

# HOW GOOD IS OUR CODE?

Dan Kohn  
Executive Director, CNCF

# Cloud Native Computing Foundation

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

## Graduated



## Incubating



## Sandbox



- 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.



We are regularly adding projects; for the most up-to-date listing of all projects visit [tlfprojects.org](https://www.linuxfoundation.org/projects)



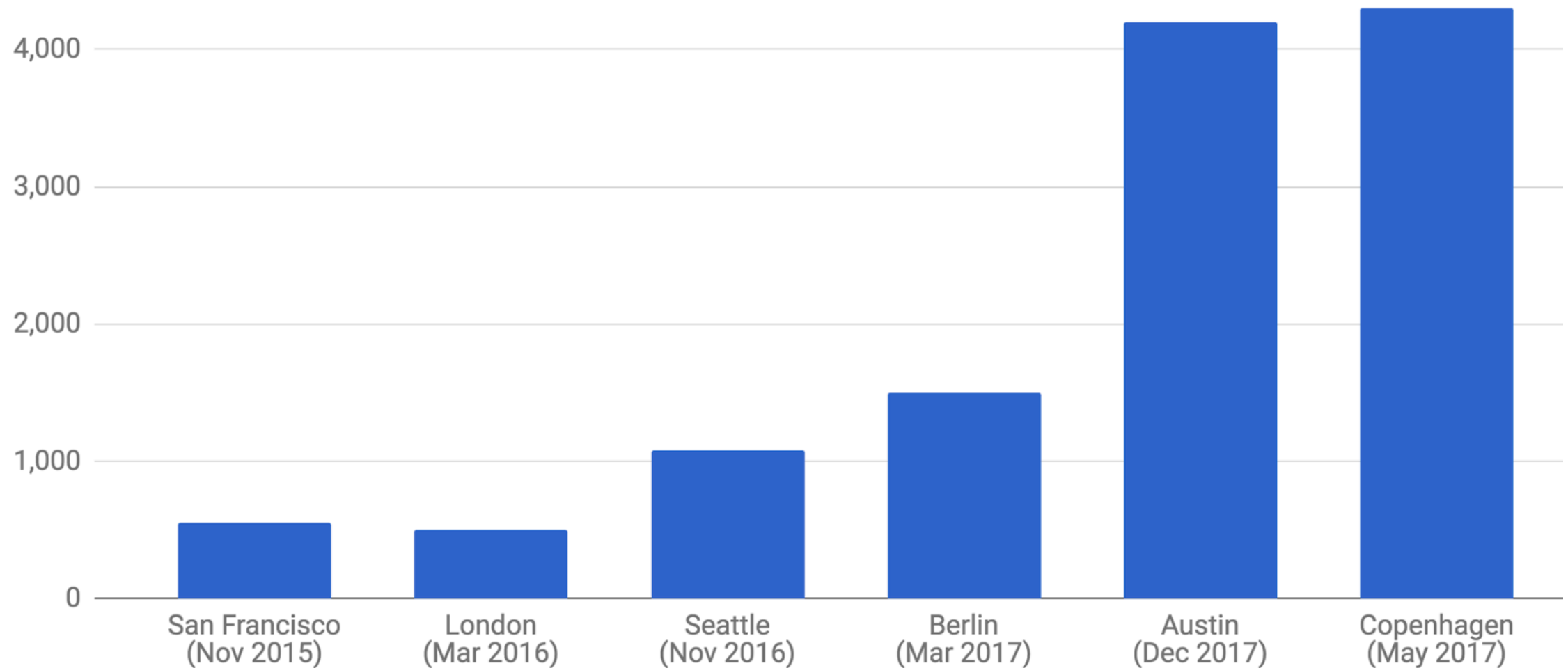
# KubeCon + CloudNativeCon

- China
  - [Shanghai](#): November 14-15, 2018
  - Sponsorships [open](#)
- North America
  - Seattle: December 11 - 13, 2018
  - Sponsorships [open](#)
- Europe
  - Barcelona: May 21 - 23, 2019



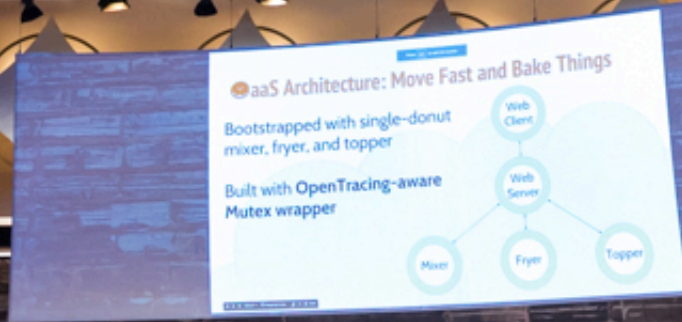


# KubeCon + CloudNativeCon Attendees



HOW GOOD IS  
OUR CODE?





SUPPORTING THE CLOUD NATIVE ECOSYSTEM

[www.cncf.io](http://www.cncf.io) | @CloudNativeFdn

ORCHESTRATION.  
CONTAINERIZATION.  
MICROSERVICES.







