Université IBM i 2018

16 et 17 mai

IBM Client Center Paris

Session **S34 - Comment développer les applications de demain ?**

Erwan Marechal - DevOps Leader France
erwan.marechal@fr.ibm.com

Benoit Marolleau – Cloud Architect IBM Systems
benoit.marolleau@fr.ibm.com

IBM **Systems** IBM **Cloud**

IBM **Client Center** Montpellier

Session S34

Comment développer les applications de demain ?

Université IBM i 2018 - 16 et 17 mai

IBM Client Center Paris

Erwan Marechal - DevOps Leader France erwan.marechal@fr.ibm.com

Benoit Marolleau – Cloud Architect IBM Systems benoit.marolleau@fr.ibm.com

DRAFT

Plan de la présentation

- Introduction - DevOps & Innovation Continue
- Les nouveaux outils et modèles de développement
  - Cloud Computing, Containers & Technologies Docker, Kubernetes , Microservices
- IBM Cloud Private
  - Présentations de ICP, IBM Microservice Builder
- Intégration avec mes applications & développements IBM i
  - Prise en compte des ces technologies et Intégration avec l'existant IBM i
  - Short Demo – ICp next to your IBM i

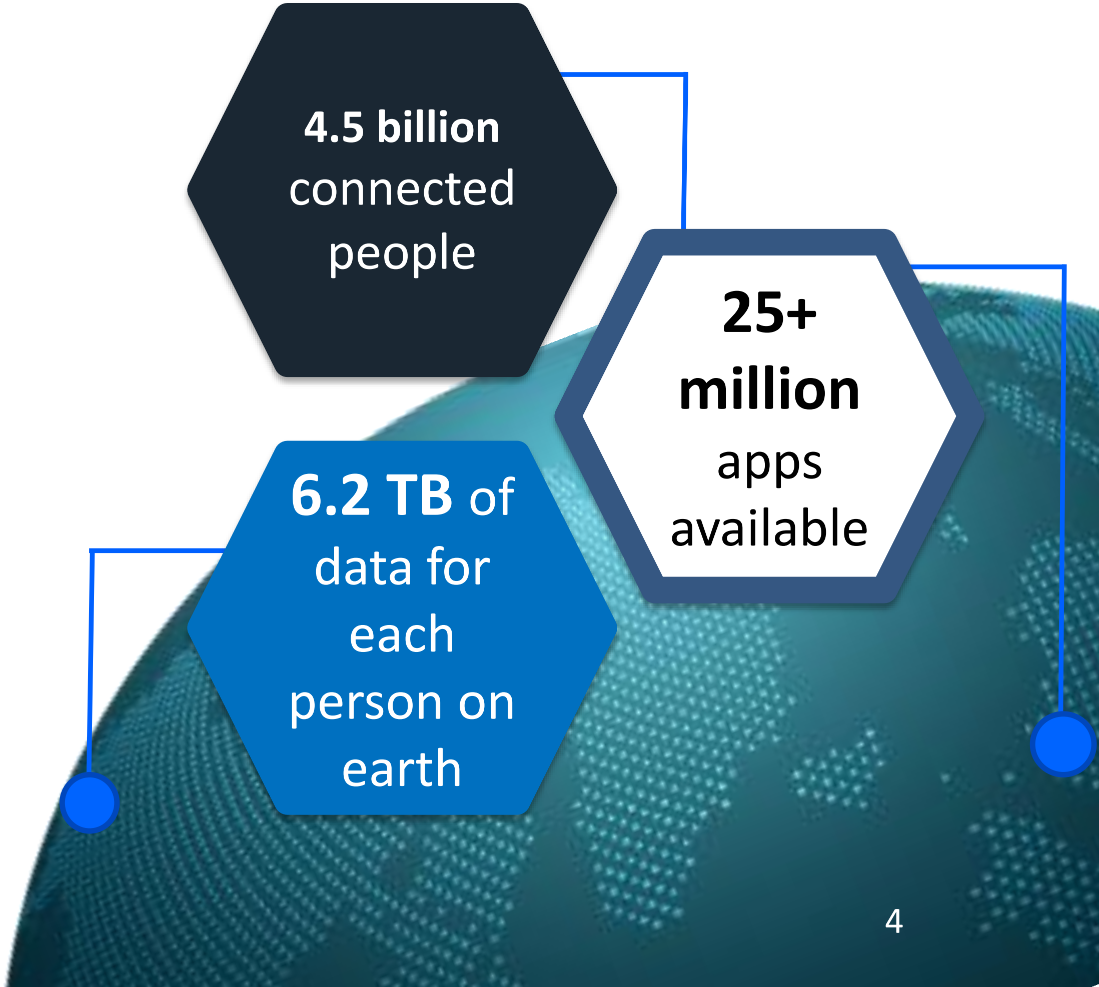
The world is becoming more connected than ever

Businesses must be ready to face the challenge

To win in the connected economy, **enterprises are focusing on interactions and value exchange** across a partner ecosystem

You need:

- A better, more compelling customer experience
- An infrastructure that scales out autonomously
- To bring teams together across a partner ecosystem
- Continuous innovation to deliver software faster, consistently, and reliably



4.5 billion connected people

25+ million apps available

6.2 TB of data for each person on earth

Source:[1] IDC: The Digital Universe of Opportunities: Rich Data and the Increasing Value of Internet of Things, April 2014

Source:[2] RisingStack: How Enterprises Benefit From Microservices Architecture, February 2016

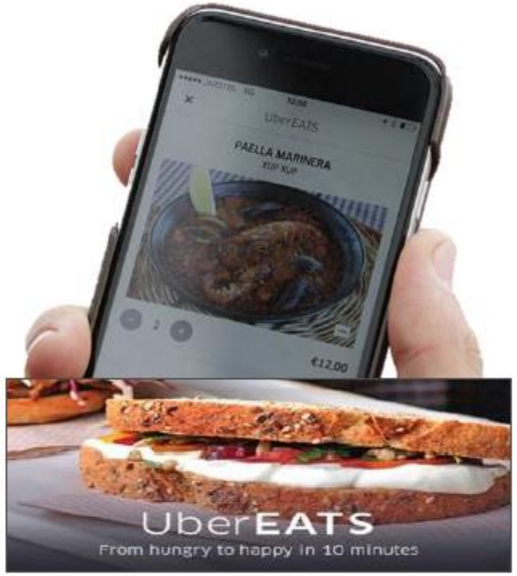
Customers and ecosystem partners expect innovative and personalized experiences

Location Aware

FREE
25%
ADMIT ONE
SALE

Starbucks now
Hi Phil! You are near Starbucks Oxford st. Currently 50% off on your favorite drink: Vanilla Latte!
slide to open

On Demand



Personalized and Engaging

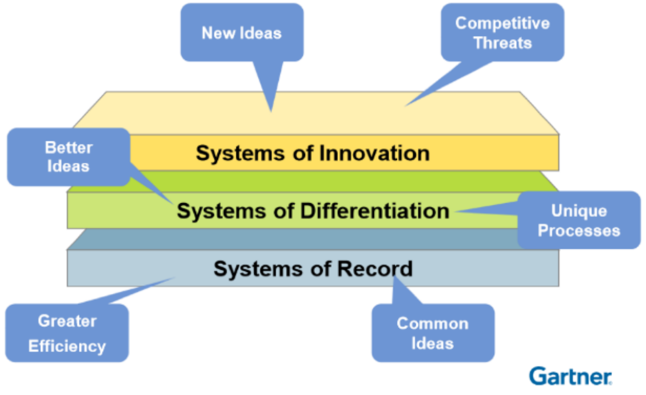
75%

of application development supporting digital business will be built not bought by 2020

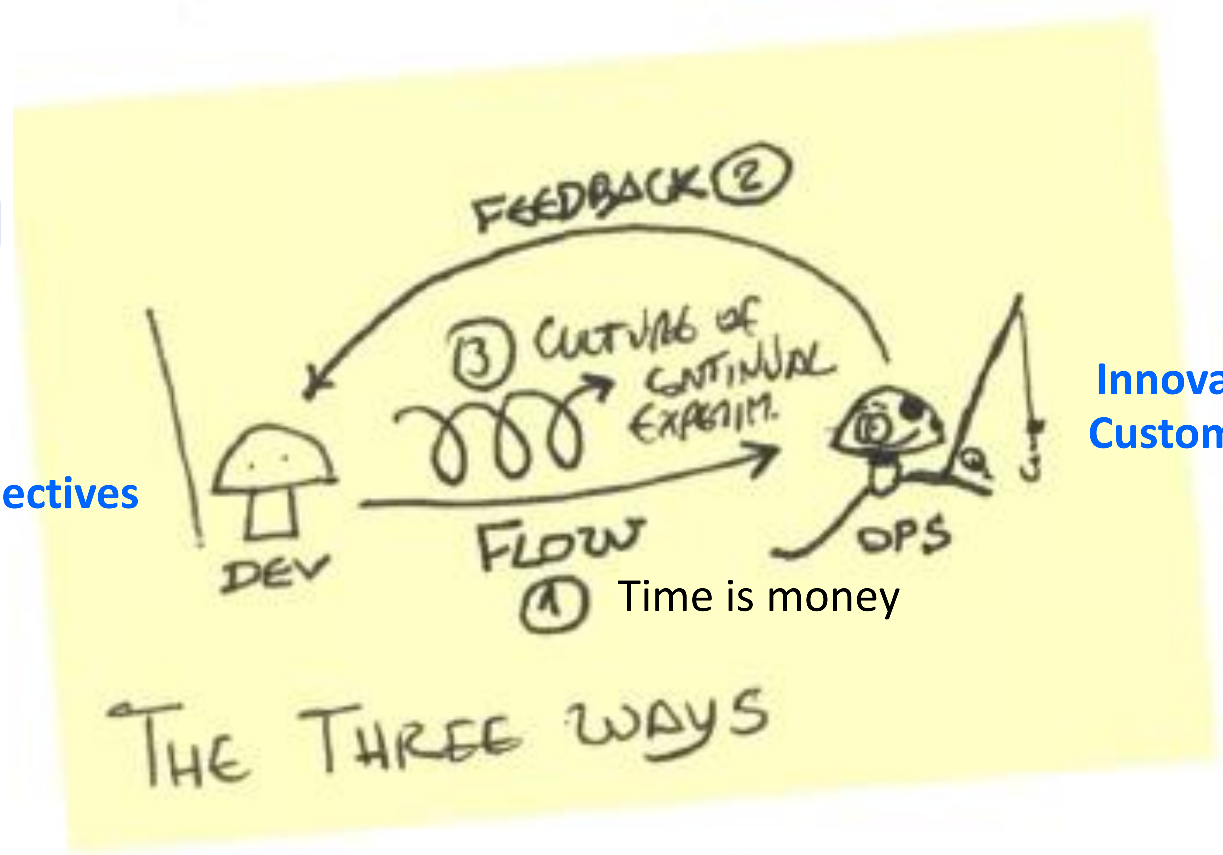
Source: Gartner

DevOps & Innovation continue

- DevOps “3 Ways”



Business Objectives & Ideas

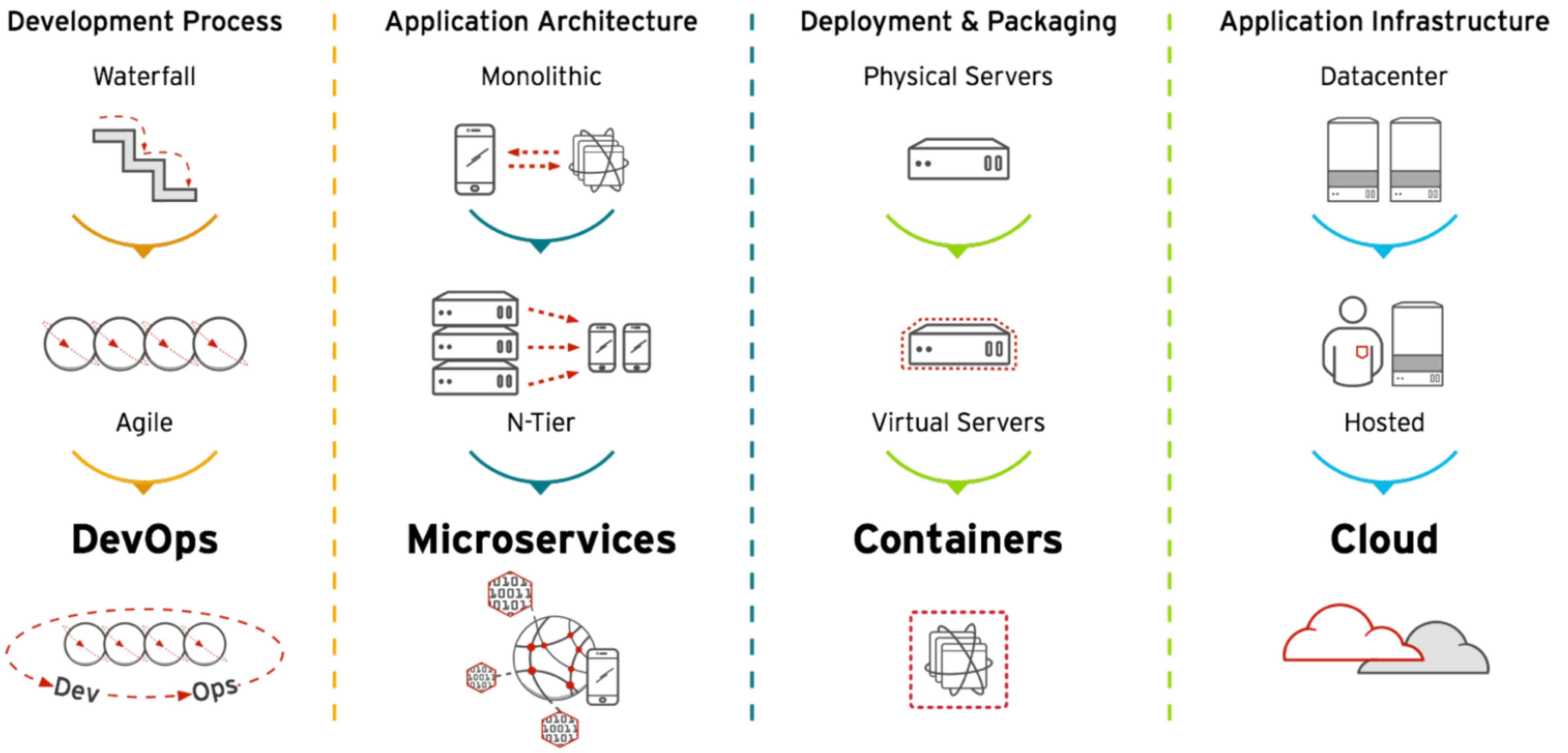


Time is money

1. Accelerate Delivery
2. Feedback Loop
3.Continuous Innovation

Innovation
Customer Experience

L’IT en phase avec les nouveaux besoins



Development Process

Waterfall

Agile

DevOps

Application Architecture

Monolithic

N-Tier

Microservices

Deployment & Packaging

Physical Servers

Virtual Servers

Containers

Application Infrastructure

Datacenter

Hosted

Cloud

Containers

Containers vs. Machines virtuelles

Build

Ship

Run

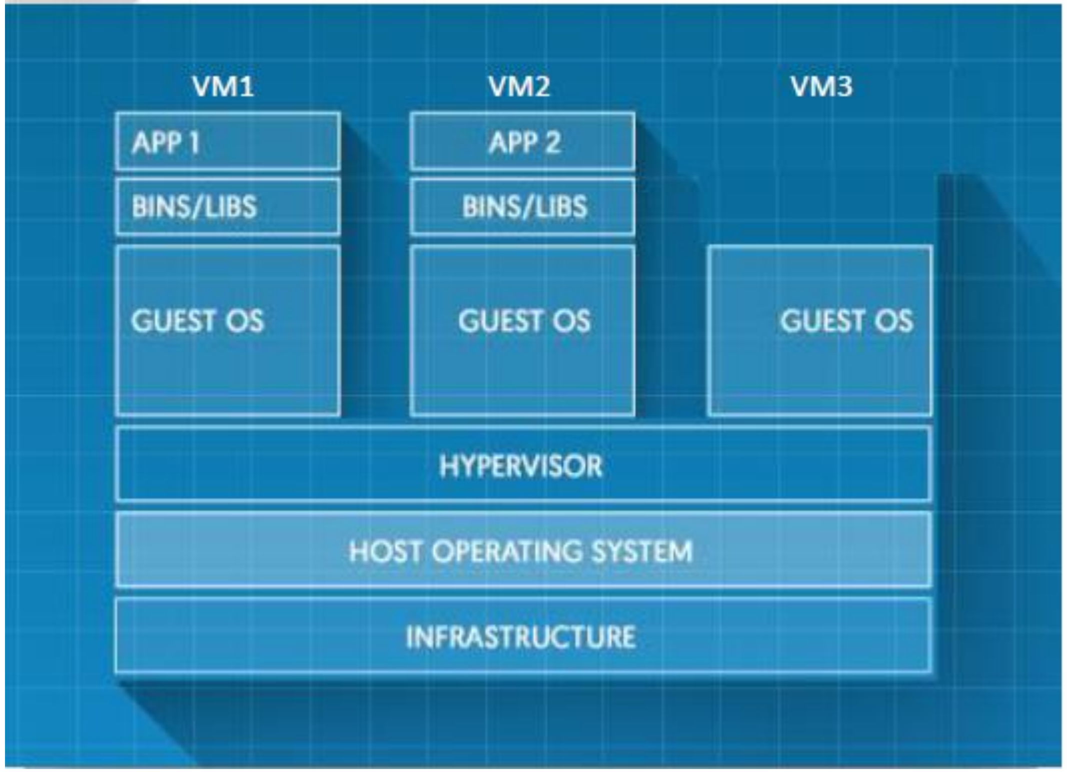
Docker: Application portability

A standard container that is loaded with goods and stays sealed until its final delivery

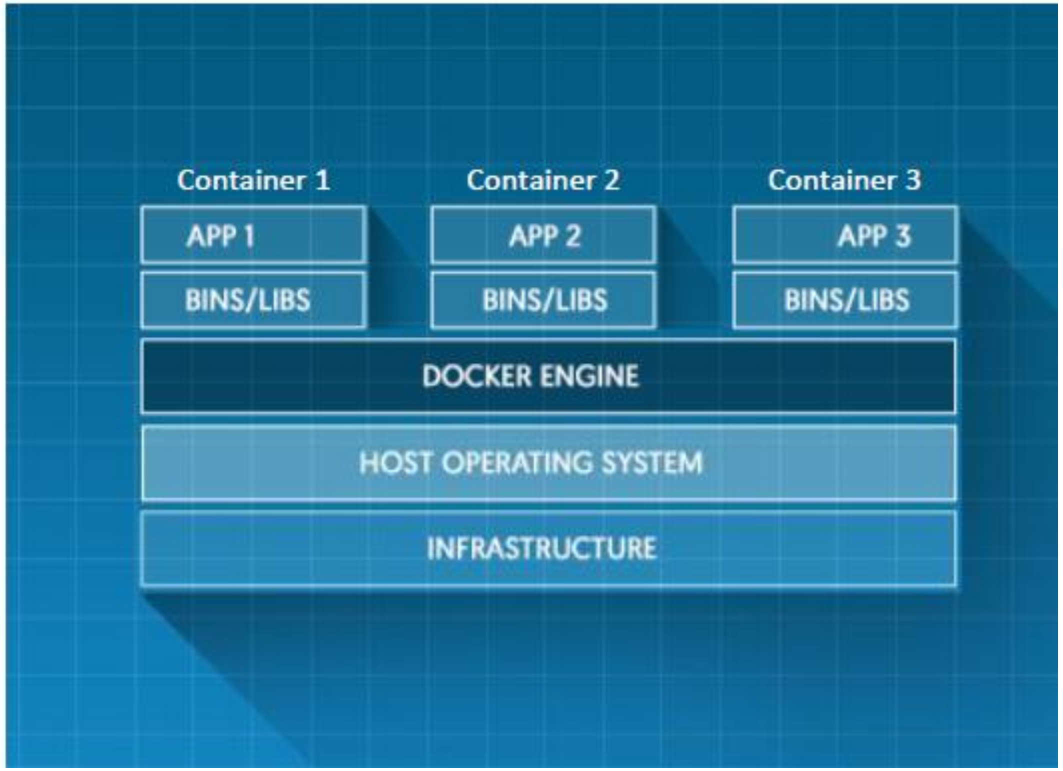
Can be **loaded, unloaded, stacked, transported** efficiently over long distances **with any transport's mode**

Containers vs Machines Virtuelles

Virtual Machines



Containers



Virtualization Pro:

- Better Security / Isolation
- Allow different Kernel between VMs
- Not Limited to Linux OS

Containers Pro:

- Better resources utilization
- Less overhead compare to VM
- Light compare to VM
- Very FAST START : No Boot
- No special hypervisor mode access required

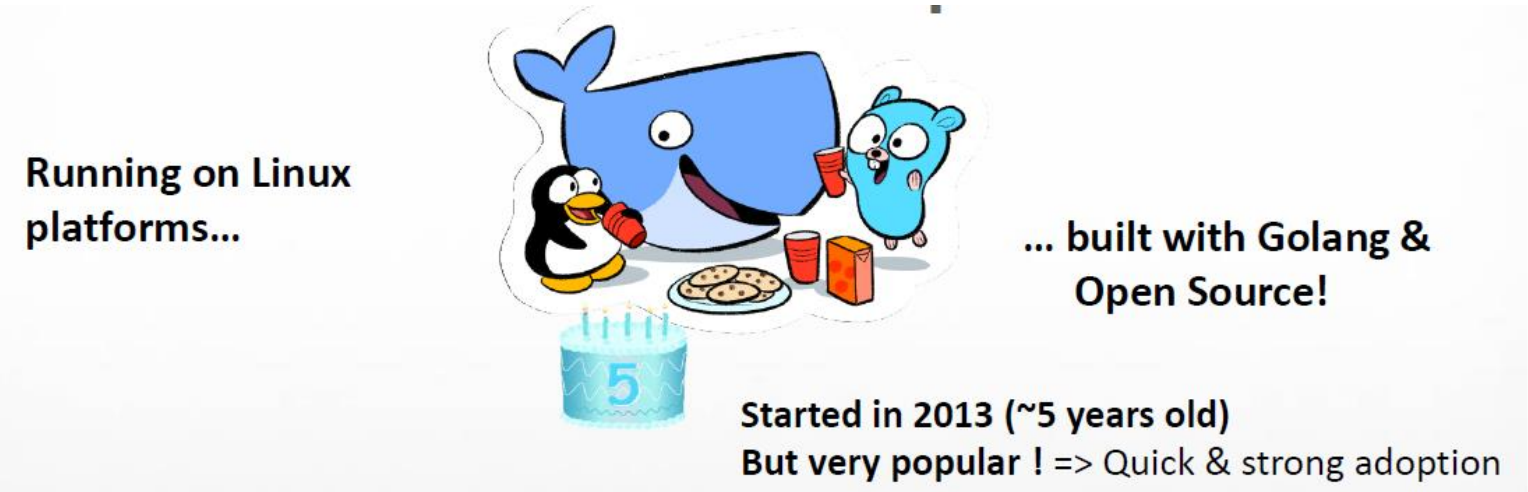
=> could be nested without performance impact.

Docker : Concepts

- **Engine**: Runs on Linux, it provides the operating environment for Docker containers.
- **Image:** Read-only templates for containers, stored and managed in a registry.

  Once instantiated a container is created.
- **Dockerfile:** Defines a Docker image as if it was code; used to re-build an image
- **Registry:** A service that allows to store and manage Docker images
- **Container:** Standard unit to package an application and its dependencies: binaries, libraries, system tools…

  So that it can be moved between environments and run without changes.



