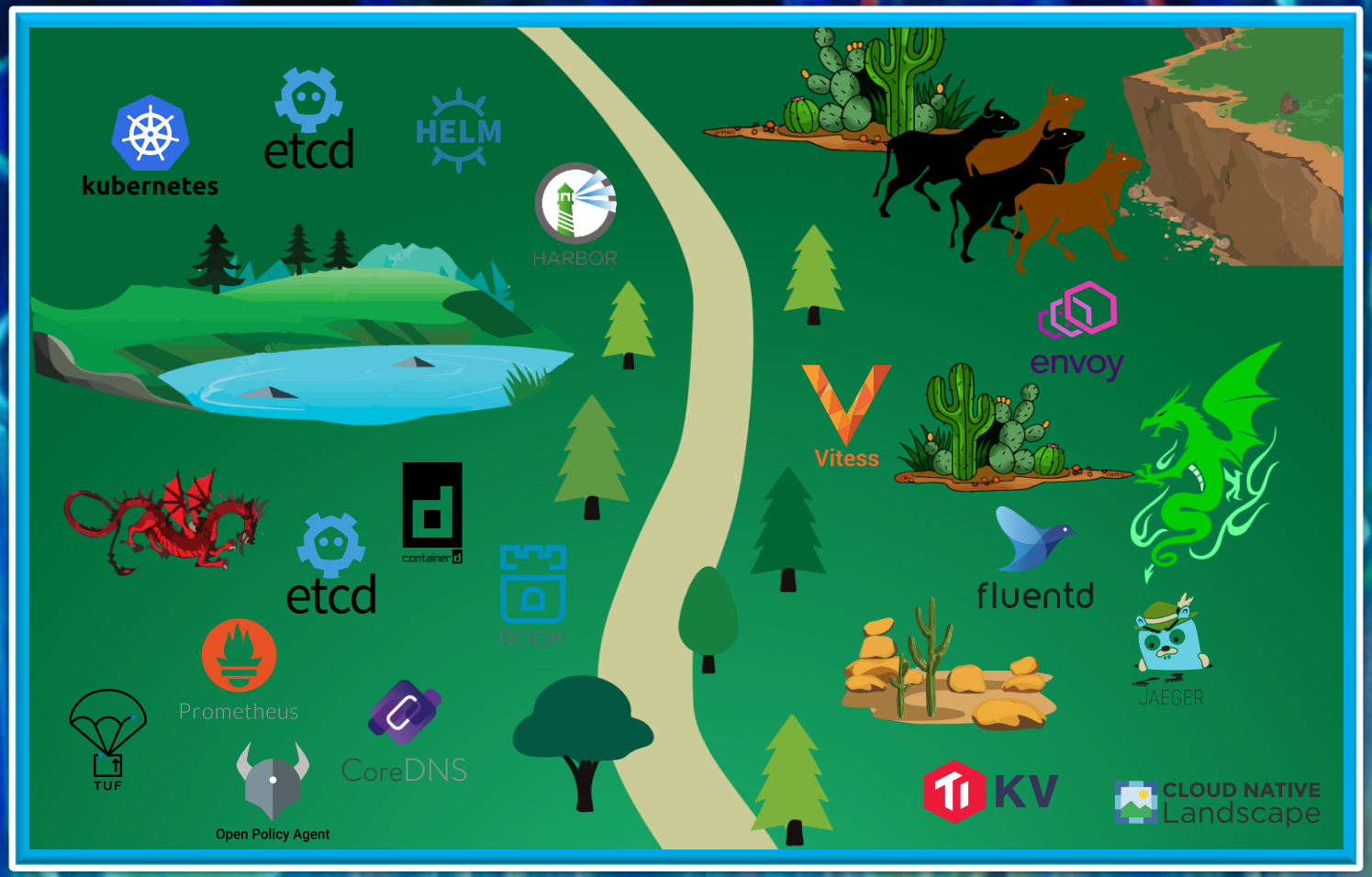


Cloud Native: OSS technology trends & top scenarios

Ian Choi (최영락)

Contents

0. Preface

Introduction – Ian Choi

- **Developer Audience Product Marketing Manager at Microsoft (from Nov 2018)**
- **Developer community activities**
  - ✓ Microsoft Ex-MVP: Cloud & Datacenter Management (July 2016 – Oct 2018)
  - ✓ OpenStack (Internationalization, Election official, Korea user group organizer)
  - ✓ K8s – Docs L10n: Korean reviewer, Facebook Developer Circles - Seoul Lead

At the end, we also had a chance to take picture with a young talented Korean programmer who work days and nights in developing the world. We also made friends from all around the world who are contributing to the advance of humanity. Microsoft Future Now 2019 was a valuable experience to us. We also would like to thank School of Business & Economics, School of Engineering and Tan Tao University for giving us an opportunity like this.

The journey to the cloud with open sources

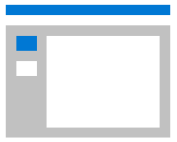
Infrastructure

"What your application runs on"

Data

"What your application works with"

Code



"What your application does"

Migrate • Innovate

Unified Management • Security • Governance • Tools • DevOps

Software Development • Open Sources • Enterprise collaboration

Q: How "Cloud Native" empowers you?

1. Cloud computing and enabling technologies

P.S. Cloud vs. Cloud Native?

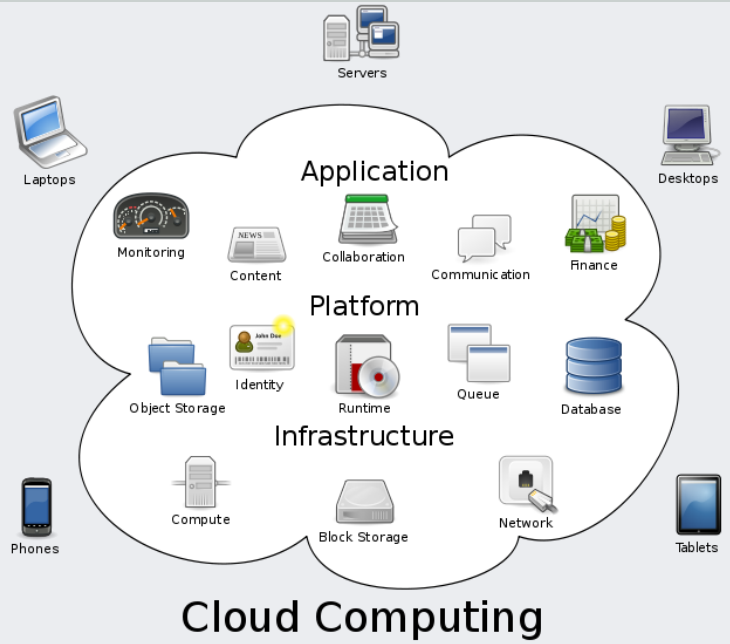
Defining cloud computing

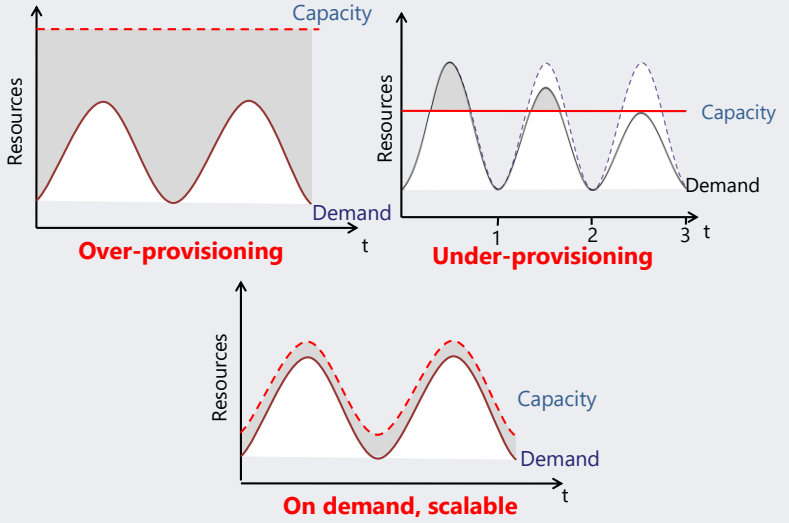
- Computing performed in a cloud?
- The National Institute of Standards and Technology (NIST) definition:

> Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

What is cloud computing?

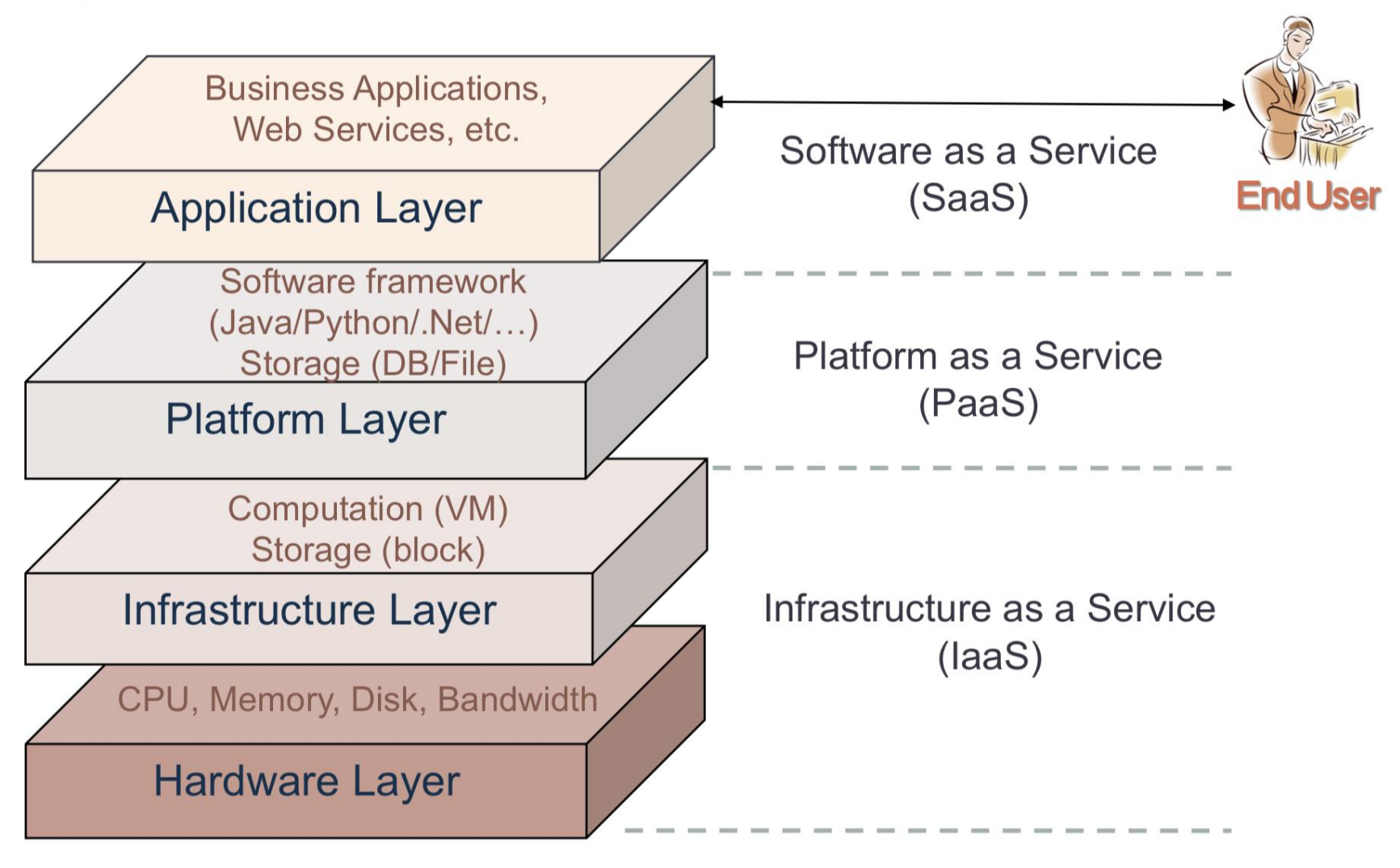
- (Anytime, anywhere) An environment where IT resources needed can be instantly created and used
  - Consumers: off-loading to cloud & using only when needed
  - Service providers: increasing revenue by enhancing resource utilization





[1] http://www.basilit.com/Website/CloudComputing.aspx
[2] *Above the Clouds: A Berkeley View of Cloud Computing*

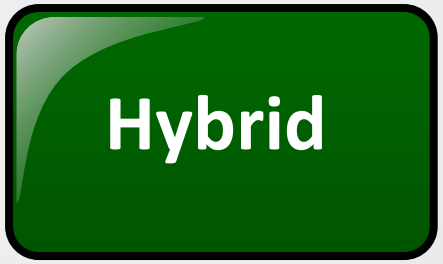
Type of clouds: service type



Type of clouds: usage type

Public cloud
Resources offered as services from a third-party company (e.g.,: Amazon AWS, Microsoft Azure, Google Cloud, KT uCloud, NAVER Cloud, TOAST) to the general public by the Internet

Private cloud
Designed for exclusive use by a single organization
Highest degree of control over performance, reliability and security



Hybrid cloud
Part of the service infrastructure runs in private clouds while the remaining part runs in public clouds to take both advantages on public & private clouds

Enabling technologies: data center virtualization

- Virtualization enables to create virtual IT infrastructure resources dynamically upon physical hardware infrastructure

Advantage

Logical separation

Centralized

Testability

Mobility

Weakness

Risky from centralization

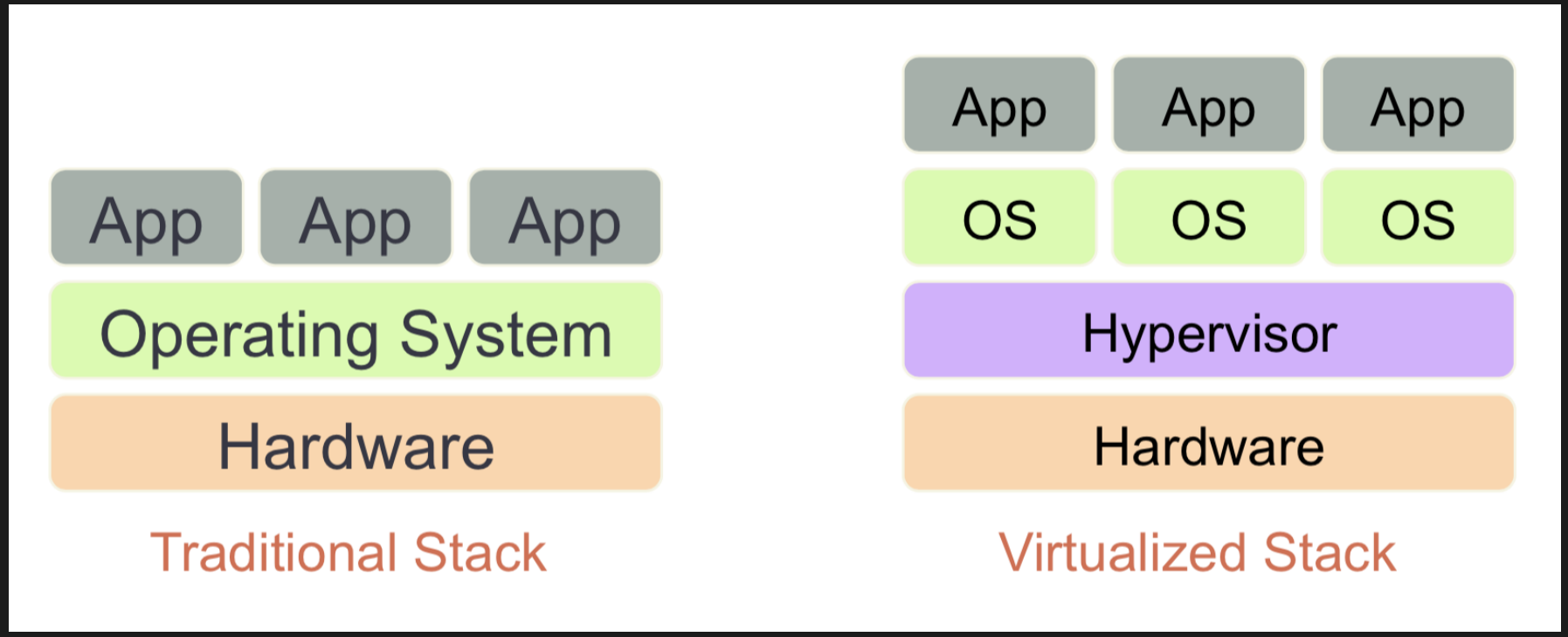
Cost for physical infrastructure

Performance degradation

Hardware support

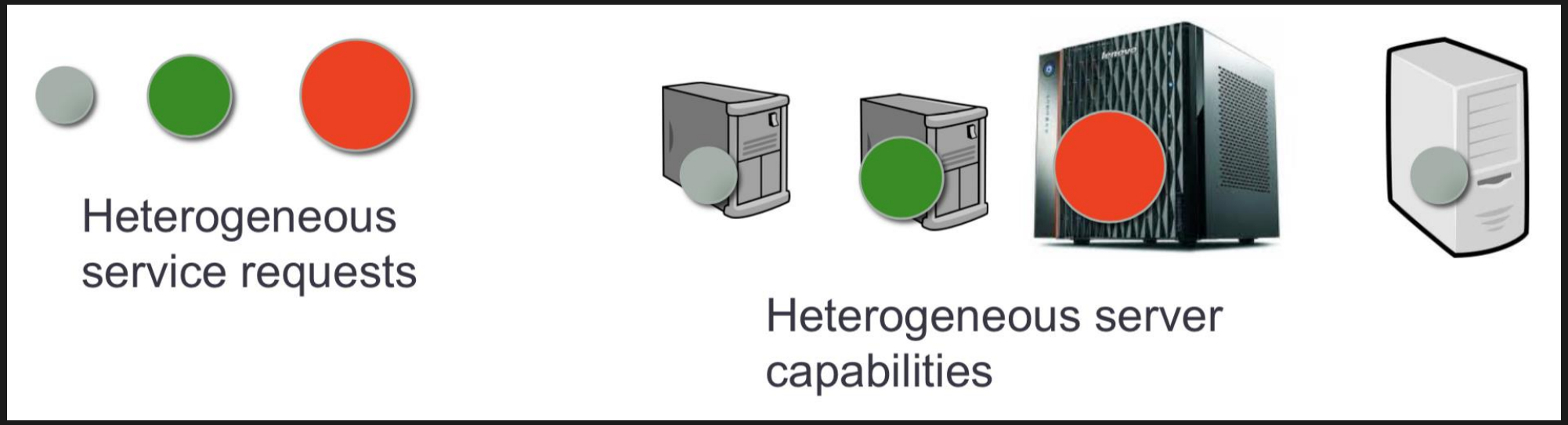
Enabling technologies: data center virtualization

- Virtualization techniques
  - Computer Virtualization is one approach to improve resource utilization and flexibility in data centers



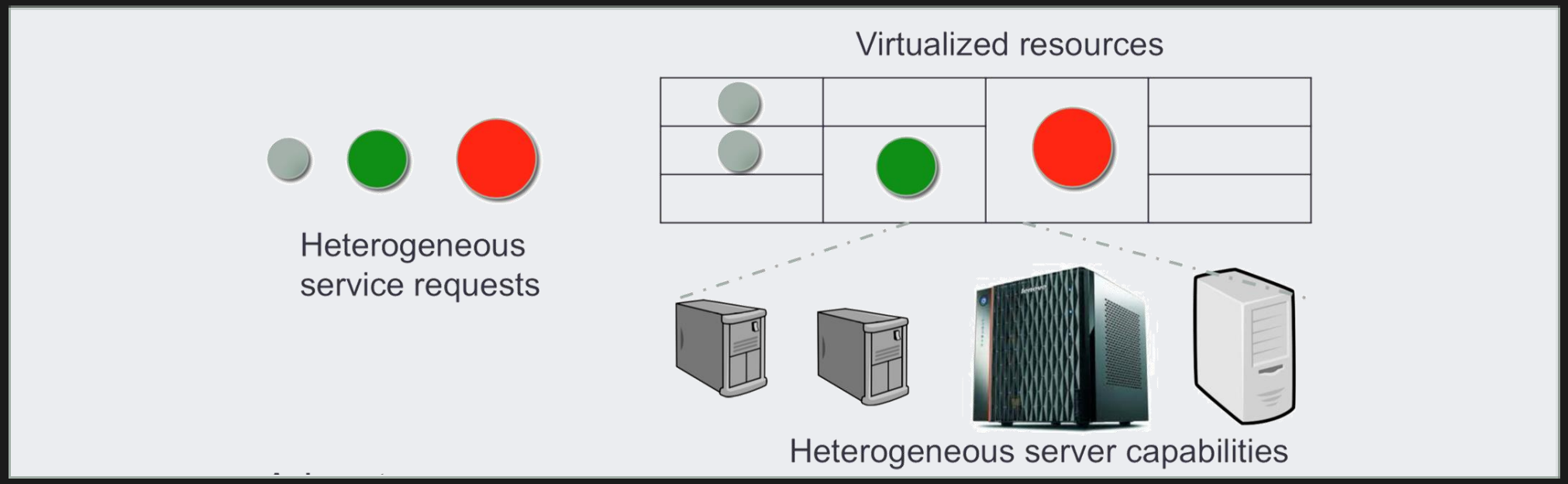
Enabling technologies: data center virtualization

- Dynamic resource sharing / consolidation
  - Before (Issues)
    - Performance/security isolation leads to waste of resources
    - Limited flexibility in shared environments



Enabling technologies: data center virtualization

- Dynamic resource sharing / consolidation
  - Now (Advantages)
    - Customizable CPU, memory, storage and networking capabilities
    - Allow apps to migrate from one machine to another
    - Power efficiency



