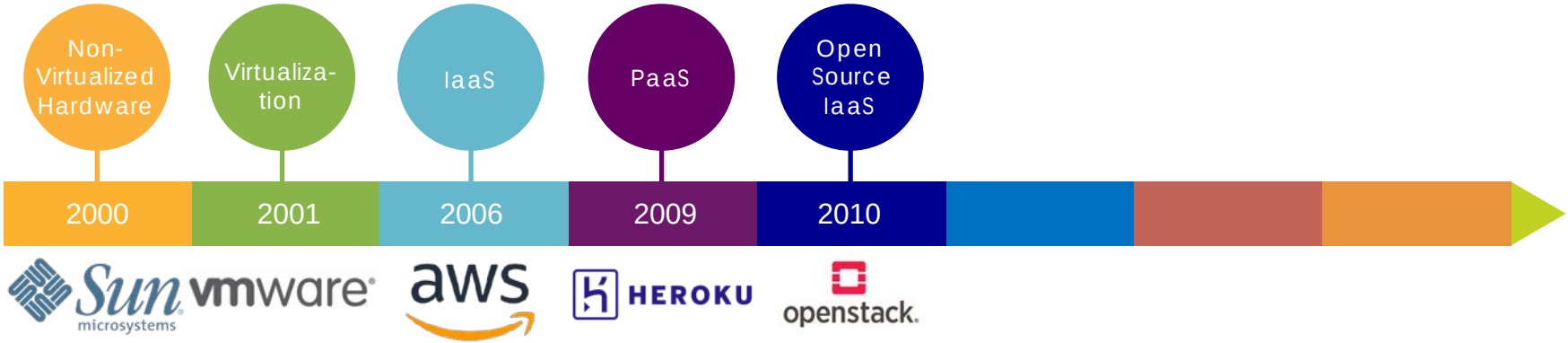
PaaS: Heroku (2009)

- Heroku popularizes Platform-as-a-Service (PaaS) with their launch in 2009
- Building block is a buildpack, which enables containerized 12-factor applications
  - The process for building the container is opaque, but:
  - Deploying new version of an app is just: git push heroku



Open Source IaaS: OpenStack (2010)

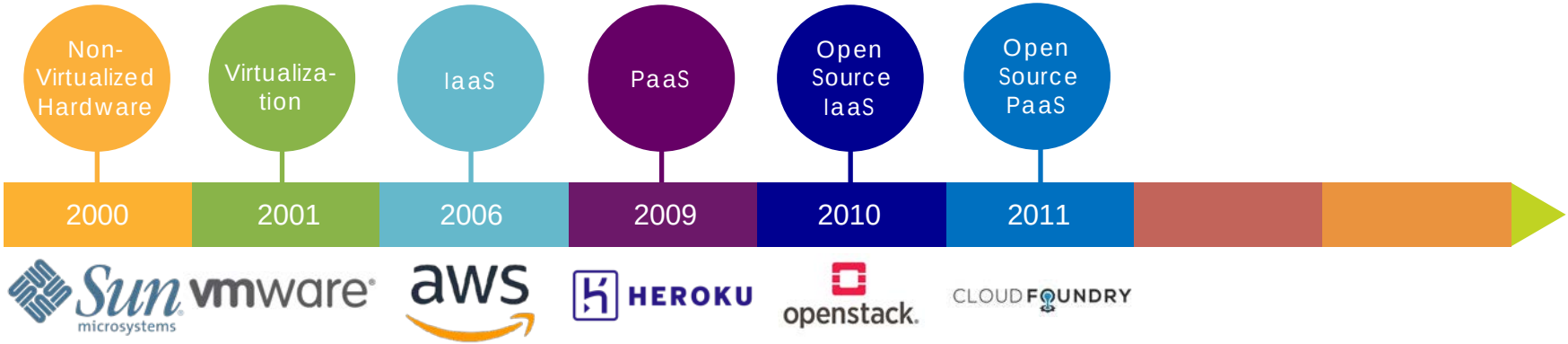
- OpenStack brings together an extraordinarily diverse group of vendors to create an open source Infrastructure-as-a-Service (IaaS)
- Competes with AWS and VMWare
- Building block remains a VM



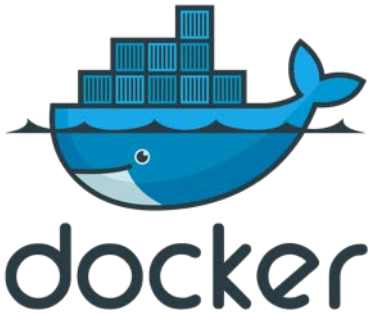
Open Source PaaS: Cloud Foundry (2011)

CLOUD FOUNDRY

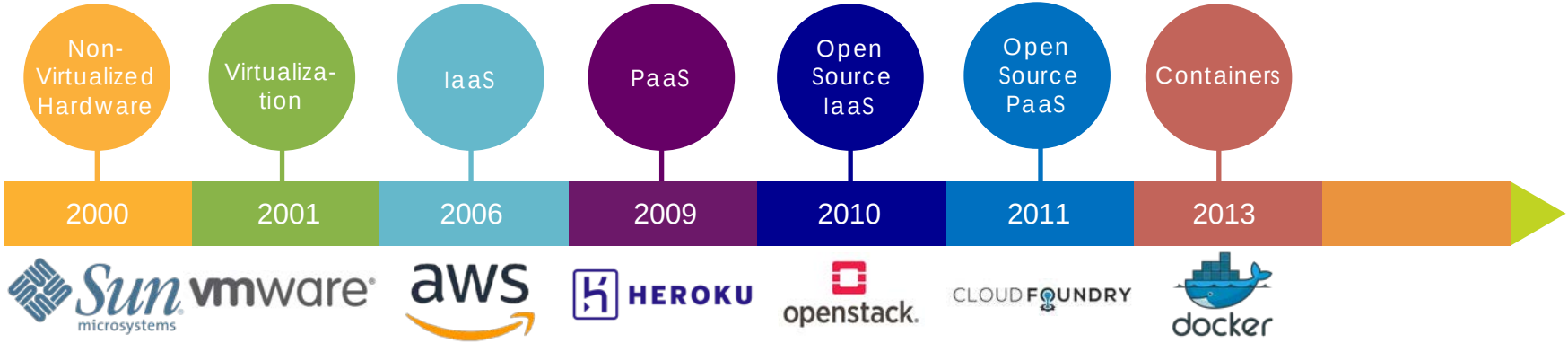
- Pivotal builds an open source alternative to Heroku's PaaS and launches the Cloud Foundry Foundation in late 2014
- Building block is Garden containers, which can hold Heroku buildpacks, Docker containers and even non-Linux OSes



Containers: Docker (2013)



- Docker combines LXC, Union File System and cgroups to create a containerization standard adopted by millions of developers around the world
- Fastest uptake of a developer technology ever
- Enables isolation, reuse and immutability

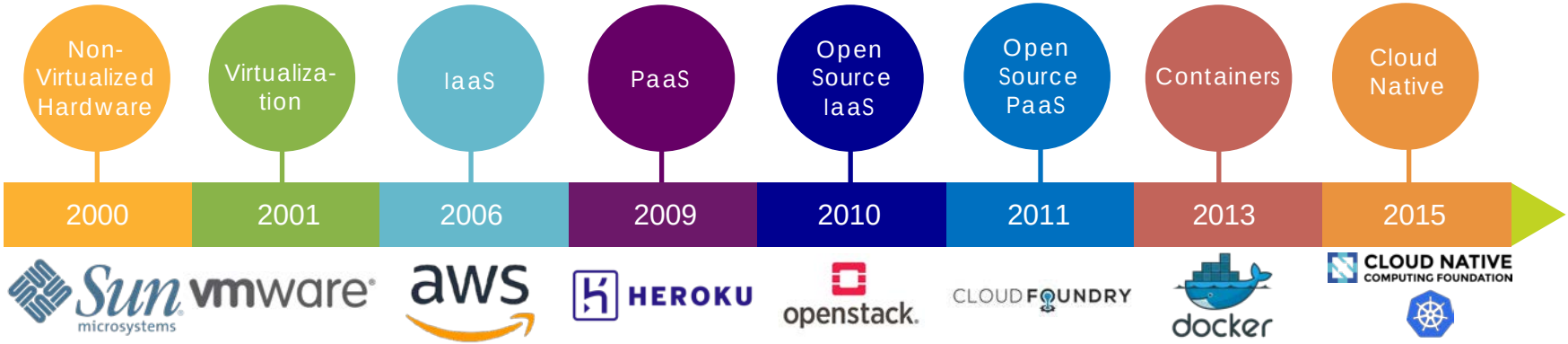


Cloud Native: CNCF (2015)

CLOUD NATIVE COMPUTING FOUNDATION

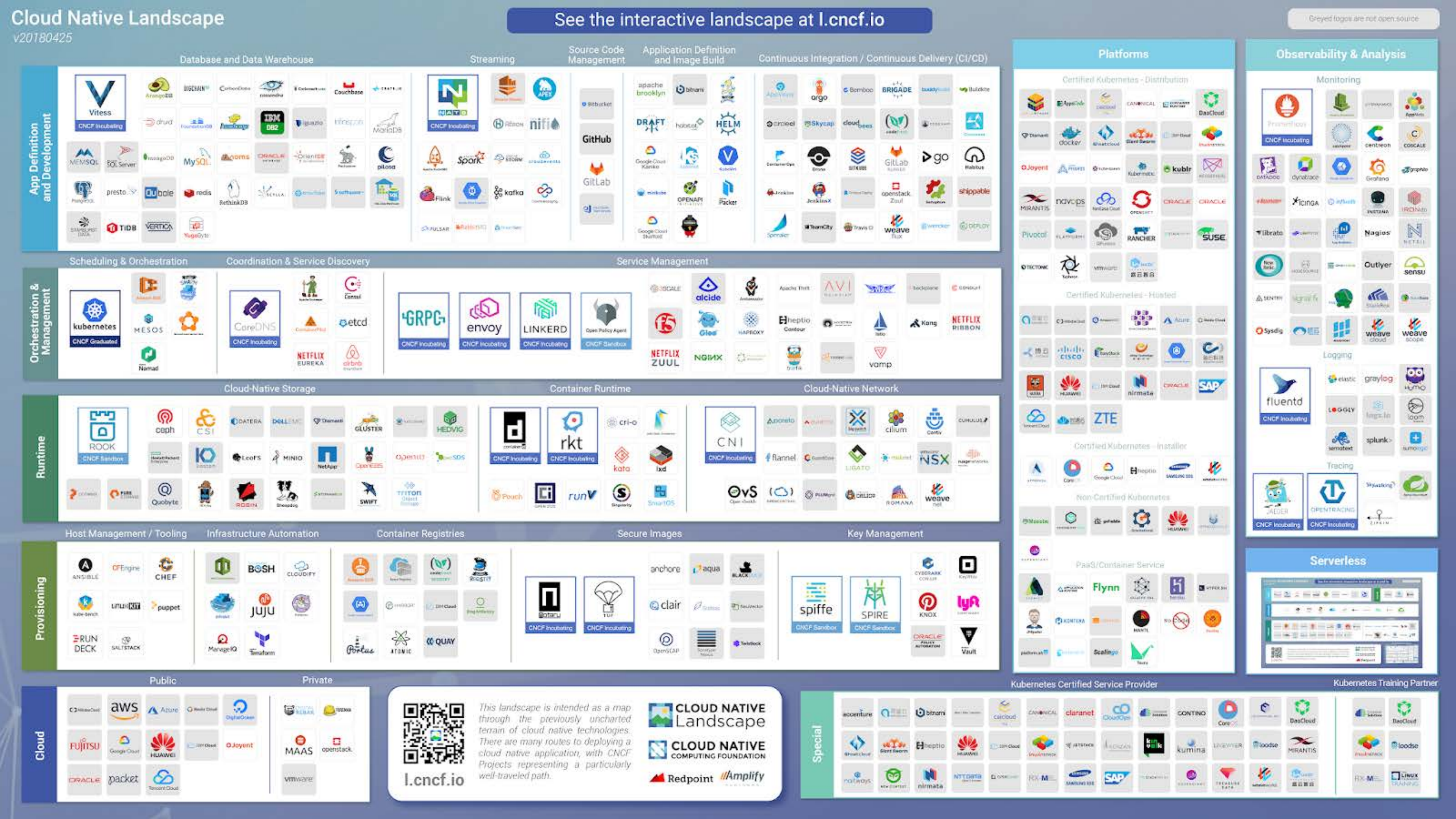
kubernetes

- Cloud native computing uses an open source software stack to:
  - segment applications into *microservices*,
  - packaging each part into its own *container*
  - and dynamically *orchestrating* those containers to optimize resource utilization



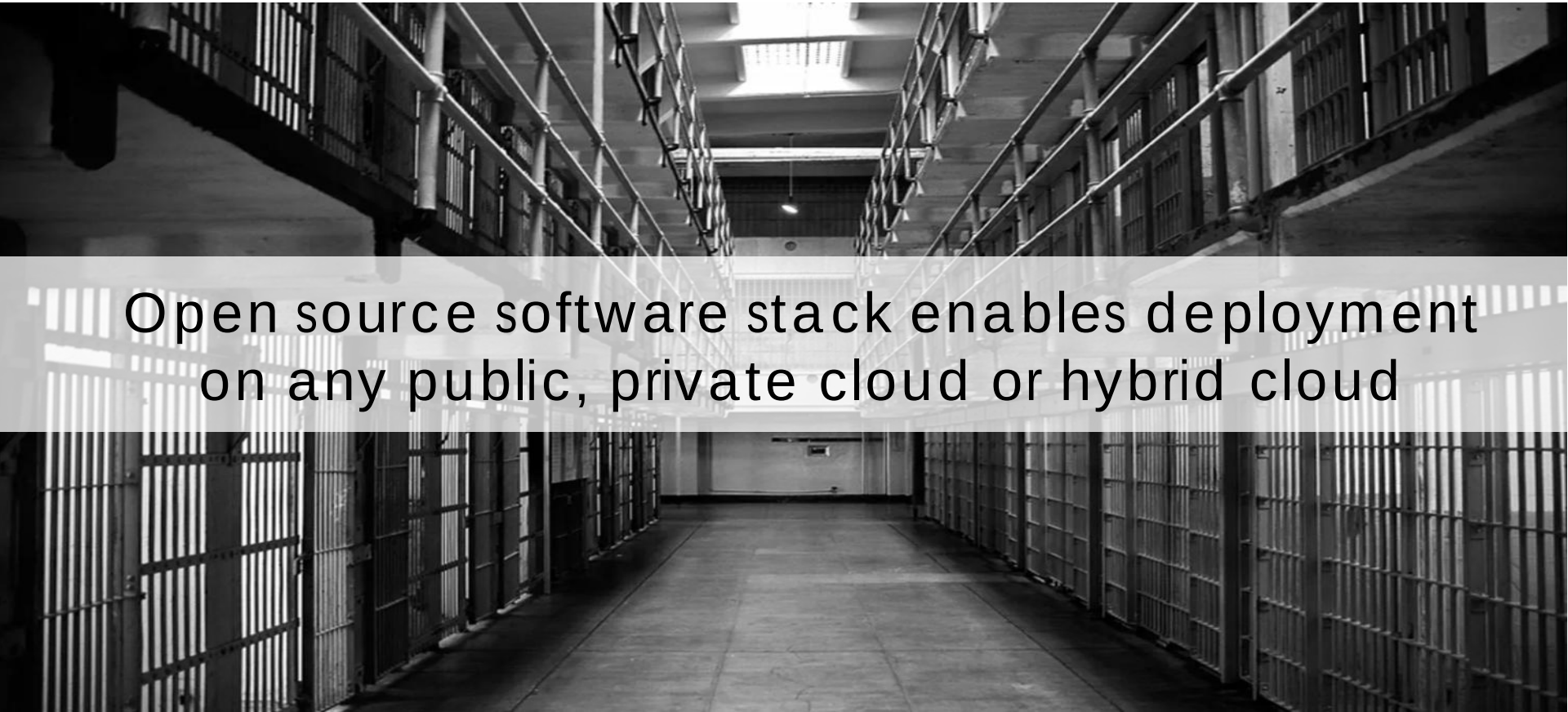
What Have We Learned?

