Evolving Cloud Native Landscape

Chris Aniszczyk, CTO (@cra)

https://cncf.io

https://linuxfoundation.org

CLOUD NATIVE
COMPUTING FOUNDATION

Agenda

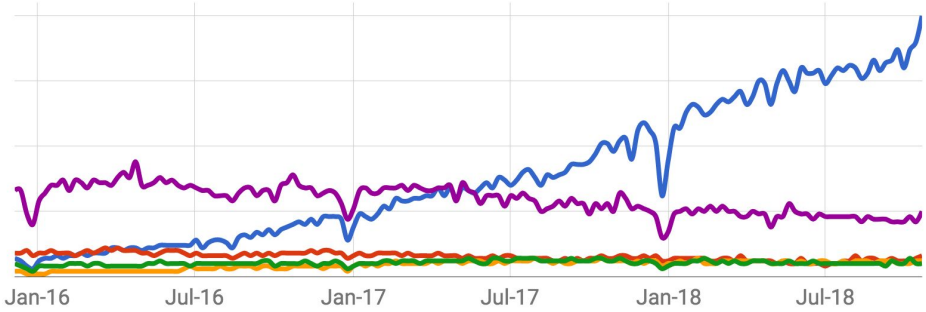
- Introduction
- What is CNCF?
- From Virtualization to Containers and Cloud Native
- Evolving Cloud Native Landscape
- VNFs to CNFs
- Q&A

Hi, I'm Chris Aniszczyk (@cra)

- CTO/COO, Cloud Native Computing Foundation (CNCF)
- Executive Director, Open Container Initiative (OCI)
- VP, Developer Relations, Linux Foundation (LF)

- *In a previous life…*
  - *Director of Open Source (@Twitter) / Sr. Eng Manager*
  - *Co-Founder of the TODO Group*
  - *Co-Founder of EclipseSource (via Code9)*
  - *Open Source Committer (Gentoo, Fedora, etc)*
  - *Principal Software Engineer, Red Hat*
  - *Senior Software Engineer, IBM*

Been a crazy ~3 years for me… Google Trends + CNCF



Kubernetes OpenStack Mesos Docker
Swarm Cloud Foundry

Cloud Native Computing Foundation (CNCF)

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

Graduated

Orchestration

Monitoring

Incubating

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

Package Management

Sandbox

Storage

Identity Spec

Identity

Policy

Serverless

Tooling

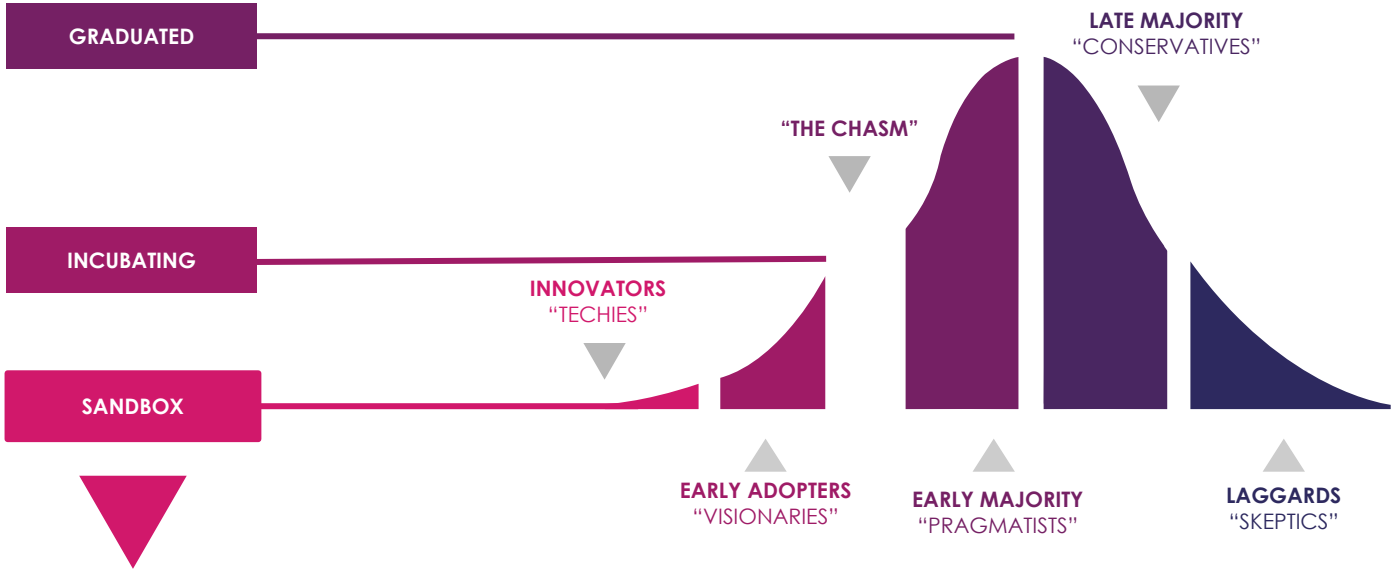
Registry

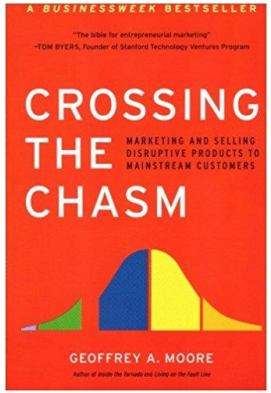
Metrics Spec

Key/Value Store

- Platinum members:

CNCF Project Maturities





SANDBOX

Identity Spec	Identity	Policy	Serverless	Tooling	Registry	Metrics Spec	Distributed K/V	Monitoring	Packaging Spec	Container Security
spiffe	SPIRE	Open Policy Agent	cloudevents	TELEPRESENCE	HARBOR	OPENMETRICS	KV	cortex	Buildpacks.io	Falco

300+ Members and Growing

Platinum Members

Alibaba Cloud, aws, Azure, cisco, Dell Technologies, docker, Fujitsu, Google Cloud, Huawei, IBM Cloud, intel, JD.com, Mesosphere, Oracle, Pivotal, redhat, Samsung SDS, SAP, vmware

Gold Members

AT&T, Baidu 百度, DigitalOcean, JFrog, Orchestrating a brighter world NEC, NetApp, salesforce, sumologic, SUSE, Tencent Cloud, ZTE

End User Members

Atlassian, Bloomberg, Capital One, Comcast, ebay, DENSO, DiDi, GitHub, Goldman Sachs, indeed, Intuit, JD.com, Morgan Stanley, NAIC, NCSOFT, Pinterest, salesforce, Spotify, Squarespace, twilio, twitter, University of Michigan, Wikimedia Foundation, Yahoo! Japan

End User Supporters

adidas, box, cruise, FORM3, Kuelap, Layer, Nasdaq, The New York Times, Olark, Pusher, reddit, ricardo.ch, SAP Concur, shopify, showmax, Spredfast, Steelhouse, stix, textkernel, THREDUP, ticketmaster, Two Sigma, Werkspot, woorank, workday, WP Engine, zalando, zendesk

Academic/Nonprofit

CableLabs, Cloud Foundry, Golden Gate University, Internet, NAIC, NIPR, SEL, University of Michigan, Wikimedia Foundation

300+ Members and Growing (Silver 1)

300+ Members and Growing (Silver 2)

