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.*

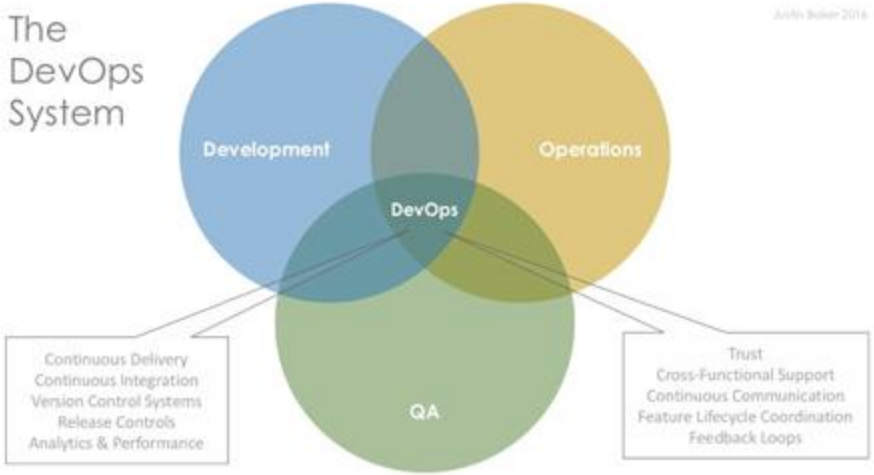
Cloud Native Computing Foundation (CNCF)

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.