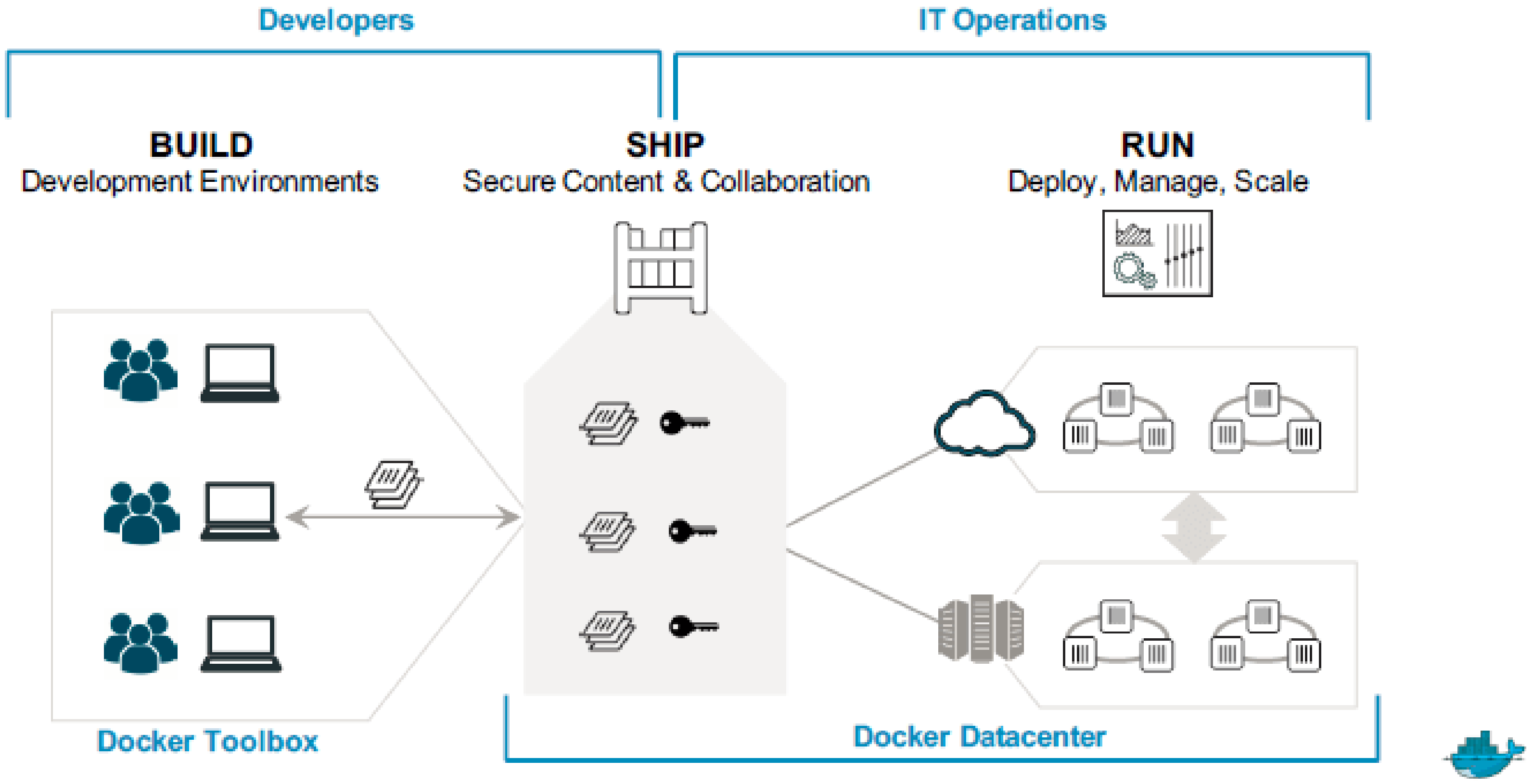
Running on Linux platforms…

… built with Golang & Open Source!

Started in 2013 (~5 years old)

But very popular ! => Quick & strong adoption

Containers & DevOps

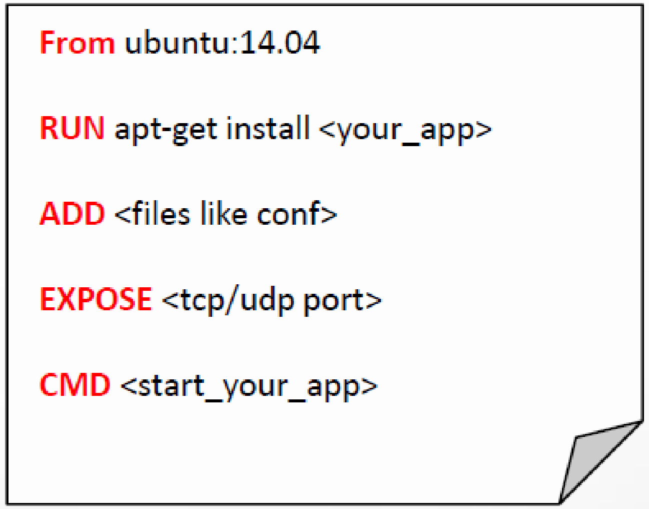


IBM Cloud amazon webservices™ Microsoft Azure

DevOps: Break down barriers between Dev and Ops teams to improve the app development process

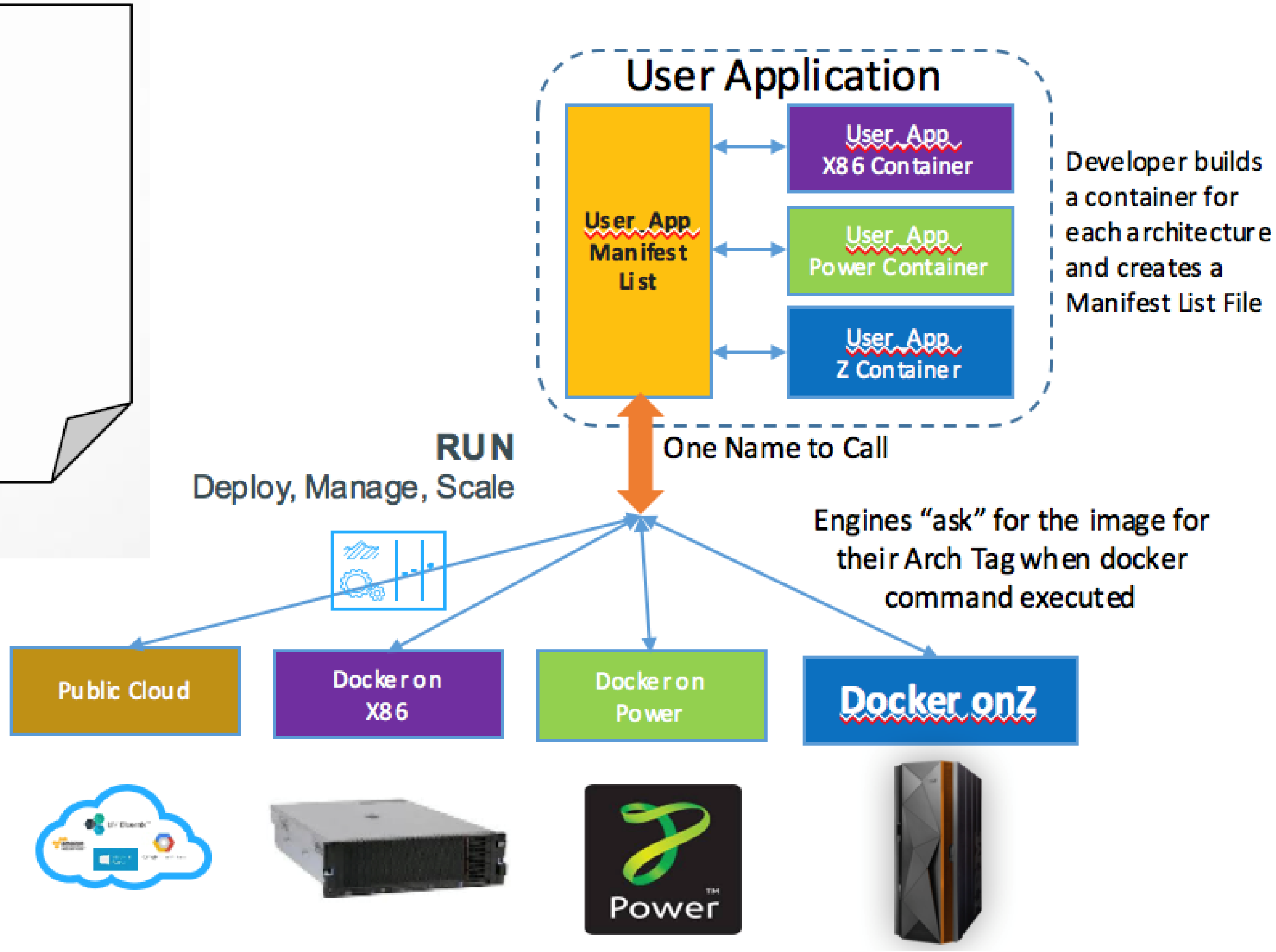
CI/CD: Enable developers to develop and test applications more quickly and within any environment

Multi-Arch & Multi-Cloud Enablement for Docker



```
From ubuntu:14.04

RUN apt-get install <your_app>

ADD <files like conf>

EXPOSE <tcp/udp port>

CMD <start_your_app>
```

Dockerfile example



User Application

User_App Manifest List

User_App X86 Container

User_App Power Container

User_App Z Container

Developer builds a container for each architecture and creates a Manifest List File

One Name to Call

RUN
Deploy, Manage, Scale

Engines "ask" for the image for their Arch Tag when docker command executed

Public Cloud

Docker on X86

Docker on Power

Docker onZ

Key use cases for Containerization

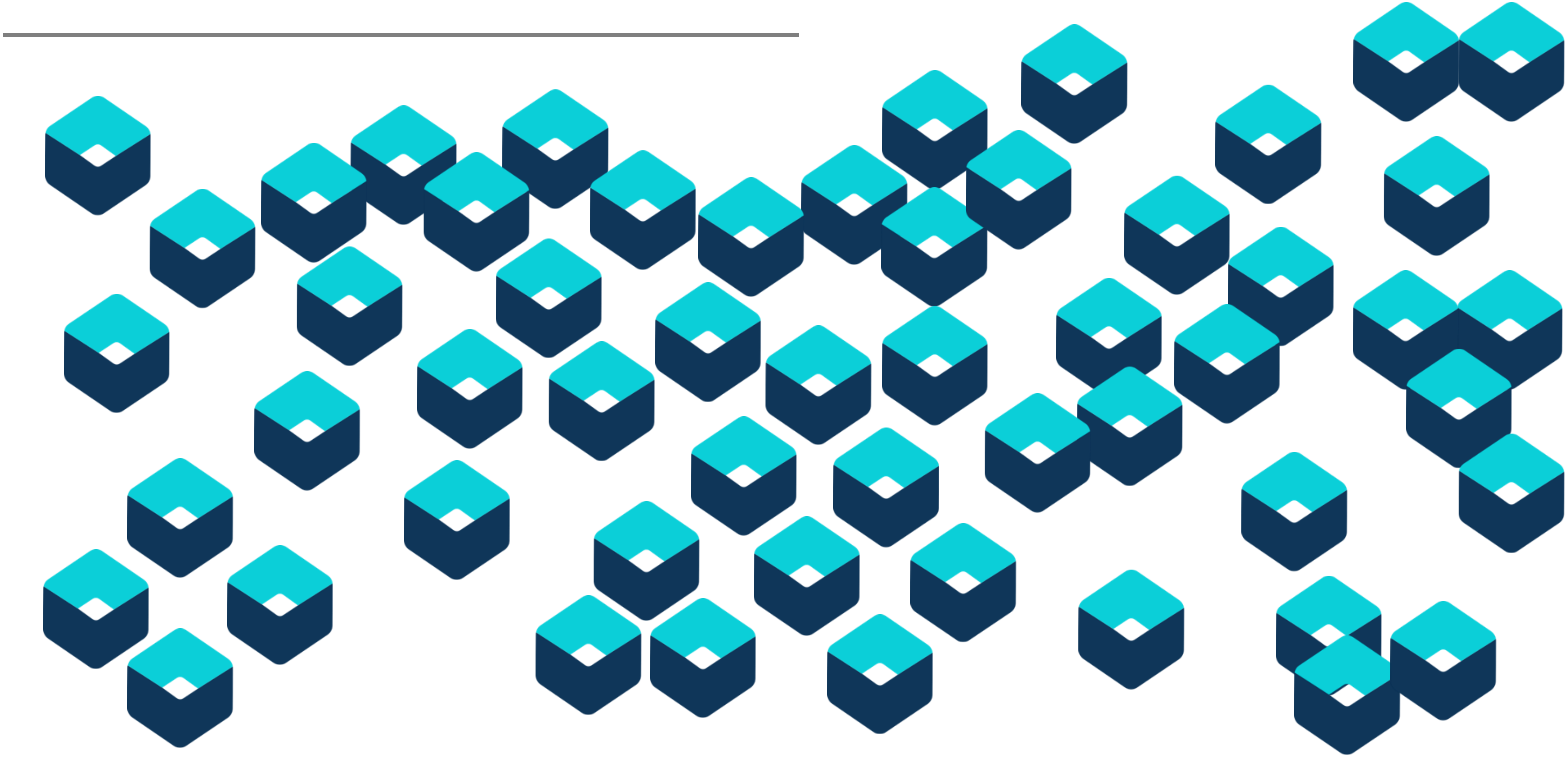
App Modernization in the Enterprise

1 **Containerize Legacy Applications**
Lift and shift for portability and efficiency

2 **Transform Legacy to Microservices**
Look for shared services to transform

3 **Accelerate New Applications**
Greenfield innovation

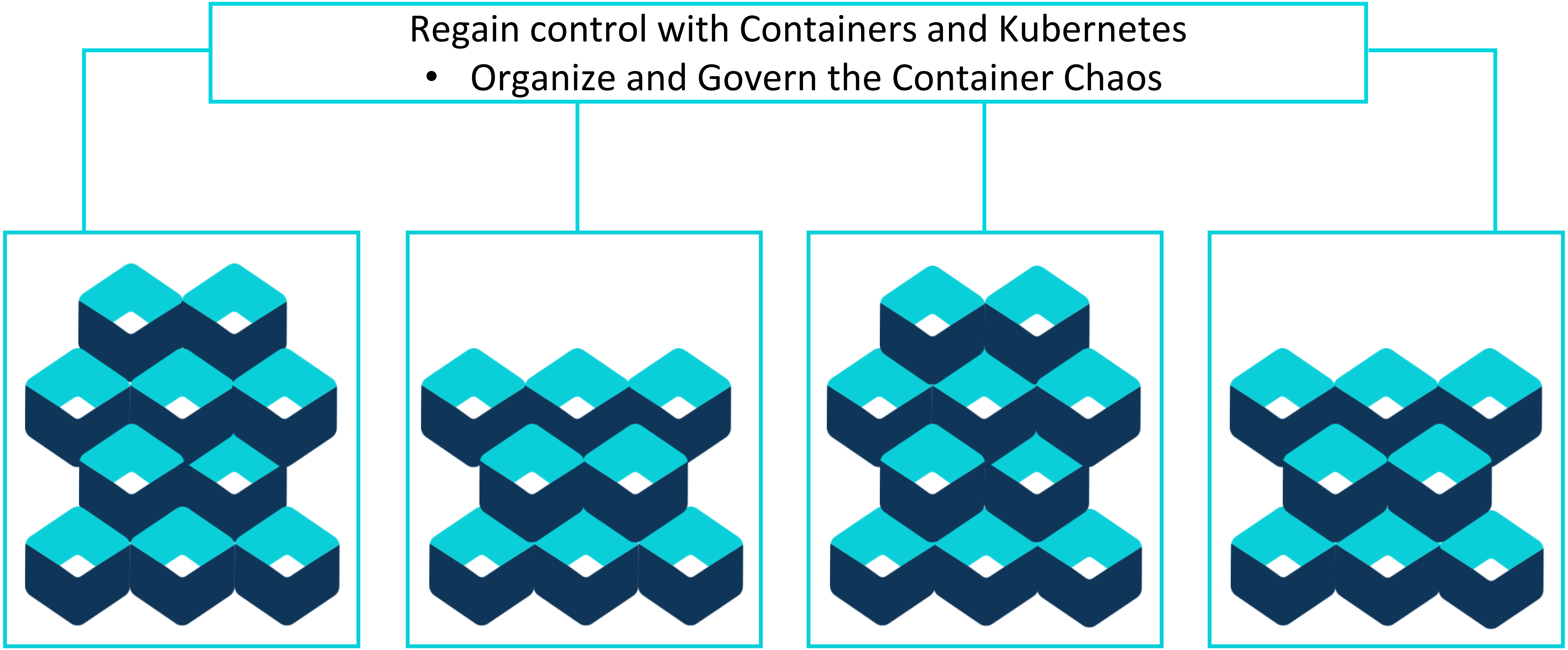
Containers are great but … can lead into lack of control & chaos



Kubernetes – (Κυβερνήτης - Captain in Greek)

Regain control with Containers and Kubernetes

- Organize and Govern the Container Chaos



What do Kubernetes really offer ?

Intelligent Scheduling

Automatically places containers based on their resource requirements and other constraints, while not sacrificing availability. Mix critical and best-effort workloads in order to drive up utilization and save even more resources.

Self Healing

Restarts containers that fail, replaces and reschedules containers when nodes die, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

Horizontal Scaling

Scale your application up and down with a simple command, with a UI, or automatically based on CPU usage.

Service Discovery and Load Balancing

No need to modify your application to use an unfamiliar service discovery mechanism. Kubernetes gives containers their own IP addresses and a single DNS name for a set of containers, and can load-balance across them.

Automated rollout and rollback

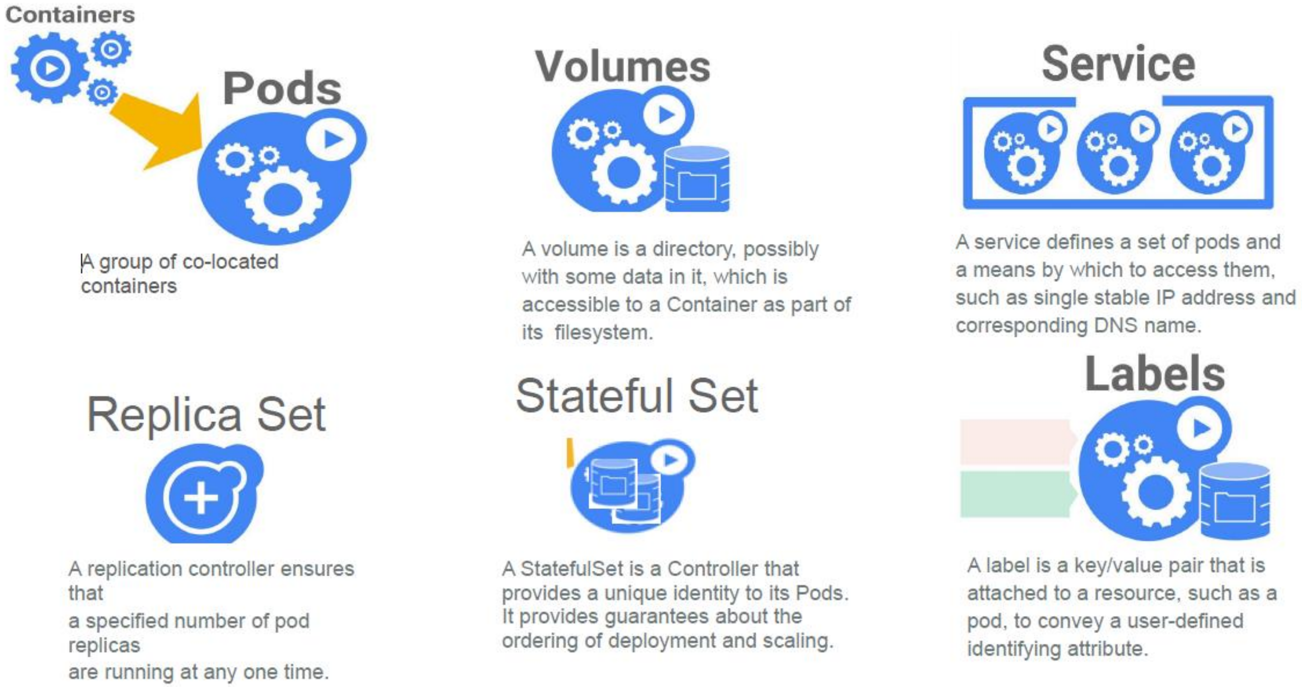
Kubernetes progressively rolls out changes to your application, while monitoring application health to ensure it doesn't kill all your instances at the same time. If something goes wrong, Kubernetes will rollback the change for you. Take advantage of a growing ecosystem of deployment solutions.

Secret and configuration management

Deploy and update secrets and application configuration without rebuilding your image and without exposing secrets in your stack configuration.

Kubernetes Concepts

- Declarative Configuration (YAML) & Decoupling
  - Services, loosely coupled apps
- Consistency / Scaling
  - Application SLA vs. OS SLA
- Abstraction layer
  - K8s is present in all Cloud Providers
  - Pods, or groups of containers =
  - Kubernetes services =
  - Namespaces =



A group of co-located containers

A volume is a directory, possibly with some data in it, which is accessible to a Container as part of its filesystem.

A service defines a set of pods and a means by which to access them, such as single stable IP address and corresponding DNS name.

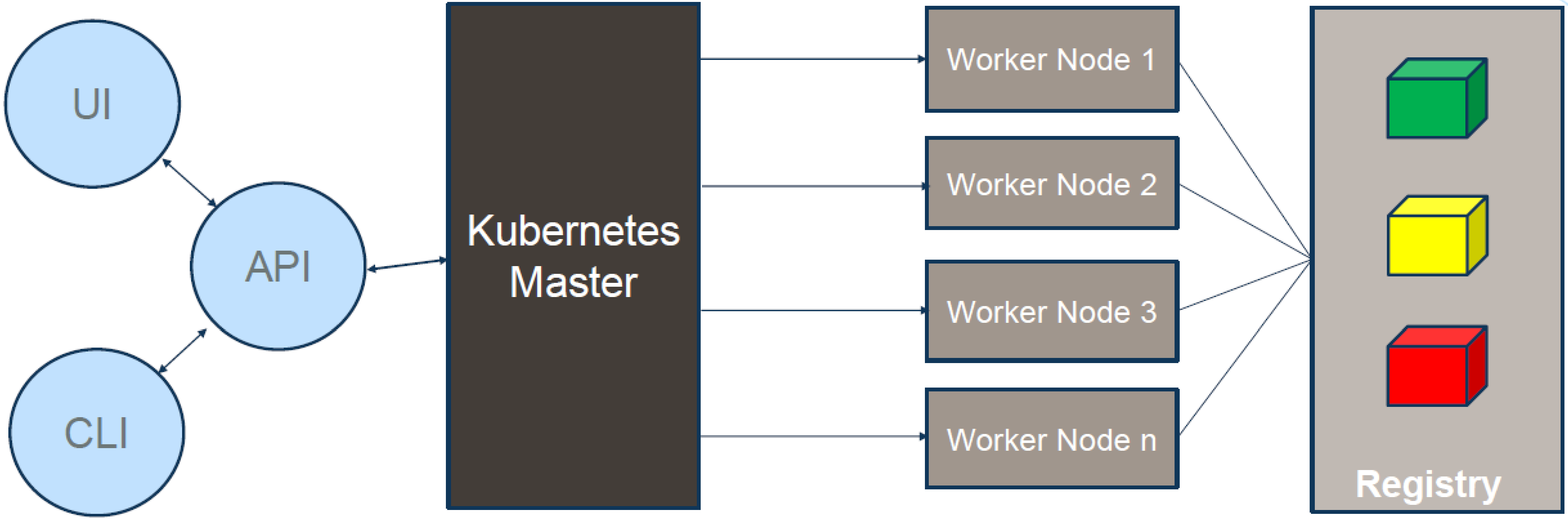
A replication controller ensures that
a specified number of pod replicas
are running at any one time.

A StatefulSet is a Controller that provides a unique identity to its Pods. It provides guarantees about the ordering of deployment and scaling.

A label is a key/value pair that is attached to a resource, such as a pod, to convey a user-defined identifying attribute.

- Efficiency
  - Machine usage optimization – distribution of application

Kubernetes Architecture



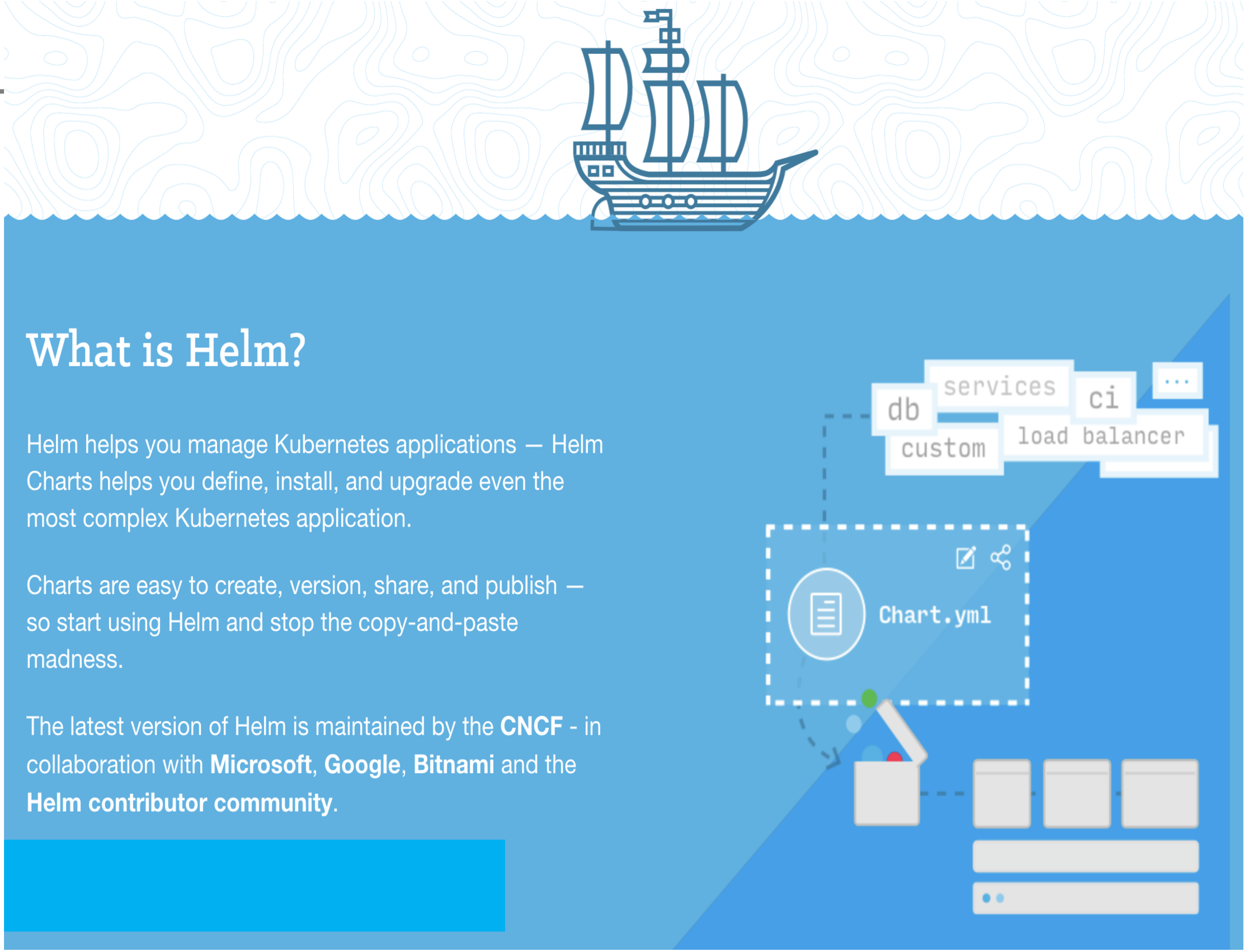
- Etcd
- API Server
- Controller Manager Server
- Scheduler Server

And HELM is …

The package manager for Kubernetes

Helm is the best way to find, share, and use software built for [Kubernetes](https://kubernetes.io).

