The IT revolution of containers

Marica Antonacci
(INFN Bari)

Outline

- What is cloud-native?
- DevOps
- What is containerization?
- Containers benefits

Three revolutions are changing the IT world

- The creation of the **cloud**
- The dawn of **DevOps**
- The wide adoption of **containers**

Together, these three waves of change are creating a new software world:

the cloud native world.

What is cloud-native

*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.*

Cloud Native Computing Foundation (CNCF)

Application development/deployment evolution

	Development Process	Application Architecture	Deployment and Packaging	Application Infrastructure
~ 1980 – ~ 1990	Waterfall	Monolithic	Physical Server	Datacenter
~ 2000	Agile	N-Tier	Virtual Servers	Hosted
~ 2010 – Now	DevOps	Microservices	Containers	Cloud

DevOps

The DevOps Workflow is all about **collaborating** together and finding ways to **automate** parts of the lifecycle.
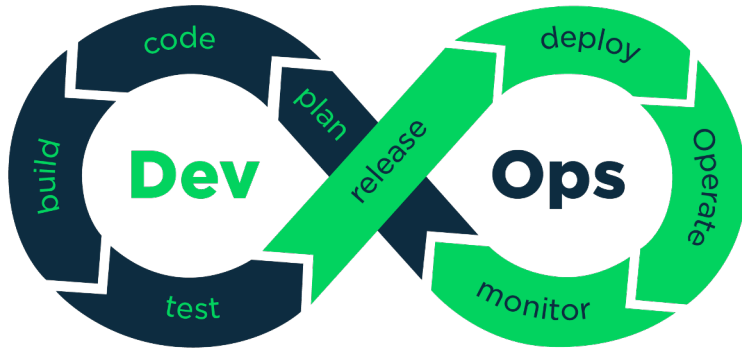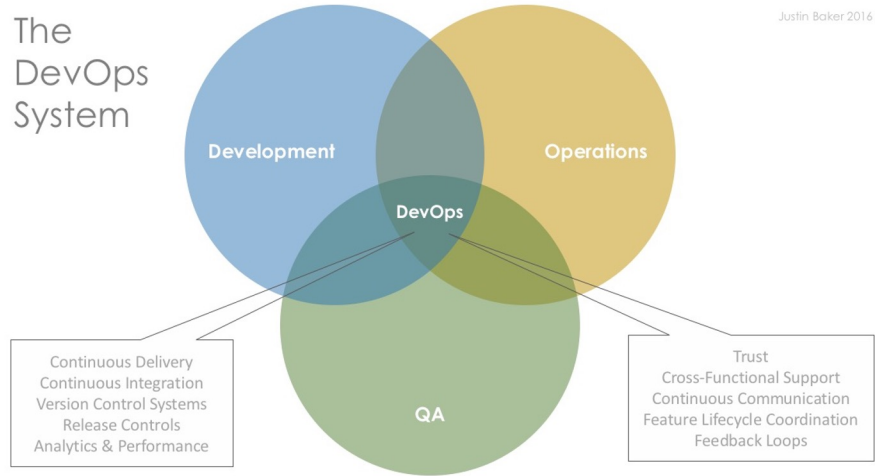
DevOps has been described variously as a culture, a mindset, a framework and a movement.

The CAMS model

Culture   Automation   Measurement   Sharing

CAMS is an acronym describing the core values of the DevOps Movement: Culture, Automation, Measurement, and Sharing. It was coined by Damon Edwards and John Willis at DevOpsDays Mountainview 2010