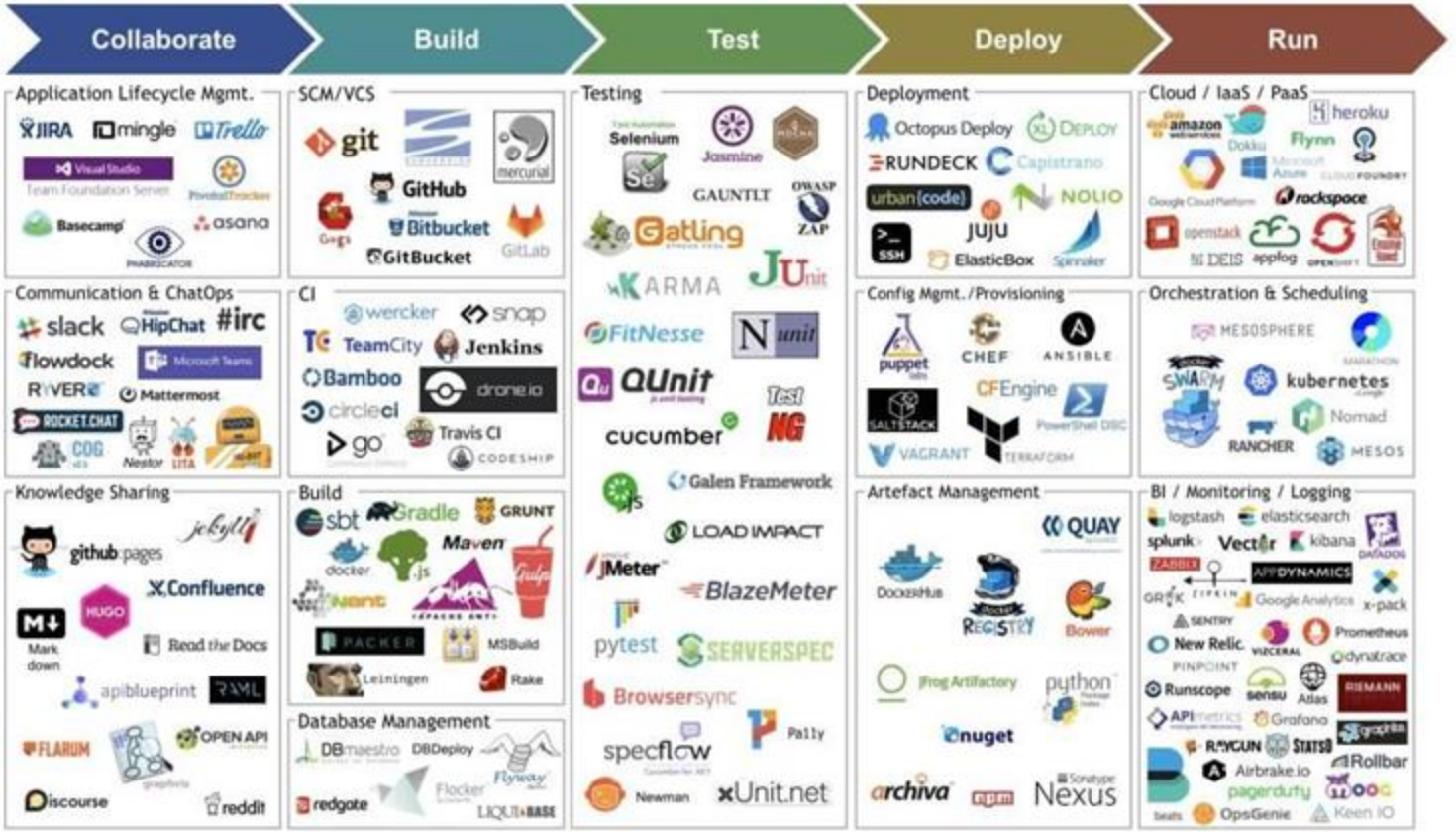


DevOps has been described variously as a culture, a mindset, a framework and a movement.

The CAMS model

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)?

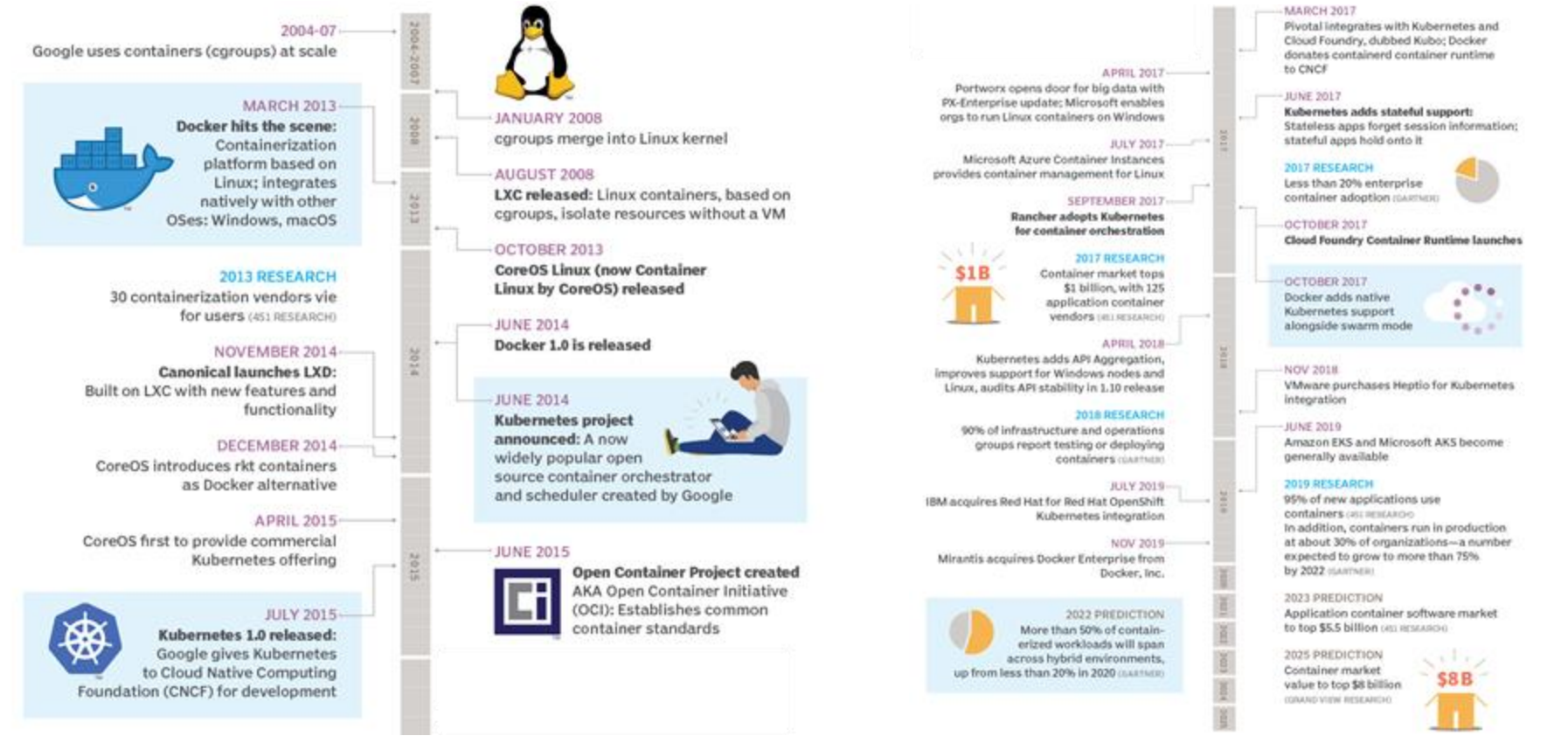
The concept is borrowed from shipping containers, which define a standard to ship goods globally.

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



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/