

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

CHAPTER 8 : CONTINUOUS INSPECTION

DR FARIZUWANA AKMA
farizuwana@utar.edu.my

Static Code Analysis

- A.k.a source code analysis
- Tool that scan source files for violations of predefined rules
- Examples:
 - Java's PMD
 - .NET FxCop

Advantages

- Cheap to run often
- Objective evaluation
- Customizable
- Work well in geographically distributed teams
- Scale more efficiently for large code bases

Reducing Code Complexity

- **Code complexity caused by:**
 - Methods that are long
 - Methods that are deeply nested conditional
- **Cyclomatic Complexity Number (CCN)**
 - Plain integer that measures complexity by counting the number of distinct paths through a method
 - CCN > 10 have higher risk of defects

Reducing Code Complexity

- Tools:
 - PMD
 - JavaNCSS
 - .NET CCMetrics
 - .NET Source Monitor

Reduce Duplicate Code

- **Copied-and-pasted code can occur in all areas, e.g.**
 - Database logics, stored procedures and views, e.g. SQL
 - Compiled source code, e.g. Java, C, C++, C#
 - Interpreted source code, e.g. ASP, JSP, Ruby, Python
 - Build scripts, e.g. make, Ant
 - Data and configuration files, e.g. ASCII, XML, XSD, DTD

Reduce Duplicate Code

- Duplicated code causes these problems:
 - Increased maintenance costs due to discovering, reporting, analysing and fixing bugs multiple times.
 - Uncertainty about the existence of other bugs (on unfound duplicate code)
 - Increased testing costs for the additional code written
- Tools:
 - PMD
 - Simian

Assess Code Coverage

- **Line (or statement) coverage**
 - Indicate that a particular line of code was exercised
 - If a code base has 10,000 non-commenting lines of code, 3,000 were exercised on a particular test run, then it is 30% code coverage
- **Branch (or path) coverage**
 - Coverage of decision points, e.g. if..else, switch

Evaluate Code Quality Continuously

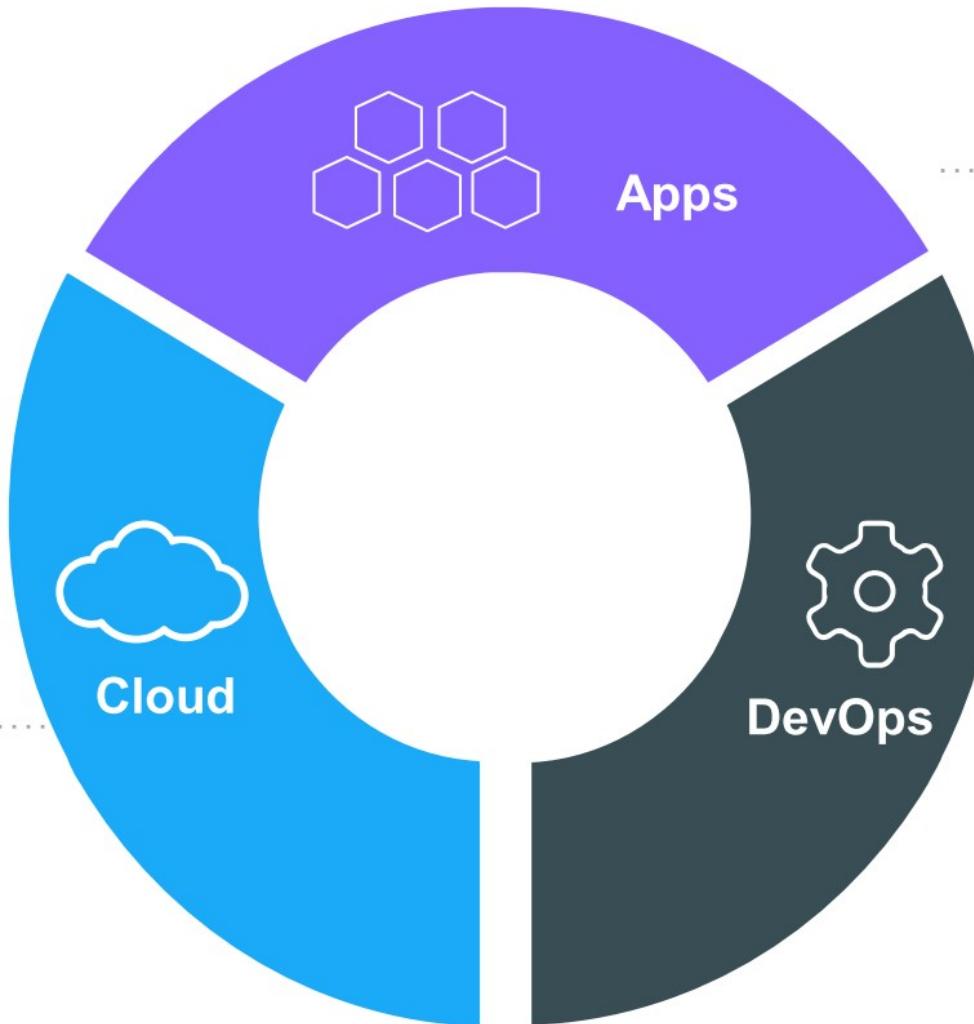
- QA use coverage report to fine-tune functional testing
- If the coverage rate drop, you can infer that:
 - The source code grew, but no corresponding tests were written
 - Test cases were removed

UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION

CHAPTER 9 : Introduction to Docker

DR FARIZUWANA AKMA
farizuwana@utar.edu.my

The IT Landscape is Changing



Movement in the cloud



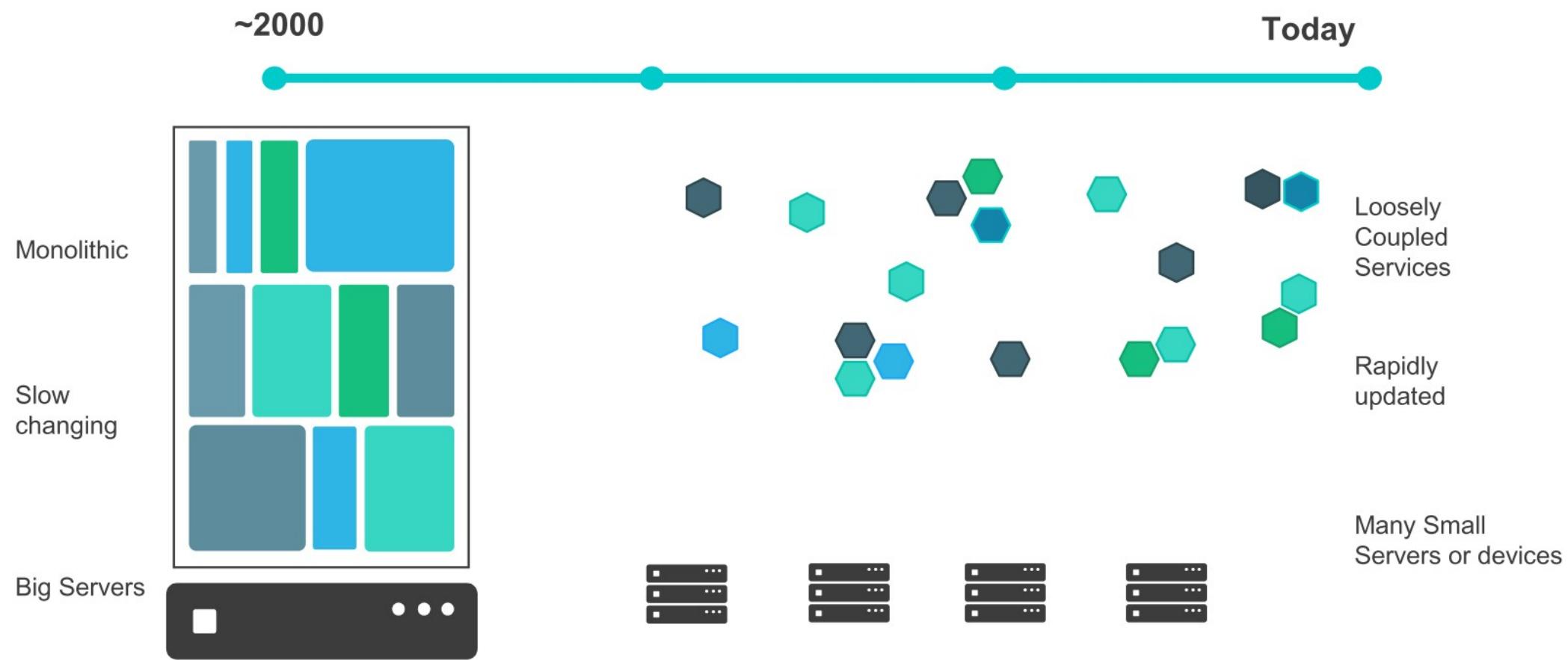
Migrate workloads to cloud

Portability across environments

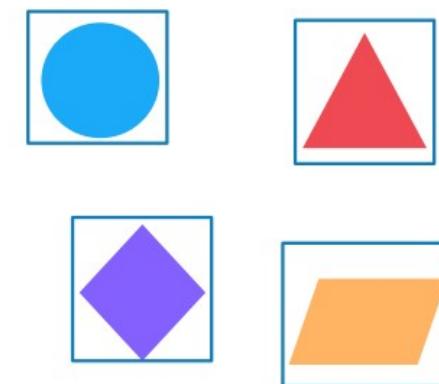
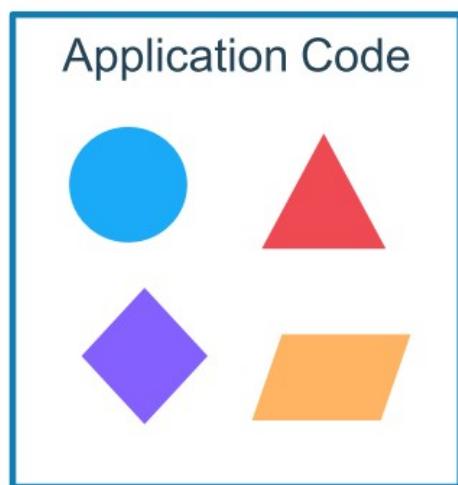
Want to avoid cloud vendor lock-in

State of App development Survey: Q1 - 2016

Applications are transforming



Application Modernization



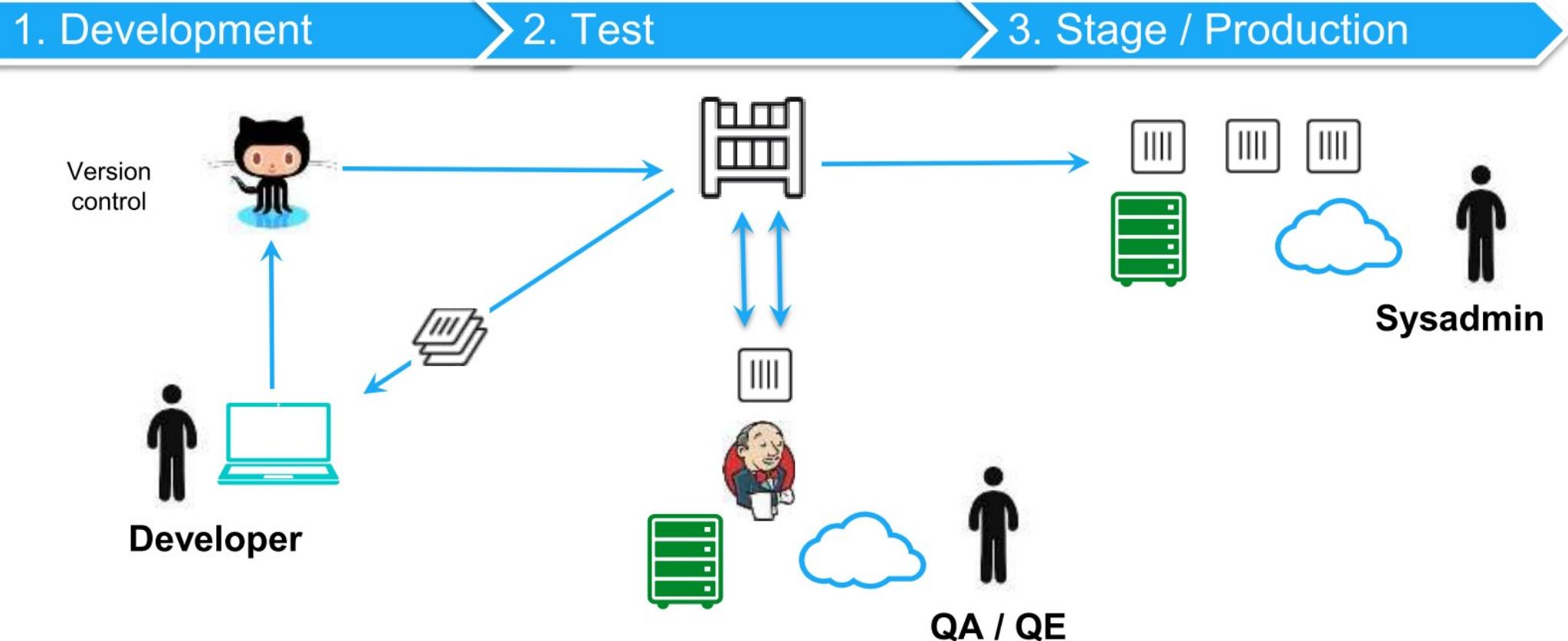
Developer Issues:

- Minor code changes require full re-compile and re-test
- Application becomes single point of failure
- Application is difficult to scale

Microservices: Break application into separate operations

12-Factor Apps: Make the app independently scalable, stateless, highly available by design

Continuous Integration and Delivery



Tug of War Between Developers and Ops



Developers

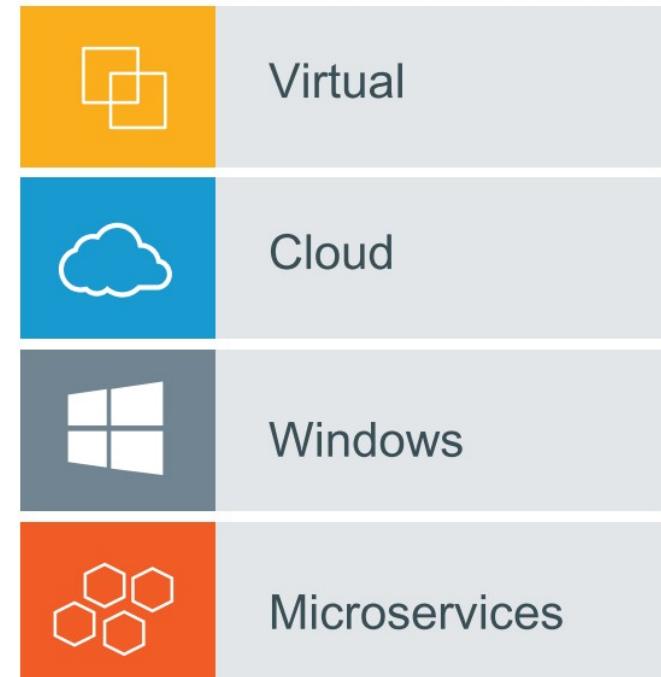
- Freedom to create and deploy apps fast
- Define and package application needs