Tells Kubernetes all it needs to know about an application its parameters and dependencies



What is Helm?

Helm helps you manage Kubernetes applications — Helm Charts helps you define, install, and upgrade even the most complex Kubernetes application.

Charts are easy to create, version, share, and publish — so start using Helm and stop the copy-and-paste madness.

The latest version of Helm is maintained by the **CNCF** - in collaboration with **Microsoft**, **Google**, **Bitnami** and the **Helm contributor community**.

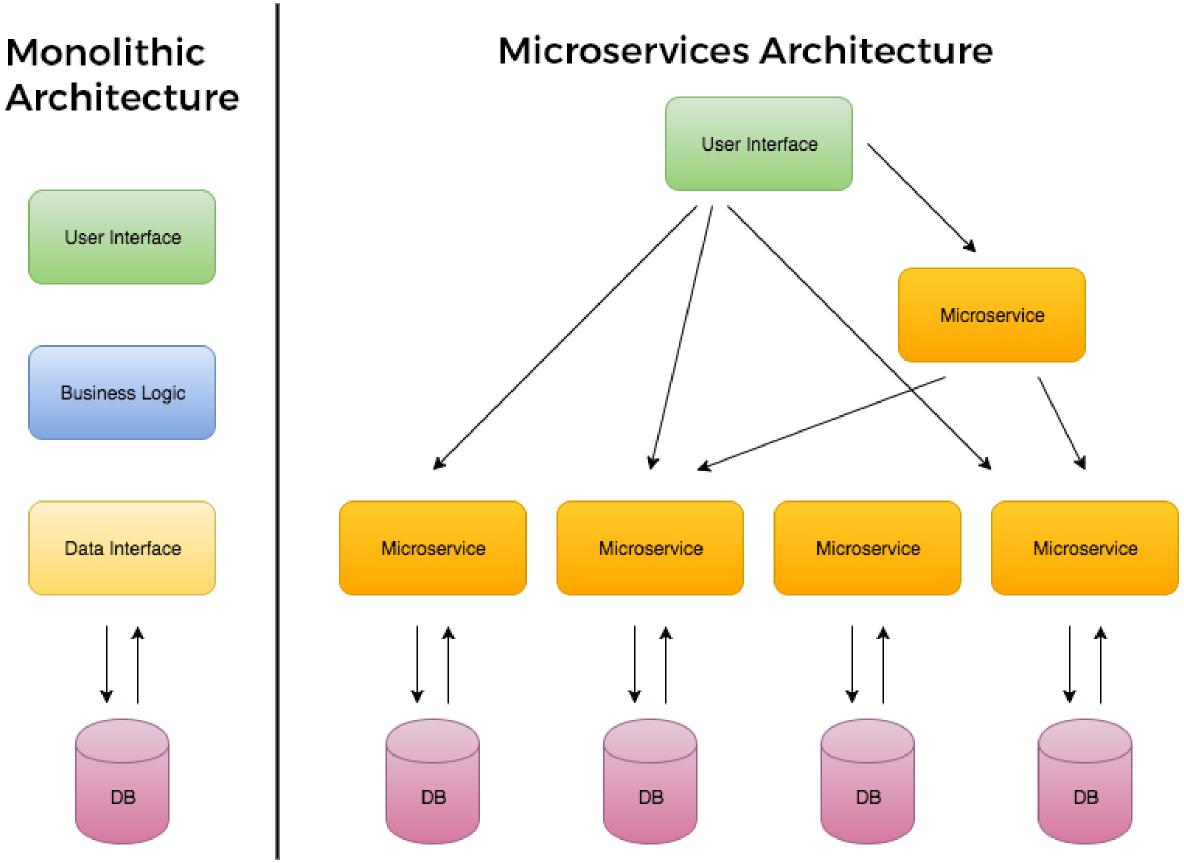
Microservices

Microservice Approach

Architectural Evolution

Spaghetti Architecture	Lasagna Architecture	Ravioli Architecture
Cut & Paste (1990's)	Layered Monolith (2000's)	Microservices (2010's)

Microservices & Cloud Native Apps


Monolithic Architecture

Microservices Architecture

Why microservices?

The microservices revolution

Connecting people and digital apps better than ever before

A microservices architecture is gaining traction for developing and delivering cloud-native workloads across public, private, and hybrid application environments

60%

60% of new apps will use cloud-enabled continuous delivery and cloud-native application architectures to enable faster innovation and business agility**

IDC FutureScape: Worldwide Cloud 2016 Predictions – Master the Raw Material of Digital Transformation, November 2015

Why?

- ✓ Decomposed into small pieces
- ✓ Loosely coupled
- ✓ Easier to scale development
- ✓ Improved fault isolation
- ✓ Each service can be developed and deployed independently
- ✓ Eliminates any long-term commitment to a technology stack

Why 12 factor apps?

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

https://12factor.net/

Code

I. Codebase
One codebase tracked in revision control, many deploys

V. Build, release, run
Strictly separate build and run stages

X. Dev/prod parity
Keep development, staging, and production as similar as possible

Deploy

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

Operate

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

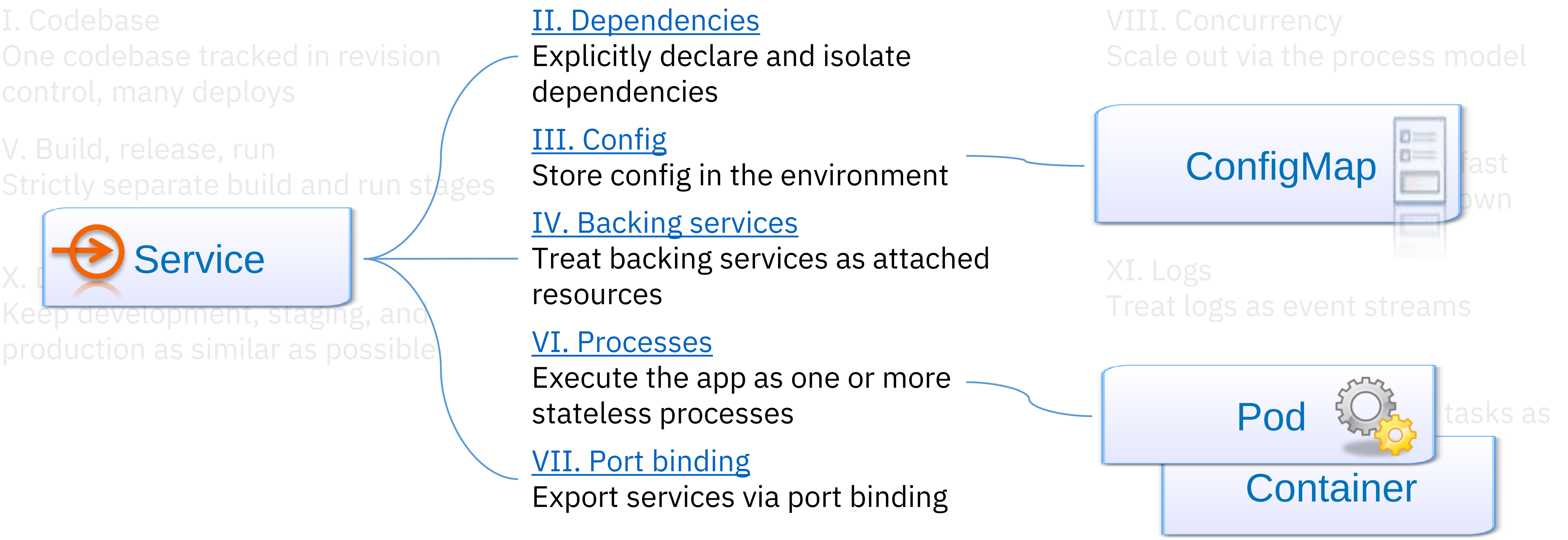
XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

Code

Deploy

Operate



I. Codebase
One codebase tracked in revision control, many deploys

V. Build, release, run
Strictly separate build and run stages

X. D
Keep development, staging, and production as similar as possible

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

fast
own

XI. Logs
Treat logs as event streams

tasks as

Code

I. Codebase
One codebase tracked in revision control, many deploys

V. Build, release, run
Strictly separate build and run stages

X. Dev/prod parity
Keep development, staging, and production as similar as possible

Deploy

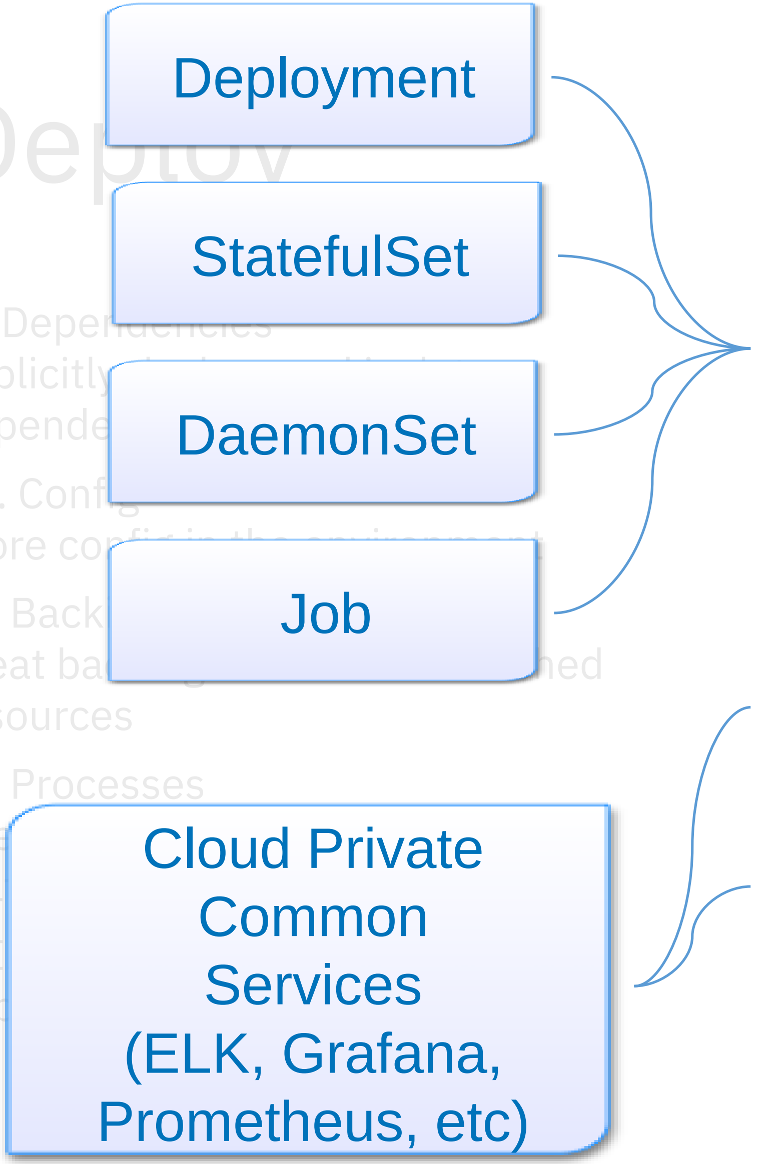
II. Depe
Explicitly
depende

III. Conf
Store co

IV. Back
Treat ba ned
resources

VI. Processes
Exe
sta

VII
Exp



Deployment

StatefulSet

DaemonSet

Job

Cloud Private Common Services (ELK, Grafana, Prometheus, etc)

Operate

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

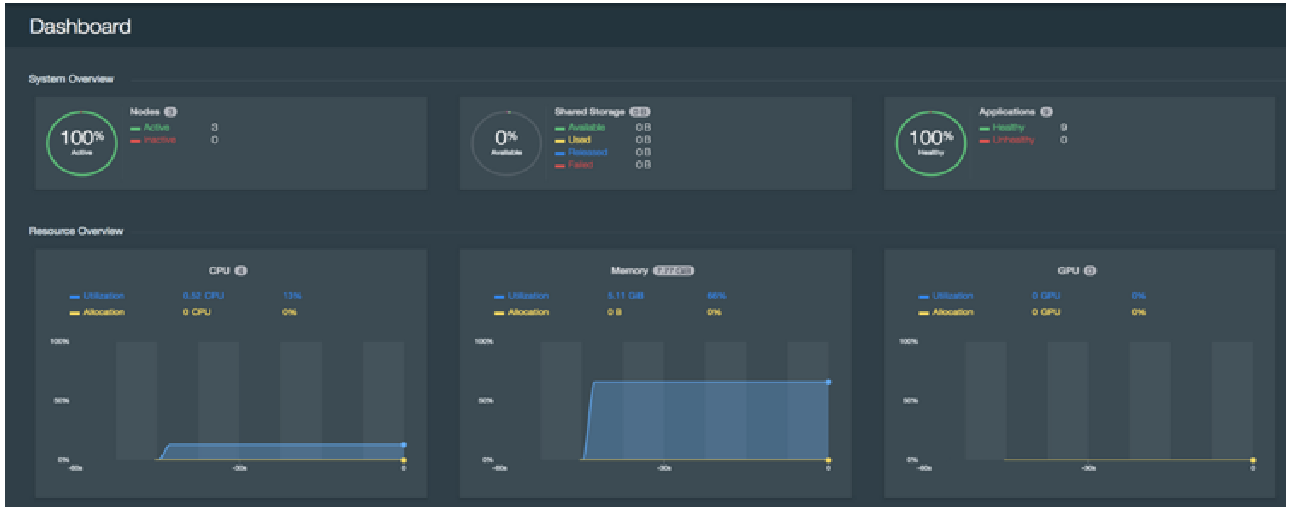
XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

IBM Cloud Private

IBM Cloud Private

- A private cloud platform for enterprises to develop and run their workloads locally
- An integrated platform consisting of PaaS and developer services necessary to create, run, and manage cloud applications
- Container infrastructure, orchestration and management
  - ✓ Resource management
  - ✓ Application life-cycle management/schedule/deployment
  - ✓ Scaling, rolling upgrade
  - ✓ Service registry/discovery
  - ✓ Distributed storage management
  - ✓ Image/software repository management
  - ✓ Configuration management
  - ✓ User/Account management



Dashboard

System Overview

Nodes: 100% Active; Active 3; Inactive 0

Shared Storage: 0% Available; Available 0 B; Used 0 B; Released 0 B; Failed 0 B

Applications: 100% Healthy; Healthy 9; Unhealthy 0

Resource Overview

	CPU		Memory		GPU	
Utilization	0.52 CPU	13%	5.11 GiB	66%	0 GPU	0%
Allocation	0 CPU	0%	0 B	0%	0 GPU	0%

kubernetes

Kubernetes based container platform

Industry leading container orchestration platform

Common Services

Simplify operations management, DevOps, and hybrid integration

IBM Middleware, Data and Analytics services

Optimize current investments and rapidly innovate

IBM **Cloud** Private Solution Overview

IBM Middleware & Open Source – e.g. Data, Analytics and Developer Services
Cloud-enabled middleware, application runtimes, messaging, databases & analytics to optimize current investments and rapidly innovate

IBM WebSphere MQ IBM DB2 IBM Data Science Experience open source

Core Operational Services

To simplify Operations Management, Security, DevOps, and hybrid integration

Grafana Prometheus

Kubernetes-based Container Platform

Industry leading container orchestration platform across private, dedicated & public clouds

kubernetes HELM

Cloud Foundry

For prescribed application development & deployment

CLOUD FOUNDRY

Terraform (CAM)

Infrastructure as Code for provisioning on public and on-prem cloud

HashiCorp Terraform

IBM Cloud Microsoft amazon web services openstack vmware

Runs on existing IaaS: vmware openstack Power Systems System Z IBM Spectrum Dell, Cisco, NetApp, Lenovo, …

IBM Cloud Private – Specific Use Cases

Airline

- Develop Chat app between tarmac personnel and pilot crews
- Makes use of on-prem APIs of flight logistics & scheduling info
- Requires low-latency
- Solution: Running microservices within containers on-prem

Industrial Client

- Requires standard deployment of software within factory environment
- Factories are geographically disperse and isolated with limited technical resources
- Solution: Leverage small ICP footprint with ability to synchronize catalogue content and approved workloads

Hospital

- Desire to leverage IBM Voice Gateway using on-prem environment for HA scalable deployment
- Analysis of text, roundtrip application
- Solution: Run IBM Voice Gateway and ICP for cloud native workloads

Bank

- Currently running main web portal on a self-managed (vanilla) Kubernetes deployment
- Need vendor to provide Kubernetes deployment in order to improve support and security posture
- Solution: Deploy ICP with full support of Kubernetes, Docker images, patch process, etc

IBM Cloud Private – Specific Use Cases

Use Case #1

Modernize and optimize existing applications

- Time to market acceleration
- Legacy or monolithic apps
- Existing WAS, MQ, DB2 infrastructure / migration
- DevOps initiatives and enterprise developers
- x86, Power and zLinux

Use Case #2

Opening up enterprise data centers to work with cloud services

- Securely open your datacenter
- GDPR
- API Economy
- Integrate public cloud services securely with your local cloud
- new web/mobile presence
- customer loyalty
- B2B initiatives

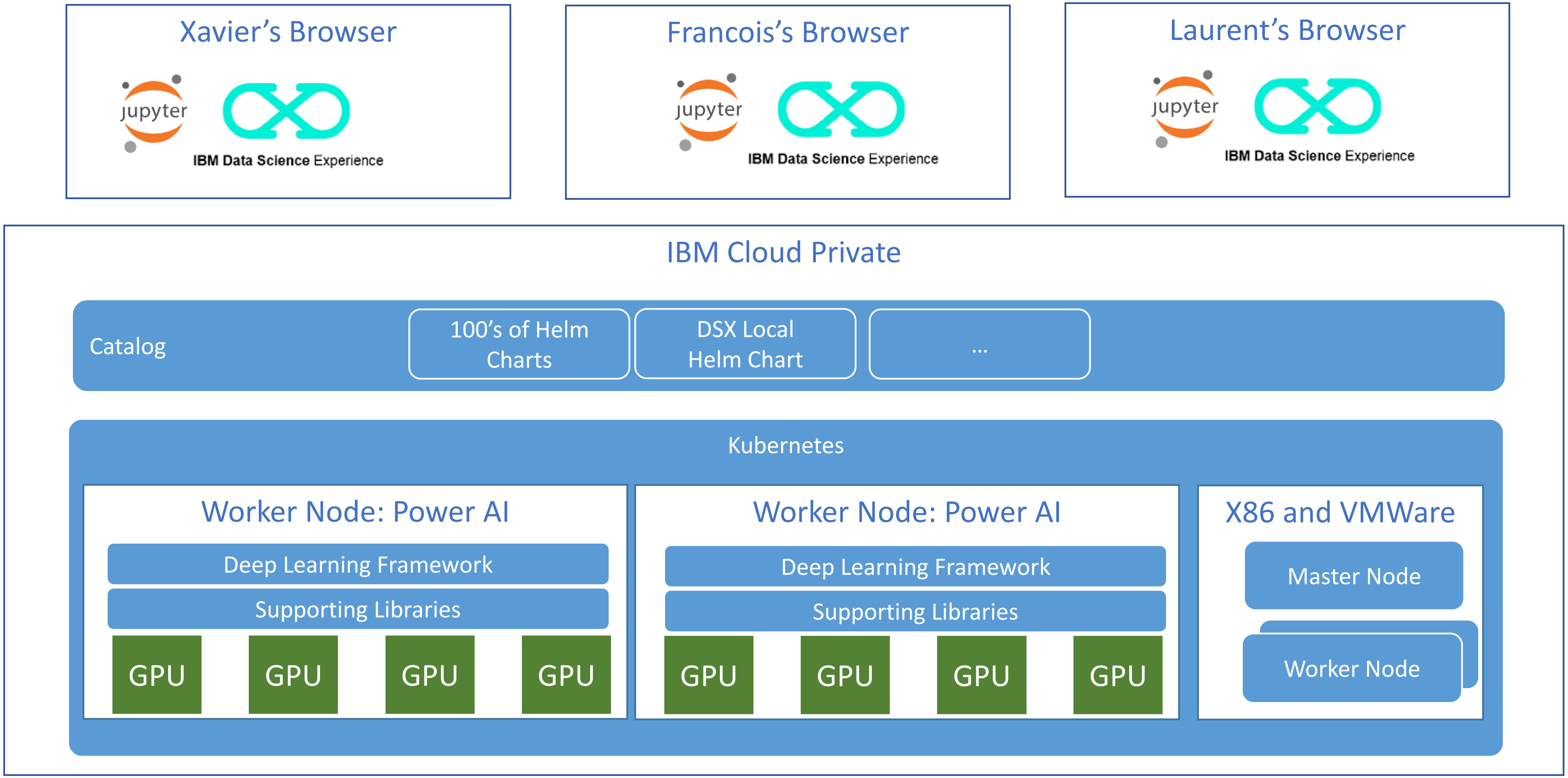
Use Case #3

Create new cloud-native applications

- New use cases
- IoT
- Blockchain
- Machine Learning
- Data science experience
- Building MicroServices

IBM Cloud Private

ICP Use case for Data Scientists



Xavier's Browser

jupyter

IBM Data Science Experience

Francois's Browser

jupyter

IBM Data Science Experience

Laurent's Browser

jupyter

IBM Data Science Experience

IBM Cloud Private

Catalog

100's of Helm Charts

DSX Local Helm Chart

…

Kubernetes

Worker Node: Power AI

Deep Learning Framework

Supporting Libraries

GPU GPU GPU GPU

Worker Node: Power AI

Deep Learning Framework

Supporting Libraries

GPU GPU GPU GPU

X86 and VMWare

Master Node

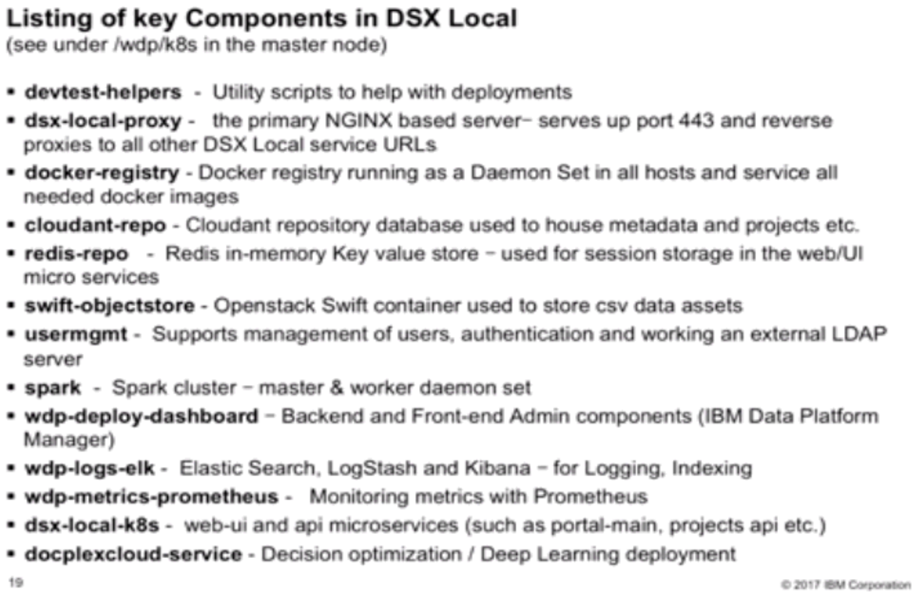
Worker Node

DSX Local Cluster & ICP – Internals (as of today: DSX 1.1.1.00 ppc64le)

Namespaces:
sysibmadm-data , sysibm-adm , dsxl-ml , ibm-private-cloud

PODs:
cloudant, redis, usermgmt, dsx-core, and ibm-nginx

Images:
27 images



Listing of key Components in DSX Local

(see under /wdp/k8s in the master node)

- **devtest-helpers** - Utility scripts to help with deployments
- **dsx-local-proxy** - the primary NGINX based server– serves up port 443 and reverse proxies to all other DSX Local service URLs
- **docker-registry** - Docker registry running as a Daemon Set in all hosts and service all needed docker images
- **cloudant-repo** - Cloudant repository database used to house metadata and projects etc.
- **redis-repo** - Redis in-memory Key value store – used for session storage in the web/UI micro services
- **swift-objectstore** - Openstack Swift container used to store csv data assets
- **usermgmt** - Supports management of users, authentication and working an external LDAP server
- **spark** - Spark cluster – master & worker daemon set
- **wdp-deploy-dashboard** – Backend and Front-end Admin components (IBM Data Platform Manager)
- **wdp-logs-elk** - Elastic Search, LogStash and Kibana – for Logging, Indexing
- **wdp-metrics-prometheus** - Monitoring metrics with Prometheus
- **dsx-local-k8s** - web-ui and api microservices (such as portal-main, projects api etc.)
- **docplexcloud-service** - Decision optimization / Deep Learning deployment

