# MQ and IIB Deployment Patterns using Docker on IBM Cloud Private

**Sandeep Chellingi**

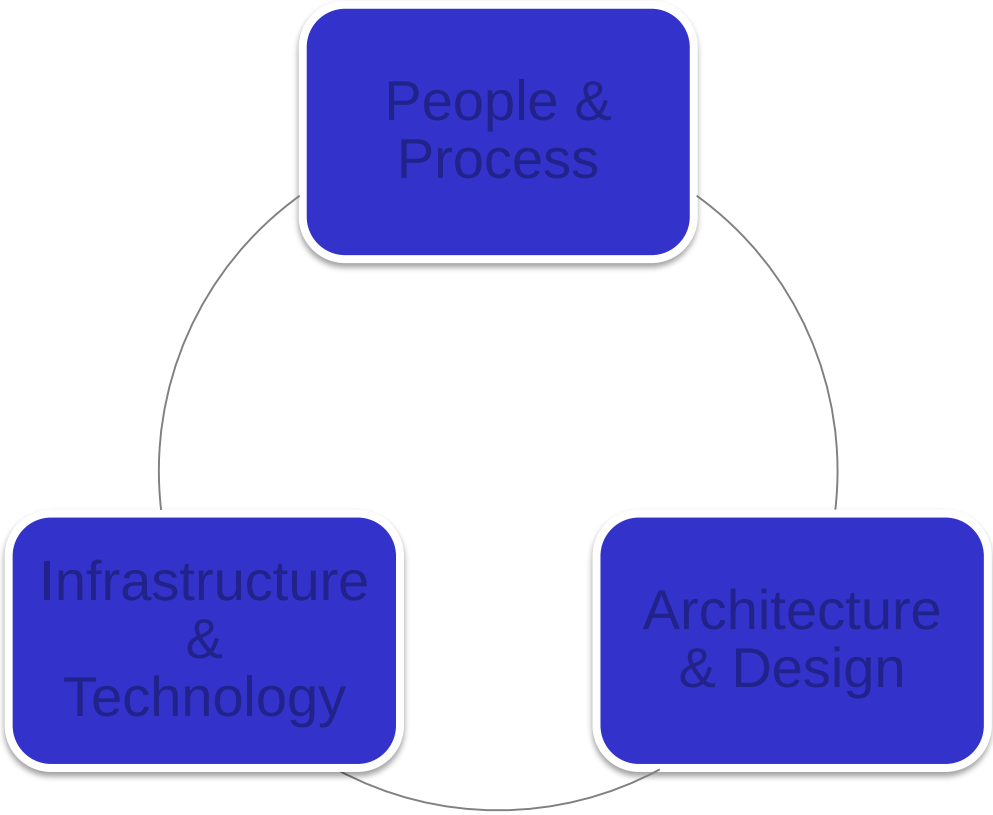
**Head of Hybrid Cloud Integration , Prolifics**

# Agenda

- Integration Modernization
- Agile Integration Architecture
- SOA vs Micro Services
- Evolution of Agile Architecture
- Containers
- High Availability Scenarios
- IBM Cloud Private

# Integration Modernization

**Modernization will impact more than just your software**



# Agile Integration Architecture

## Agile Integration Architecture drives the change

| | | |
|---|---|---|
| Fine grained deployment | → | Speed development & simplify mgmt |
| Decentralized Ownership | → | Accelerate agility & innovation |
| Cloud native infrastructure | → | Provide resiliency and scalability |