DevOps tools
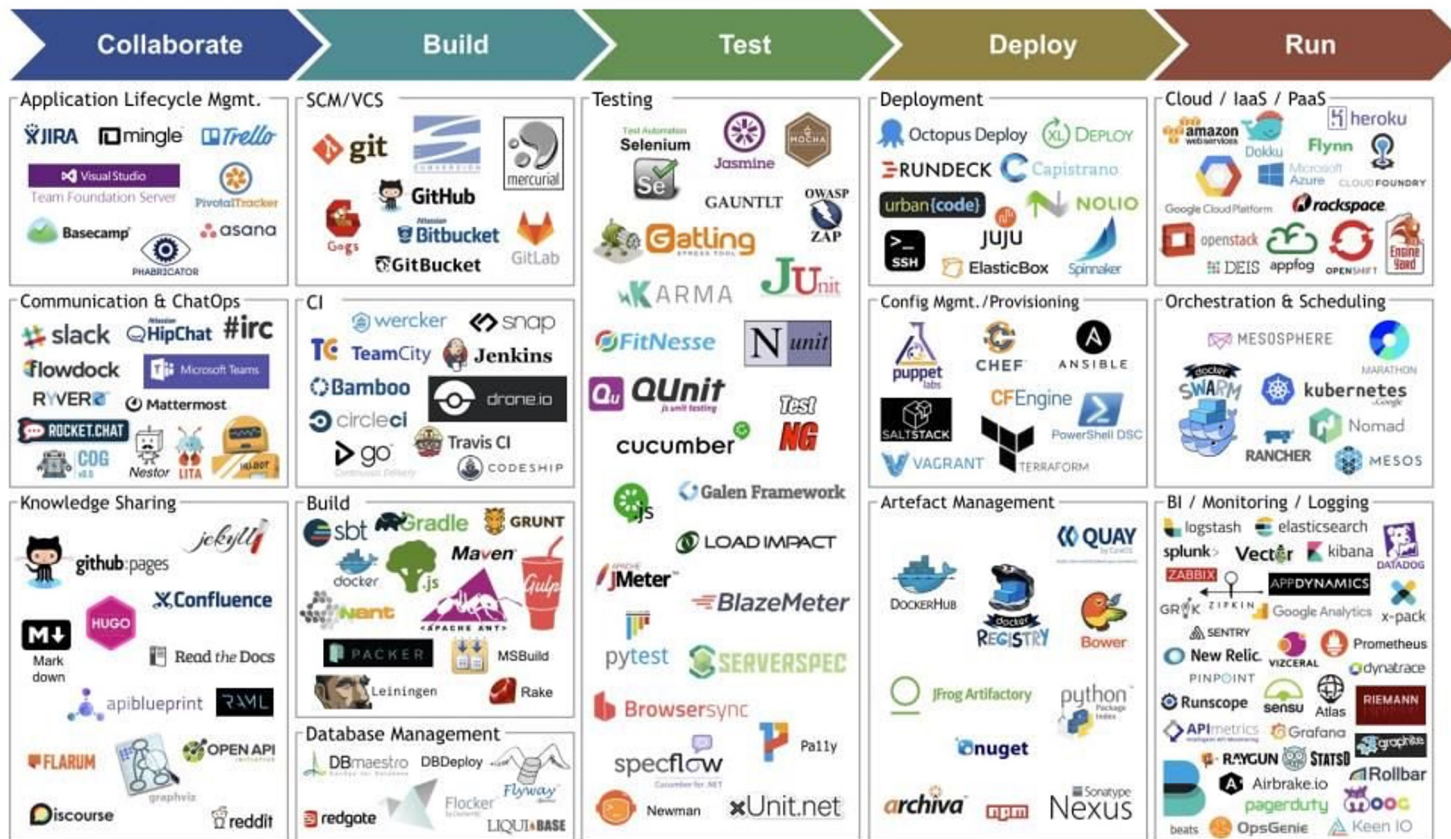
What is container(ization)?

The concept is borrowed from shipping containers, which define a standard to ship goods globally.

Containerization is a process of **packaging** your application together with its **dependencies** into one package (a container).

Such a package can then be run pretty much anywhere: on-premises server, virtual machine in the cloud, developer's laptop.

By abstracting the infrastructure, containerization allows you to make your application truly **portable** and **flexible**.