CLOUD  
NATIVE  
CON  
Europe 2017



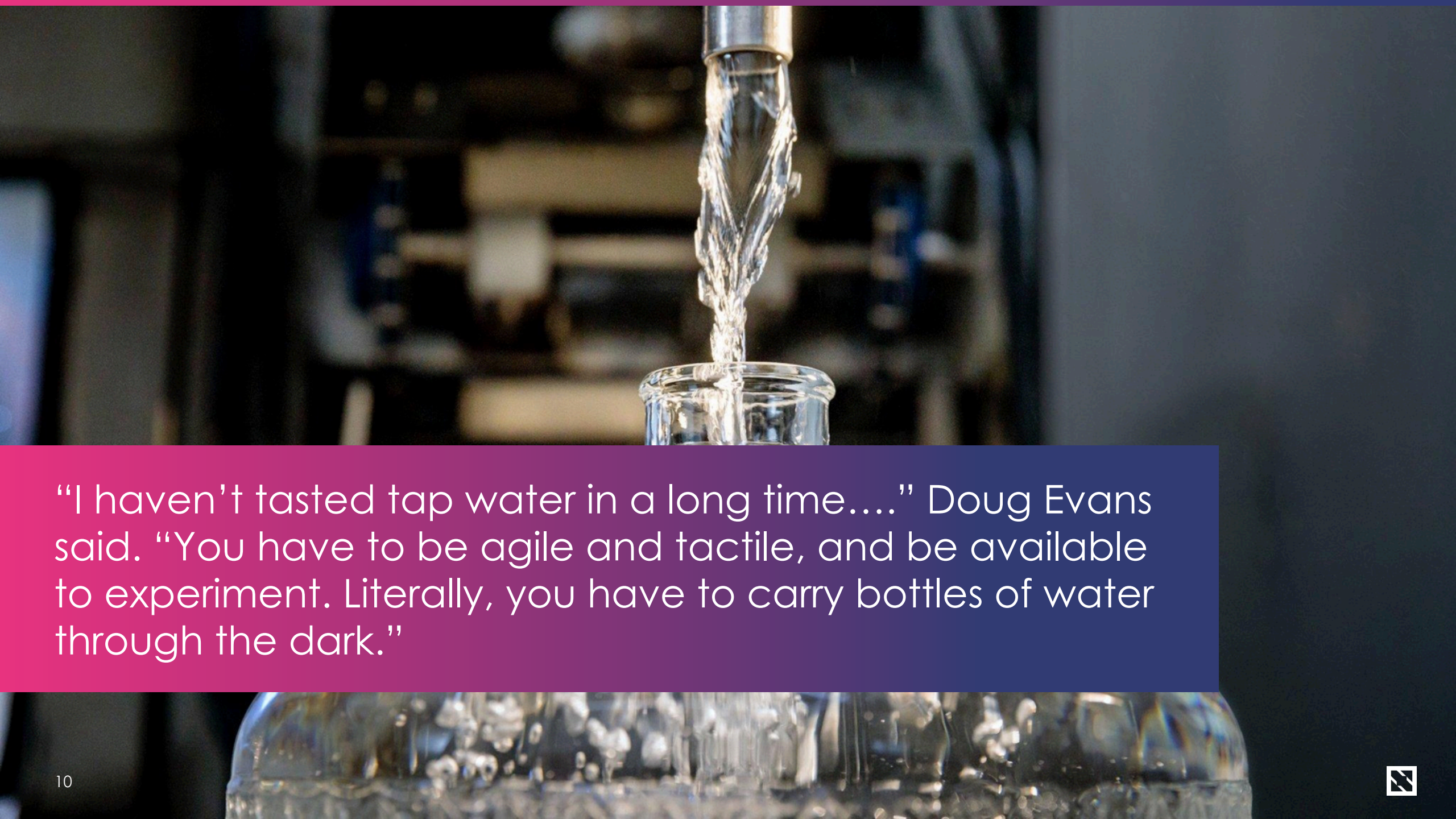
KubeCon  
A CNCF EVENT

ORCHESTRATION.  
CONTAINERIZATION.  
MICROSERVICES.



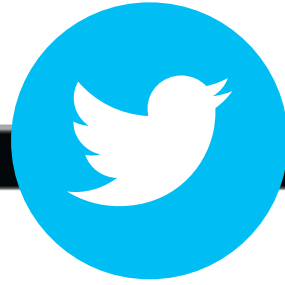




A close-up photograph of water being poured from a chrome faucet into a clear glass bottle. The water is captured in mid-pour, creating a dynamic, splashing effect. The background is dark and out of focus, showing hints of a kitchen or laboratory setting with shelves and equipment. A semi-transparent purple and blue gradient box is overlaid on the lower half of the image, containing white text.

