Getting the Most out of IIB and App Connect Enterprise (ACE)!

Christopher Frank

IBM Cloud - Integration & Messaging

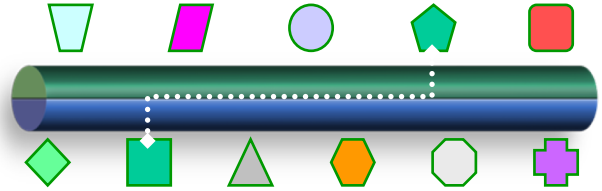
Copyright IBM 2018

Agenda

- Where We Are
- What's New in V11?
- Container Orchestration
- ACE and Containers
- ACE and MQ
- Summary

Copyright IBM 2018

IBM Integration Bus

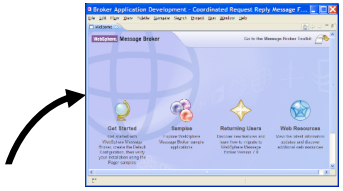


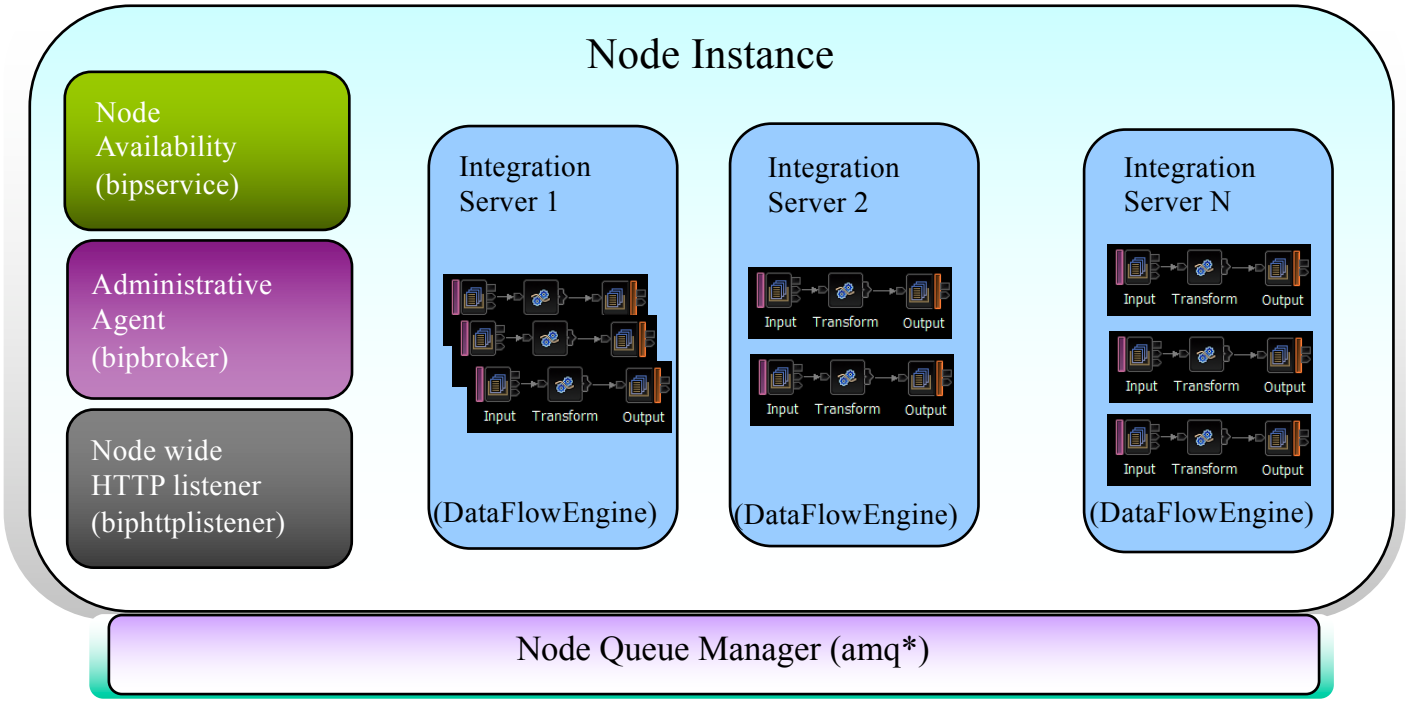
- **Universal Connectivity FROM anywhere, TO anywhere**
  - Simplify application connectivity for a flexible & dynamic infrastructure

- **Comprehensive Protocols, Transports, Data Formats & Processing**
  - Connect to applications, services, systems and devices
    - MQ, JMS 1.1, HTTP(S), SOAP, REST, File (incl. FTP, FTE, ConnectDirect), Database, TCP/IP, MQTT, CICS, IMS, SAP, SEBL, .NET, PeopleSoft, JDEdwards, SCA, CORBA, email…
  - Understand the broadest range of data formats
    - Binary (C/COBOL), XML, CSV, DFDL, JSON, Industry (SWIFT, EDI, HL7…), IDOCs, user-defined
  - Built-in suite of request processors
    - Route, Filter, Transform, Enrich, Monitor, Publish, Decompose, Sequence, Correlate, Detect…