19

© 2017 IBM Corporation

Prefix/Suffix	image.repository	image.tag
cloudantRepo	privatecloud-cloudant-repo	v3.13.428
dsxConnectionBack	dsx-connection-back	1.0.4
dsxCore	dsx-core	v3.13.10
dsxScriptedML	privatecloud-dsx-scripted-ml	v0.01.2
filemgmt	filemgmt	1.0.2
hdpzeppelinDsxD8a2ls2x	hdpzeppelin-dsx-d8a2ls2x	v1.0.10
jupyterDsxD8a2ls2x	jupyter-dsx-d8a2ls2x	v1.0.11
jupyterDsxD8a3ls2x	jupyter-dsx-d8a3ls2x	v1.0.7
jupyterGpuPy35	jupyter-gpu-py35	v1.0.9
mlOnlineScoring	privatecloud-ml-online-scoring	v3.13.6
mlPipelinesApi	privatecloud-ml-pipelines-api	v3.13.4
mllib	ml-libs	v3.13.30
nginxRepo	privatecloud-nginx-repo	v3.13.6
pipeline	privatecloud-pipeline	v3.13.3
portalMachineLearning	privatecloud-portal-machine-learning	v3.13.20
portalMlaas	privatecloud-portal-mlaas	v3.13.17
redisRepo	privatecloud-redis-repo	v3.13.431
repository	privatecloud-repository	v3.13.2
rstudio	privatecloud-rstudio	v3.13.8
spark	spark	1.5.1
sparkClient	spark-client	v1.0.2
sparkaasApi	sparkaas-api	v1.3.14
spawnerApiK8s	privatecloud-spawner-api-k8s	v3.13.5
usermgmt	privatecloud-usermgmt	v3.13.5
utilsApi	privatecloud-utils-api	v3.13.5
wmlBatchScoring	wml-batch-scoring	v3.13.2
wmlIngestion	privatecloud-wml-ingestion	v3.13.2

Good news: ICP/K8s manages everything for you ☺

IBM Cloud Private Editions

Community

Platform

- Kubernetes (+ Helm)
- Core services
- Content catalog (Containers)

Freely Available in Docker Hub

Cloud Native

Platform

- Kubernetes (+ Helm)
- Core services
- Content catalog (Containers)

Cloud Foundry (Optional)

IBM Enterprise Software

- Microservice Builder
- WebSphere Liberty
- IBM SDK for node.js
- Cloud Automation Manager

Enterprise

Platform

- Kubernetes (+Helm)
- Core services
- Content catalog (Containers)

Cloud Foundry (Optional)

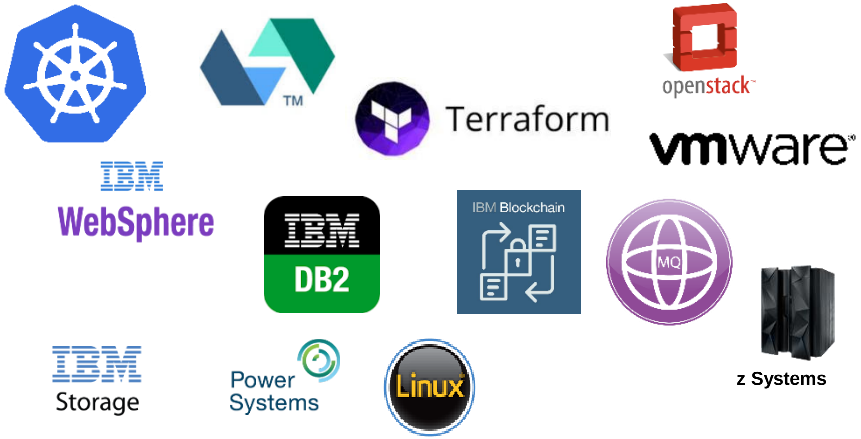
IBM Enterprise Software

Cloud Native Edition, plus:

+ WAS ND
+ MQ Advanced
+ API Connect Professional

IBM Cloud Private v2.1 (Example on POWER)

Exceptional density, performance and economics for the next generation of business transformation and optimized cognitive services



- A rich stack of built-in tools and services for Developers and powerful enterprise-grade management tools for Operators
- Single Kubernetes-based platform to address new application development as well as optimize existing applications– supporting developer agility and operational simplicity
- Automate, deploy, scale and manage containerized applications across multiple architectures, delivering better performance, density and efficiency
- Built-in dashboards and analytics – simplifying operations and management
- Add-in services to connect APIs and data, monitor and manage events - delivering enhanced integration with critical enterprise applications

- Deploy in any LE Linux partition including PowerVM, KVM on Power and AHV with Nutanix
- Deliver better performance for data and cognitive services, i.e.:
  - ✓ 157% higher container density and 145% more throughput compared to x86 when running WAS Liberty*
  - ✓ Record setting speed and accuracy for deep learning training – 16 days down to 7 hours -- a 58x speedup!*

GA: 10/24/17 Announce: 11/1/17

Cloud Native

Platform
- Kubernetes
- Core services (logging, etc)
- Cloud Automation Manager
- Content catalog

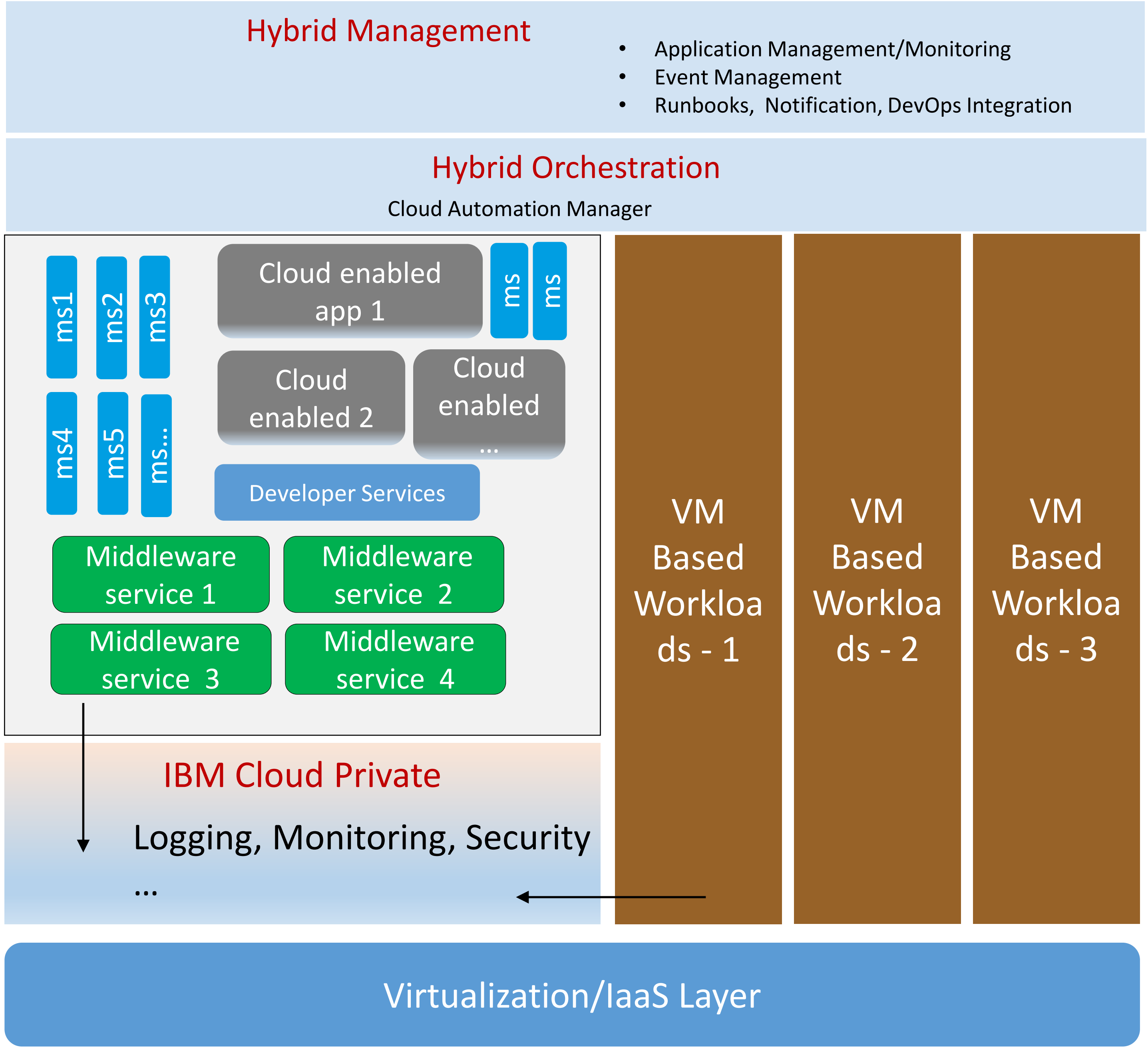
Bundled IBM Software
- Microservice Builder
- Liberty

Data Science

Platform
- Kubernetes/Cloud Foundry
- Core services (logging, etc)
- Cloud Automation Manager
- Content catalog

Bundled IBM Software
- DB2 Warehouse Enterprise Edition
- Data Science Experience Enterprise Edition

The Architect's view - Bringing it all together at the Enterprise level



Cloud Native application logic (Innovating)

- Microservices

Middleware Services (some IBM and some from the open source world)

- Each instance supports 1..n microservices
- IIB, DB2, Open Databases like Redis, Mongo, Messaging, API C, Datapower

Cloud Enabled (Modernizing)

- Monolithic applications made to run in containers
- Brought from WAS ND or WAS Base to Liberty

Add new function, expose APIs

VM Based Workloads

- WAS Base, WAS ND, BPM and others as necessary, CAM provisioned

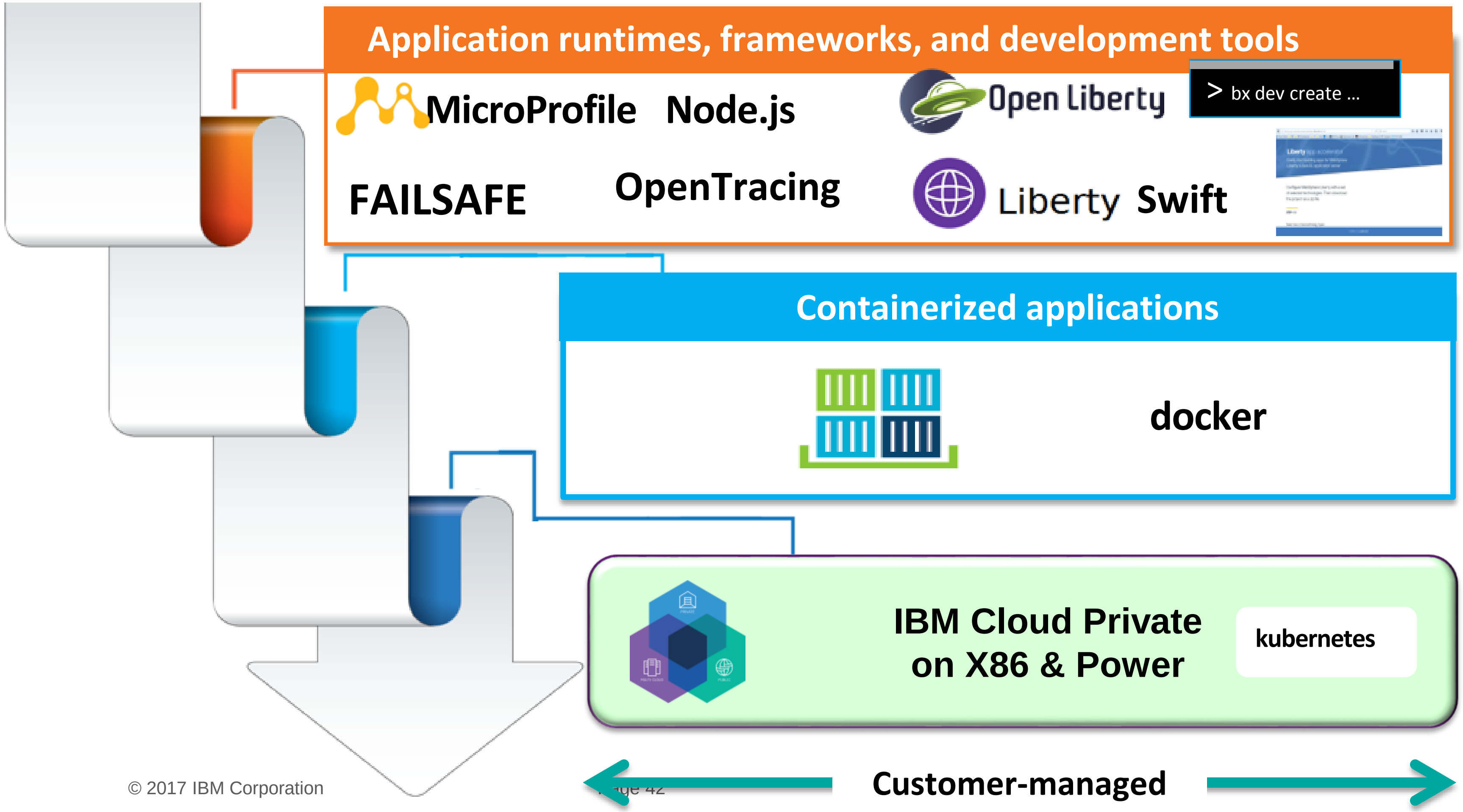
All holistically plugged into existing Enterprise Facilities for Management, Monitoring and Security

Microservices Builder

Microservice acceleration with Microservice Builder

Accelerate the creation and deployment of microservice, hybrid, and containerized applications, targeting Kubernetes-based Docker clouds like IBM Cloud Private

“The Microservice Builder pipeline runs on a Jenkins image in a Docker container that is deployed to Kubernetes using Helm. It is designed to integrate with GitHub, GitHub Enterprise, or other Git services that are supported by the Jenkins GitHub plug-ins”



Application runtimes, frameworks, and development tools

```
> bx dev create ...
```

Containerized applications

IBM Cloud Private on X86 & Power

Customer-managed

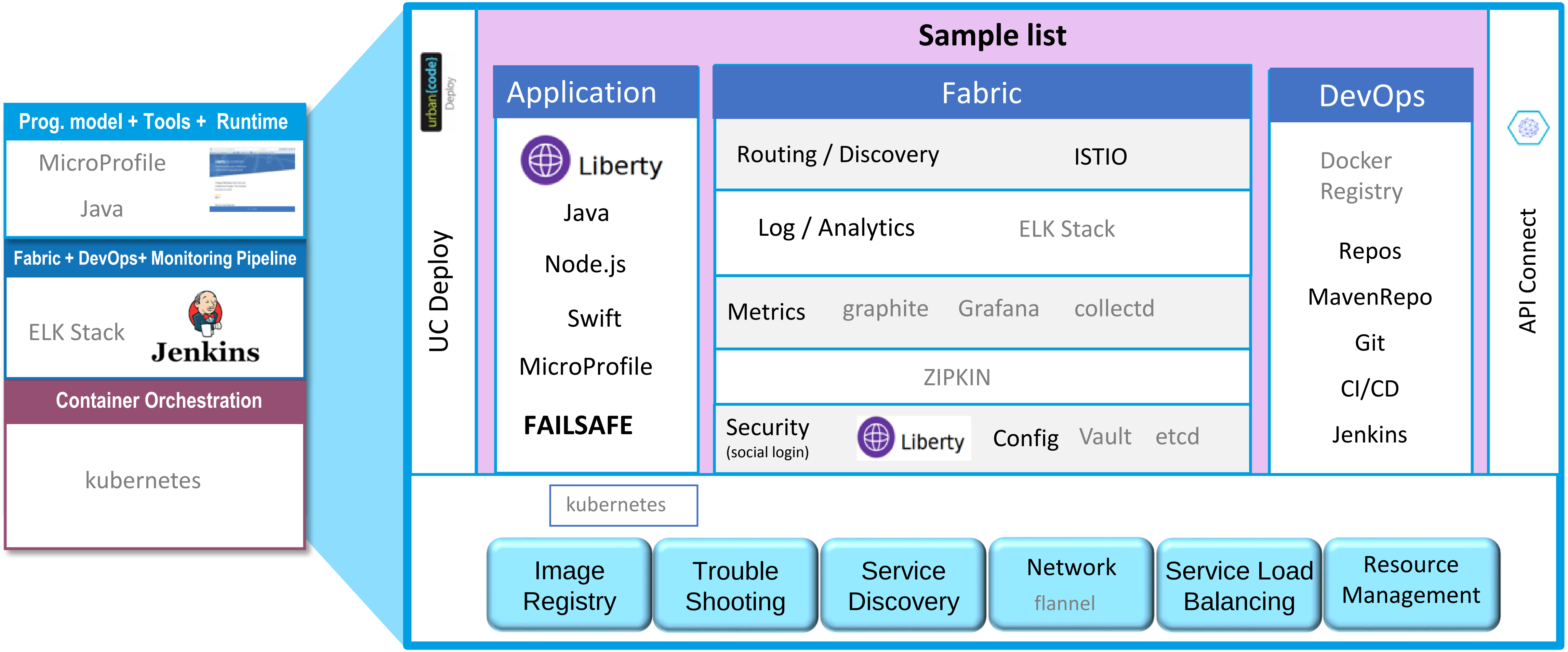
Integrated Resilience

Application and Infrastructure working together

- Pre-integrated Fabric, DevOps tools, build automation, end-to-end security on IBM Cloud Private
- WebSphere Liberty provides MicroProfile technologies to simplify and encapsulate portable behavior for fault tolerance, health check and metrics endpoints
- Advanced resiliency with integrated ISTIO supporting Canary testing

Microservice Builder: A comprehensive environment

Turnkey solution delivers Runtime + Tools + Open Source + DevOps + Fabric + choice of Container Orchestration – on-prem or off-prem



Prog. model + Tools + Runtime

MicroProfile

Java

Fabric + DevOps+ Monitoring Pipeline

ELK Stack

Jenkins

Container Orchestration

kubernetes

UC Deploy

urban{code} Deploy

Sample list

Application

Liberty

Java

Node.js

Swift

MicroProfile

FAILSAFE

Fabric

Routing / Discovery — ISTIO

Log / Analytics — ELK Stack

Metrics — graphite — Grafana — collectd

ZIPKIN

Security (social login) — Liberty — Config — Vault — etcd

DevOps

Docker Registry

Repos

MavenRepo

Git

CI/CD

Jenkins

API Connect

kubernetes

Image Registry

Trouble Shooting

Service Discovery

Network flannel

Service Load Balancing

Resource Management

Enterprise ready platform with IBM support!

Microservice Builder: Gets you ahead of the revolution

Focus on app development, not the framework

Developer Center

Get ahead of the

REVOLUTION

Get started now

Get ahead of the microservices revolution.
Focus on app development, not the framework.

In 3 steps

Create and run your microservices, hybrid and containerized apps

1 Download and install

2 Create and code

3 Build and run

Your team is looking for ways to shorten **development cycles, reduce complexity** and **minimize costs**. Microservice Builder can benefit your team in three ways:

Agile dev and test
Support rapid hybrid and cloud-native application development and testing cycles with greater agility, scalability and security

Lightning fast delivery
Leverage a continuous delivery pipeline to accelerate software delivery from weeks to days, and days to minutes

Cloud agnostic deploy
Reduce costs and complexity with portability across IBM and other cloud providers including public, private and hybrid clouds

Innovate with

SPEED

Set up your environment, fabric and DevOps pipeline in Minutes with guidance through a dedicated:

- developerWorks Developer Center
- Knowledge Center documentation
- Slack community

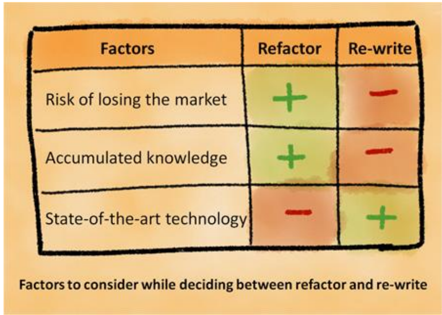
IBM support entitlement with WebSphere Application Server

https://developer.ibm.com/microservice-builder

Et mes applications IBM i ?

Mes Applications IBM i & les applications Cloud

- ICP = solution basée sur des standards Open pour créer et déployer rapidement mes applications Cloud Native & Micro-service.
- Il faut complémenter cette solution avec des processus et outils “DevOps” (automatisation, test..)
  - CI/CD : Microservice Builder, …
- Coté IBM i, quelques Challenges :
  1. Comprendre son applicatif
  2. Moderniser ses applications, appliquer les bonnes pratiques pour les nouvelles
     - Rewriting vs. Refactoring
     - Data Centric Approach – Utiliser les fonctions Db2 for i & OS,…
     - Modular Design vs. Architectural Monolith
     - Approche **DevOps** – Toolchain CI/CD souhaitable (obligatoire?) prenant en compte **l’environnement IBM i.**
  3. Intégration IBM i avec les Cloud Native Apps
     - Puissance des outils & frameworks Open Source sur IBM i.
     - Exposition et valorisation des applications & données IBM i via des standards - Web Services & API / Microservices sur IBM i



Factors	Refactor	Re-write
Risk of losing the market	+	–
Accumulated knowledge	+	–
State-of-the-art technology	–	+

Factors to consider while deciding between refactor and re-write

Mes Applications IBM i & les applications Cloud

- ❑ Pourquoi une approche modulaire, voire micro services sur IBM i ?
  - ❑ Travail collaboratif et intégration applicative facilitée, notamment dans le cas d'applications complexes.
  - ❑ Gain de temps dans des projets innovants avec des changements applicatifs et améliorations fréquents.
  - ❑ Valable sur des développements purement IBM i (RPG ILE, Java, Node.js, PHP) ou hybride (IBM i + Cloud – SoE – SoR).