“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



# OUR SOFTWARE IS NOT AS GOOD AS SQLITE



<https://www.sqlite.org/testing.html>

- 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

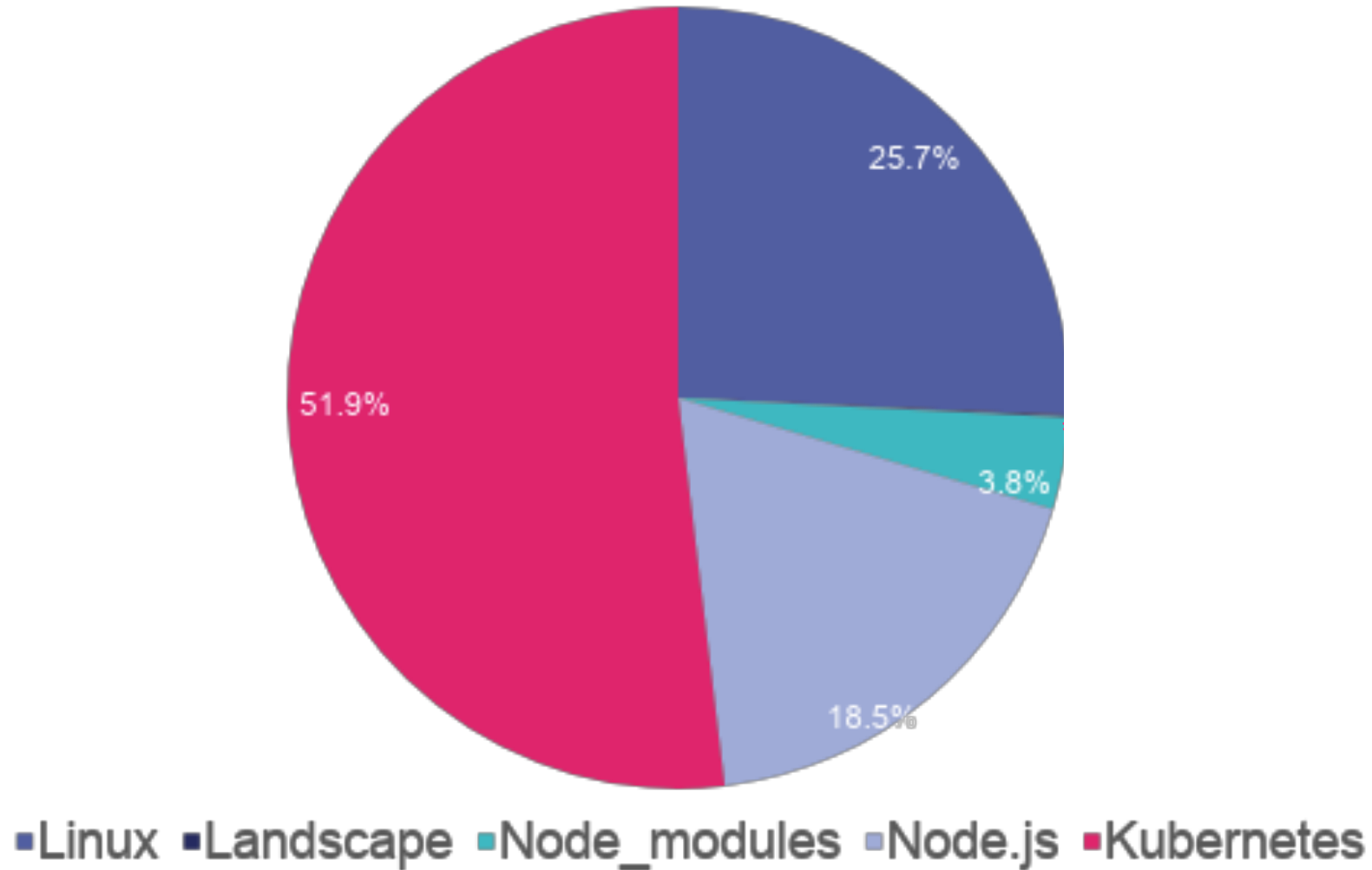
OUR CODE