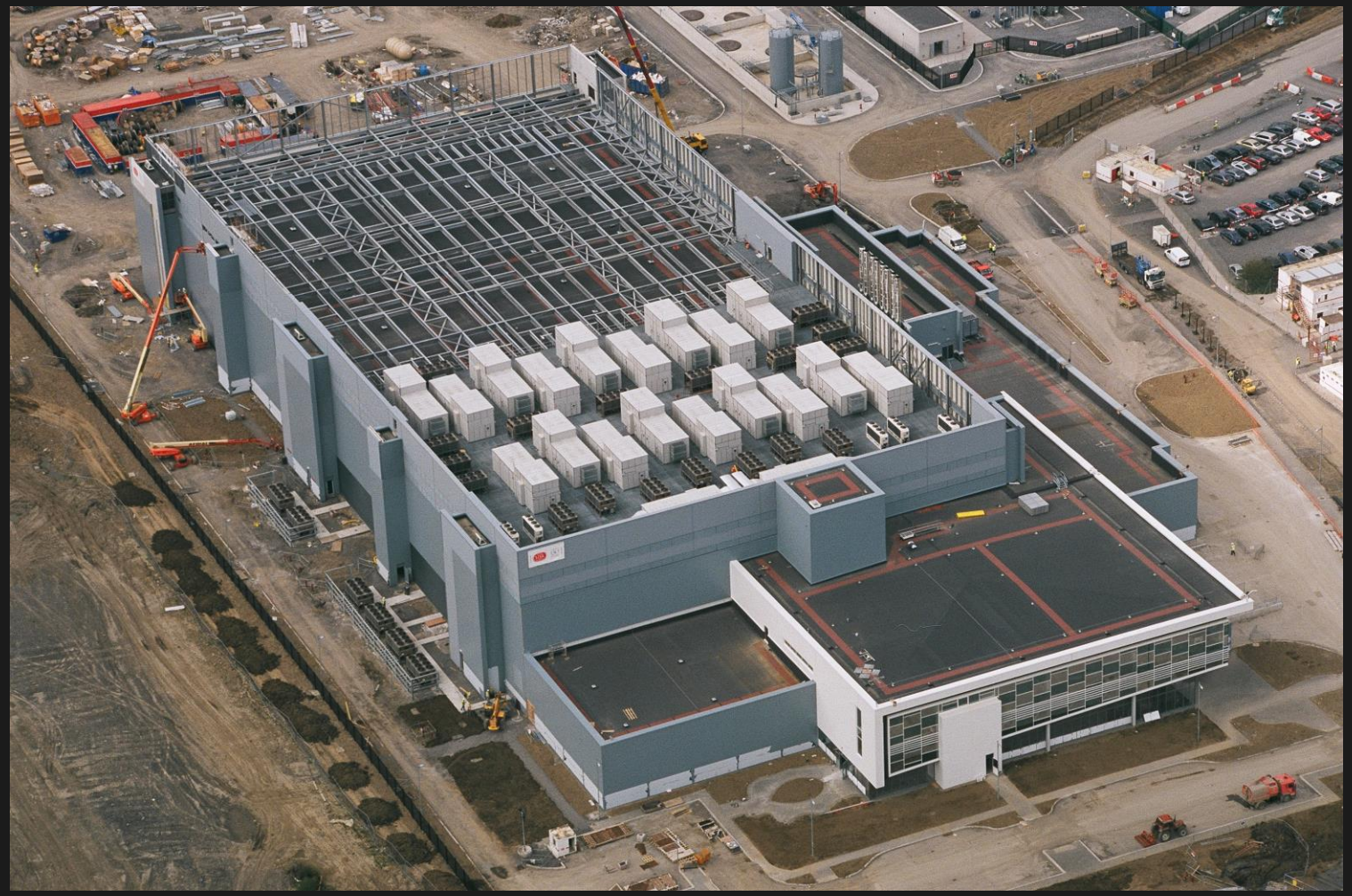
Enabling technologies: data center virtualization

- Data center

Maximilien Brice, © CERN

Enabling technologies: data center virtualization

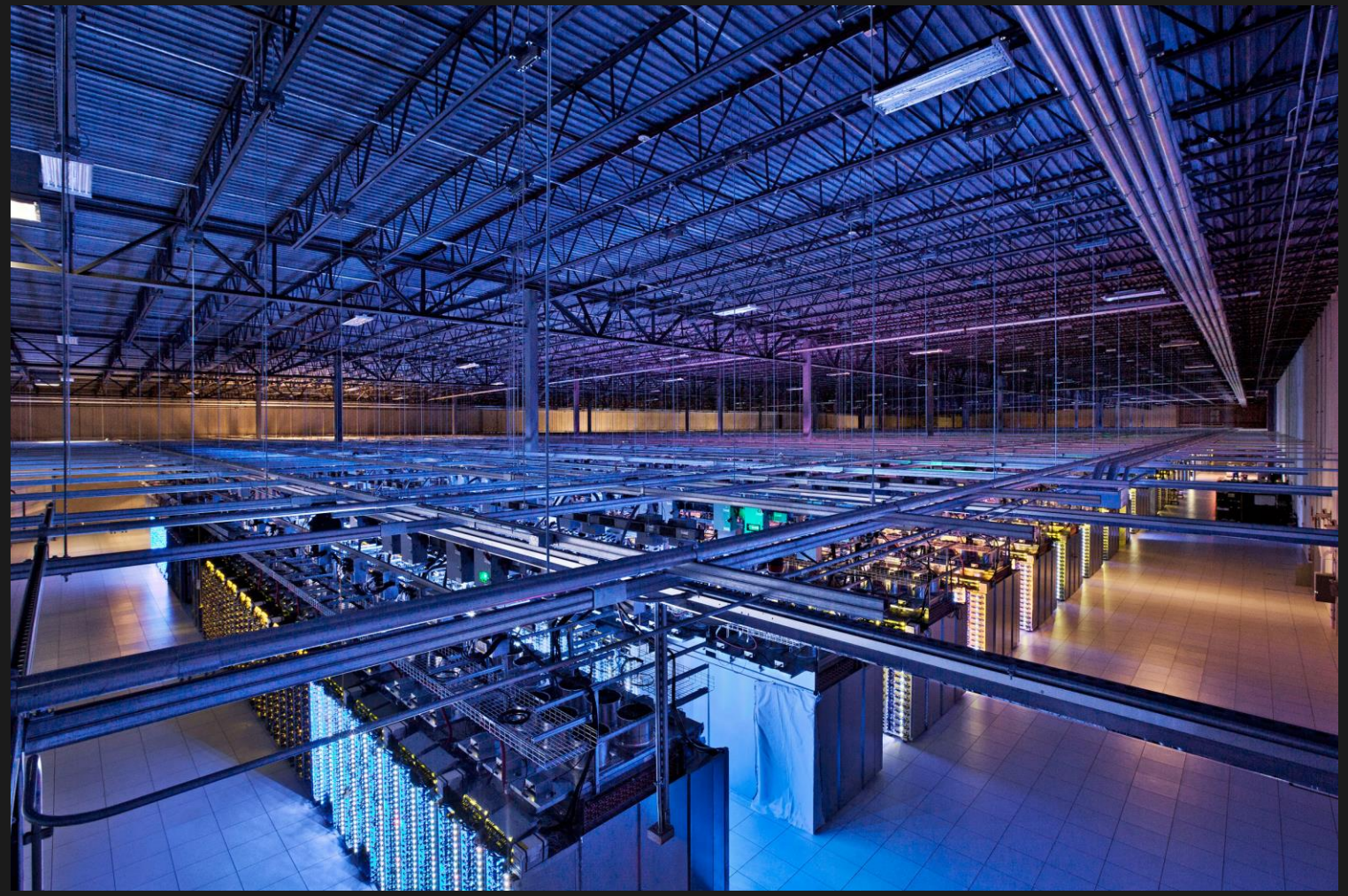
- Data center



Microsoft Data Center (Dublin, Ireland)

Enabling technologies: data center virtualization

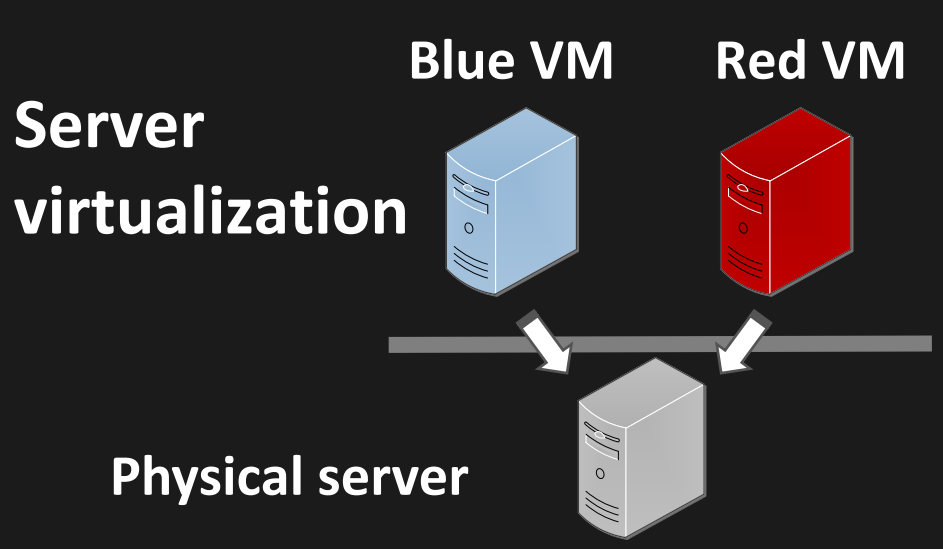
- Data center

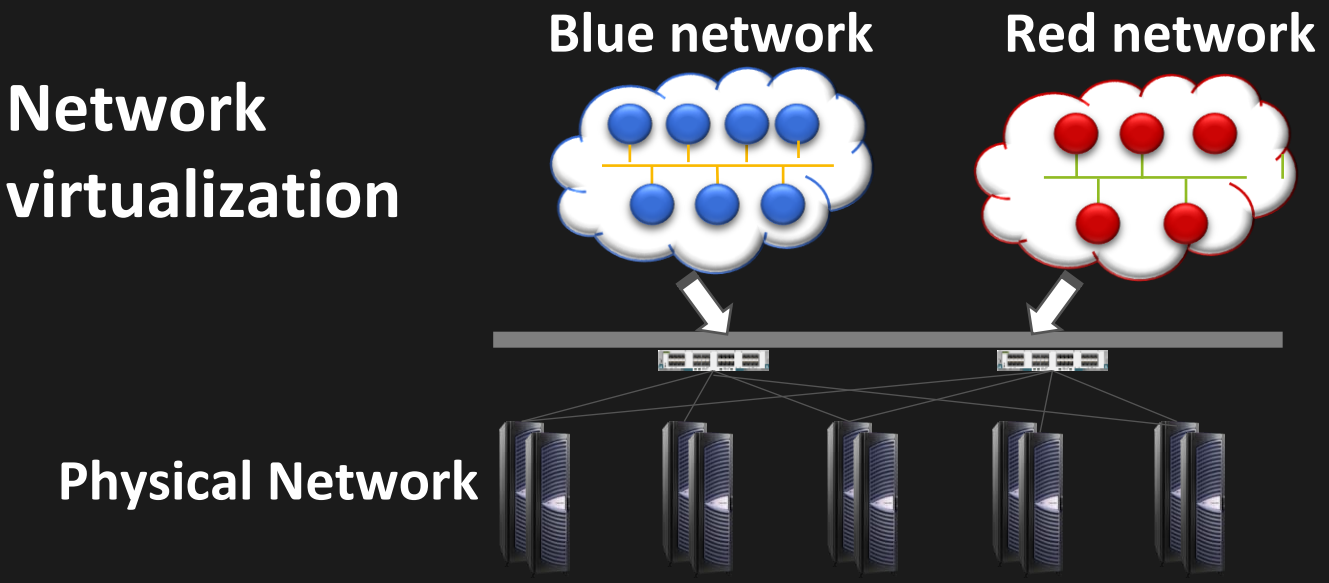


A Google Data Center (Iowa, US)

Enabling technologies: data center virtualization

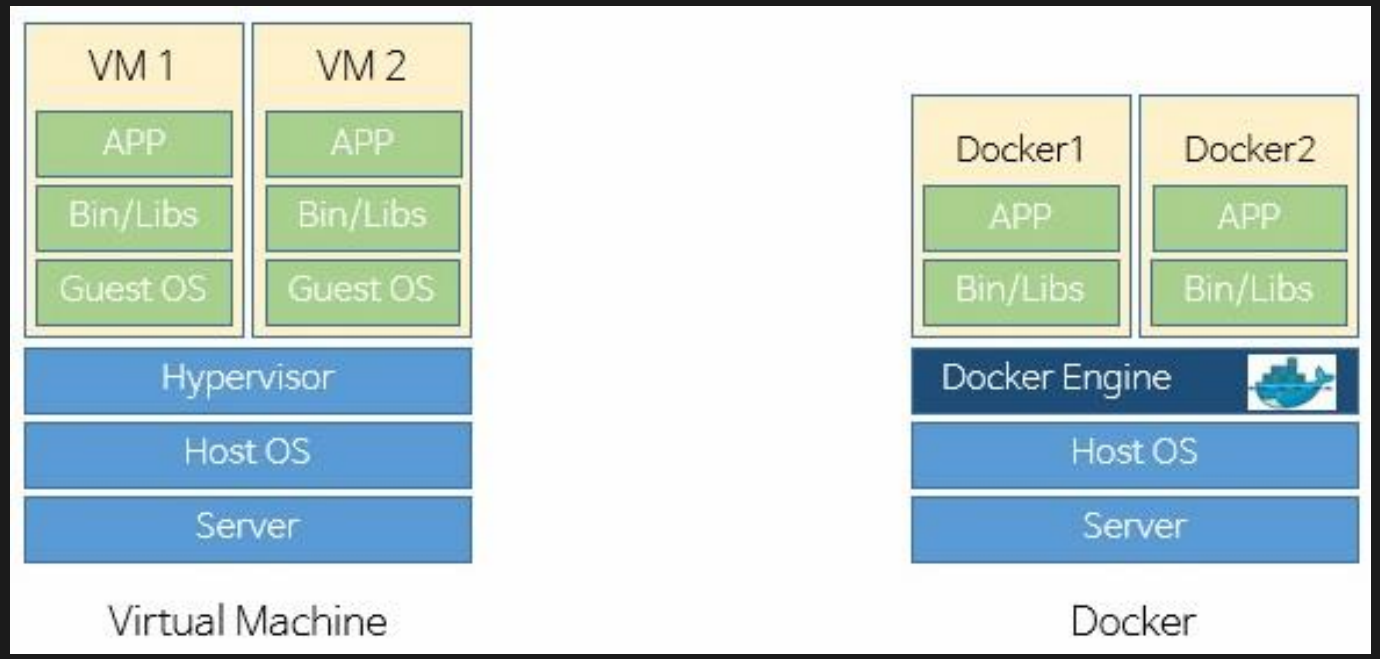
- Server virtualization has been possible through
  - Hypervisors: VMware ESXi, Microsoft Hyper-V, Citrix XenServer, KVM, Xen, …
  - Hardware support: Inter VT/VT-x, AMD-V
- Network virtualization enables to create separated virtual networks for dedicated set of virtual machines.

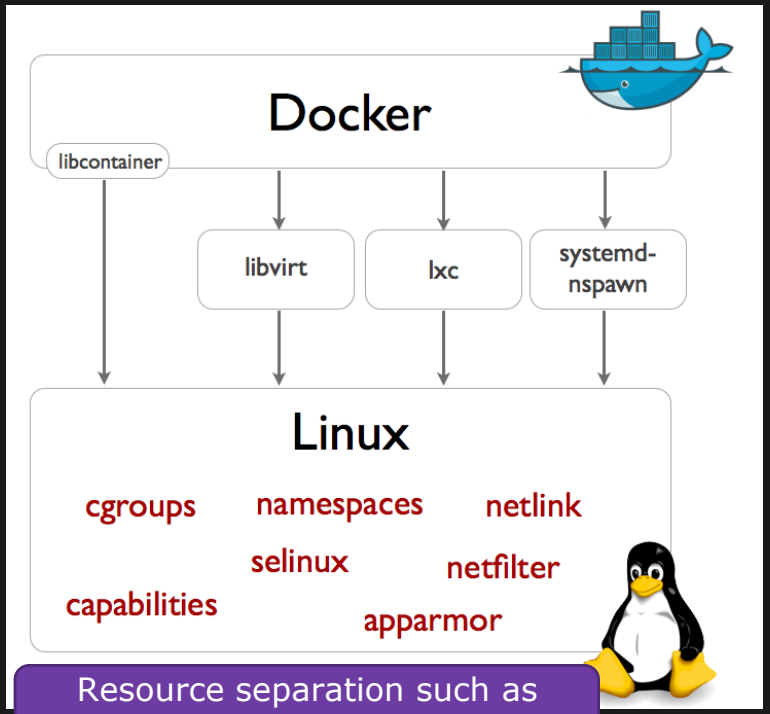




Enabling technologies: virtual machines & containers

- Containers
  - Sharing host OS kernel
  - Logical separation on the top of host OS kernel
- Docker is the most popular Linux container technology

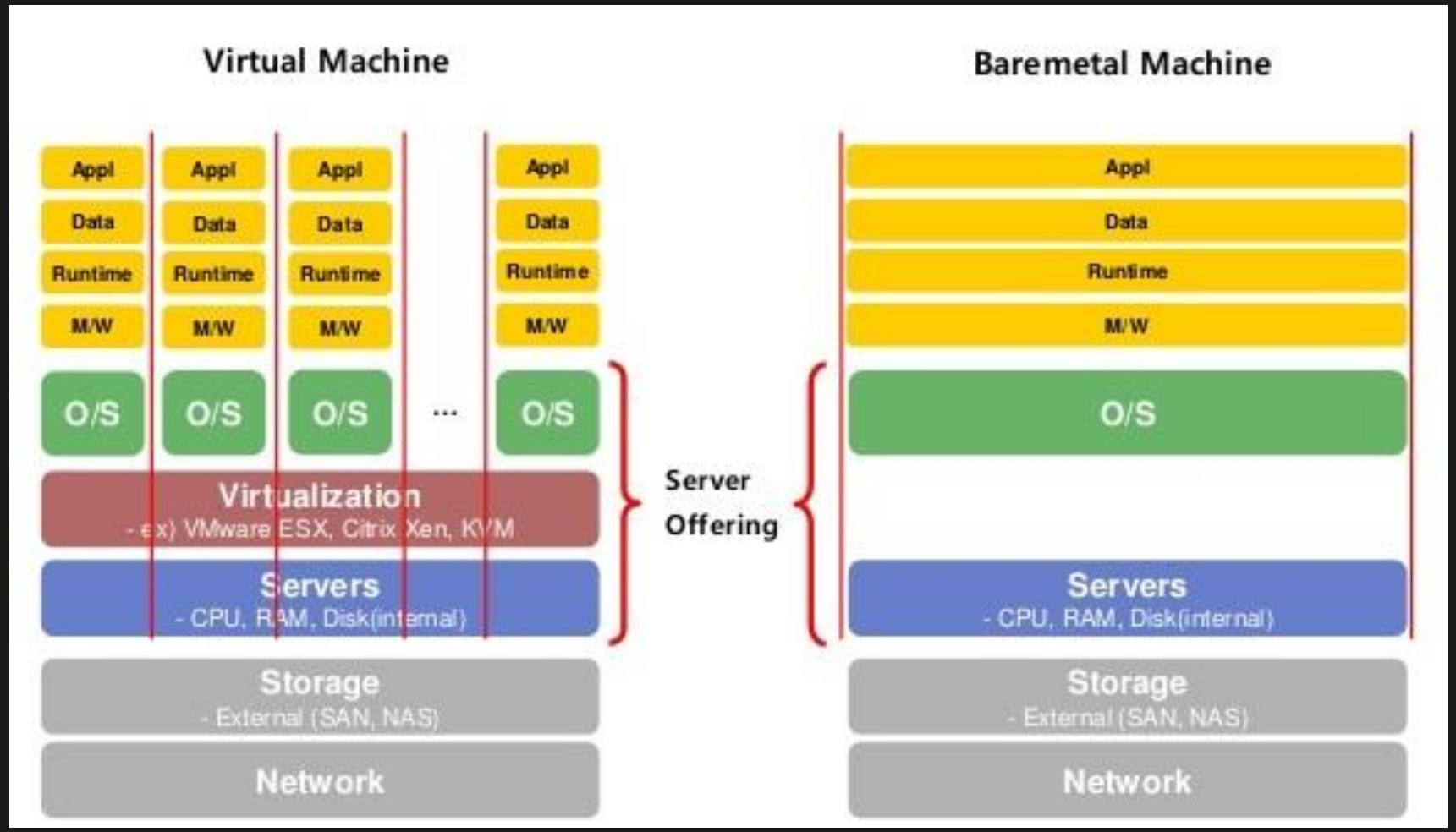




Resource separation such as CPUs, memories, and networks

Enabling technologies: bare metal

- Bare metal
  - Providing “server”
  - How to manage bare metal hardware?



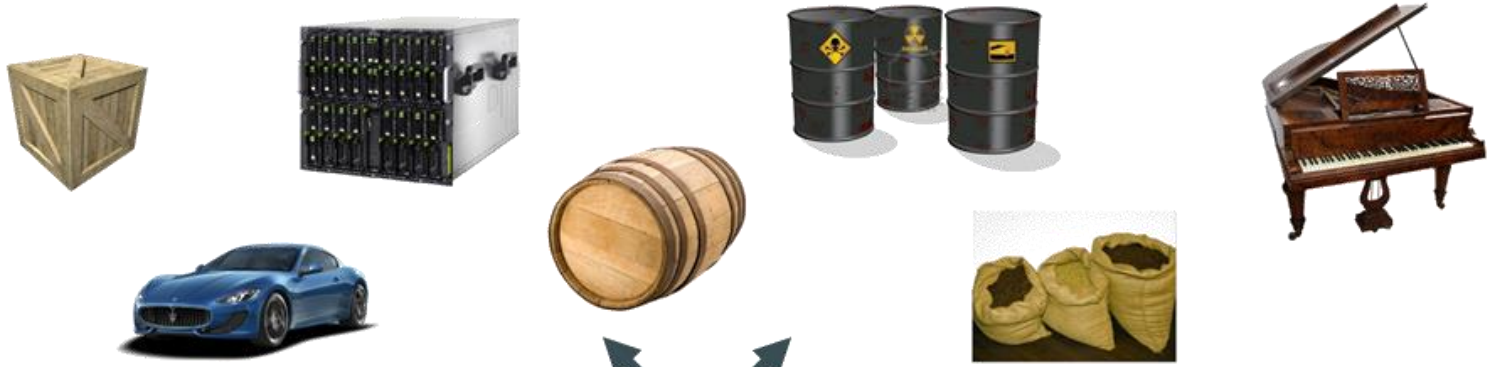
2. Where containers came from?

Q: Virtualization vs. Container technology?

a very brief History of Shipping

Cargo transport pre-1960

Multiplicity of goods



Do I worry about how goods interact?

Multiplicity of methods for transporting & storing

Can I transport quickly and smoothly?

Cargo transport pre-1960

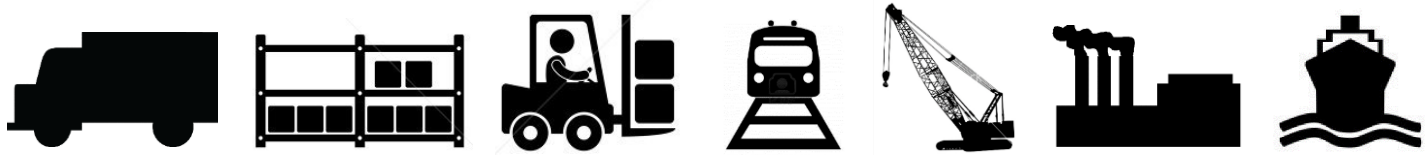
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

The Shipping Container analogy brings a solution

Multiplicity of goods

Do I worry about how goods interact?

Multiplicity of methods for transporting & storing



Can I transport quickly and smoothly?

