



IS4301 Agile IT with DevOps – Lecture 10

Adjunct Professor Foong Sew Bun

Department of Information Systems and Analytics

National University of Singapore

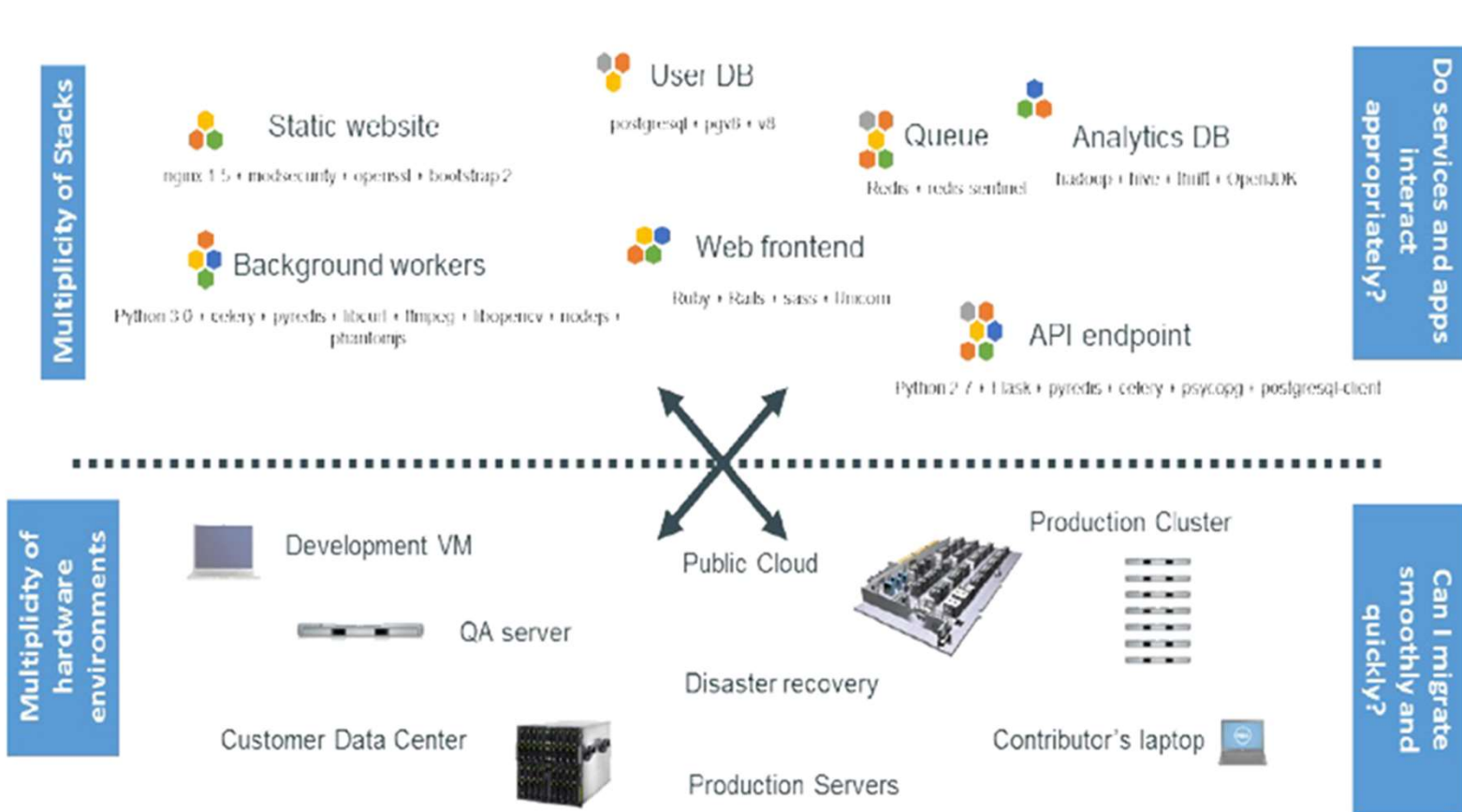
Learning Objectives

At the end of this lecture, you will understand important foundational technologies used in agile development and deployment:













- What are containers?
- Why are containers important?
- Are containers and virtual machines different?
- Standards of containers
- CI/CD tool chains



Why Containerize Applications and Services?