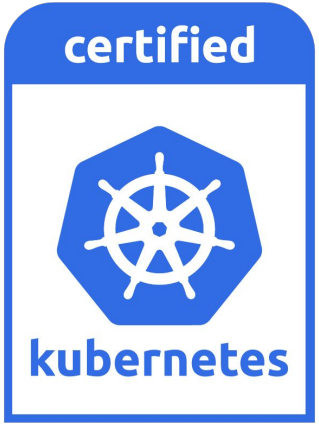
67 Companies in the End User Community

 |

Plus 7 non-public members

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 1.10 & 1.11
  - https://www.cncf.io/certification/software-conformance/



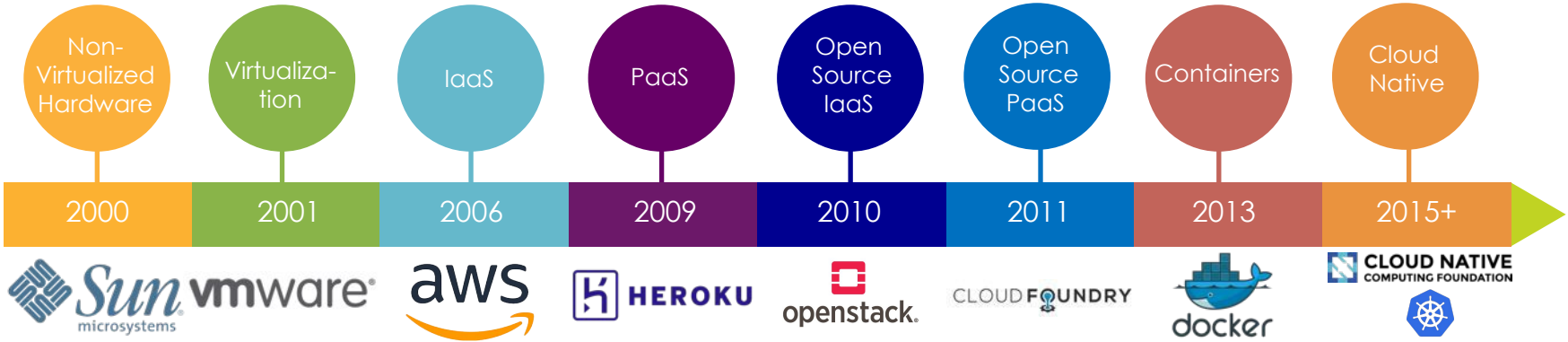
55 Certified Kubernetes Partners

CNCF? From Virtualization to Cloud Native

CLOUD NATIVE COMPUTING FOUNDATION

kubernetes

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



 |

Why You Should Host Your Project at CNCF

- A neutral home increases contributions
- Endorsement by CNCF's Technical Oversight Committee
- Engagement with End User and Service Provider Communities
- Full-time press and analyst relations teams
- Tens of thousands of dollars per year in documentation, case study and other support services
- Maintain your committers and define your own governance, as long as its neutral
- Full-time staff eager to assist
- World-class events team, track at KubeCon + CloudNativeCon around the world, and custom events for your project
- Worldwide meetup groups
- Cloud resources for CI and scale testing

Cloud Native Landscape

2018-11-02T15:40:41Z 98f8067

See the interactive landscape at l.cncf.io

Greyed logos are not open source

App Definition and Development

Database

Streaming & Messaging

Application Definition & Image Build

Continuous Integration & Delivery

Orchestration & Management

Scheduling & Orchestration

Coordination & Service Discovery

Remote Procedure Call

Service Proxy

API Gateway

Service Mesh

Runtime

Cloud-Native Storage

Container Runtime

Cloud-Native Network

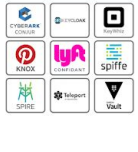
Provisioning

Automation & Configuration

Container Registry

Security & Compliance

Key Management



Platform

Certified Kubernetes - Distribution

Certified Kubernetes - Hosted

Certified Kubernetes - Installer

Non-Certified Kubernetes

PaaS/Container Service

Observability and Analysis

Monitoring

Logging

Tracing

Chaos Engineering

Serverless

Cloud

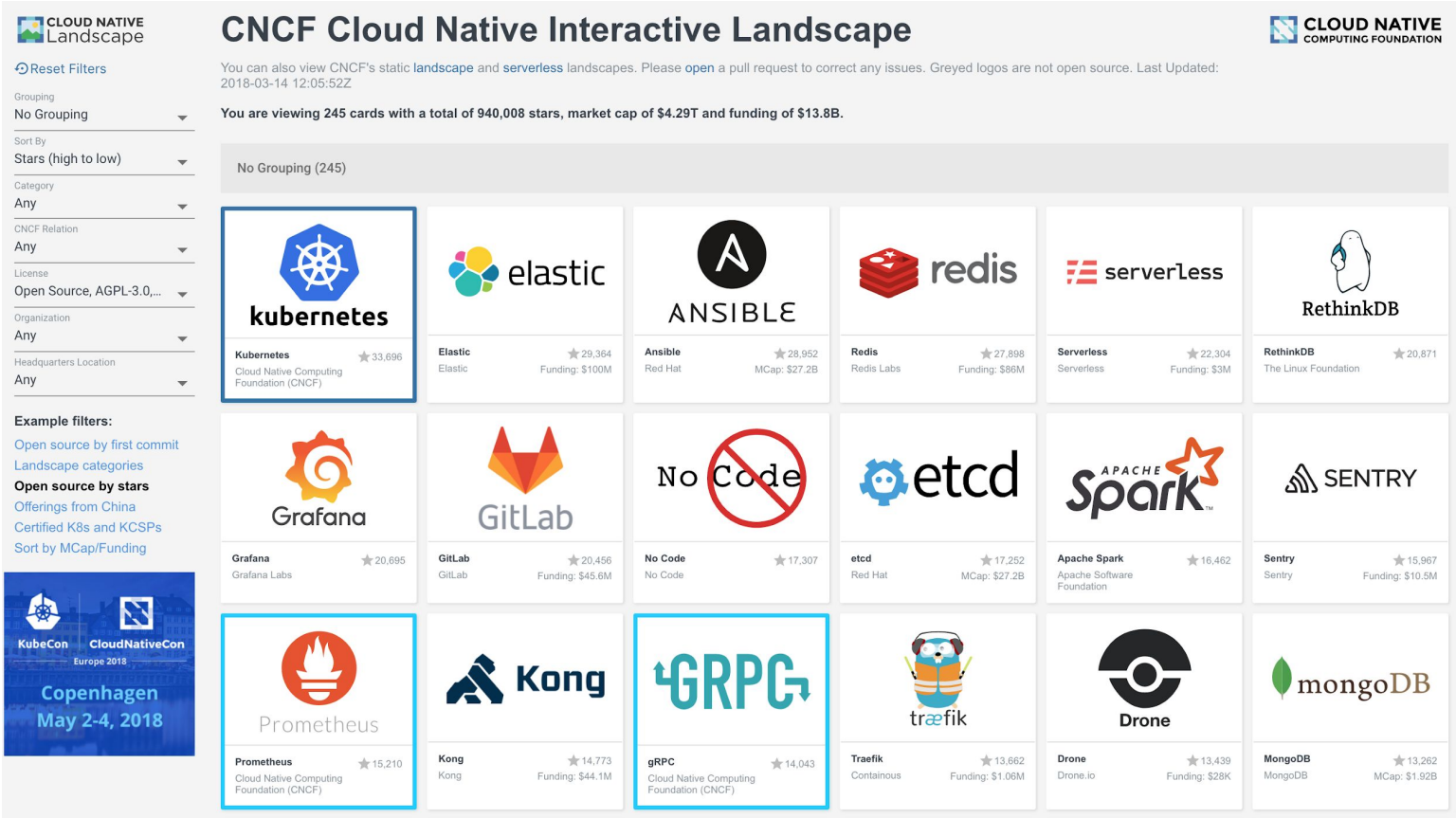
Public

Special

Kubernetes Certified Service Provider

Kubernetes Training Partner

Try it now at https://l.cncf.io



 |

Cloud Native Trail Map

Trail Map: l.cncf.io

CLOUD NATIVE COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape 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

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider

cncf.io/kcsp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally

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

v20180425

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

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

3. ORCHESTRATION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer
- cncf.io/ck

kubernetes — CNCF Graduated

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

Prometheus — CNCF Incubating
fluentd — CNCF Incubating
OPENTRACING — CNCF Incubating
JAEGER — CNCF Incubating

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

envoy — CNCF Incubating
CoreDNS — CNCF Incubating
LINKERD — CNCF Incubating

6. NETWORKING

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

CNI — CNCF Incubating

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.