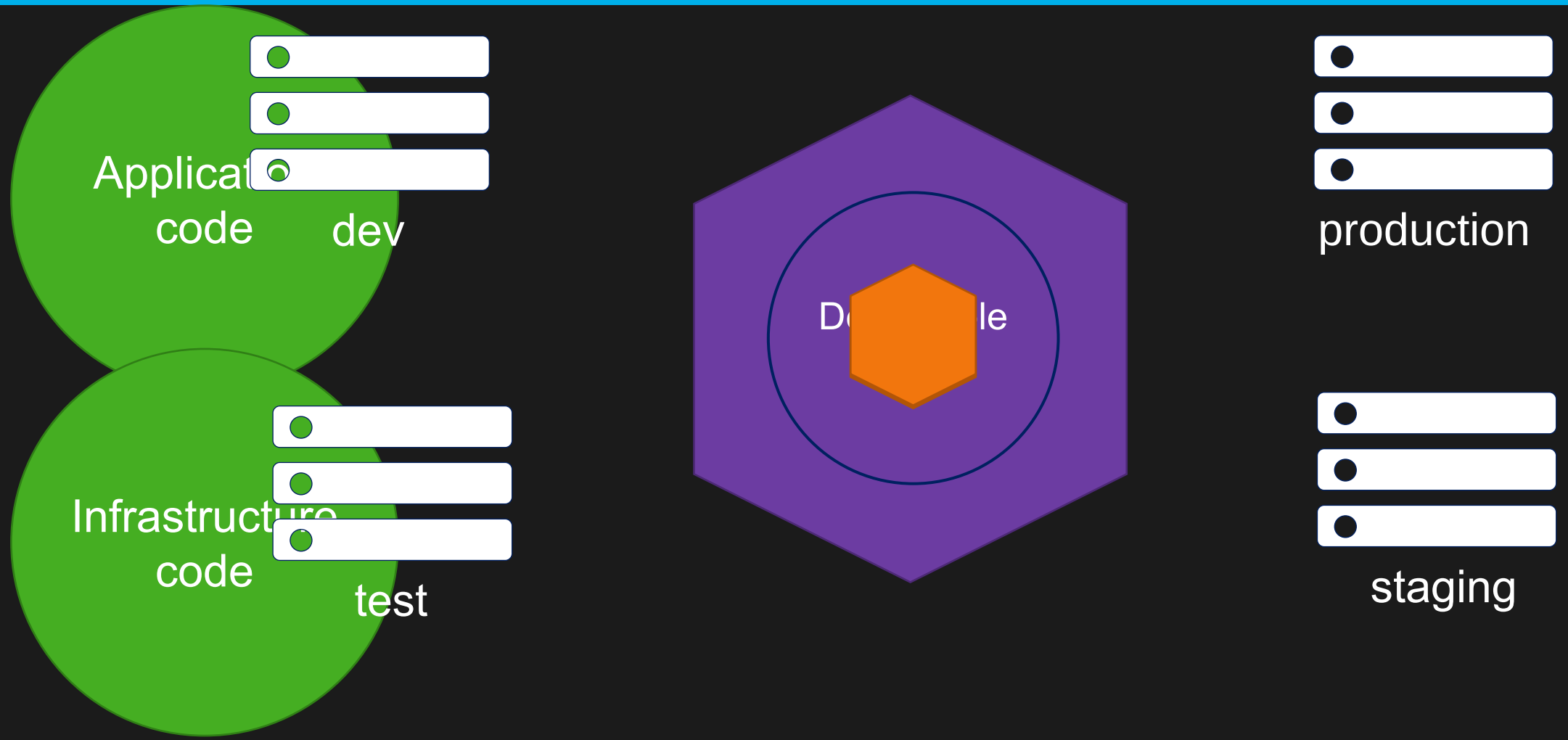

MAERSK SEALAND
K LINE
COSCO
HANJIN
SEA STAR
EVERGREEN
UASC
CAPITAL
APL
MAERSK


NYK ATLAS
PANAMA


so
What's
The Big Deal ?

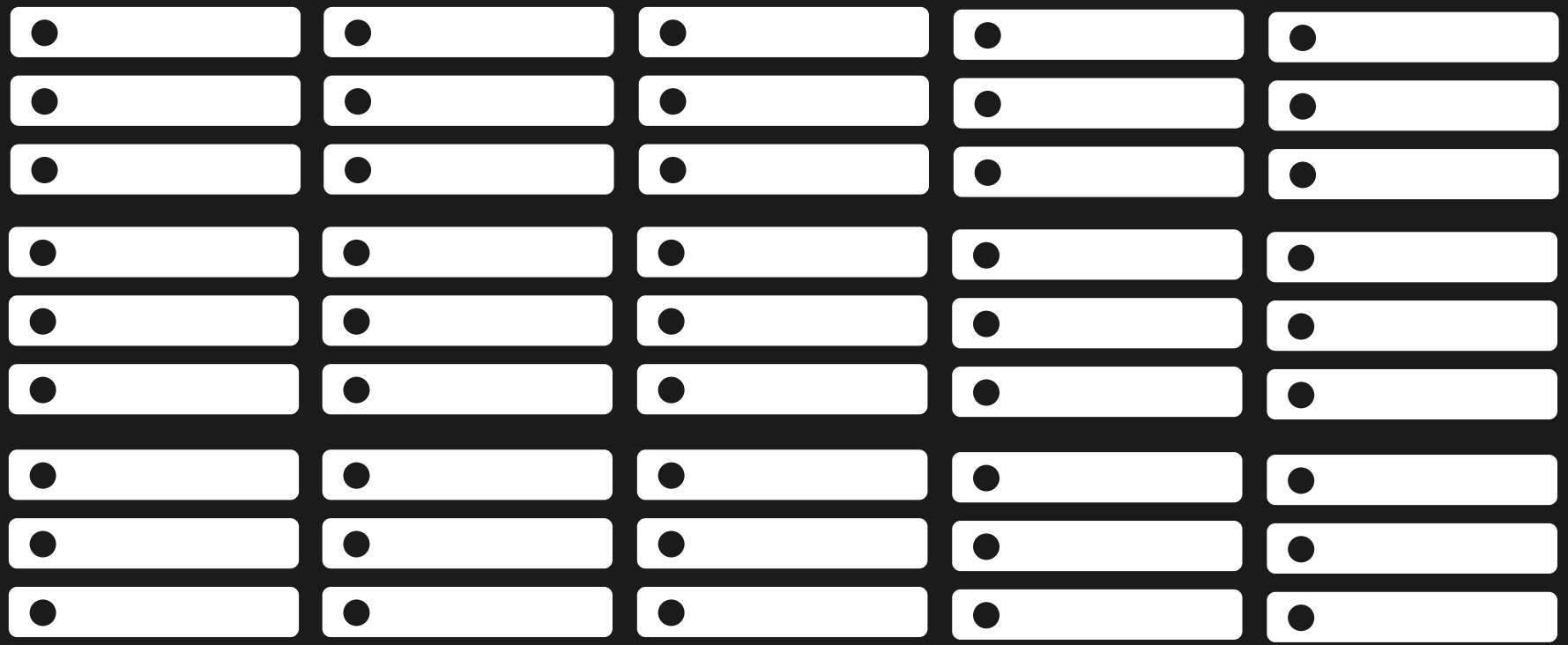
containers for code

Containers for code

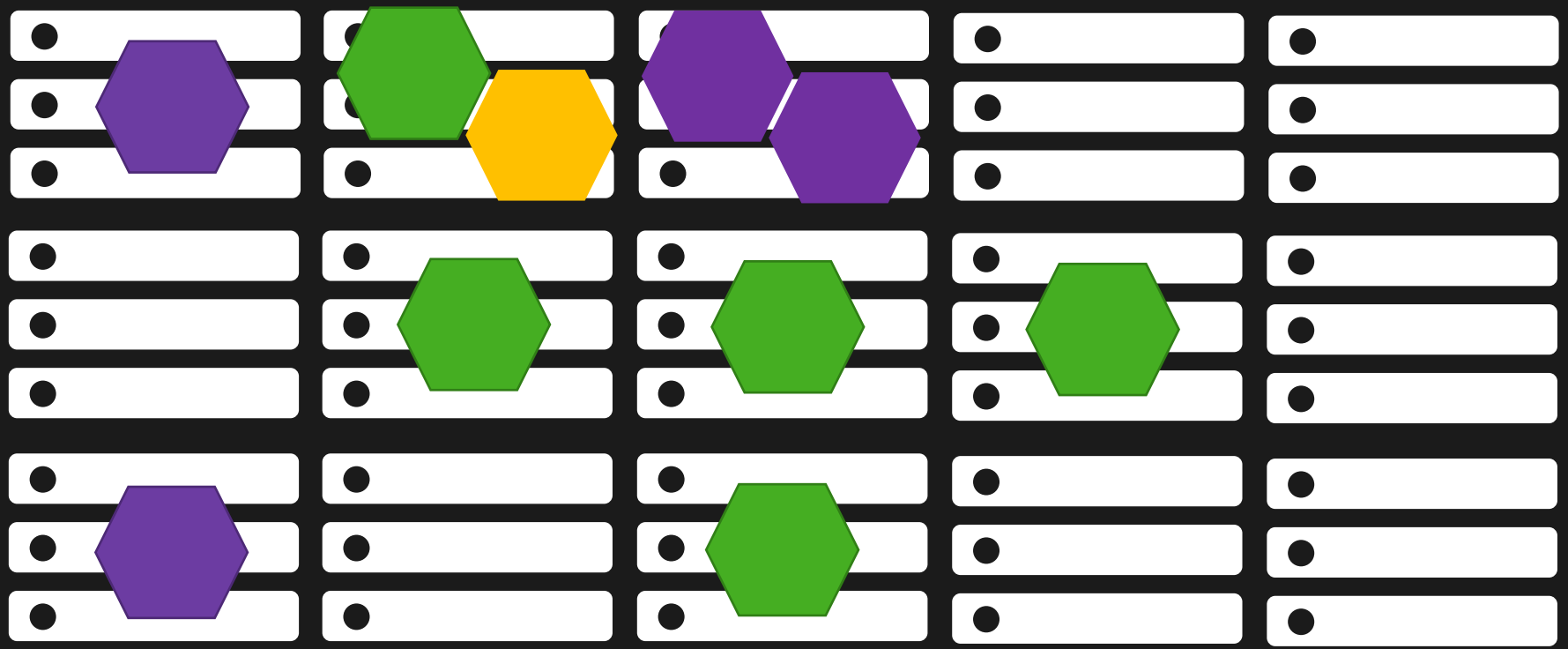


density & efficiency

Data center without containers

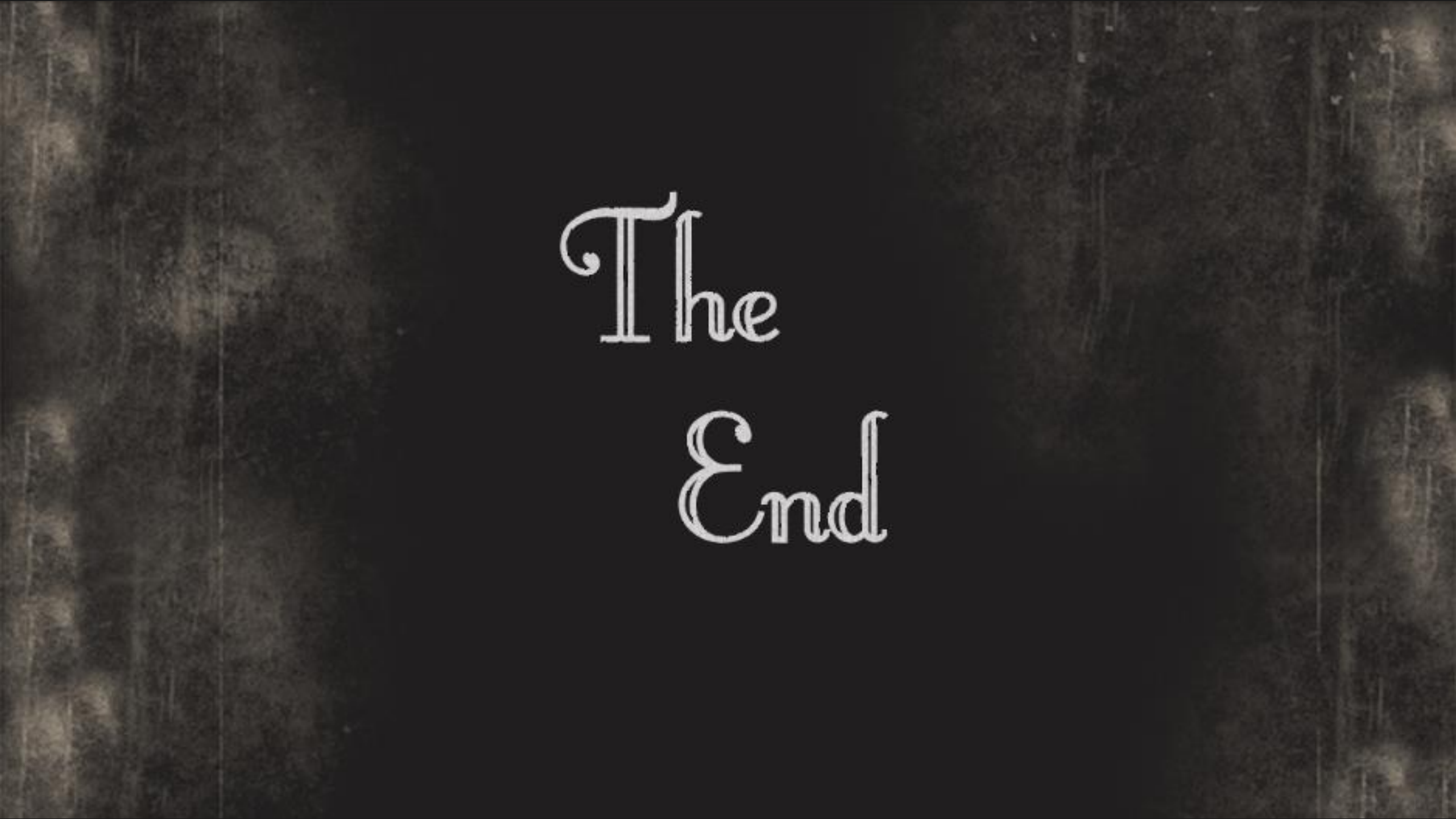


Data center with containers

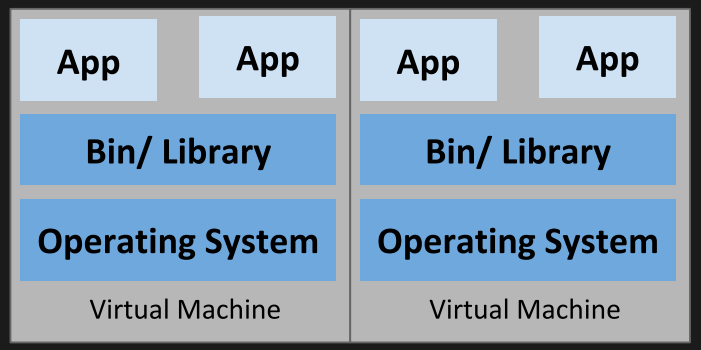


build once
run anywhere


micro services

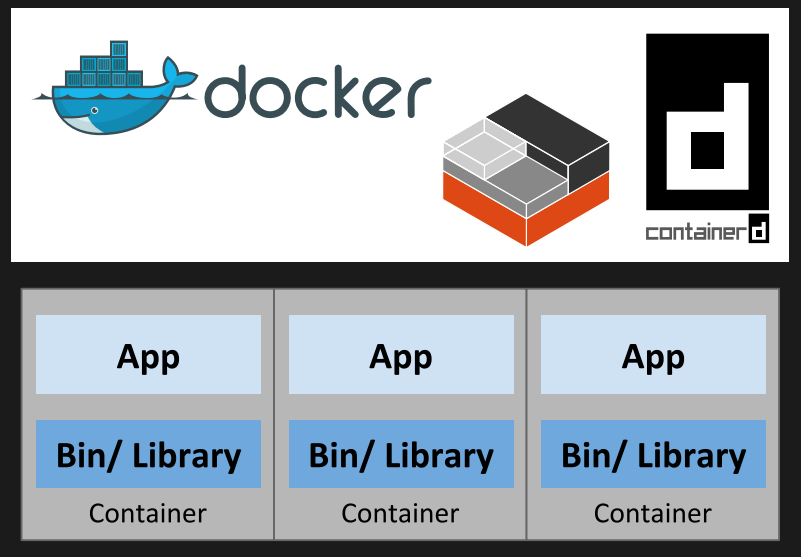

The
End

Cloud Native Computing Foundation (CNCF) in 2015



Deployment on VMs **vs.**

KVM Microsoft Hyper-V vmware Xen



Deployment on containerized environment

ANNOUNCEMENTS

New Cloud Native Computing Foundation to drive alignment among container technologies

Posted on June 21, 2015

AT&T, Box, Cisco, Cloud Foundry Foundation, CoreOS, Cycle Computing, Docker, eBay, Goldman Sachs, Google, Huawei, IBM, Intel, Joyent, Kismatic, Mesosphere, Red Hat, Switch SUPERNAP, Twitter, Univa, VMware and Weaveworks join new effort to build and maintain cloud native distributed systems

SAN FRANCISCO, Calif., July 21, 2015 – The Linux Foundation, the nonprofit organization dedicated to accelerating the growth of Linux and collaborative development, today announced the Cloud Native Computing Foundation.

CLOUD NATIVE COMPUTING FOUNDATION

+

3. Cloud Native and Kubernetes, 1st CNCF Graduated Project

Q: How “Cloud Native” is different from cloud computing?

Definition of “Cloud Native” by CNCF

https://github.com/cncf/toc/blob/main/DEFINITION.md

CNCF Cloud Native Definition v1.0

Approved by TOC: 2018-06-11

العربية (Arabic) | 中文版本 (Chinese) | עברית (Hebrew) | 日本語版 (Japanese) | 한국어 (Korean) | Deutsch (German) | Español (Spanish) Français (French) | Polski (Polish) | Português Brasileiro (Portuguese-BR) | Português de Portugal (Portuguese-PT) | Русский (Russian) | Bahasa Indonesia (Indonesian) | Türkçe (Turkish)

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

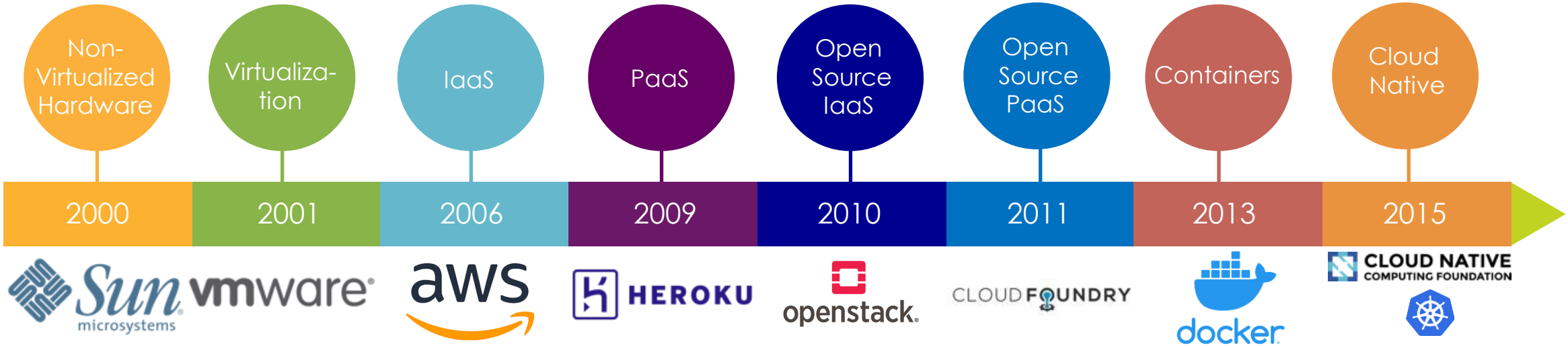
The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

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



 | (Source: Cloud Native Computing Foundation)

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

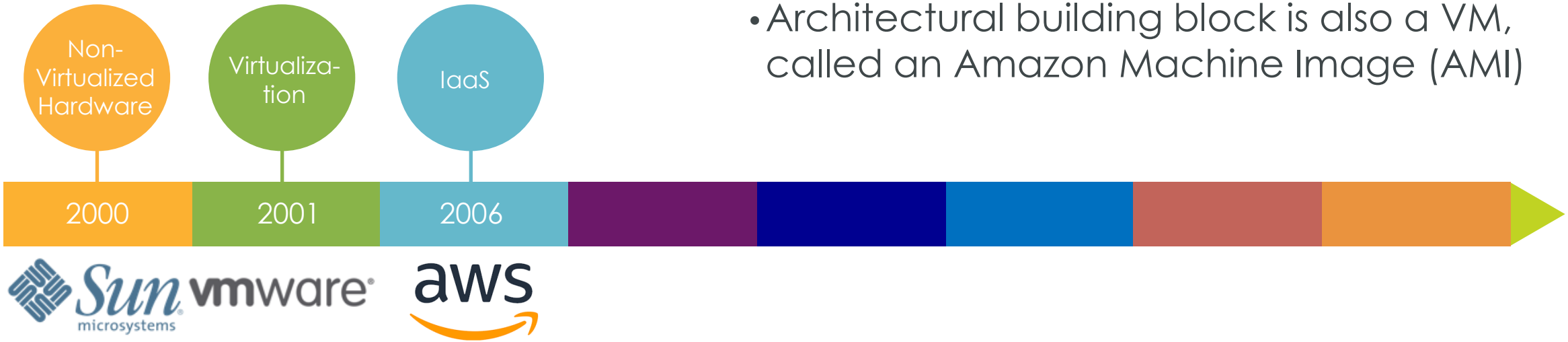
 | (Source: Cloud Native Computing Foundation)

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

(Source: Cloud Native Computing Foundation)

IaaS: AWS (2006)

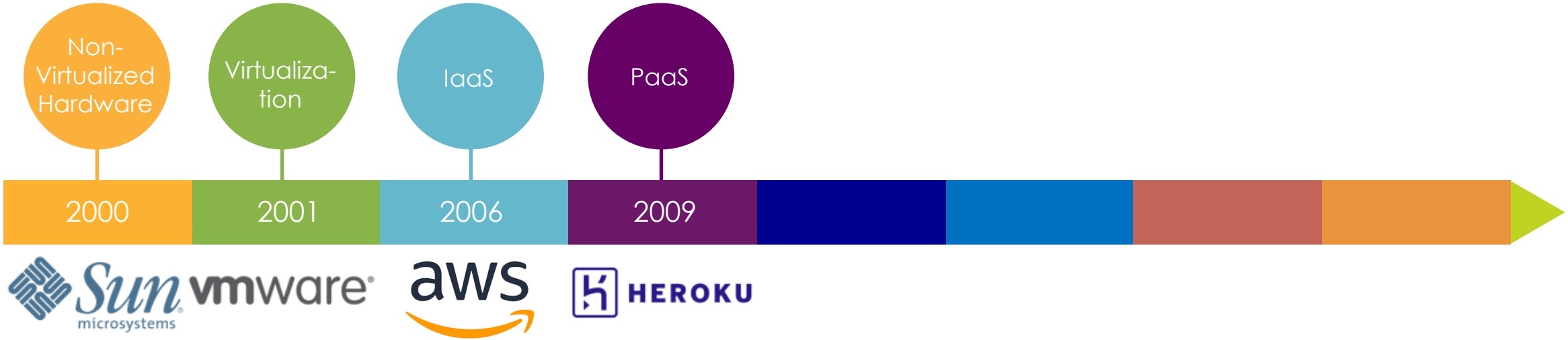


- Amazon Web Services (AWS) creates the Infrastructure-as-a-Service market by launching Elastic Compute Cloud (EC2) in 2006
- Rent servers by the hour
- Convert CapEx to OpEx
- Architectural building block is also a VM, called an Amazon Machine Image (AMI)

(Source: Cloud Native Computing Foundation)

PaaS: Heroku (2009)