Matrix of Services for Containerization

	Static website	?	?	?	?	?	?	?
	Web frontend	?	?	?	?	?	?	?
	Background workers	?	?	?	?	?	?	?
	User DB	?	?	?	?	?	?	?
	Analytics DB	?	?	?	?	?	?	?
	Queue	?	?	?	?	?	?	?
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
								

Cargo Transport – 1960s



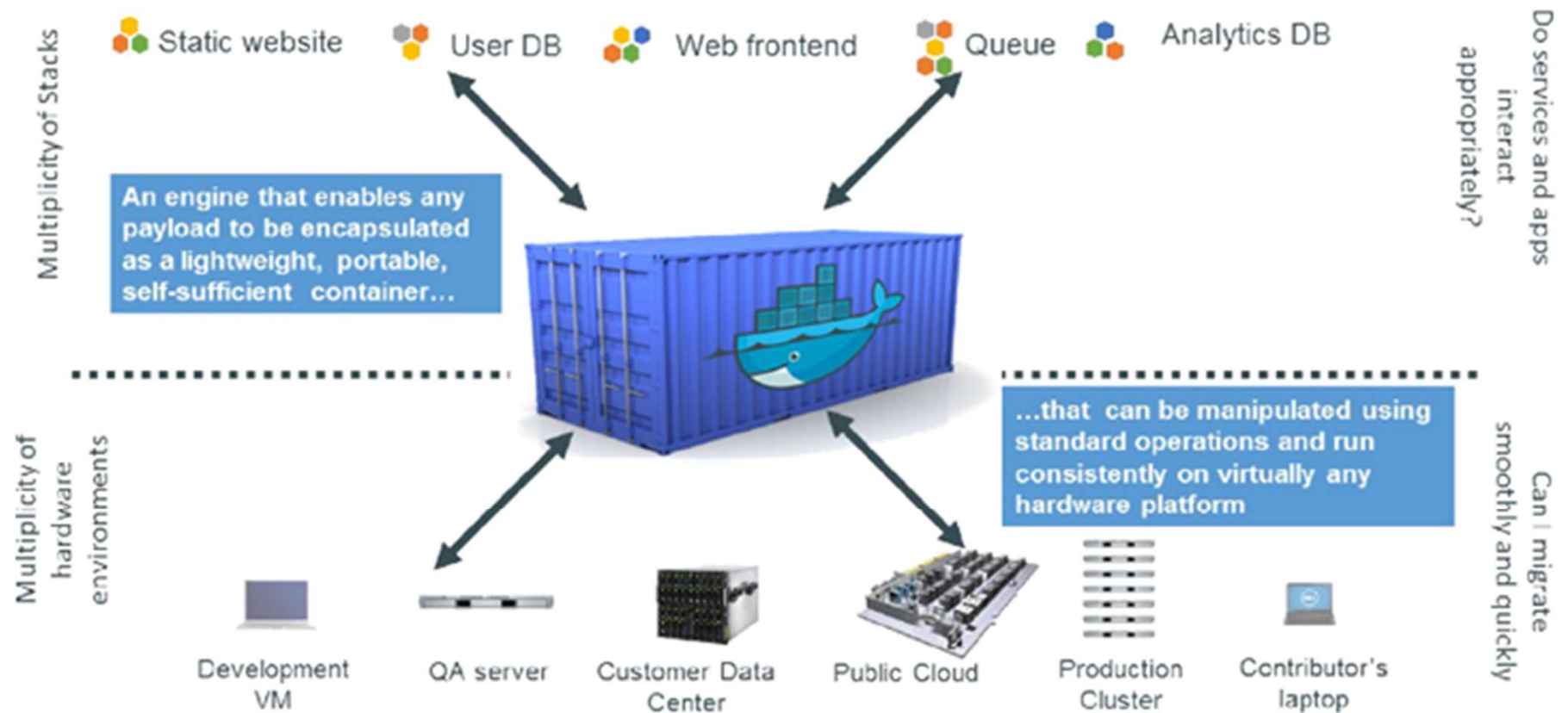
Similar Matrix for Containerizing Cargo Transport