- Core Building Block:
  - Servers ➡ Virtual Machines ➡ Buildpacks ➡ Containers
- Isolation Units
  - From heavier to lighter weight, in spin-up time and size
- Immutability
  - From pets to cattle
- Provider
  - From closed source, single vendor to open source, cross-vendor



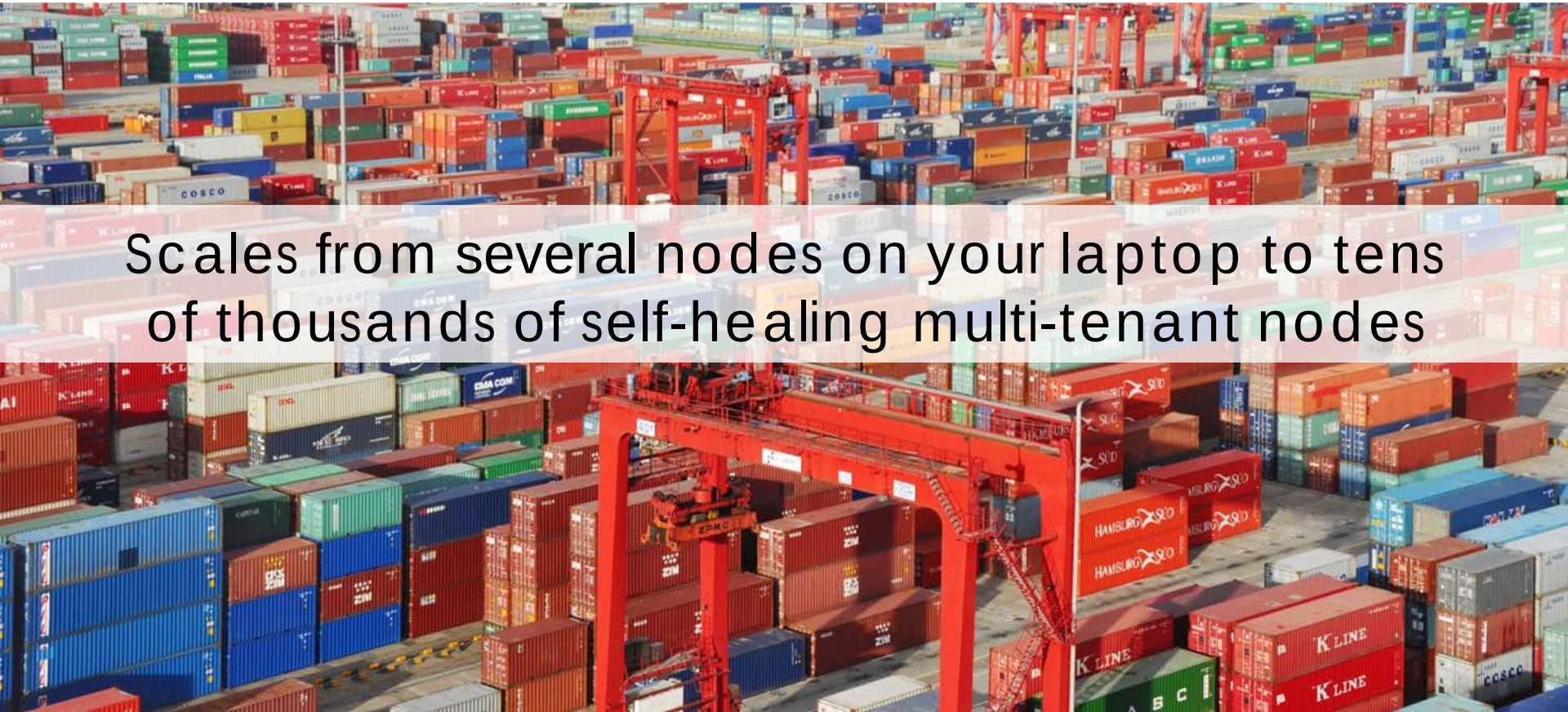
Cloud Native Landscape
v20180425
See the interactive landscape at l.cncf.io
Greyed logos are not open source
App Definition and Development
Database and Data Warehouse
Streaming
Source Code Management
Application Definition and Image Build
Continuous Integration / Continuous Delivery (CI/CD)
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Service Management
Runtime
Cloud-Native Storage
Container Runtime
Cloud-Native Network
Provisioning
Host Management / Tooling
Infrastructure Automation
Container Registries
Secure Images
Key Management
Cloud
Public
Private
Platforms
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
Non-Certified Kubernetes
PaaS/Container Service
Observability & Analysis
Monitoring
Logging
Tracing
Serverless
Kubernetes Certified Service Provider
Kubernetes Training Partner
Special
CNCF Graduated
CNCF Incubating
CNCF Sandbox
kubernetes
Vitess
NATS
CoreDNS
GRPC
envoy
LINKERD
Open Policy Agent
ROOK
containerd
rkt
CNI
Notary
TUF
spiffe
SPIRE
Prometheus
fluentd
JAEGER
OPENTRACING
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
l.cncf.io
CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify

Avoid Vendor Lock-in



Open source software stack enables deployment on any public, private cloud or hybrid cloud

Enable Unlimited Scalability



Scales from several nodes on your laptop to tens of thousands of self-healing multi-tenant nodes

Increase Agility and Maintainability

By splitting applications into microservices with explicitly described dependencies

Improve Efficiency and Resource Utilization

Via a central orchestrating process that dynamically manages and schedules microservices

Achieve Resiliency

To failures of individual containers, machines, and even data centers and to varying levels of demand

Cloud Native Enables High Performance

High-performing teams deploy more frequently and have much faster lead times.

They make changes with fewer failures, and recover faster from failures.

200x more frequent deployments

2,555x shorter lead times

3x lower change failure rate

24x faster recovery from failures

2016 State of DevOps Report Infographic from Puppet

Greenfield Applications

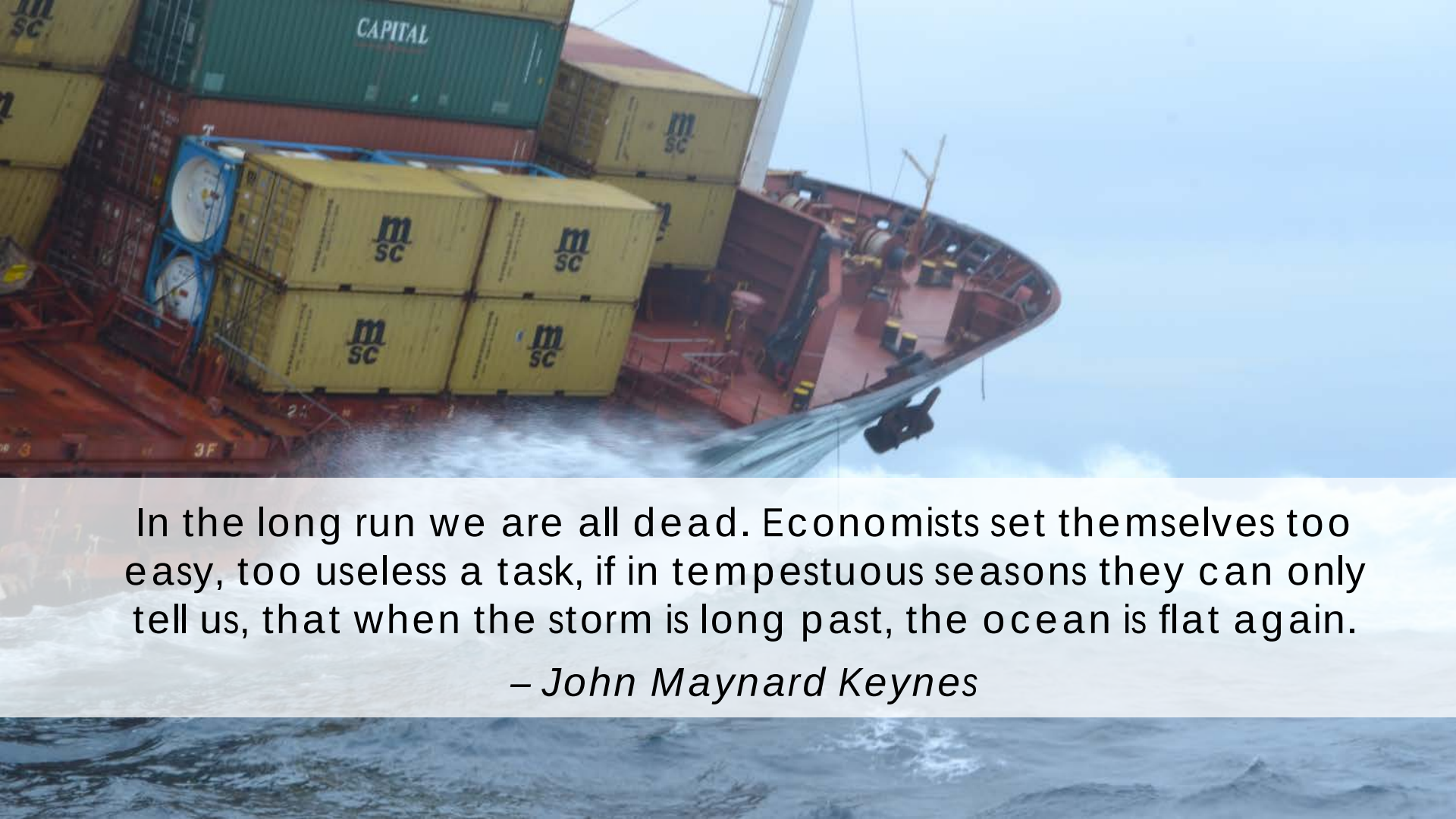
Cloud Native application architectures are the default choice for greenfield applications

Greenfield Application Design

- The leading choice for cloud native orchestration is:

kubernetes

  - Selected by top enterprises and startups like Bloomberg, Box, Reddit and the New York Times
  - One of the highest development velocity projects in the history of open source
  - Amazing group of technology giants and startups cooperating to rapidly improve

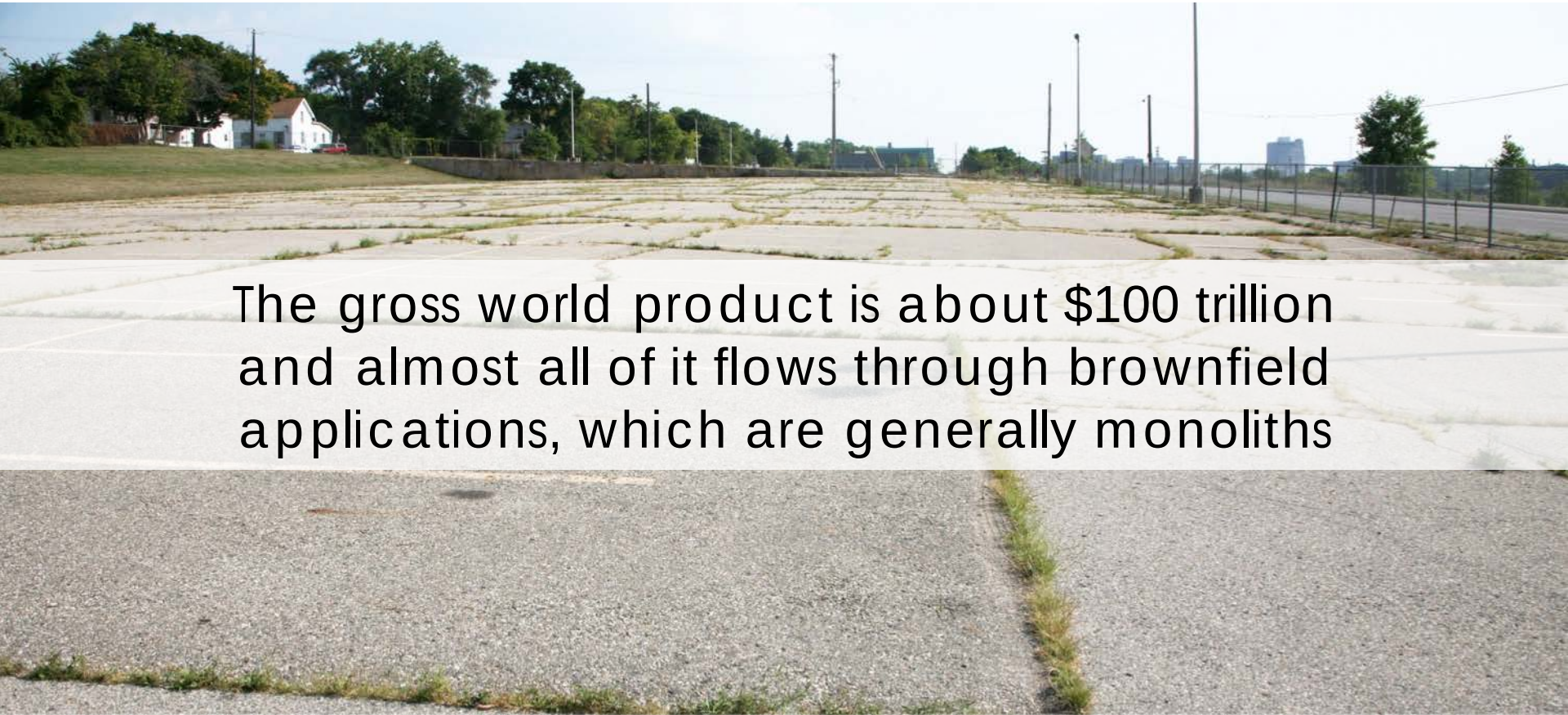


In the long run we are all dead. Economists set themselves too easy, too useless a task, if in tempestuous seasons they can only tell us, that when the storm is long past, the ocean is flat again.