# Typical benefits sought from a move to microservices


**Agility**

Faster iteration cycles, bounded contexts, autonomous teams

**Scalability**

Elastic scalability, workload orchestration, cloud infrastructure

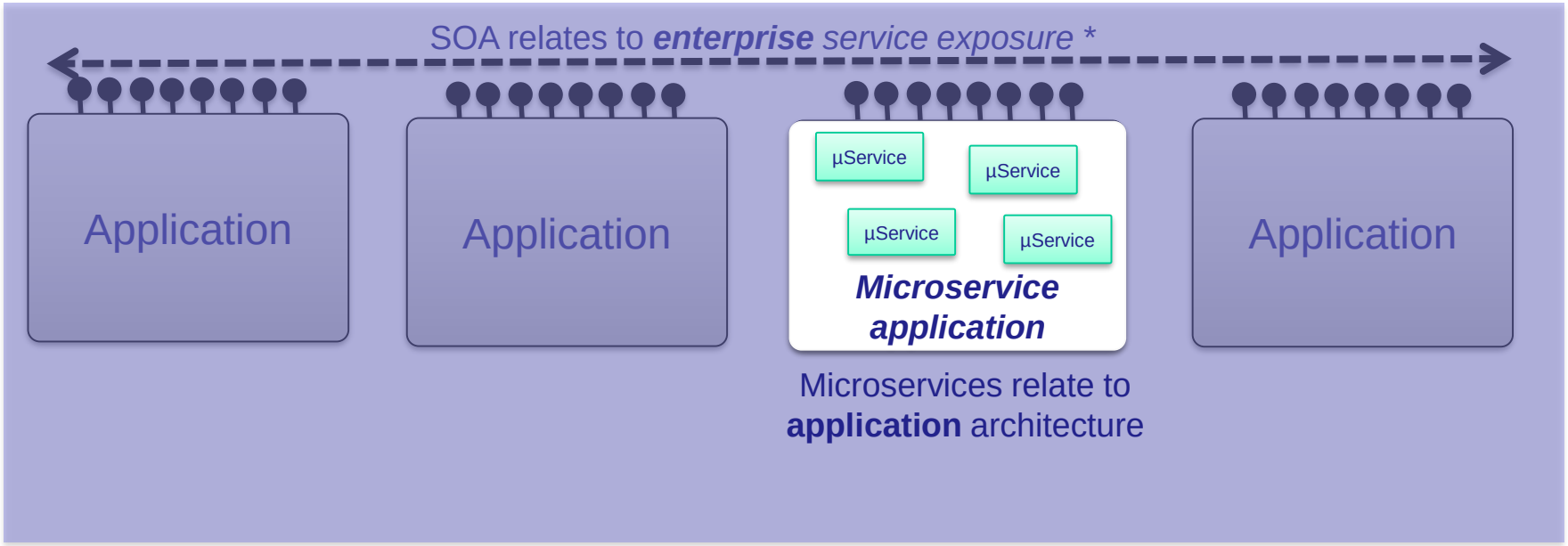
**Resilience**

**Minimized dependencies, discrete failover, fail fast, start fast**

# Difference between SOA and Micro services

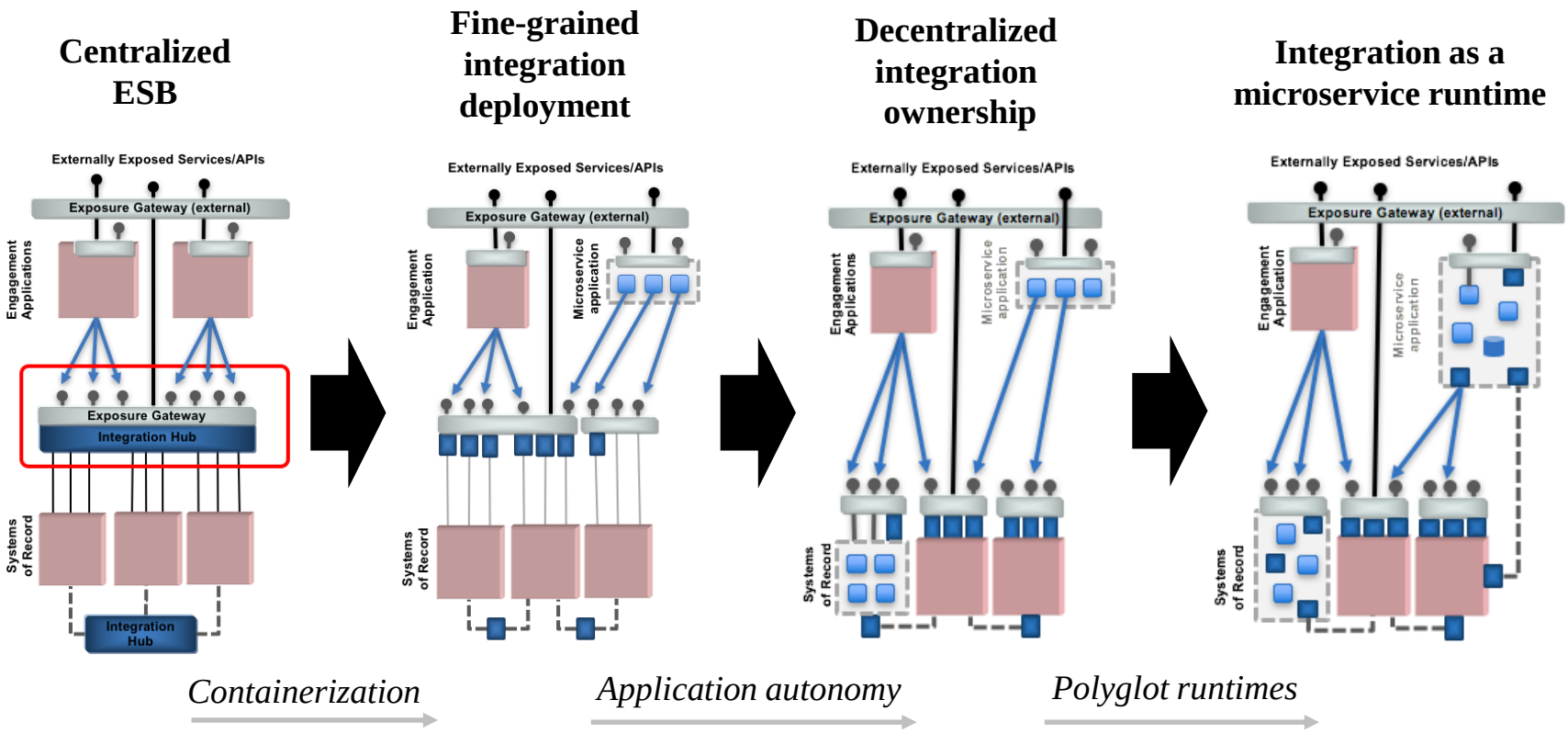
**Service oriented architecture (SOA) and microservices architecture relate to different scopes**

**They are *complementary*, rather than *competing***




Webinar based on above paper (55 mins) http://ibm.biz/MicroservicesVsSoaFullWebinar

# Evolution to agile integration



# Benefits of a container-based approach

**Build Agility**
- Higher build velocity
- Faster maintenance cycles
- Consistency across environments
- Independent component deployment
- Simplified testing of isolated components

**Fine-grained Resilience**
- Safe independent deployment, removing risk of destabilizing existing components.
- Disposable components enabling rapid start/stop for simple HA and scaling.
- Fit for purpose discrete topologies

**Infrastructure Optimization**
- Maximized component/resource density
- Lower overheads than virtual machine isolation
- Dynamic and elastic provisioning of resources (CPU, memory, persistent volumes)
- Usage based licensing models

**Operational Consistency**
- Standardized infrastructure platform skillset across products
- Platform based load balancing
- Platform based high availability
- Platform based scaling via policy
- Platform based logging/monitoring

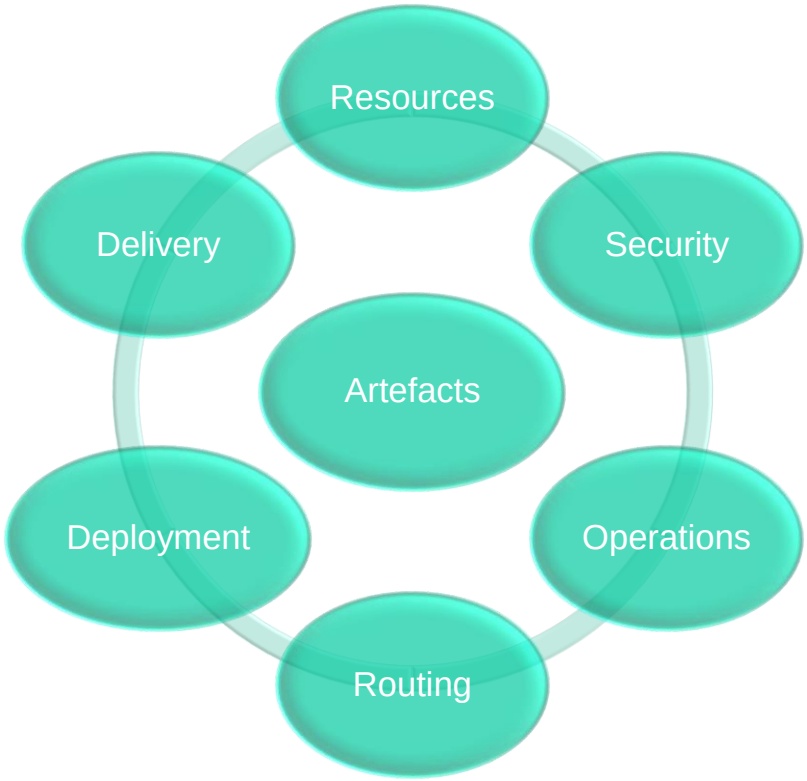
**Component Portability**
- Containers can be re-distributed dynamically across nodes within a given cloud
- Images can be built and run on any cloud
- Focus on open containerisation standards such as Docker and Kubernetes
- Enables multi-cloud scenarios

**Scalability & Continuous Availability**
- Fine-grained dynamic scaling of individual functionality
- Implicit high availability based on replication policy and built in re-instatement
- Provided consistently across all types of components

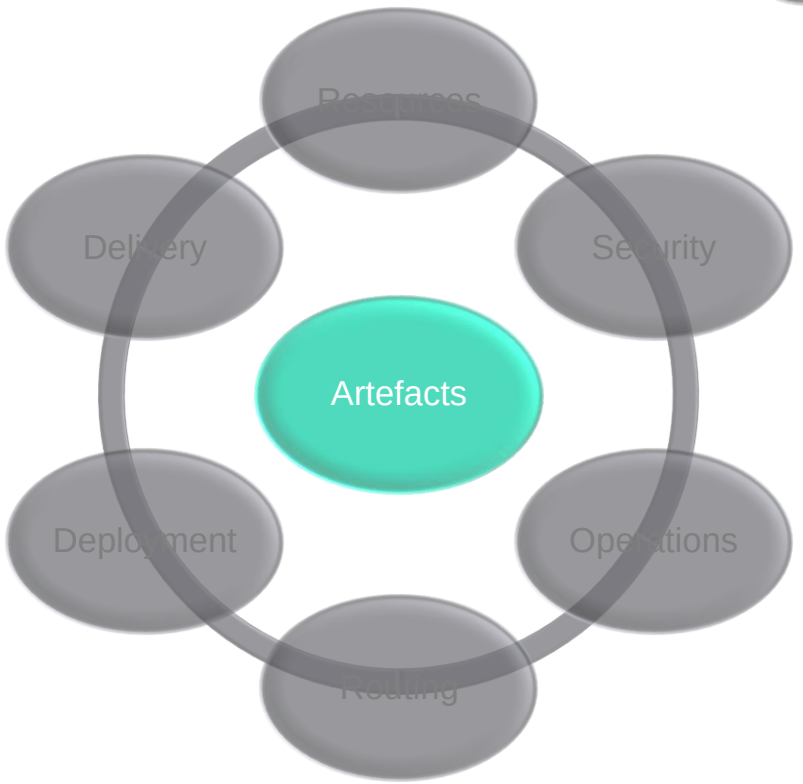
# Operational consistency of container based solutions