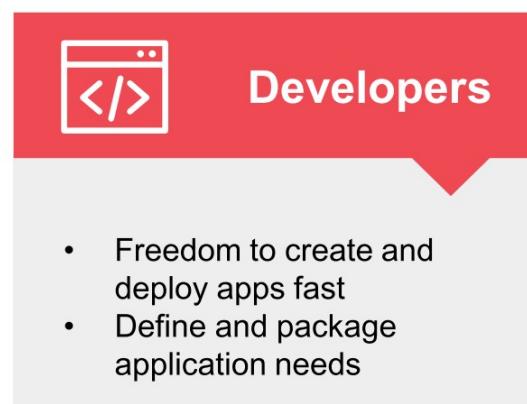
IT Operations

- Quickly and flexibly respond to changing needs
- Standardize, secure, and manage

Organizations Must Deal with Diverse Technology



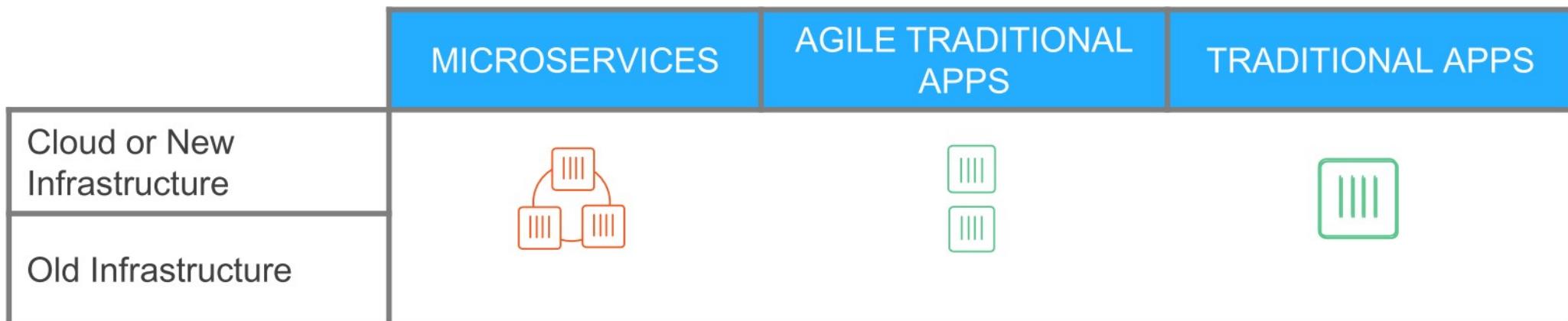
and diverse organizations...



The Myth of Bi-Modal IT

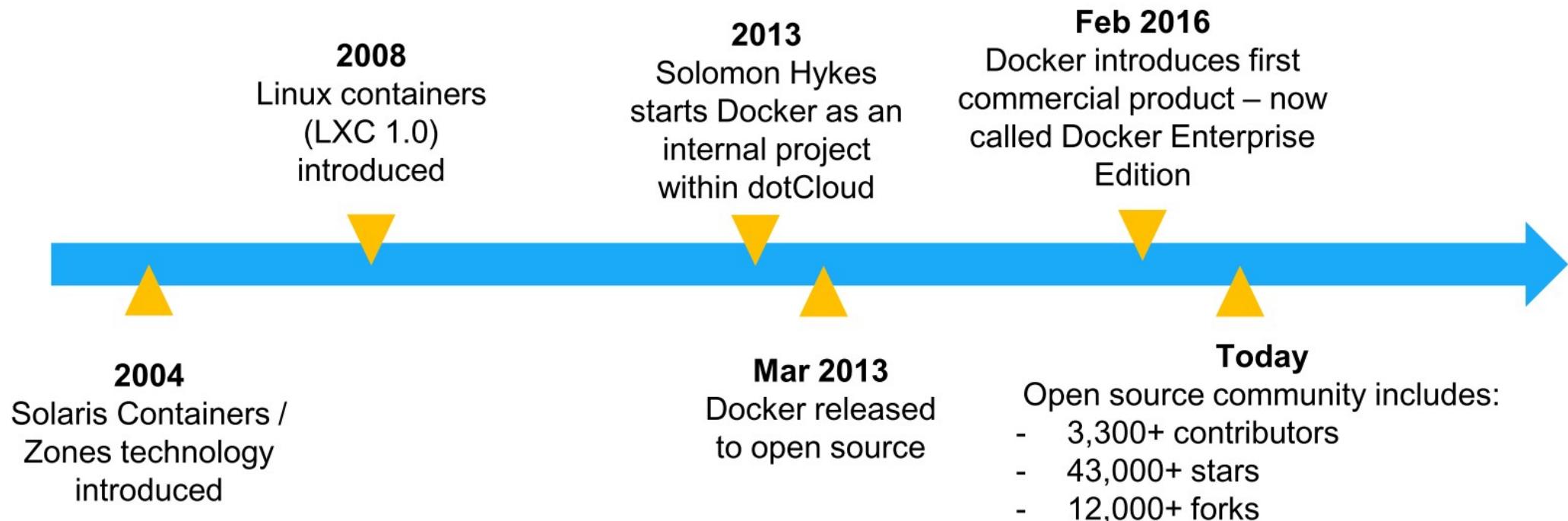
	MICROSERVICES	TRADITIONAL APPS
Cloud or New Infrastructure	You are either here..	
Old Infrastructure		...or here

Enabling a Journey

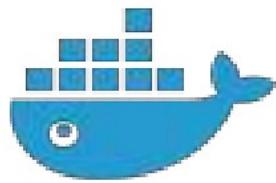


...that is past AND future proof

History of Docker



Incredible adoption in just 4 years



14M

Docker
Hosts

900K

Docker
apps

77K%

Growth in
Docker job
listings

12B

Image pulls
Over 390K%
Growth

3300

Project
Contributors

The Docker Family Tree



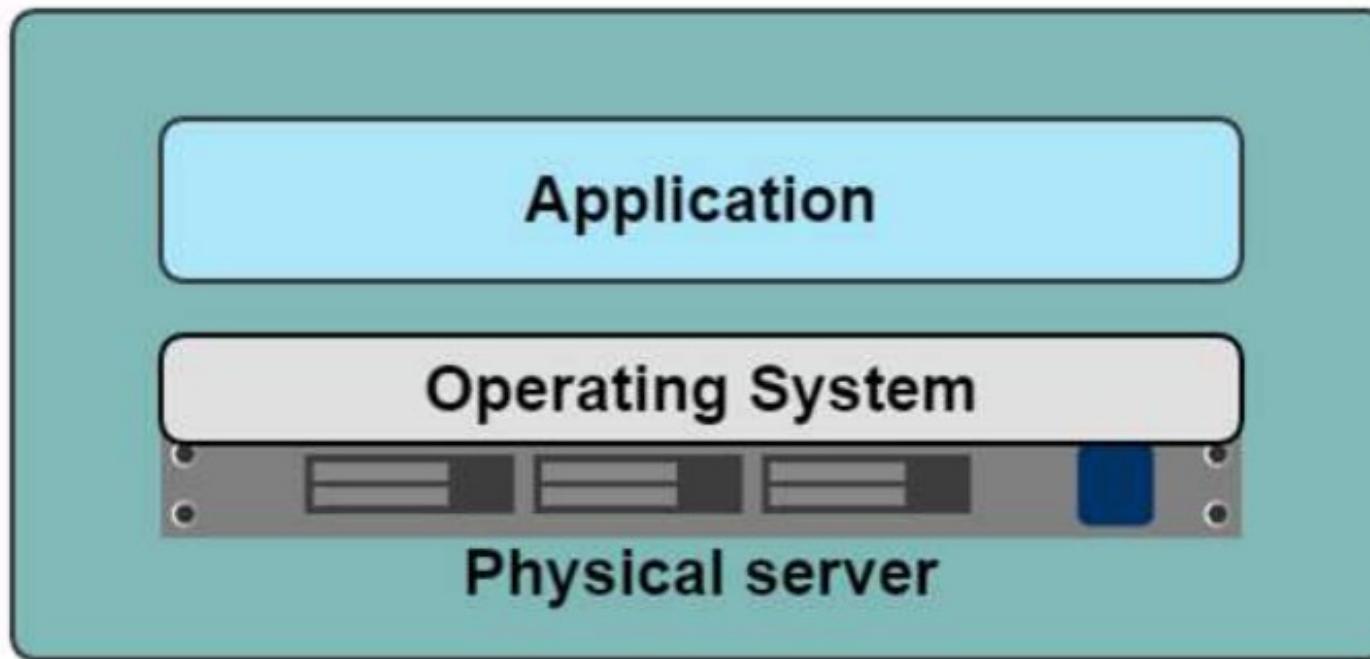
Open source **framework** for assembling core components that make a container platform