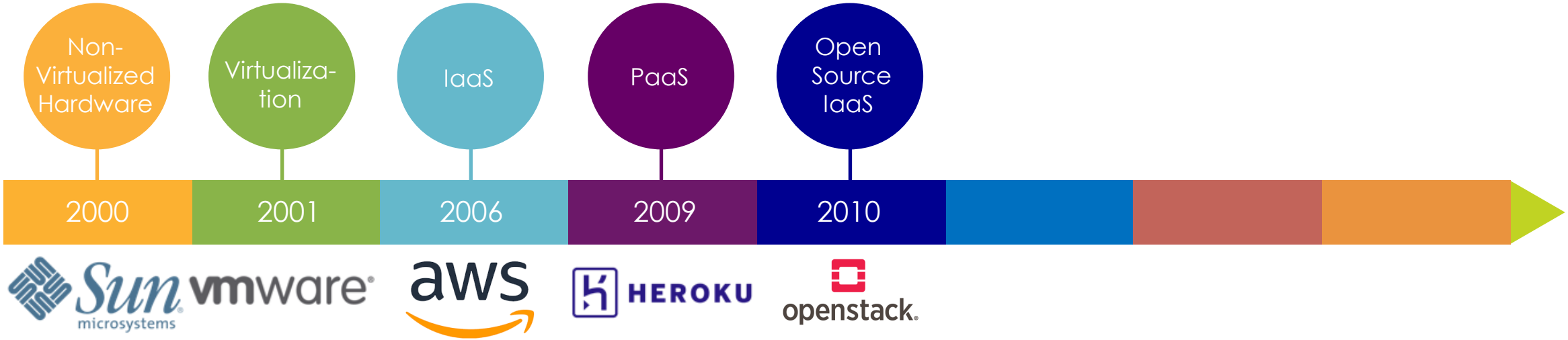
- Heroku popularizes Platform-as-a-Service (PaaS) which their launch in 2009
- Building block is a buildpack, which enables containerized 12-factor applications
  - The process for building the container is opacue, but:
  - Deploying new version of an app is just: **git push heroku**



(Source: Cloud Native Computing Foundation)

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

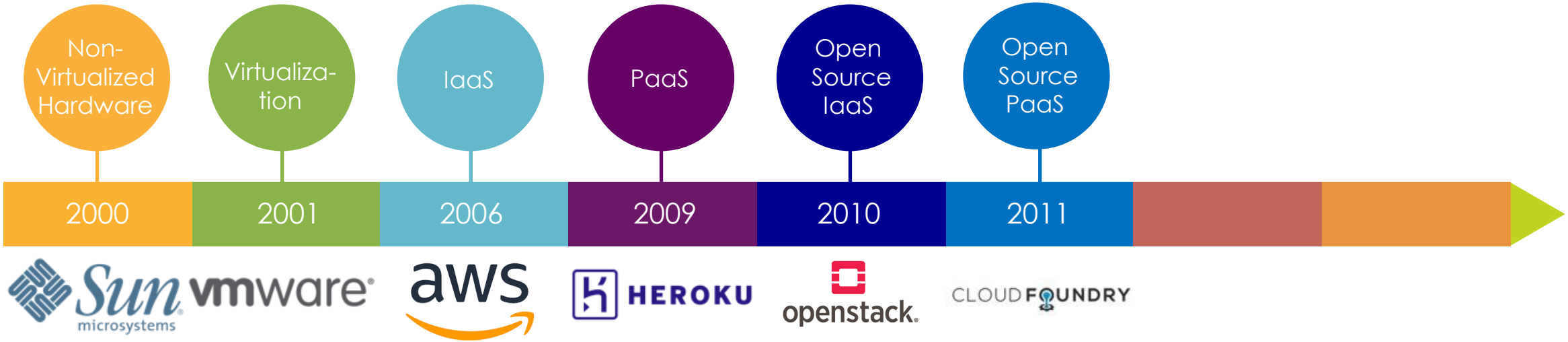


(Source: Cloud Native Computing Foundation)

Open Source PaaS: Cloud Foundry (2011)

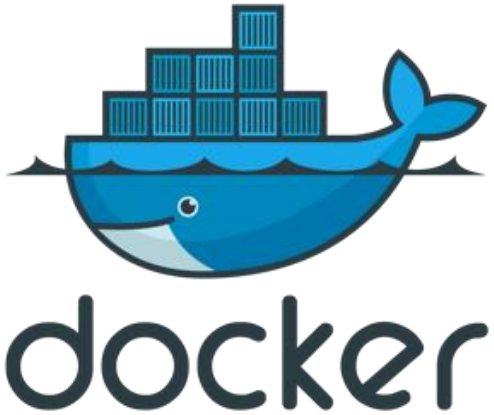
CLOUD FOUNDRY

- Pivotal brings 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

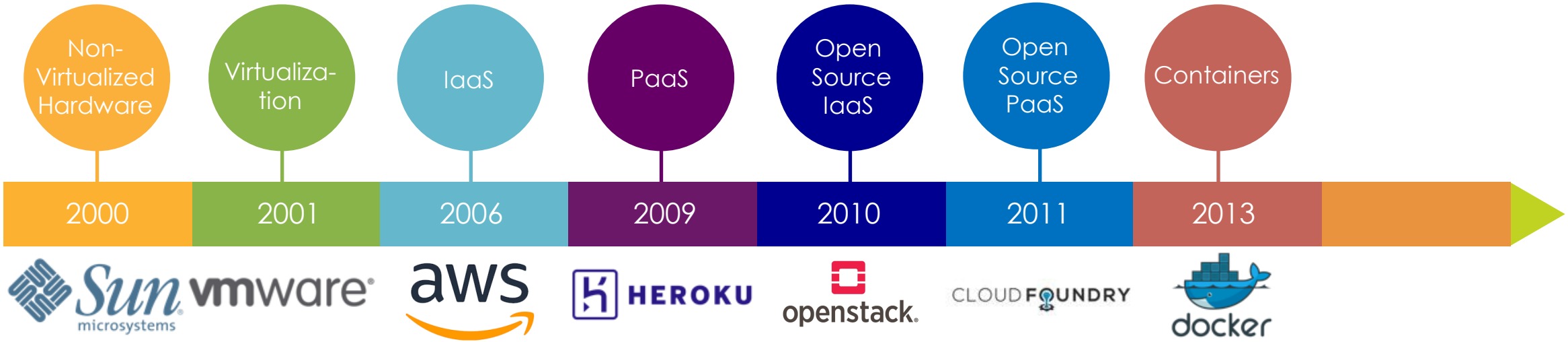


 | (Source: Cloud Native Computing Foundation)

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 immuability



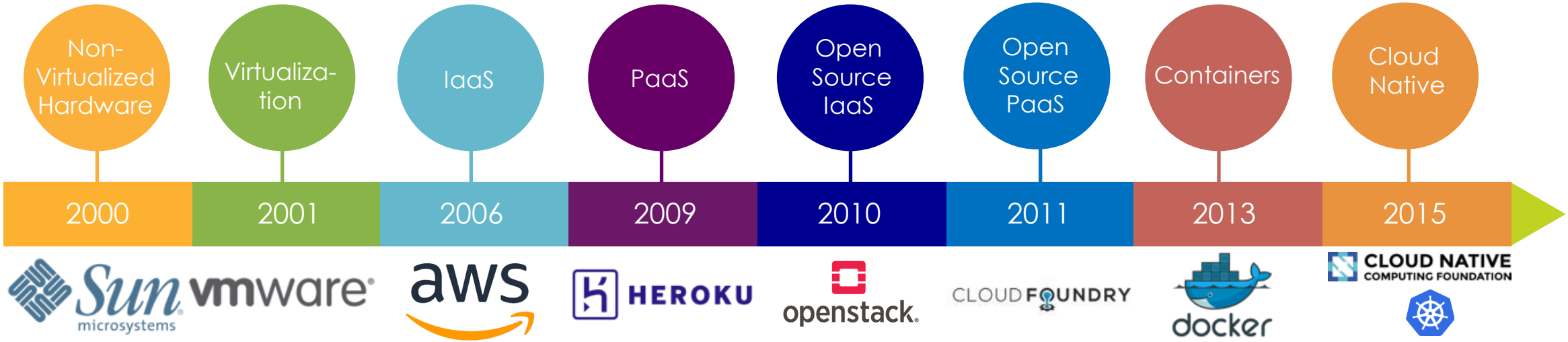
 | (Source: Cloud Native Computing Foundation)

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



(Source: Cloud Native Computing Foundation)

Cloud Native and open sources

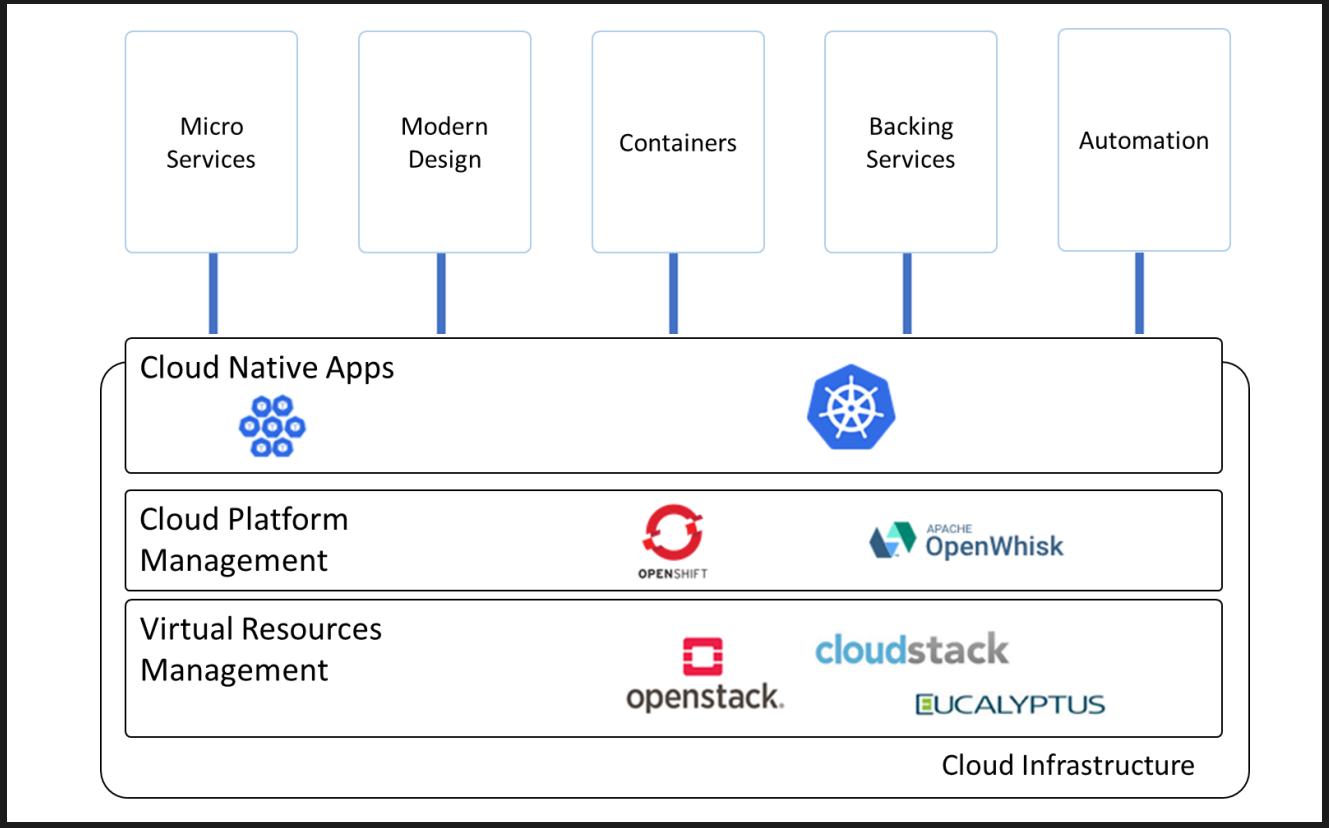
- There are so many open sources on cloud native
- Kubernetes is first "Graduated Project" in CNCF!

ANNOUNCEMENTS

Cloud Native Computing Foundation announces Kubernetes® as first graduated project

Posted on March 6, 2018

Container orchestration system widely deployed at scale with numerous global organizations

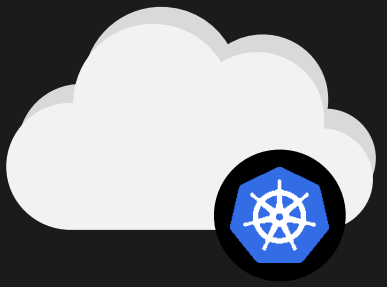


What is Kubernetes(k8s)?

- **Kubernetes**: “open-source orchestration software for deploying, managing, and scaling containers”
- “κυβερνήτης” in Greek, meaning helmsman or pilot
  - K8s: an abbreviation results from counting the eight letters between the "K" and the "s"

[Brief History]

- development and design are heavily influenced by Google's Borg system
- Kubernetes v1.0 release: July 21, 2015 (Founder: Joe Beda, Brendan Burns Craig McLuckie)
- GitHub is a main repository for open source contributions, and new releases come for every 3-6 months (latest release: v1.21 on April 8, 2021)
- See Large-scale cluster management at Google with Borg for detailed background and idea



Kubernetes: the industry leading “container” orchestrator

Portable

Public, private, hybrid, multi-cloud

Extensible

Modular, pluggable, hookable, composable

Self-healing

Auto-placement, auto-restart, auto-replication, auto-scaling

Pets

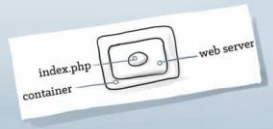
Legacy Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health
Infrastructure is a permanent fixture in the data center
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access
Infrastructure requires several different teams to coordinate and provision the full environment
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns

Cattle

Cloud-Friendly Infrastructure
Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is stateless, ephemeral, and transient
Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is a operating expenditure that charges only for services when they are consumed

Containers

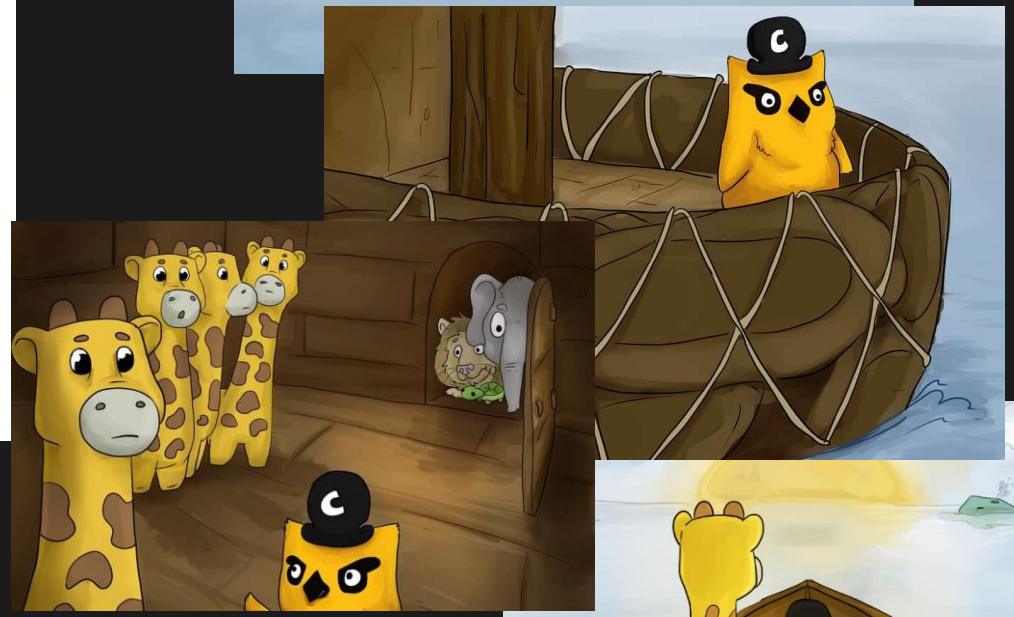
- Need to be managed
- Networking is hard
- Containers must be scheduled, distributed, and load balanced
- And the data's got to persist *somewhere*



Kubernetes

- Phi-Beta-Kappa: Philsophia Biou Kubernetes (Love of Wisdom Pilots life)

kubernetes



Kubernetes: the leading orchestrator shaping the future app development and management

It's widely used

Kubernetes is in production for global companies across industries[1]

Capital One	eBay	SAP
New York Times	Pokémon Go	Spotify

Vendor-neutral

A variety of cloud providers offer robust Kubernetes support

Azure	AWS
VMWare	Red Hat

It's community-supported

There's a huge community of active contributors supporting Kubernetes[3]

35,000 contributors since 2016	1.1 million contributions since 2016

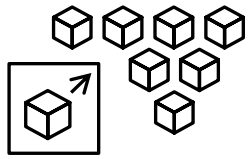
[1]Kubernetes.io. "Kubernetes User Case Studies." [2]CNCF. "Kubernetes Is First…" [3]CNCF. K8s Project Journey

Cloud Independence

4. Top scenarios with cloud native journey

Top scenarios with cloud native journey

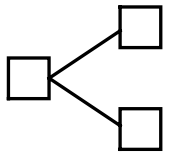
Lift and shift to containers



Cost saving

without refactoring your app

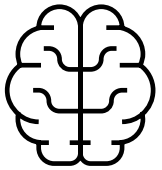
Microservices



Agility

Faster application development

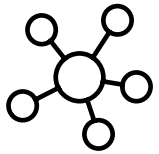
Machine learning



Performance

Low latency processing

IoT



Portability

Build once, run anywhere

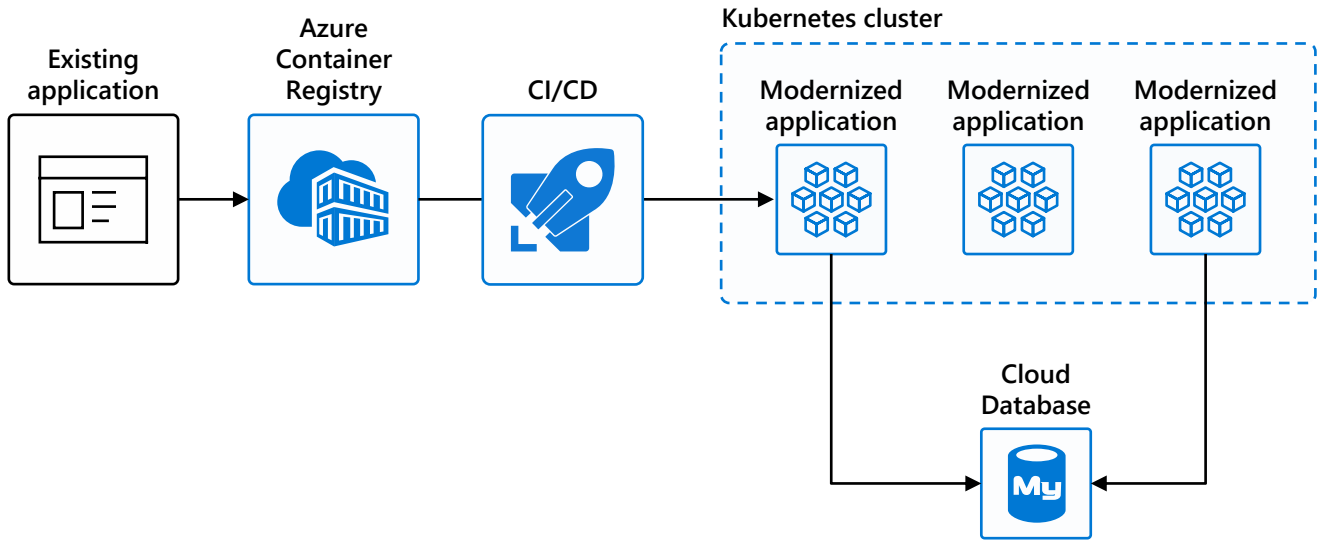
Secure DevOps

Automation

Deliver code faster and securely at scale

App modernization without code changes

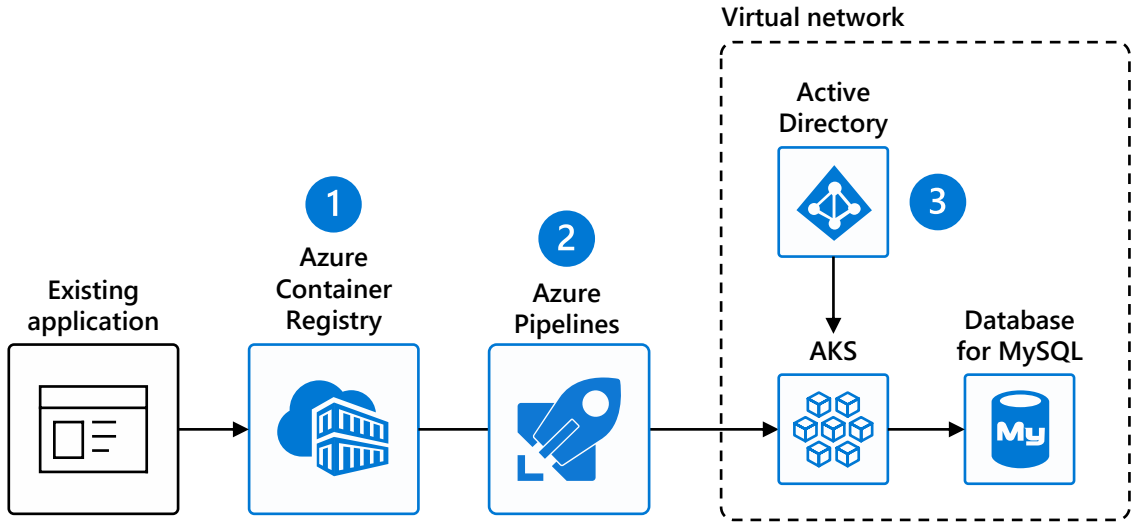
- Speed application deployments by using container technology
- Defend against infrastructure failures with container orchestration
- Increase agility with continuous integration and continuous delivery