Runtime specific

Provided by platform

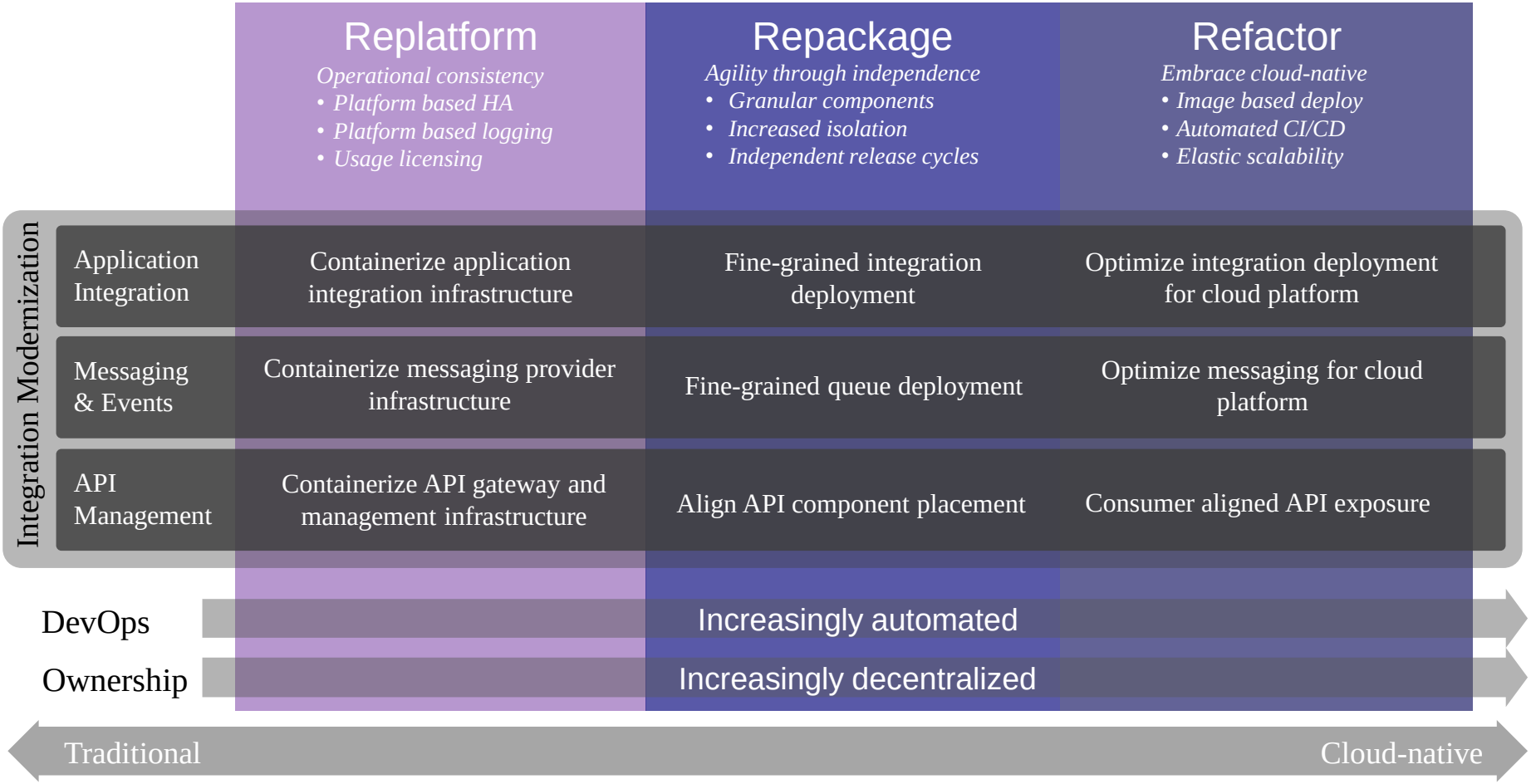



Traditional infrastructure
(Pets)



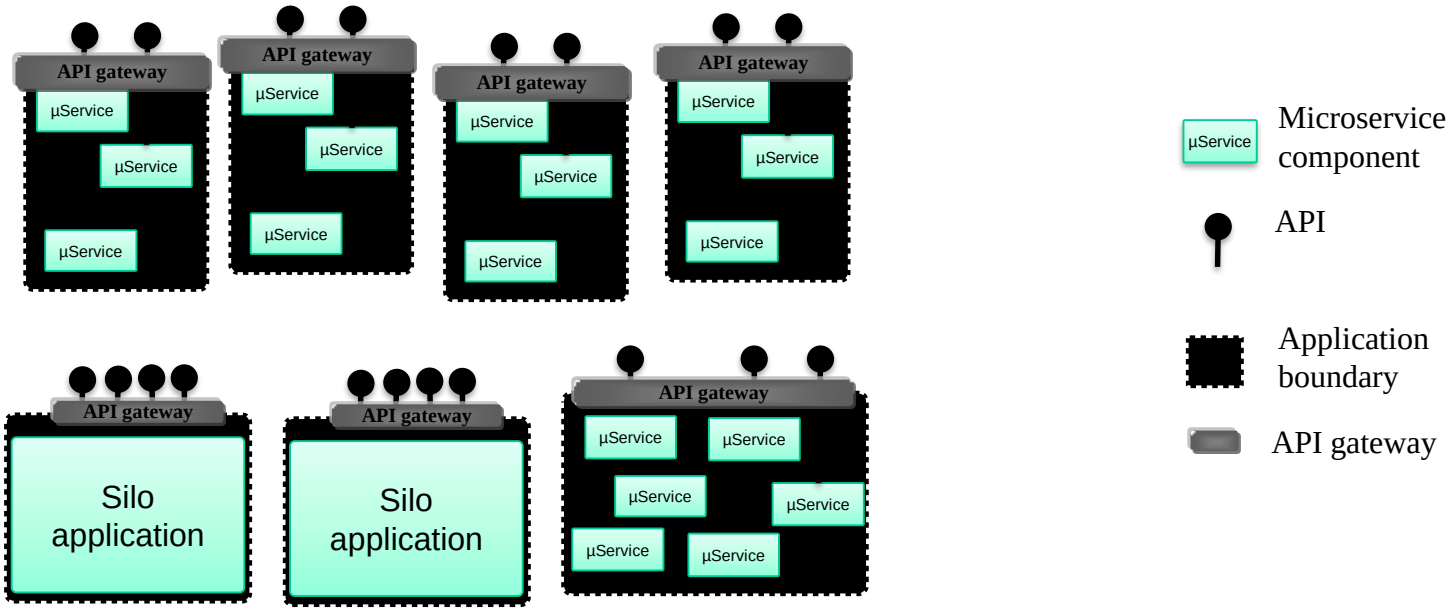

Cloud native infrastructure
(Cattle)