Intended for:
Open source contributors + ecosystem developers



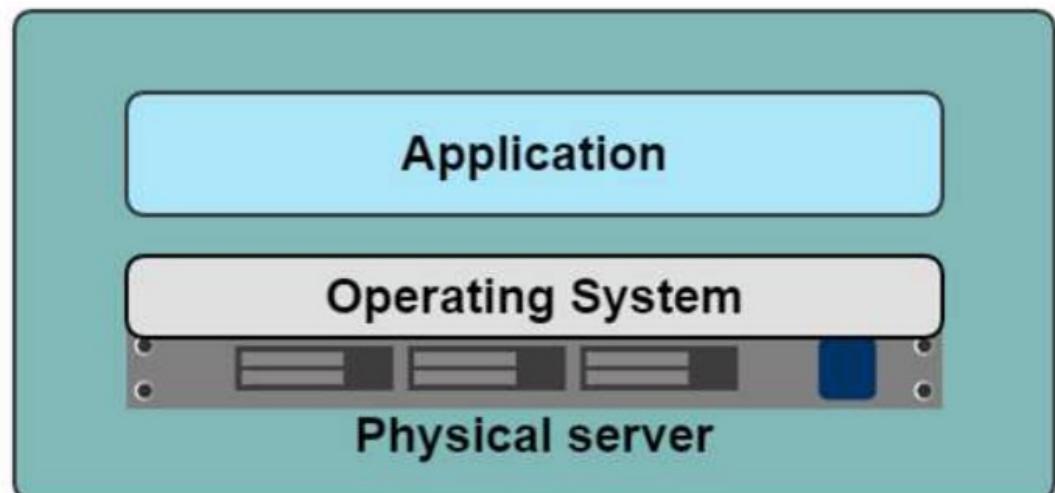
A History Lesson: In the Dark Ages

One Application on One Physical Server



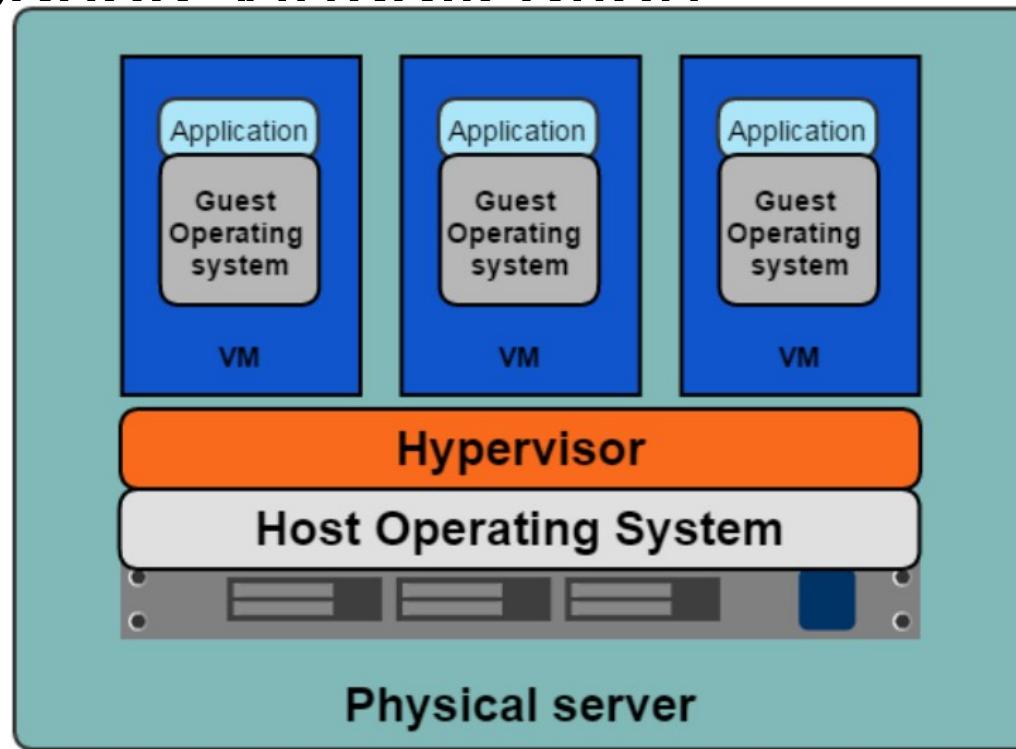
Historical limitations of application deployment

- Slow deployment times
- Huge costs
- Wasted resources
- Difficult to scale
- Difficult to migrate
- Vendor lock in



A History Lesson: Hypervisor-based Virtualization

- One physical server can contain multiple applications
 - Each application runs in a virtual machine (VM)
- ## Hypervisor-based Virtualization



Benefits of VMs

- Better resource pooling
 - One physical machine divided into multiple virtual machines
- Easier to scale
- VMs in the cloud
 - Rapid elasticity
 - Pay as you go model