App modernization without code changes

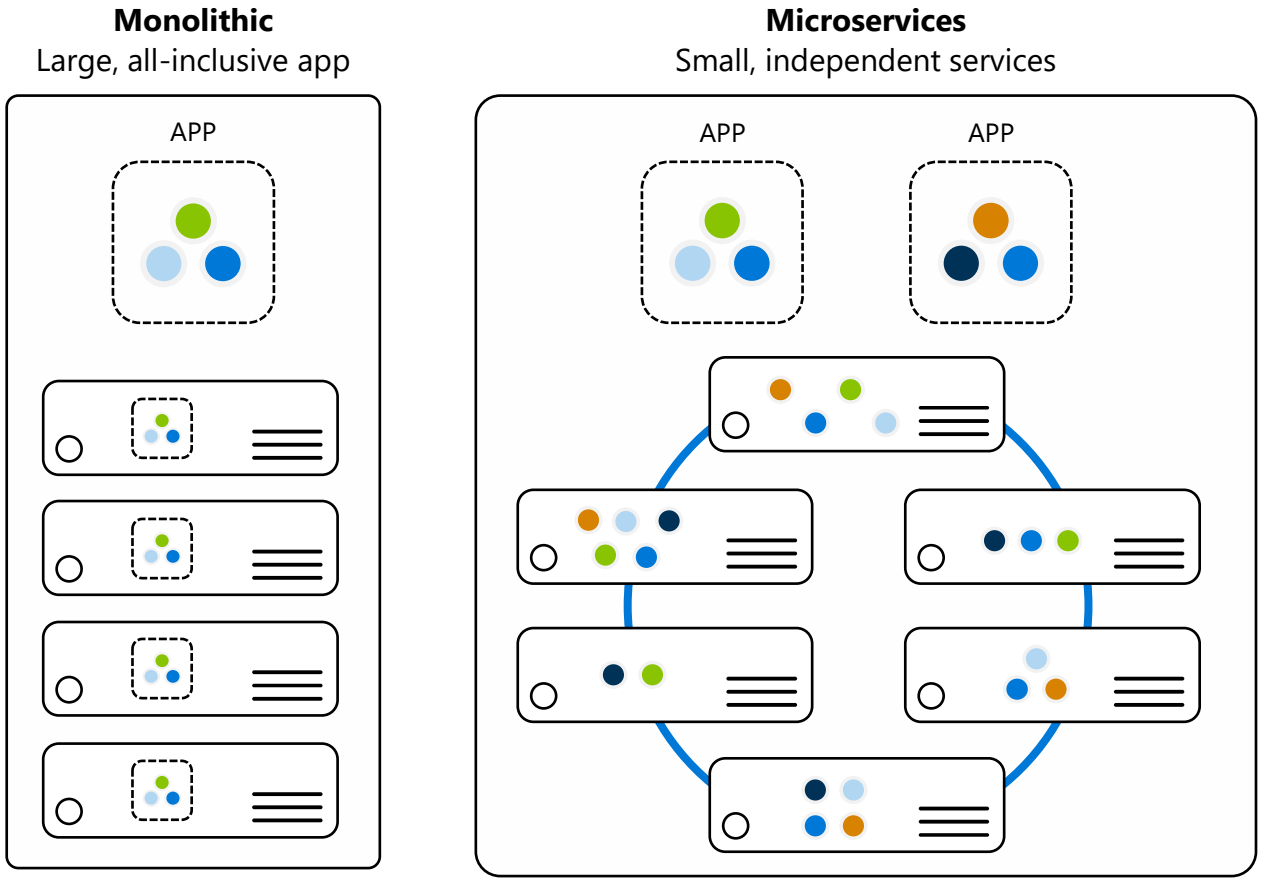
Capabilities

1. Use **Azure Container Registry** to store container images and Helm charts for your modernized applications, replicated globally for low latency image serving
2. Integrate AKS with **Azure Pipelines** or other Kubernetes ecosystem tooling to enable continuous integration/continuous delivery (CI/CD)
3. Enhance security with **Azure Active Directory** and RBAC to control access to AKS resources



Microservices: for faster app development

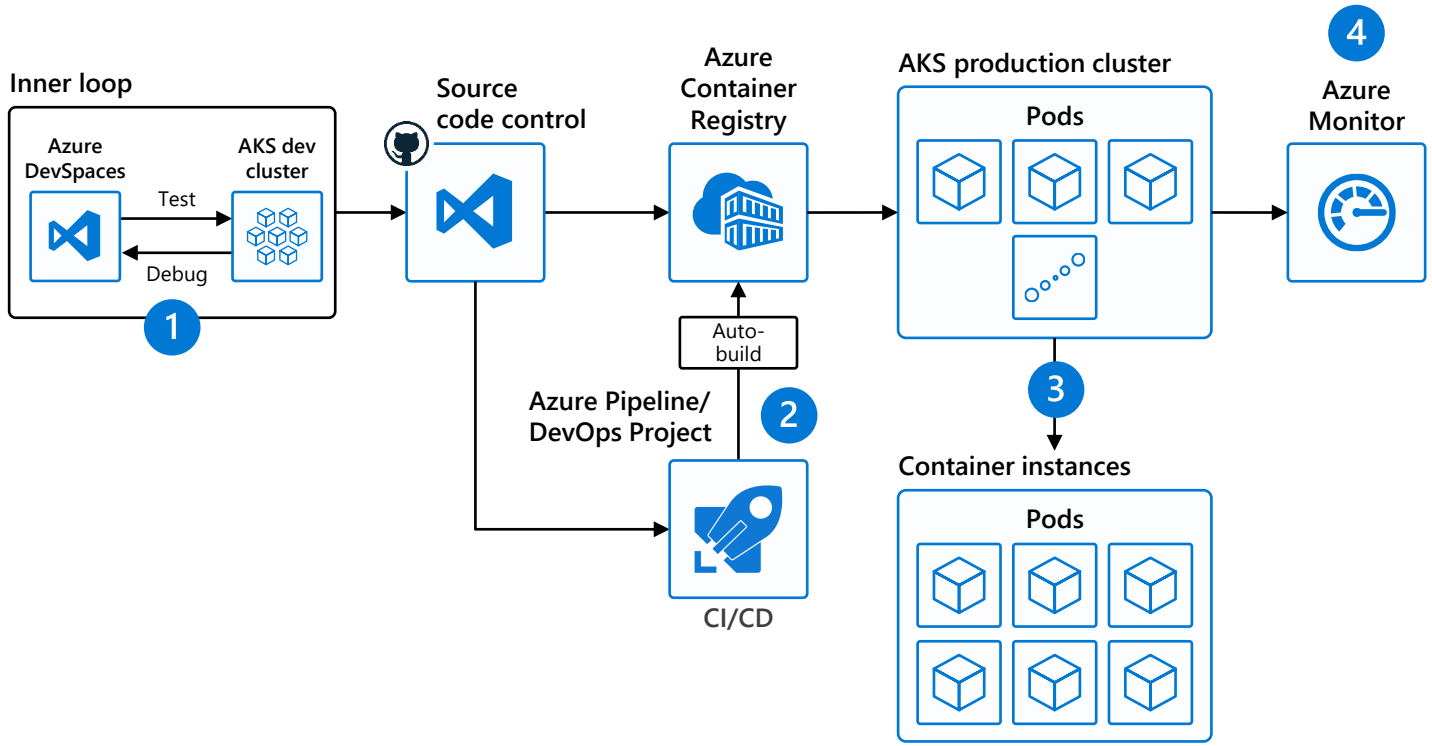
- Independent deployments
- Improved scale and resource utilization per service
- Smaller, focused teams



Microservices for faster app development

Capabilities

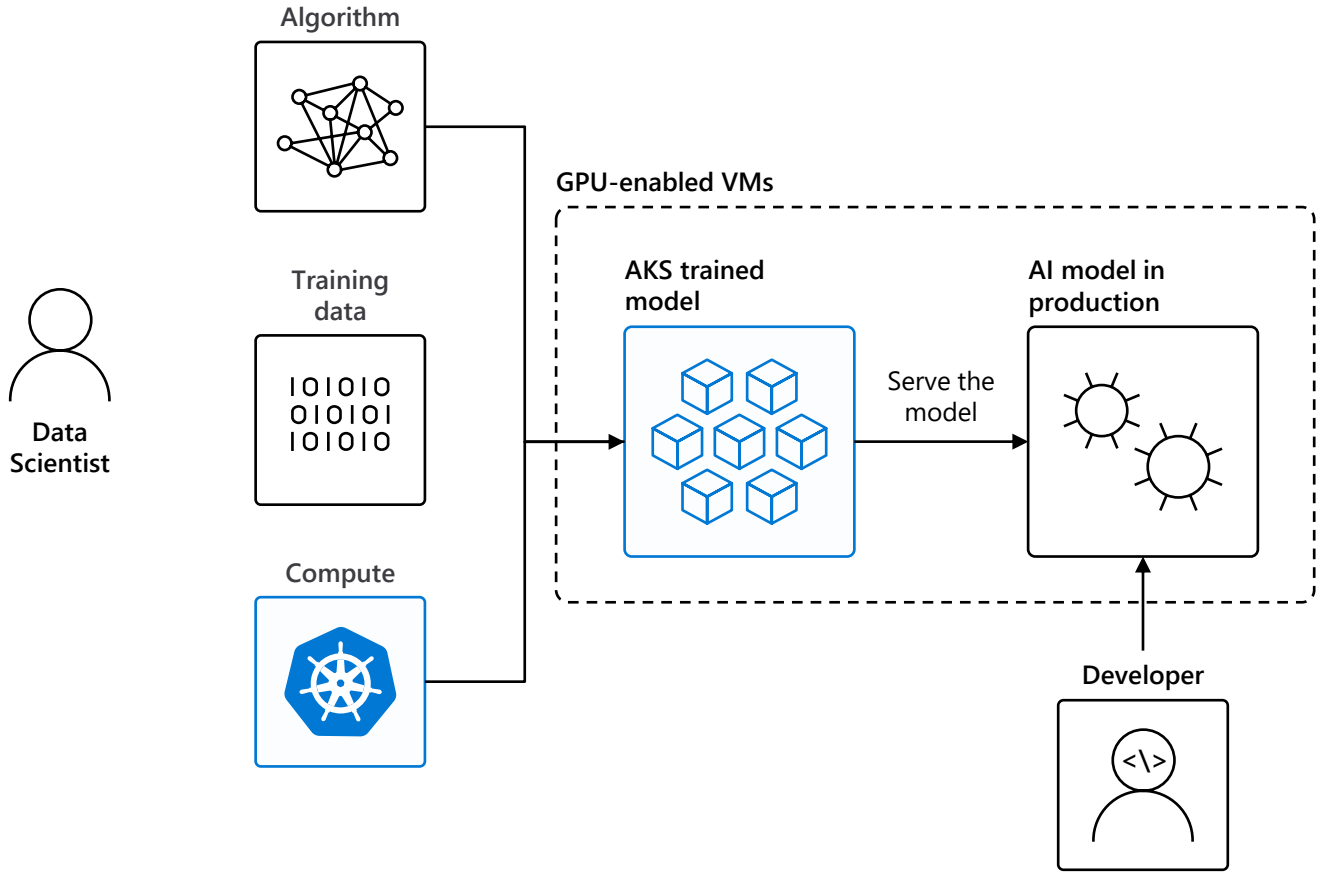
1. Use **Azure Dev Spaces** to iteratively develop, test, and debug microservices targeted for AKS clusters.
2. **Azure DevOps** has native integration with Helm and helps simplifying continuous integration/continuous delivery (CI/CD)
3. **Virtual node**—a Virtual Kubelet implementation—allows fast scaling of services for unpredictable traffic.
4. **Azure Monitor** provides a single pane of glass for monitoring over app telemetry, cluster-to-container level health analytics.



https://github.com/Microsoft/SmartHotel360-AKS-DevSpaces-Demo

Data scientist in a box

- Quick deployment and high availability
- Low latency data processing
- Consistent environment across test, control and production

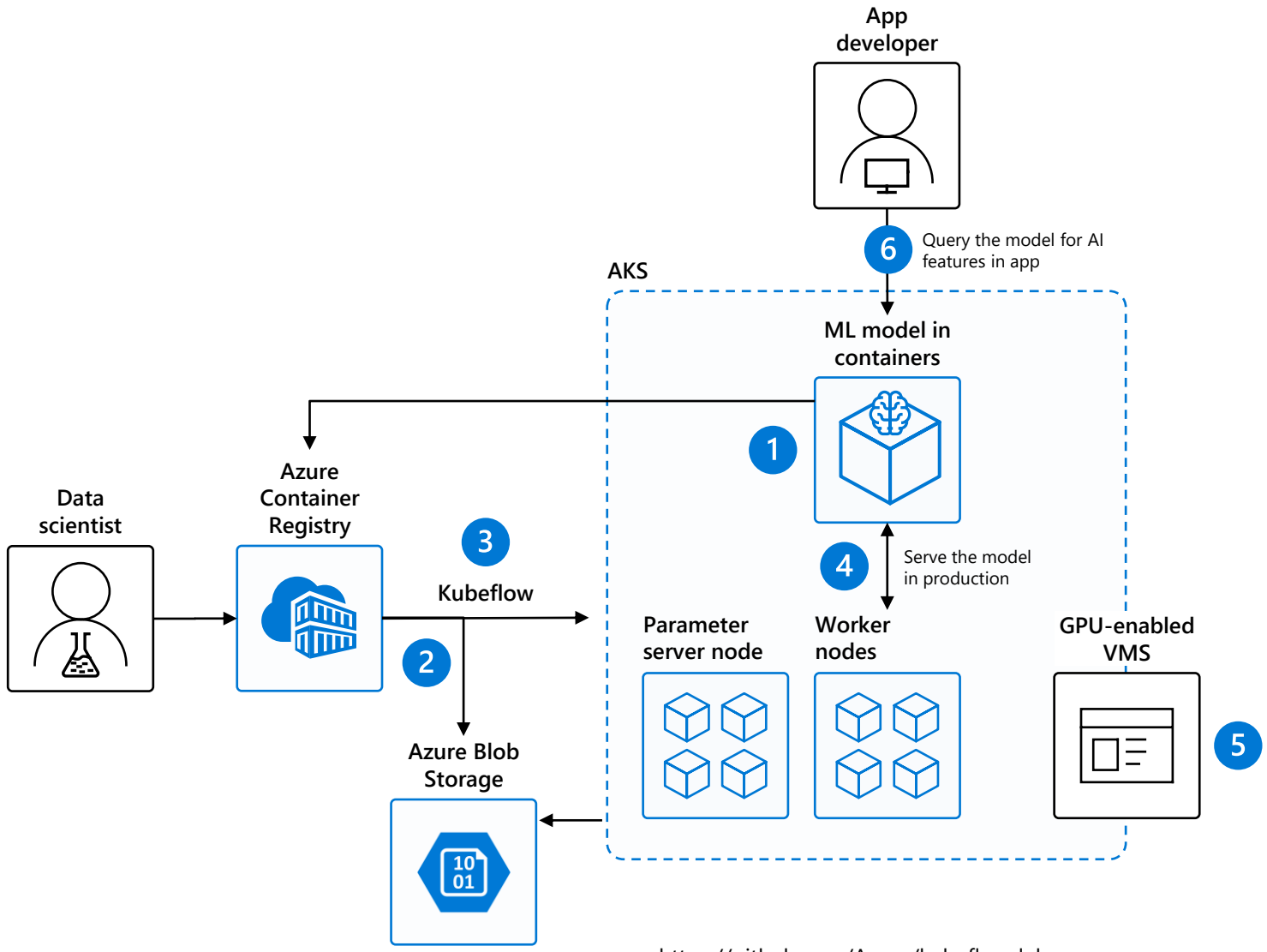


https://github.com/Azure/kubeflow-labs

Data scientist in a box

Capabilities

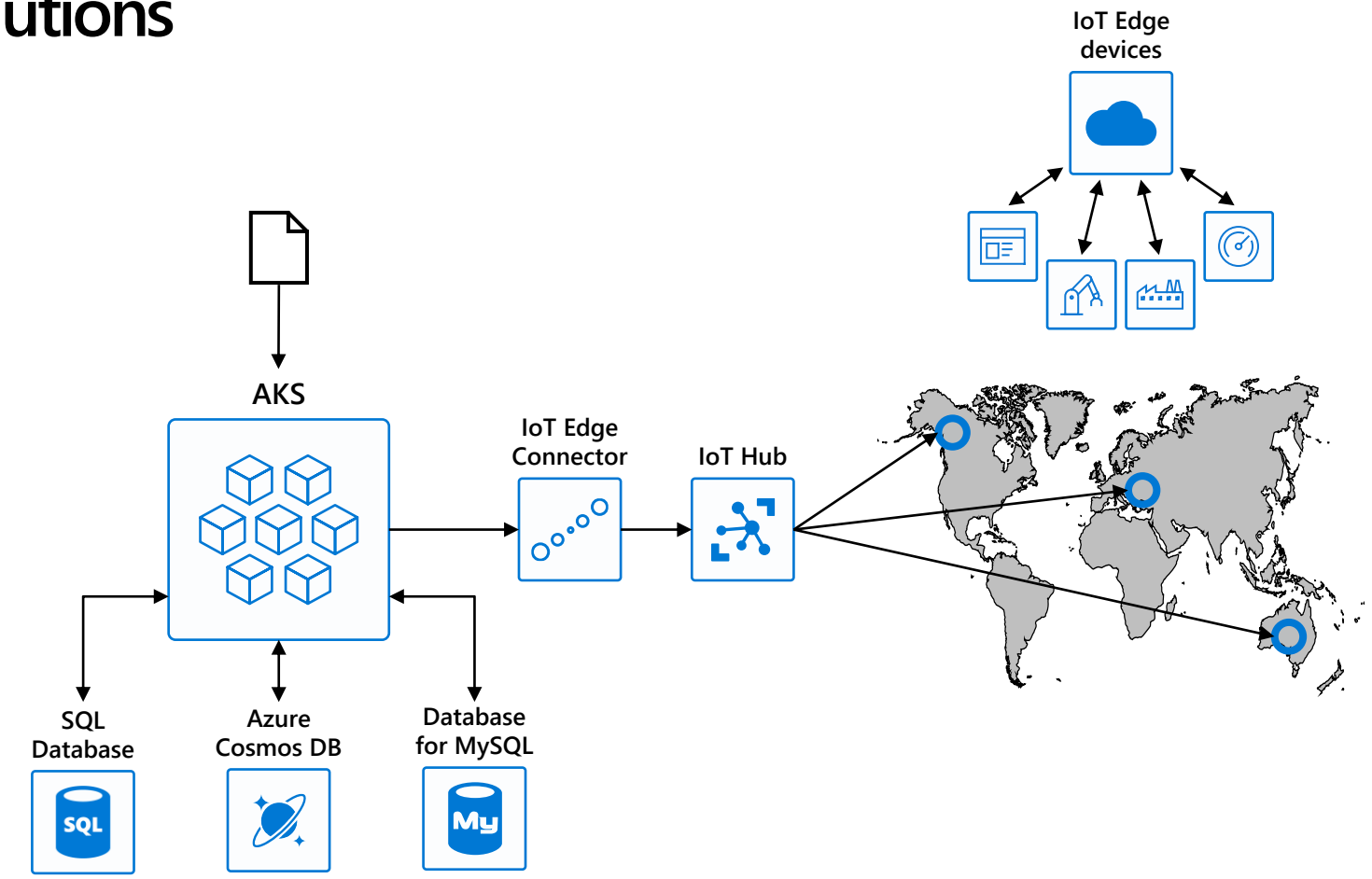
1. Package ML model into a container and publish to **Azure Container Registry**
2. **Azure Blob Storage** hosts training data sets and trained model
3. Use **Kubeflow** to deploy training job to AKS, distributed training job to AKS includes Parameter servers and Worker nodes
4. Serve production model using **Kubeflow**, promoting a consistent environment across test, control and production
5. AKS supports **GPU enabled VM**
6. Developer can build features querying the model running in AKS cluster



https://github.com/Azure/kubeflow-labs

Scalable Internet of Things solutions

- Portable code, runs anywhere
- Elastic scalability and manageability
- Quick deployment and high availability



Scalable Internet of Things solutions

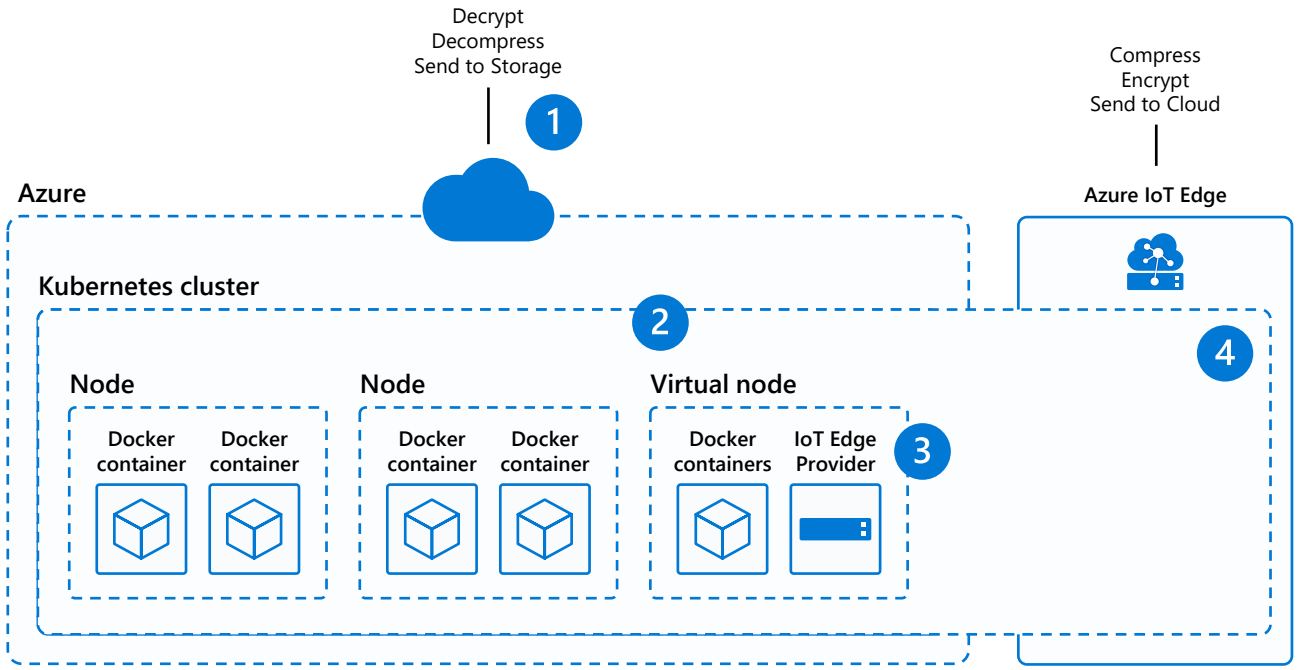
Capabilities

1. **Azure IoT Edge** encrypts data and send to Azure, which then decrypts the data and send to storage

2. **Virtual node**, an implementation of Virtual Kubelet, serves as the translator between cloud and Edge

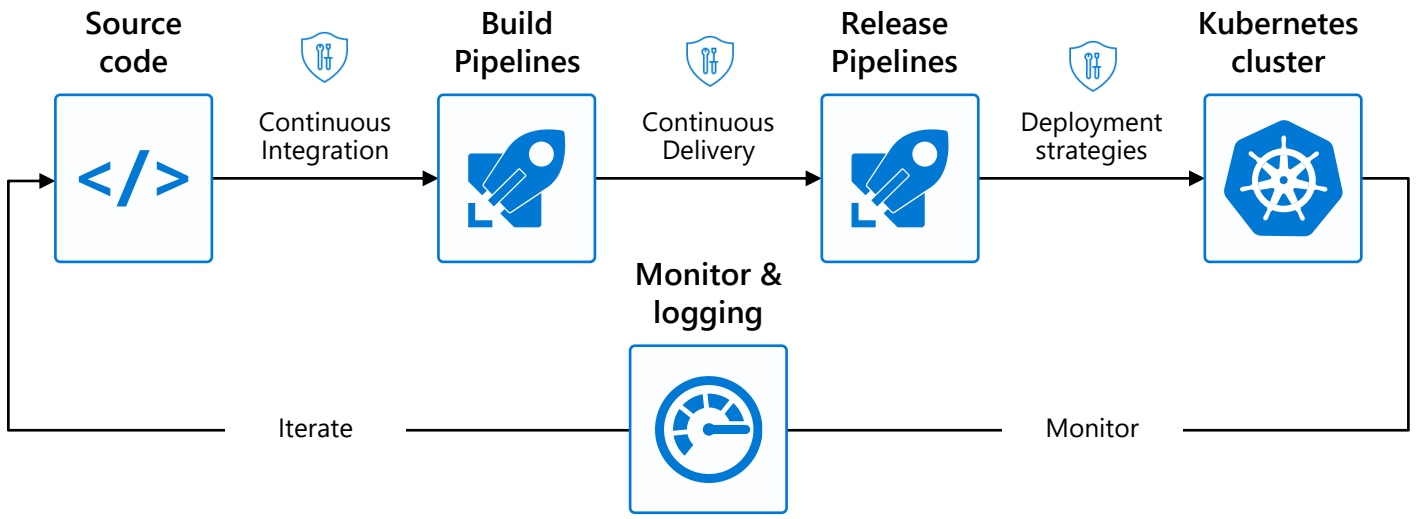
3. **IoT Edge Provider in virtual node** redirects containers to IoT Edge and extend AKS cluster to target millions of edge devices