# The scope of Integration Modernization



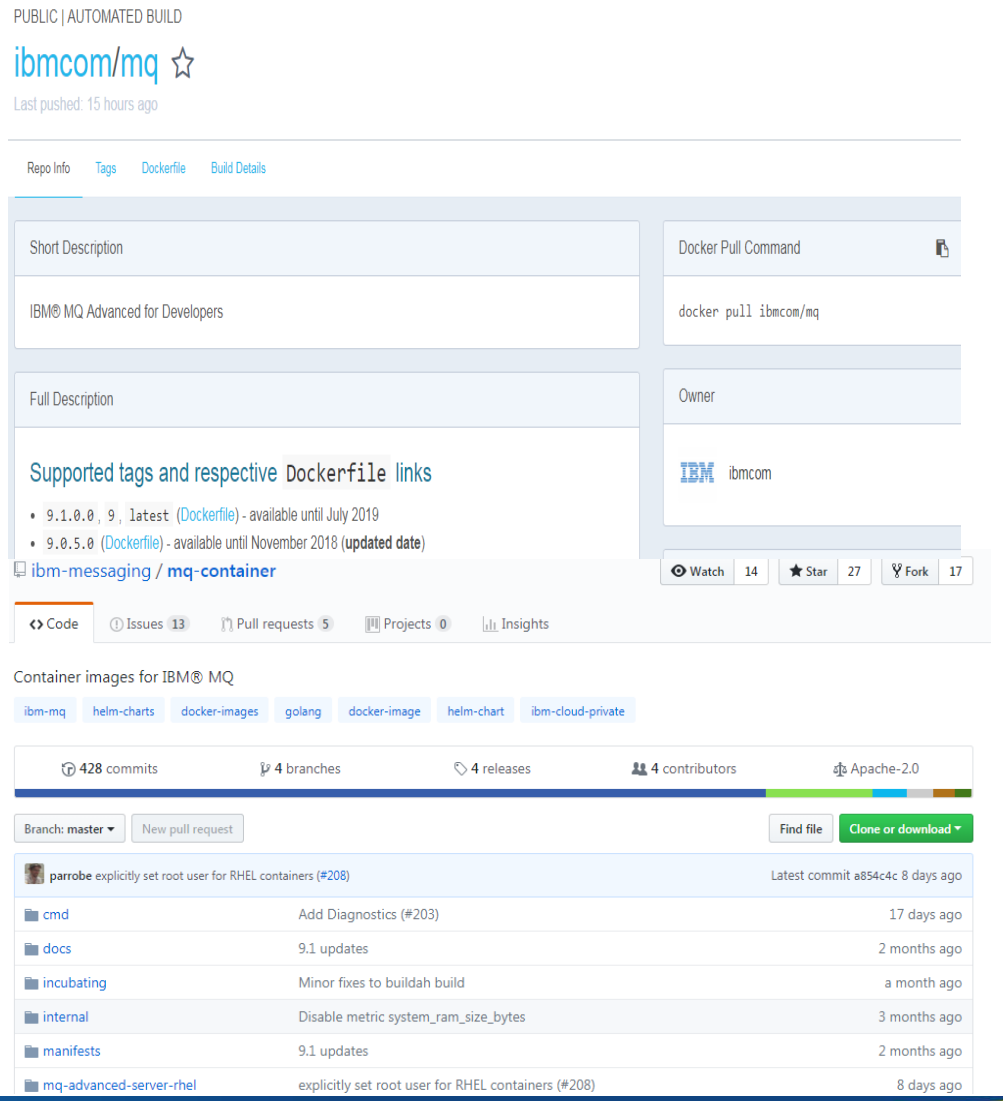
# Boundaries make complex environments manageable

*__Managed API gateways__ define and enforce application boundaries*



# Running MQ in Containers

- MQ has been supporting Docker containers since 2015 with images on Docker Hub and Docker Store and sample code on Github
- Recently it has been demonstrating how to get most of docker containers and kubernetes providers like Redhat open shift, Pivotal container service
- MQ Advanced is available as a fully supported product with IBM Cloud Private, a Kubernetes-based solution from IBM



# MQ container orchestration support

| | IBM Cloud Public | IBM Cloud Private | | Other Container Services (Docker Hub/Store) | Other Container Orchestration services (e.g.OpenShift) | Other |
|---|---|---|---|---|---|---|
| Component/arch | x86_64 | x86_64 | ppc64le (POWER) | s390x (z/Linux) | x86_64 | x86_64 | * |
| MQ Advanced Server Production | ● | ★ | ▲ | ▲ | ● | ▲ | ▲ |
| Client (dev & prod) | ▲ | ▲ | ▲ | ▲ | ▲ | ▲ | ▲ |
| MFT & AMS & MQTT | ▲ | ▲ | ▲ | ▲ | ▲ | ▲ | ▲ |
| SDK | ▲ | ▲ | ▲ | ▲ | ● | ▲ | ▲ |
| MQ Explorer | ▲ | ▲ | ▲ | ▲ | ● | ▲ | ▲ |
| Salesforce Bridge | ▲ | ▲ | ▲ | ▲ | ● | ▲ | ▲ |
| Blockchain bridge | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| RDQM | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |
| MQ IPT | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ | ✖ |

★ **MQ supported with image and sample available**
Supported, Image and Helm chart available

● **MQ supported with sample**
Supported, and you need to build your own image (samples/blog available)

▲ **MQ supported with no sample**
Supported, and you need to build your own image.

✖ **Not supported**

# Running IIB/ACE in Docker containers

Q IIB  Dashboard  Explore  Organizations  Create  sandeep17jy

PUBLIC REPOSITORY

## ibmcom/iib ☆

Last pushed: 5 months ago

Repo Info  Tags

| Short Description | Docker Pull Command |
| --- | --- |
| Official IBM Integration Bus for Developers image | `docker pull ibmcom/iib` |


CONTAINERS

ot4i / ace-docker  Watch 8  Star 9  Fork 11

Code  Issues 4  Pull requests 1  Projects 0  Insights

*No description, website, or topics provided.*

18 commits  1 branch  0 releases  2 contributors  EPL-2.0

Branch: master  New pull request  Find file  Clone or download

dan robinson Merge branch 'master' of https://github.com/ot4i/ace-docker  Latest commit f947ce1 on Jun 4

| | | |
| --- | --- | --- |
| 11.0.0.0/ace/ubuntu-1604 | Merge branch 'master' of https://github.com/ot4i/ace-docker | 4 months ago |
| CLA.md | Update repo layout and add CLA and README | 5 months ago |
| LICENSE | Initial commit | 5 months ago |
| README.md | Further README updates | 5 months ago |

# Helm Charts

## IIB Helm Charts

readme.md

### IBM INTEGRATION BUS

IBM® Integration Bus is a market-leading lightweight
way for systems and applications to communicate wit
business value, reduce IT complexity and save money
choices, skills and interfaces to optimize the value of

### Introduction

This chart deploys a single IBM Integration Bus for Developers integrat
IBM Cloud Private or other Kubernetes environment.

### Installing the Chart

To install the chart with the release name `foo`:

```
helm install --name foo ibm-integration-bus-dev --set license=accept
```

### Configuration

The following table lists the configurable parameters of the `ibm-integration-bus-dev` chart and their default values.