- ❑ S'inscrit dans une méthodologie **DevOps , pour les applications Cloud (SoE) mais également IBM i**

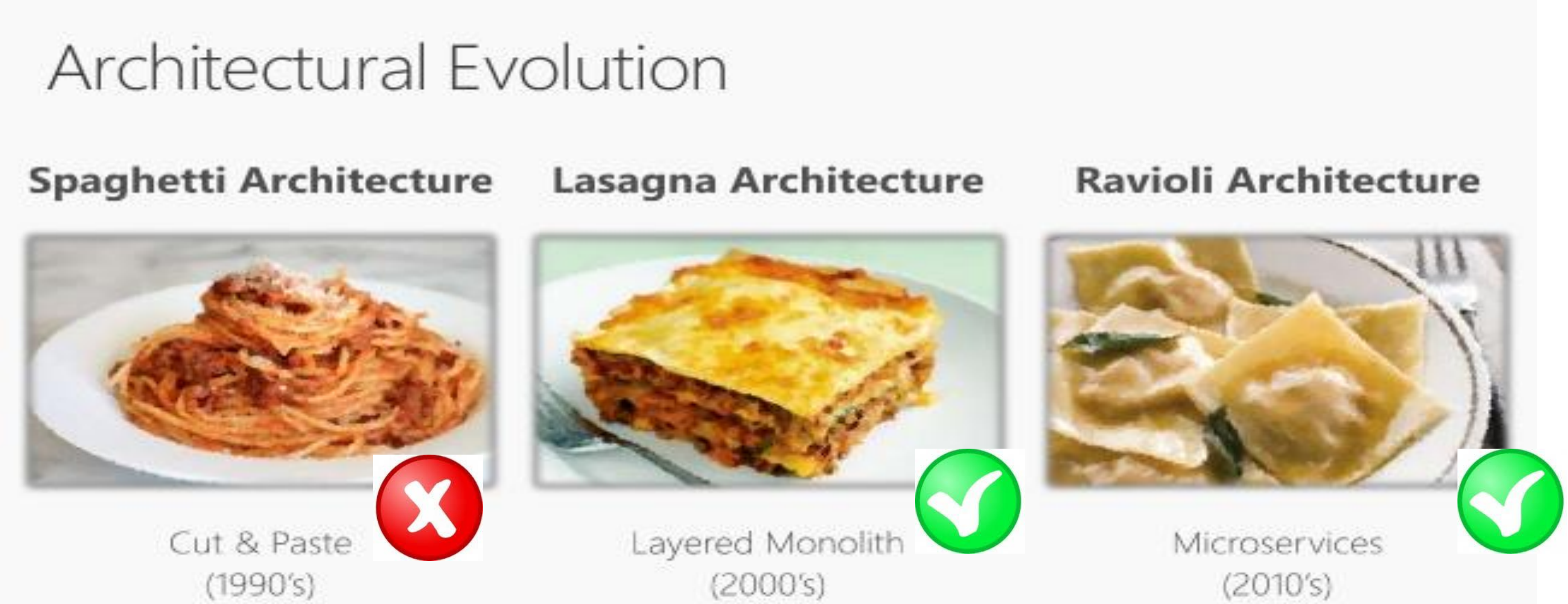
  → nécessite l'esprit DevOps…et les outils adéquats : **Toolchain, Delivery Pipeline**

  → Solutions IBM, Open Source et Editeurs

- ❑ Voir les sessions sur le thème « **IBM i & DevOps** » - Université IBM i 2018:

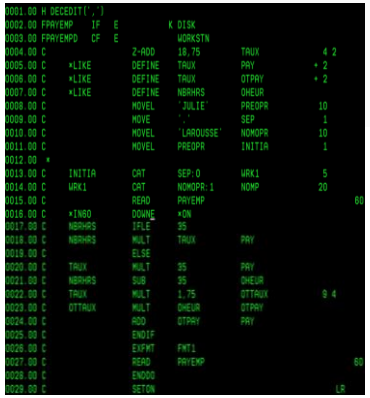
ARCAD SOFTWARE — IBM — itheis — Rocket — SMA solutions BEYOND AUTOMATION™

Mes Applications IBM i & les applications Cloud



Architectural Evolution

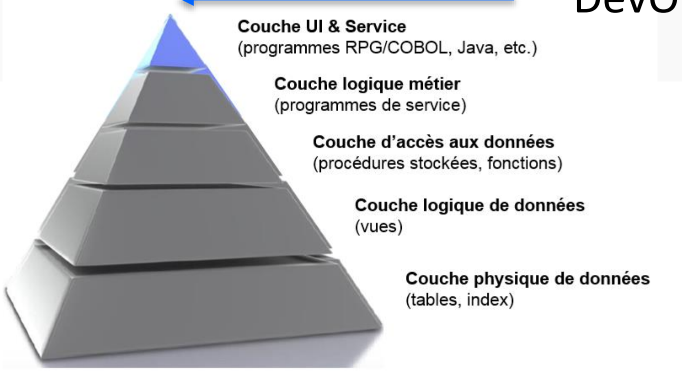
Spaghetti Architecture	Lasagna Architecture	Ravioli Architecture
Cut & Paste (1990's)	Layered Monolith (2000's)	Microservices (2010's)



```
0001.00 H DECEDIT(',')
0002.00 FPAYEMP    IF   E           K DISK
0003.00 FPAYEMPD   CF   E             WORKSTN
0004.00 C                    Z-ADD     18,75         TAUX            4 2
0005.00 C         *LIKE      DEFINE    TAUX          PAY           + 2
0006.00 C         *LIKE      DEFINE    TAUX          OTPAY         + 2
0007.00 C         *LIKE      DEFINE    NBRHRS        OHEUR
0008.00 C                    MOVEL     'JULIE'       PREOPR         10
0009.00 C                    MOVE      '.'           SEP             1
0010.00 C                    MOVEL     'LAROUSSE'    NOMOPR         10
0011.00 C                    MOVEL     PREOPR        INITIA          1
0012.00  *
0013.00 C         INITIA     CAT       SEP:0         WRK1            5
0014.00 C         WRK1       CAT       NOMOPR:1      NOMP           20
0015.00 C                    READ      PAYEMP                             60
0016.00 C         *IN60      DOWNE     *ON
0017.00 C         NBRHRS     IFLE      35
0018.00 C         NBRHRS     MULT      TAUX          PAY
0019.00 C                    ELSE
0020.00 C         TAUX       MULT      35            PAY
0021.00 C         NBRHRS     SUB       35            OHEUR
0022.00 C         TAUX       MULT      1,75          OTTAUX          9 4
0023.00 C         OTTAUX     MULT      OHEUR         OTPAY
0024.00 C                    ADD       OTPAY         PAY
0025.00 C                    ENDIF
0026.00 C                    EXFMT     FMT1
0027.00 C                    READ      PAYEMP                             60
0028.00 C                    ENDDO
0029.00 C                    SETON                                      LR
```

App Centric Monolith,
Single Program

DevOps Ready



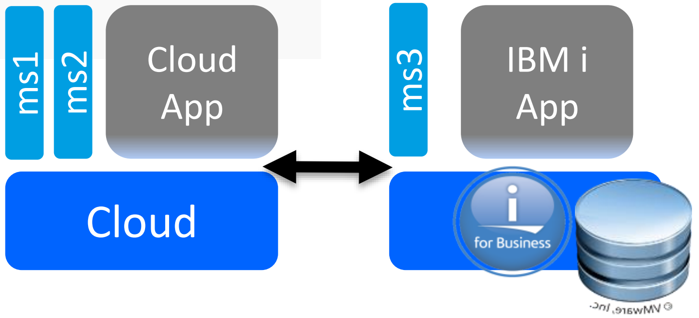
Couche UI & Service
(programmes RPG/COBOL, Java, etc.)

Couche logique métier
(programmes de service)

Couche d'accès aux données
(procédures stockées, fonctions)

Couche logique de données
(vues)

Couche physique de données
(tables, index)



Data Centric, Modular & Layered, Modern Techno, Design Patterns (MVC…)

Mes Applications IBM i & les applications Cloud

Tout est prêt pour créer des solutions Cloud Native & Microservices intégrées à l’environnement IBM i:

- ❑ Nouveaux langages et frameworks sur IBM i : RPG Free, Python, Ruby, Node.js et bien d’autres (.NET) etc.

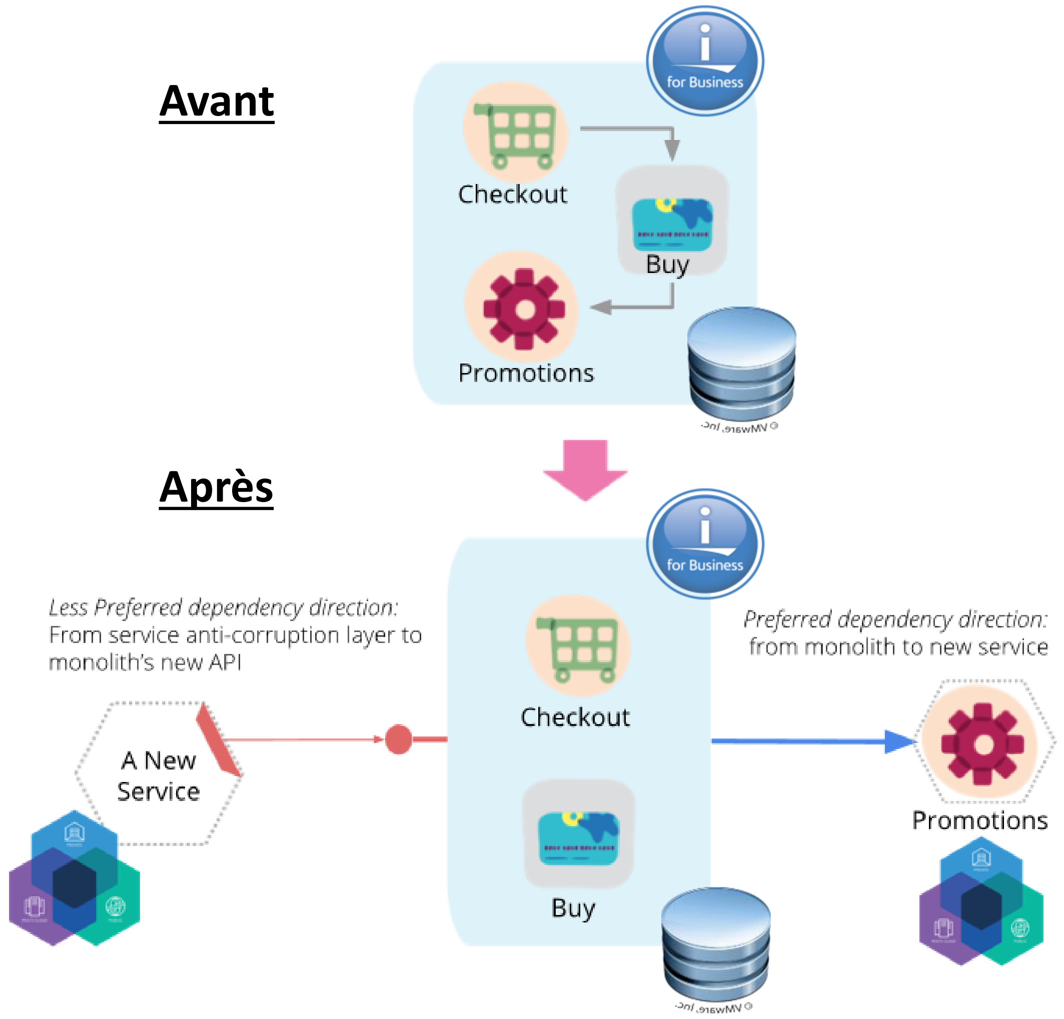
- ❑ Les technologies d’intégration sont disponibles de base sur IBM i
  - Integrated Web Service Server (WebSphere Liberty)
  - Integrated Application Server (WebSphere Liberty)
  - Open Source frameworks (Node.js, NGINX) avec accès natifs aux objets et à la base de données.

Mes Applications IBM i & les applications Cloud

Tout est prêt pour créer des solutions Cloud Native & Microservices intégrées à l’environnement IBM i:

- ❑ Nouveaux langages et frameworks sur IBM i : RPG Free, Python, Ruby, Node.js et bien d’autres (.NET) etc.

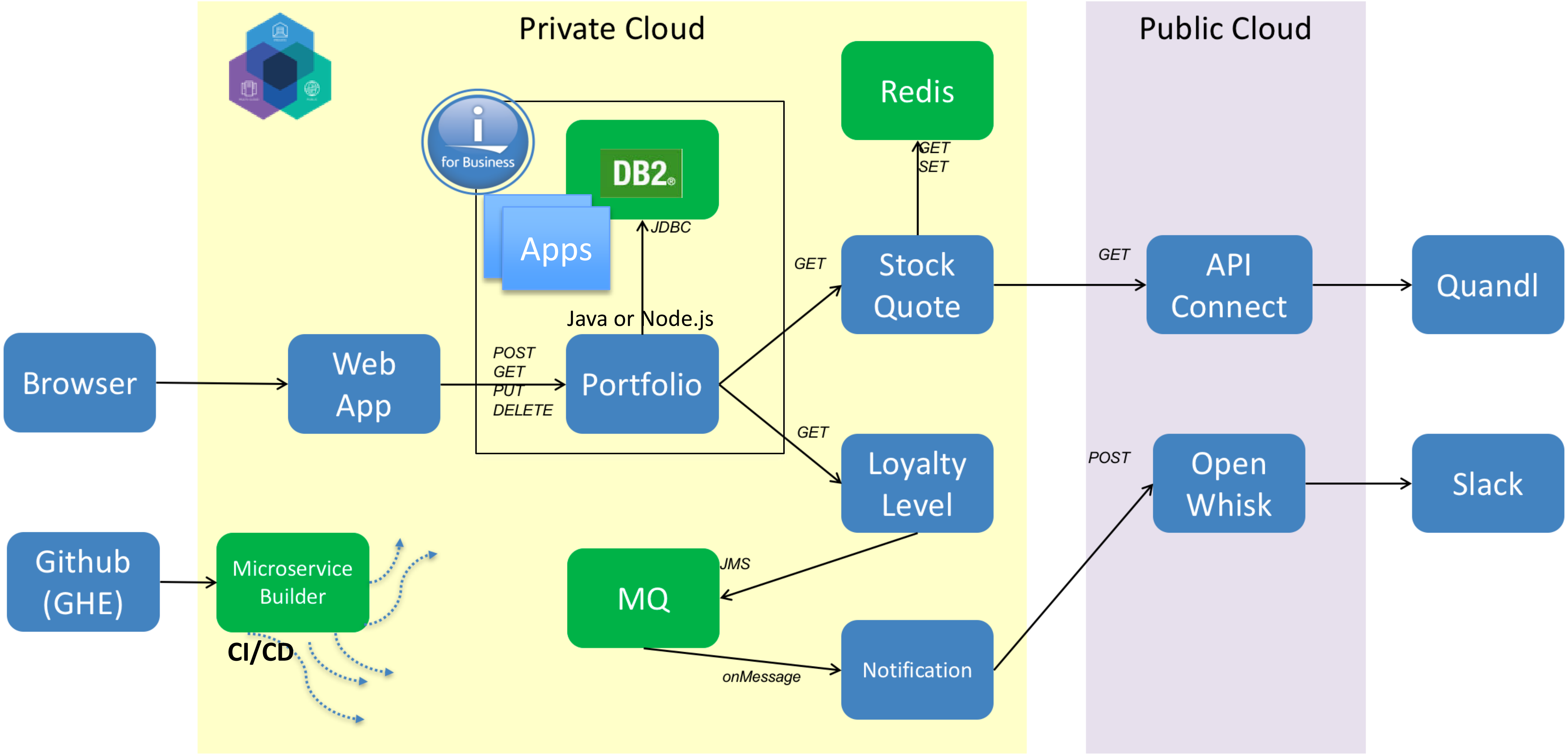
- ❑ Les technologies d’intégration sont disponibles de base sur IBM i
  - Integrated Web Service Server (WebSphere Liberty)
  - Integrated Application Server (WebSphere Liberty)
  - Open Source frameworks (Node.js, NGINX) avec accès natifs aux objets et à la base de données.


Avant
for Business
Checkout
Buy
Promotions
Après
for Business
Less Preferred dependency direction:
From service anti-corruption layer to monolith's new API
A New Service
Checkout
Buy
Preferred dependency direction:
from monolith to new service
Promotions

Mes Applications IBM i & les applications Cloud

Application micro-services avec integration IBM i

Stock Trader App

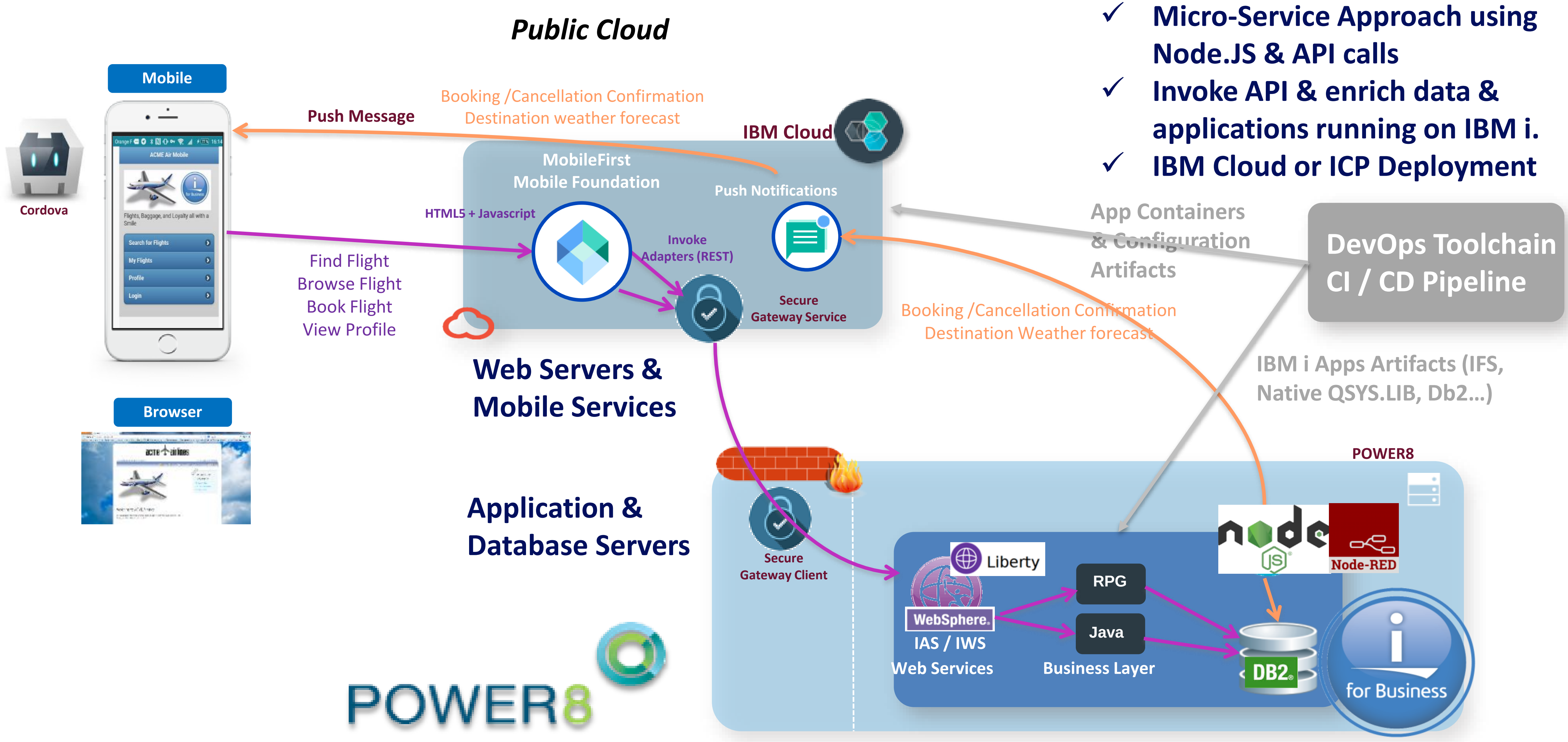


Original source: https://github.com/IBMStockTrader/

Example: Develop new Mobile Services on IBM i with IBM Cloud

Hybrid Application – DevOps & Microservices

- ✓ Micro-Service Approach using Node.JS & API calls
- ✓ Invoke API & enrich data & applications running on IBM i.
- ✓ IBM Cloud or ICP Deployment

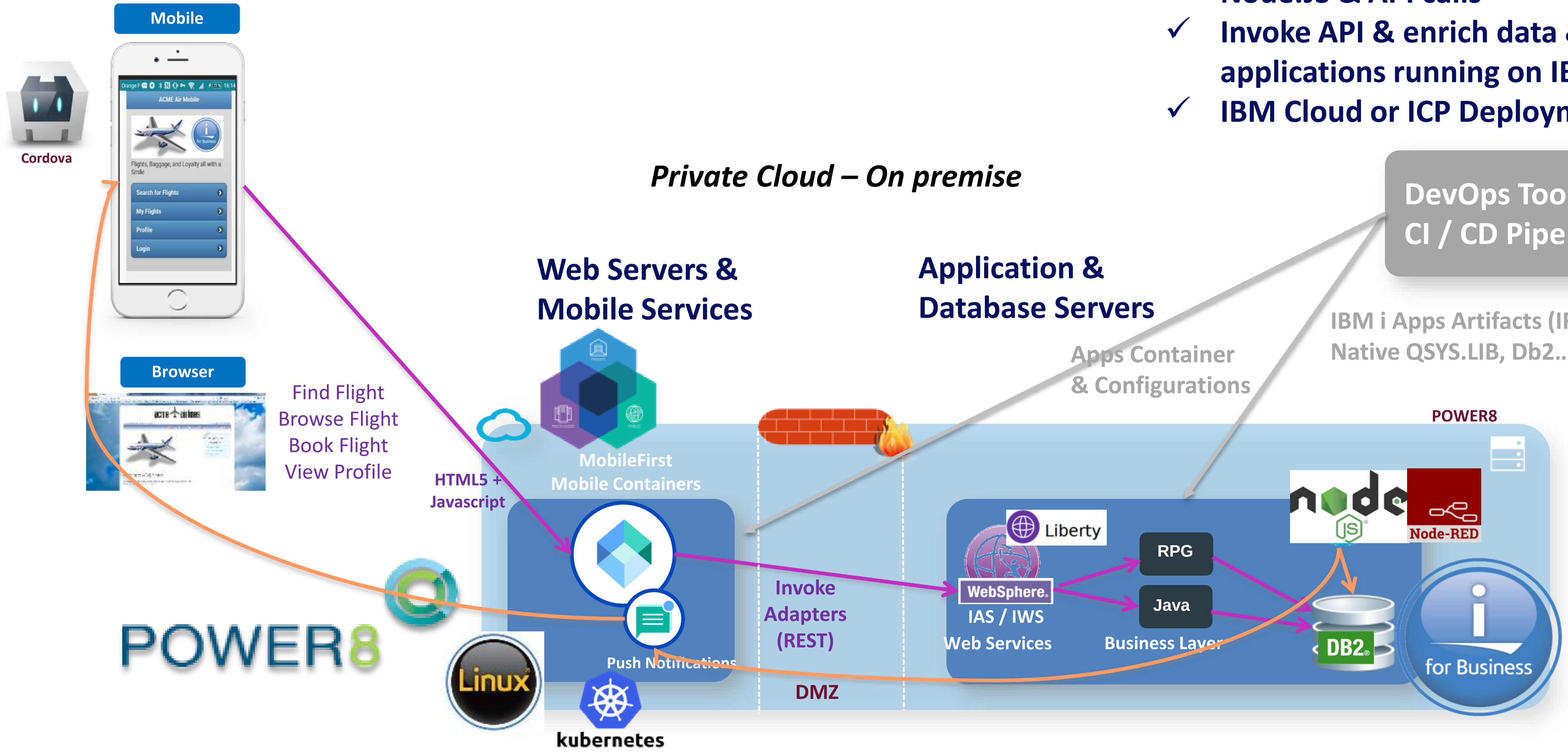


Example: Develop new Mobile Services on IBM i with IBM Cloud

Hybrid Application – DevOps & Microservices

- ✓ Micro-Service Approach using Node.JS & API calls
- ✓ Invoke API & enrich data & applications running on IBM i.
- ✓ IBM Cloud or ICP Deployment

Private Cloud – On premise



Web Servers & Mobile Services

Application & Database Servers

DevOps Toolchain CI / CD Pipeline

Apps Container & Configurations

IBM i Apps Artifacts (IFS, Native QSYS.LIB, Db2…)

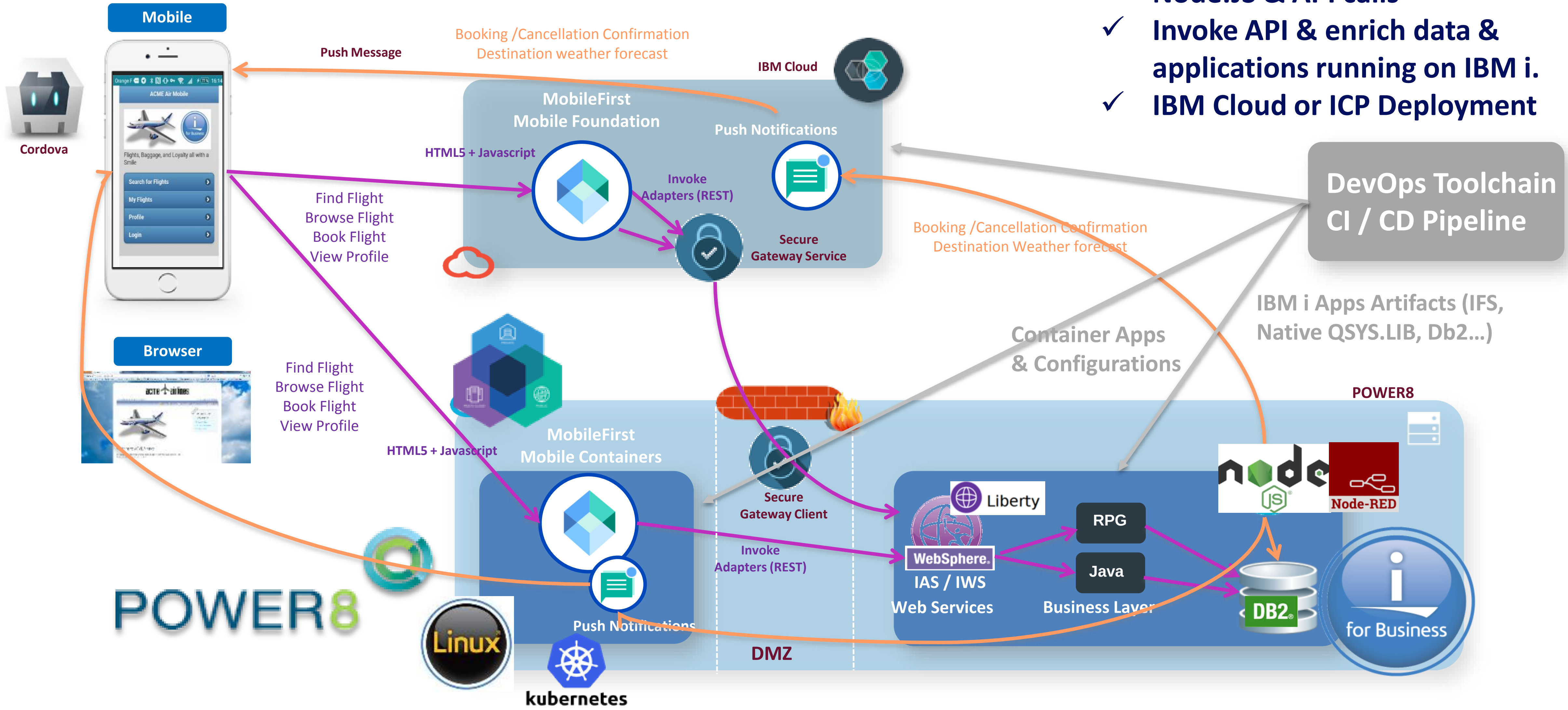
Find Flight
Browse Flight
Book Flight
View Profile

Invoke Adapters (REST)

Example: Develop new Mobile Services on IBM i with IBM Cloud

Hybrid Application – DevOps & Microservices

- ✓ Micro-Service Approach using Node.JS & API calls
- ✓ Invoke API & enrich data & applications running on IBM i.
- ✓ IBM Cloud or ICP Deployment

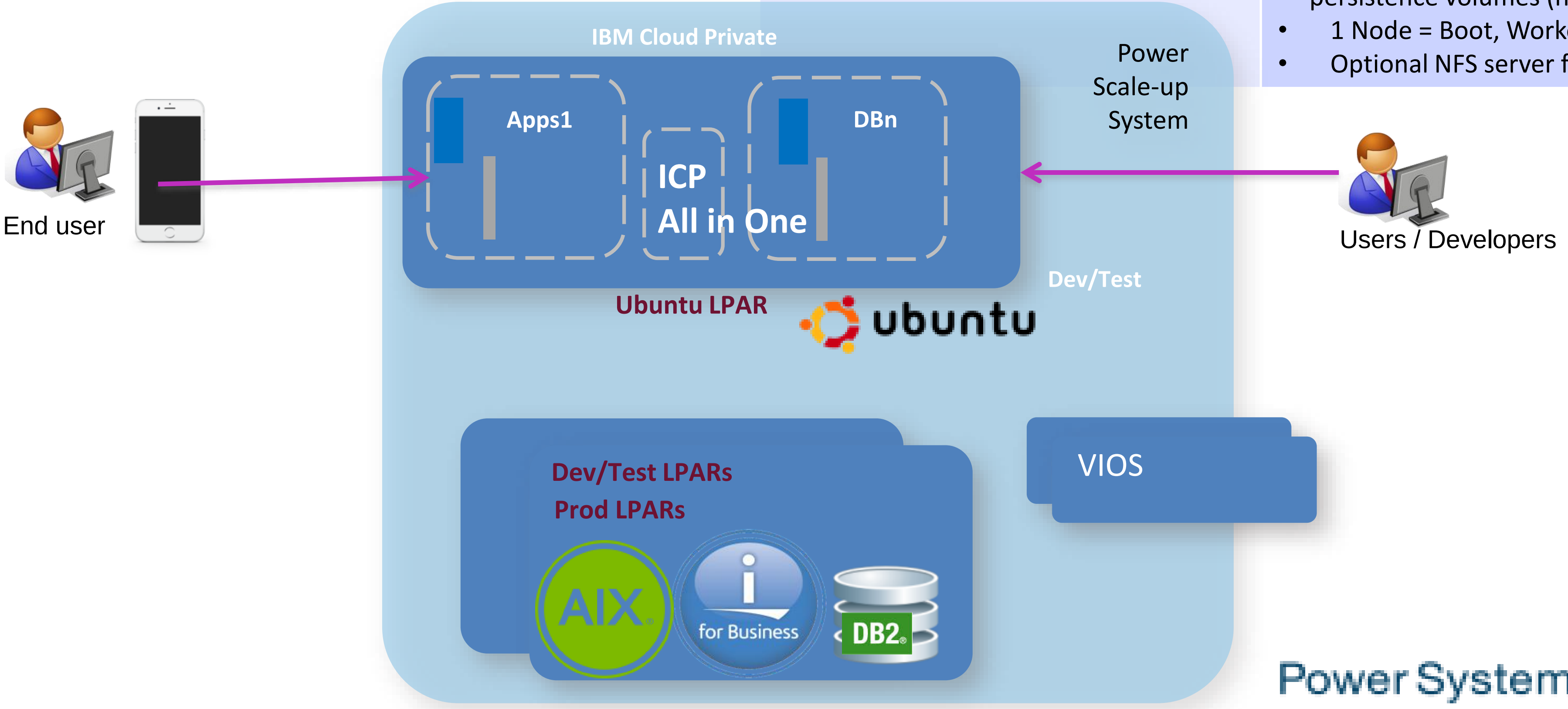


Comment Démarrer?