4. Consistent update, manage, and monitoring as one unit in AKS using single pod definition



Secure DevOps

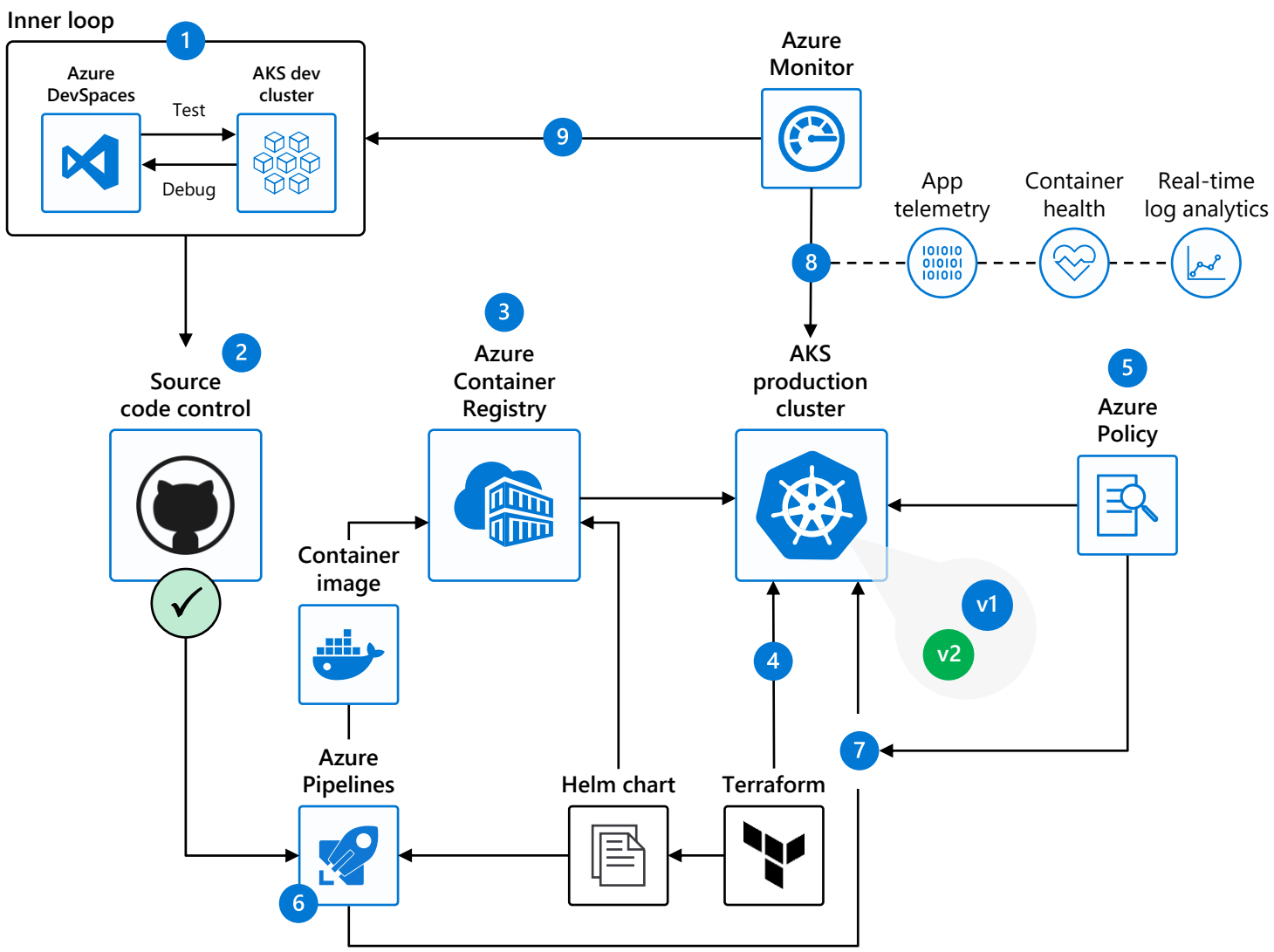
- Deliver code faster with Kubernetes and CI/CD
- Accelerate the feedback loop with constant monitoring
- Balance speed and security with continuous security and deep traceability



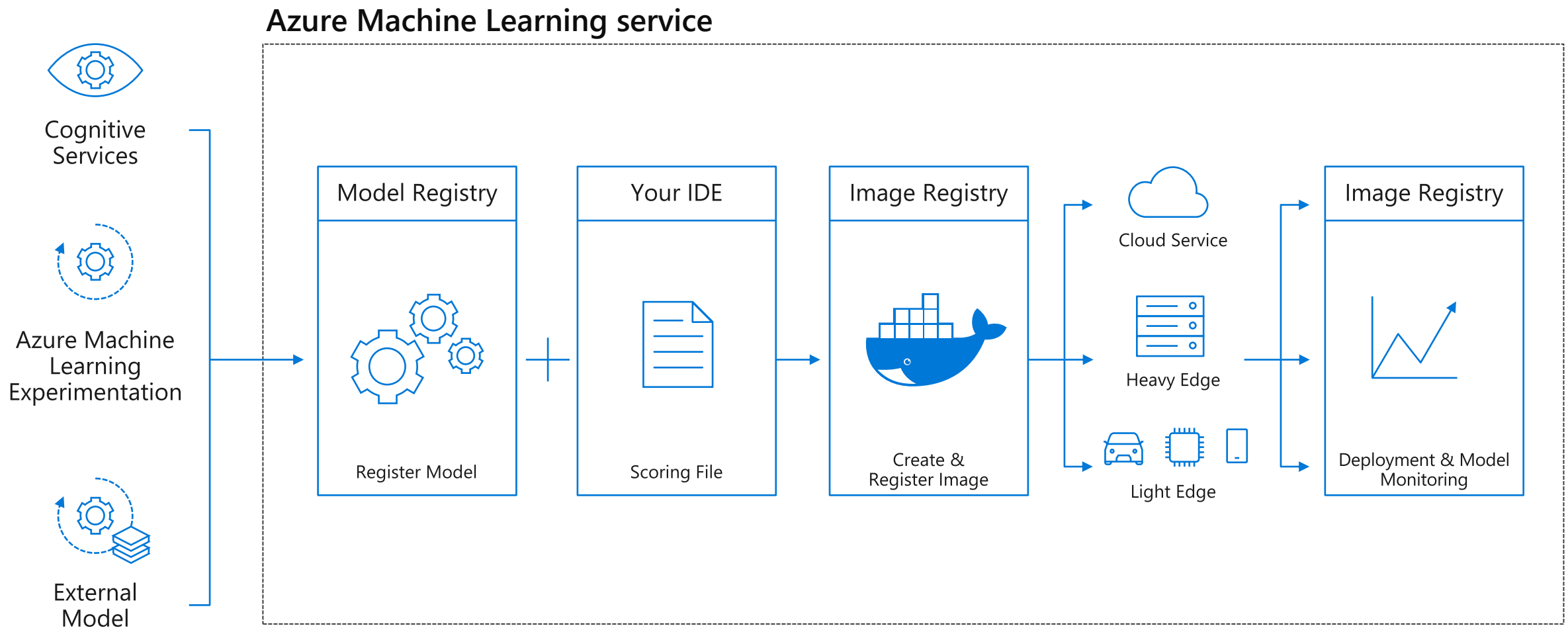
Secure DevOps

Capabilities

1. Developers rapidly iterate, test, and debug different parts of an application together in the same Kubernetes cluster
2. Code is merged into a GitHub repository, after which automated builds and tests are run by Azure Pipelines
3. Container image is pushed to Azure Container Registry
4. Kubernetes clusters are provisioned using tools like Terraform; Helm charts, installed by Terraform, define the desired state of app resources and configurations
5. Operators enforce policies to govern deployments to the AKS cluster
6. Release pipeline automatically executes pre-defined deployment strategy with each code change
7. Policy enforcement and auditing is added to CI/CD pipeline using Azure Policy
8. App telemetry, container health monitoring, and real-time log analytics are obtained using Azure Monitor
9. Insights used to address issues and fed into next sprint plans

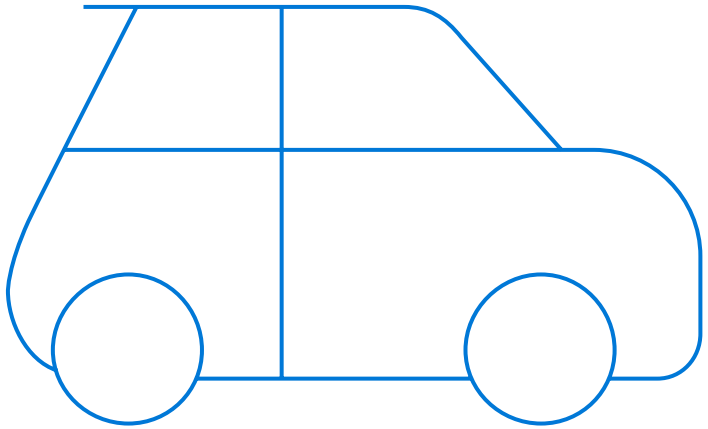


Case: Deploy Azure ML models at scale



Building your own AI models

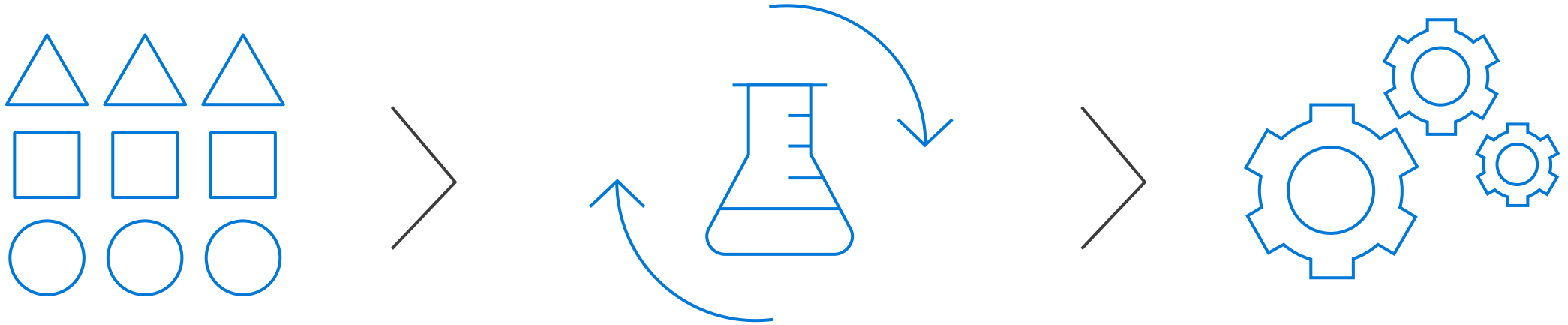
Transforming Data into Intelligence



Q: How much is this car worth?