Vitess — CNCF Incubating

8. MESSAGING

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

gRPC — CNCF Incubating
NATS — CNCF Incubating

9. CONTAINER RUNTIME

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

containerd — CNCF Incubating
rkt — CNCF Incubating

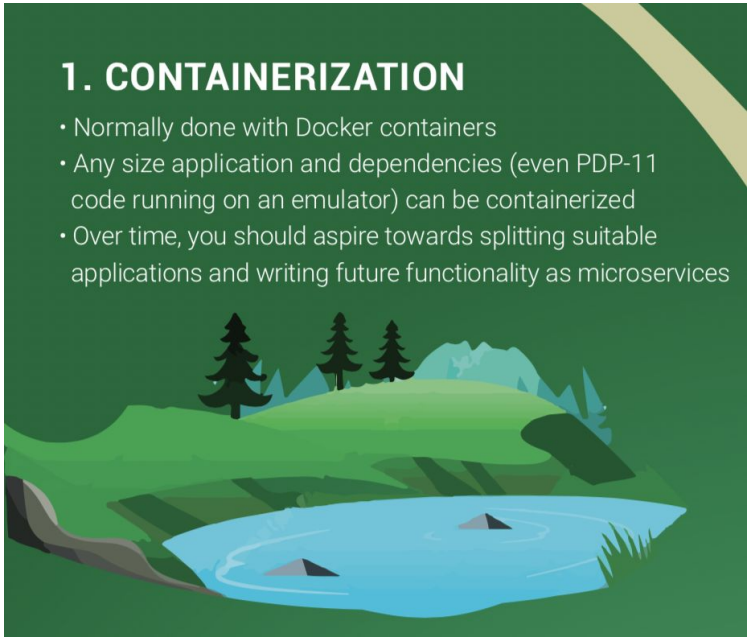
10. SOFTWARE DISTRIBUTION

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

Notary — CNCF Incubating
TUF — CNCF Incubating

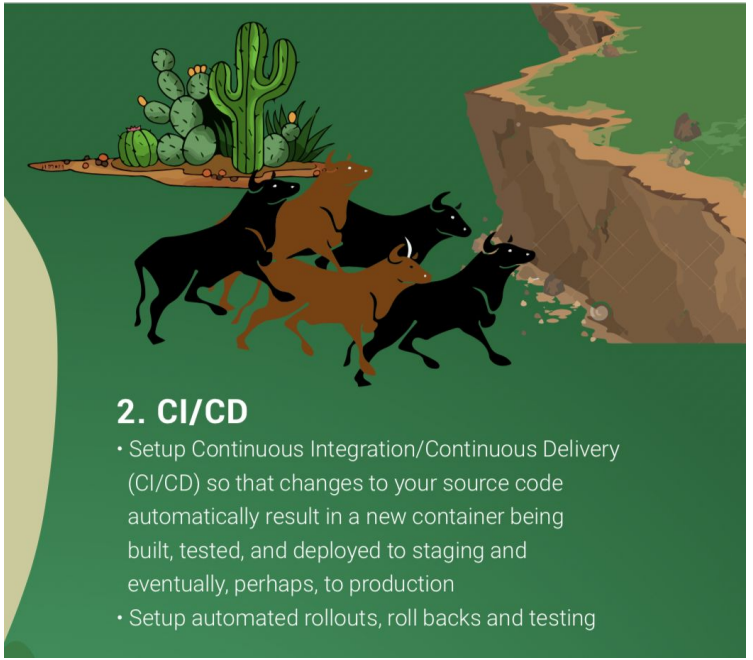
Step 1: Containerization

- You can't do cloud native without containerizing your application!
- Any size application will do!
- Containerize your legacy: Ticketmaster has a PDP-11 emulator running inside a container!



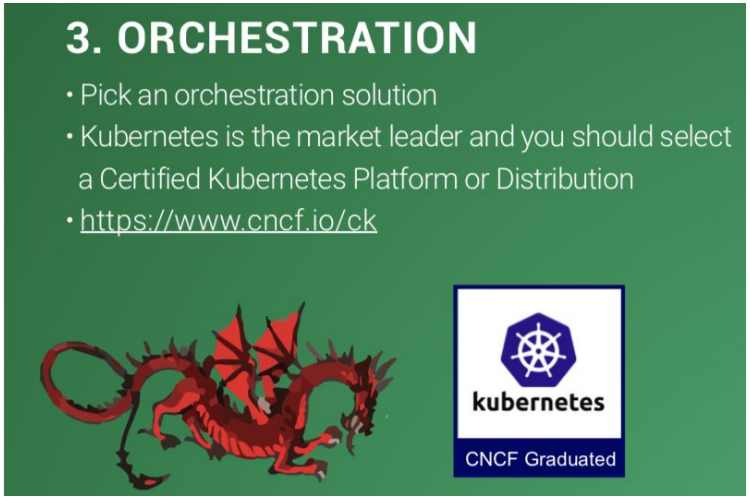
Step 2: CI/CD

- Setup Continuous Integration and 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
- CNCF doesn't offer a CI/CD solution but many options out there: Jenkins, GitLab CI, Spinnaker, etc



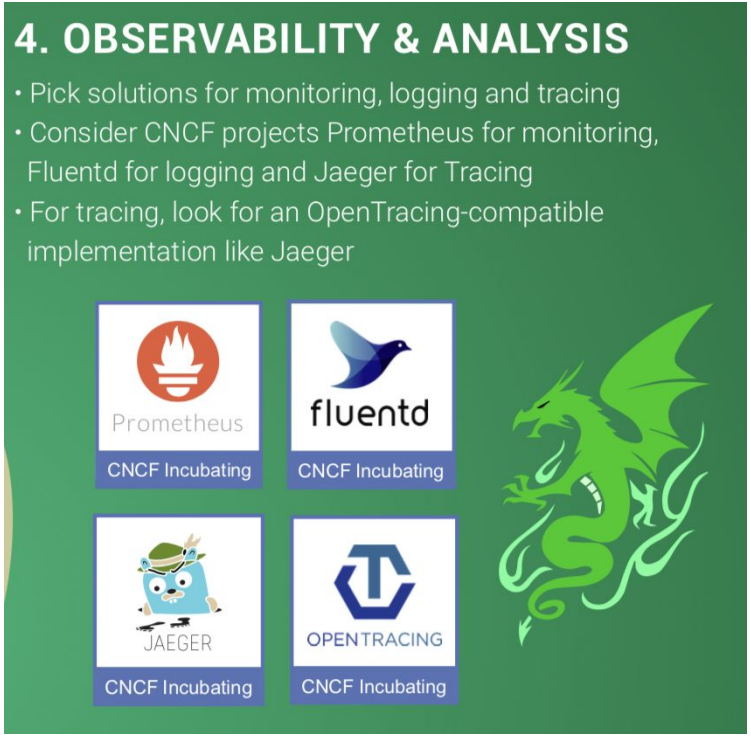
Step 3: Orchestration

- Pick an orchestration solution
- Kubernetes is the market leader and you should select a Certified Kubernetes Platform or Distribution: https://www.cncf.io/ck
- However, other orchestration solutions exist: Mesos, Pouch, Nomad, Docker Swarm