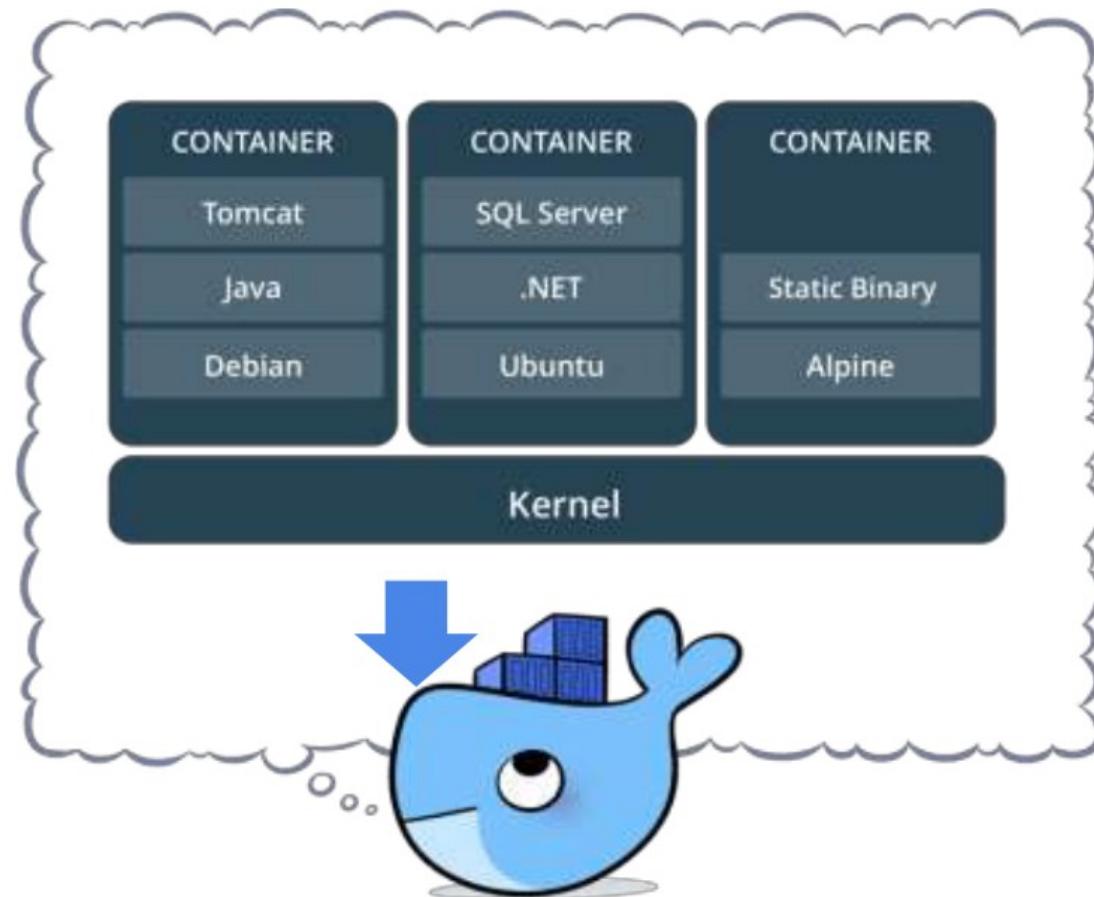
Limitations of VMs

- Each VM stills requires
 - CPU allocation
 - Storage
 - RAM
 - An entire guest operating system
- The more VMs you run, the more resources you need
- Guest OS means wasted resources
- Application portability not guaranteed

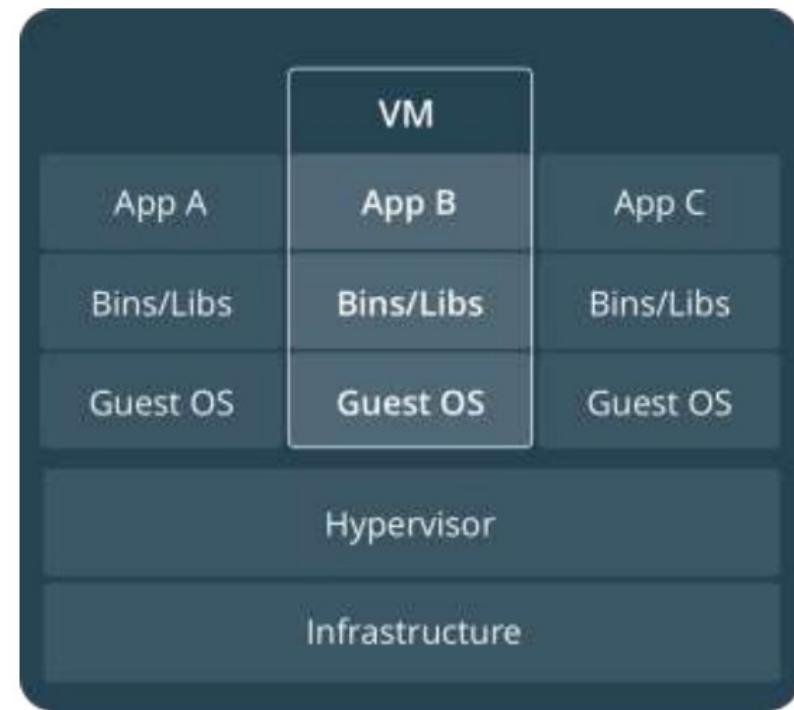


What is a container?

- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works with all major Linux and Windows Server



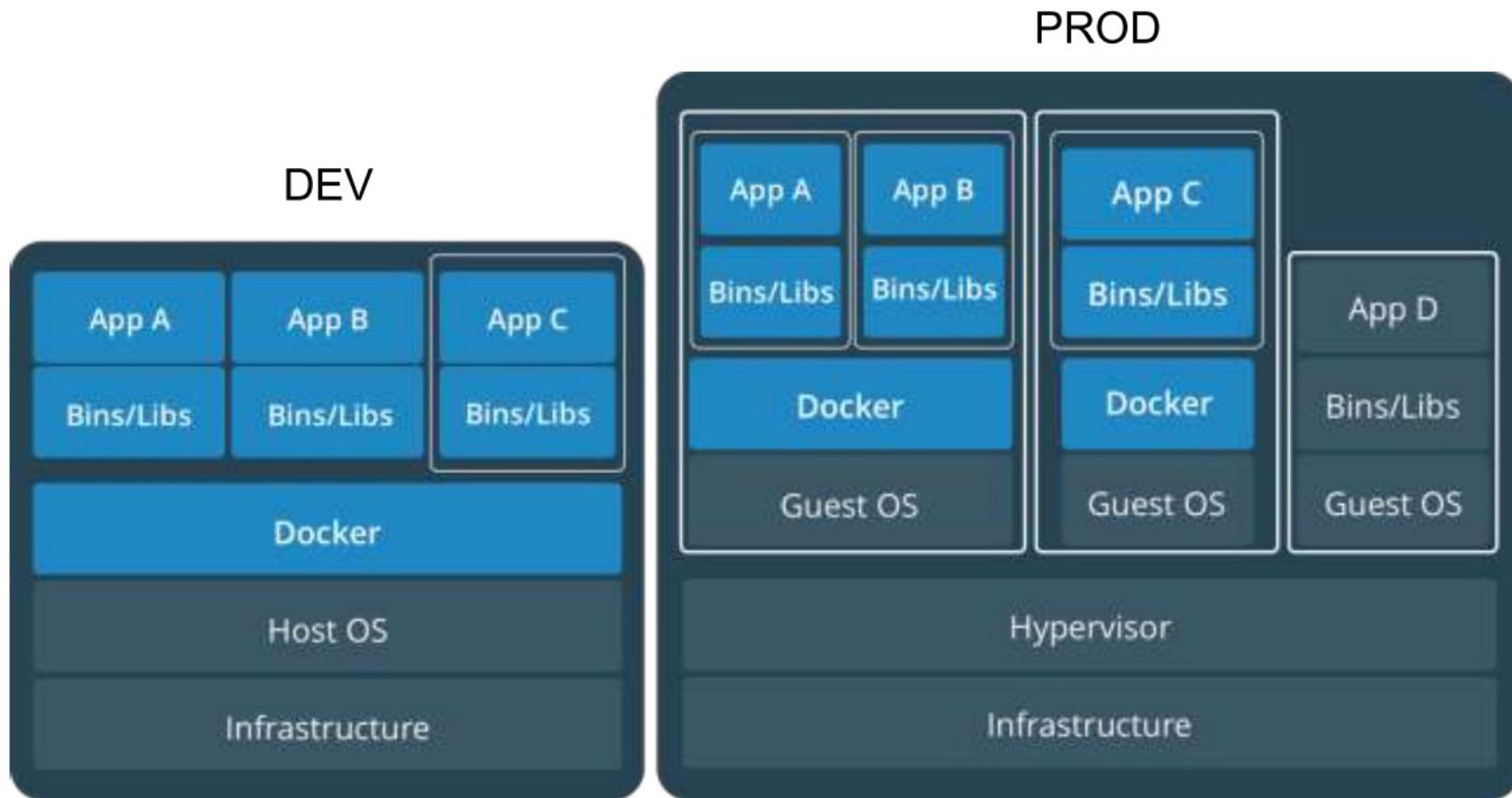
Comparing Containers and VMs



Containers are an app level construct

VMs are an infrastructure level construct to turn one machine into many servers

Containers and VMs together



Containers and VMs together provide a tremendous amount of flexibility for IT to optimally deploy and manage apps.