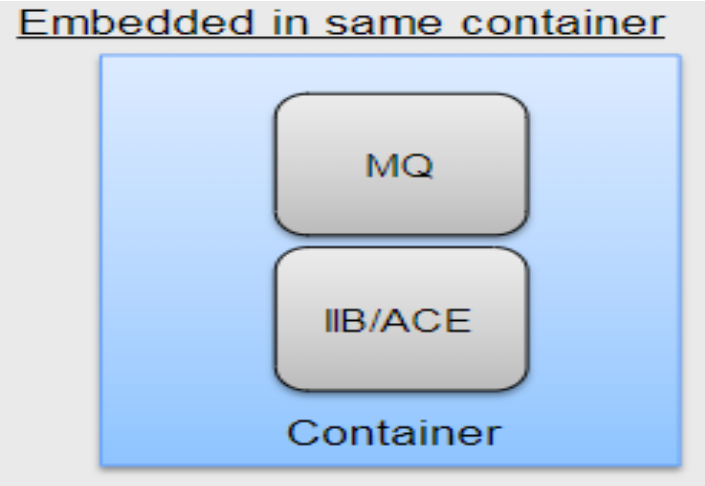
| Parameter | Description | Default |
|---|---|---|
| `license` | Set to `accept` to accept the terms of the IBM license | `Not accepted` |
| `image.repository` | Image full name including repository | `ibmcom/iib` |
| `image.tag` | Image tag | `10.0.0.10` |
| `image.pullPolicy` | Image pull policy | `IfNotPresent` |
| `image.pullSecret` | Image pull secret, if you are using a private Docker registry | `nil` |
| `service.name` | Name of the Kubernetes service to create | `qmgr` |
| `service.type` | Kubernetes service type exposing ports, e.g. `NodePort` | `NodePort` |
| `resources.limits.cpu` | Kubernetes CPU limit for the Queue Manager container | `2` |
| `resources.limits.memory` | Kubernetes memory limit for the Queue Manager container | `2048Mi` |
| `resources.requests.cpu` | Kubernetes CPU request for the Queue Manager container | `1` |
| `resources.requests.memory` | Kubernetes memory request for the Queue Manager container | `512Mi` |
| `nodename` | IBM Integration Bus integration node name | IIB_NODE |
| `servername` | IBM Integration Bus integration node name | IIB_SERVER |

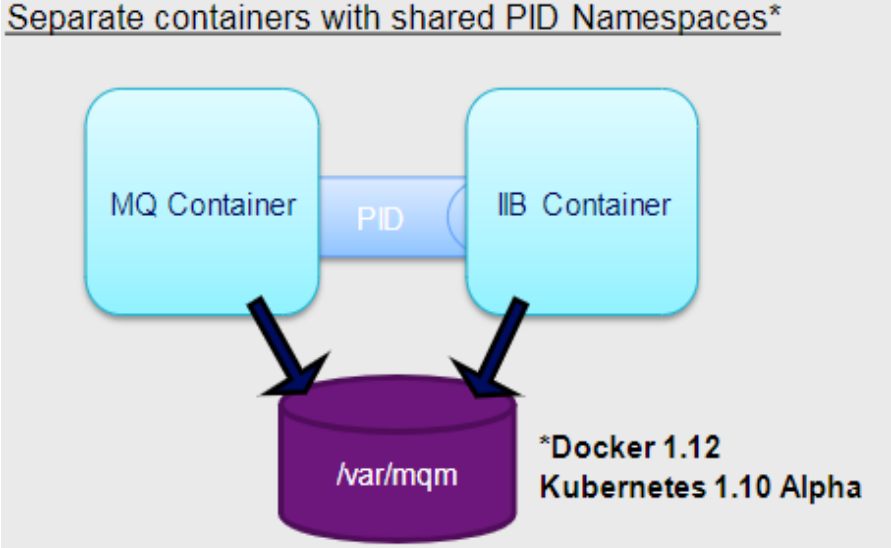
# How and why does IIB/ACE use MQ ?

- As an asynchronous messaging provider
- By certain nodes to maintain state (Collector, Resync, etc)
- As a co-coordinator for global (two phase commit) transactions

# Running MQ and IIB/ACE in containers

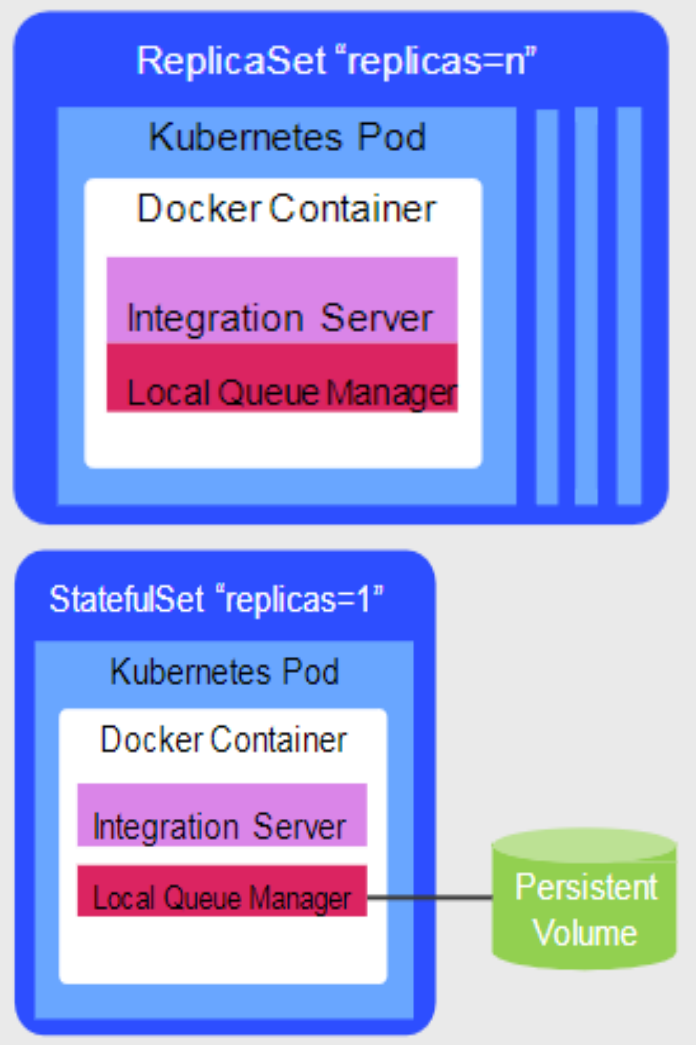
- Use cases where IIB/ACE can use Client bindings connections then run MQ and IIB/ACE in separate containers.
- Use cases where IIB/ACE requires local (server) bindings connections can be deployed in different options
  - Embedded in Same Container
  - Separate Container with same PIS Namespace