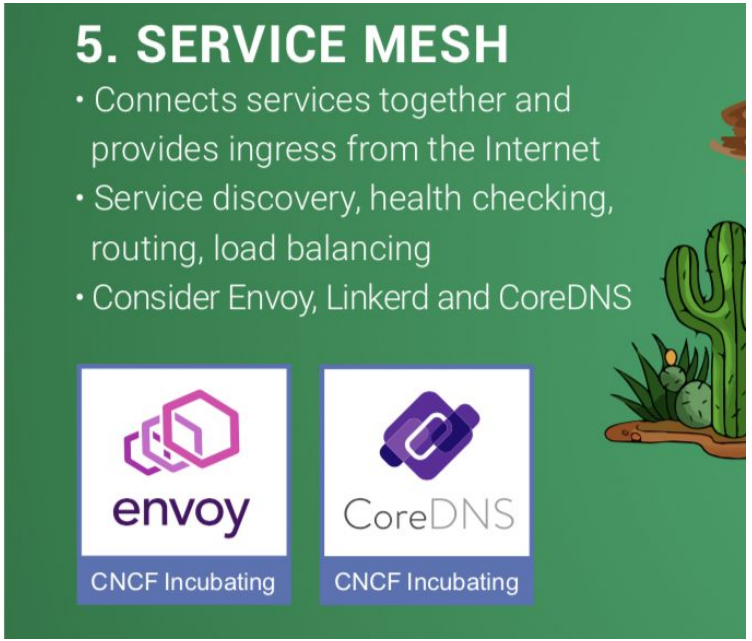
Step 4: Observability

- Observability is table stakes for cloud native at scale: Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus (prometheus.io) for monitoring
- Fluentd (fluentd.io) for logging
- For tracing, look for an OpenTracing-compatible (opentracing.io) implementation like Jaeger (jaegertracing.io)



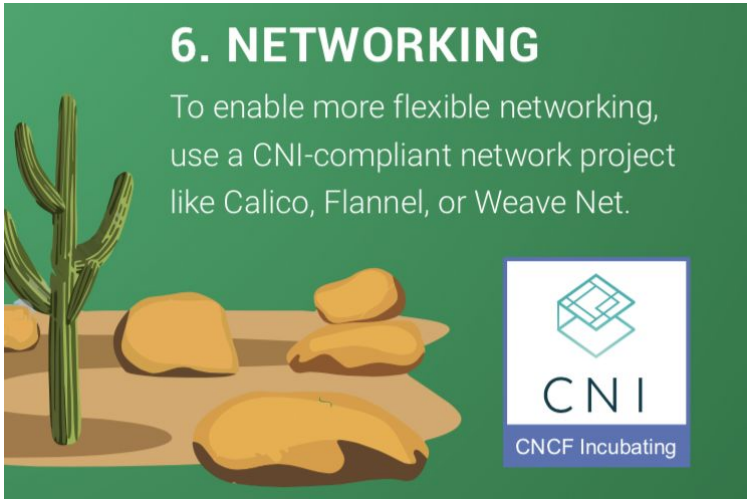
Step 5: Service Mesh

- Connects services together and monitors ingress from the Internet
- Service discovery, health checking, routing, load balancing
- Consider Envoy (envoyproxy.io) , Linkerd (linkerd.io) and CoreDNS (coredns.io)



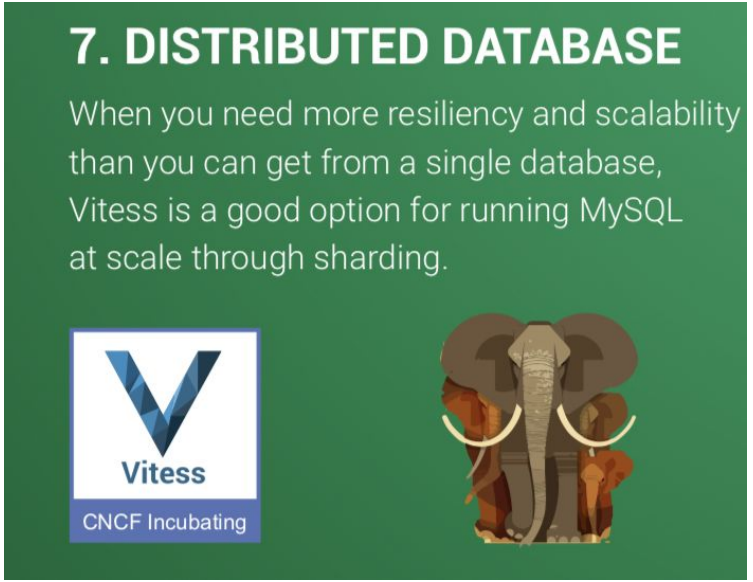
Step 6: Networking

- To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net
- https://github.com/containernetworking/cni



Step 7: Distributed Database

- When you need more resiliency and scalability than you can get from a single database, Vitess (vitess.io) is a good option for running MySQL at scale through sharding.
- TiKV is a distributed key value store: https://github.com/tikv/tikv

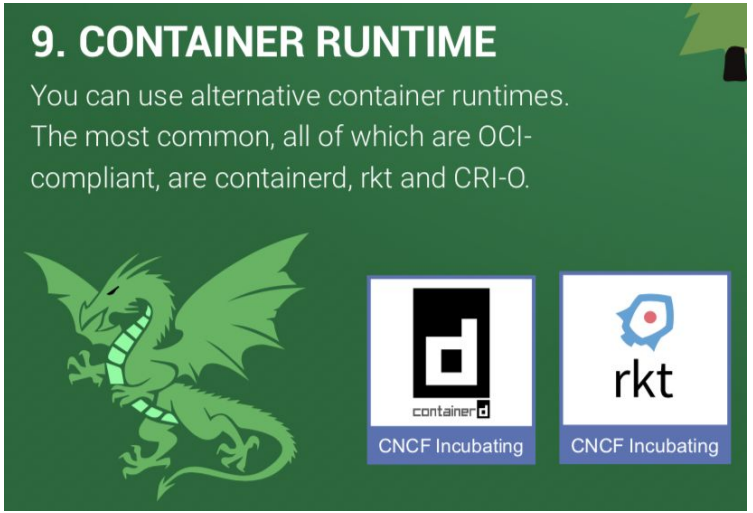


Step 8: Messaging

- When you need higher performance than JSON-REST, consider using gRPC: grpc.io
- NATS is a publish/subscribe message-oriented middleware: nats.io

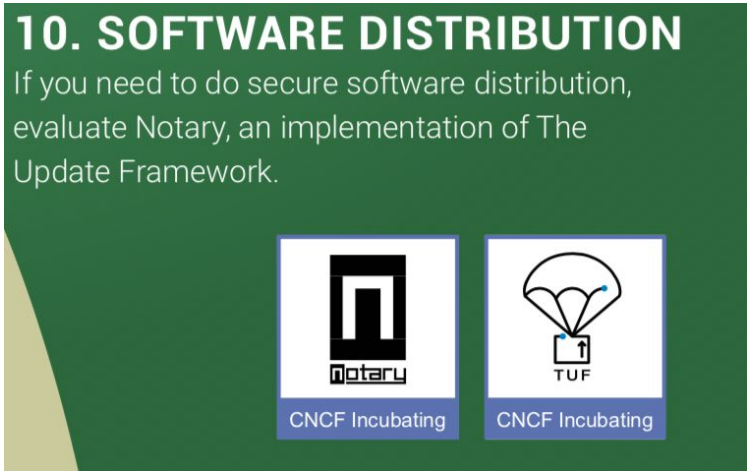
Step 9: Container Runtimes

- You can use alternative container runtimes. The most common, all of which are OCI- compliant, are containerd (containerd.io), rkt (rkt.io)and CRI-O (cri-o.io).
- Alibaba's PouchContainer runtime also makes for an option:



Step 10: Software Distribution

- If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.
- TUF is designed to work as part of a larger software distribution framework and provides resilience to key or server compromises. Using a variety of cryptographic keys for content signing and verification, TUF allows security to remain as strong as is practical against a variety of different classes of attacks. TUF is used in production by Docker, LEAP, App Container, Flynn, OTAInfo, ATS Solutions, and VMware.
- Harbor is a container registry

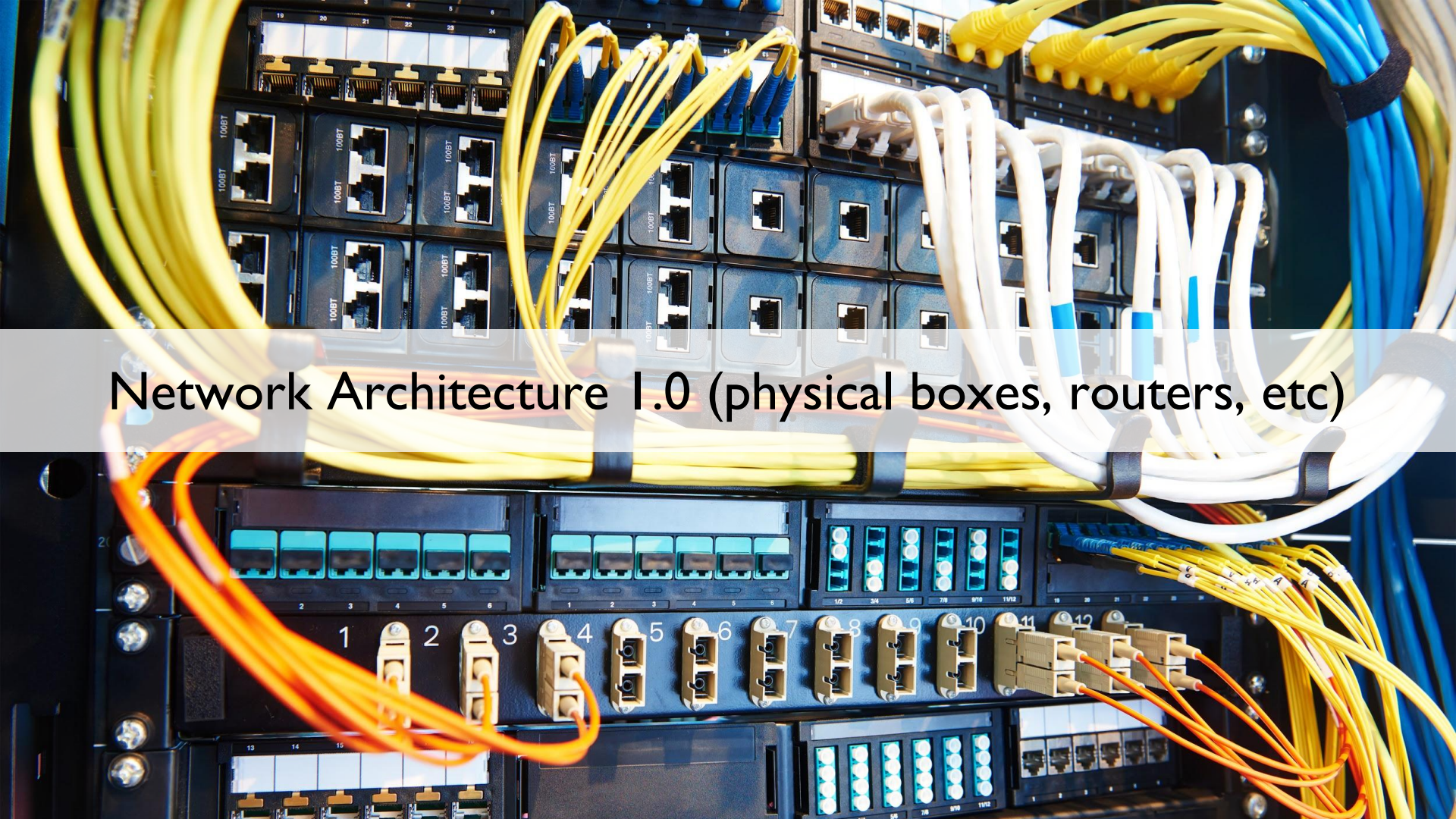


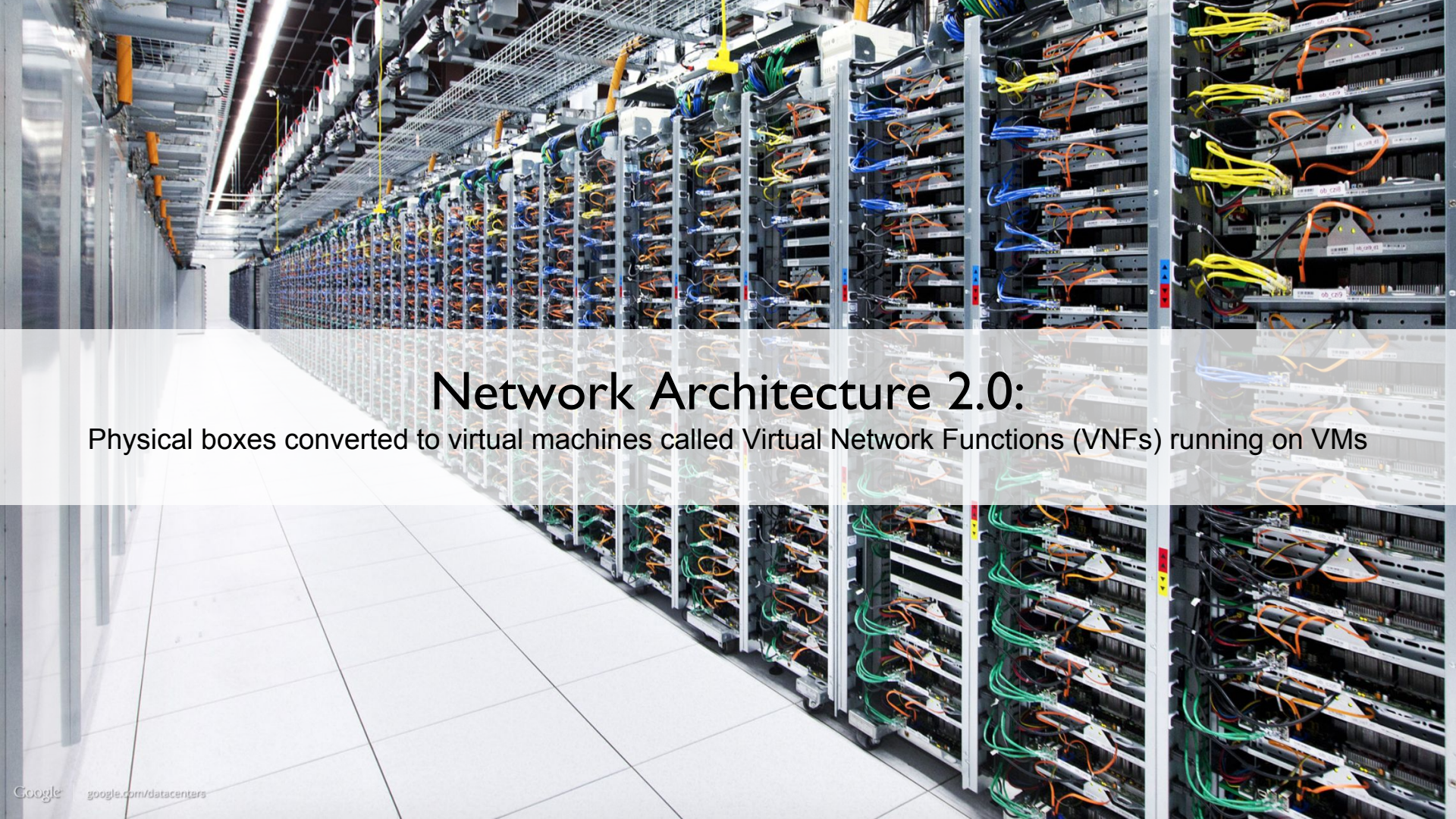
Cloud Native Trail Map Summary

- There are many paths to cloud native and the CNCF is doing our best to guide companies on their journey

- The landscape and trail map are open source, please contribute issues and improvements: https://github.com/cncf/landscape

VNFs to CNFs


Network Architecture 1.0 (physical boxes, routers, etc)