Key Benefits of Docker Containers

Speed

- No OS to boot = applications online in seconds

Portability

- Less dependencies between process layers = ability to move between infrastructure

Efficiency

- Less OS overhead
- Improved VM density

The Challenge

Multiplicity of
Stacks

 Static website
nginx 1.5 + modsecurity + openssl + bootstrap 2

 Background workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

 User DB
postgresql + pgv8 + v8

 Queue
Redis + redis-sentinel

 Analytics DB
hadoop + hive + thrift + OpenJDK

 Web frontend
Ruby + Rails + sass + Unicorn

 API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

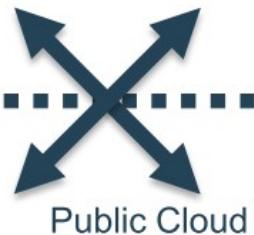
Do services and
apps interact
appropriately?

Multiplicity of
hardware
environments

 Development VM

 QA server

Customer Data Center



Disaster recovery



Production Cluster



Contributor's laptop



Production Servers

Can I migrate
smoothly and
quickly?

The Matrix From Hell

	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 Pre-1960

Multiplicity of Goods



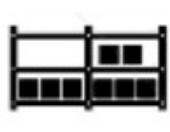
Do I worry about how goods interact (e.g. coffee beans next to spices)

Multiplicity of transporting/storying



Can I transport quickly and smoothly (e.g. from boat to train to truck)

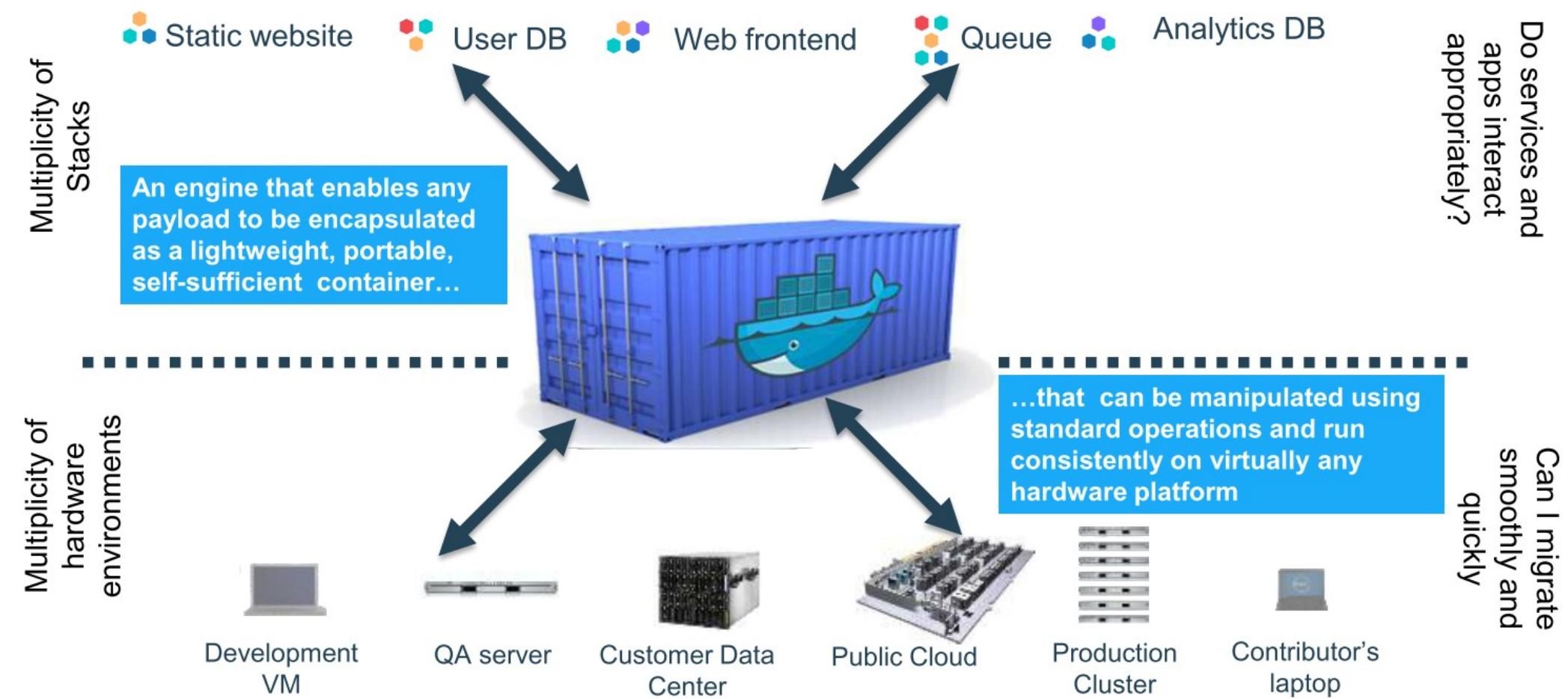
Also a matrix from hell

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

Solution: Intermodal Shipping Container



Docker is a shipping container system for code



Docker eliminates the matrix from Hell

	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	
								

Docker Basics



Image

The basis of a Docker container. The content at rest.



Container

The image when it is ‘running.’ The standard unit for app service



Engine

The software that executes commands for containers. Networking and volumes are part of Engine. Can be clustered together.