INTERACTIVE  
LANDSCAPE  
40 K SLOCs

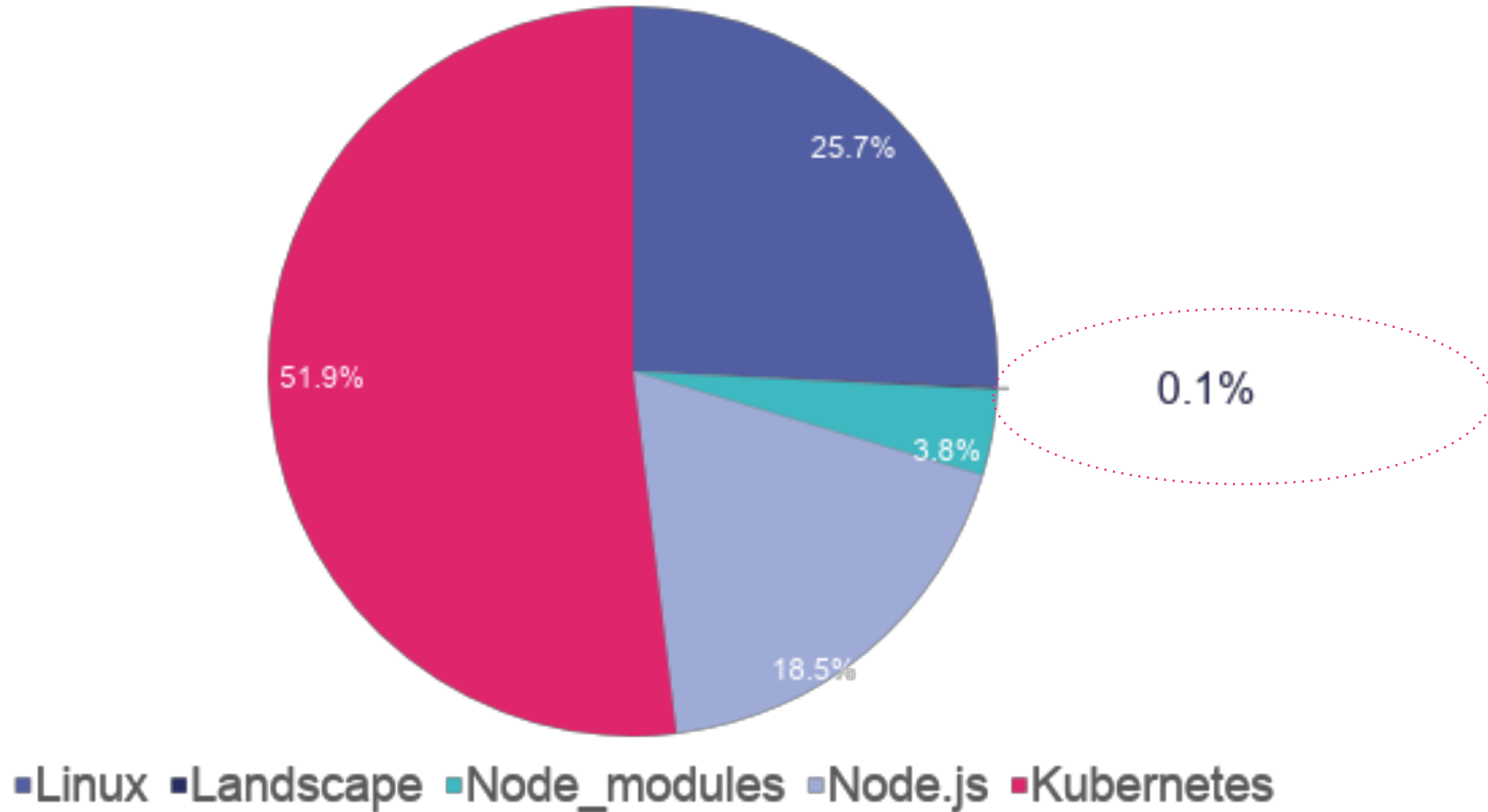


# OUR APPLICATION SOFTWARE STACK

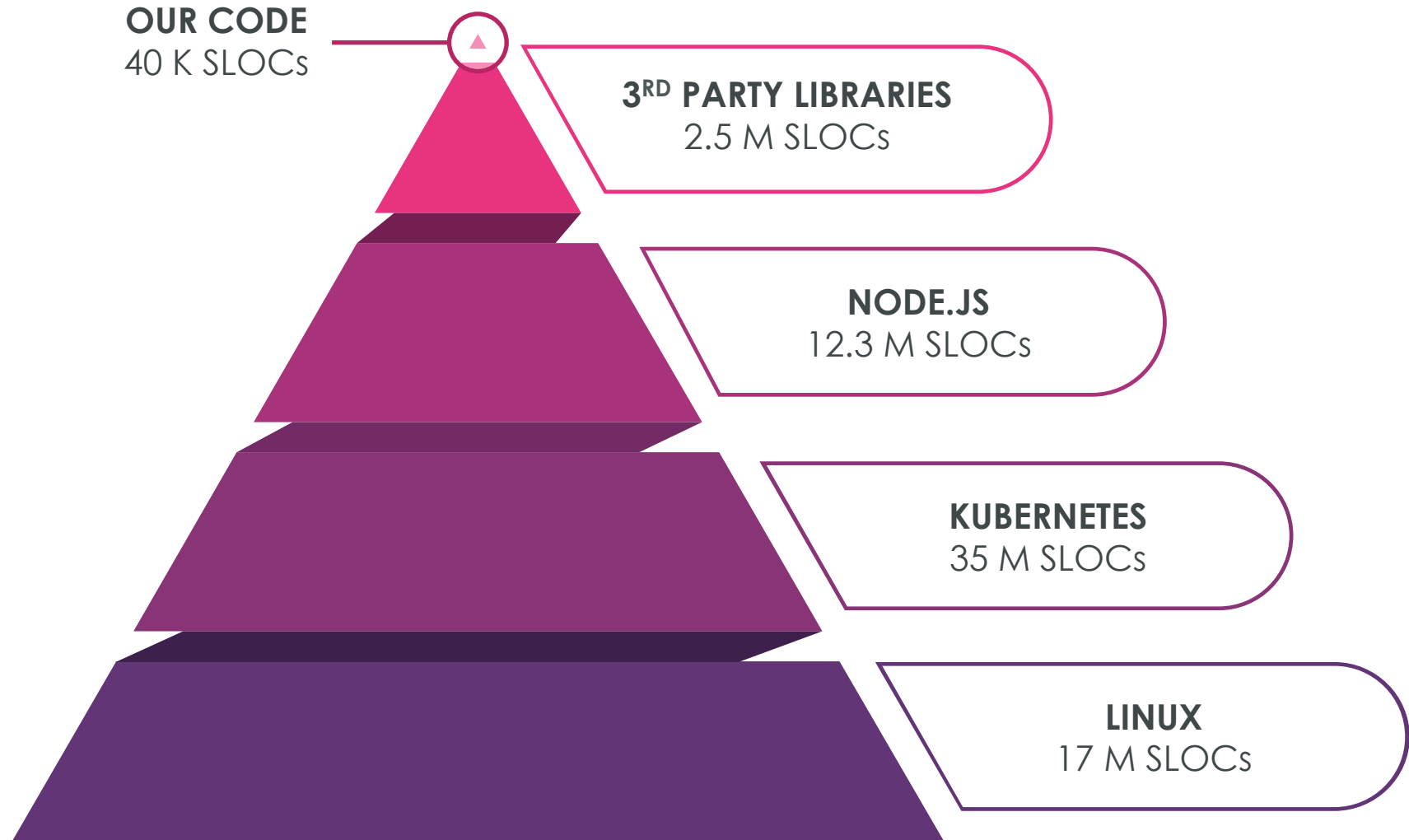




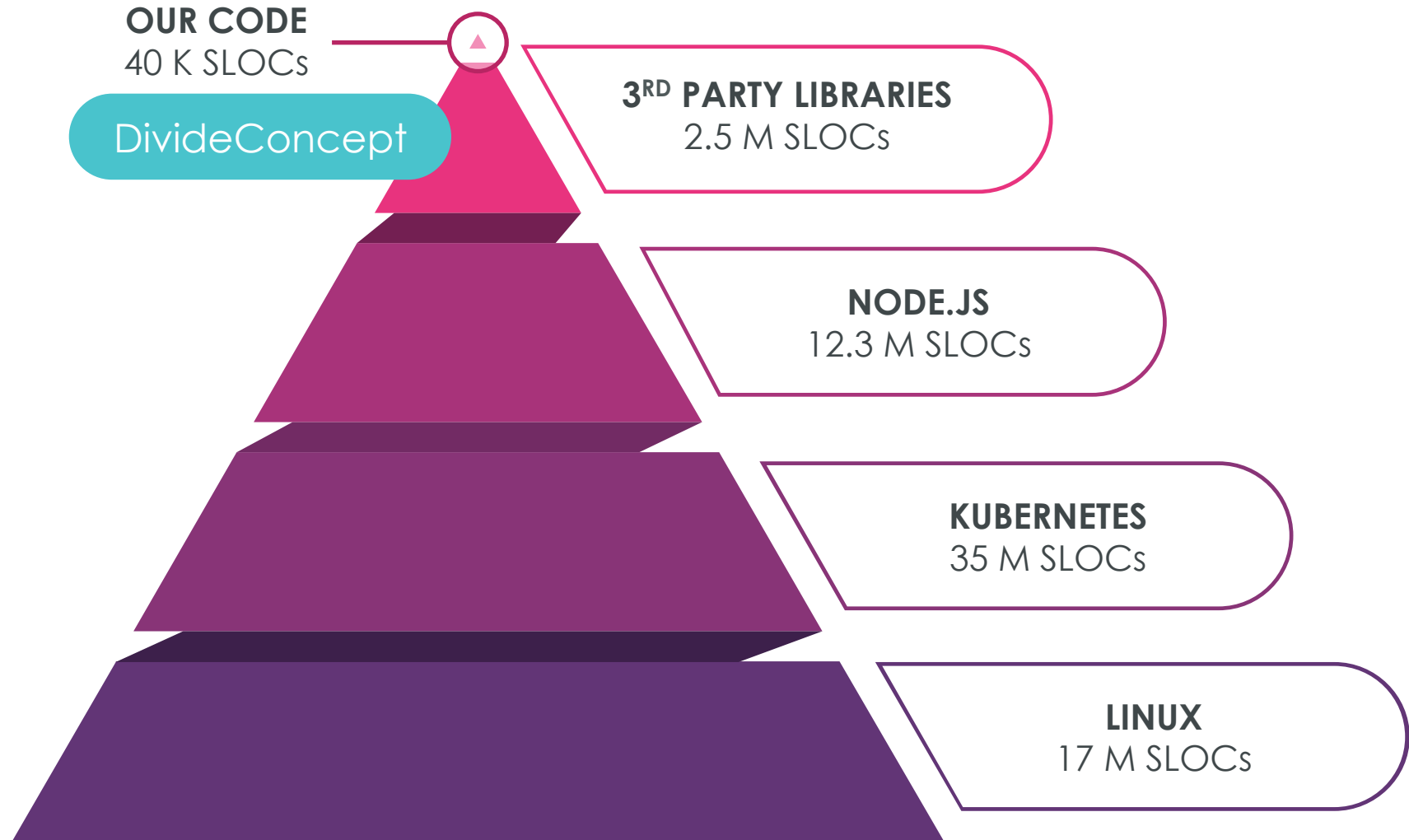
# OUR CODE IS <0.1% OF OUR SOFTWARE STACK