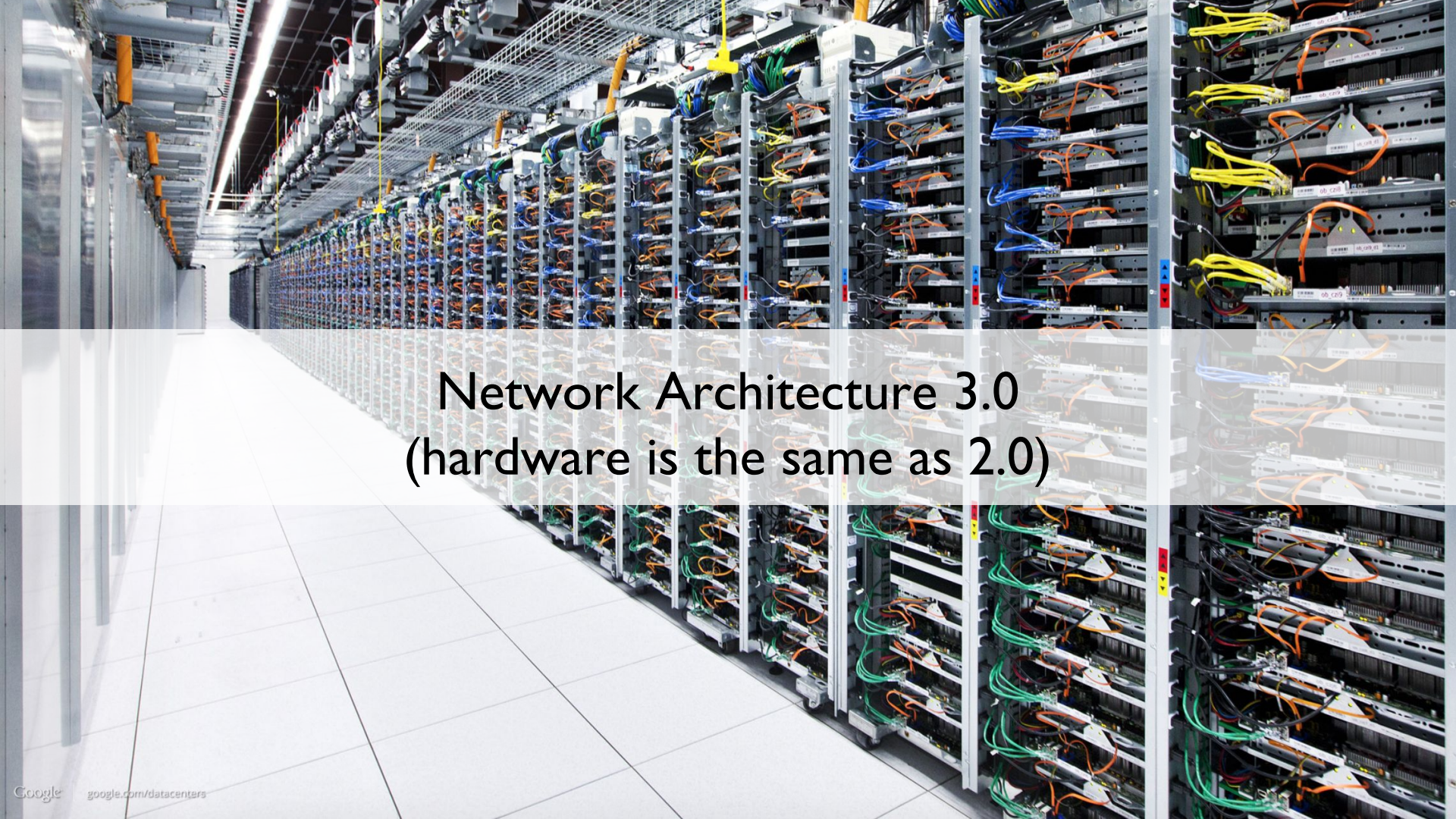


Network Architecture 2.0:

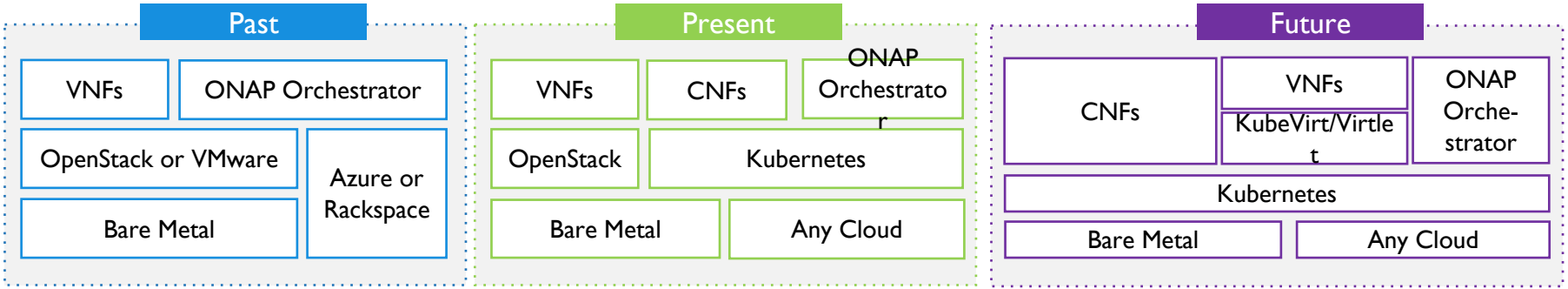
Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMs

Network Architecture Evolution

- **1.0:** Separate physical boxes for each component (e.g., routers, switches, firewalls)
- **2.0:** Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack
- **3.0:** Cloud-native Network Functions (CNFs) running on Kubernetes on public, private, or hybrid clouds


Network Architecture 3.0
(hardware is the same as 2.0)
Google
google.com/datacenters

Evolving from VNFs to CNFs



- ONAP Amsterdam (Past) runs on OpenStack, VMware, Azure or Rackspace
- ONAP Casablanca (Present) runs on Kubernetes and so works on any public, private or hybrid cloud
- Virtual Network Functions (VNFs) are virtual machines that run on OpenStack or VMware, or can be run on K8s via KubeVirt or Virtlet

Benefits of CNFs?