Copyright IBM 2018

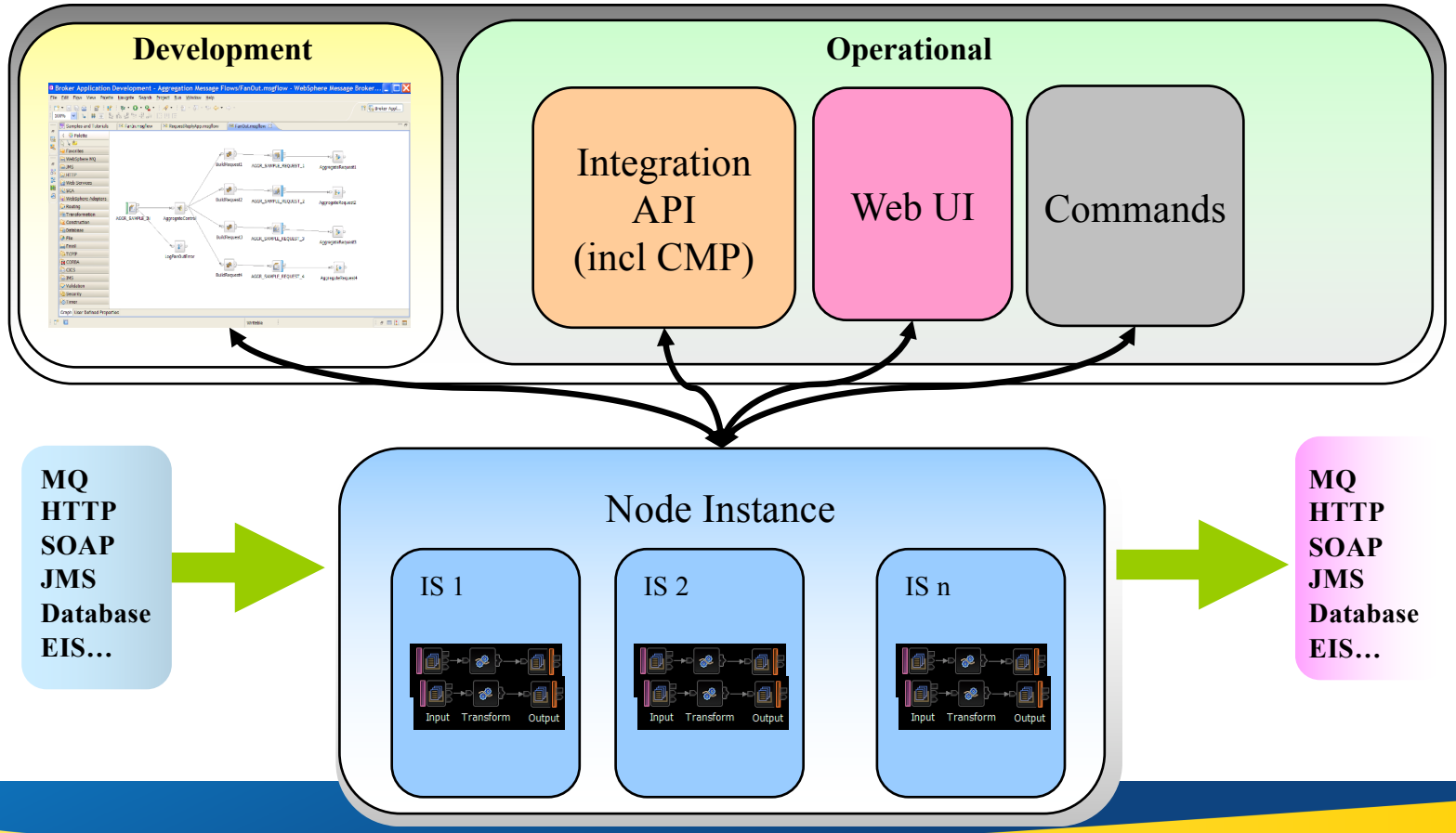
Under the Covers





Copyright IBM 2018

A Node Instance – Development and Operation

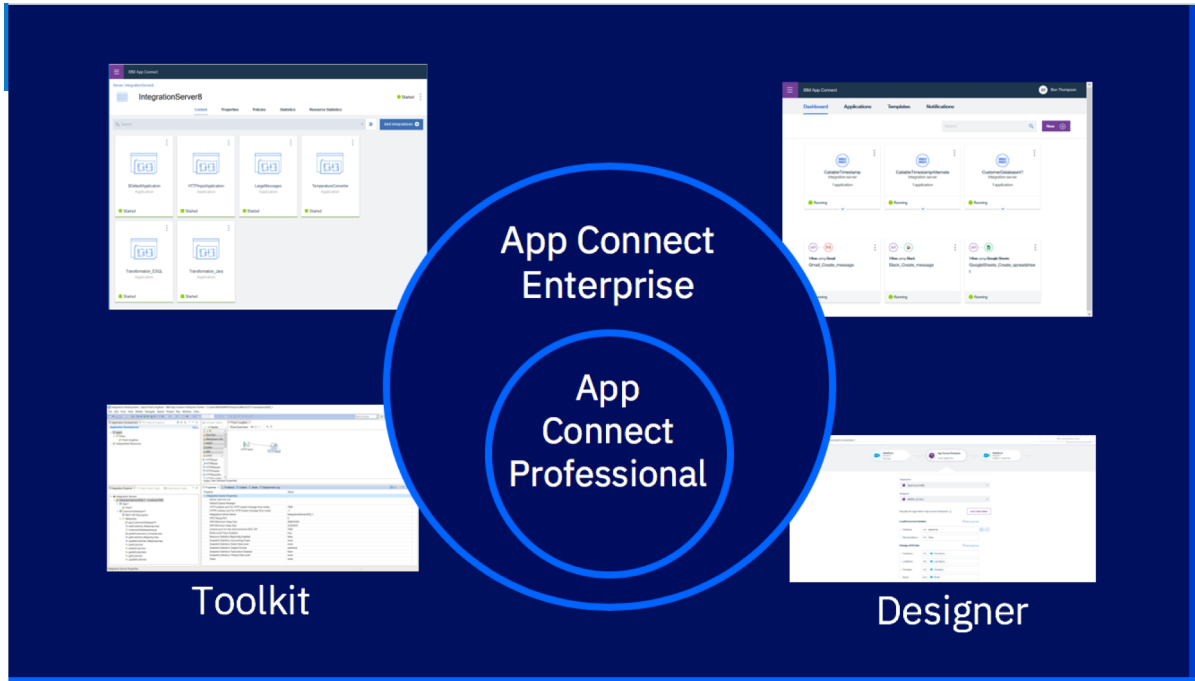


Copyright IBM 2018

#MaximizeYourGame

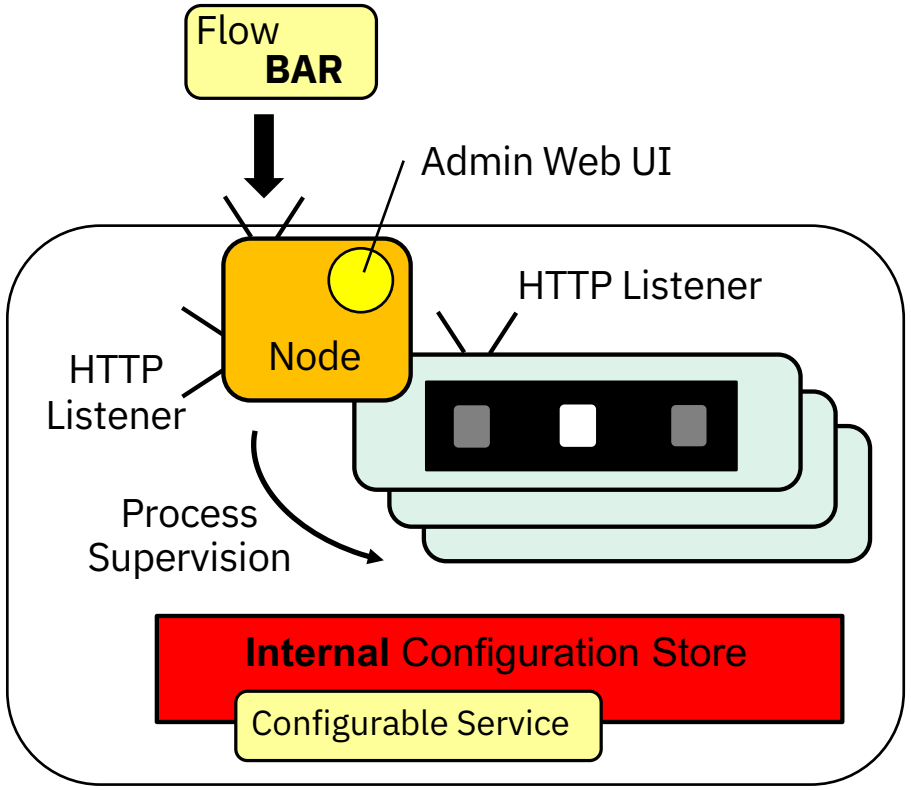
What's Changed with V11?

Copyright IBM 2018



- Extended connectivity support with App Connect Professional and the App Connect connectors on IBM Cloud (adds over 100 smart connectors)
- Leverage extensive pre-built connectors to popular SaaS apps for Marketing, CRM, Finance, HR, Analytics, Project management, ERP. New connectors added each month
- App Connect Enterprise users can also take advantage of their entitlement to App Connect Professional which supports a range of endpoints from pre-packaged applications to custom endpoints through the ability to create custom connectors using the Connector Development Kit

How Things Were (V9/V10 and Before)



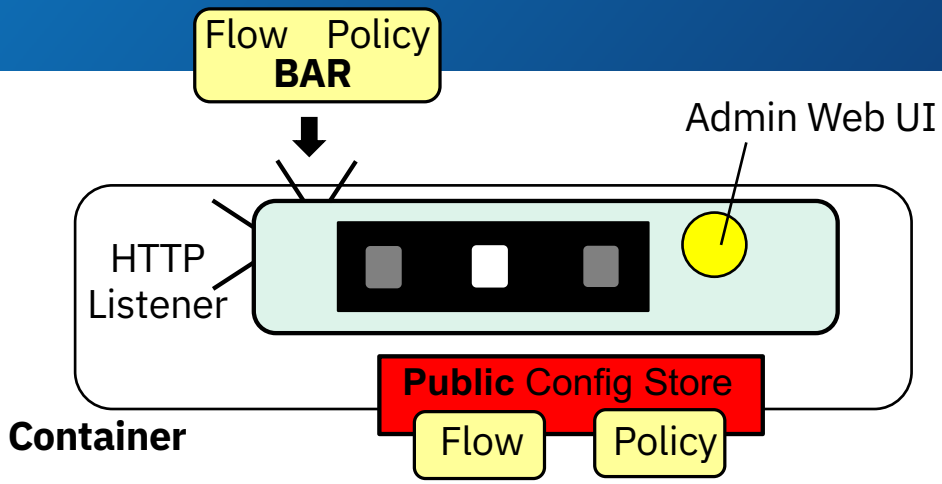
Physical Machine or VM

- **Integration Node was the Control Point**
  - Deployment, Admin, CLI, etc
  - Configurable Services & Policies
  - Process Supervision
  - Configuration Store
- **Could (and often did) include Locally-Bound Queue Manager**
  - MQ inbound/outbound message traffic
  - Global Transaction Coordinator
  - Used by some nodes to maintain state (Collector. Timer, etc)
- **Usually Deployed on a Server or VM**
  - Due to size and complexity
  - Docker Containers supported since 2015