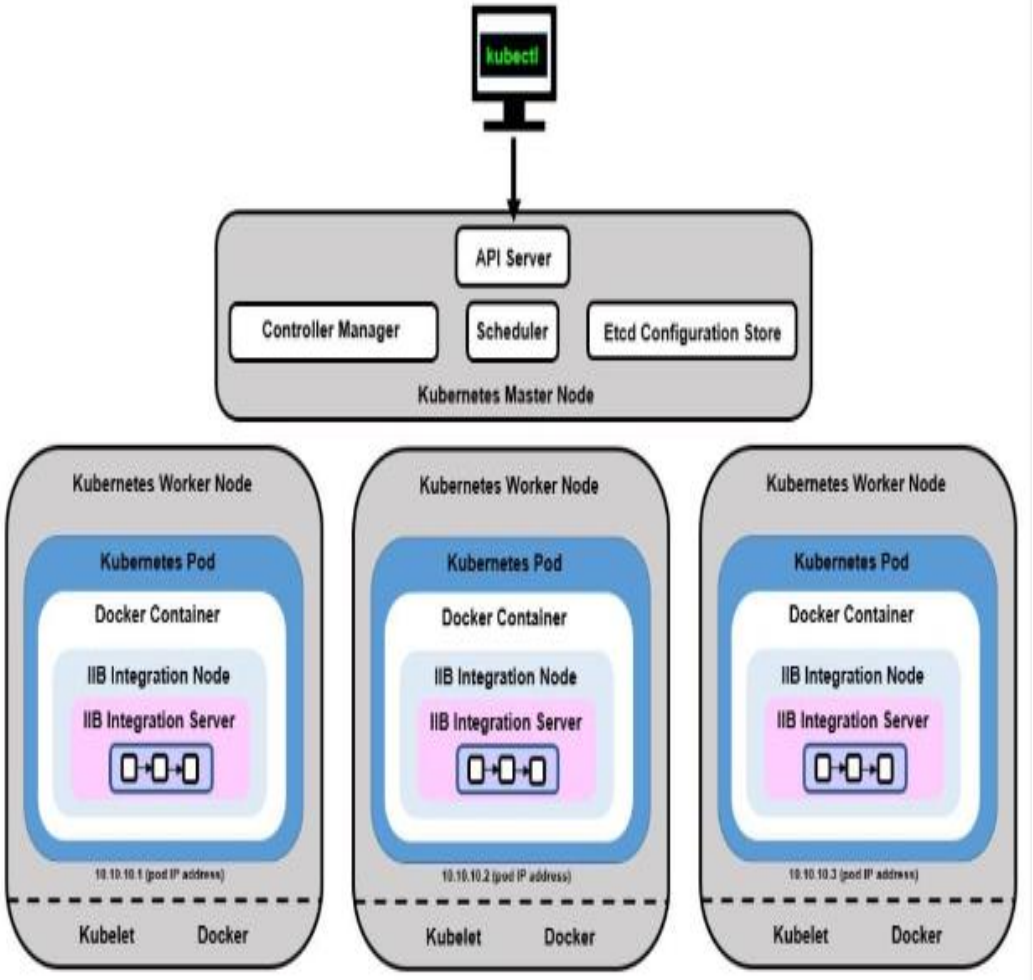
# Running MQ and IIB/ACE in same containers

- replicas =n
  - no persistent volume claim
  - HA by replication (continuous availability)
  - Elastic horizontally scalability
  - Non-durable use of EDA*nodes
  - No 2 Phase Commit.
- replicas =1
  - persistent volume claim
  - HA by reinstatement
  - Manual horizontal scalability
  - Durable use of EDA*nodes
  - 2 Phase Commit



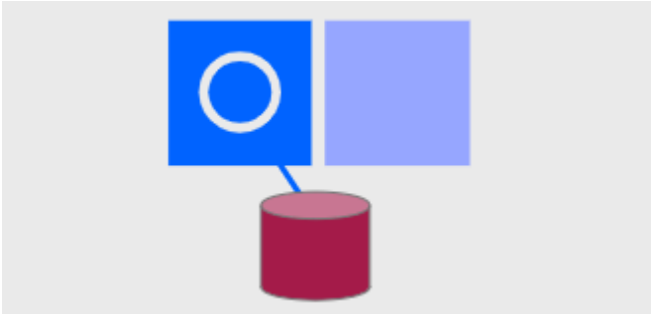
# IIB running in Kubernetes

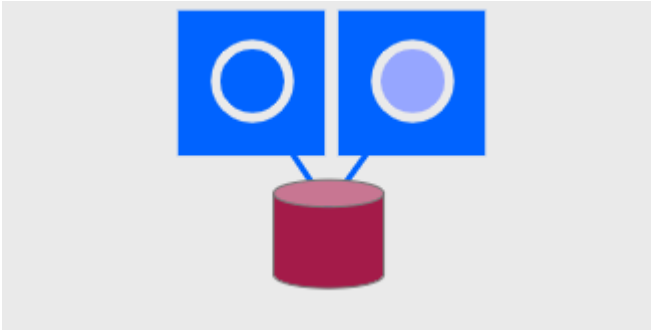
- The IBM Cloud Container Service provides a Kubernetes-based public cloud solution
- IBM Cloud Private provides a Kubernetes-based private cloud solution for running in your own datacenter

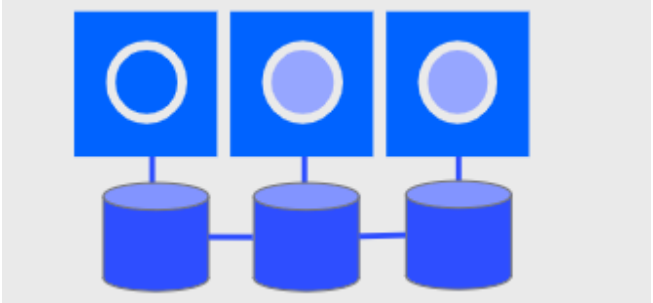


# MQ HA Availability on Docker/Kubernetes

- **Single resilient queue manager**
  - Cloud manages failover to somewhere with spare capacity
  - Networked storage (block or filesystem), managed by separate subsystem
- **MQ Multi-Instance Queue Manager**
  - Active – Standby pair ,MQ Manages Failover
  - Shared Network Storage managed by different sub System
- **Replicated data queue manager**
  - “Shared Nothing” approach ,MQ manages failover
  - Local block storage, synchronously replicated by MQ
  - Not supported on Containers

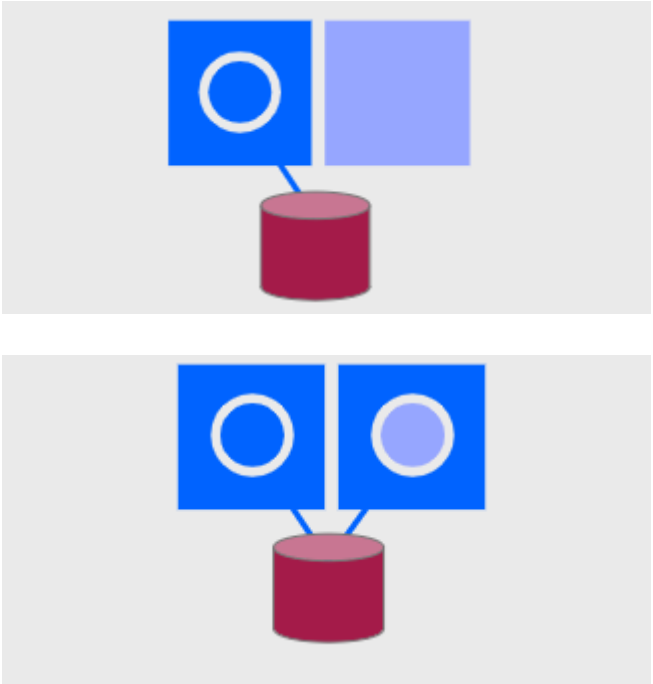






# IIB/ACE HA Availability on Cloud

- **Single resilient Integration Node (v10) or Integration Server (v11)**
  - Cloud manages fail-over to somewhere with spare capacity
  - Networked storage (if needed) managed by separate subsystem
- **Multi-instance Integration Node (v10)**
  - Requires local MQ
  - MQ manages fail-over
  - ACE v11 not supported (yet)
  - Networked storage (filesystem), managed by separate subsystem



# IBM Cloud Private Solution

**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





**Core Operational Services**

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



**Kubernetes-based Container Platform**

Industry leading container orchestration platform



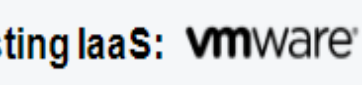
**Cloud Foundry**

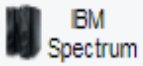
For prescribed application development & deployment


**Terraform (CAM)**

Infrastructure as Code for multi-cloud provisioning to public and on-prem private clouds

**Runs on existing IaaS:**






Dell, Cisco, NetApp, Lenovo, ...