- Cost savings (with public, private, and hybrid clouds)
- Development velocity (cloud native)
- Resilience (to failures of individual CNFs, machines, and even DCs)

Challenges of evolving CNFs to VNFs

- Moving from network functionality from physical hardware to encapsulating the software in a virtual machine (P2V) is generally easier than containerizing the software (P2C or V2C)
- Many network function virtualization VMs rely on kernel hacks or otherwise do not restrict themselves to just the stable Linux kernel userspace ABI
  - They also often need to use DPDK or SR-IOV to achieve sufficient performance
- Containers provide nearly direct access to the hardware with little or no virtualization overhead
  - But they expect containerized applications to use the stable userspace Linux kernel ABI, not to bypass it

CNFs Discussion Areas

- The strength of no longer being locked into specific OSs
  - Any version of Linux >3.10 is acceptable
- Multi-interface pods vs. Network Service Mesh
- Complete parity for IPv6 functionality and dual-stack support in K8s
- Security, and specifically recommendations from Google and Jess that come into play when hosting untrusted, user-provided code
  - Possible use of isolation layers such as gVisor or Kata
- Scheduling container workloads with network-related hardware constraints (similar to what's been done for GPUs)
  - Network-specific functionality like QOS
-

CNFs Demos

- VNFs vs. CNFs
  - Working on a demo of boot-time and throughput of VNFs on OpenStack vs. CNFs on Kubernetes, where the networking code and underlying hardware is identical
  - Will deliver opens source installers and Helm charts
- Cloud-native Customer Premises Equipment (CCPE) Project
  - Modify the ONAP vCPE use case and VNF deployment to show VNF vs. CNF deployments of chained network functions

Getting Involved / Q&A

CLOUD NATIVE
COMPUTING FOUNDATION

Contributing to CNCF for Networking and Telcos

- See the landscape
  - CNI: https://github.com/containernetworking/cni
  - Envoy: https://github.com/envoyproxy/envoy
  - Linkerd: https://github.com/linkerd/linkerd
- Kubernetes IoT/Edge WG
  - https://github.com/kubernetes/community/tree/master/wg-iot-edge
- Cloud-native Network Functions (CNFs)
  - #cnfs on slack.cncf.io
  - https://github.com/cncf/cnfs

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

KubeCon + CloudNativeCon

- China
  - Shanghai: November 13-15, 2018
  - Sponsorships open
- North America
  - Seattle: December 10-13, 2018
  - Sponsorships open
- 2019
  - Barcelona: May 20-23, 2019
  - Shanghai: June 26-28, 2019
  - San Diego: November 18-21, 2019

Thank you! Q&A?

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

Appendix

A Brief History of the Cloud

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

 |

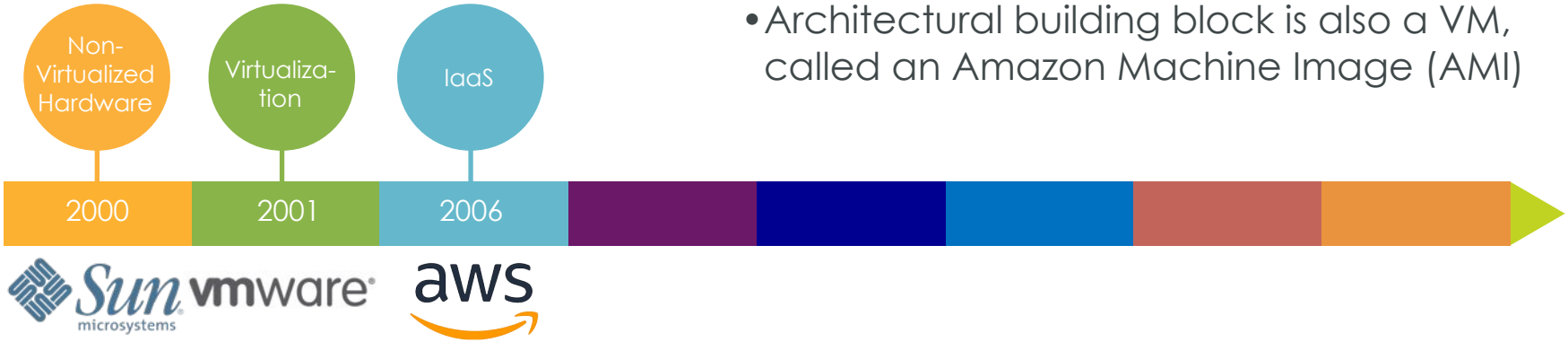
Virtualization: VMWare (2001)

vmware®

- 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

Sun microsystems vmware®

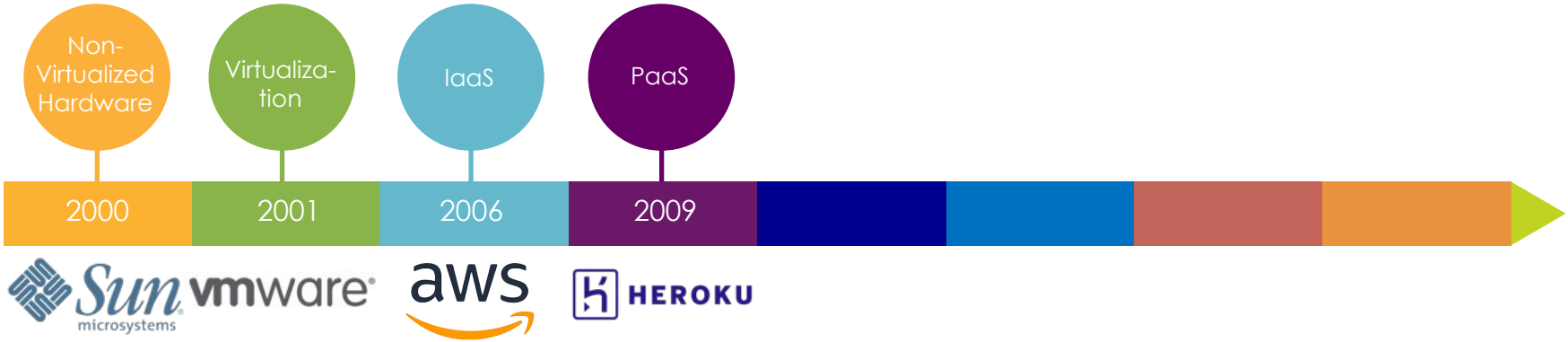
IaaS: AWS (2006)



- 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)

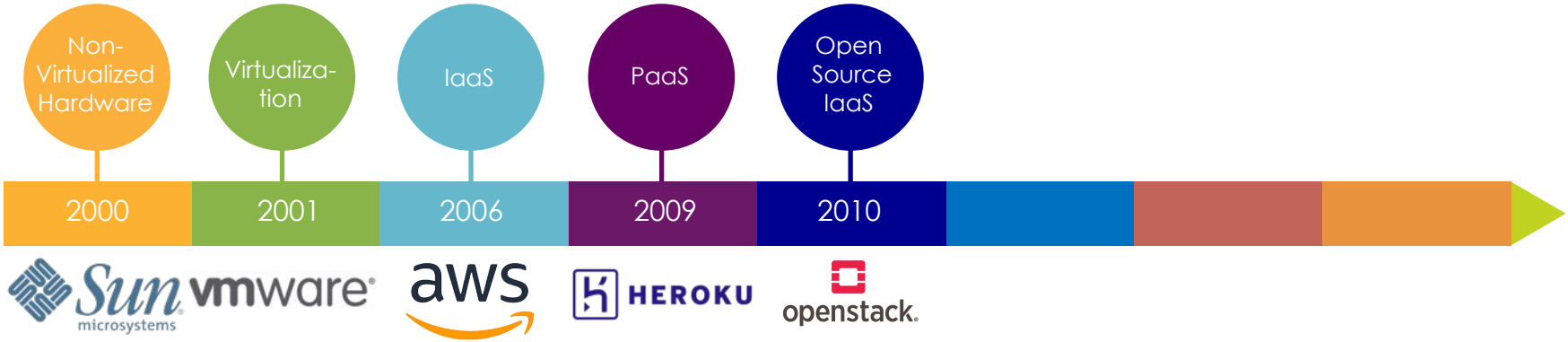
PaaS: Heroku (2009)