# ALL OF THIS CODE IS VULNERABLE

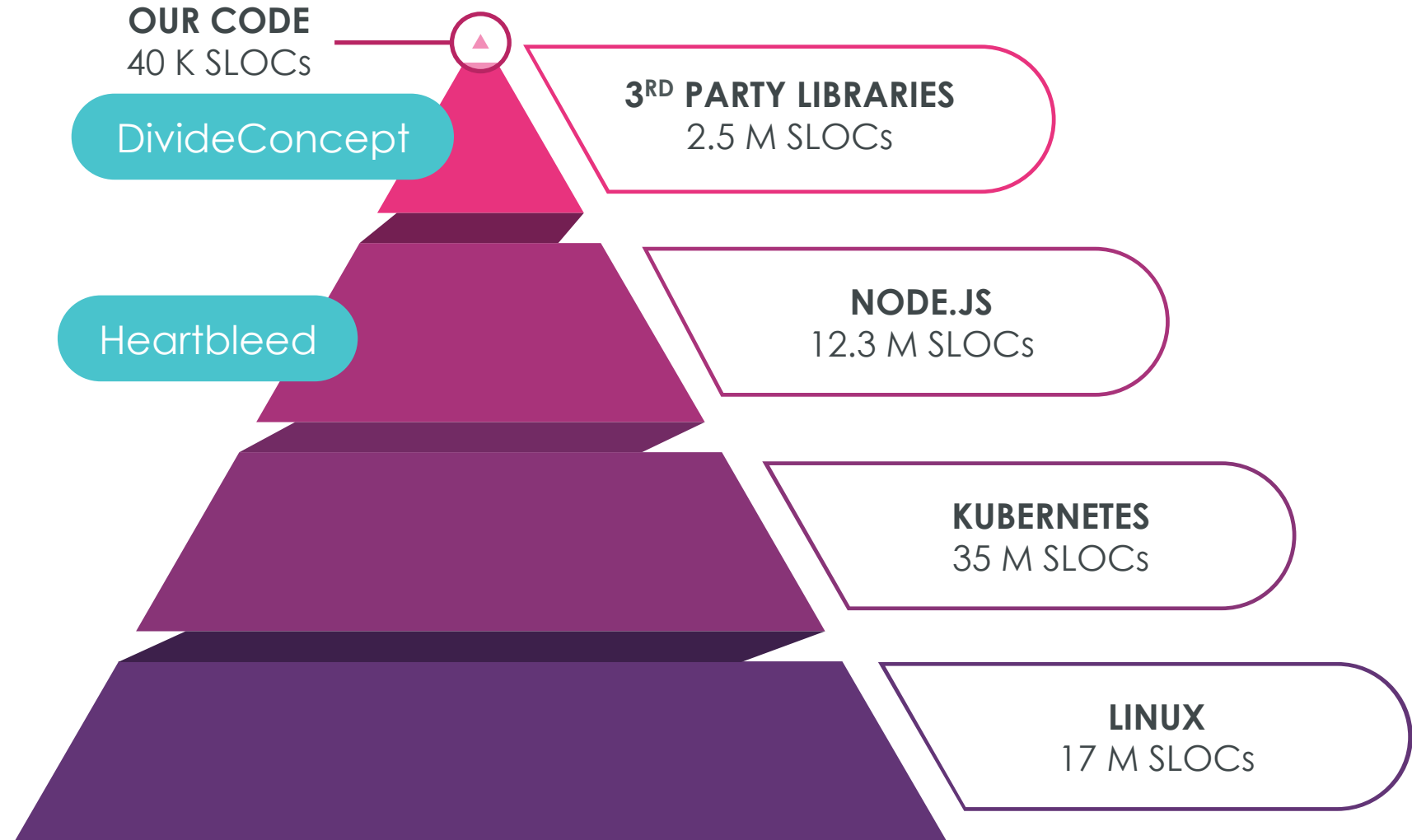


# SPECTRE & MELTDOWN IN LINUX

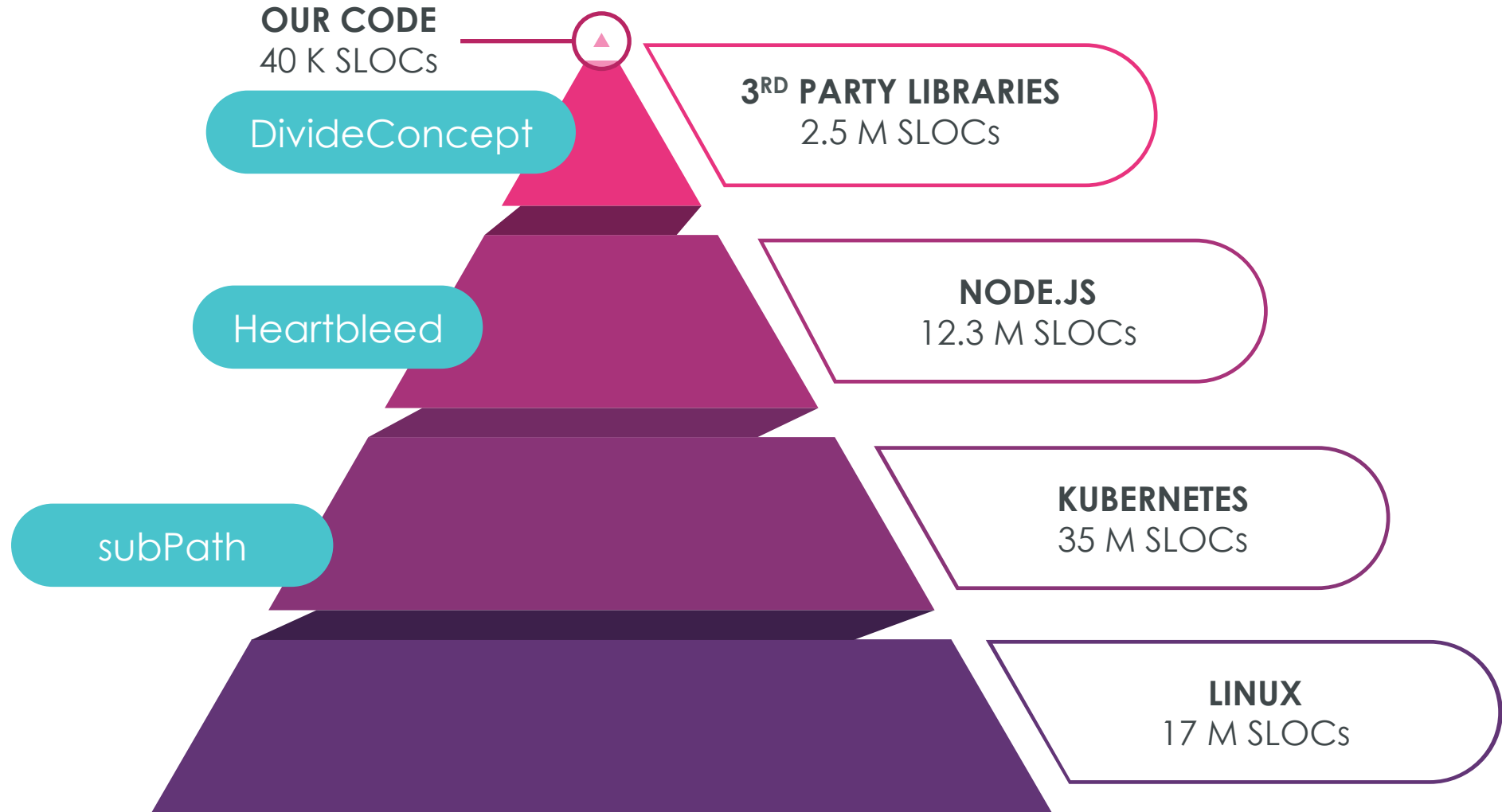




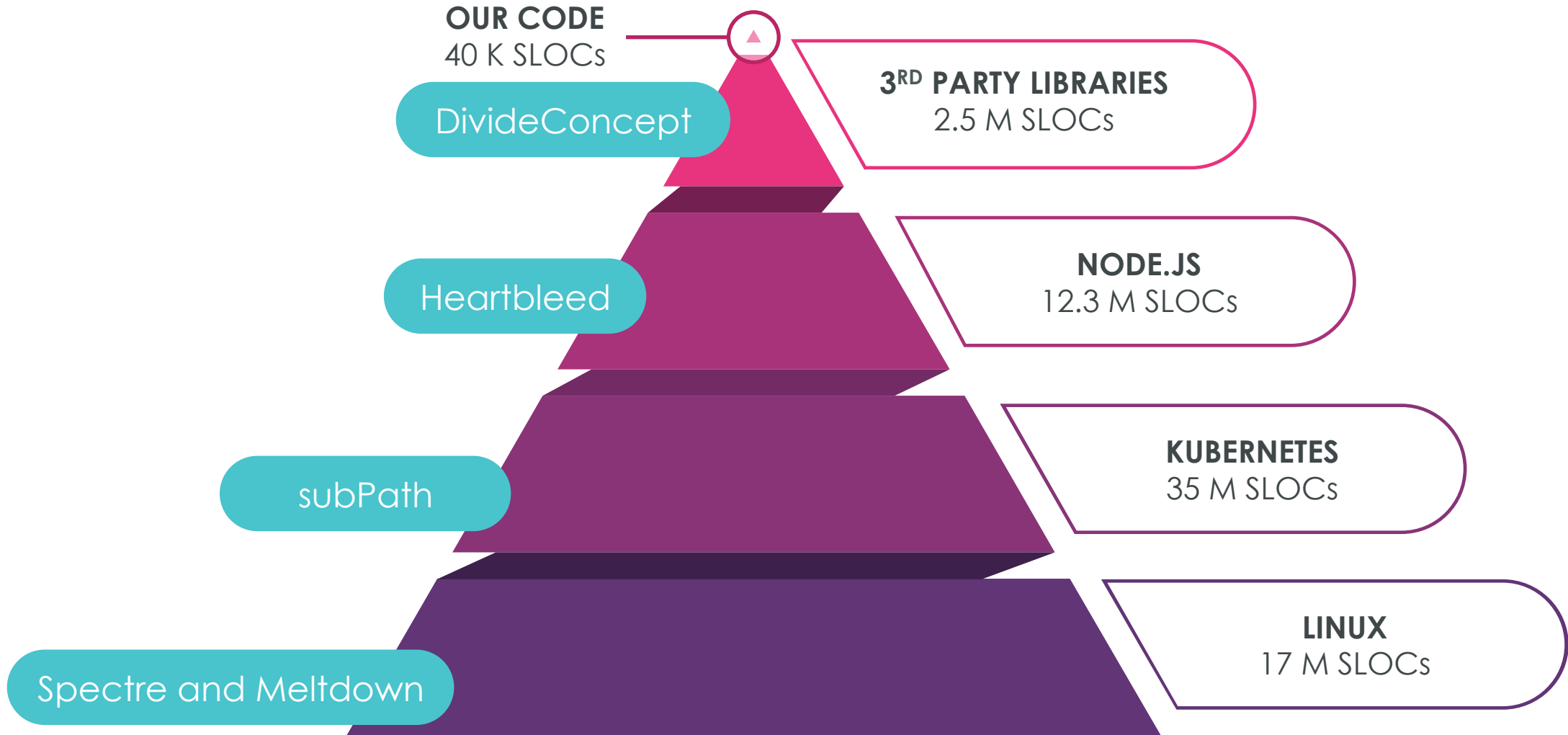
# SPECTRE & MELTDOWN IN LINUX



# 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?

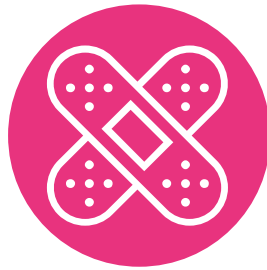




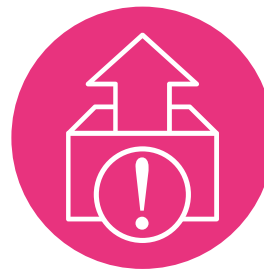
The power of **open source** is the ability to leverage **thousands** of other **developers** that are **finding bugs** and **making fixes** to the software we depend on



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?




The Answer is  
**CONTINUOUS  
INTEGRATION  
(CI)**



**WHAT KIND  
OF TESTS  
SHOULD CI  
RUN?**

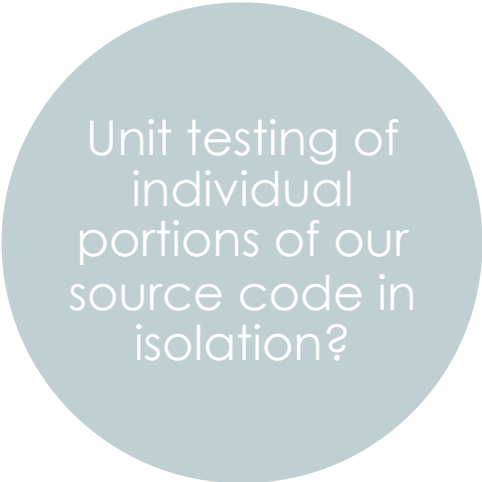
# WHAT KIND OF TESTS SHOULD CI RUN?




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?

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



# 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?

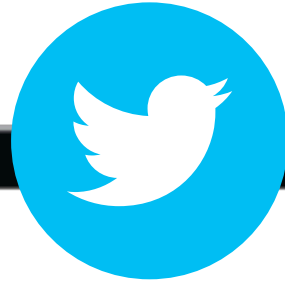
# HOW GOOD IS OUR CODE?