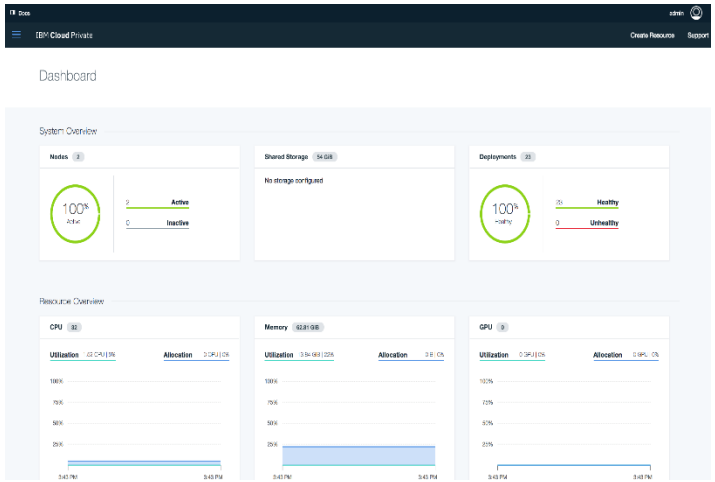
# IBM Cloud Private Overview

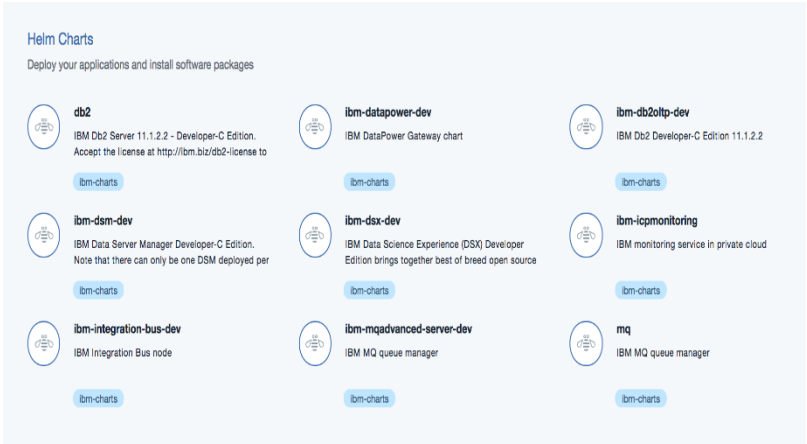
A customer-managed Private Cloud software solution based on Kubernetes, Docker and Cloud Foundry technology that runs on customer-provided infrastructure (or in Public Cloud IAAS)

A platform to run containerized versions of IBM Software such as Datapower, IIB, MQ, DB2, Cloudant, Data Science Experience (Apache Spark), Blockchain

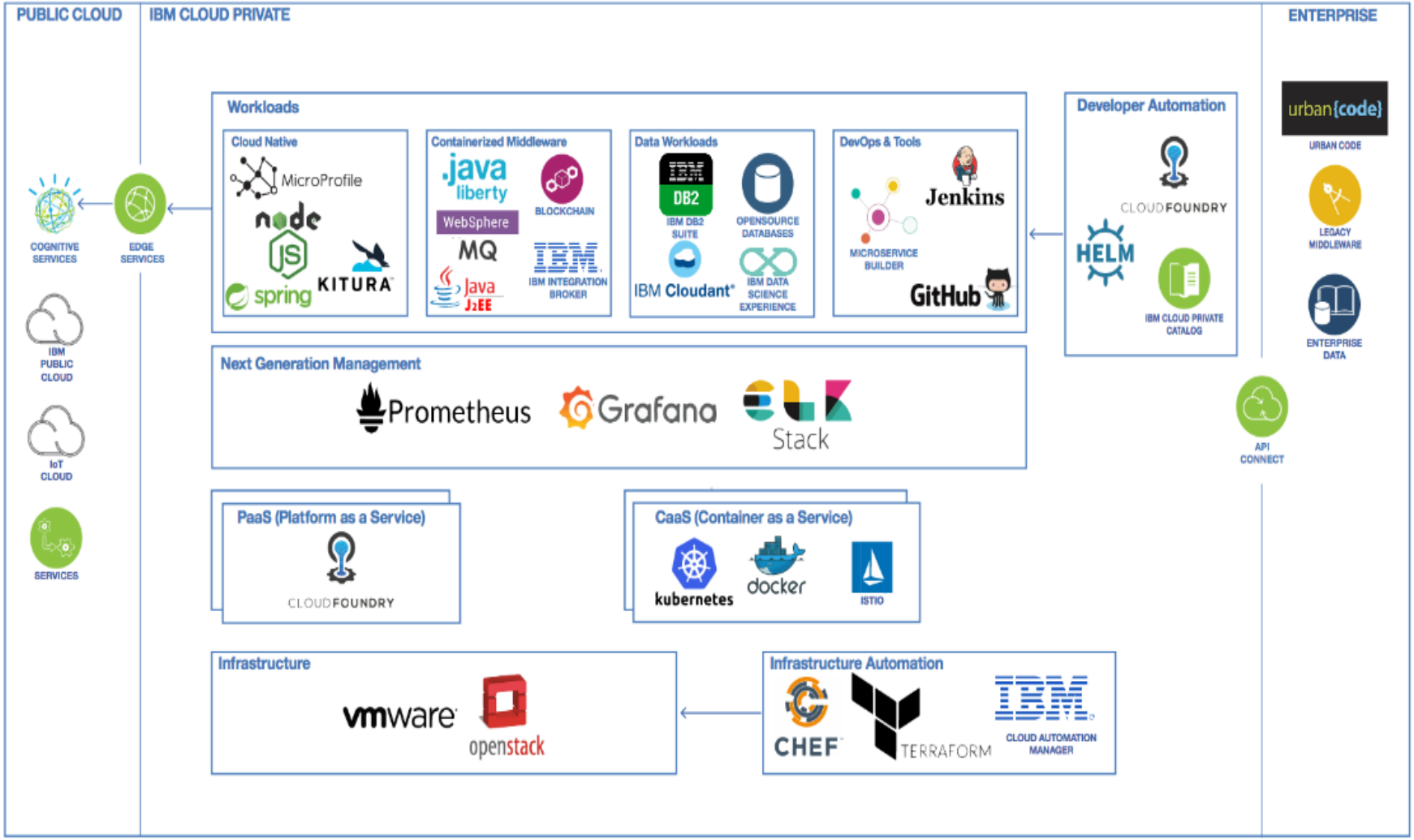
A platform to build Cloud Native, Stateless, 12 Factor apps including powerful developer tools to jump-start projects

A platform to run Modernized and Containerized Legacy Applications including tools and services to help transform code.