Container technology evolution

2004-07
Google uses containers (cgroups) at scale

MARCH 2013
Docker hits the scene: Containerization platform based on Linux; integrates natively with other OSes: Windows, macOS

2013 RESEARCH
30 containerization vendors vie for users (451 RESEARCH)

NOVEMBER 2014
Canonical launches LXD: Built on LXC with new features and functionality

DECEMBER 2014
CoreOS introduces rkt containers as Docker alternative

APRIL 2015
CoreOS first to provide commercial Kubernetes offering

JULY 2015
Kubernetes 1.0 released: Google gives Kubernetes to Cloud Native Computing Foundation (CNCF) for development

---

JANUARY 2008
cgroups merge into Linux kernel

AUGUST 2008
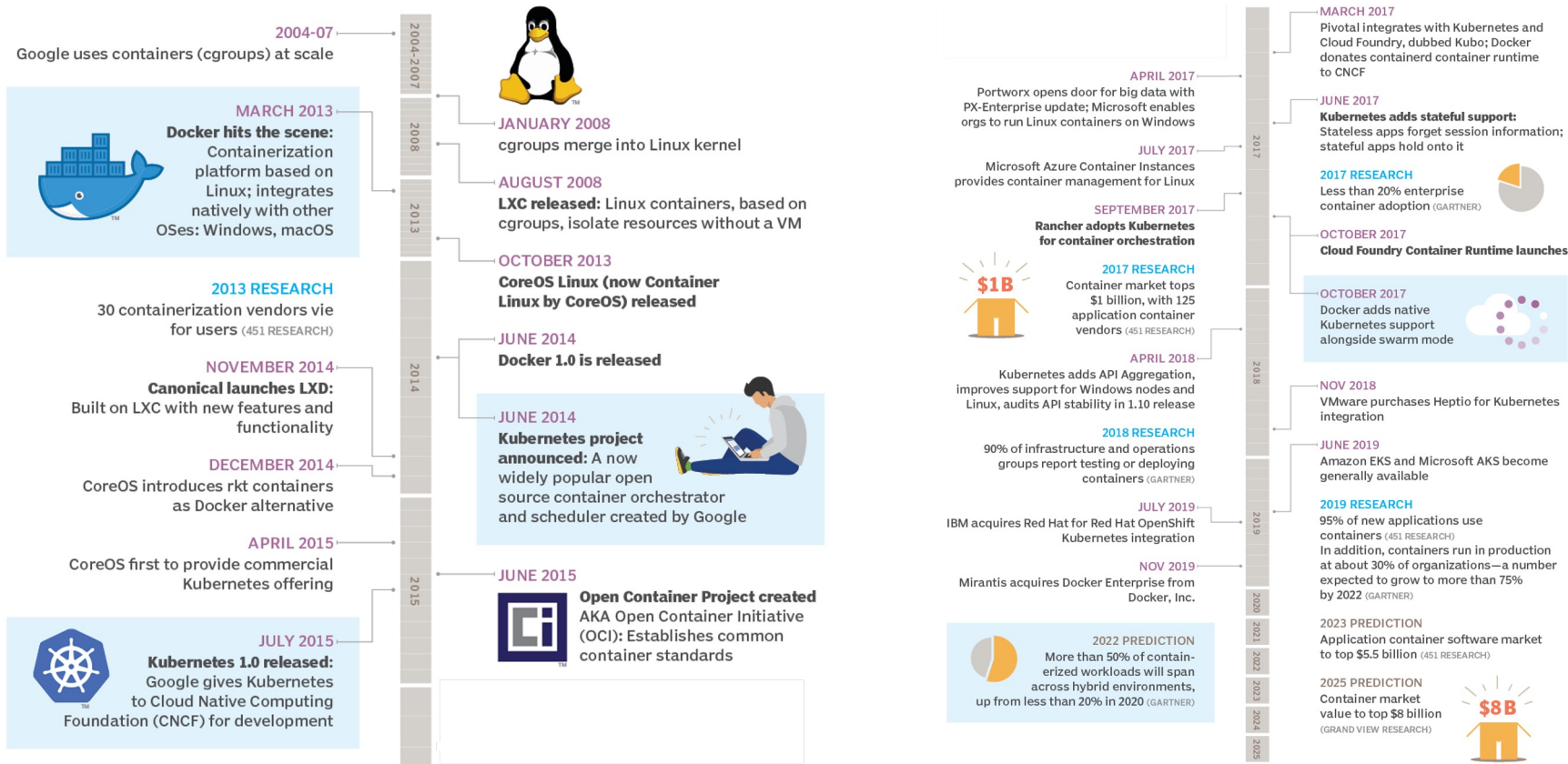**LXC released:** Linux containers, based on cgroups, isolate resources without a VM