Copyright IBM 2018

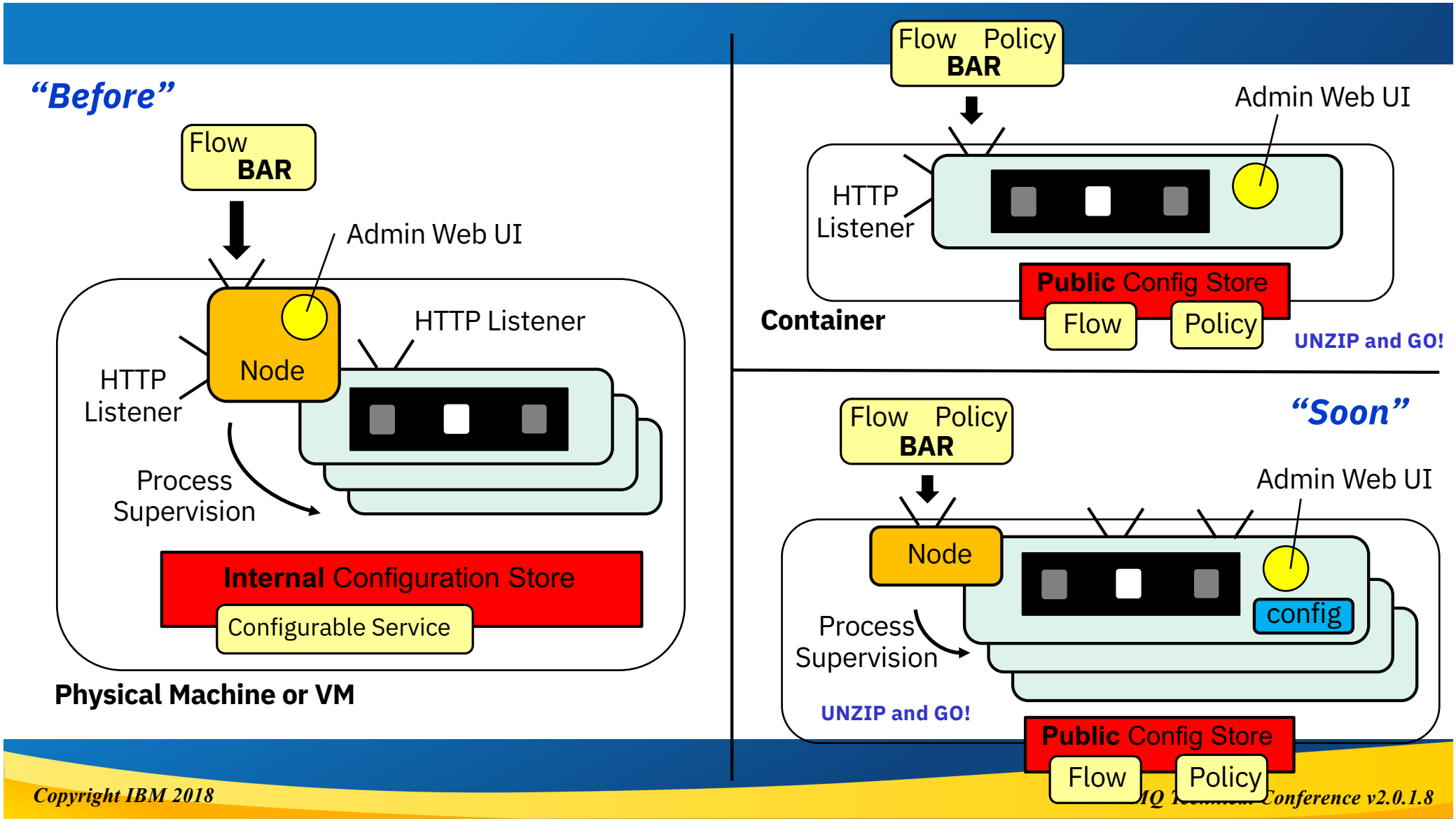
What's Changed in V11?

- **"Autonomous" Integration Servers**
  - No Managing Integration Node required
  - Single OS Process
  - Deployment, Admin, CLI, etc
  - "Public" Configuration Store (flows, policies, etc)
- **No Process Supervision**
  - External Assumed (e.g. Orchestration Engine)
- **Local Queue Manager still an option…**
  - …But Not Recommended (clients preferred)
  - No Global Transaction Coordinator
  - Restrictions on some nodes (Collector, Timer, etc)
- **Redesigned for Deployment in a Container**
  - Single process, reduced footprint



How do I migrate to this???

Copyright IBM 2018



Copyright IBM 2018

#MaximizeYourGame

Container Orchestration

Copyright IBM 2018

Container Orchestration – What's the Big Deal?

- Ease the management and monitoring of large numbers of disparate containers, distributed across multiple hosts
- Container scheduling
  - Placement of containers on the most appropriate host based on constraints
  - Rescheduling if a container or a host fails
  - Ability to rollout/rollback updates whilst maintaining availability
- Deployment of multiple containers that form an application from version controlled configuration
- Policies for placement, security, performance, HA
- Routing of inbound and inter-container requests (service discovery and routing)
- Enterprise integration with existing tools e.g. CI/CD and SSO/RBAC

Copyright IBM 2018

Orchestration Engines

- **Kubernetes** – DOB July 21, 2015
  - Came out of Google's internal Borg project
  - 100% open source, written in GO
  - Can work with any containers including Docker
  - Very powerful, but may take some time to master
  - A large and diversified echo system
  - A number of sites in production

aka K8s

- **Docker Swarm** – DOB November 03, 2015
  - Open source, written in GO
  - Only for orchestrating Docker containers
  - Built-in security and real simple to learn and use
  - Less powerful than Kubernetes
  - Vibrant echo system
  - Some sites in production

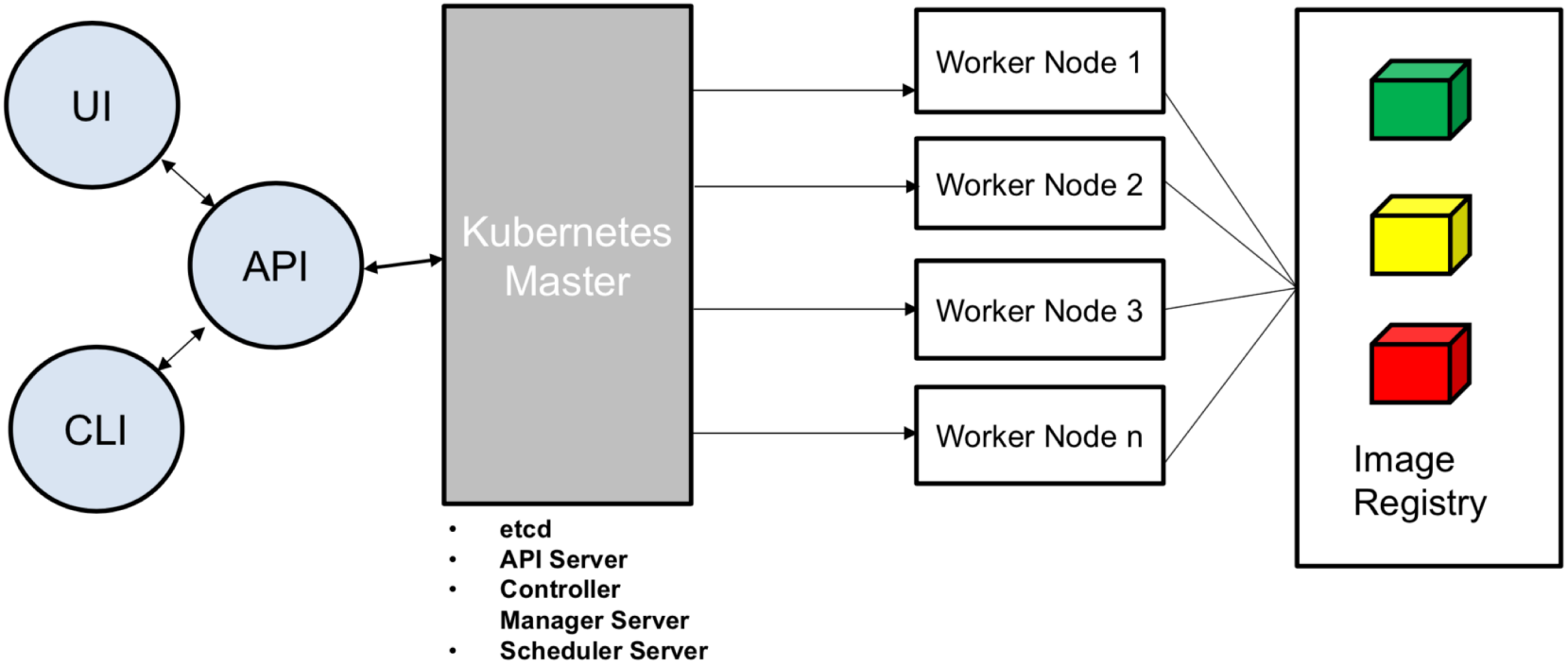
Copyright IBM 2018

Why Kubernetes?

- **Kubernetes has a clear governance model** managed by the Linux Foundation. Google is actively driving the product features and roadmap, while allowing the rest of the ecosystem to participate.
- **A growing and vibrant Kubernetes ecosystem** provides confidence to enterprises about its long-term viability. IBM, Huawei, Intel, and Red Hat are some of the companies making prominent contributions to the project.
- **The commercial viability of Kubernetes makes it an interesting choice for vendors.** We expect to see new offerings announced over the next several months.
- **Despite the expected growth in commercial distributions, Kubernetes avoids dependency and vendor lock-in** through active community participation and ecosystem support.
- **Kubernetes supports a wide range of deployment options.** Customers can choose between bare metal, virtualization, private, public, and hybrid cloud deployments. It enjoys a wide range of delivery models across on-premises and cloud-based services.
- **The design of Kubernetes is more operations-centric** than developer-orientated, which makes it the first choice of DevOps teams.