- 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)

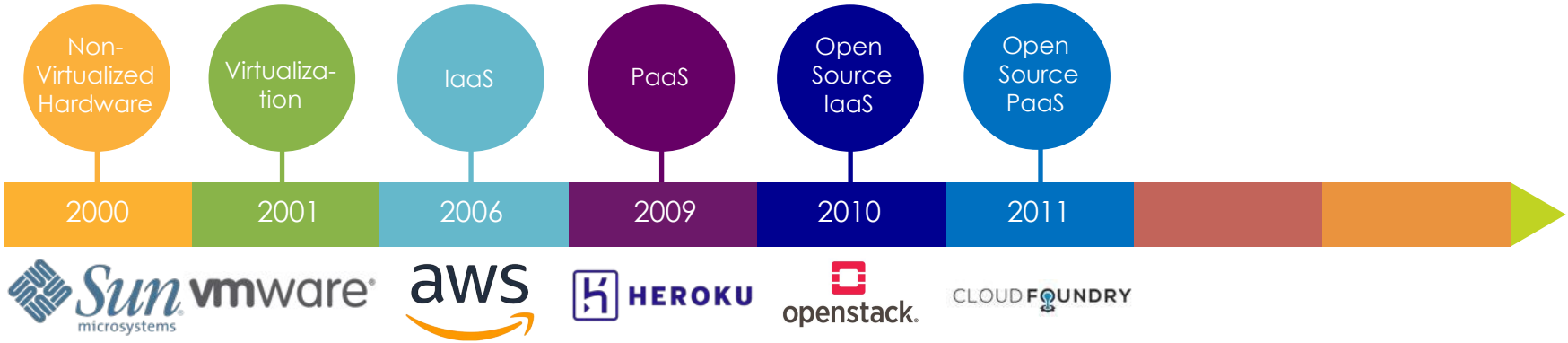
- OpenStack brings together an extraordinarily diverse group of vendors to create an open source Infrastructure-as-a-Service (IaaS)
- Competes with AWS and VMWare
- Building block remains a VM



Open Source PaaS: Cloud Foundry (2011)

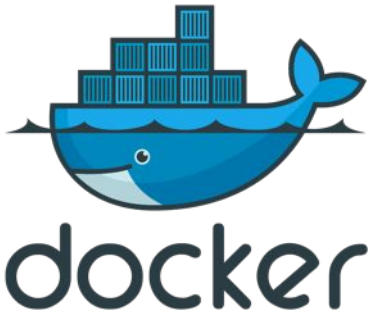
CLOUD FOUNDRY

- 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

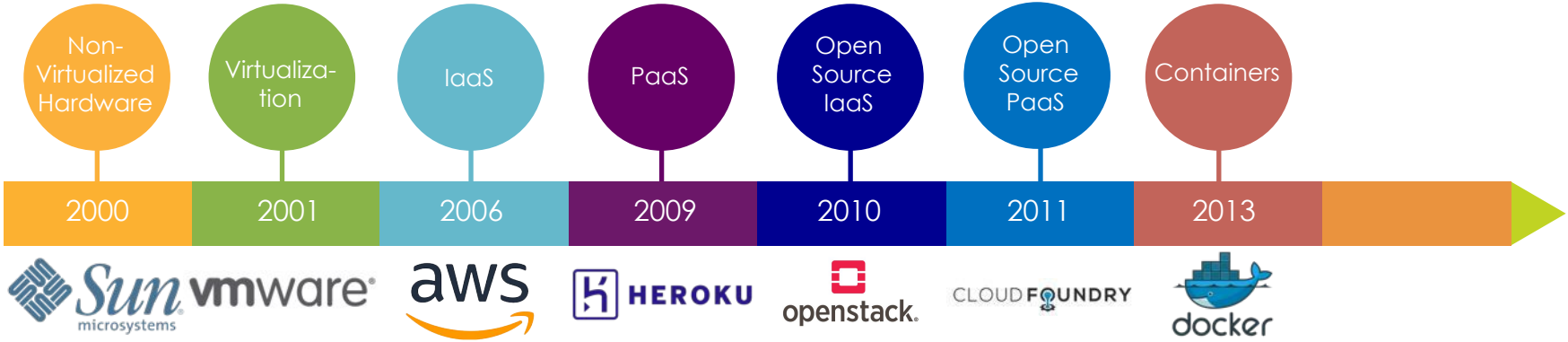


 |

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



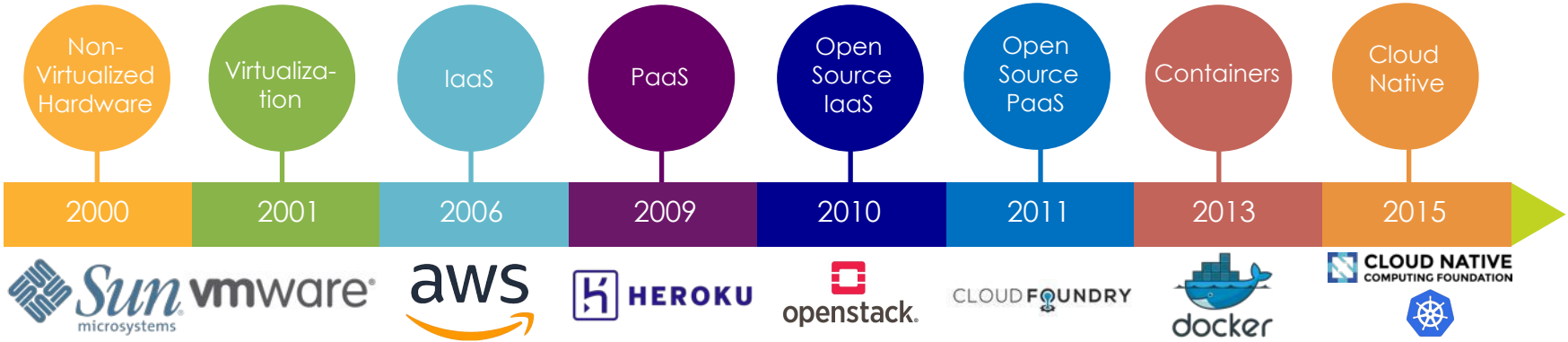
 |

Cloud Native: CNCF (2015)

CLOUD NATIVE COMPUTING FOUNDATION

kubernetes

- Cloud native computing uses an open source software stack to:
  - segment applications into *microservices*,
  - package each part into its own *container*
  - and dynamically *orchestrate* 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

 |

What About PaaS?

- OpenShift, Huawei CCE, Deis, and Apprenda are examples of PaaS's built on top of cloud native platforms
- Many new applications start out as 12-factor apps deployable on a PaaS
  - In time they sometimes outgrow PaaS
  - And some apps never fit a PaaS model
- PaaS on top of cloud native supports both