Building your own AI models

Transforming data into intelligence



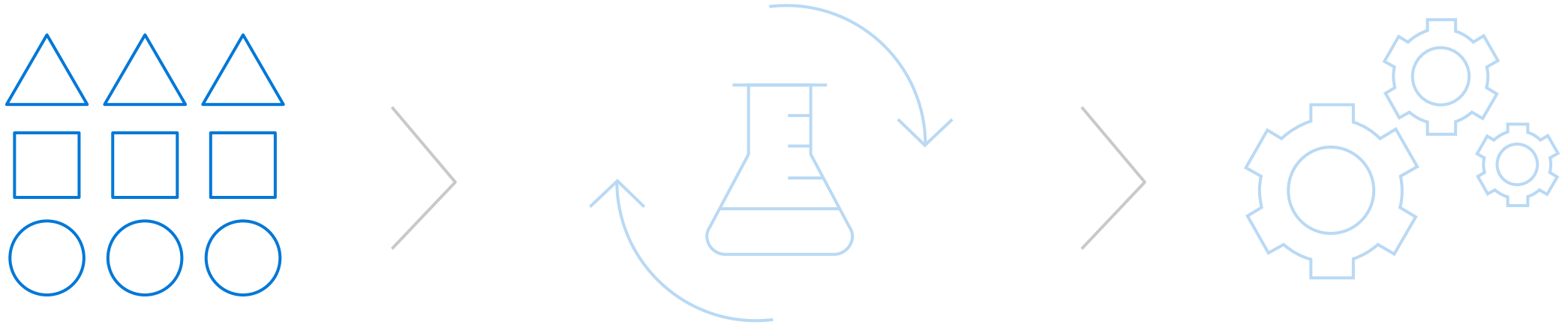
Prepare data

Build and train

Deploy

Building your own AI models

Transforming data into intelligence



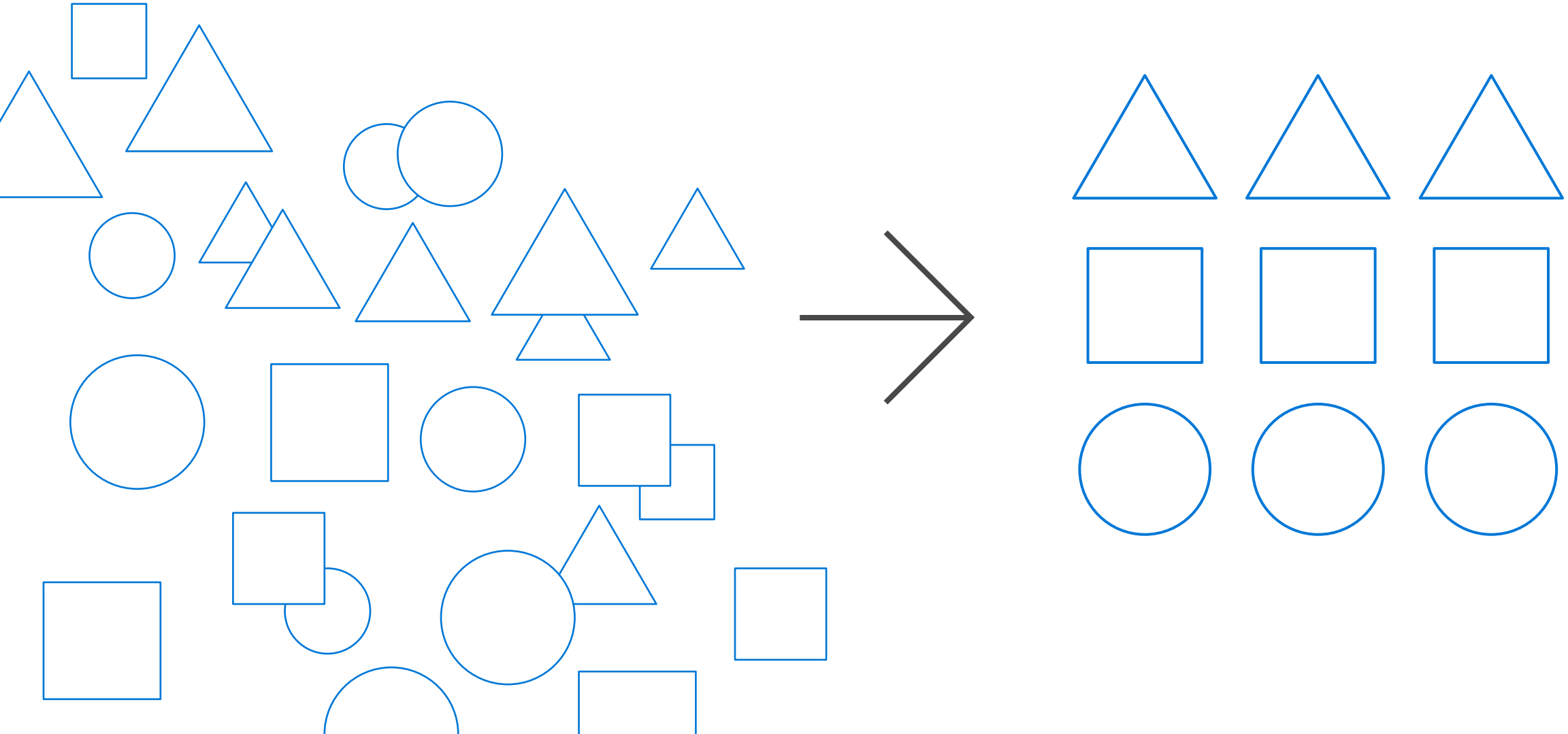
Prepare data

Build and train

Deploy

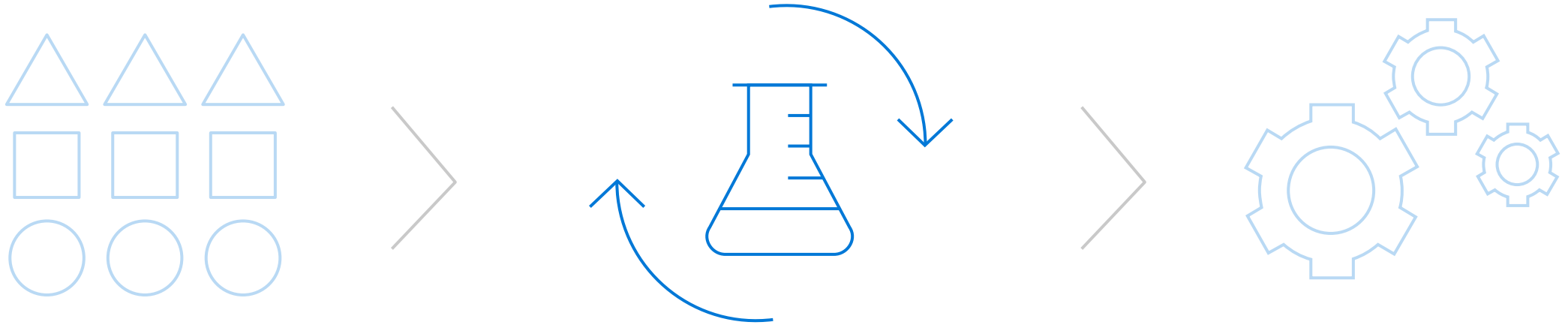
Building your own AI models

Step 1: Prepare data



Building your own AI models

Transforming data into intelligence



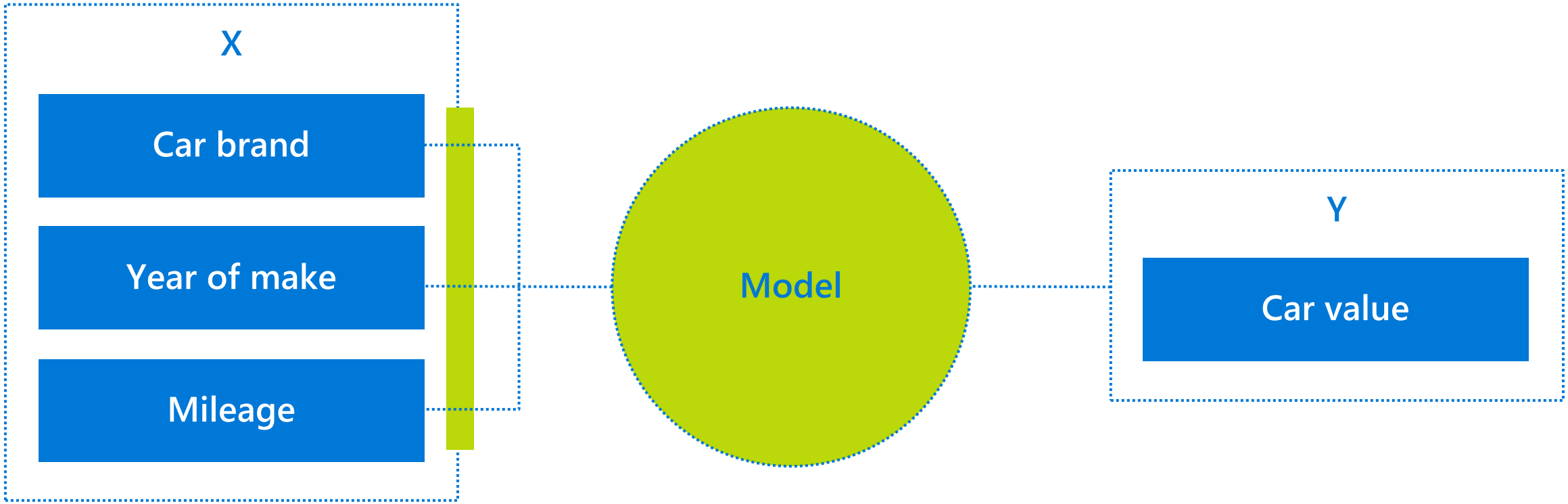
Prepare data

Build and train

Deploy

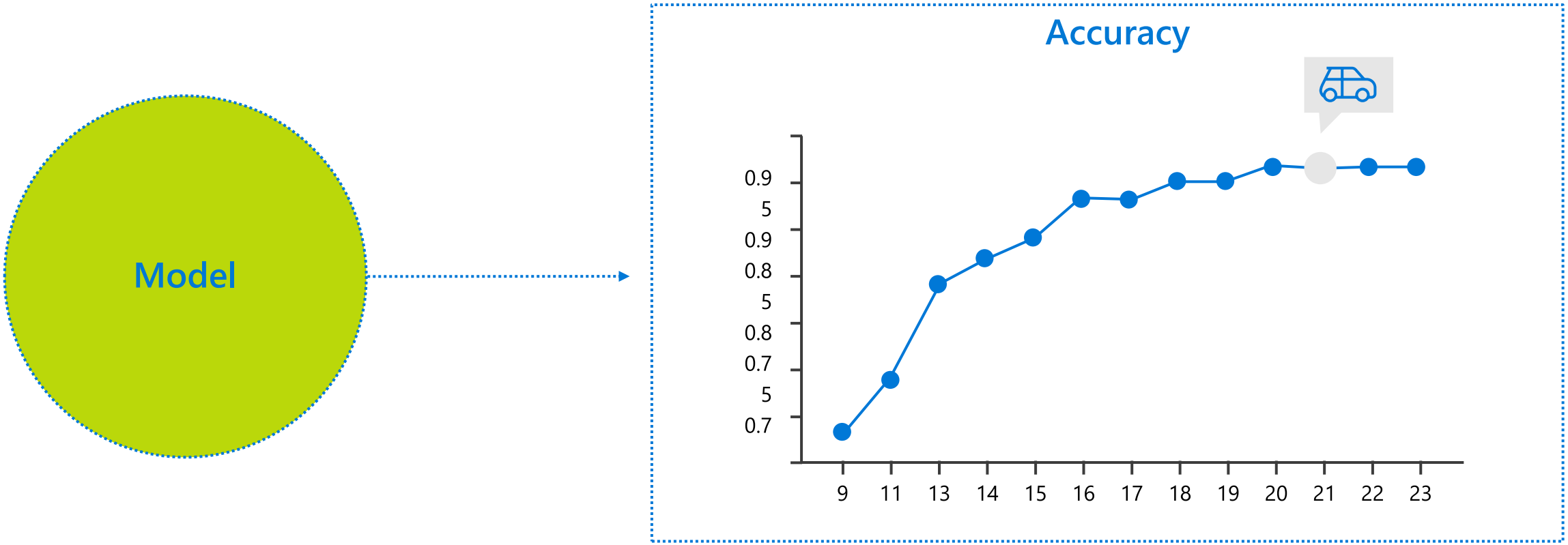
Building your own AI models

Step 2: Build and Train



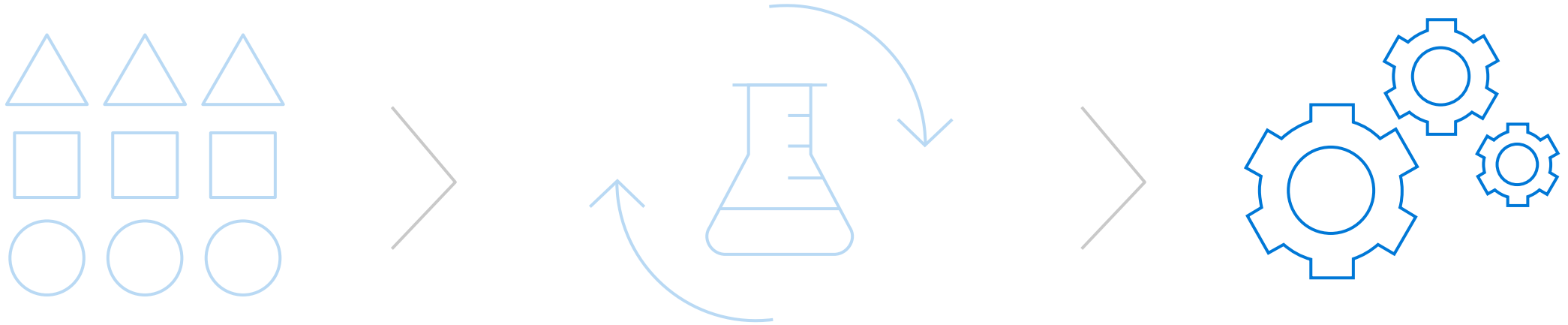
Building your own AI models

Step 2: Build and train



Building your own AI models

Transforming data into intelligence



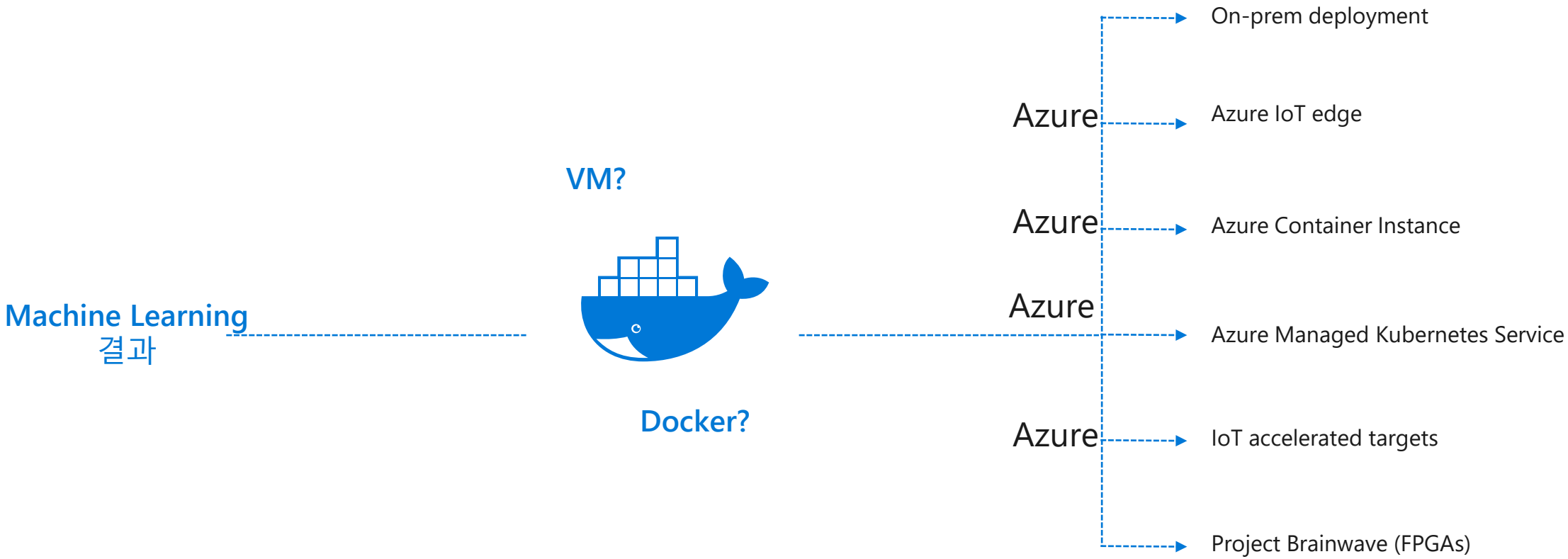
Prepare data

Build and train

Deploy

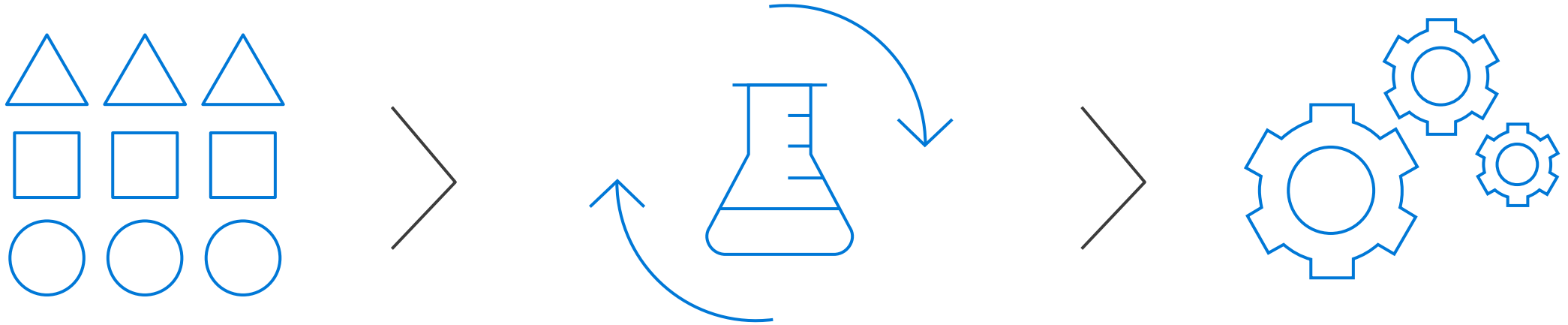
Building your own AI models

Step 3: Deploy



Building your own AI models

Transforming data into intelligence



Prepare data

Build and train

Deploy

Building your own AI models

Transforming data into intelligence

SQL DB

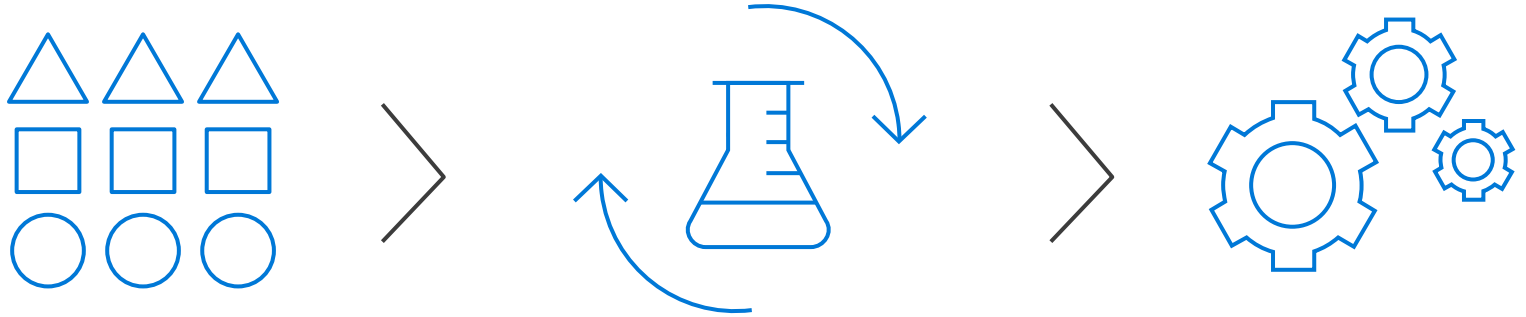
Cosmos DB

Datawarehouse

Data lake

Blob storage

...



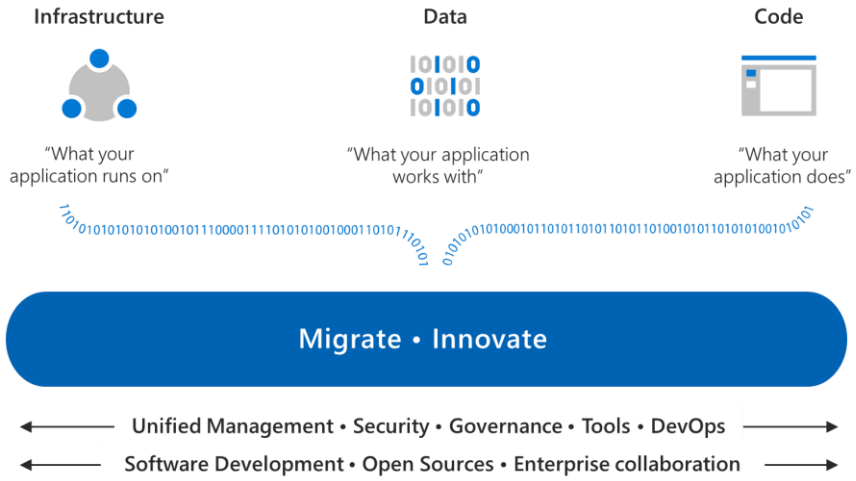
Prepare data

Build and train

Deploy

5. Closing

Cloud native journey with your cloud environment



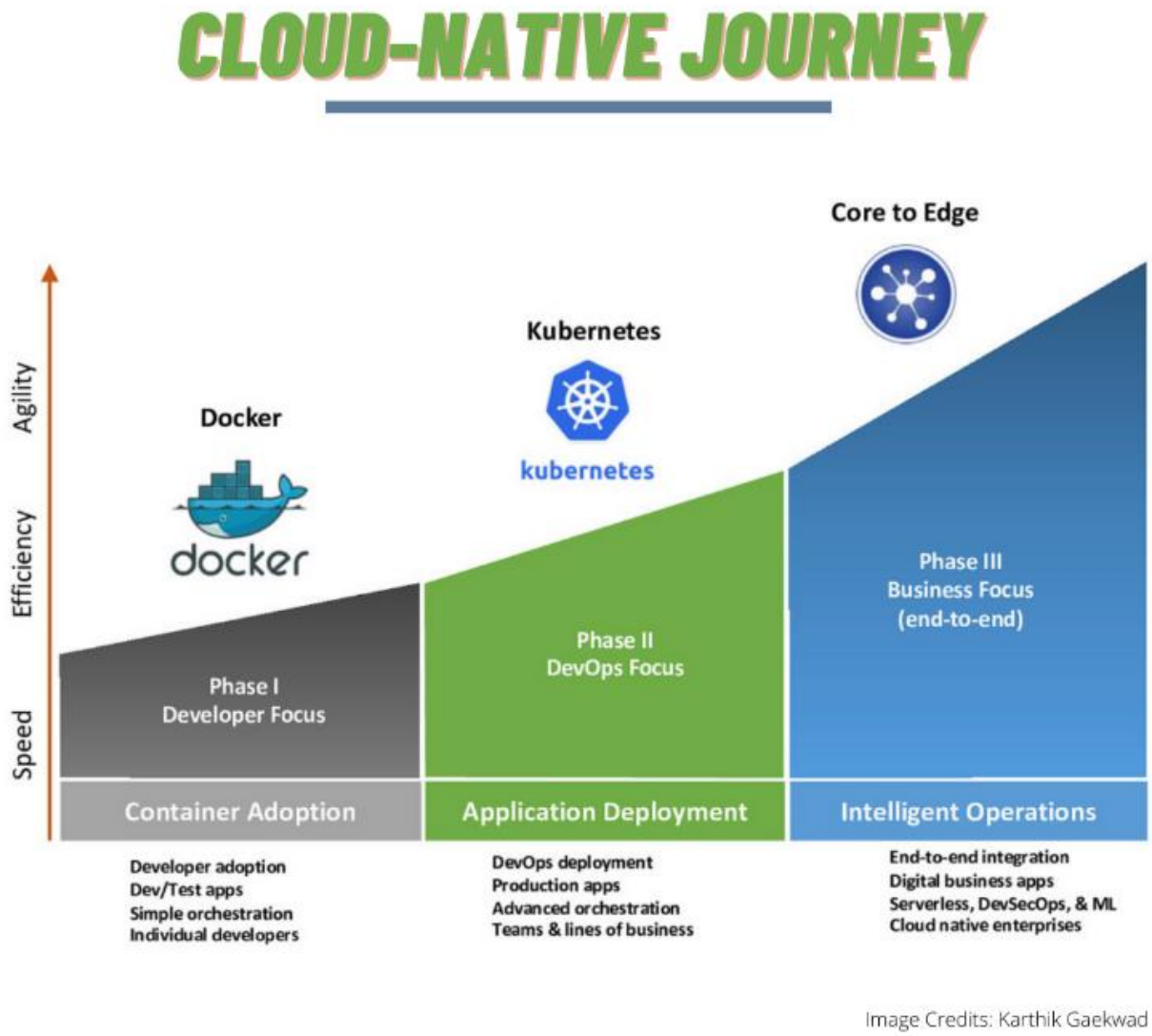


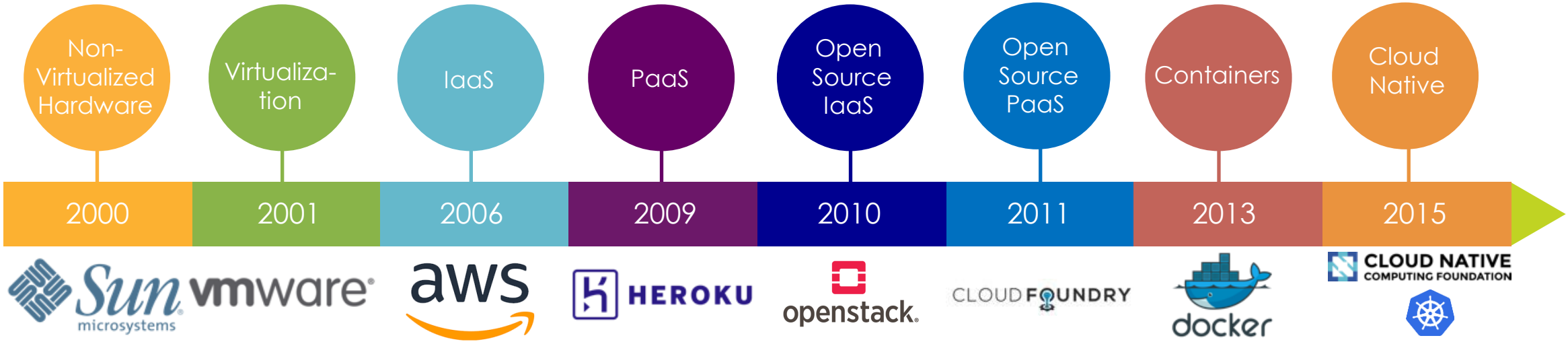
Image Credits: Karthik Gaekwad

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



(Source: Cloud Native Computing Foundation)

CNCF in 2021

Graduated Projects

Container Runtime

Service Discovery

Network Proxy

Key/Value Store

Logging

Registry

Package Management

Distributed Tracing

Policy

Orchestration

Monitoring

Storage

Key/Value Store

Software Update Spec

Storage

Incubating Projects

Continuous Integration & Deployment

Packaging Spec

Serverless

Networking API

Ingress Controller

Monitoring

Container Runtime

Image Distribution

Container Security

Remote Procedure Call

Edge

Service Mesh

Messaging

Security

GitOps

Distributed Tracing API

Operator Lifecycle Management

Identity Spec

Identity

Monitoring

(Source: Cloud Native Computing Foundation)

CNCF in 2021

Sandbox Projects

(Source: Cloud Native Computing Foundation)

CNCF is End User driven open source - 140+ organizations

(Source: Cloud Native Computing Foundation)

Plus 14 non-public members

The world's largest cloud and software companies

Platinum Members

Gold Members

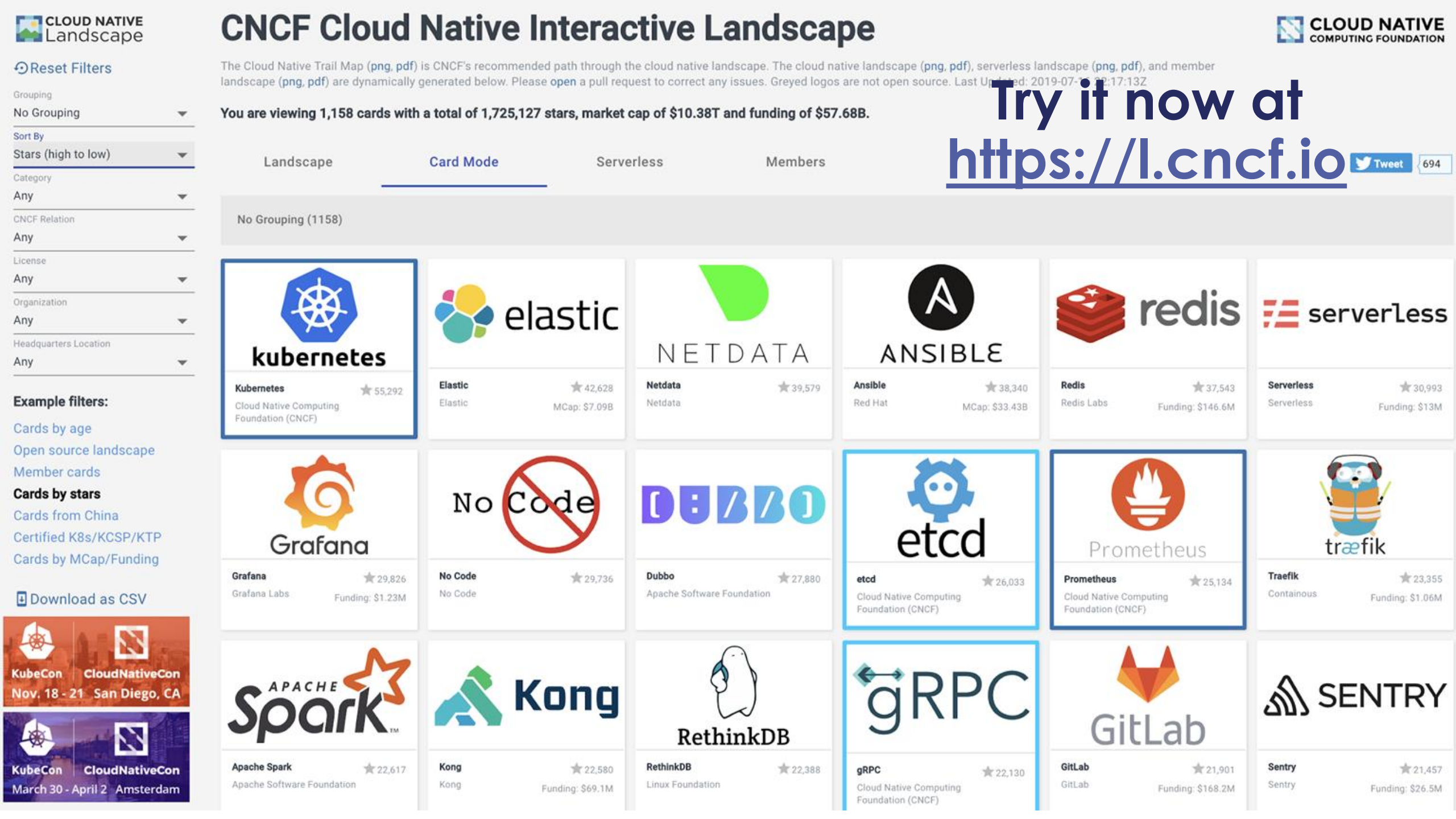
Academic/Nonprofit Members

(Source: Cloud Native Computing Foundation)

and innovative startups - 630+ members

Silver Members

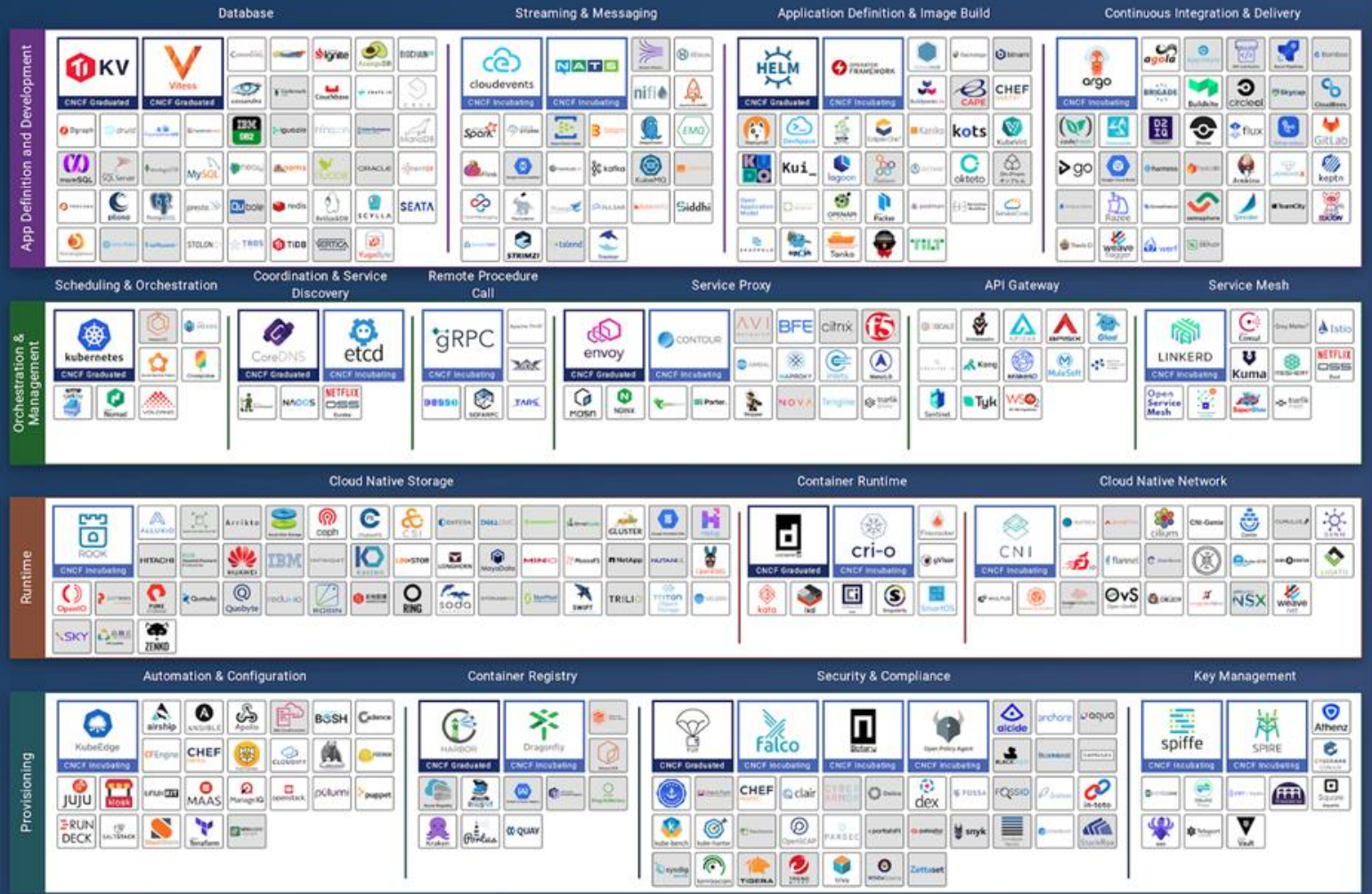
[See all Silver members in the member landscape!]