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


















































Solution: International Shipping Containers



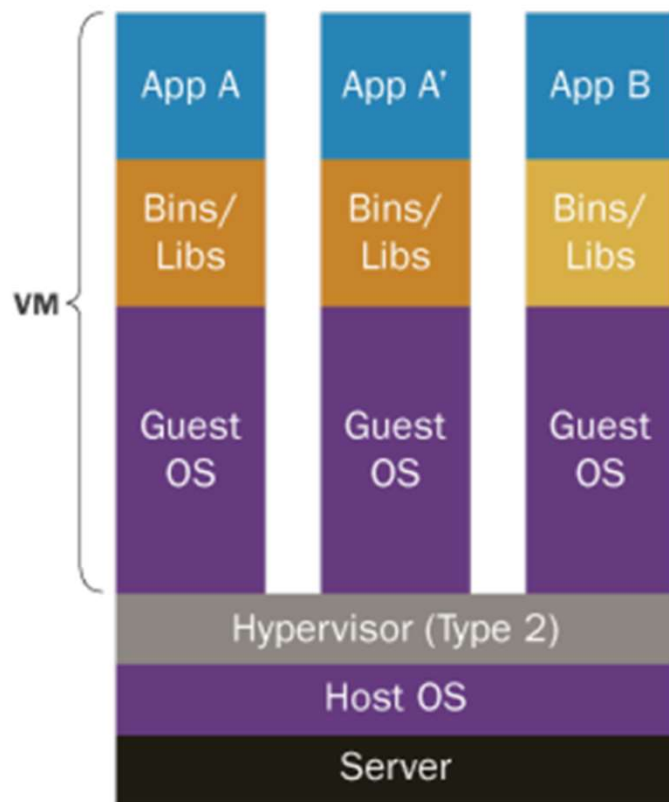
Containers as Code



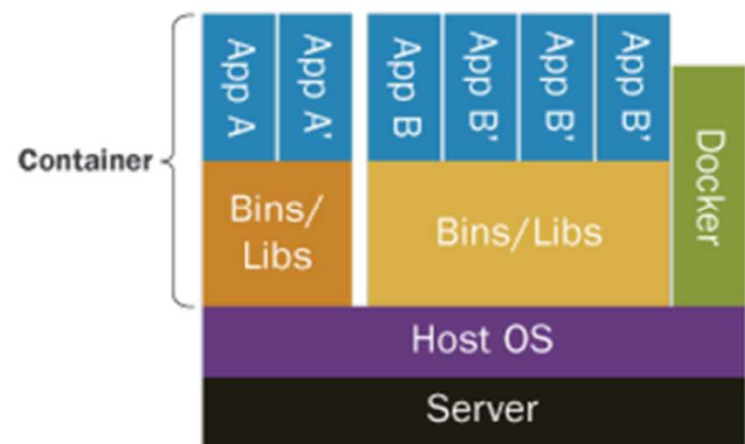
Matrix

	Static website							
	Web frontend							
	Background workers							
	UserDB							
	Analytics DB							
	Queue							
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
								