## NOT GOOD ENOUGH

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

A man with a beard and a dark cap is seen from the side, looking at a chalkboard. The chalkboard has three terms written on it in a stylized, hand-drawn font. The background is dark and textured, resembling a brick wall.

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)

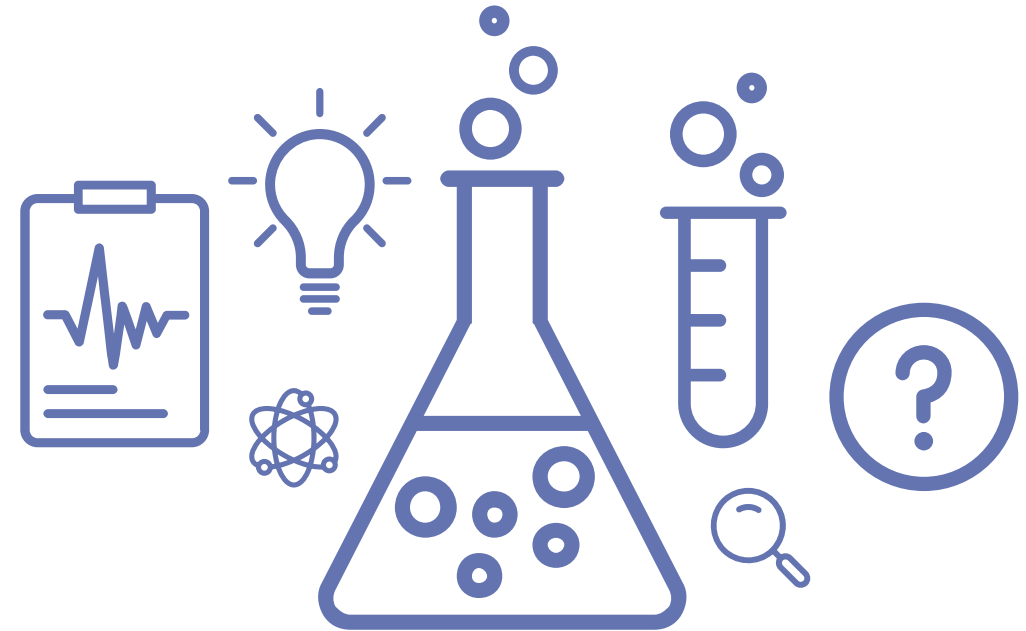
**Continuous  
integration (CI)**  
just means  
constant testing

But what is  
testing?