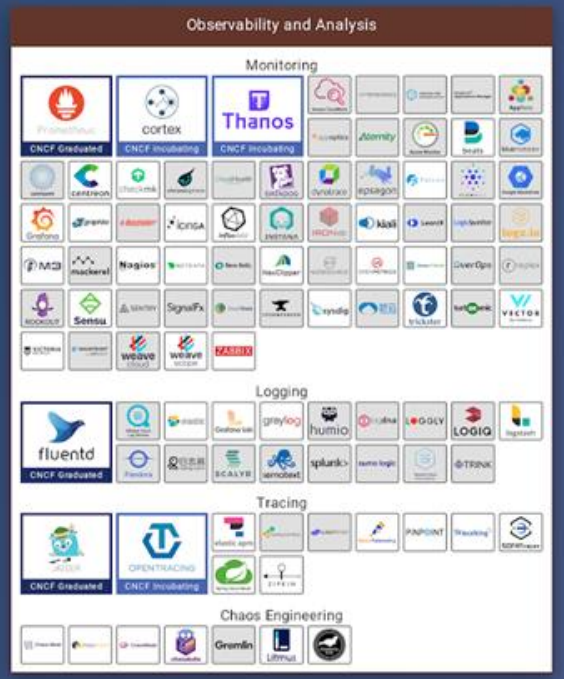

CLOUD NATIVE Landscape
CNCF Cloud Native Interactive Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Reset Filters
Grouping
No Grouping
Sort By
Stars (high to low)
Category
Any
CNCF Relation
Any
License
Any
Organization
Any
Headquarters Location
Any
Example filters:
Cards by age
Open source landscape
Member cards
Cards by stars
Cards from China
Certified K8s/KCSP/KTP
Cards by MCap/Funding
Download as CSV
The Cloud Native Trail Map (png, pdf) is CNCF's recommended path through the cloud native landscape. The cloud native landscape (png, pdf), serverless landscape (png, pdf), and member landscape (png, pdf) are dynamically generated below. Please open a pull request to correct any issues. Greyed logos are not open source. Last Updated: 2019-07-31 08:17:13Z
You are viewing 1,158 cards with a total of 1,725,127 stars, market cap of $10.38T and funding of $57.68B.
Try it now at https://l.cncf.io
Tweet 694
Landscape
Card Mode
Serverless
Members
No Grouping (1158)
kubernetes
Kubernetes 55,292
Cloud Native Computing Foundation (CNCF)
elastic
Elastic 42,628
Elastic MCap: $7.09B
NETDATA
Netdata 39,579
Netdata
ANSIBLE
Ansible 38,340
Red Hat MCap: $33.43B
redis
Redis 37,543
Redis Labs Funding: $146.6M
serverless
Serverless 30,993
Serverless Funding: $13M
Grafana
Grafana 29,826
Grafana Labs Funding: $1.23M
No Code
No Code 29,736
No Code
Dubbo
Dubbo 27,880
Apache Software Foundation
etcd
etcd 26,033
Cloud Native Computing Foundation (CNCF)
Prometheus
Prometheus 25,134
Cloud Native Computing Foundation (CNCF)
træfik
Traefik 23,355
Containous Funding: $1.06M
APACHE Spark
Apache Spark 22,617
Apache Software Foundation
Kong
Kong 22,580
Kong Funding: $69.1M
RethinkDB
RethinkDB 22,388
Linux Foundation
gRPC
gRPC 22,130
Cloud Native Computing Foundation (CNCF)
GitLab
GitLab 21,901
GitLab Funding: $168.2M
SENTRY
Sentry 21,457
Sentry Funding: $26.5M
KubeCon CloudNativeCon
Nov. 18 - 21 San Diego, CA
KubeCon CloudNativeCon
March 30 - April 2 Amsterdam

CNCF Cloud Native Landscape
2020-10-05T19:14:12Z e98f4c72

Overwhelmed? Please see the CNCF Trail Map. That and the interactive landscape are at l.cncf.io

Greyed logos are not open source

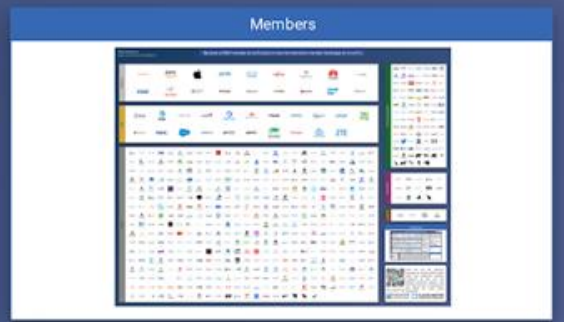




CLOUD NATIVE Landscape
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint Amplify

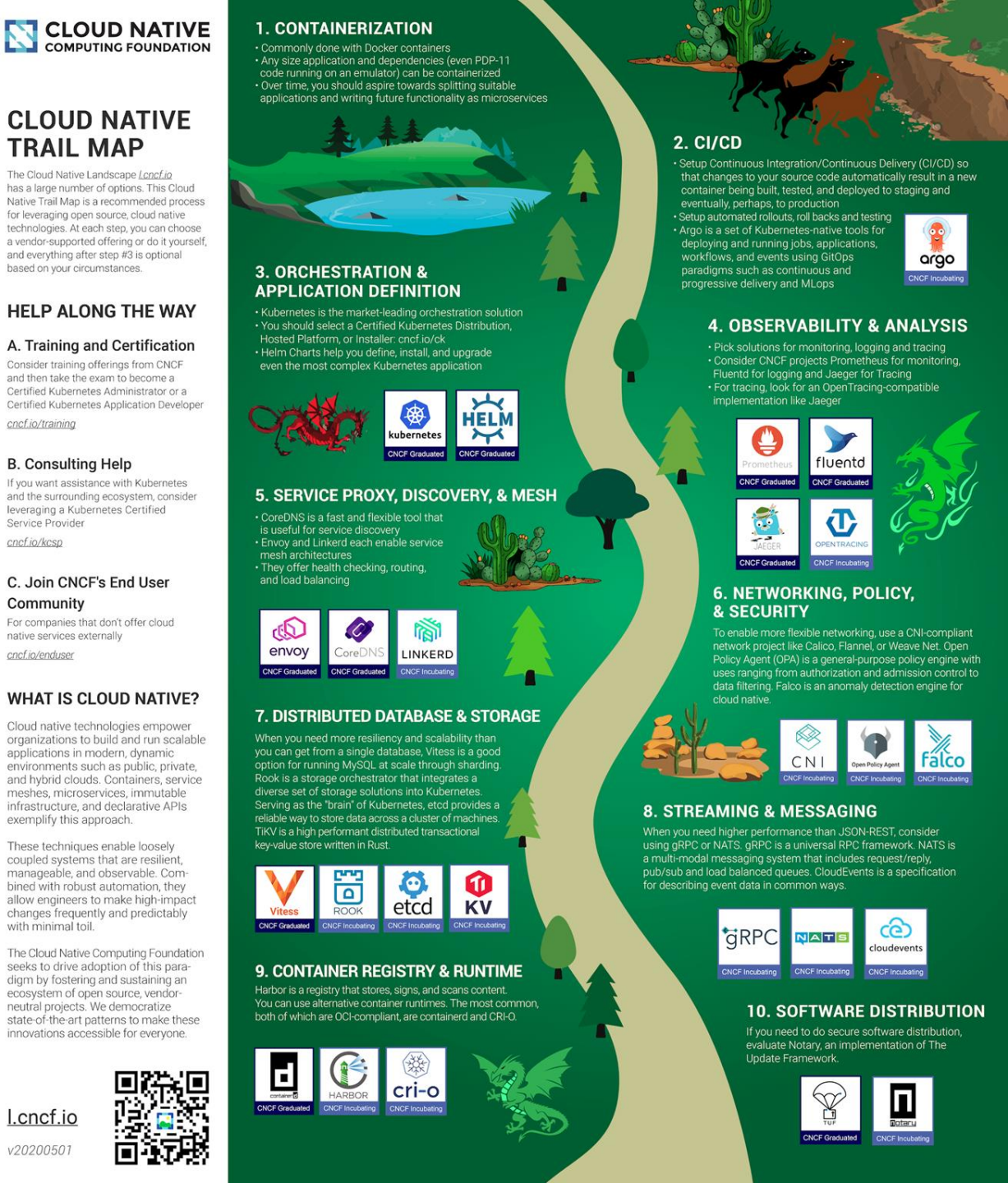
l.cncf.io

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.

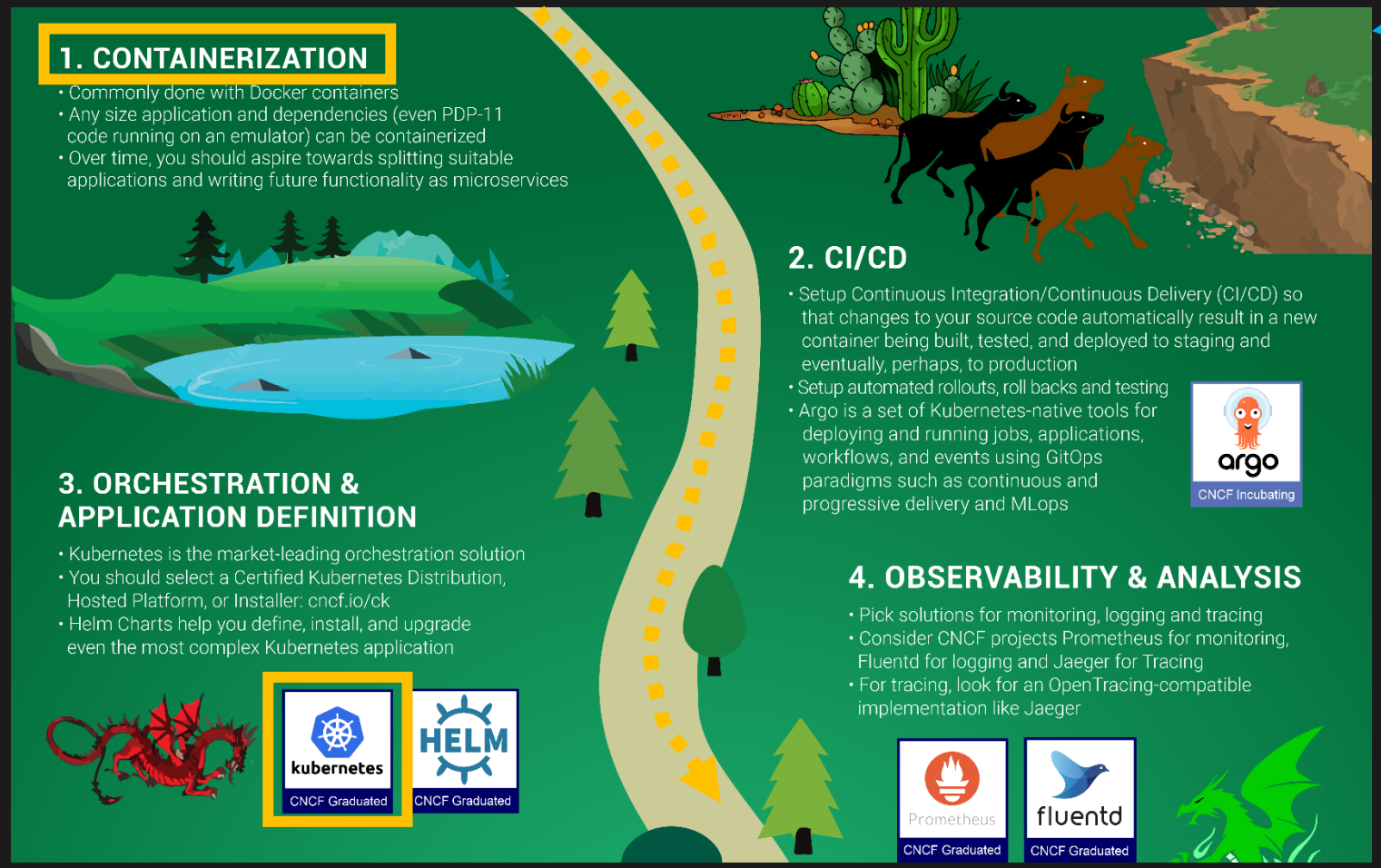


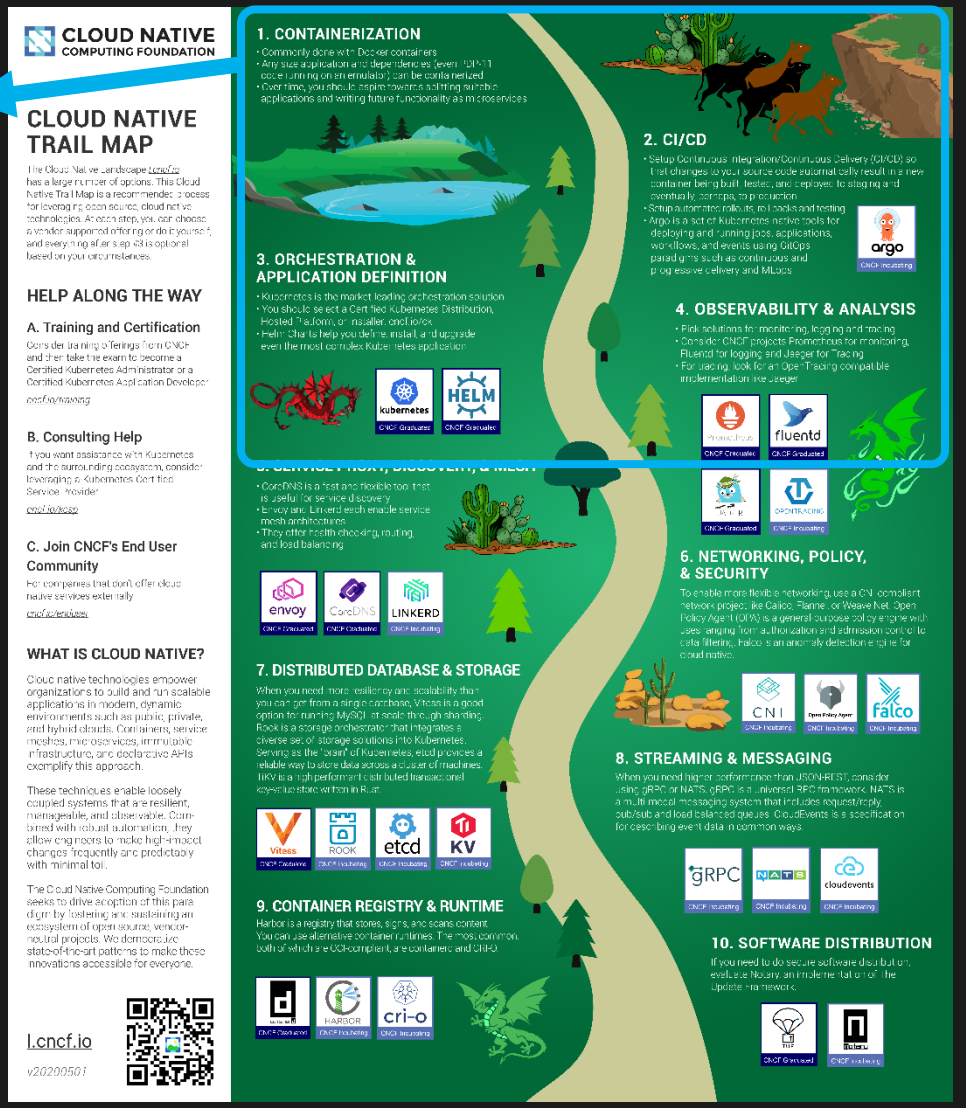
Cloud Native Trail Map

Trail Map: l.cncf.io

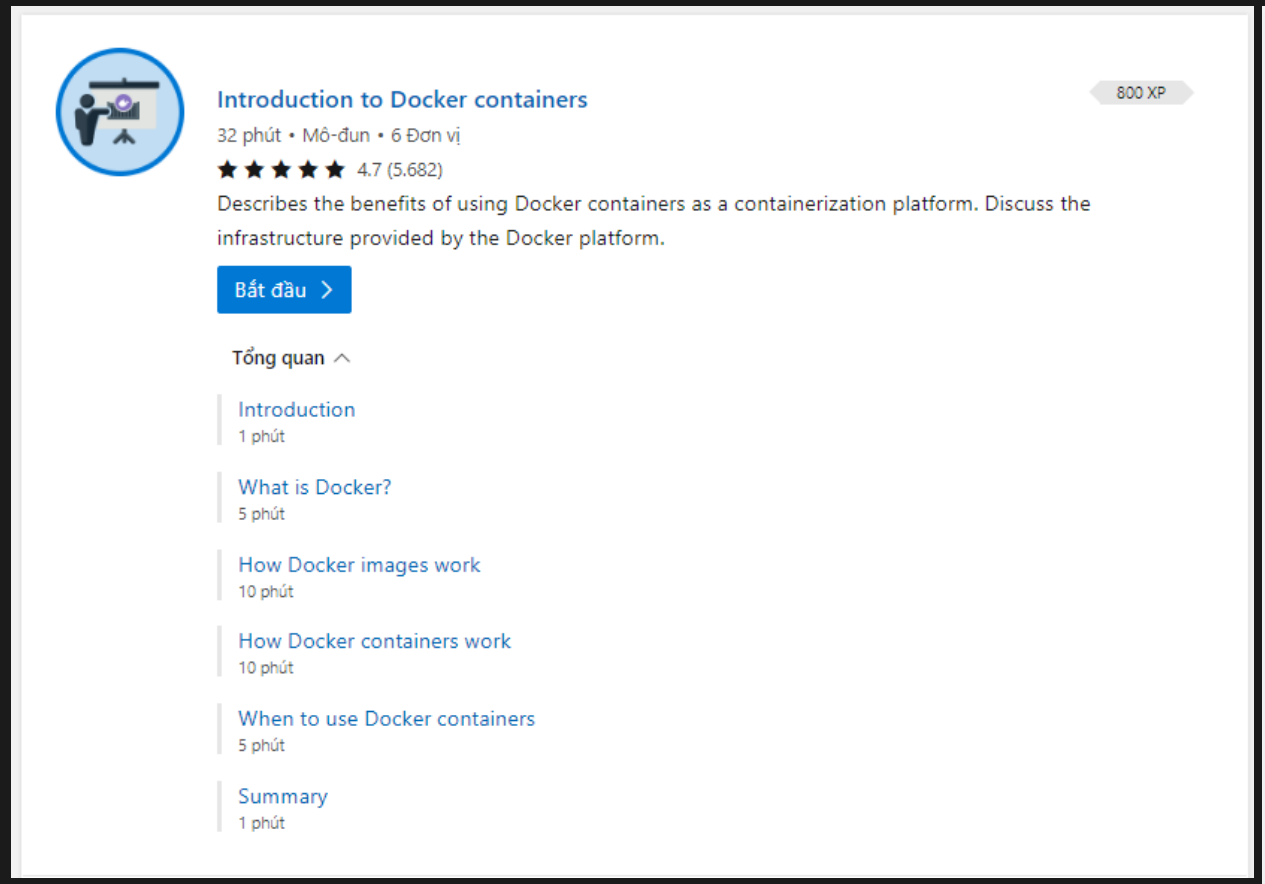


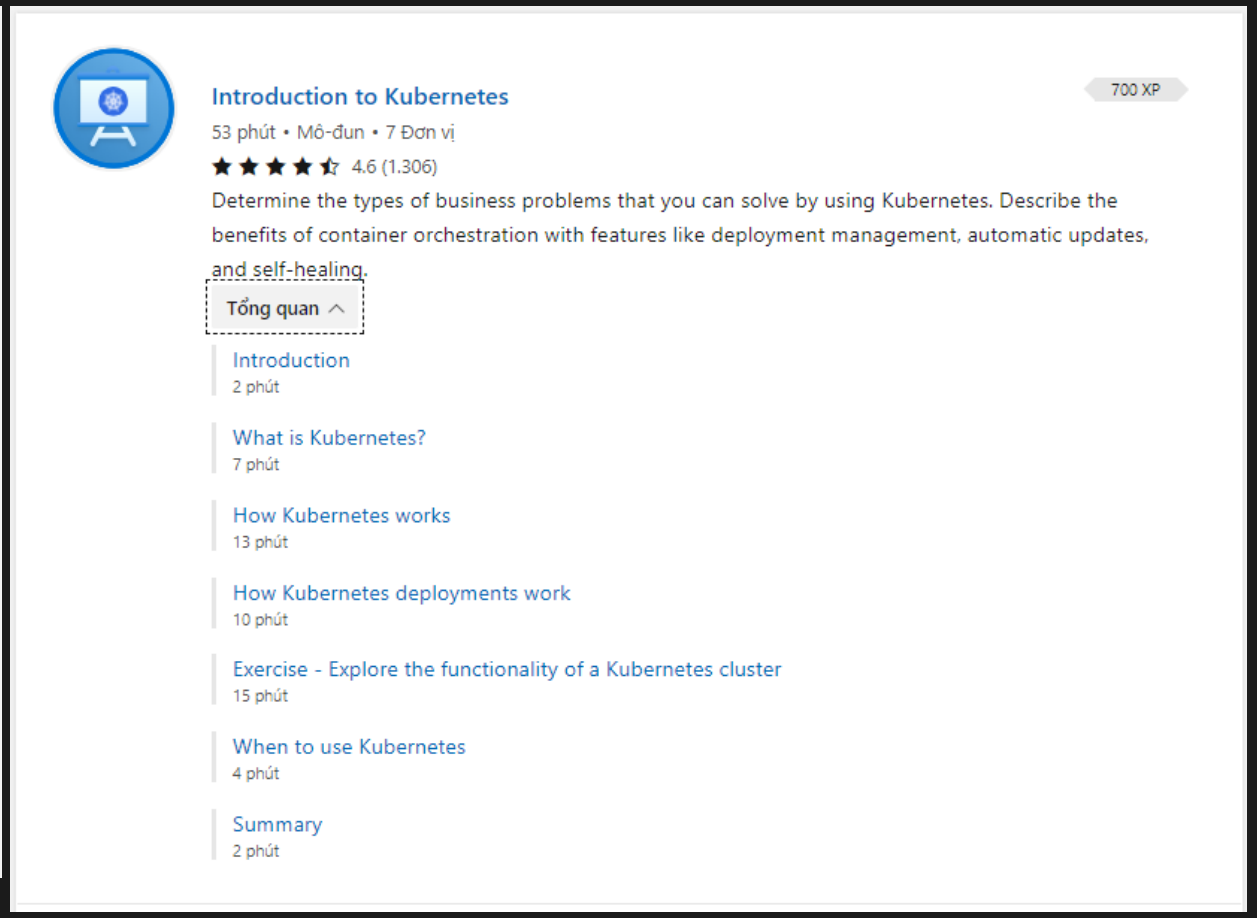
“Cloud Native”: Trail Map





How to more learn containers & Kubernetes?





https://aka.ms/learn-containers-and-aks-with-azure

Finished!

THANK YOU very much!