Registry

Stores, distributes and manages Docker images



Control Plane

Management plane for container and cluster orchestration

Foundation: Docker Engine

Integrated Security

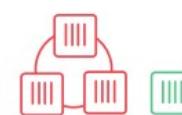


Docker Engine

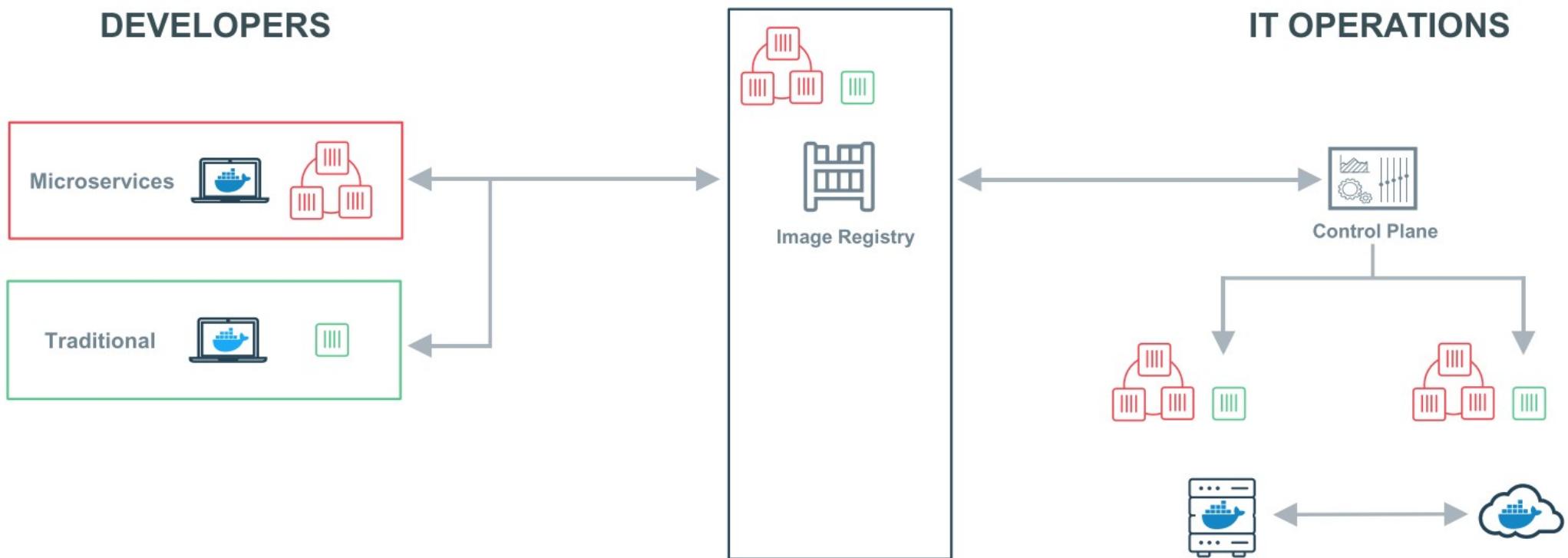
DEVELOPERS



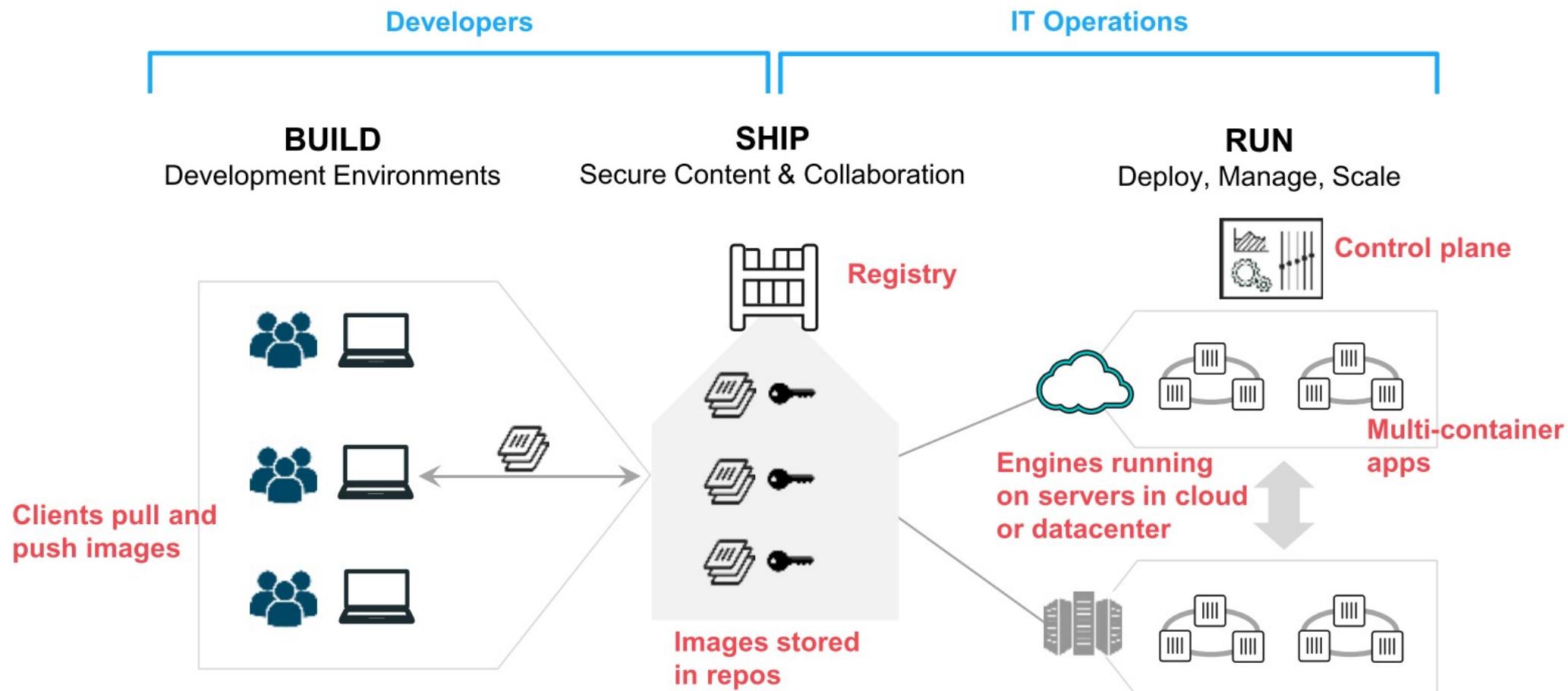
IT OPERATIONS



Building a Software Supply Chain



Containers as a Service



Building a Secure Supply Chain

Container App Lifecycle Workflow

Private Image Registry	Secure Access and User Management	Application and Cluster Management
Image Scanning and Monitoring	Content Trust and Verification	Policy Management
Security	Network	Volumes
Distributed State	Container Runtime	Orchestration