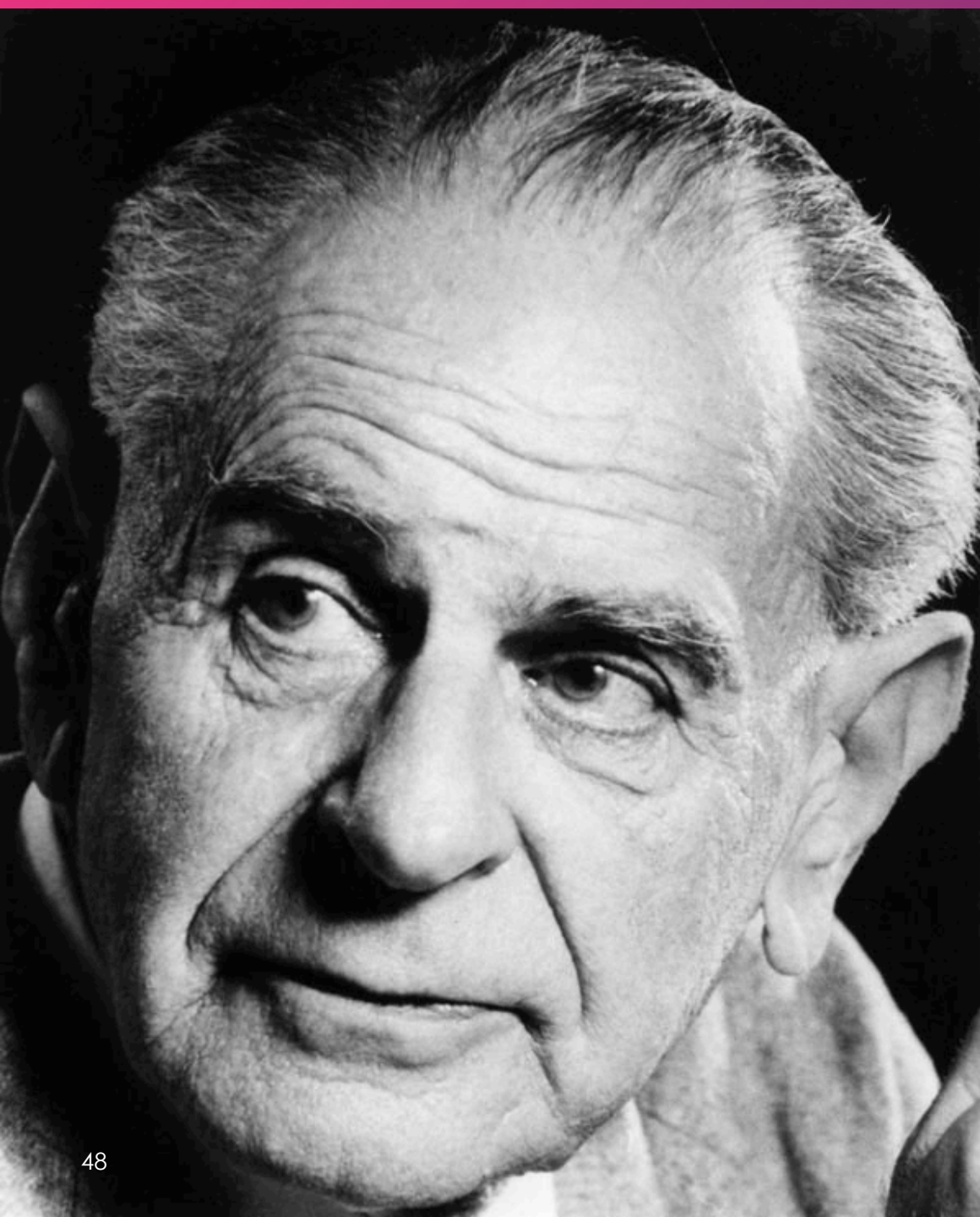


# TESTING IS LIKE SCIENCE

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**





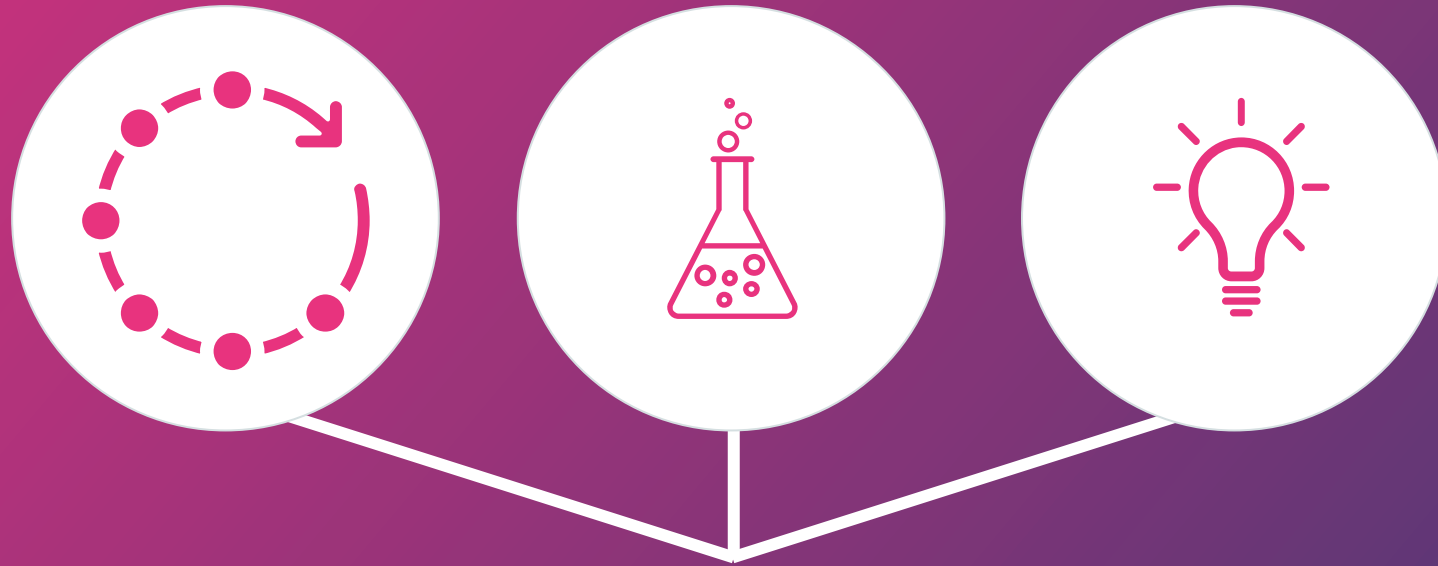


Karl Popper  
defined science as

**BEING TESTABLE  
AND FALSIFIABLE**



What do **Continuous Integration**, **Science**,  
**and Entrepreneurship** all have in common?



What do **Continuous Integration, Science, and Entrepreneurship** all have in common?



They **each** require **comparing** an **idealized conception** to the often **brutal truth of objective reality**

# HOW DOES CONTINUOUS INTEGRATION FIT INTO THE CLOUD NATIVE JOURNEY?

# Cloud Native Trail Map

Trail Map: [l.cncf.io](https://l.cncf.io)



## CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape [l.cncf.io](https://l.cncf.io) has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

### HELP ALONG THE WAY

#### A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer [cncf.io/training](https://cncf.io/training)

#### B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider [cncf.io/kcsp](https://cncf.io/kcsp)

#### C. Join CNCF's End User Community

For companies that don't offer cloud native services externally [cncf.io/enduser](https://cncf.io/enduser)

### WHAT IS CLOUD NATIVE?

Cloud-native technologies, such as containers and microservices, empower organizations to develop and deploy scalable, agile applications and services in dynamic, distributed environments. By taking into account these characteristics, such systems are designed to be resilient, elastic, and loosely coupled, via manageable abstractions and declarative APIs, thereby enabling effective, reliable automation. This allows engineers to observe the applications and to safely make impactful changes, and results in processes and workflows that fully take advantage of these environments and minimize toil.

The Cloud Native Computing Foundation seeks to drive adoption of these techniques by fostering an ecosystem of open-source, vendor-neutral projects that align with these objectives, and which are portable to public, private, and hybrid clouds. We democratize the state-of-the-art patterns and practices to ensure innovations remain open and accessible for everyone.

[l.cncf.io](https://l.cncf.io)

v20180604



### 1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

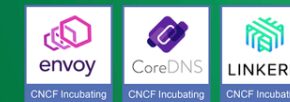
### 3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: [cncf.io/ckd](https://cncf.io/ckd)
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application



### 5. SERVICE MESH AND DISCOVERY

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



### 7. DISTRIBUTED DATABASE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding.



### 9. CONTAINER RUNTIME

You can use alternative container runtimes. The most common, all of which are OCI-compliant, are containerd, rkt and CRI-O.



### 2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing

### 4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



### 6. NETWORKING

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net.



### 8. MESSAGING

When you need higher performance than JSON-RPC, consider using gRPC. NATS is publish/subscribe message-oriented middleware.