Starter configuration – Single node “All in one”

- **CE (Free) Edition. For Dev/Test, no HA.**
- Cloud or Enterprise Editions = catalog++
- ICP on PowerVM (LPAR) or 1 Scale-out L / LC / CS System (VM)
- Cloud Foundry on x86 nodes only (1H 2018)

Current	Target*
IBM i / AIX / Linux LPARs production	IBM i / AIX / Linux LPARs production
IBM i / AIX / Linux LPARs dev/test	IBM i / AIX / Linux LPARs dev/test
	• **ICP 2.1 on Power Community Edition** • **1 Node: LPAR Ubuntu 16.04 LTS** • **8 core / 16GB / disk 200+ GB** Documentation: 4 cores / 8GB • SAN Storage or internal disks • hosted by VIOS (SAN/LAN), NFS Server for persistence volumes (need storage) • 1 Node = Boot, Worker, Proxy, Master • Optional NFS server for persistence



Comment continuer?

Multi-VM config with HA – Config Example

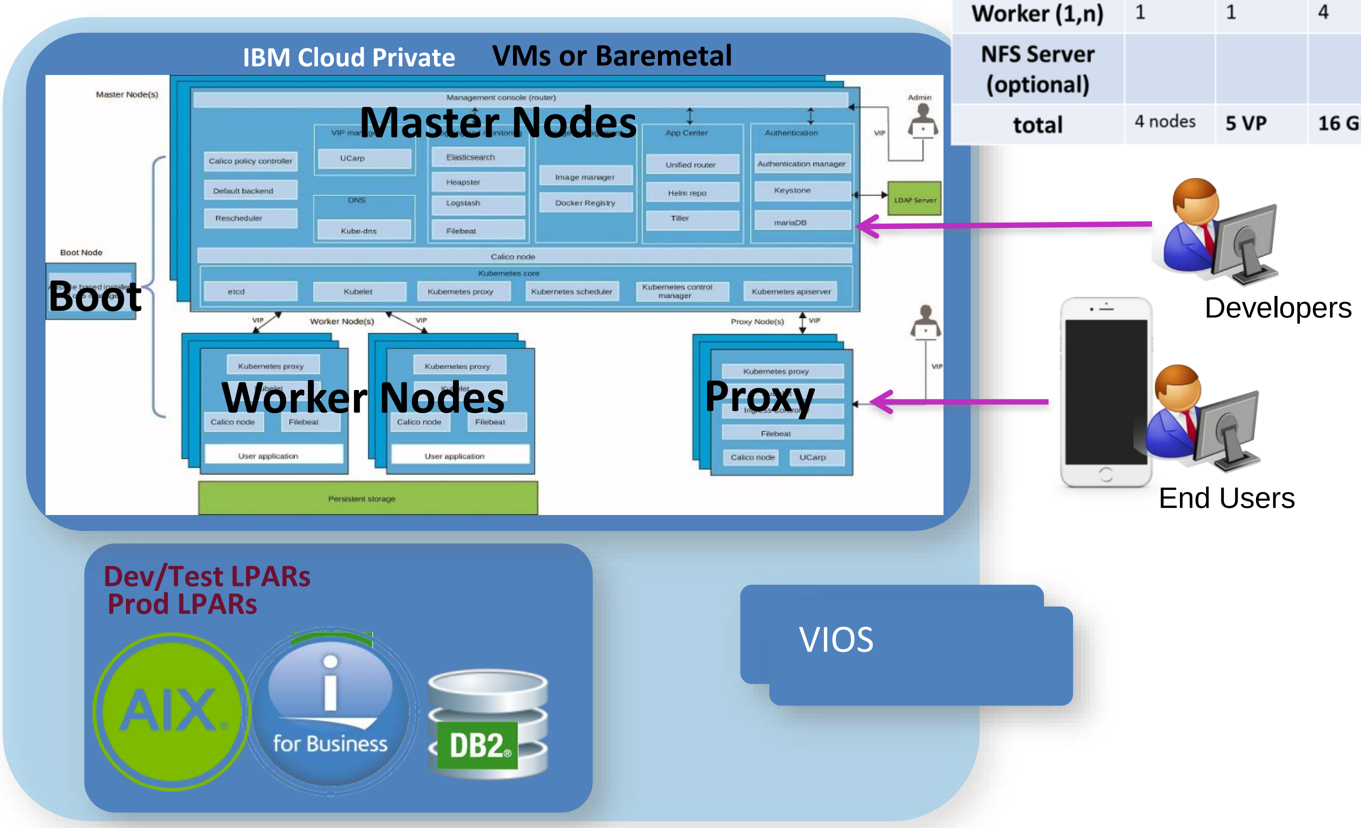
- ICP on PowerVM (LPARs) / Scale-out Systems / Nutanix HCI
- HA Mode for Master & Proxy nodes

Note: Worker nodes can support mixed architectures. You can add worker nodes into a single cluster that run on Linux™ 64-bit, Linux™ on Power® 64-bit LE and IBM® Z platforms.

- Persistence:
  - NFS v4
  - Prefer Glusterfs
- Node = VM (LPAR or KVM VM)
  or Baremetal
- Example on PowerVM:
  - 11 nodes on 11 LPARs
- Example on L / CS / LC servers:
  - 11 nodes on 4 servers

Current	Target*
IBM i / AIX / Linux LPARs production	IBM i / AIX / Linux LPARs production
	ICP 2.1 on Power - Cloud or Enterprise Edition **11 x OS: Ubuntu 16.04 LTS or RHEL 7.2+**

	Minimal 4 Node Config				11 Node Config Example			
Role	#Nodes	CPU (VP)	Mem (GB)	Disk (GB)	#Nodes	CPU (VP)	Mem (GB)	Disk (GB)
Boot (1)	1	1	4	100	1	2	8	80
Master(1,3,5)	1	1	4	150	3	4	16	60
Proxy (1,3,5)	1	2	4	50	3	2	4	60
Management (1, optional)	0	4	8	100	1	8	16	150
Worker (1,n)	1	1	4	100	3	2	8	100
NFS Server (optional)					1	2	8	1024
total	4 nodes	**5 VP**	**16 GB**	**400 GB**	11 nodes	**32 VP**	**116 GB**	**900 GB**



IBM **Client Center** Montpellier

200
EXPERTS
IT Specialists, Architects, Designers, Project Managers

1500
ENGAGEMENTS
Think Industry Showcases
Explore & Co-Create in IBM Studios
Experience - Demo, Benchmark

2500
CLIENTS
From 78 countries

IBM **Client Center** Montpellier

thirty
years

thirty years

OSDB @ Montpellier Cognitive Systems Lab

Positioning IBM Power Systems at the heart of the Cognitive Era

Power System Linux Center

Engage with Clients, ISVs & Partners to leverage the benefits of Linux on Power, using **Open Sources** Solutions

mongoDB

hadoop

HORTONWORKS

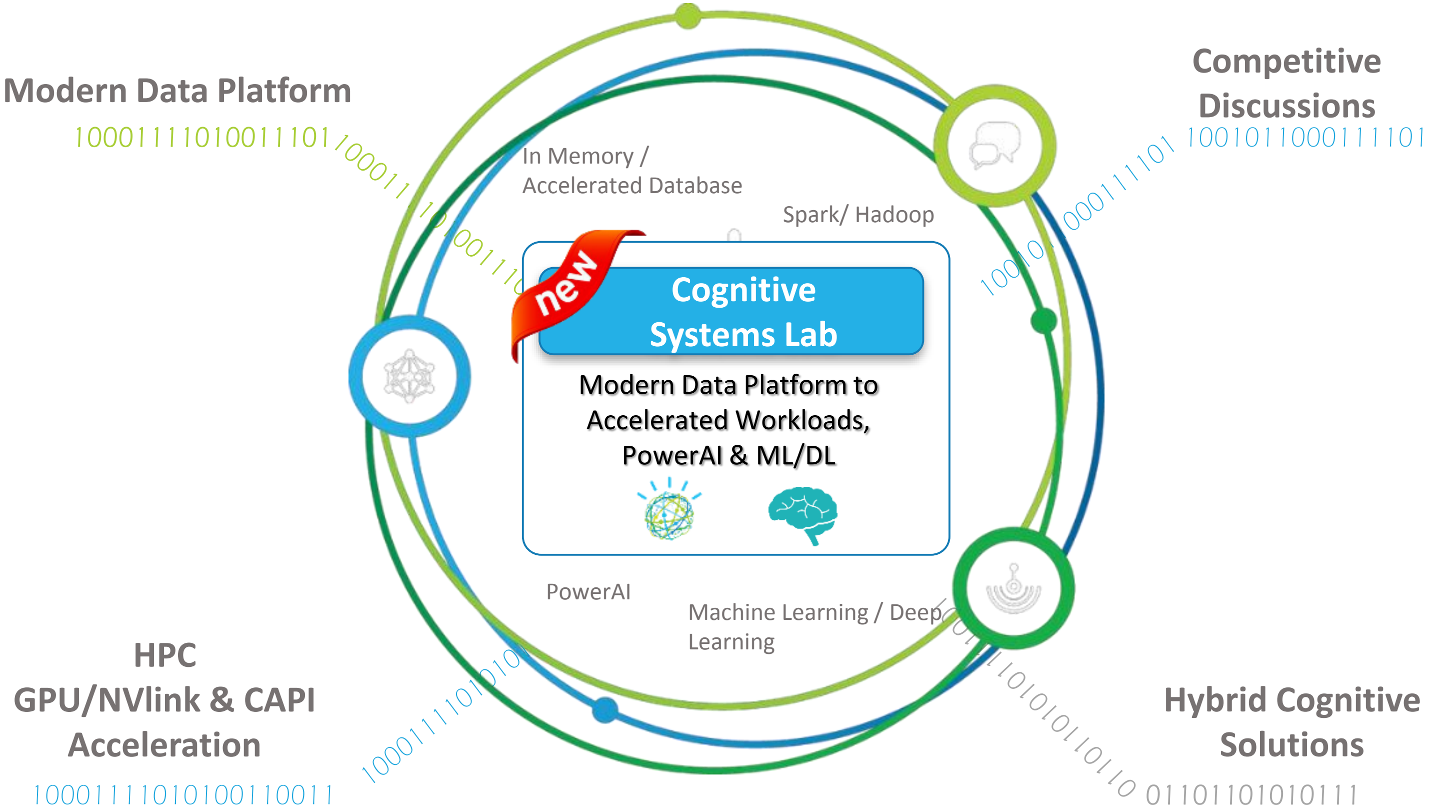
Power Acceleration for High Performance

HPC & HPDA

From traditional HPC on **POWER/GPU** technologies to FPGA based Acceleration with **CAPI/SNAP**

XILINX

NVIDIA



new Cognitive Systems Lab

Modern Data Platform to Accelerated Workloads, PowerAI & ML/DL

Power Acceleration for ISV's

Leverage OpenSource Databases in the Competition with Oracle.

Competitive & TCO Eagle Teams

Leverage TCO Eagle Team Studies & Competitive Expertise (X86/Oracle)

Software Defined Infrastructure

Leverage Software Define Infrastructure to expend Modern Data Platform Capabilities (Spectrum Scale, ESS, LSF...)

What we will deliver

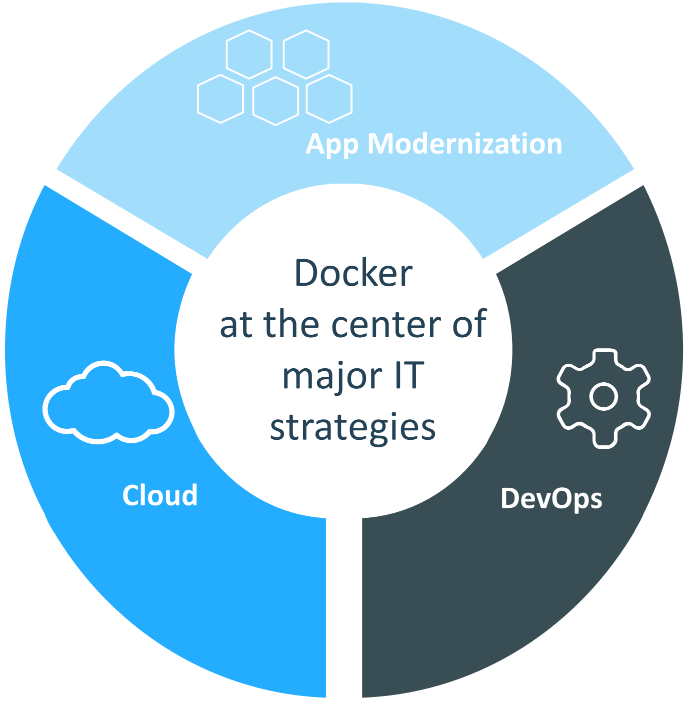
Highly Technical Skills & Hardware (S822LC for HPC & Big Data) available

Talk & Demonstrate	Exploration & Design Workshops	PoT/ POC/Benchmarks	Advanced Technical Support

Backup Slides

- Docker

Docker is the leading software container platform



- Founded in 2013 as Linux developer tool
- Fundamentally solves the “works on my machine” problem
- Container industry inventor, leader and innovator
- Transform app and infrastructure security, portability, agility and efficiency

Docker Enterprise Edition (EE) and Community Edition (CE)

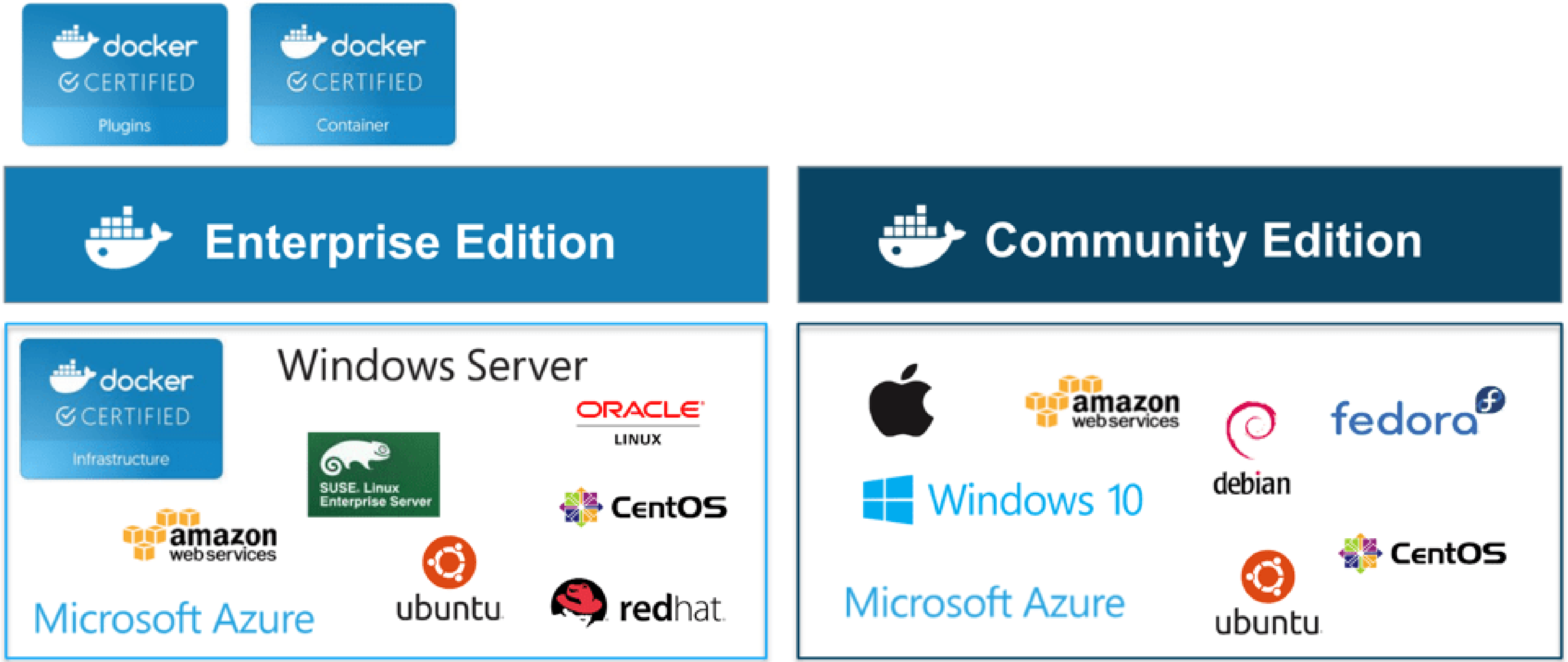
Enterprise Edition (EE)

- CaaS enabled platform subscription (integrated container orchestration, management and security)
- Enterprise class support
- Quarterly releases, supported for one year each with backported patches and hotfixes.
- Certified Infrastructure, Plugins, Containers

Community Edition (CE)

- Free Docker platform for “do it yourself” dev and ops
- Monthly Edge release with latest features for developers
- Quarterly release with maintenance for ops

Docker EE vs CE: Platform Support



Enterprise Edition

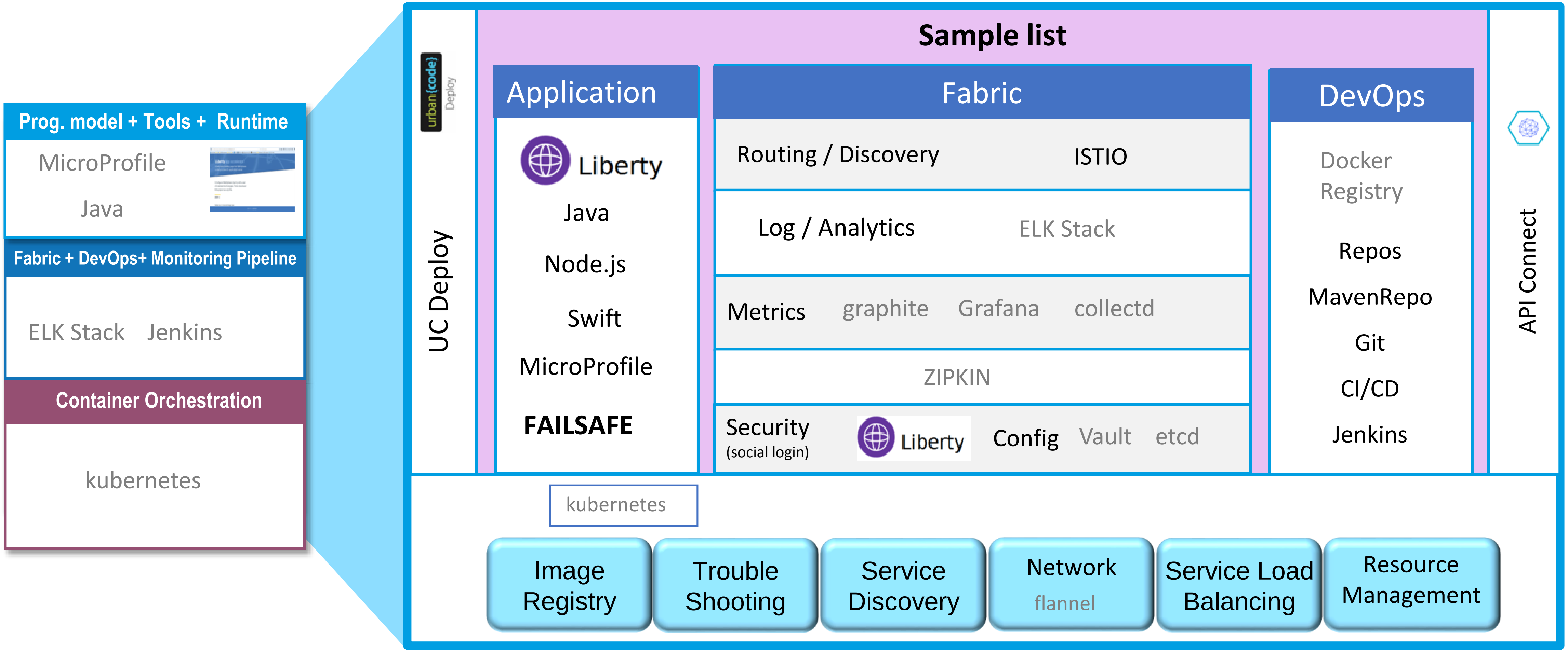
Community Edition

Backup Slides

- Microservice Builder

Microservice Builder: A comprehensive environment

Turnkey solution delivers Runtime + Tools + Open Source + DevOps + Fabric + choice of Container Orchestration – on-prem or off-prem



Prog. model + Tools + Runtime

MicroProfile

Java

Fabric + DevOps+ Monitoring Pipeline

ELK Stack Jenkins

Container Orchestration

kubernetes

urban{code} Deploy

UC Deploy

Sample list

Application

Liberty

Java

Node.js

Swift

MicroProfile

FAILSAFE

Fabric

Routing / Discovery ISTIO

Log / Analytics ELK Stack

Metrics graphite Grafana collectd

ZIPKIN

Security (social login) Liberty Config Vault etcd

DevOps

Docker Registry

Repos

MavenRepo

Git

CI/CD

Jenkins

API Connect

kubernetes

Image Registry

Trouble Shooting

Service Discovery

Network flannel

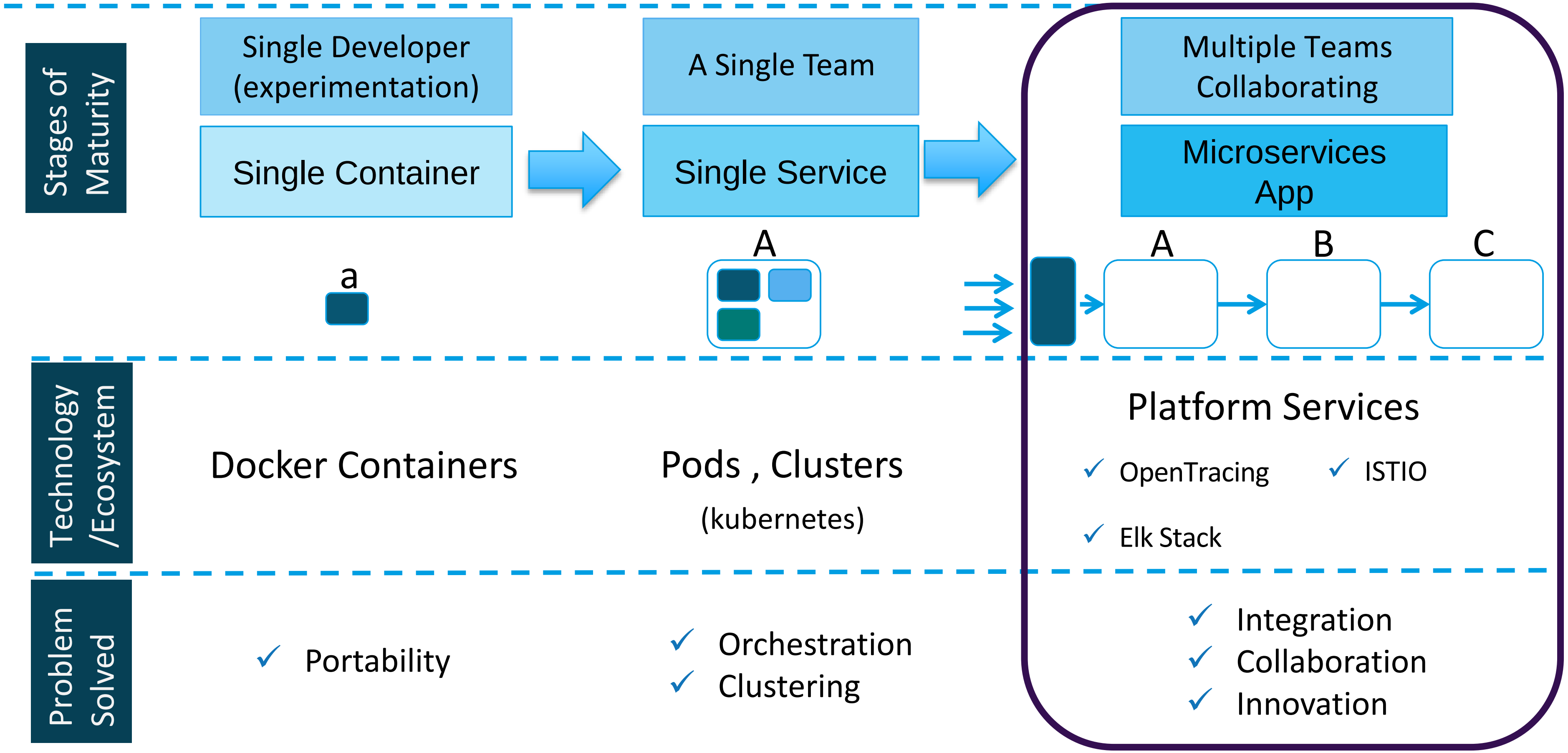
Service Load Balancing

Resource Management

Enterprise ready platform with IBM support!