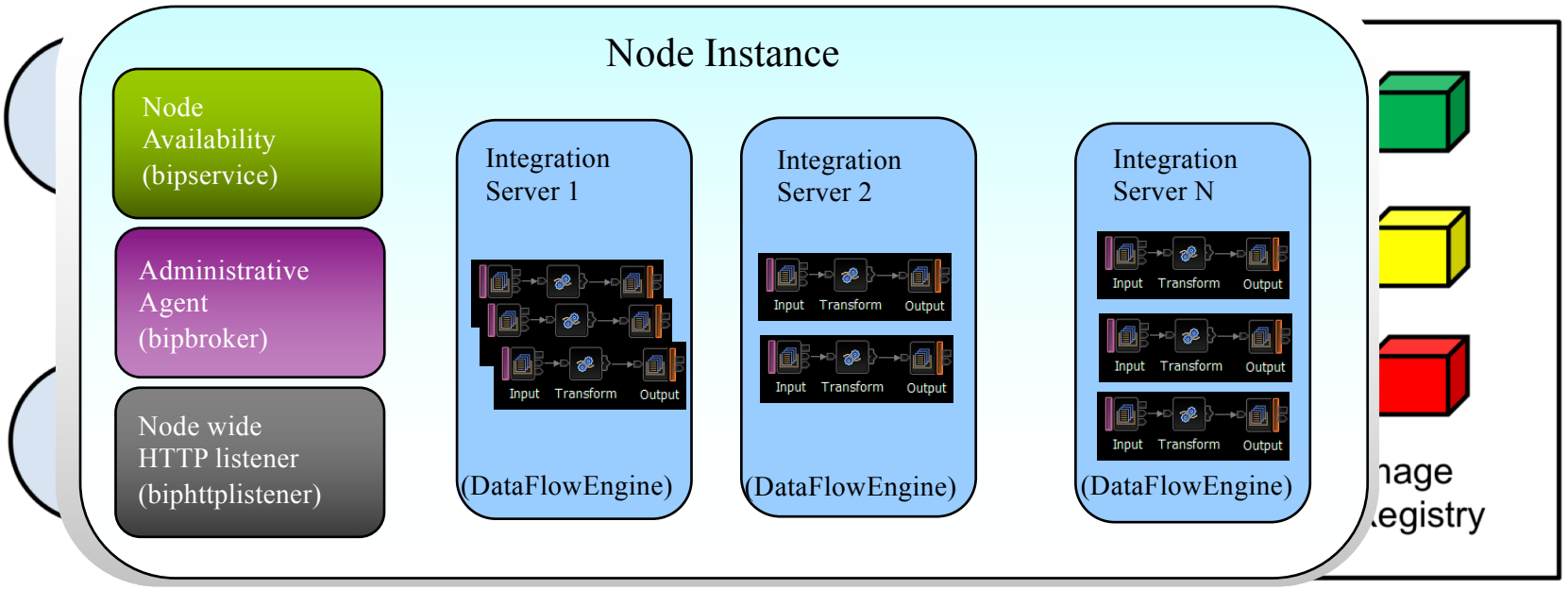
Copyright IBM 2018

Kubernetes Architecture



Copyright IBM 2018

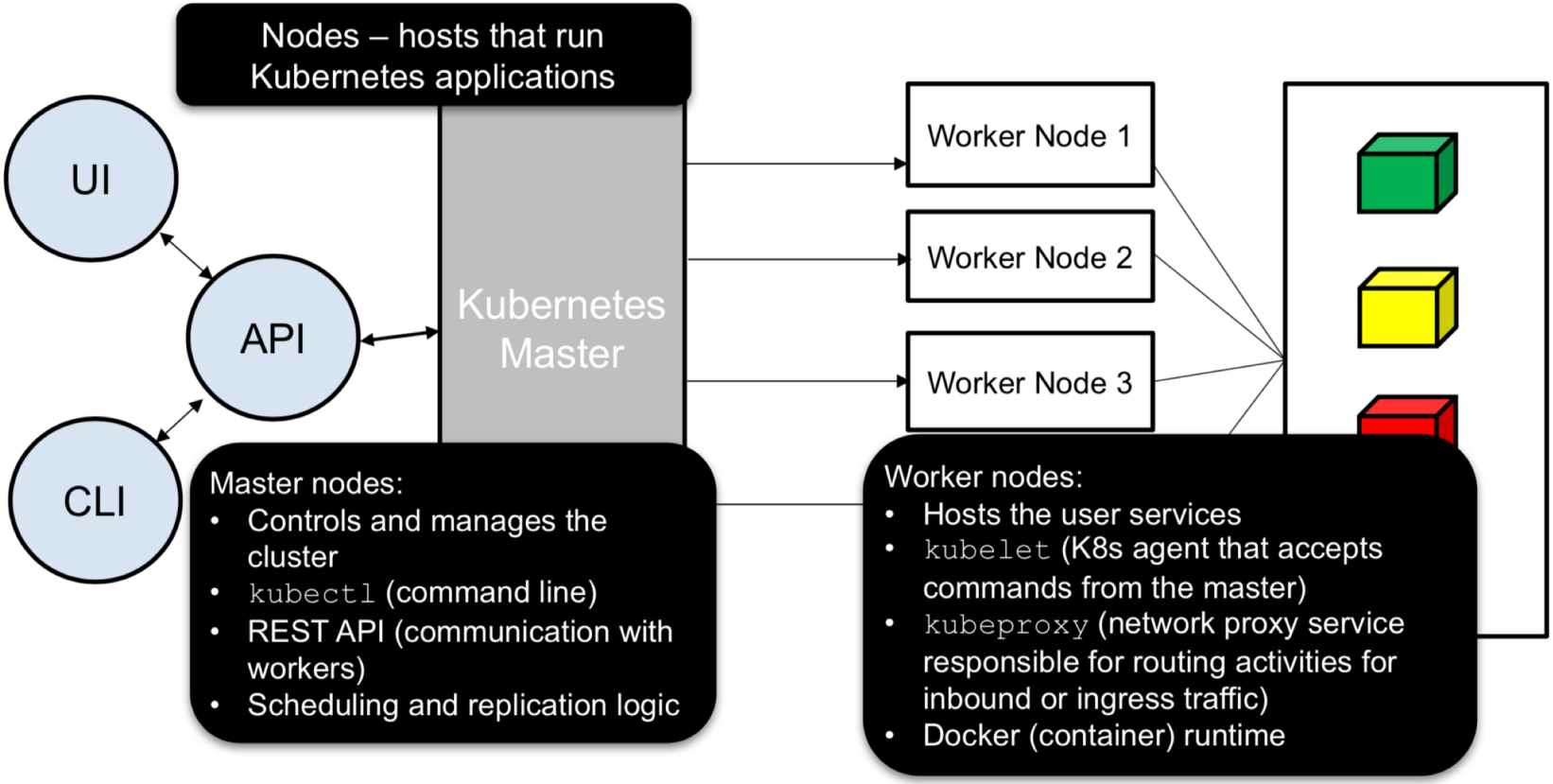
Kubernetes Architecture



- **Controller Manager Server**
- **Scheduler Server**

Copyright IBM 2018

Kubernetes Architecture



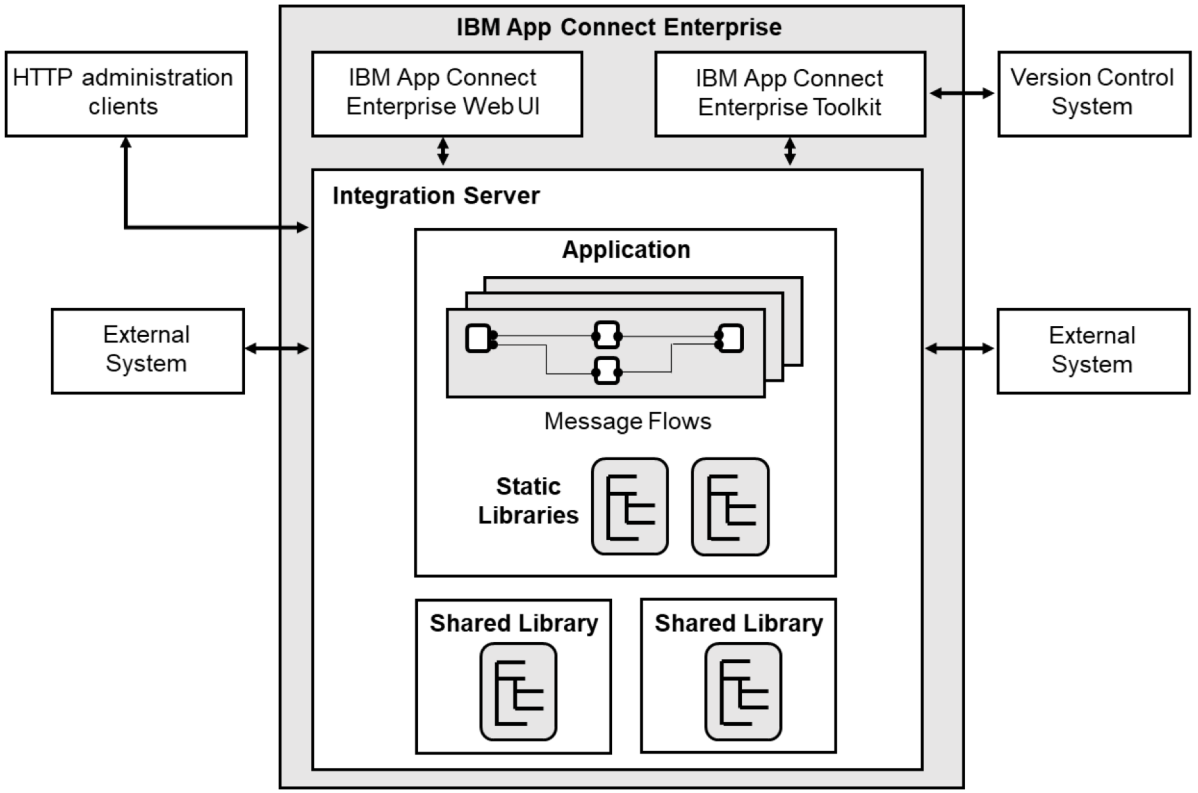