Enterprise Edition



Docker Engine



Usable
Security

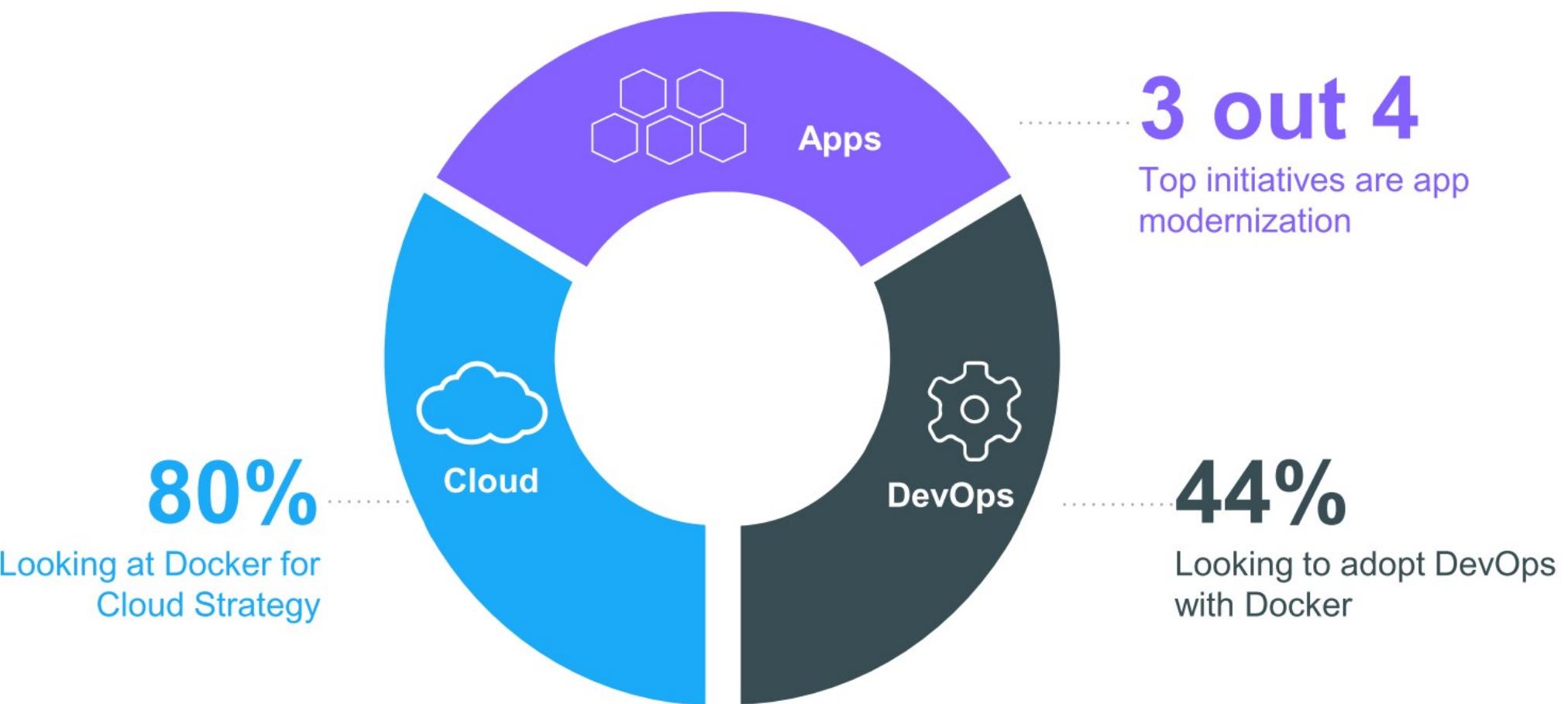


Trusted
Delivery



Portable

Docker Aligns to Multiple IT Initiatives



Docker Is in the Enterprise



Service Provider



Healthcare & Science



Financial Services



Tech



Insurance



Public Sector



Docker delivers agility, security and cost savings



Hardened containers deliver new levels of security to monoliths on the transition to microservices



Transform monoliths to secure and agile DevOps environments



Reduce maintenance costs by 10X for legacy, commercial and new apps

Docker delivers agility, resiliency, portability security and cost savings for all applications

Commercial Off
The Shelf Apps

Homegrown
Traditional Apps

Microservices
Apps

13X

More software releases

65%

Reduction in developer
onboarding time

~47%

Reduction in VMs, OS licensing
and Server costs

Eliminate

“works on my machine”
issues

62%

Report reduction in MTTR

10X

Cost reduction in maintaining
existing applications

One platform and one journey for all applications

1

Traditional apps in containers

Gain portability, efficiency and security



2

Transform to Microservices

Look for shared services to transform



3

Accelerate New Applications

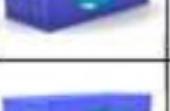
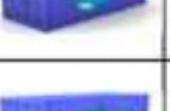
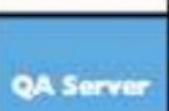
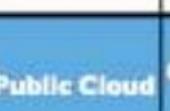
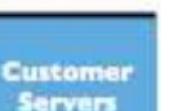
Greenfield innovation



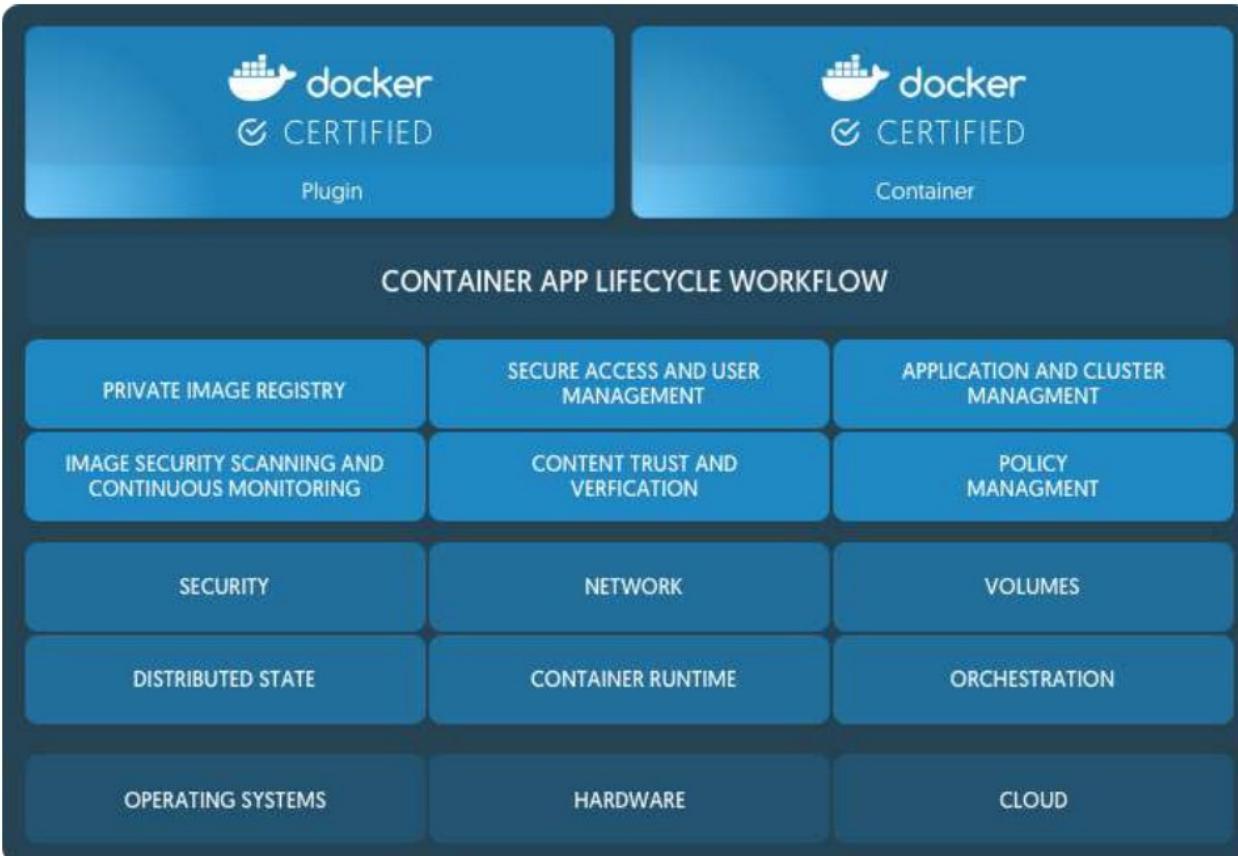
Multiple Stacks, Multiple Stages = Complexity

	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	

Solving the deployment matrix

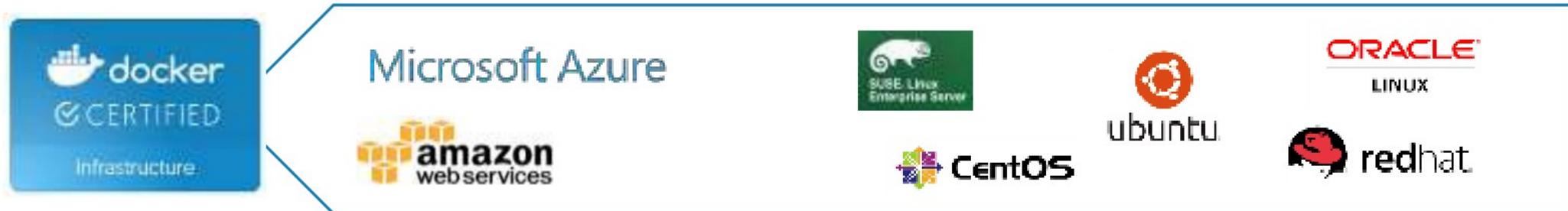
	Static website	Web frontend	Background workers	User DB	Analytics DB	Queue	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers	
Static website														
Web frontend														
Background workers														
User DB														
Analytics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

Docker Enterprise Edition (EE) is the CaaS enabled platform for developers and IT



- Integrated orchestration, security and management
- Predictable quarterly releases with one year of support and maintenance
- Security patches and hotfixes backported to all supported versions
- Enterprise class support (9am-6pm or 24x7x365)
- Certified Infrastructure, Containers and Plugins

Certified technology extend the platform with assurances and support



END OF LECTURE 11