– John Maynard Keynes

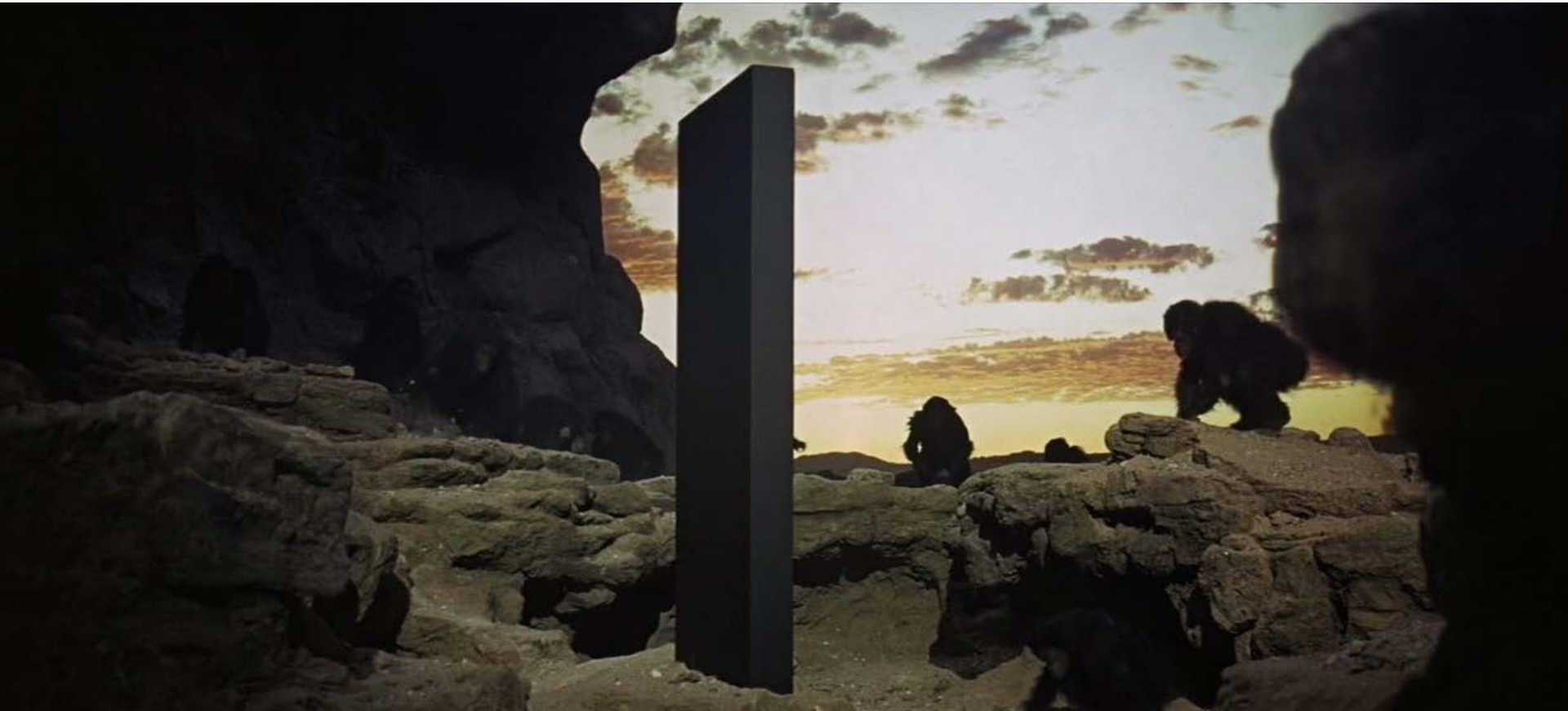
...too easy, too useless a task...

The Real World Consists of Brownfield Applications



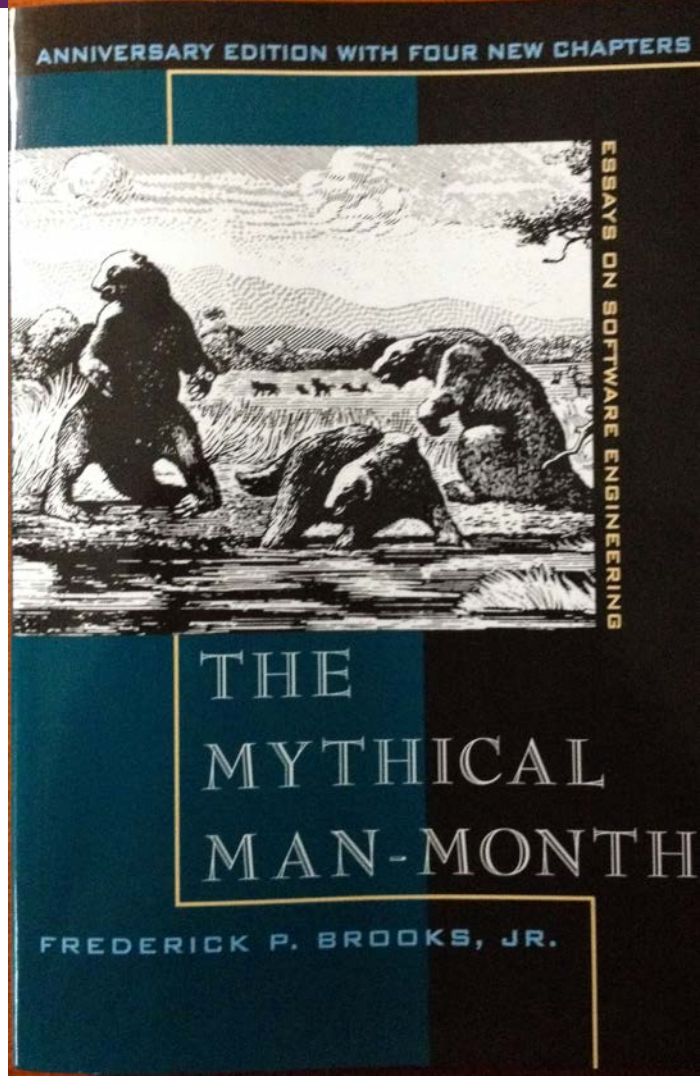
The gross world product is about $100 trillion and almost all of it flows through brownfield applications, which are generally monoliths

Nearly All Production Applications Are Monoliths

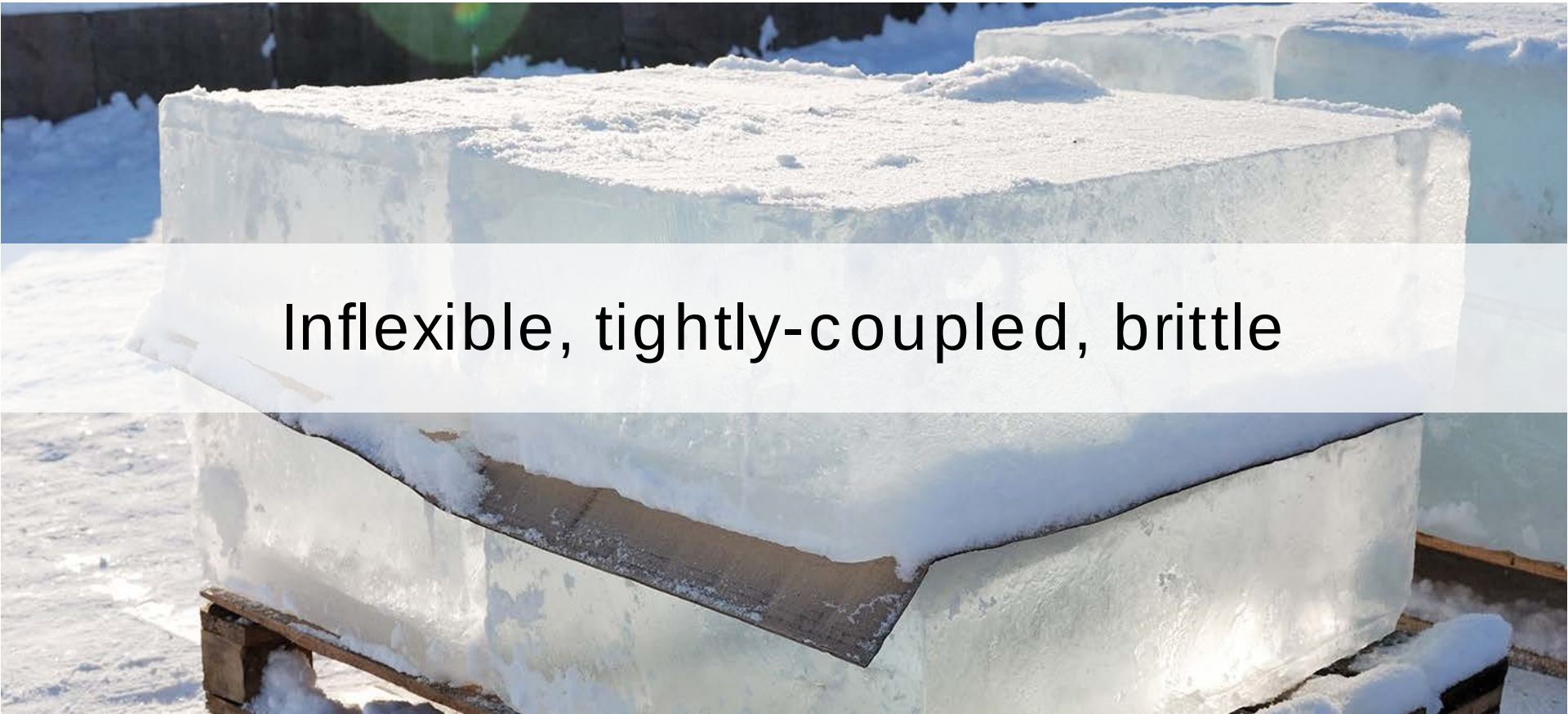


How About a Rewrite?

- The Second System Syndrome says that many rewrites end in failure
- The existing system evolves faster than the second one can catch up



Monoliths Are the Antithesis of Cloud Native

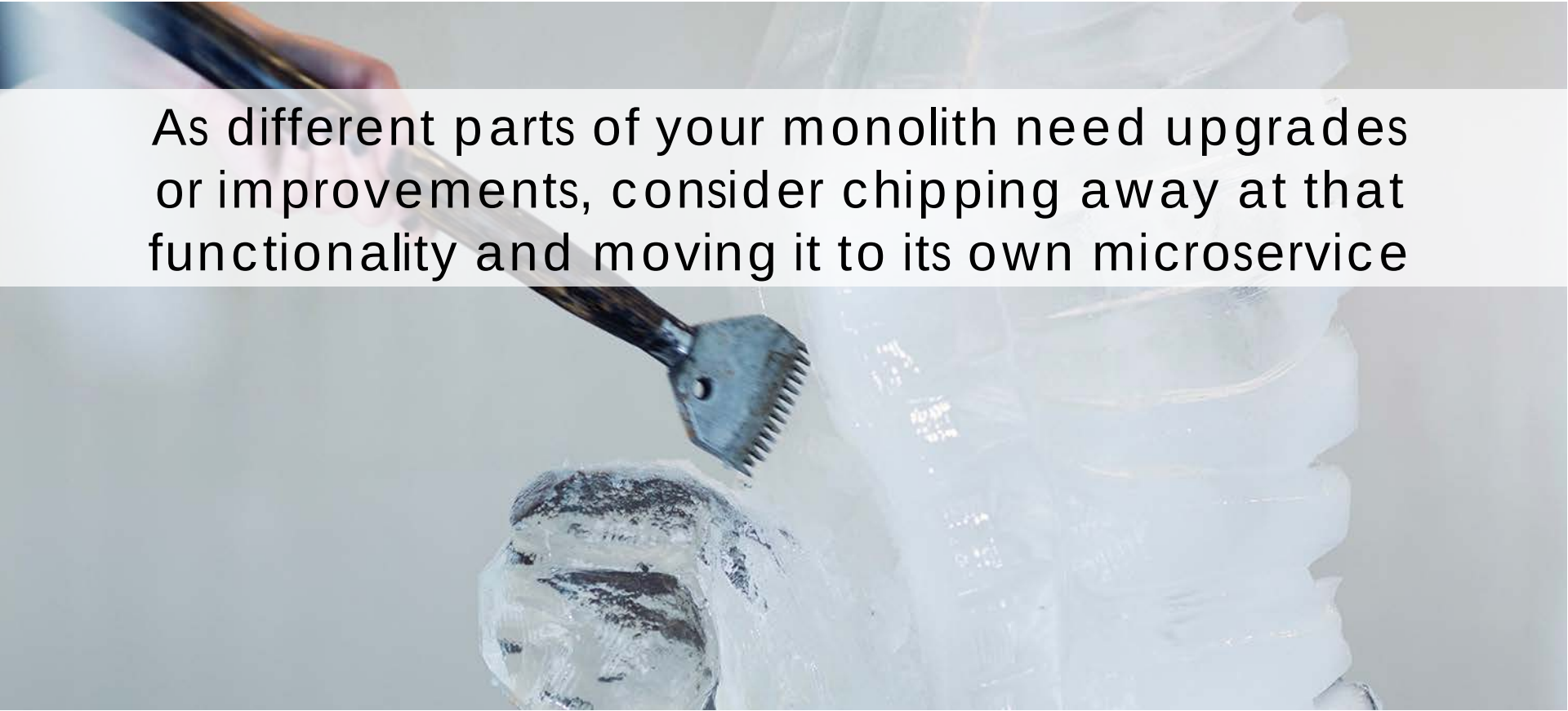


Step One Is to Stop Digging

Step Two is Lift-and-Shift Your Monolith

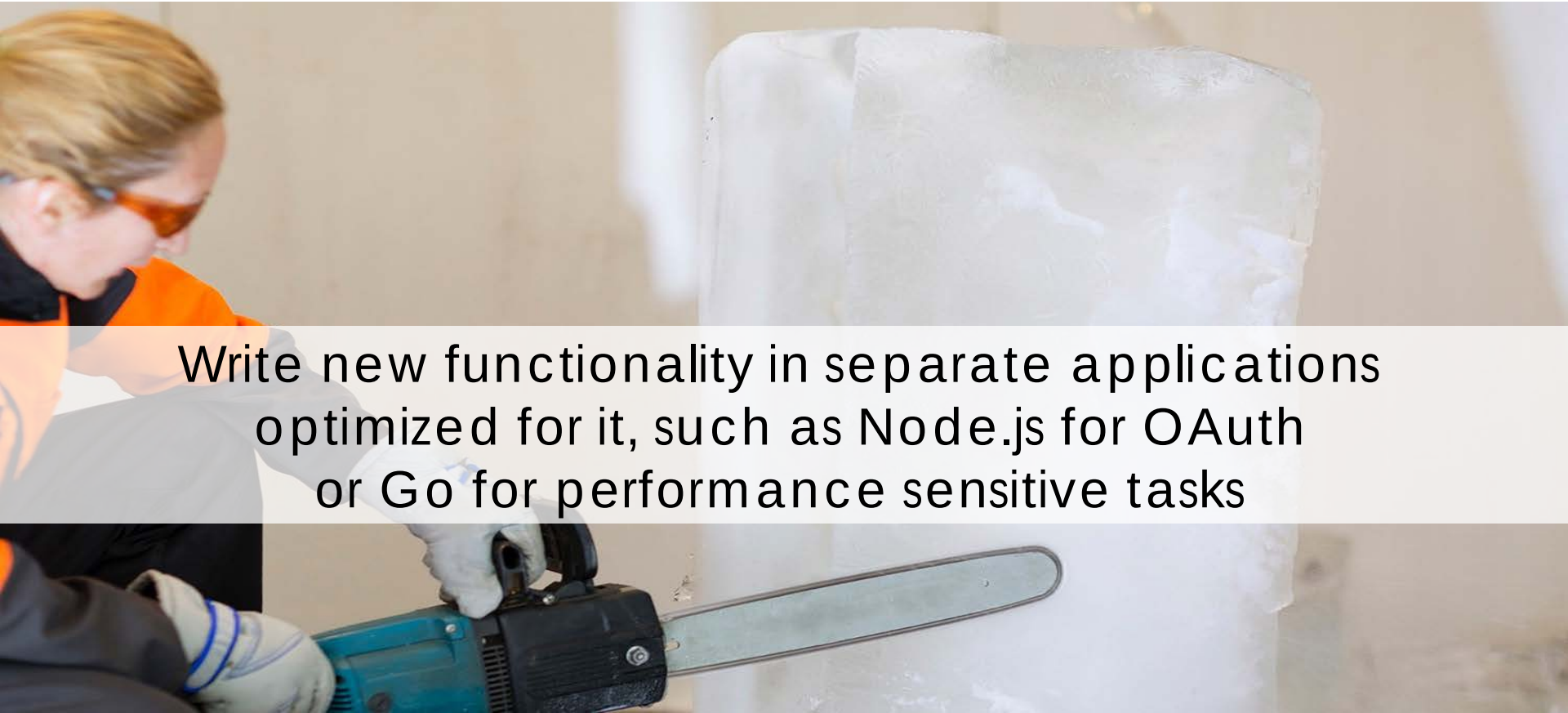
StatefulSets are helpful for providing stable, persistent storage and unique network identifiers to your monolith