Copyright IBM 2018

#MaximizeYourGame

ACE and Containers

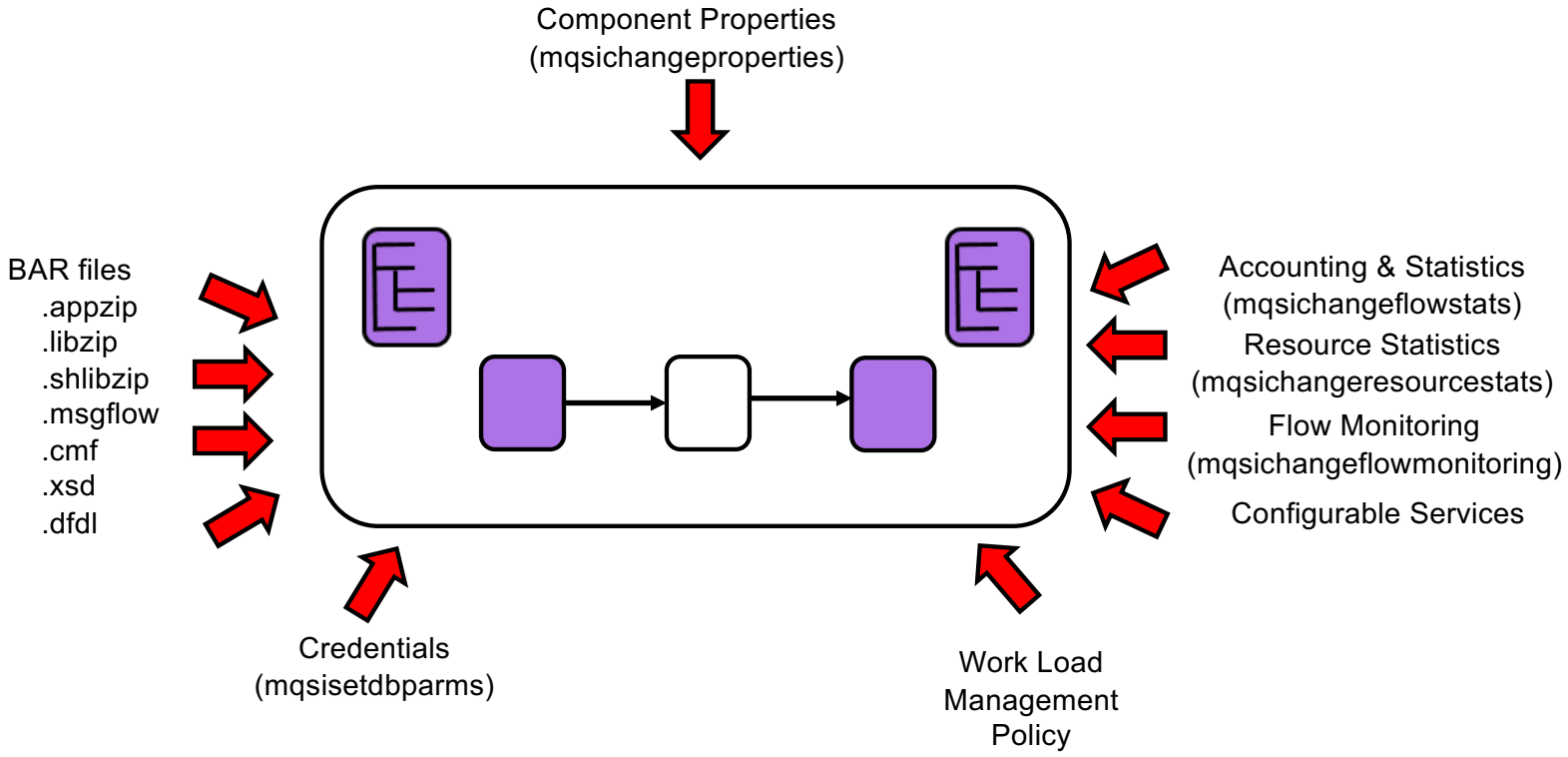
Copyright IBM 2018

App Connect Enterprise – Integration Server Internals



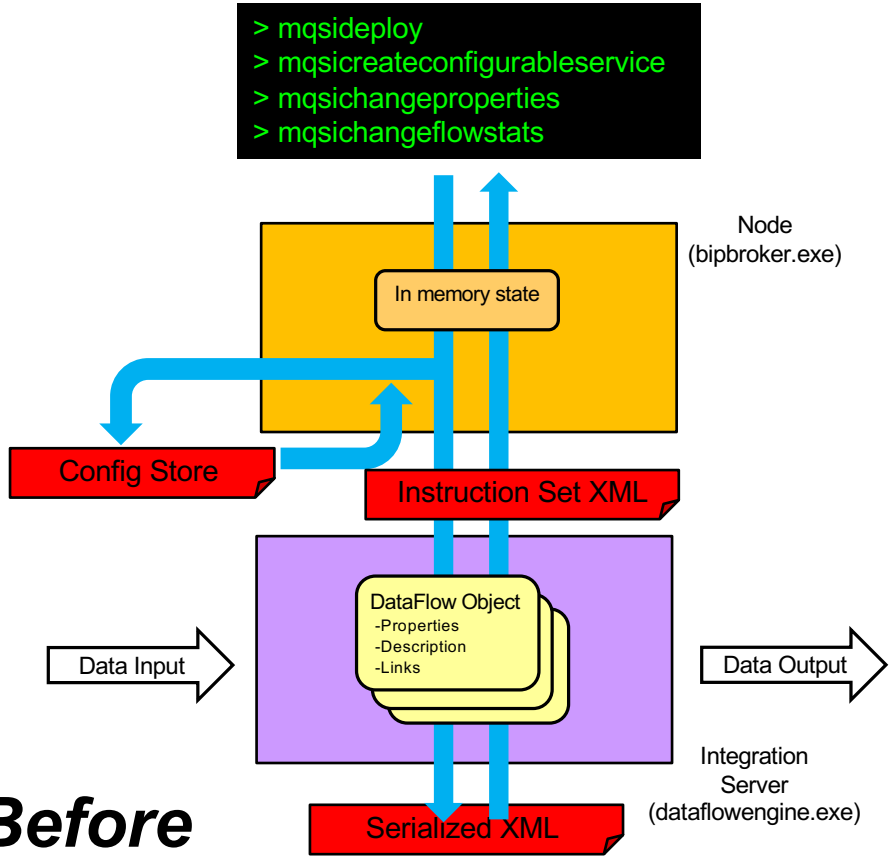
Copyright IBM 2018

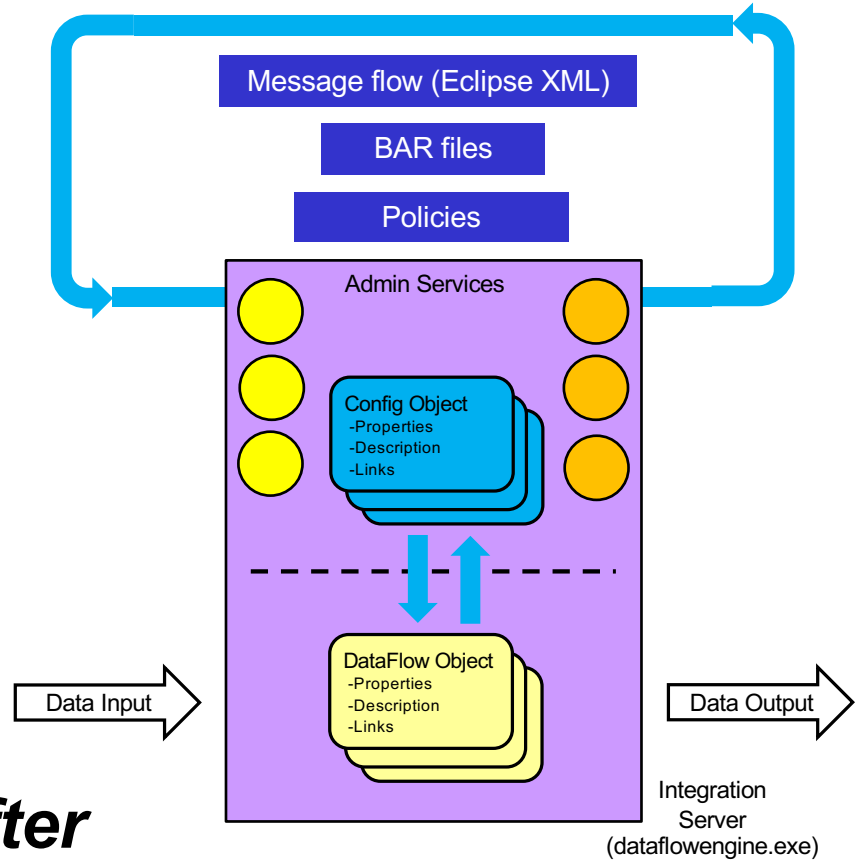
Current Stateful Configuration of an Integration Server