OCTOBER 2013
CoreOS Linux (now Container Linux by CoreOS) released

JUNE 2014
Docker 1.0 is released

JUNE 2014
Kubernetes project announced: A now widely popular open source container orchestrator and scheduler created by Google

JUNE 2015
Open Container Project created AKA Open Container Initiative (OCI): Establishes common container standards

---

APRIL 2017
Portworx opens door for big data with PX-Enterprise update; Microsoft enables orgs to run Linux containers on Windows

JULY 2017
Microsoft Azure Container Instances provides container management for Linux

SEPTEMBER 2017
Rancher adopts Kubernetes for container orchestration

2017 RESEARCH
Container market tops $1 billion, with 125 application container vendors (451 RESEARCH)

APRIL 2018
Kubernetes adds API Aggregation, improves support for Windows nodes and Linux, audits API stability in 1.10 release

2018 RESEARCH
90% of infrastructure and operations groups report testing or deploying containers (GARTNER)

JULY 2019
IBM acquires Red Hat for Red Hat OpenShift Kubernetes integration

NOV 2019
Mirantis acquires Docker Enterprise from Docker, Inc.

2022 PREDICTION
More than 50% of containerized workloads will span across hybrid environments, up from less than 20% in 2020 (GARTNER)

MARCH 2017
Pivotal integrates with Kubernetes and Cloud Foundry, dubbed Kubo; Docker donates containerd container runtime to CNCF

JUNE 2017
Kubernetes adds stateful support: Stateless apps forget session information; stateful apps hold onto it

2017 RESEARCH
Less than 20% enterprise container adoption (GARTNER)

OCTOBER 2017
Cloud Foundry Container Runtime launches

OCTOBER 2017
Docker adds native Kubernetes support alongside swarm mode

NOV 2018
VMware purchases Heptio for Kubernetes integration

JUNE 2019
Amazon EKS and Microsoft AKS become generally available

2019 RESEARCH
95% of new applications use containers (451 RESEARCH)
In addition, containers run in production at about 30% of organizations—a number expected to grow to more than 75% by 2022 (GARTNER)

2023 PREDICTION
Application container software market to top $5.5 billion (451 RESEARCH)

2025 PREDICTION
Container market value to top $8 billion (GRAND VIEW RESEARCH)

$1B

$8B

Containers are a natural fit for DevOps

- **Improved development/deployment pipeline**
  - Containers allow developers to create predictable runtime environments, including all software dependencies required by an application component, isolated from other applications on the same machine. The old adage "it worked on my machine" is no longer a concern with container technology.

- **Improved resource utilization**
  - Containers do not require a separate operating system and therefore use fewer resources. VMs are typically a few GB in size, but containers commonly weigh only tens of megabytes, making it possible for a server to run many more containers than VMs.

- **Increased portability**
  - Containers can run anywhere, as long as the container engine supports the underlying operating system—it is possible to run containers on Linux, Windows, MacOS, and many other operating systems. Containers can run in virtual machines, on bare metal servers, locally on a developer's laptop. They can easily be moved between on-premise machines and public cloud, and across all these environments, continue to work consistently.

- **Improved scalability—up and down**
  - Containers make it easy to horizontally scale distributed applications. **Container orchestrators** can perform smart scaling, running only the number of containers you need to serve application loads, while taking into account resources available to the container cluster.

References

- Cloud Native Computing Foundation: https://www.cncf.io/

- Devopedia: https://devopedia.org/devops

- The Twelve-Factor App: https://12factor.net/