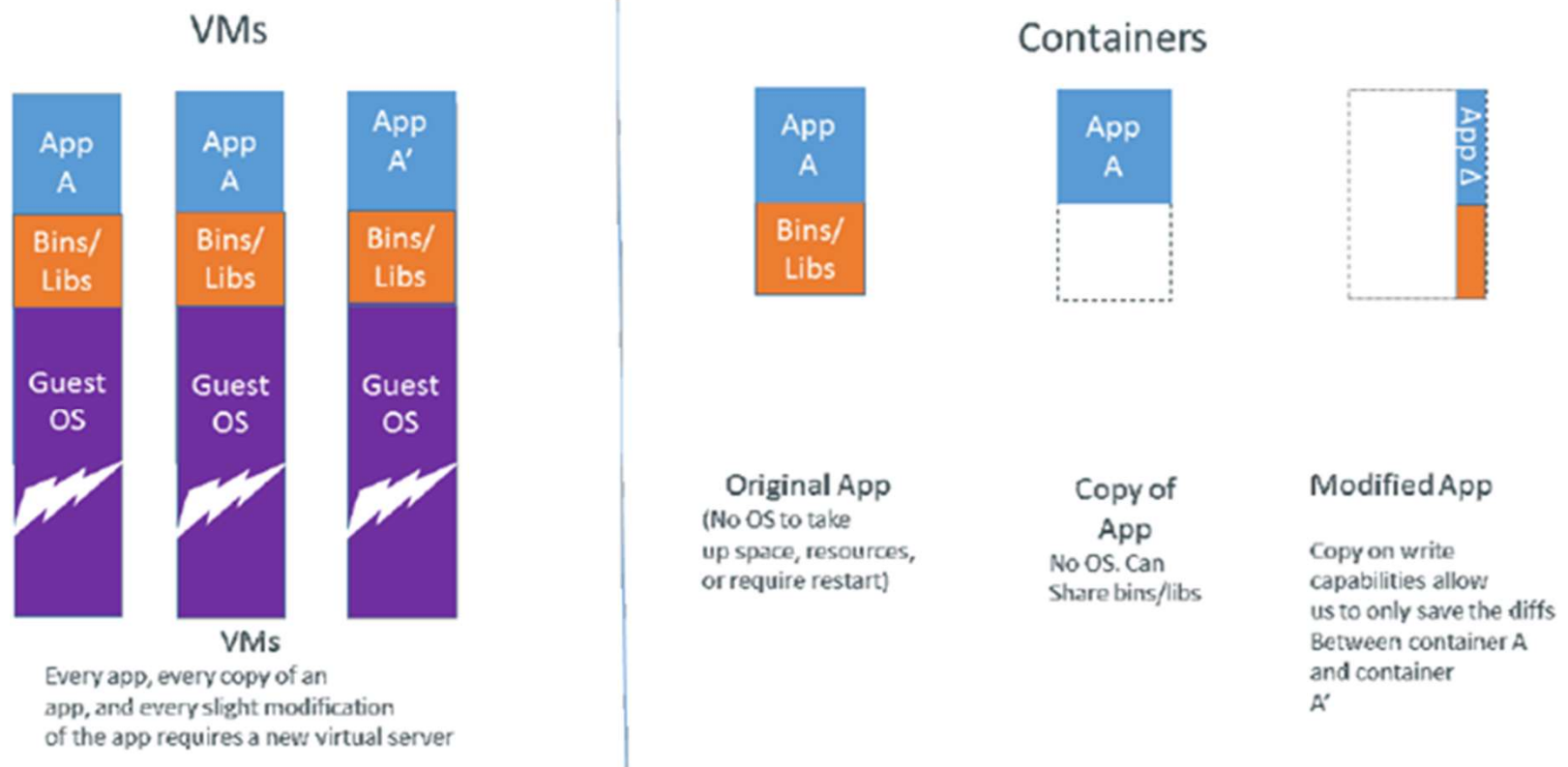
Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