Chip Away at the Monolith



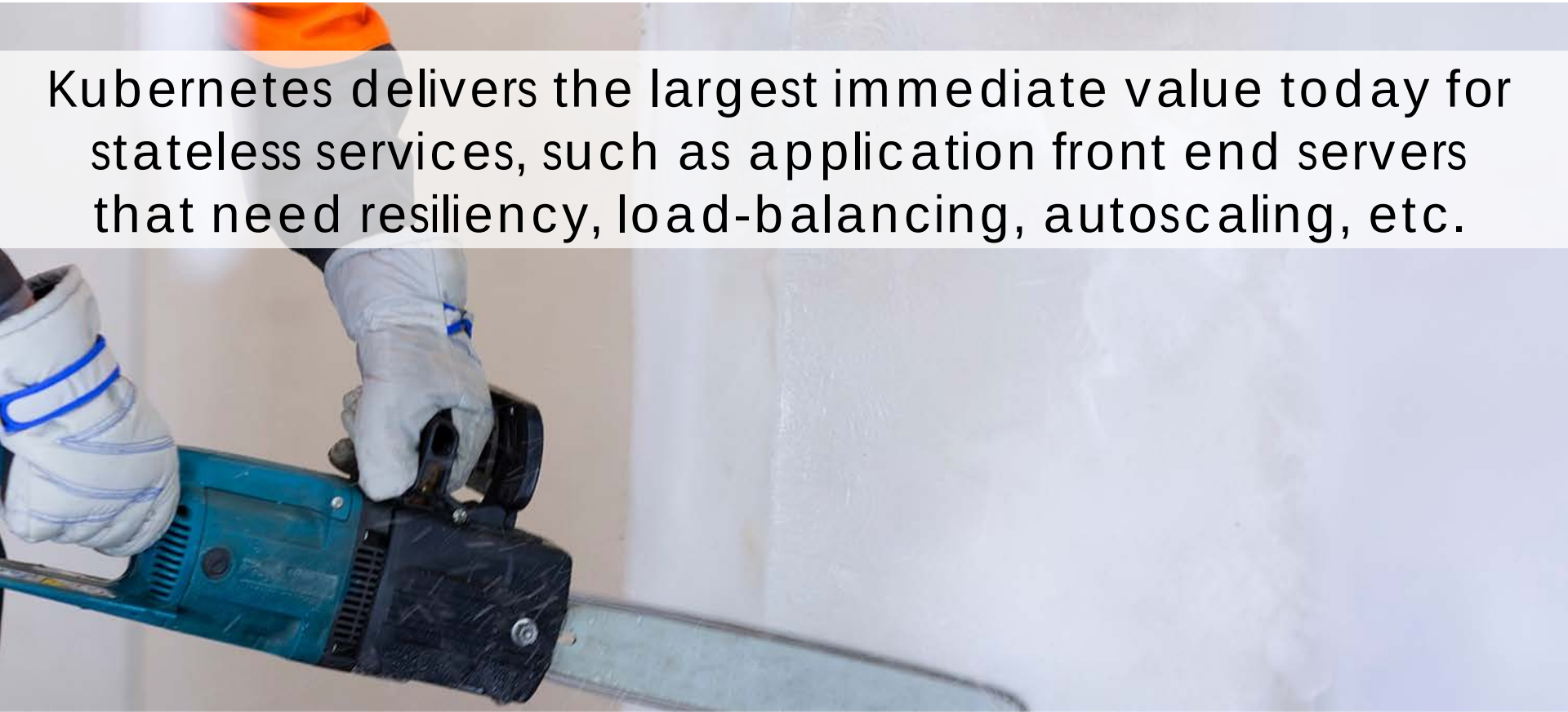
As different parts of your monolith need upgrades or improvements, consider chipping away at that functionality and moving it to its own microservice

New Functionality Can Go in New Apps



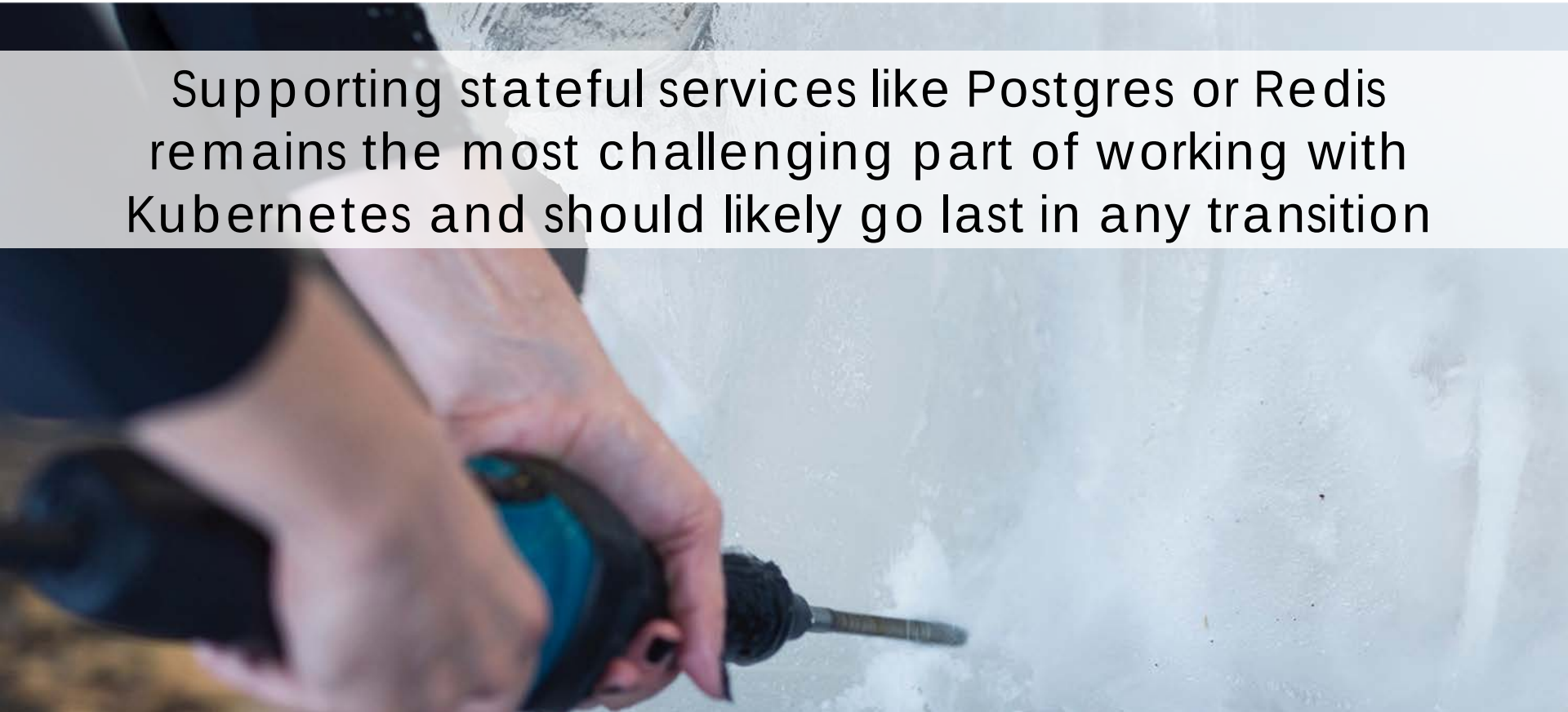
Write new functionality in separate applications optimized for it, such as Node.js for OAuth or Go for performance sensitive tasks

Start with Stateless Services



Kubernetes delivers the largest immediate value today for stateless services, such as application front end servers that need resiliency, load-balancing, autoscaling, etc.

Transition Your Data Stores Last



Supporting stateful services like Postgres or Redis remains the most challenging part of working with Kubernetes and should likely go last in any transition

Consider Complementary Projects

Consider the constellation of CNCF projects
(and their competitors) to provide related services

Prometheus OpenTracing Fluentd Linkerd gRPC CoreDNS containerd rkt CNI Envoy Jaeger TUF Notary

Evolving Your Monolith into a Microservice

And Eventually You Have a Collection

Kubernetes and Greenfield

Avoid the “soft bigotry of low expectations” of thinking that you need to do a greenfield rewrite to get the benefits of cloud native

Kubernetes and Brownfield

kubernetes ❤ brownfield applications

More Info on Evolving Monoliths

- Box
  - https://architecht.io/box-co-founder-on-moving-to-microservices-and-the-promise-of-kubernetes-a49f01b1c0c0
- Key Bank
  - https://developers.redhat.com/blog/2016/10/27/the-fast-moving-monolith-how-we-sped-up-delivery-from-every-three-months-to-every-week/
- TicketMaster
  - https://www.linux.com/news/learn/kubernetes/ticketmaster-chooses-kubernetes-stay-ahead-competition
- Wikipedia
  - https://kubernetes.io/case-studies/wikimedia/
- GitHub
  - https://githubengineering.com/kubernetes-at-github/

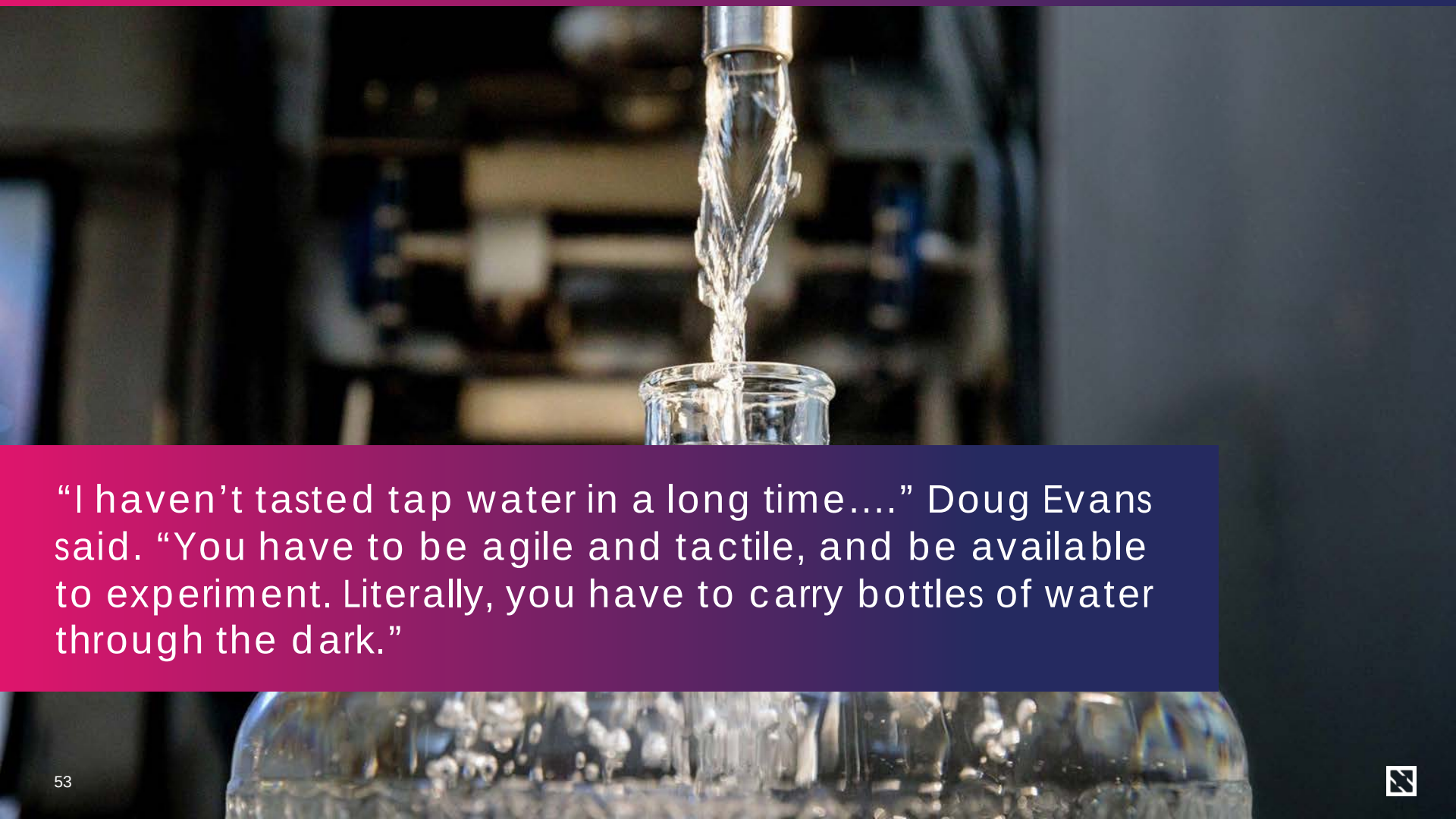
HOW GOOD IS OUR CODE?