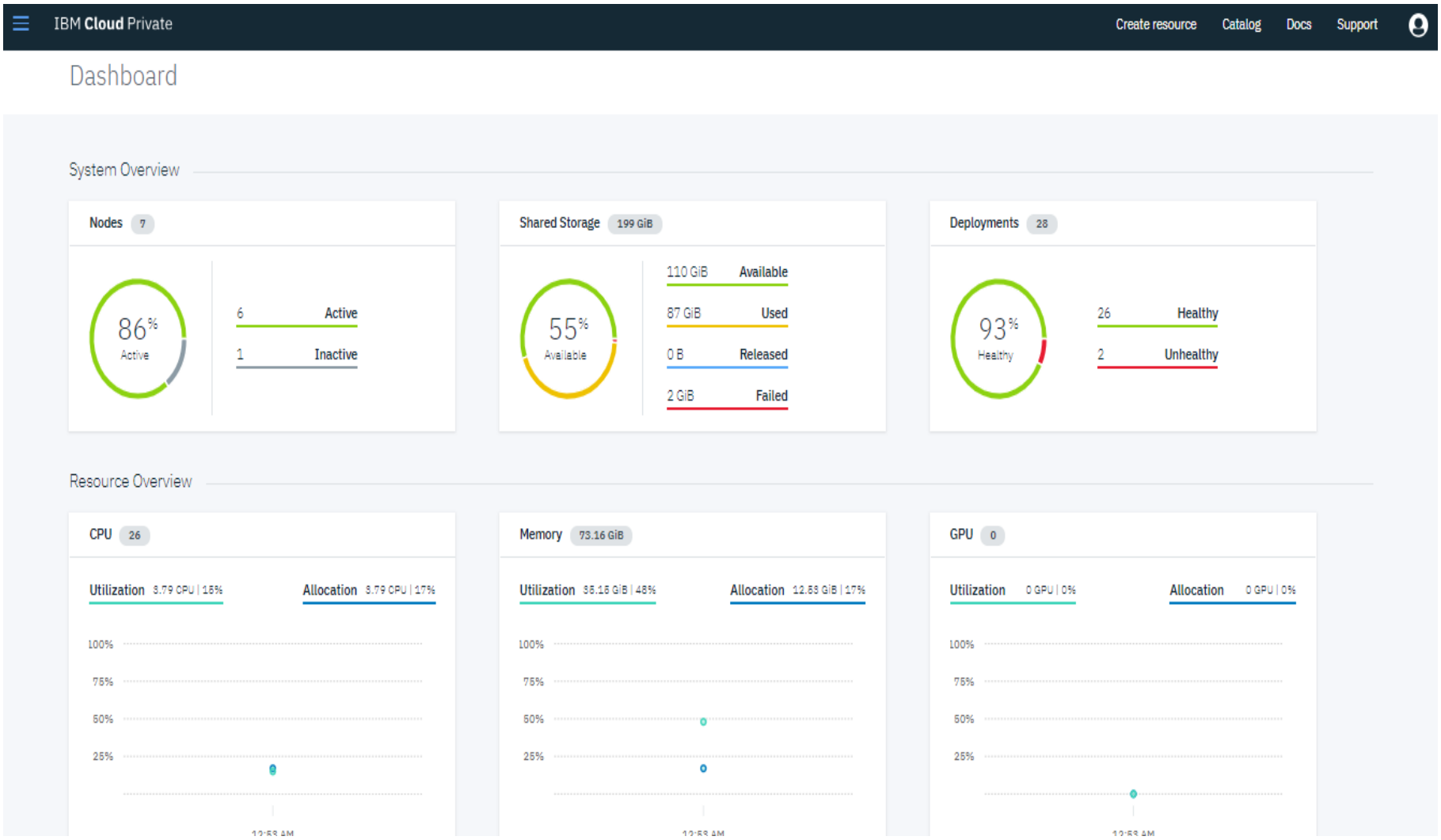


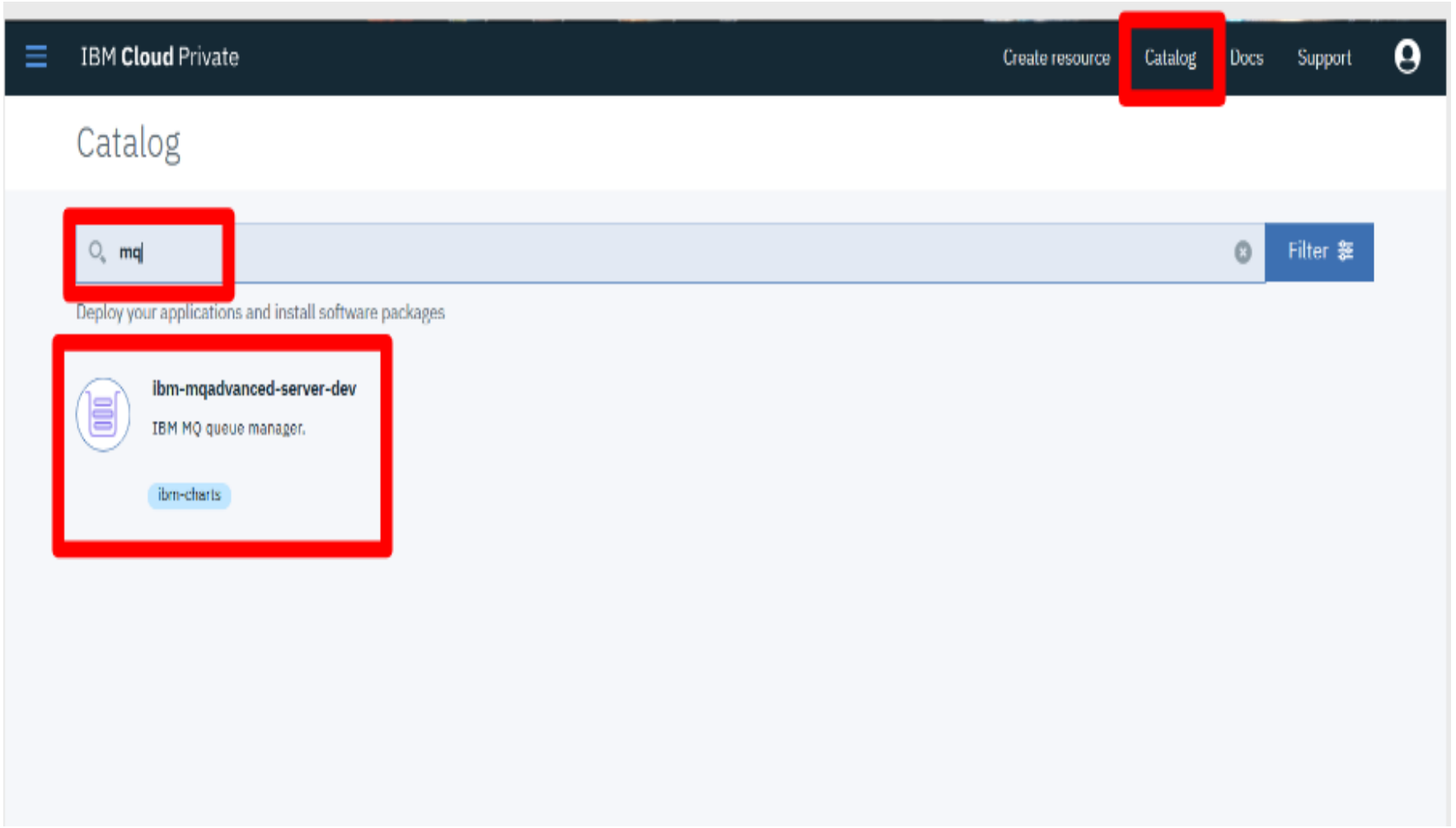
# IBM Cloud Private Architecture



# IBM Cloud Private Dashboard



# IBM Cloud Private Helm Catalog



# Questions & Answers