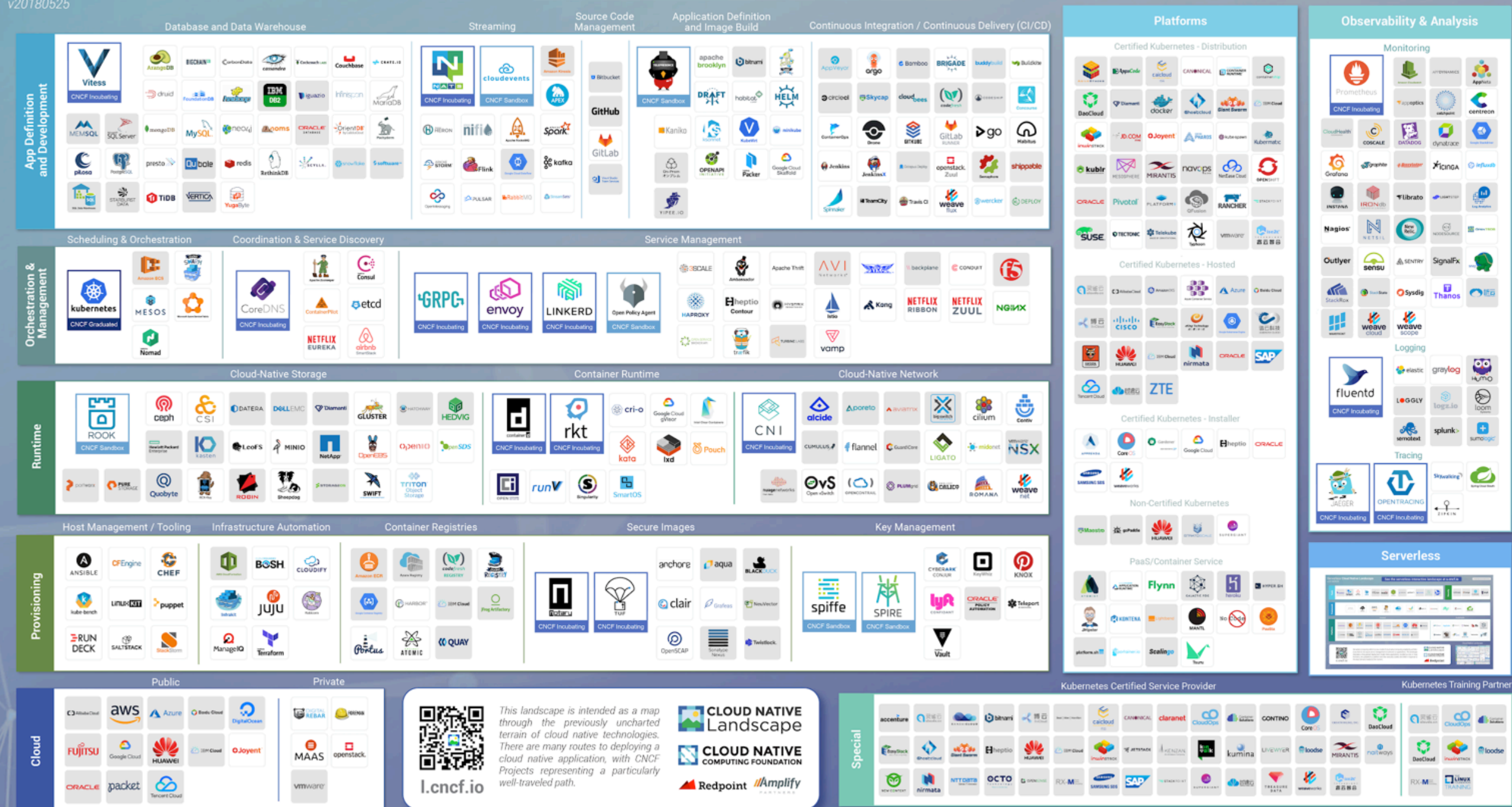


### 10. SOFTWARE DISTRIBUTION

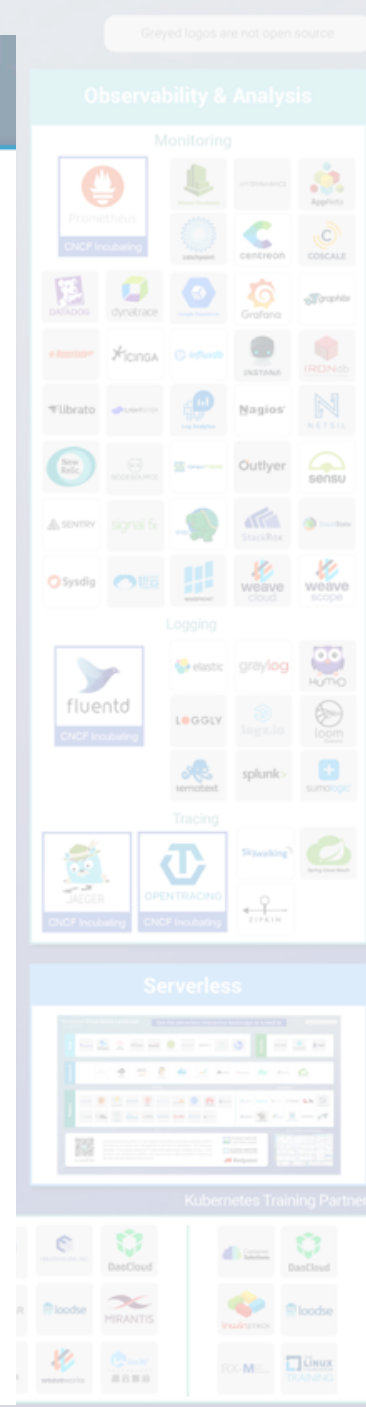
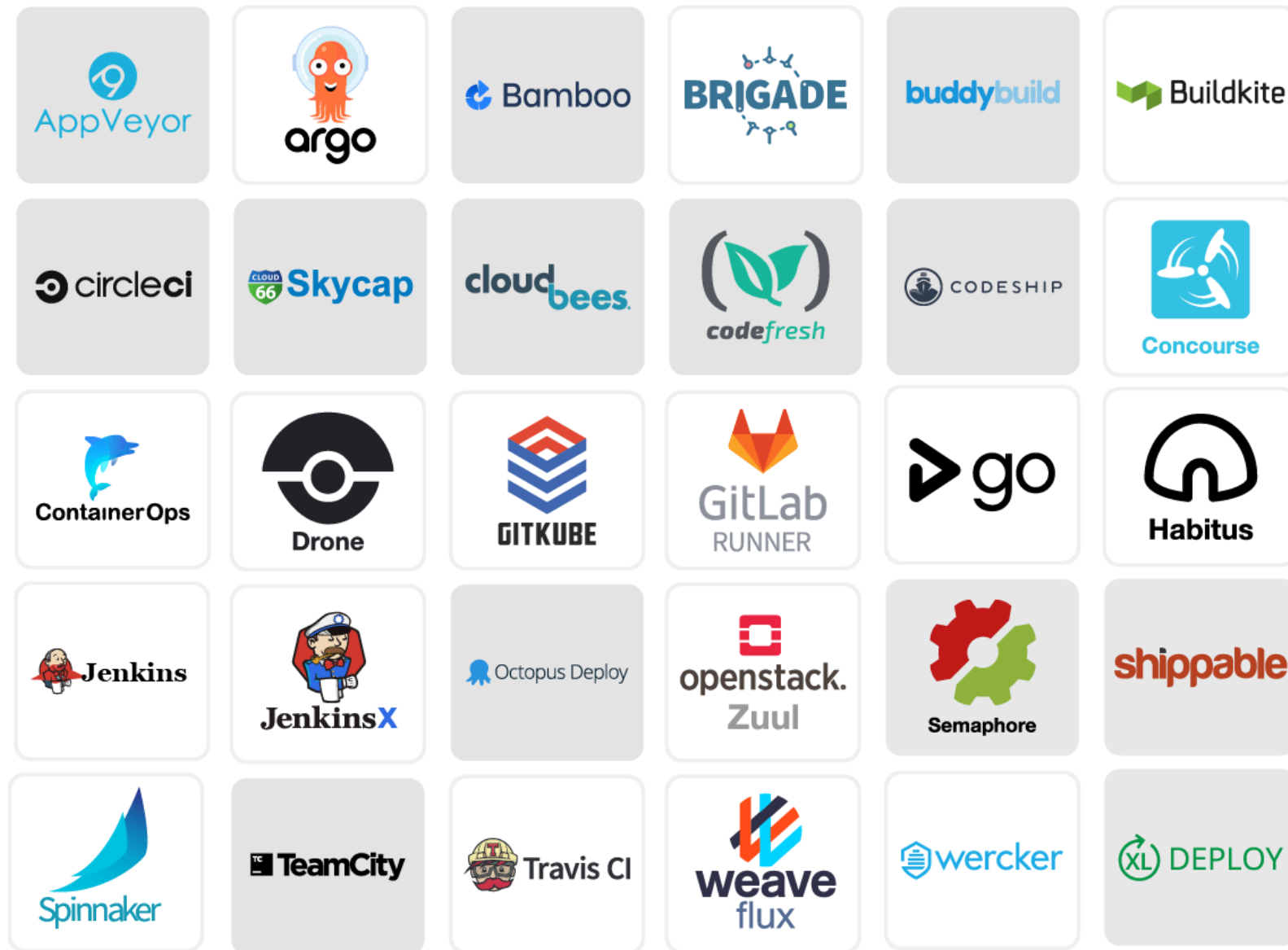
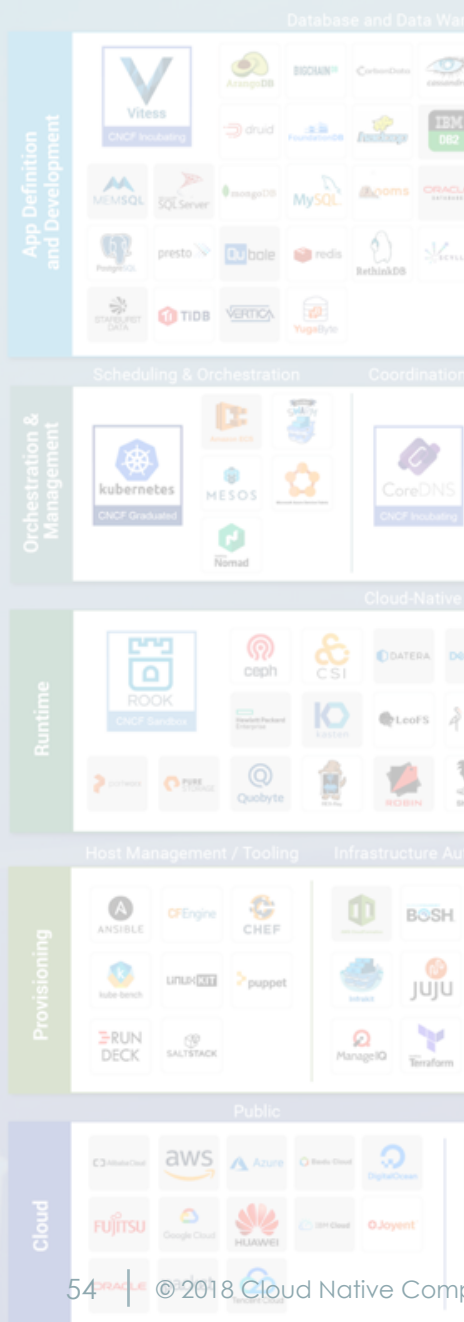
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.







# Continuous Integration / Continuous Delivery (CI/CD)







PLEASE TRY THE  
INTERACTIVE  
LANDSCAPE NOW:

**[l.cncf.io](http://l.cncf.io)**