Dan Kohn
Executive Director, CNCF


SaaS Architecture: Move Fast and Bake Things
Bootstrapped with single-donut mixer, fryer, and topper
Built with OpenTracing-aware Mutex wrapper
Web Client
Web Server
Mixer
Fryer
Topper
CLOUD NATIVE CON
KubeCon
SUPPORTING THE CLOUD NATIVE ECOSYSTEM
www.cncf.io | @CloudNativeFdn
ORCHESTRATION.
CONTAINERIZATION.
MICROSERVICES.
gRPC


CLOUD NATIVE CON Europe 2017
KubeCon
A CNCF EVENT
ORCHESTRATION.
CONTAINERIZATION.
MICROSERVICES.

https://www.youtube.com/watch?v=5lutHF5HhVA



“I haven’t tasted tap water in a long time….” Doug Evans said. “You have to be agile and tactile, and be available to experiment. Literally, you have to carry bottles of water through the dark.”


qntm
@qntm
Follow
Our new software consultancy produces what we call "raw code", guaranteed NOT to have passed through CI or any kind of onerous "testing". The result is a palpably richer and more authentic software experience
7:08 PM - 4 Jan 2018

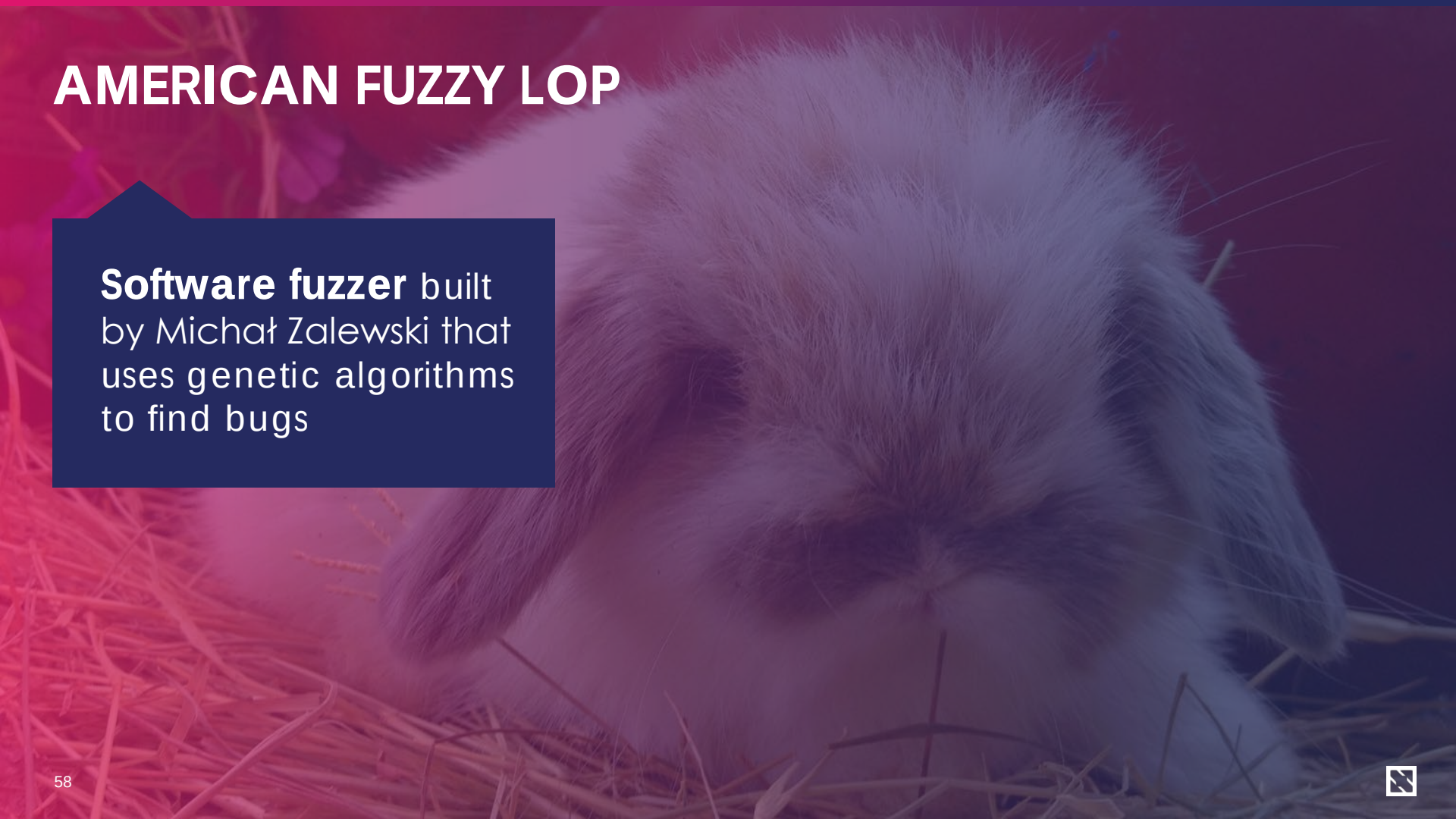
CLOUD NATIVE
COMPUTING FOUNDATION

OUR SOFTWARE IS NOT AS GOOD AS SQLITE

- Developed mainly by one highly-regarded developer, Richard Hipp
- 100% branch test coverage
- Millions of test cases
- 1,000 times as much test code as product code

AMERICAN FUZZY LOP

Software fuzzer built by Michał Zalewski that uses genetic algorithms to find bugs



SQLITE STILL HAS BUGS!

When Zalewski ran American Fuzzy Lop against SQLite, he found **22 bugs(!!!) in 30 minutes of work**

Note that SQLite quickly fixed all of the bugs and incorporated AFL into their release process

But our code is **not as good** as SQLite's!

HOW BIG IS OUR APP?