Light Weight Containers

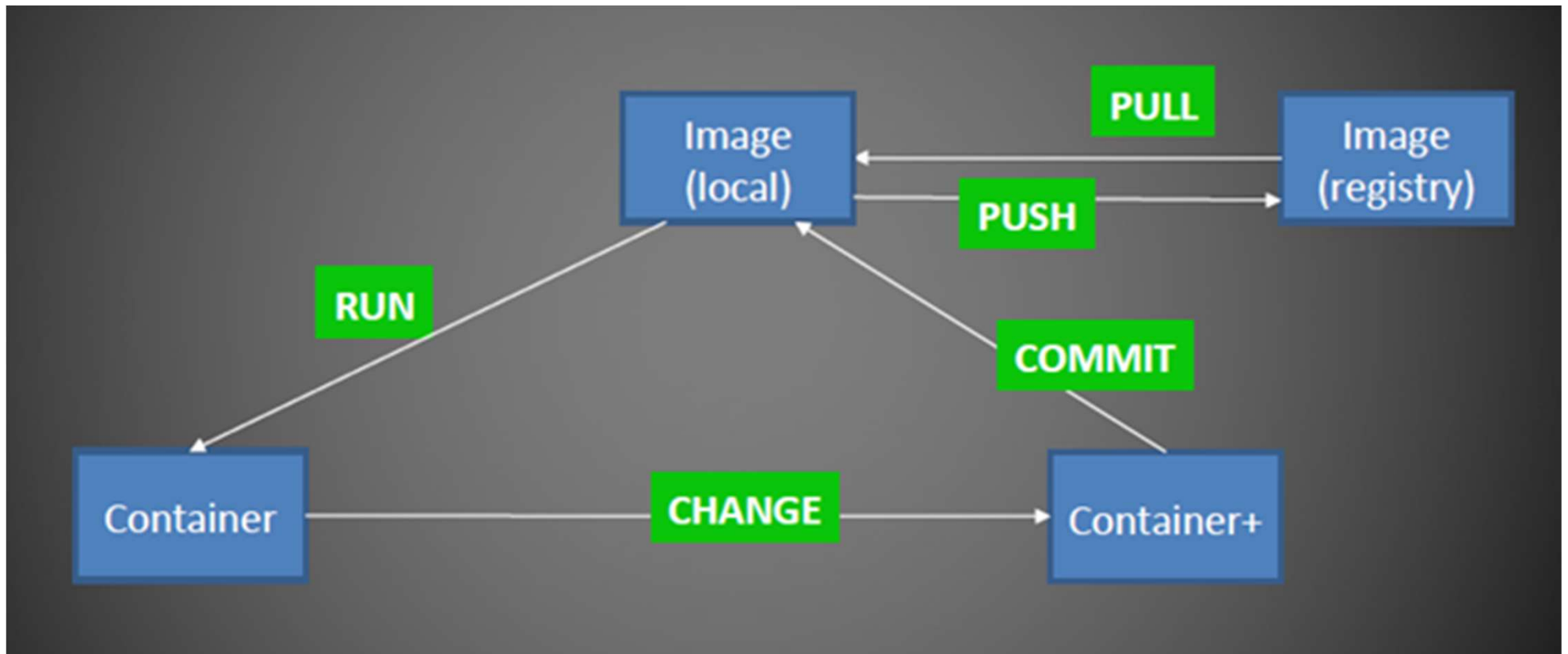




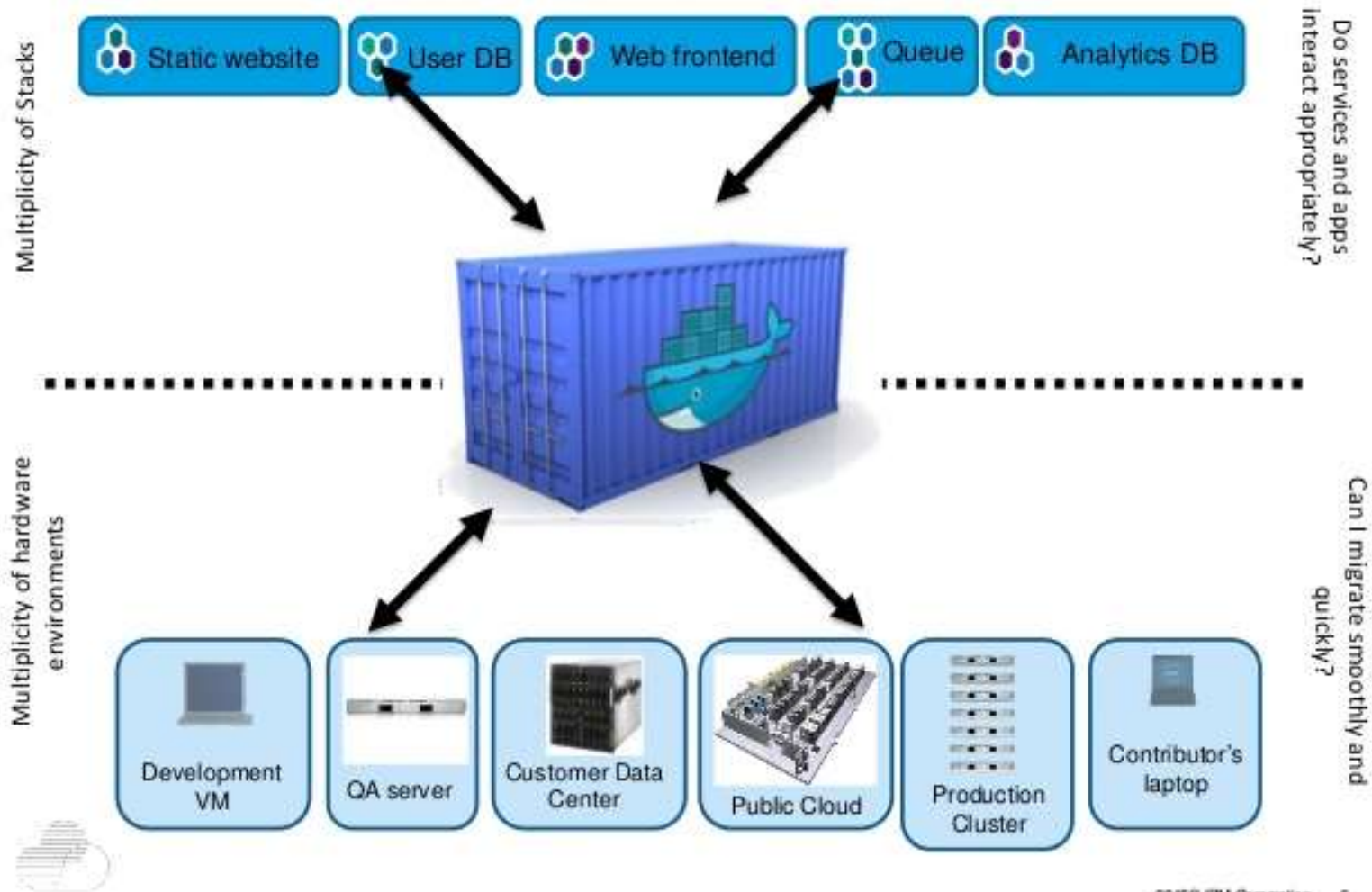
Summary of Container Benefits

- **Agile application creation and deployment:** Increased ease and efficiency of container image creation compared to VM image use.
- **Continuous development, integration, and deployment:** Provides for reliable and frequent container image build and deployment with quick and easy rollbacks (due to image immutability).
- **Dev and Ops separation of concerns:** Create application container images at build/release time rather than deployment time, thereby decoupling applications from infrastructure.
- **Environmental consistency across development, testing, and production:** Runs the same on a laptop as it does in the cloud.
- **Cloud and OS distribution portability:** Runs on Ubuntu, RHEL, CoreOS, on-prem, Google Container Engine, and anywhere else.
- **Application-centric management:** Raises the level of abstraction from running an OS on virtual hardware to run an application on an OS using logical resources.
- **Loosely coupled, distributed, elastic, liberated micro-services :** Applications are broken into smaller, independent pieces and can be deployed and managed dynamically – not a fat monolithic stack running on one big single-purpose machine.
- **Resource isolation:** Predictable application performance.
- **Resource utilization:** High efficiency and density

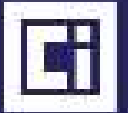
Image and Containers



Docker is a **shipping container** system for code



What is the Open Container Initiative (OCI)



An open source community (hosted by the Linux Foundation) for building a vendor-neutral, portable and open specification/runtime that delivers on the promise of containers as a source of application portability backed by a certification program.



OPEN CONTAINER
INITIATIVE

OCI Membership (46 organizations as of May 2016)



Kubernetes

- Kubernetes is an open-source platform designed to automate deploying, scaling and operating application containers.
- Consistent object model and API supporting many leading cloud providers like AWS and Google.



kubernetes

Apache Mesos and Mesosphere



- Created in 2009 at UC Berkeley, hardened in Twitter
- Top level Apache project
- Mesosphere, Twitter, Airbnb are major users/contributors
- Scales to 10000+ nodes, production grade
- Google officially endorsed Mesos for Kubernetes
- Built in containerization, including docker
- Packages and support through Mesosphere
- [Mesosphere](#) is an enterprise software OEM that sells a "data center operating system" also built on Mesos and providing cluster management, container orchestration, service discovery and build automation for elastic computing. Mesosphere's offerings are in large-scale use at Yelp, Verizon and Bloomberg, to name a few adopters.