Microservice Builder: Built for the enterprise journey

Standardizing the technology stack across Dev and Ops



Stages of Maturity	Single Developer (experimentation) Single Container	A Single Team Single Service	Multiple Teams Collaborating Microservices App
Technology /Ecosystem	Docker Containers	Pods , Clusters (kubernetes)	Platform Services ✓ OpenTracing ✓ ISTIO ✓ Elk Stack
Problem Solved	✓ Portability	✓ Orchestration ✓ Clustering	✓ Integration ✓ Collaboration ✓ Innovation

Microservice Builder: The support advantage

Overview of support policy for open source technologies bundled with Microservice Builder

Customers can raise PMRs/Tickets per the normal process

1. IBM support determines if the problem is reported in a documented scenario from the Knowledge Center and is associated with the integration of OSS components with WebSphere Liberty:
   - IBM dev team determines if the problem is in the IBM code integrating with the OSS, not OSS fixes.
   - IBM provides guidance and fix in the integration code

2. IBM determines that a problem is in the open source or is an undocumented scenario
   - IBM will hand off customers to external forums for any products that are not already supported by an IBM team

3. IBM and customer will subscribe, track and monitor open source community for fix or upgrade

4. When a fix becomes available through the open source community:
   - IBM tests and delivers the fix in the next Microservice Builder Update

(ELK) Elastic, Log Analytics, Kibana

Jenkins

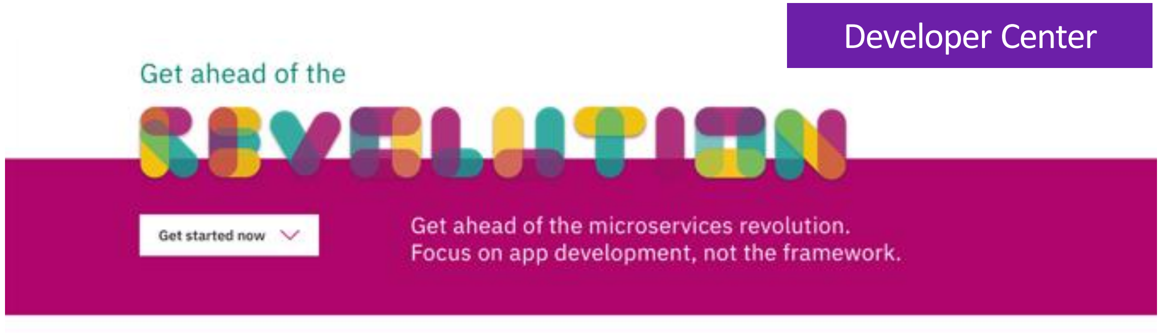
ZIPKIN

GIT

Microservice Builder Knowledge Center: http://ibm.biz/microservicebuilderdoc

Microservice Builder: Gets you ahead of the revolution

Focus on app development, not the framework



Developer Center

Get ahead of the

REVOLUTION

Get started now

Get ahead of the microservices revolution.
Focus on app development, not the framework.

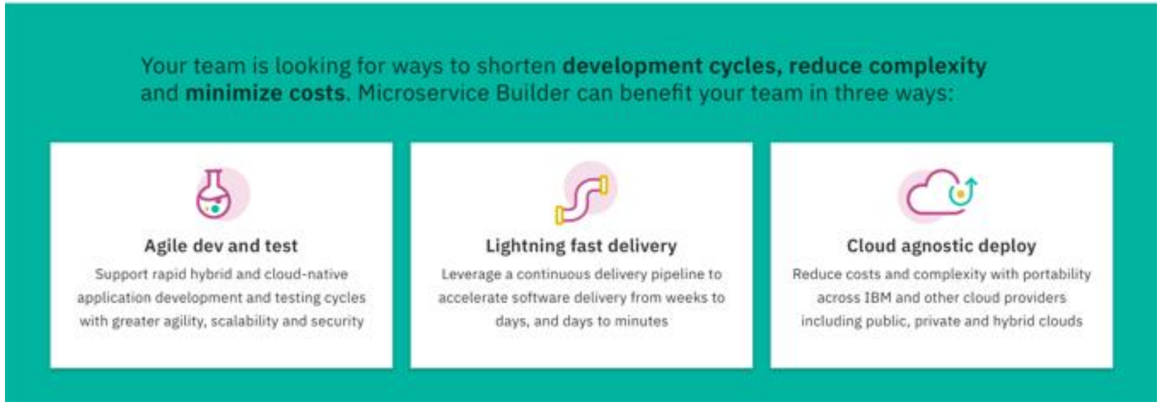
In 3 steps

Create and run your microservices, hybrid and containerized apps

1 Download and install

2 Create and code

3 Build and run



Your team is looking for ways to shorten **development cycles, reduce complexity** and **minimize costs**. Microservice Builder can benefit your team in three ways:

Agile dev and test
Support rapid hybrid and cloud-native application development and testing cycles with greater agility, scalability and security

Lightning fast delivery
Leverage a continuous delivery pipeline to accelerate software delivery from weeks to days, and days to minutes

Cloud agnostic deploy
Reduce costs and complexity with portability across IBM and other cloud providers including public, private and hybrid clouds

Innovate with

SPEED

Set up your environment, fabric and DevOps pipeline in Minutes with guidance through a dedicated:

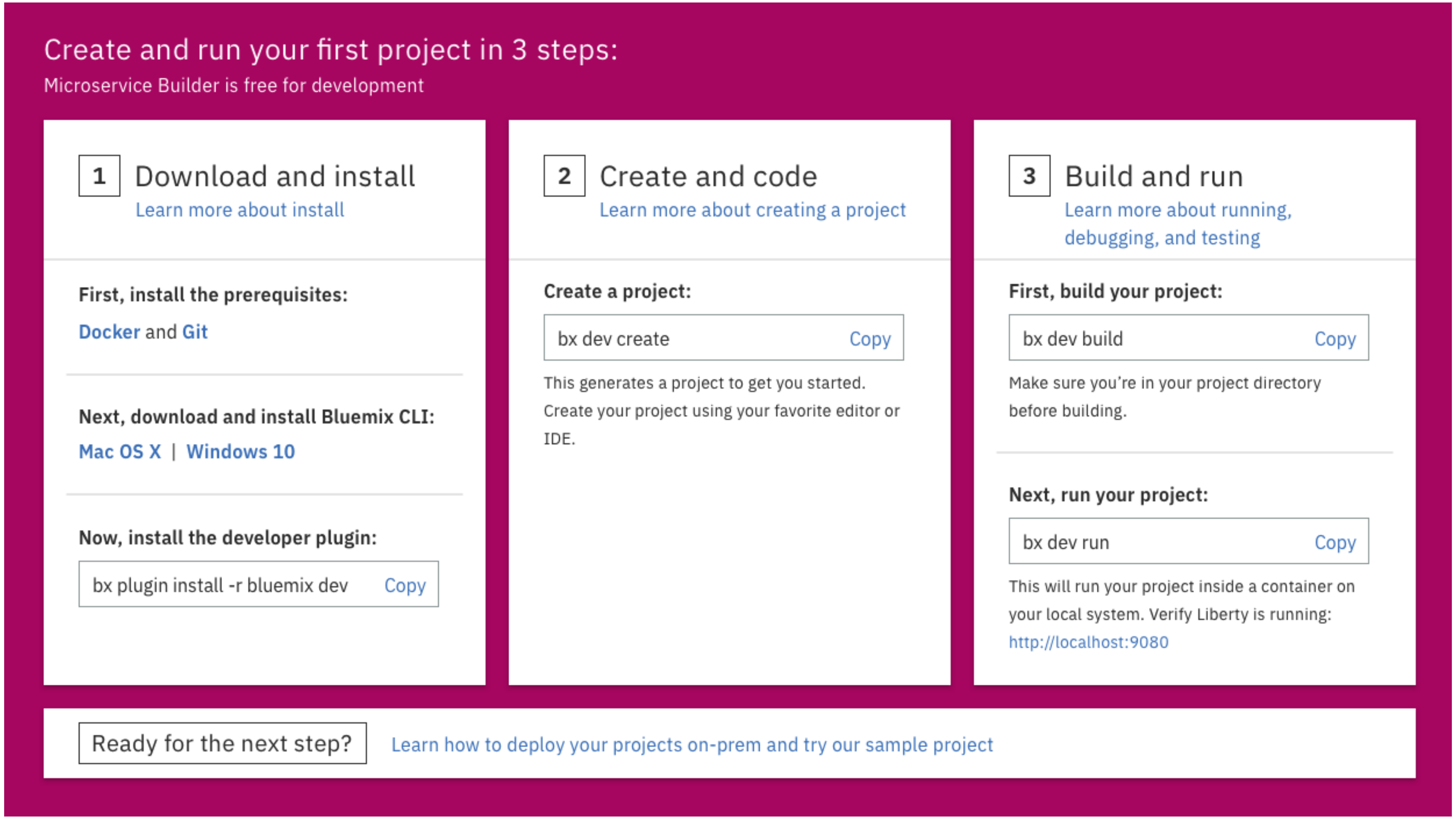
- developerWorks Developer Center
- Knowledge Center documentation
- Slack community

IBM support entitlement with WebSphere Application Server

https://developer.ibm.com/microservice-builder

Developer CLI

Quickly create, build and run new services with minimal pre-reqs



Create and run your first project in 3 steps:

Microservice Builder is free for development

1 Download and install

Learn more about install

First, install the prerequisites:

Docker and Git

Next, download and install Bluemix CLI:

Mac OS X | Windows 10

Now, install the developer plugin:

```
bx plugin install -r bluemix dev
```

2 Create and code

Learn more about creating a project

Create a project:

```
bx dev create
```

This generates a project to get you started. Create your project using your favorite editor or IDE.

3 Build and run

Learn more about running, debugging, and testing

First, build your project:

```
bx dev build
```

Make sure you're in your project directory before building.

Next, run your project:

```
bx dev run
```

This will run your project inside a container on your local system. Verify Liberty is running: http://localhost:9080

Ready for the next step? Learn how to deploy your projects on-prem and try our sample project

"create" walks developer through choice of a pattern (e.g. microservice) and language (Java, Node or Swift) and generates project

"build" uses a containerized set of tools to build the application and package it in a Docker image

"run" executes the Docker image locally

CLI and Docker are the only prerequisites

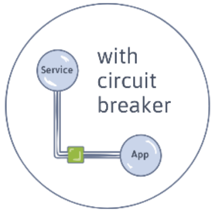
Eclipse | MICROPROFILE™ Microservice innovation

- Vendor-neutral programming model, designed in the open, for Java microservices
- Provide core capabilities for building fault tolerant, scalable, microservices
- Increasing the rate and pace of innovation beyond Java EE

Standardizing microservices in enterprise Java via the MicroProfile community

Config	Fault Tolerance	Health Check	Health Metrics	Open Tracing
externalize configuration to improve portability	build robust behavior to cope with unexpected failures	ensure services are running and meeting SLAs	understand the interactions between services while running	resolve problems in complex distributed systems

Invite your developers to join the MicroProfile community and influence the future http://microprofile.io/



Fault tolerance in action! Learn today using Open Liberty's interactive guides https://openliberty.io/guides/circuit-breaker.html

Microservice Builder entitlement requirements

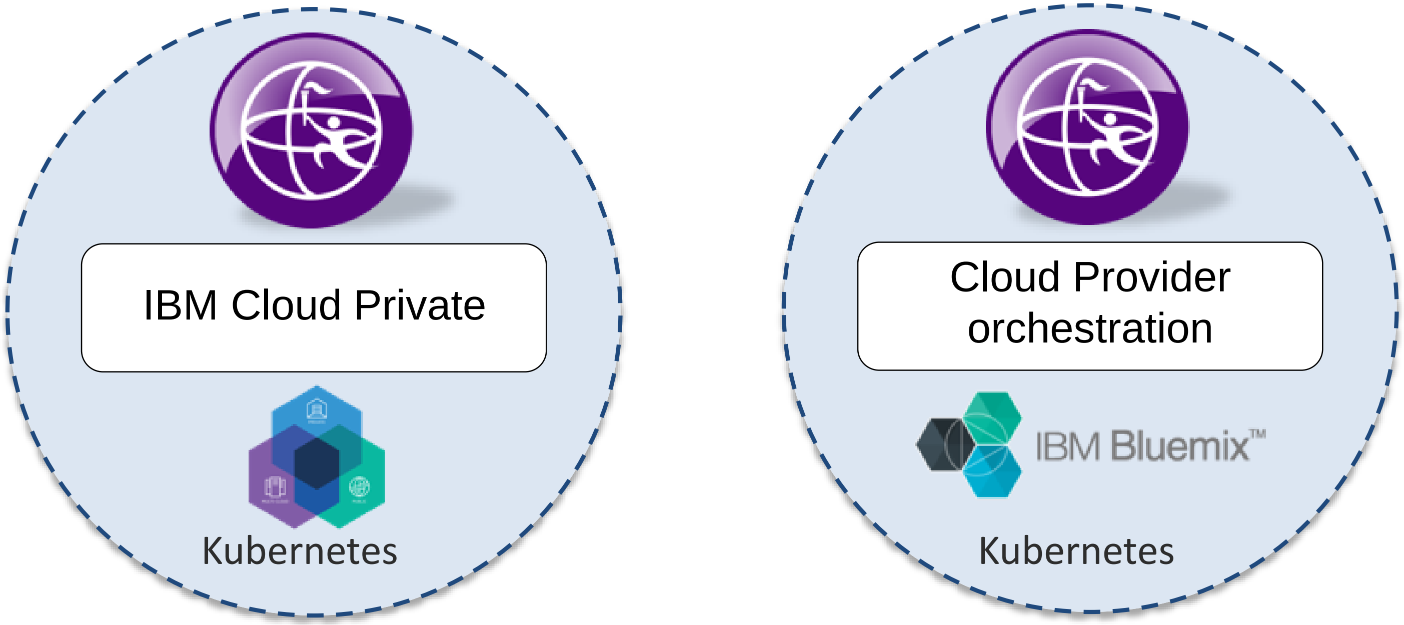
- No charge for Development
- Paid options for test and production:

		End-to-end support entitlement	Liberty Core	WAS aka (Base)	Included in IBM Cloud Private
For BYO-Cloud, purchase Liberty Core and WAS only. Customers bring their own fabric, pipeline, security, Istio, Docker		Microprofile.io programming model, Java runtime	✓	✓	✓
		Oauth security with OpenID Connect security	✓	✓	✓
		Web Profile features	✓	✓	✓
		Java Full platform features (e.g. JMS, data source)		✓	✓
Liberty ND entitlement included in purchase of IBM Cloud Private	Pipeline & Fabric	Integrated Pipeline w Jenkins			✓
		Integrated end-to-end tracing with zipkin and opentracing			✓
		Integrated Elastic search, Log Monitoring			✓
	Istio	Advanced resiliency, canary testing, app editioning, health mgmt, adv scaling policies			✓
		Support for Docker and Docker engine – via IBM Cloud Private			✓

Microservice Builder - Quick guide for buyers

Common capabilities = Programming model, tools, Java runtime, security

✓ **Purchase IBM Cloud Private**



✓ **Purchase WAS Liberty Core**

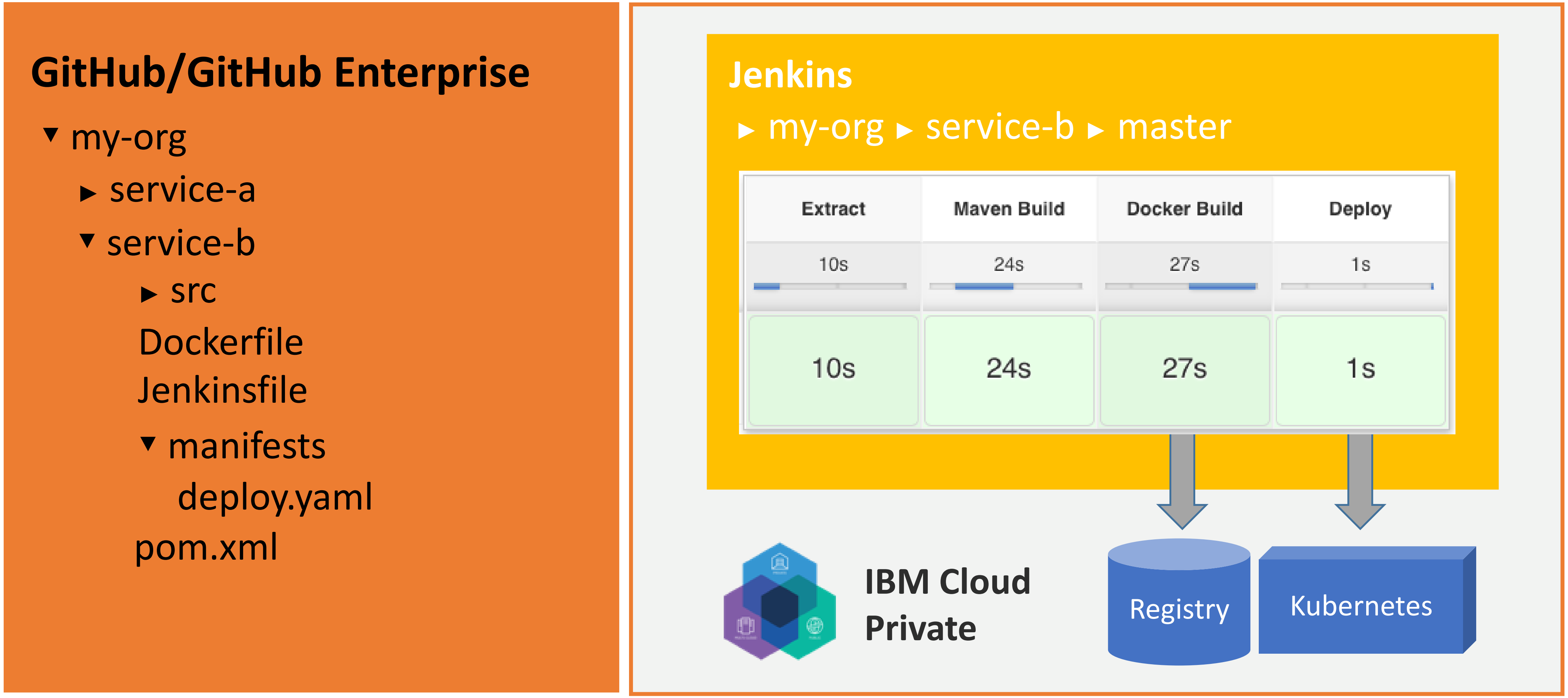
✓ **Purchase WAS**

Included services: *Integrated pipeline, monitoring, log analytics, tracing tools, app editioning, auto-scaling…*

Test and deploy on Bluemix Container Services or other 3rd party clouds (using BYOL for WebSphere Liberty) and purchase equivalent services

Microservice Builder pipeline

Automated build and deployment of microservices for rapid delivery



GitHub/GitHub Enterprise

- ▼ my-org
  - ▶ service-a
  - ▼ service-b
    - ▶ src
    - Dockerfile
    - Jenkinsfile
    - ▼ manifests
      - deploy.yaml
    - pom.xml

Jenkins

▶ my-org ▶ service-b ▶ master

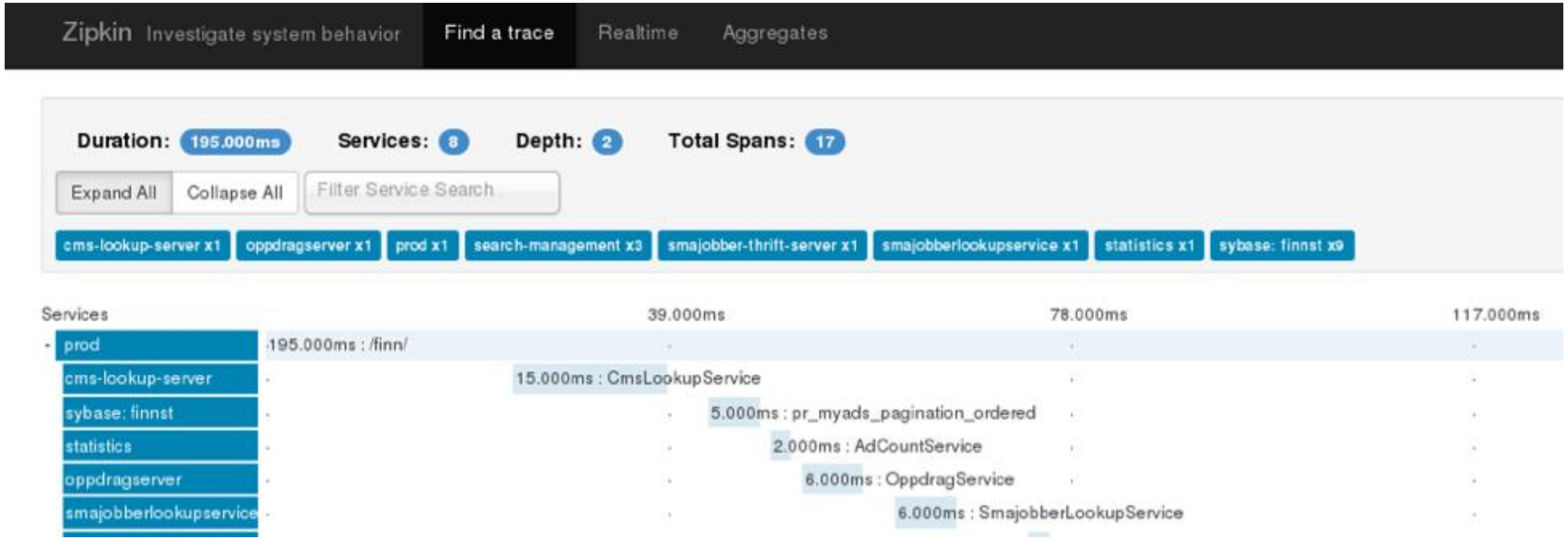
Extract	Maven Build	Docker Build	Deploy
10s	24s	27s	1s
10s	24s	27s	1s

IBM Cloud Private

Microservice Builder fabric

Runtime components to support microservice deployments

- Fabric provides shared Liberty key and trust stores to set up trust for inter-service communication
- Distributed tracing enables tracking of requests as they traverse microservices for problem determination and causal analysis
  - Applications use Open Tracing API to delineate trace spans
  - Fabric provides Zipkin server for collection and visualization of spans



Duration: 195.000ms Services: 8 Depth: 2 Total Spans: 17

cms-lookup-server x1 | oppdragserver x1 | prod x1 | search-management x3 | smajobber-thrift-server x1 | smajobberlookupservice x1 | statistics x1 | sybase: finnst x9

Services	39.000ms	78.000ms	117.000ms
prod	195.000ms : /finn/		
cms-lookup-server	15.000ms : CmsLookupService		
sybase: finnst	5.000ms : pr_myads_pagination_ordered		
statistics	2.000ms : AdCountService		
oppdragserver	6.000ms : OppdragService		
smajobberlookupservice	6.000ms : SmajobberLookupService		

Targeting "Build" and "Run & Manage"

What's New in 3q2017

LEARN

BUILD

RUN & MANAGE

✓ Deploy with confidence across staged environments

Flexible pre-integrated pipeline enables deployments across all your environments while ensuring proper testing prior to service availability

✓ Ease of problem determination

<u>10 steps down to 1!</u>
Developers can instrument distributed tracing easily, allowing admins to easily identify the root cause of a misbehaving service without involving a developer

✓ Secure microservices with interoperable JWT

Easily secure microservices with JWT via configuration or CDI injection in an interoperable and standard way