LINUX

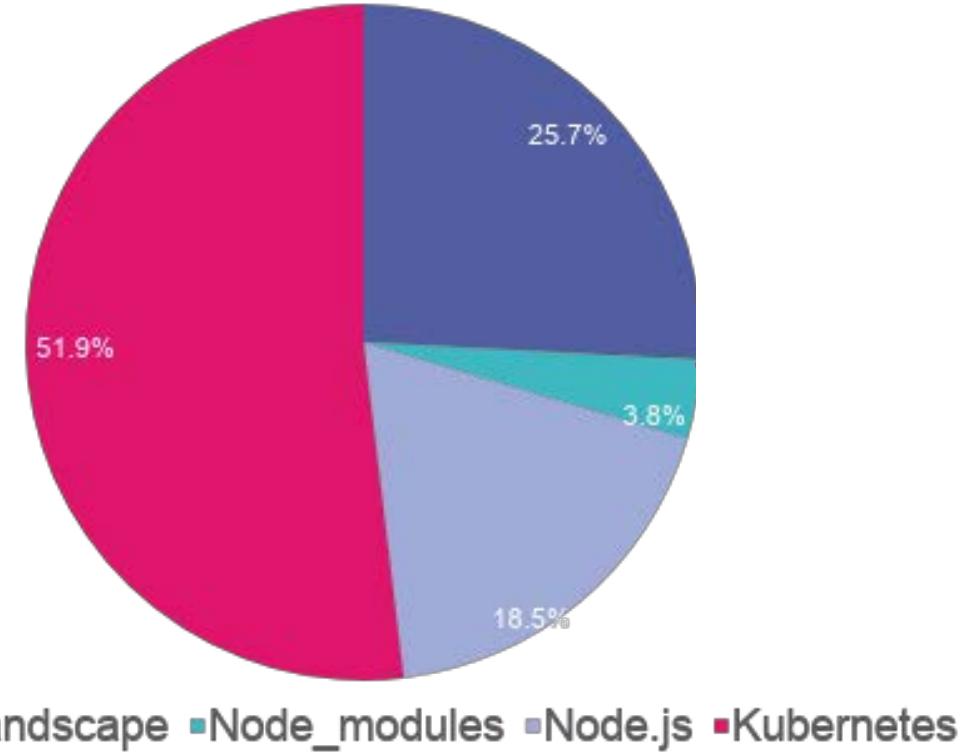
DEPLOYMENT PLATFORM

FRAMEWORK

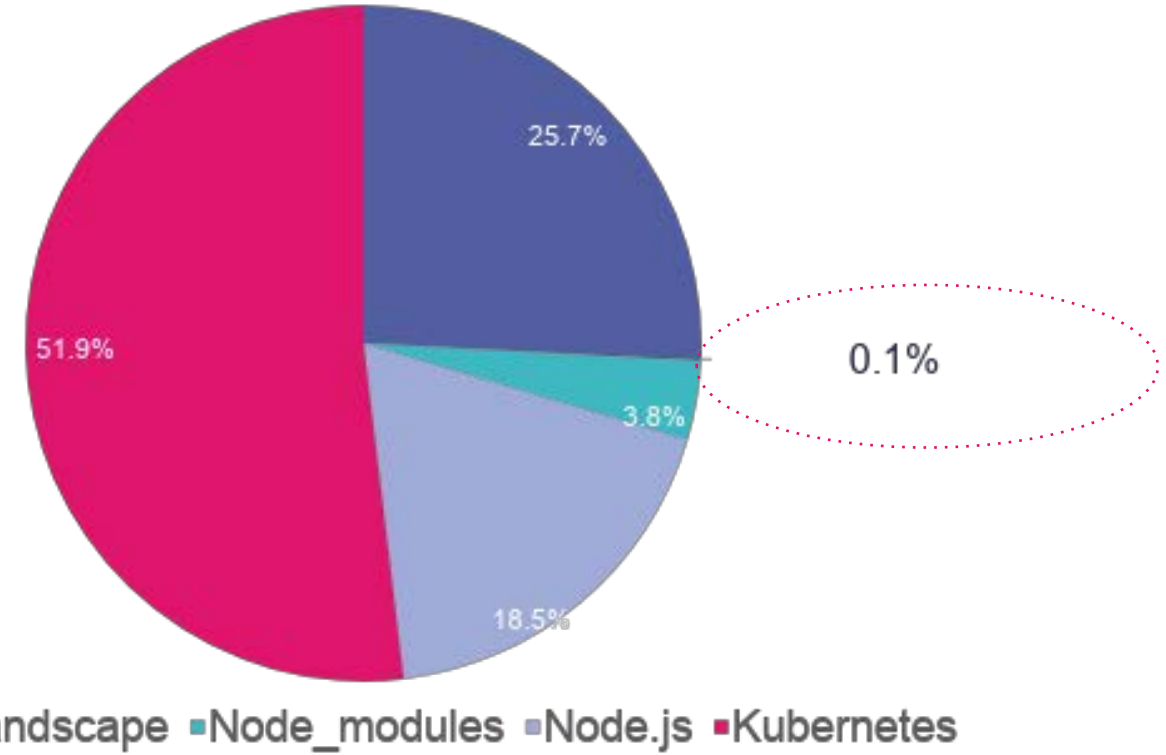
3RD PARTY LIBRARIES

OUR CODE IS ONLY 40 K SLOCS

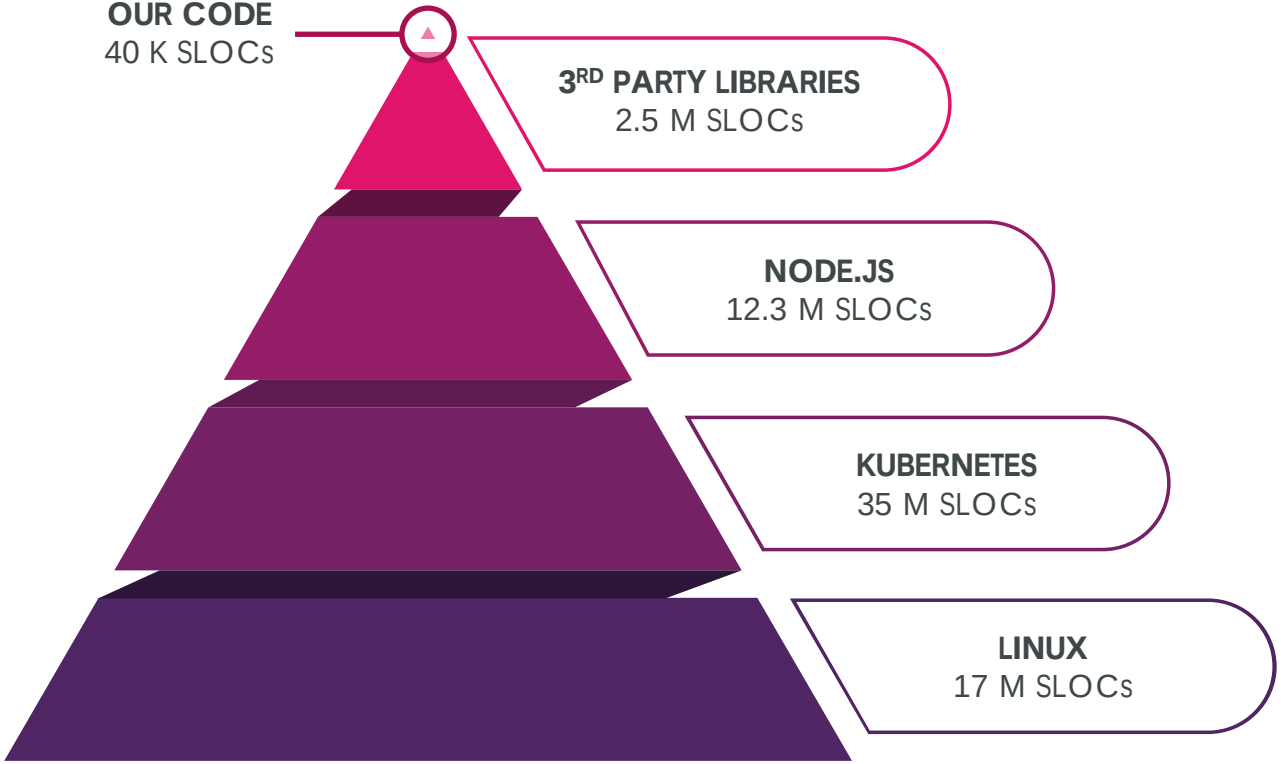
OUR APPLICATION SOFTWARE STACK



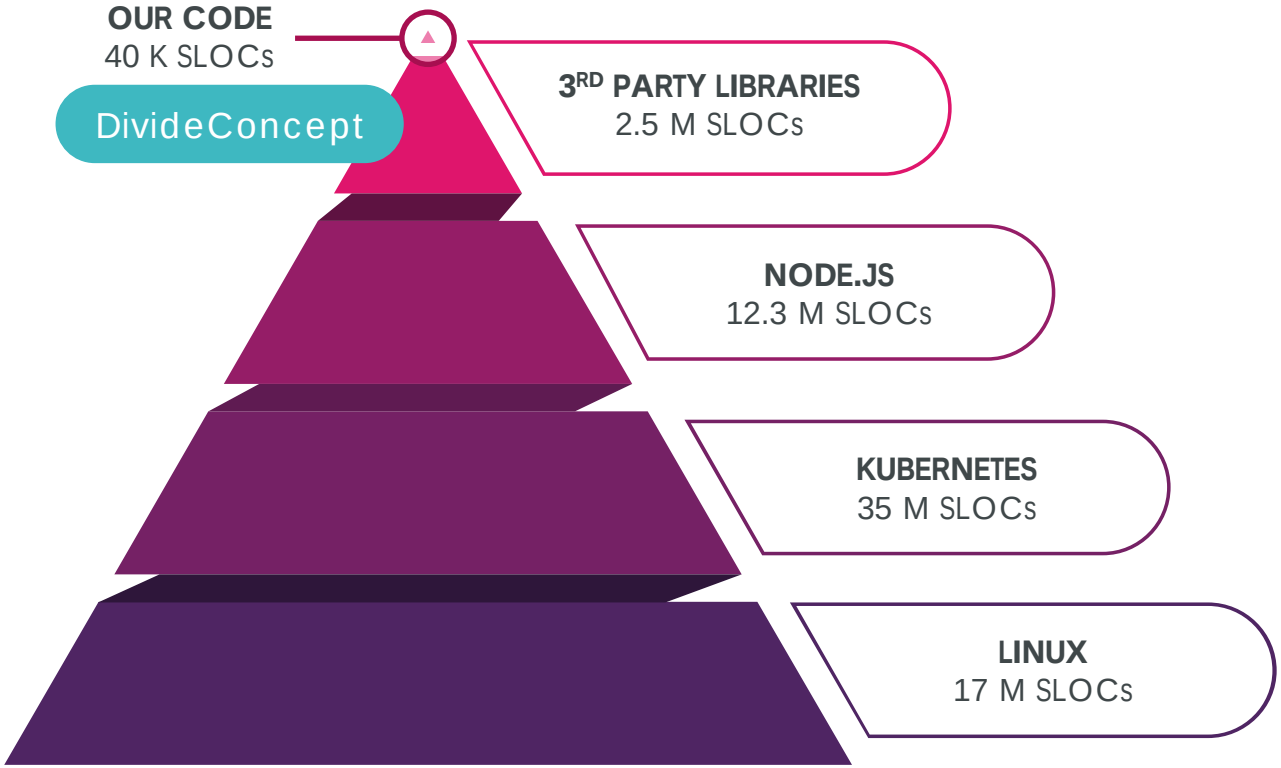
OUR CODE IS <0.1% OF OUR SOFTWARE STACK



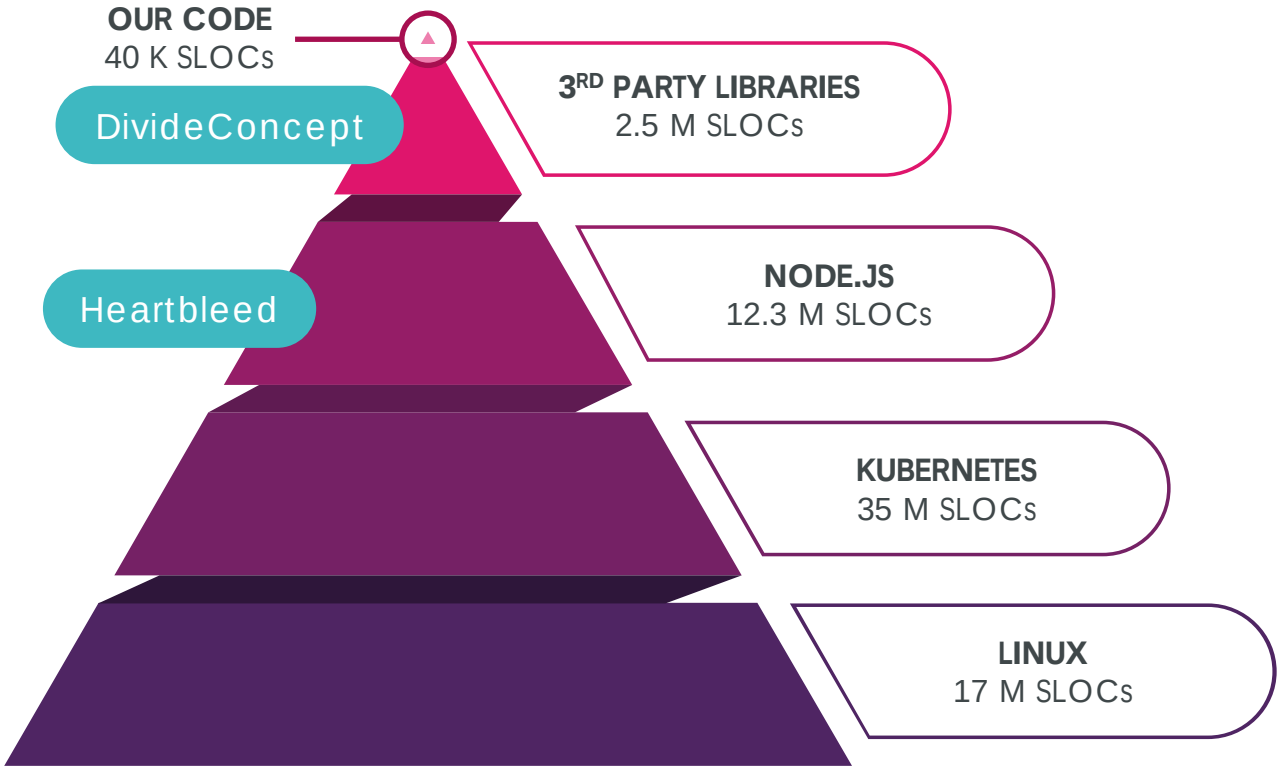
ALL OF THIS CODE IS VULNERABLE



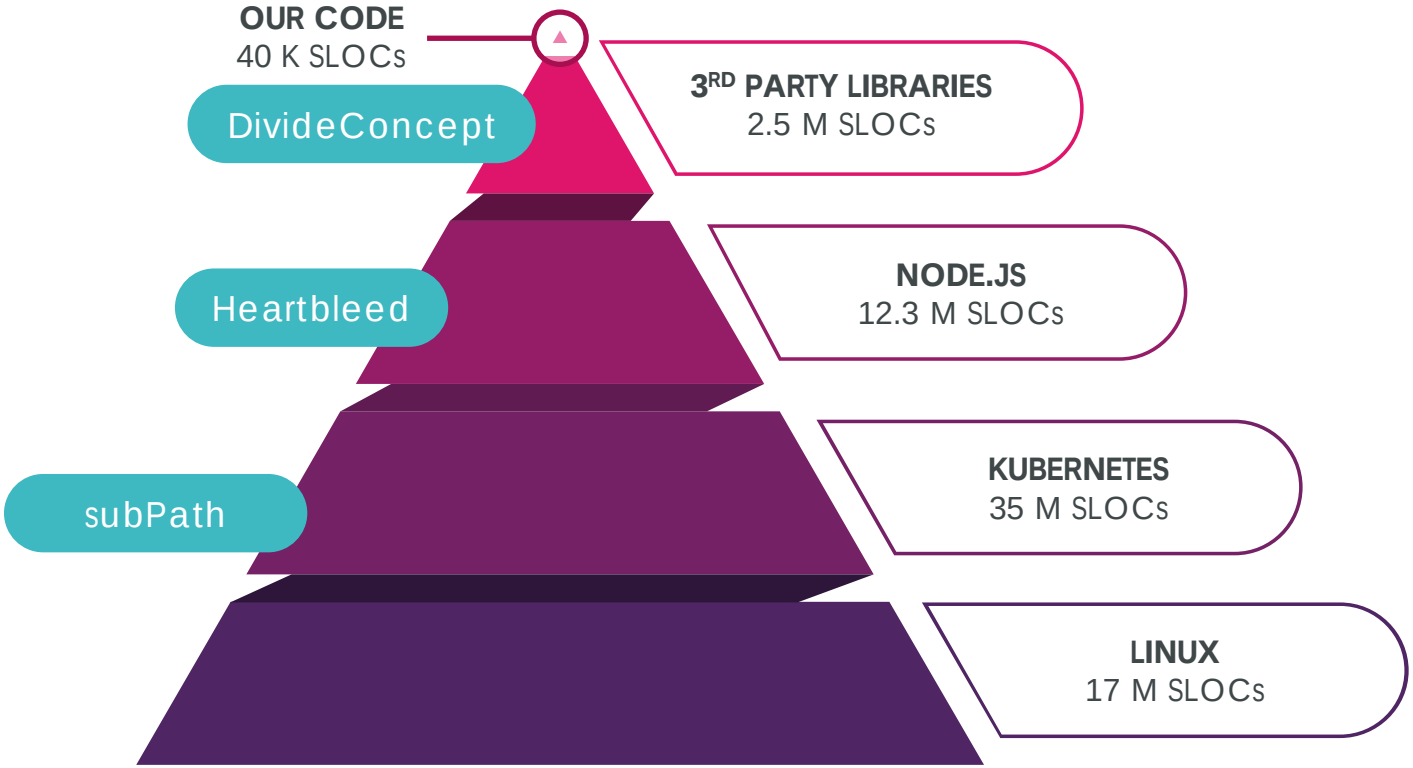
SPECTRE & MELTDOWN IN LINUX



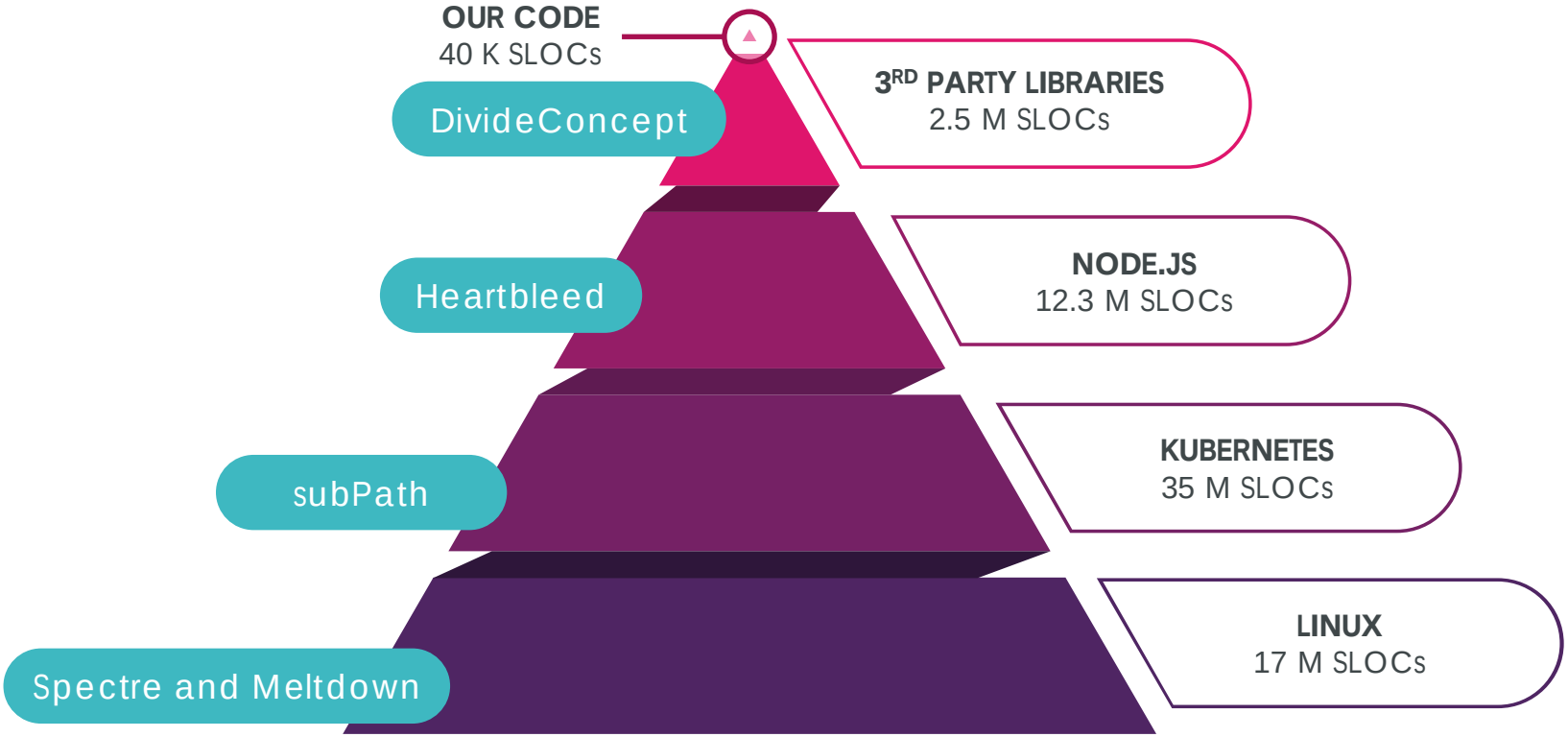
SPECTRE & MELTDOWN IN LINUX



SPECTRE & MELTDOWN IN LINUX



SPECTRE & MELTDOWN IN LINUX



The power of **open source** is the ability to leverage **thousands** of other **developers** that are **finding bugs** and **making fixes** to the software we depend on

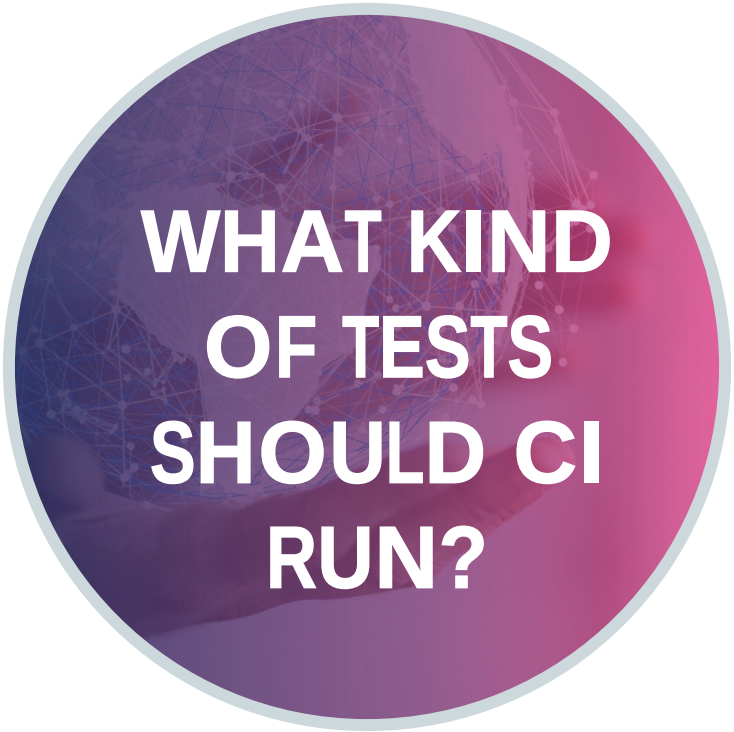
But a software **patch** does **not help** until we have **deployed** it into production

How can we have the **confidence** that that deployment **won't break** anything?

The power of **open source** is the ability to leverage **thousands** of other **developers** that are **finding bugs** and **making fixes** to the software we depend on

But a software **patch** does **not help** until we have **deployed** it into production

How can we have the **confidence** that that deployment **won't break** anything?