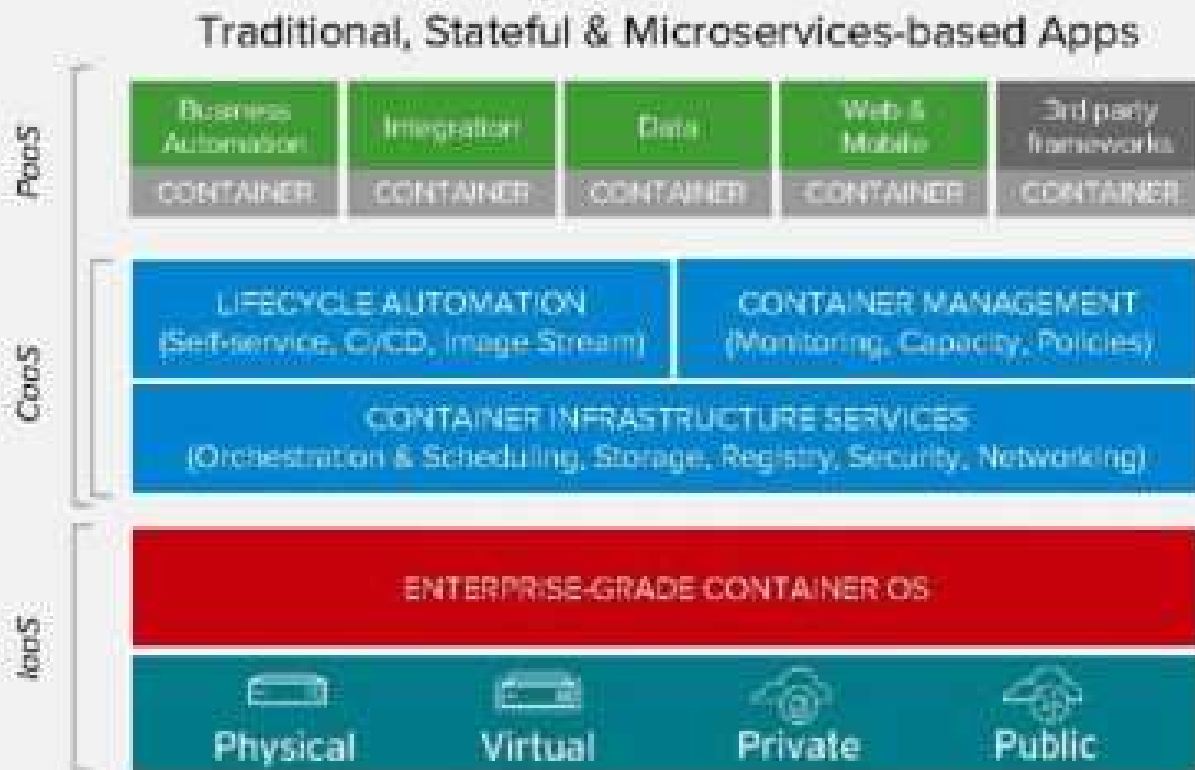
RED HAT CONTAINER PORTFOLIO



Red Hat Application
Services (JBoss)

Red Hat OpenShift
Container Platform
(incl. CloudForms)