Copyright IBM 2018

Architecture changes in App Connect Enterprise





Copyright IBM 2018

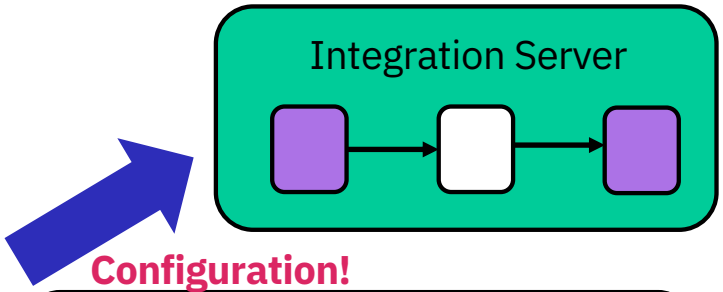
Welcome to your new friend - server.config.yaml

```
---
# ACE server configuration file
#
# General notes :
#   Integration Server will load server.conf.yaml from directory set via --work-dir
#   File paths may be taken as absolute, or relative to the integration server's
work directory

serverConfVersion: 1
adminRestApiPort: 7600
defaultQueueManager: ''
httpConnectorPort: 0
httpsConnectorPort: 0
trace: none # set 1 of : none|service|diagnostic
traceNodeLevel: none
StatsSnapNodeDataLevel: basic # choose 1 of : none|basic|advanced
StatsSnapOutputFormat: "csv,bluemix,usertrace" # comma separated list of :
csv,bluemix,json,xml,usertrace
StatsSnapPublicationOn: false
StatsSnapThreadDataLevel: none # choose 1 of : none|basic
StatsSnapAccountingOrigin: ''
resourceStatsReportingOn: false

UserVariables: # equivalent to providing extra user variables on the command
line for IntegrationServer
  Thing1: value
  Thing2: value

EnvironmentVariables: # equivalent to setting environment variables used
during application source load and run
  Thing1: value
  Thing2: value
```

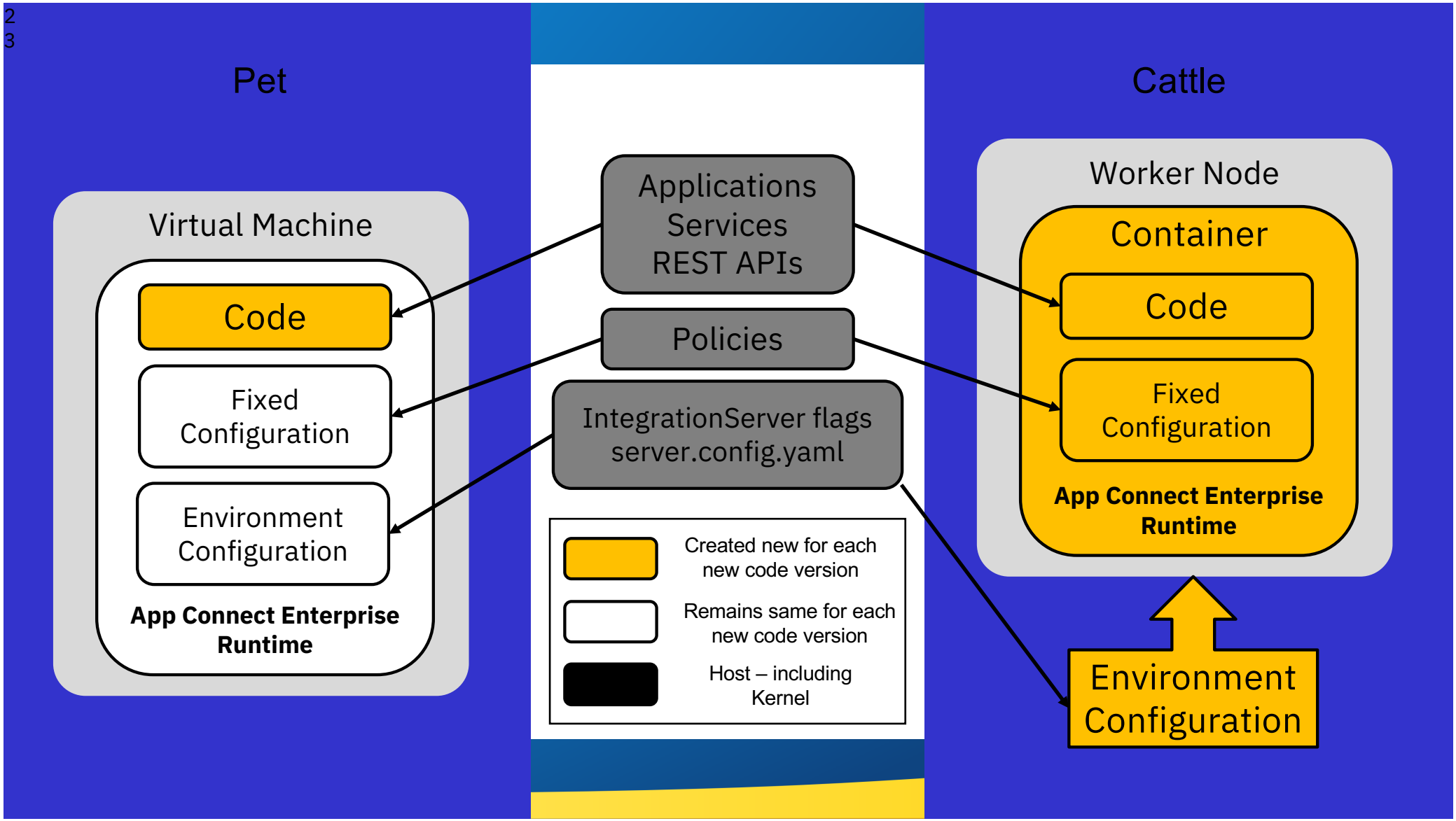


Configuration!