WHAT KIND
OF TESTS
SHOULD CI
RUN?

WHAT KIND OF TESTS SHOULD CI RUN?

Unit testing of individual portions of our source code in isolation?

WHAT KIND OF TESTS SHOULD CI RUN?

Unit testing of individual portions of our source code in isolation?

Integration testing, where we work with external systems like a database?

WHAT KIND OF TESTS SHOULD CI RUN?

WHAT KIND OF TESTS SHOULD CI RUN?

Unit, testing individual portions of our source code in isolation?

Integration testing, where we work with external systems like a database?

Regression testing, where we add a test after a failure?

Smoke testing, also known as build verification testing?

WHAT KIND OF TESTS SHOULD CI RUN?

Unit, testing individual portions of our source code in isolation?

Integration testing, where we work with external systems like a database?

Regression testing, where we add a test after a failure?

Smoke testing, also known as build verification testing?

All of the above.

HOW GOOD IS OUR CODE?

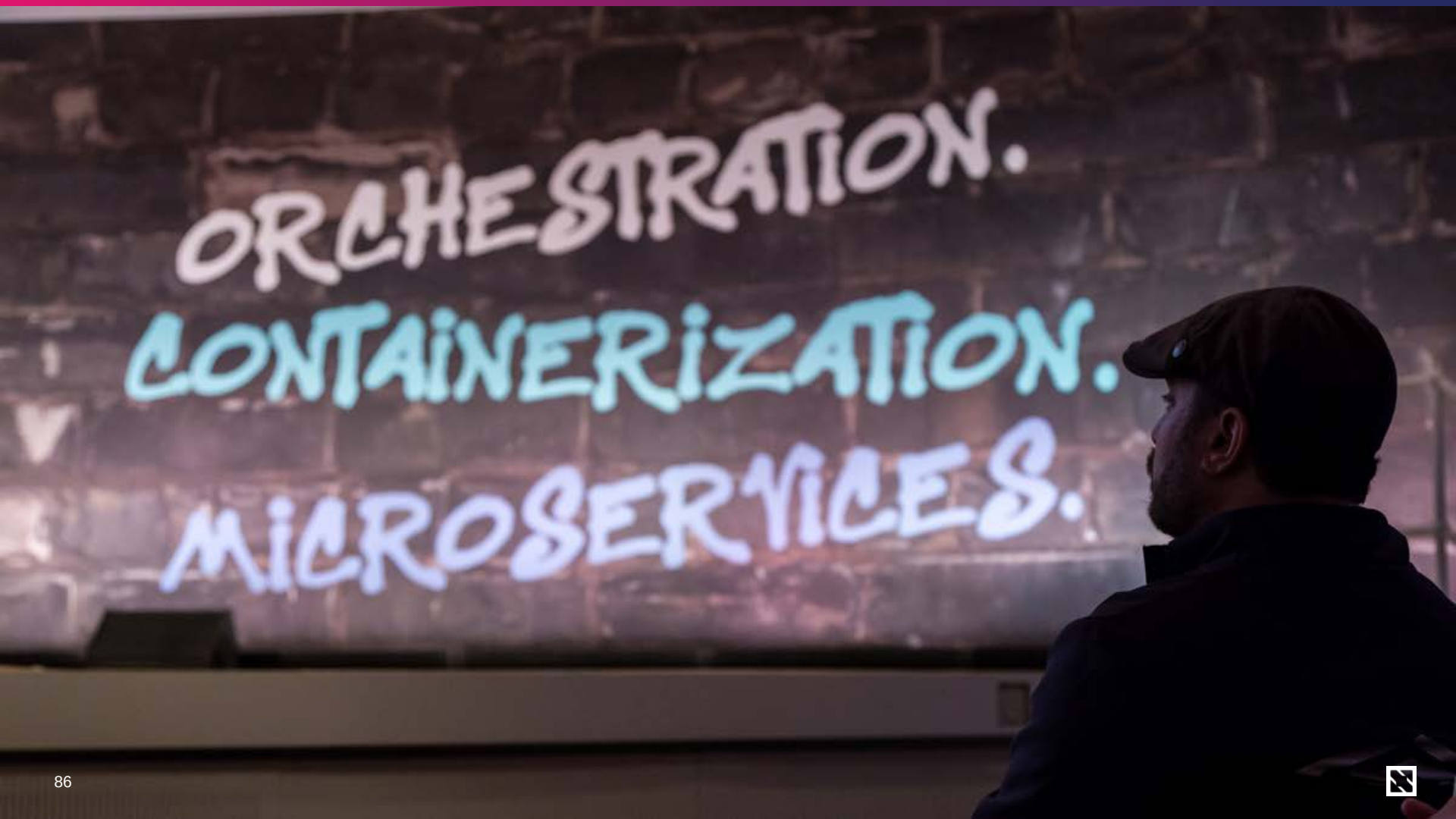
CLOUD NATIVE
COMPUTING FOUNDATION

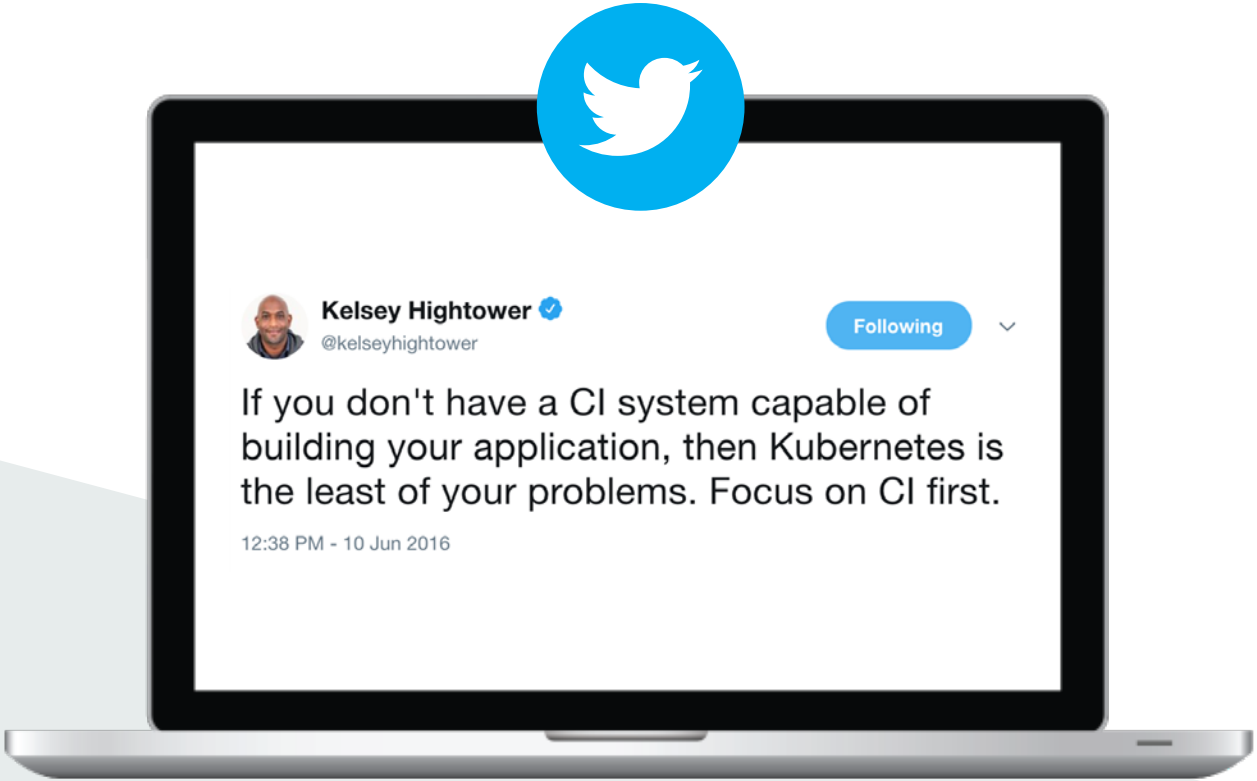
HOW GOOD IS OUR CODE?

NOT GOOD ENOUGH

We need to build in the systems and processes that enable us to continuously improve it

CLOUD NATIVE
COMPUTING FOUNDATION


ORCHESTRATION.
CONTAINERIZATION.
MICROSERVICES.


Kelsey Hightower
@kelseyhightower
Following
If you don't have a CI system capable of building your application, then Kubernetes is the least of your problems. Focus on CI first.
12:38 PM - 10 Jun 2016

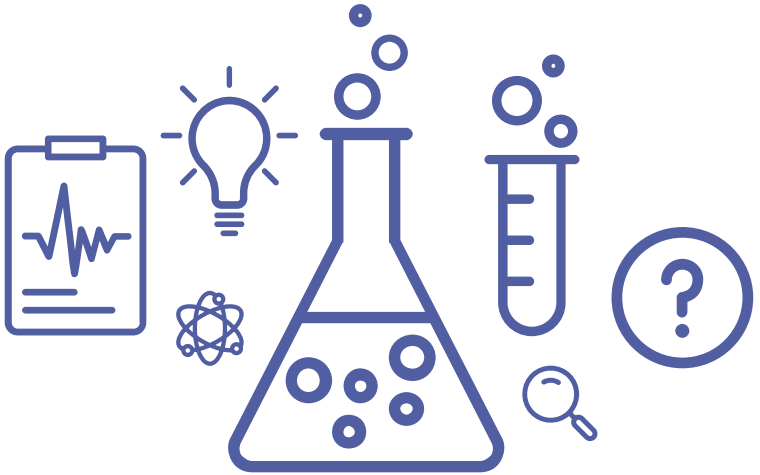
CONTINUOUS INTEGRATION (CI)

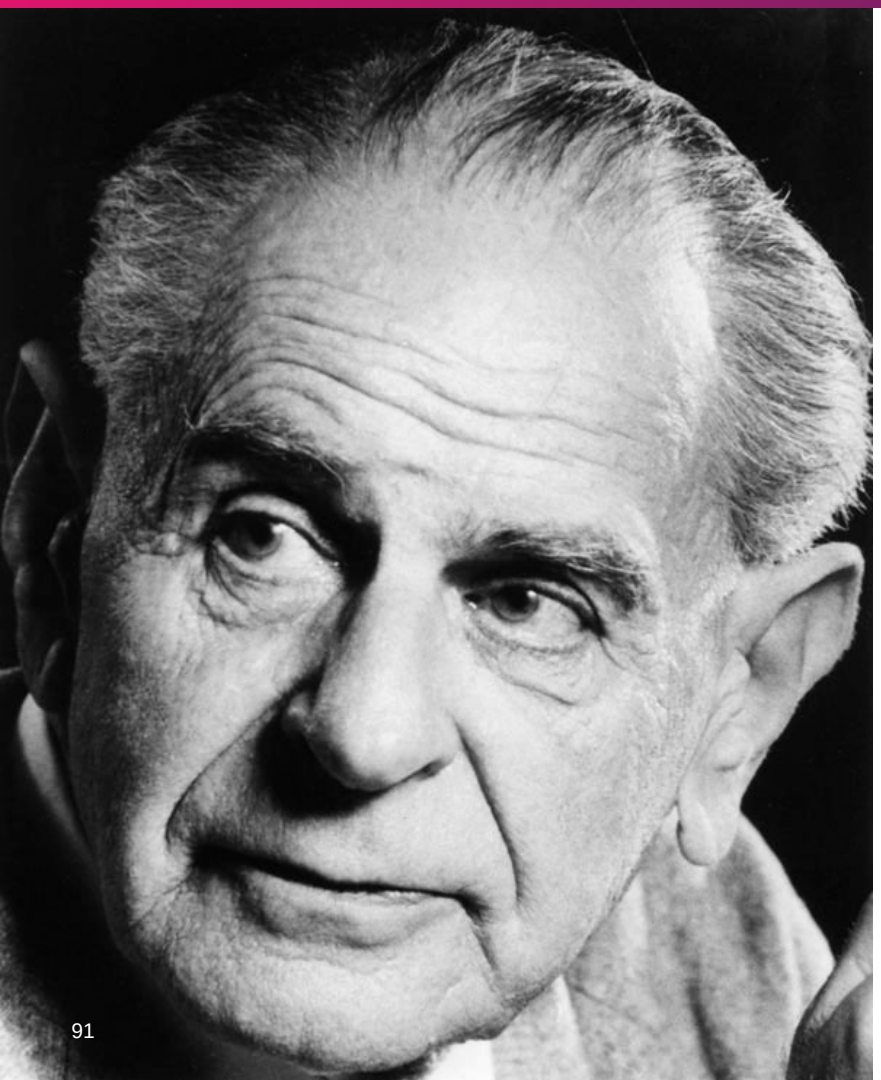
Continuous integration (CI) just means constant testing

But what is testing?

TESTING IS LIKE SCIENCE

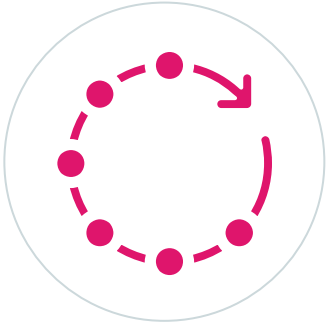
We have a hypothesis of what we believe our code should do, but we don't know for sure until we **test against objective reality**

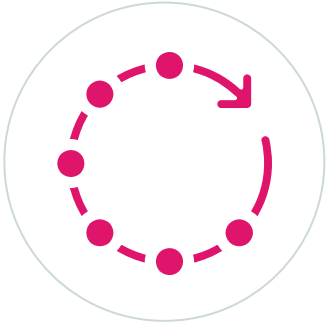




Karl Popper
defined science as

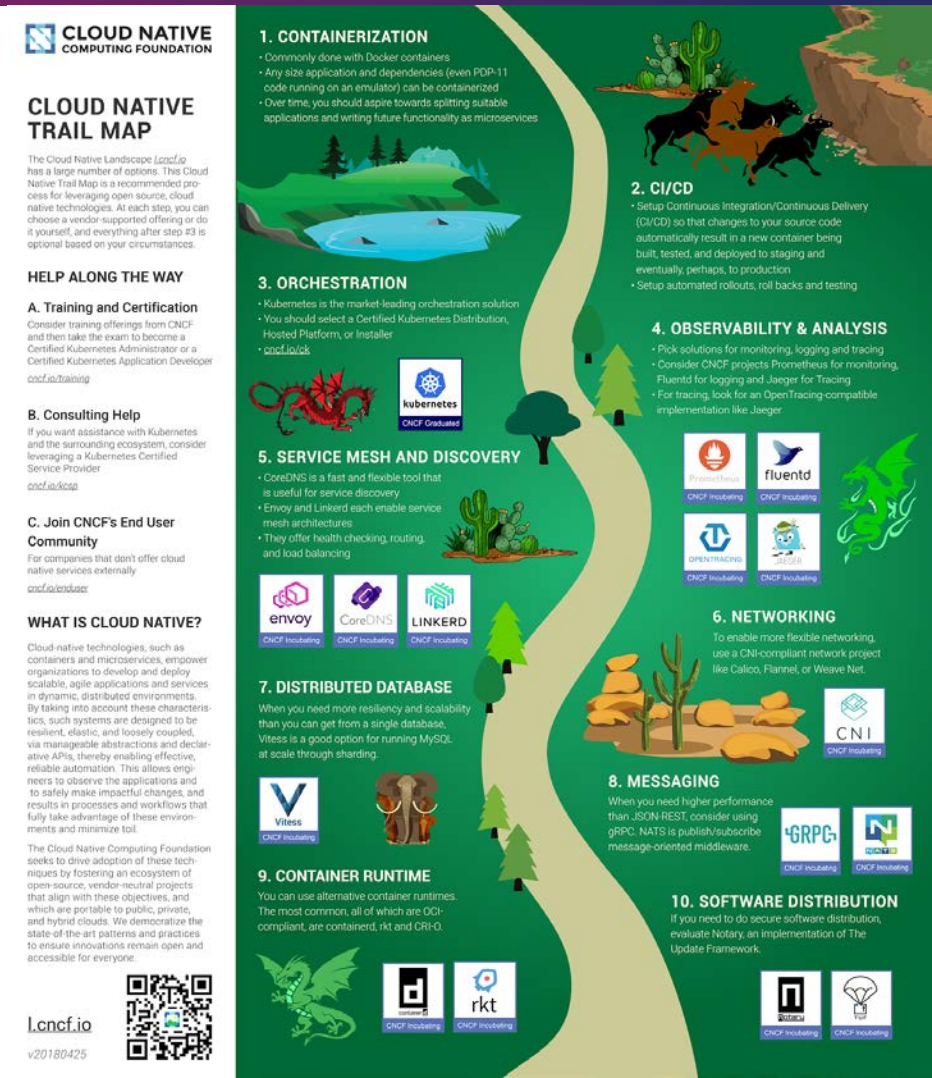
BEING TESTABLE AND FALSIFIABLE





HOW DOES CONTINUOUS INTEGRATION FIT INTO THE CLOUD NATIVE JOURNEY?