Red Hat Enterprise
Linux & Atomic Host



Red Hat
Registry

CloudForms

Ansible

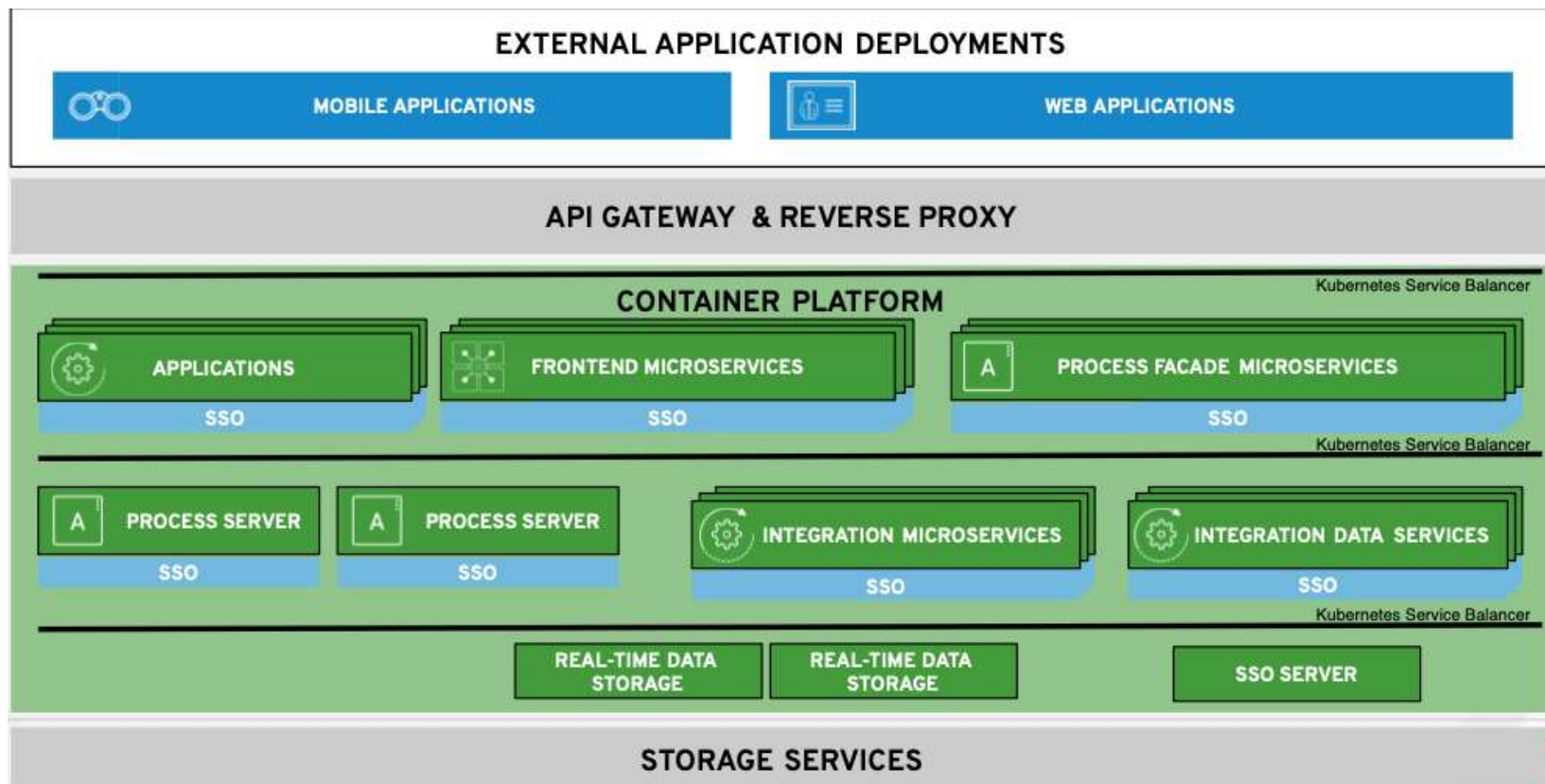
Storage

Satellite

Developer
Studio

CDK

Integration of RedHat Container Platform



DevOps Tools Market Map

Source Code Management



Continuous Integration



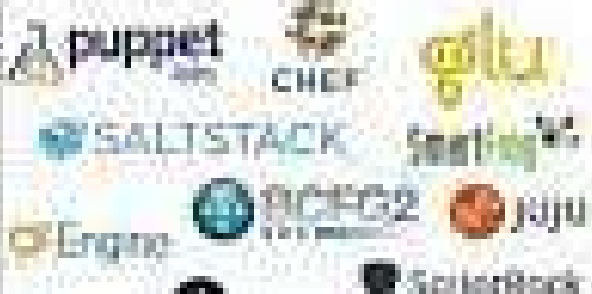
Testing



Containers



Configuration Management



Deployment



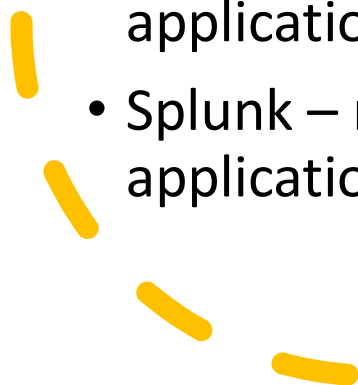
Monitoring





Popular Monitoring Tools

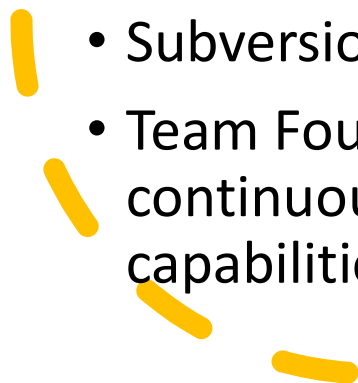
- Graphite – open source tool for storing data and rendering it graphically
- Logstash – open source tools for managing logs and other event data from your systems
- Nagios – monitoring and alerting tool for servers, switches, applications and services
- Splunk – monitoring and visualizing data from web sites, servers, applications, networks.





Popular Version Control Tools

- Git – open source and distributed version control system, allows code check in and merges while working offline
- Mercurial – similar functionalities as Git, though Git is more popular among developers
- Perforce – proprietary version control system that supports Git
- Subversion – open source version control
- Team Foundation Server – Microsoft version control system, includes continuous integration, issue tracking and project management capabilities





Popular Continuous Integration Tools

- Bamboo – Proprietary tool from Atlassian that runs builds and tests
- Jenkins, Hudson – open source continuous integration tools with automated continuous build and monitoring of externally run jobs like cron jobs
- Go – open source continuous integration tool created by continuous delivery consulting firm from ThoughtWorks, which offers paid support for Go
- Team City – proprietary continuous integration tool that integrates with Git and Mercurial
- Travis – proprietary continuous integration tool



Popular Tools

- Configuration Management tool
 - Puppet – open source as well as proprietary. Can include open source projects such as Beaker for automated acceptance test and r10k, an automated module deployment tool
- Code Review Tools
 - GitHub – online system for code review, collaboration and code management. Available for free and paid services
 - Stash – proprietary tool from Atlassian for reviewing code in Git, with features for enterprise usage like enhanced security services
 - Gerrit – web based code review system that enables online code review using Git