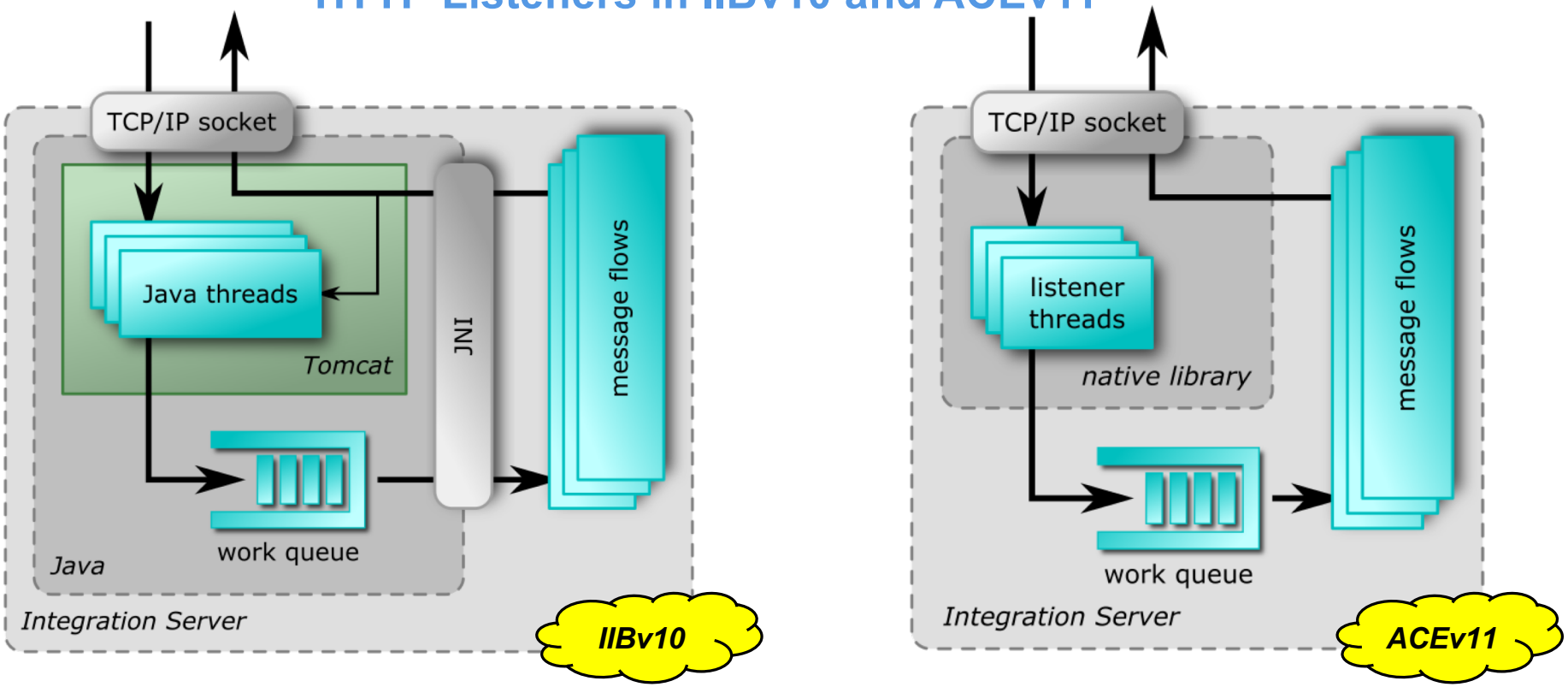
```
IntegrationServer:
   activeUserExitListi: ''
   inactiveUserExitList: ''
   unnamedTraceLevel: none # choose 1 of : none|debugTrace
   unnamedUserTraceLevel: none # choose 1 of :
none|debugTrace
   traceNodeLevel: true
   processorArchitecture: ''
   consoleMode: ''
   httpNodesUseEmbeddedListener: ''
   soapNodesUseEmbeddedListener: ''
   threadLocalProxyNameManagers: ''
   injectionMod: ''
   failedMessageWaitTime: ''
   StatsSnapPublicationOn: ''
   StatsSnapThreadDataLevel: ''
   StatsSnapNodeDataLeve: ''
   StatsSnapOutputFormat: ''
   StatsSnapAccountingOrigin: ''
   resourceStatsReportingOn: ''
```

```
ResourceManagers:
  JVM:
    resourceStatsReportingOn: ''
    jvmVerboseOption: none
    jvmDisableClassGC: ''
    jvmEnableIncGC: ''
    jvmShareClasses: ''
    jvmNativeStackSize: -1
    jvmJavaOSStackSize: -1
    jvmMinHeapSize: 33554432
    jvmMaxHeapSize: 268435456
    jvmDebugPort: 19790
    jvmSystemProperty: ''
    keystoreType: ''
    keystoreFile: ''
    keystorePass: ''
    truststoreType: ''
    truststoreFile: ''
    truststorePass: ''
    crlFileList: ''
    enableCRLDP: ''
    kerberosConfigFile: ''
    kerberosKeytabFile: ‘’
```

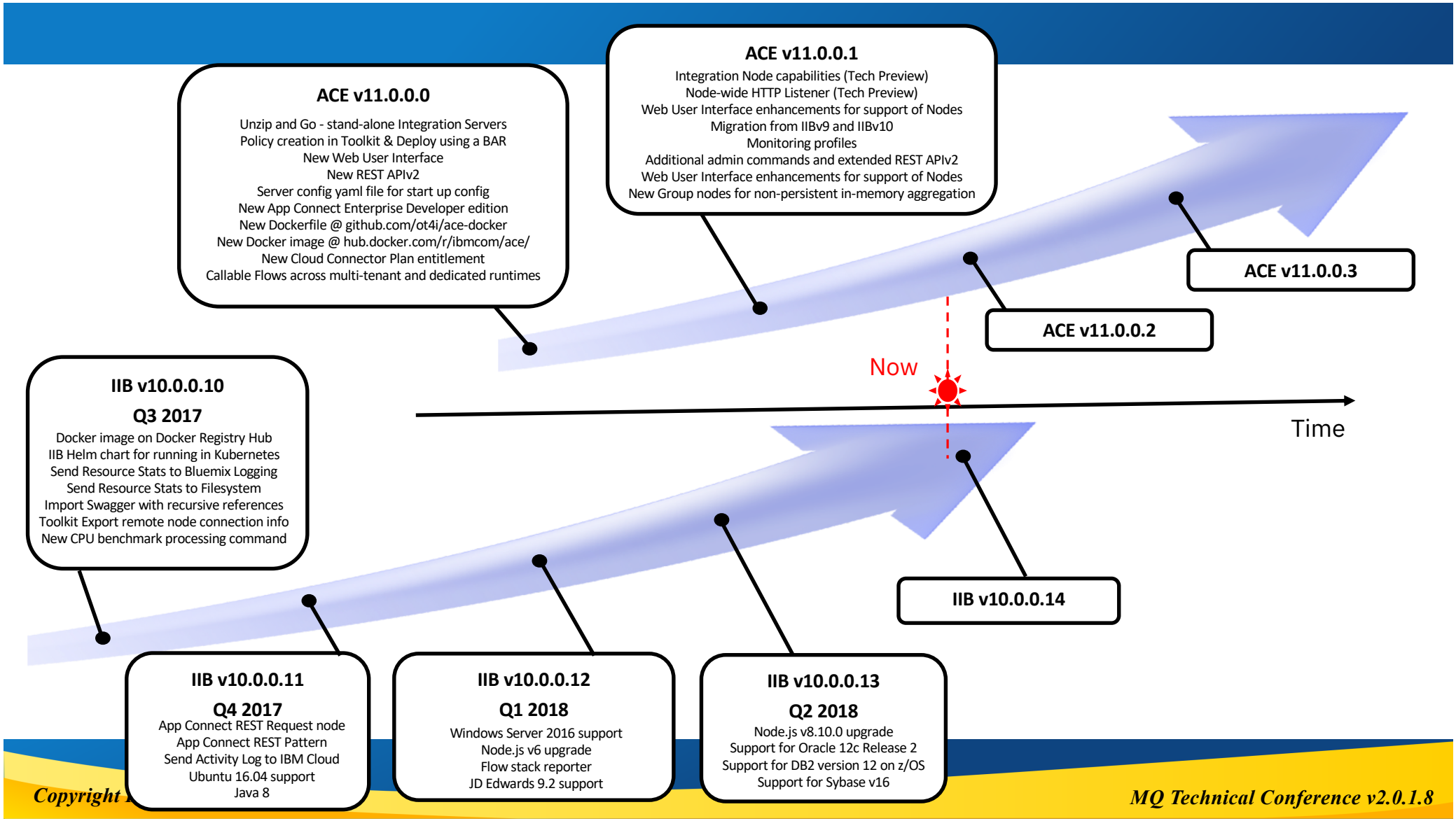
Copyright IBM 2018


Pet
Virtual Machine
Code
Fixed Configuration
Environment Configuration
App Connect Enterprise Runtime
Applications Services REST APIs
Policies
IntegrationServer flags server.config.yaml
Created new for each new code version
Remains same for each new code version
Host – including Kernel
Cattle
Worker Node
Container
Code
Fixed Configuration
App Connect Enterprise Runtime
Environment Configuration

HTTP Listeners in IIBv10 and ACEv11



Copyright IBM 2018


ACE v11.0.0.0
Unzip and Go - stand-alone Integration Servers
Policy creation in Toolkit & Deploy using a BAR
New Web User Interface
New REST APIv2
Server config yaml file for start up config
New App Connect Enterprise Developer edition
New Dockerfile @ github.com/ot4i/ace-docker
New Docker image @ hub.docker.com/r/ibmcom/ace/
New Cloud Connector Plan entitlement
Callable Flows across multi-tenant and dedicated runtimes
ACE v11.0.0.1
Integration Node capabilities (Tech Preview)
Node-wide HTTP Listener (Tech Preview)
Web User Interface enhancements for support of Nodes
Migration from IIBv9 and IIBv10
Monitoring profiles
Additional admin commands and extended REST APIv2
Web User Interface enhancements for support of Nodes
New Group nodes for non-persistent in-memory aggregation
ACE v11.0.0.3
ACE v11.0.0.2
Now
Time
IIB v10.0.0.10
Q3 2017
Docker image on Docker Registry Hub
IIB Helm chart for running in Kubernetes
Send Resource Stats to Bluemix Logging
Send Resource Stats to Filesystem
Import Swagger with recursive references
Toolkit Export remote node connection info
New CPU benchmark processing command
IIB v10.0.0.14
IIB v10.0.0.11
Q4 2017
App Connect REST Request node
App Connect REST Pattern
Send Activity Log to IBM Cloud
Ubuntu 16.04 support
Java 8
IIB v10.0.0.12
Q1 2018
Windows Server 2016 support
Node.js v6 upgrade
Flow stack reporter
JD Edwards 9.2 support
IIB v10.0.0.13
Q2 2018
Node.js v8.10.0 upgrade
Support for Oracle 12c Release 2
Support for DB2 version 12 on z/OS
Support for Sybase v16