CLOUD NATIVE TRAIL MAP



Cloud Native Landscape

v20180425

See the interactive landscape at l.cncf.io

Greyed logos are not open source

App Definition and Development

Database and Data Warehouse

Vitess (CNCF Incubating)

Streaming

NATS (CNCF Incubating)

Source Code Management

Application Definition and Image Build

Continuous Integration / Continuous Delivery (CI/CD)

Orchestration & Management

Scheduling & Orchestration

kubernetes (CNCF Graduated)

Coordination & Service Discovery

CoreDNS (CNCF Incubating)

Service Management

gRPC (CNCF Incubating) · envoy (CNCF Incubating) · LINKERD (CNCF Incubating) · Open Policy Agent (CNCF Sandbox)

Runtime

Cloud-Native Storage

ROOK (CNCF Sandbox)

Container Runtime

containerd (CNCF Incubating) · rkt (CNCF Incubating)

Cloud-Native Network

CNI (CNCF Incubating)

Provisioning

Host Management / Tooling

Infrastructure Automation

Container Registries

Secure Images

Notary (CNCF Incubating) · TUF (CNCF Incubating)

Key Management

spiffe (CNCF Sandbox) · SPIRE (CNCF Sandbox)

Cloud

Public

Private

Platforms

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

Non-Certified Kubernetes

PaaS/Container Service

Kubernetes Certified Service Provider

Observability & Analysis

Monitoring

Prometheus (CNCF Incubating)

Logging

fluentd (CNCF Incubating)

Tracing

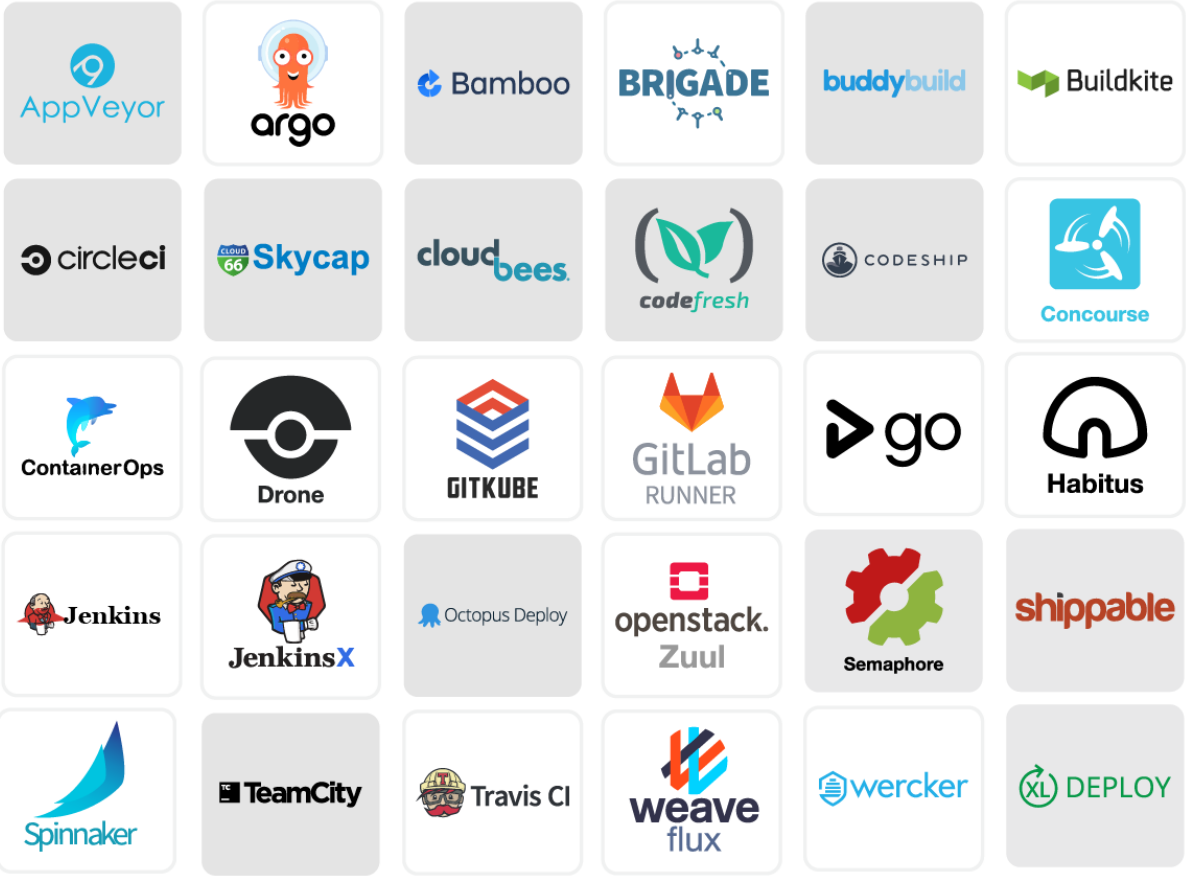
JAEGER (CNCF Incubating) · OPENTRACING (CNCF Incubating)

Kubernetes Training Partner

Serverless

Special

Continuous Integration / Continuous Delivery (CI/CD)



PLEASE TRY THE INTERACTIVE LANDSCAPE NOW:

l.cncf.io

Recent News

Certified Kubernetes Conformance

- CNCF launched a software conformance program for Kubernetes
  - Implementations run conformance tests and upload results
  - New mark and more flexible use of Kubernetes trademark for conformant implementations
  - Taking submissions now for K8s 1.9 & 1.10
  - https://www.cncf.io/certification/software-conformance/

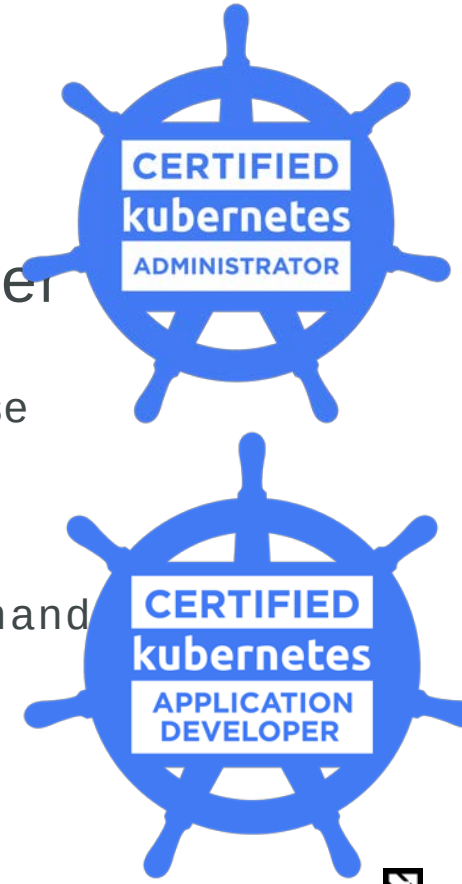
58 Certified Kubernetes Partners

CNCF MOOC and Online Training

- Free Introduction to Kubernetes self-paced course offered with edX
- Kubernetes Fundamentals course
  - Content maps to Certified Kubernetes Administrator (CKA) exam
  - $299, intermediate level
- Open source curriculum available for companies offering training
  - CKA Exam coupons available with a bulk discount

Online, Proctored Kubernetes Exams

- Certified Kubernetes Administrator (CKA)
  - Over 1,500 registrations already
  - https://www.cncf.io/certification/expert/cka/
- Certified Kubernetes Application Developer (CKAD)
  - Certifies that users can design, build, configure, and expose cloud native applications for Kubernetes
  - https://www.cncf.io/certification/expert/cka/ckad/
- Both tests
  - Tests consist of a set of scenarios to resolve from the command line over 3 hours; there is no multiple choice
  - Each exam is $300
  - Quarterly exam updates to match K8s releases



Kubernetes Certified Service Provider

A pre-qualified tier of vetted service providers who have deep experience helping enterprises successfully adopt Kubernetes through support, consulting, professional services and/or training.

Benefits

- Placement at the top of https://kubernetes.io/partners/
- Monthly private meetings with cloud native project leaders, TOC members, and representatives from the Governing Board
- Access to leads from the kubernetes.io for end users looking for support

Requirements

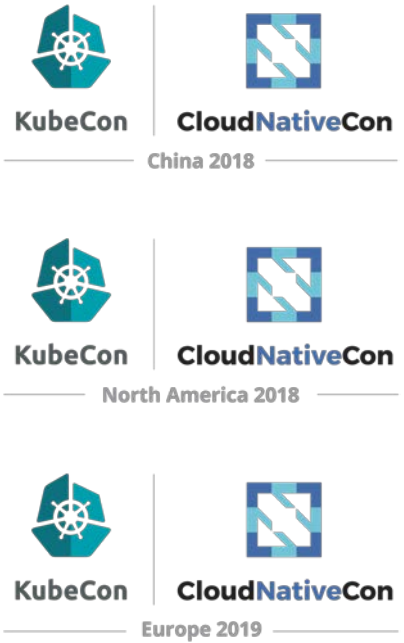
- Three or more certified engineers
- Demonstrable activity in the Kubernetes community including active contribution
- Business model to support enterprise end users

https://www.cncf.io/certification/kcsp/

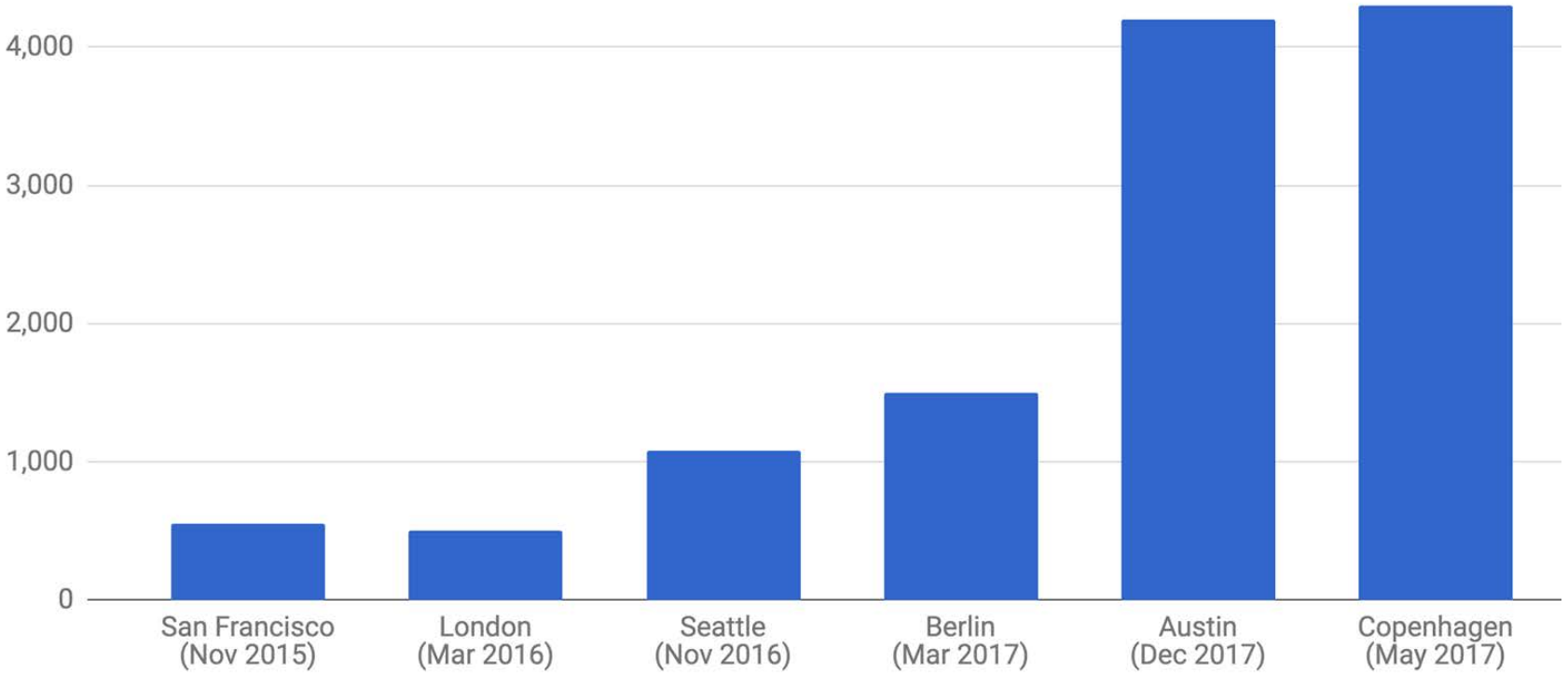
44 KCSP Partners

KubeCon + CloudNativeCon

- China
  - Shanghai: November 14-15, 2018
  - Sponsorships open
- North America
  - Seattle: December 11 - 13, 2018
  - Sponsorships open
- Europe
  - Barcelona: May 21 - 23, 2018



KubeCon + CloudNativeCon Attendees



Joining the CNCF is easy!

Join now: https://cncf.io/join

or

General Inquiries: info@cncf.io
PR: pr@cncf.io
Event Sponsorships: sponsor@cncf.io
Membership: memberships@cncf.io