✓ Sense the health of your microservices

Developers can easily instrument a standard health check URL allowing for a consistent way for admins to configure service monitors

✓ PowerLinux support

Fabric and pipeline now available on PowerLinux, enabling continuous delivery and securing of external Liberty service connections and a pre-integrated Zipkin distributed tracing system

✓ Intelligent management for a microservice world

Canary deployments
Start incorporating live-testing strategies in your workflow through support of canary testing with Istio and IBM Cloud Private

Autoscaling
Proactively manage increases and decreases in demand for your microservice applications using IBM Cloud Private – automatically based on predefined thresholds

Microservice Builder resources

Helpful Links
Marketplace page: https://www.ibm.com/us-en/marketplace/microservice-builder
developerWorks page: https://developer.ibm.com/microservice-builder/
Knowledge Center page: http://ibm.biz/microservicebuilderdoc
Demo Video: http://ibm.biz/BdiCjT
Infographic: https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WS912361USEN&

For more content related information visit our wiki page: https://ibm.biz/Bdjqns

Backup Slides

- ICP

IBM value added to Kubernetes

Core

Intelligent Scheduling

Self-healing

Horizontal scaling

Service discovery & load balancing

Automated rollouts and rollbacks

Secret and configuration management

IBM

Simplified Cluster Management

Container Security & Isolation

Design Your Own Cluster

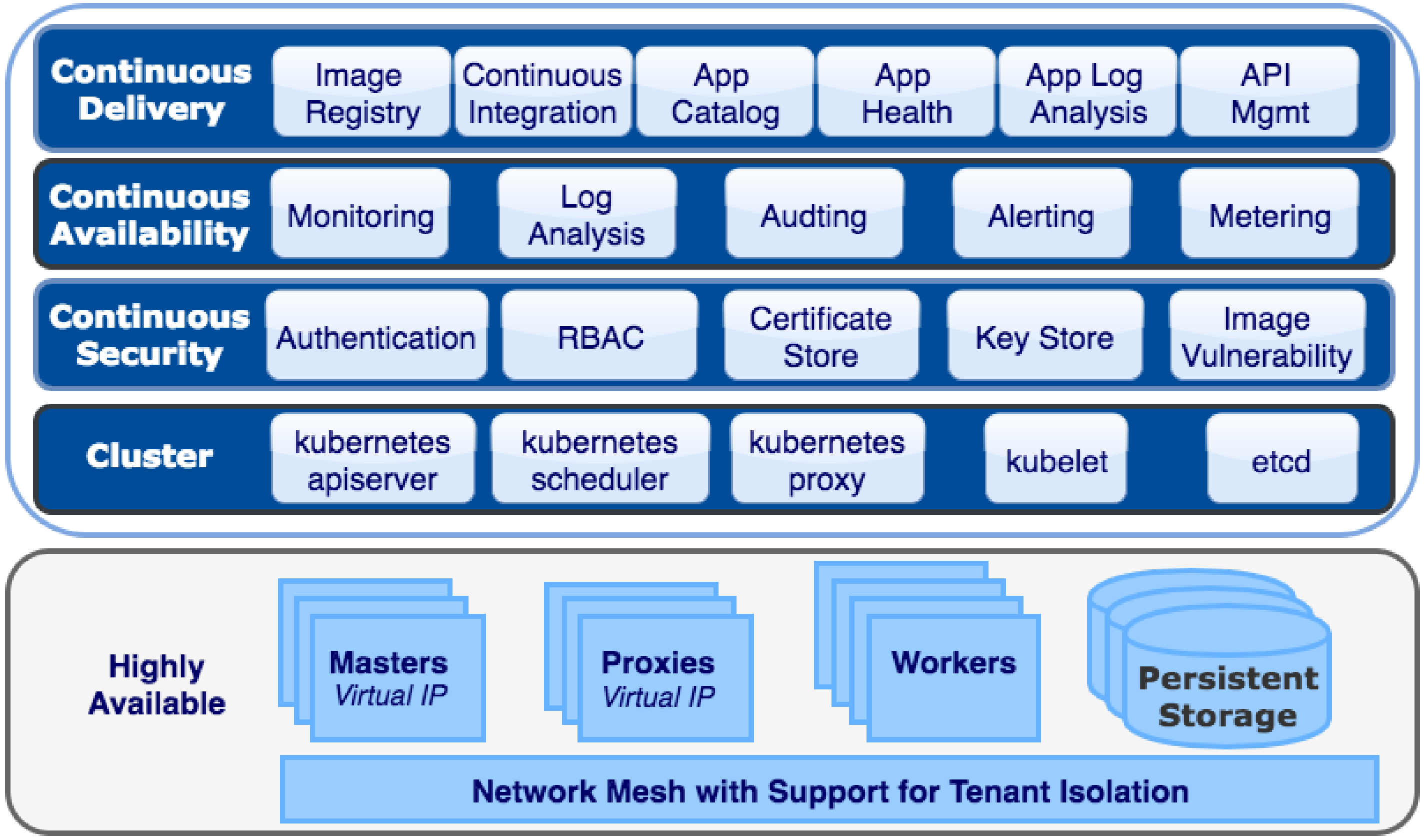
Leverages IBM Cloud & Watson

Native Kubernetes Experience

Integrated Operational Tools

IBM Cloud Private – Kubernetes Architecture

- **<u>ICP Node types:</u>**
- **Boot** (x86/Power):
- Used to Install and scale an ICP platform.
- **Master** (x86/Power):
- Used to Manage your cluster and schedule and Monitor your deployment. (HA possible with multiple master)
- **Worker** (x86/Power/Zlinux):
- System which runs your containers. Can be arch mixed (x86,Power,Z)
- **Proxy** (x86/Power):
- Transmits external request to the services created inside your cluster. (HA possible with multiple proxy)

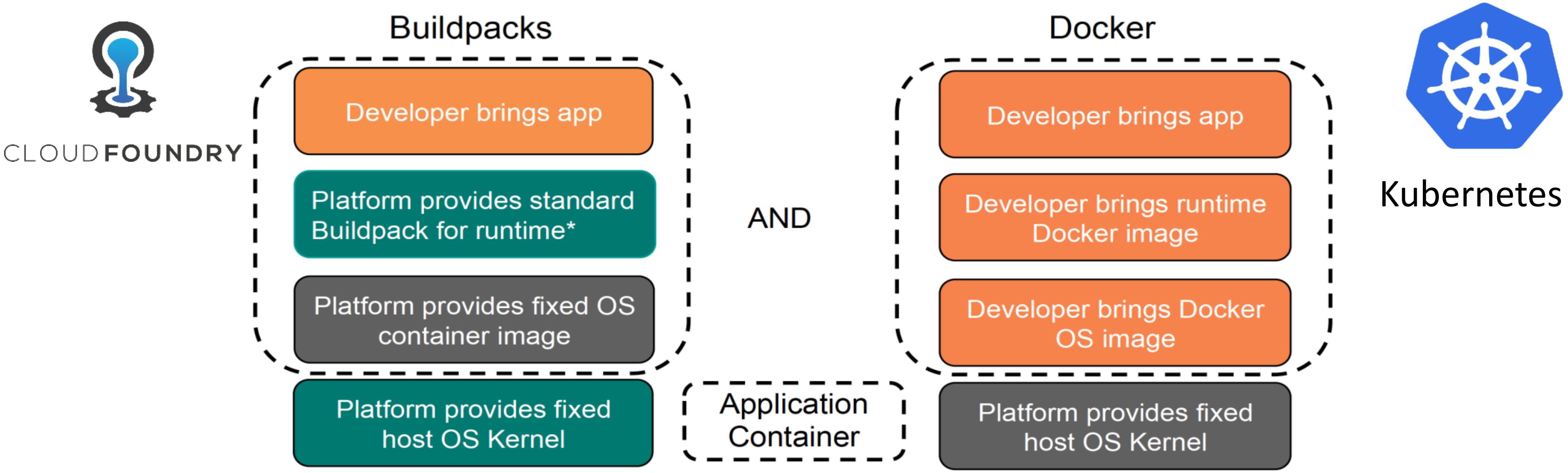


[View Detailed Component List and Versions](https://www.ibm.com/support/knowledgecenter/SSBS6K_1.2.0/getting_started/components.html)

https://www.ibm.com/support/knowledgecenter/SSBS6K_1.2.0/getting_started/components.html

All Node types can be installed on the same HW Instance.

Cloud Foundry vs Kubernetes


CLOUDFOUNDRY
Buildpacks
Developer brings app
Platform provides standard Buildpack for runtime*
Platform provides fixed OS container image
Platform provides fixed host OS Kernel
AND
Application Container
Docker
Developer brings app
Developer brings runtime Docker image
Developer brings Docker OS image
Platform provides fixed host OS Kernel
Kubernetes

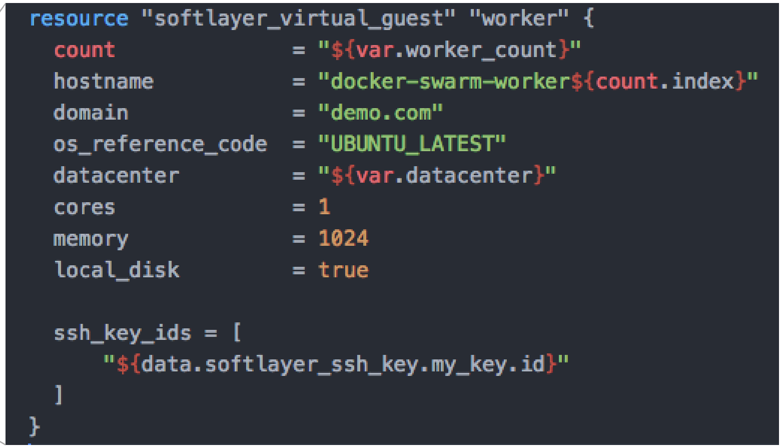
*Node .Js, Tomcat, Pearl, Ruby on Rails …

- **CF and Docker Containers with Kubernetes (K8) are operating on different levels of abstraction**

- **Choices to make**
  - Select K8 when you want the flexibility to control all of the underlying technology, and deploy whatever you'd like wherever you'd like at large scale – Optimizing for Performance & Scalability.
  - Chose CF if you prefer to write code, and have the platform take care of all of the "plumbing" to get the code into a running application, monitor it's health, and scale at the expense of control and flexibility – Optimizing for Speed & Simplicity

- **IBM's recommendation** – **Use Both platforms at their best**
  - Do rapid prototyping on Cloud Foundry PaaS
  - Harden prototypes & deploy on an Enterprise Grade K8 platform

Terraform – IBM Cloud Automation Manager (CAM)

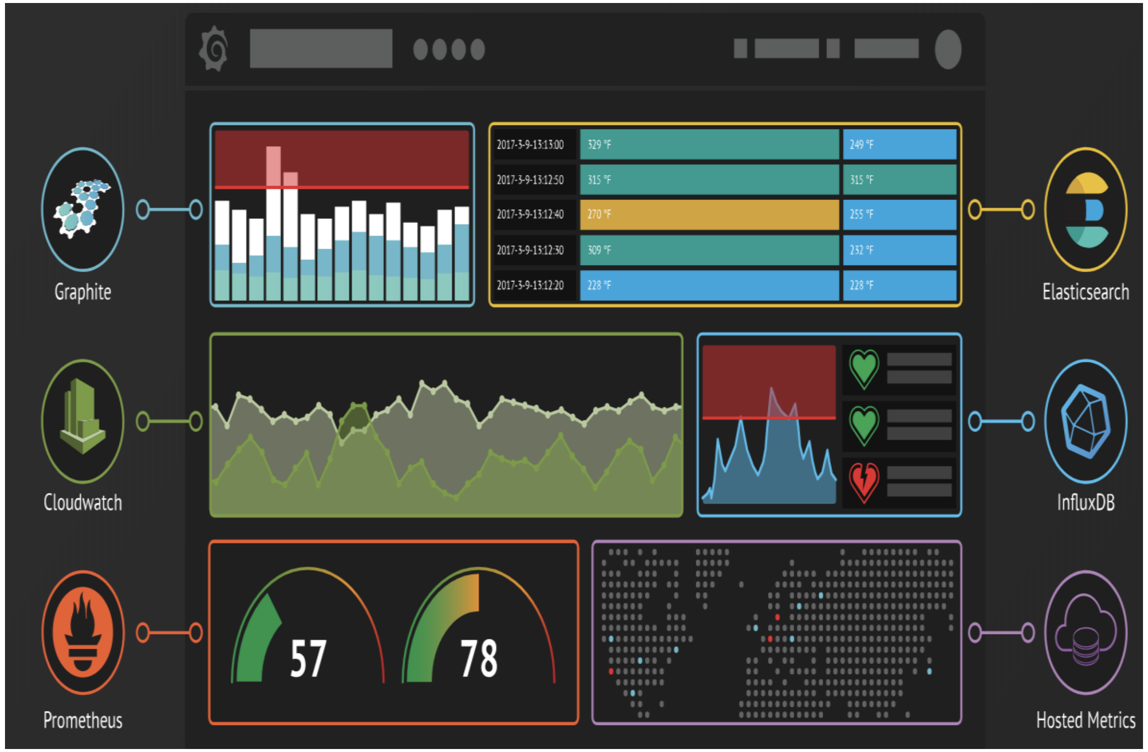


```
resource "softlayer_virtual_guest" "worker" {
  count              = "${var.worker_count}"
  hostname           = "docker-swarm-worker${count.index}"
  domain             = "demo.com"
  os_reference_code  = "UBUNTU_LATEST"
  datacenter         = "${var.datacenter}"
  cores              = 1
  memory             = 1024
  local_disk         = true

  ssh_key_ids = [
      "${data.softlayer_ssh_key.my_key.id}"
  ]
}
```

Terraform is an infrastructure as code software. It allows users to define a datacenter infrastructure in a high-level configuration language, from which it can create an execution plan to build the infrastructure in a service provider such as IBM SL , AWS, Google, MSFT as well as Open Stack & VMWare

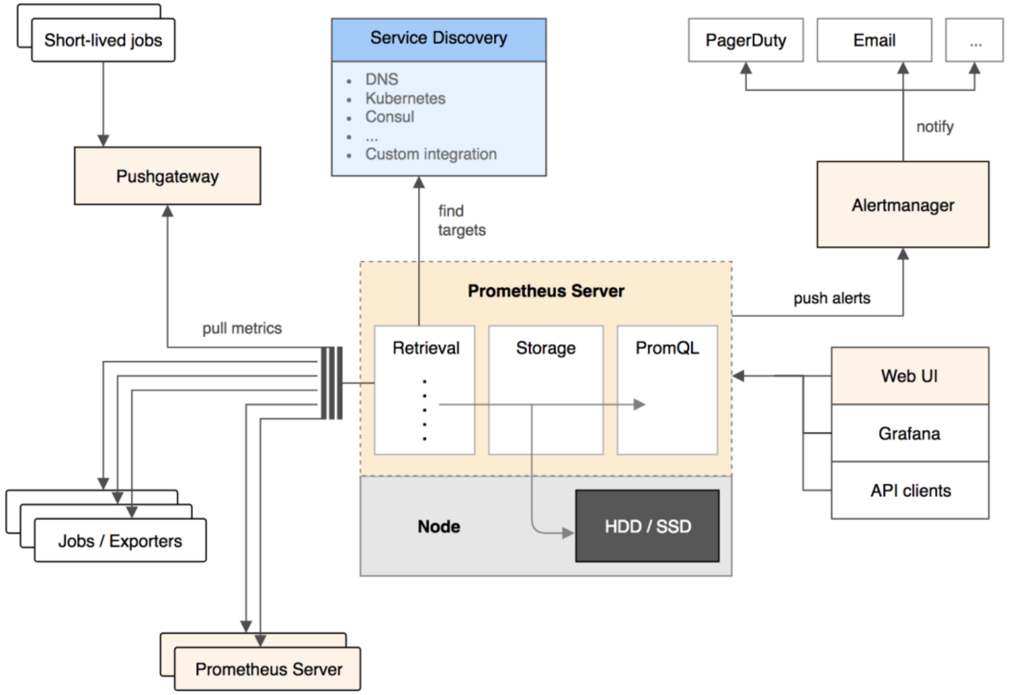
Grafana & Prometheus

Grafana : The **open** platform for **beautiful** analytics and monitoring



2017-3-9-13:13:00	329 °F	249 °F
2017-3-9-13:12:50	315 °F	315 °F
2017-3-9-13:12:40	270 °F	255 °F
2017-3-9-13:12:30	309 °F	232 °F
2017-3-9-13:12:20	228 °F	228 °F

Prometheus : is an open-source systems **monitoring and alerting** toolkit



Cloud Computing

Market Tre

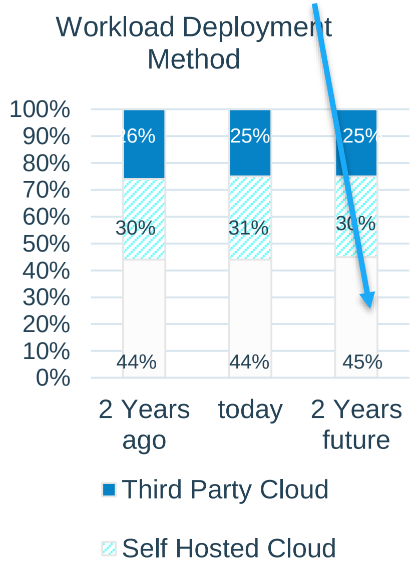
Private Cloud has a vital role
Transformations using DevOps leveraging PaaS & Containers

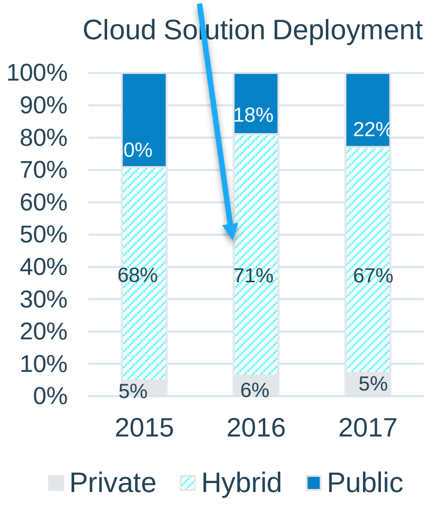
On Premise servers are still important

Hybrid is the main cloud solution model

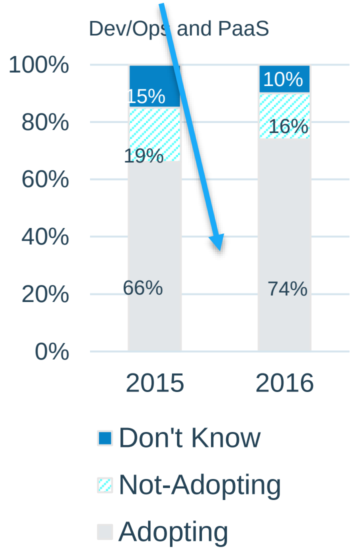
Dev/Ops and PaaS development

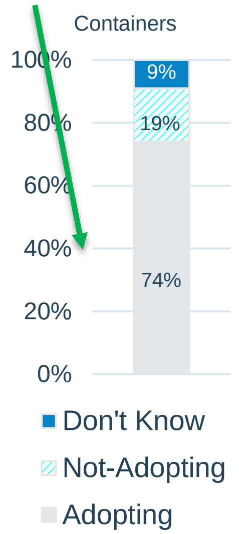
Container based Deployment





Hybrid is the interconnection of multiple compute environments





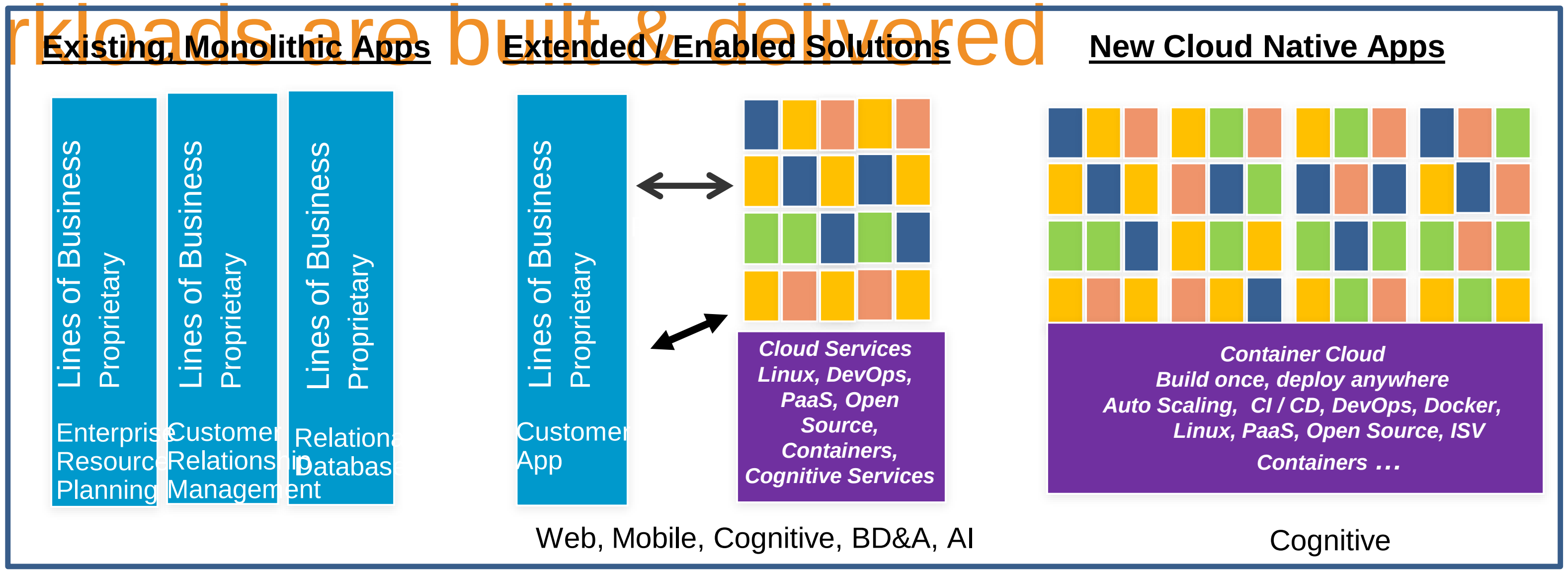
Source: IBM Institute for Business Value Analysis

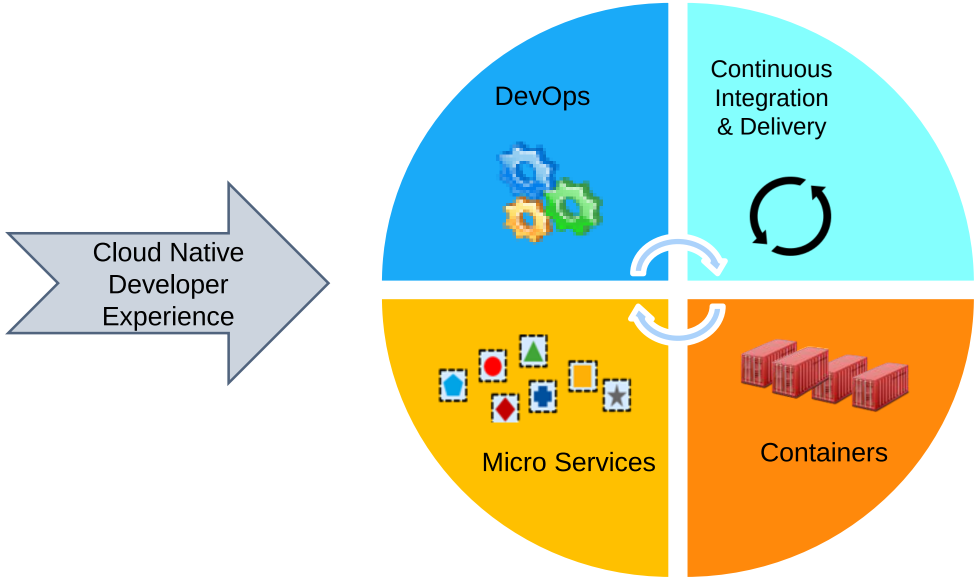
Source: RightScale 2017 State of the Cloud Report

Source: RightScale 2017 State of the Cloud Report

Source: 2016 MI Survey

Cloud is changing how workloads are built & delivered


Existing, Monolithic Apps
Extended / Enabled Solutions
New Cloud Native Apps
Lines of Business Proprietary
Enterpris Resource Planning
Lines of Business Proprietary
Customer Relationship Management
Lines of Business Proprietary
Relationa Databas
Lines of Business Proprietary
Customer App
Cloud Services Linux, DevOps, PaaS, Open Source, Containers, Cognitive Services
Container Cloud
Build once, deploy anywhere
Auto Scaling, CI / CD, DevOps, Docker, Linux, PaaS, Open Source, ISV
Containers …
Web, Mobile, Cognitive, BD&A, AI
Cognitive


Cloud Native Developer Experience
DevOps
Continuous Integration & Delivery
Micro Services
Containers

By 2018, Over **60%** of **New Apps** Will Use Cloud-Enabled Continuous Delivery and

Cloud-Native Application

Architectures to Enable Faster Innovation and Business Agility.
(***IDC Prediction)***

Why Cloud ?

1. **Cloud** has evolved as a strategy for disruption driven by continuous delivery.

2. Cloud elasticity enables **microservices** architectures to scale out quickly, but also roll new updates out at immense speeds.

3. **Data** becomes the fuel for business innovation.

4. **AI** becomes the catalyst to turn data into **brilliant** user experiences.

5. **Profit**!