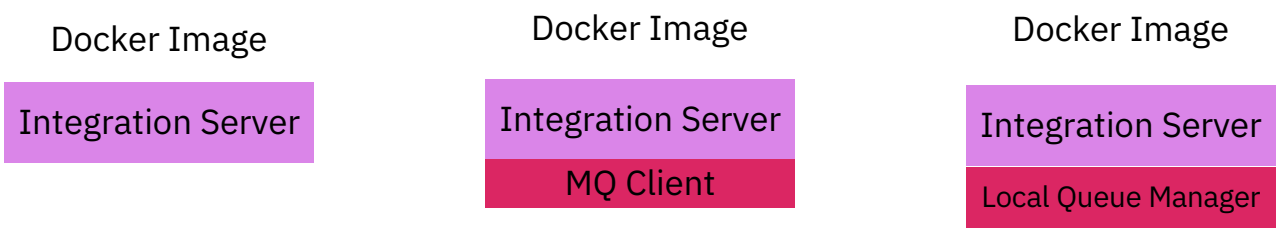
Copyright

#MaximizeYourGame

ACE, Containers and MQ

Copyright IBM 2018

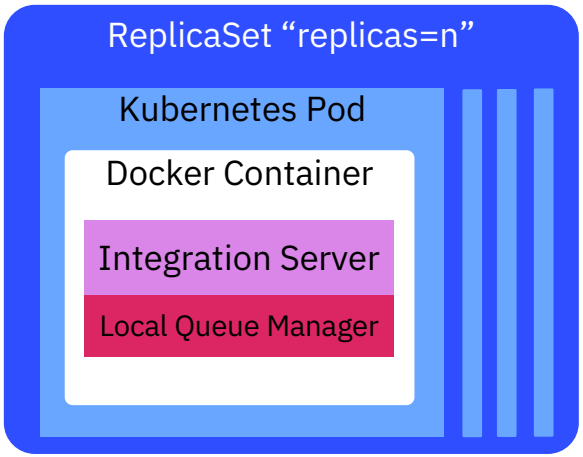
Docker Image Options for ACE and MQ



MQ connectivity	Yes (HTTP API)	Yes (client binding)	Yes (server binding)
1PC for MQ	No	Yes	Yes
EDA nodes	No	No	Yes
2PC	No	No	Yes
Horizontally scalable	Yes	Yes	Yes (with loss of sequencing)
Persistent volume	Not required	Not required	Required (if durability desired)
Start up	Fast	Fast	Slower
Disk space	Smallest	Medium	Largest

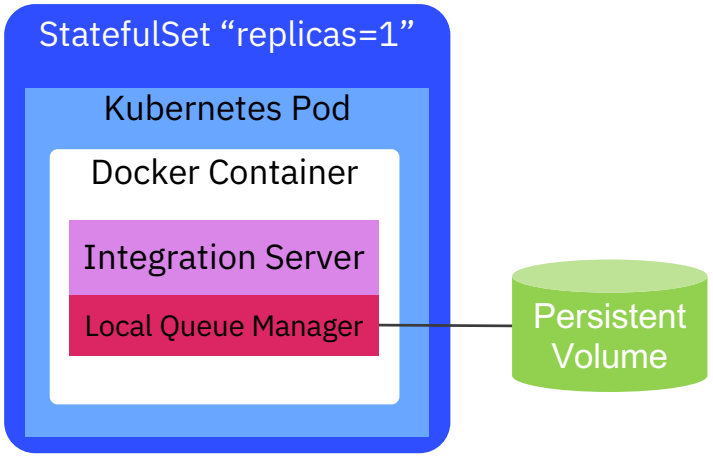
Copyright IBM 2018

Same ACE/MQ image
…but different configuration and usage



replicas = n
no persistent volume claim

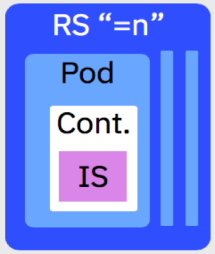
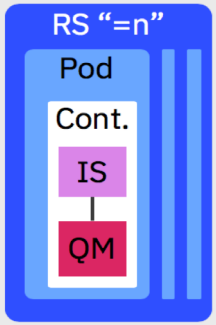
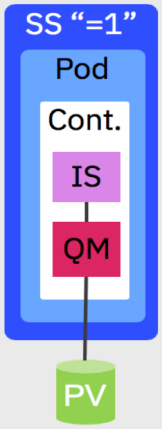
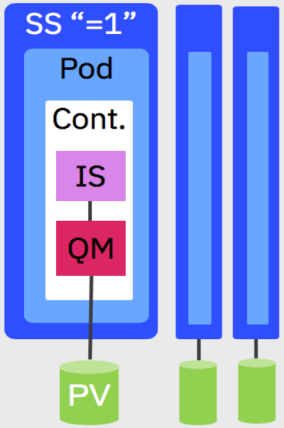
- HA by replication (**continuous** availability)
  - **Elastic** horizontally scalability
  - **Non-durable** use of EDA nodes
    - No 2 Phase Commit



replicas = 1
persistent volume claim

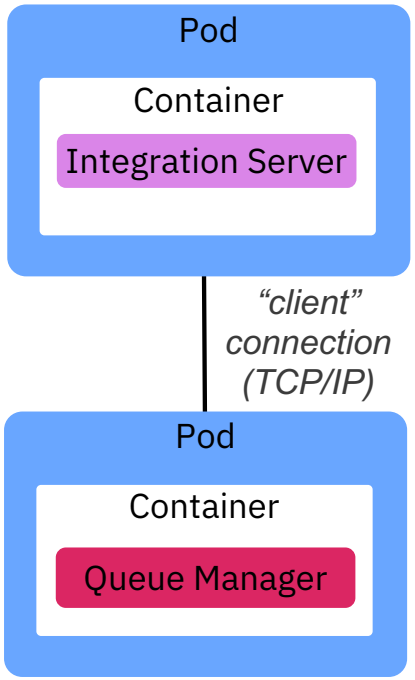
- HA by reinstatement
  - **Manual** horizontal scalability
  - **Durable** use of EDA nodes
    - **2 Phase Commit**

Copyright IBM 2018

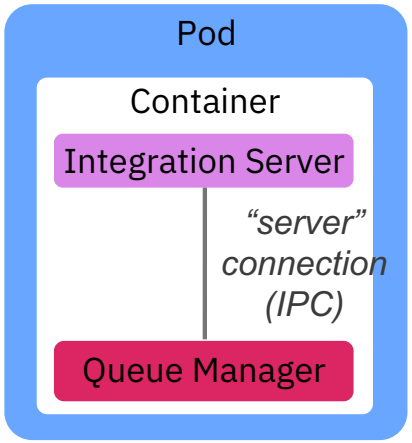
Scalable no MQ Client	Scalable with MQ Client	Scalable, with non-persistent local MQ	Single instance with persistent local MQ	Active/active with persistent independent local MQ	Traditional (with node)
					Shown separately
Stateless	Stateless	Stateless	Stateful	Stateful	Stateful
Continuous availability	Continuous availability	Continuous availability	High Availability	Continuous Service Availability High Message Availability	Continuous Availability
Dynamic horizontal scaling	Dynamic horizontal scaling	Dynamic horizontal scaling	Vertical scaling only	Non-dynamic horizontal scaling	Non-dynamic horizontal scaling
No EDA	No EDA	Enables EDA, but note that must be replica=1 for sequencing	Enables EDA and 2PC	Enables EDA and 2PC	Enables EDA and 2PC

Copyright IBM 2018

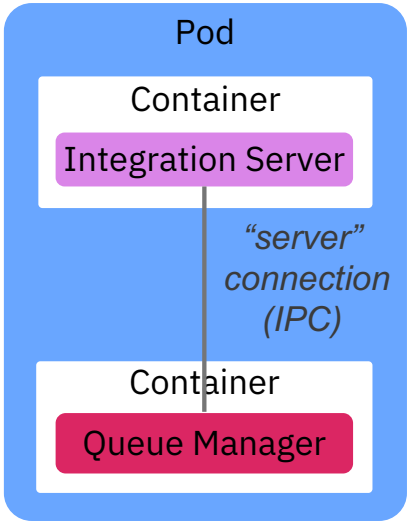
Containers and pods for ACE and MQ



- Standard ACE and MQ containers can be used
- Does not enable event nodes
- Does not enable 2PC



- Requires combined ACE/MQ image – not ideal from container design point of view. Prefer to only have one core purpose/process per container.
- Enables event nodes
- Enables 2PC (MQ, ODBC, JMS, JDBC, CICS)
- Ties topology of ACE and MQ together.



- Standard ACE and MQ images can be used
- Enables event nodes
- Enables 2PC (MQ, ODBC)
- Requires Docker 1.12 or later and Kubernetes 1.10 (alpha feature)

TCP/IP = Network based inter-communication
IPC = Inter Process Communication (via shared memory)

Copyright IBM 2018

The Benefits of servers that don't need a local queue manager

- **SIZE:** The size of the installation is dramatically reduced, and thereby the size of the Docker image. This reduces build times due to the reduced image creation time, and deployment times as a smaller image is transported out to the environments.

- **MEMORY:** The running container-based on the image uses significantly less memory usage as it has no processes associated with the MQ server. Cloud infrastructure used for container-based deployment is often charged based on memory rather than CPU so this can have a significant impact on running cost.

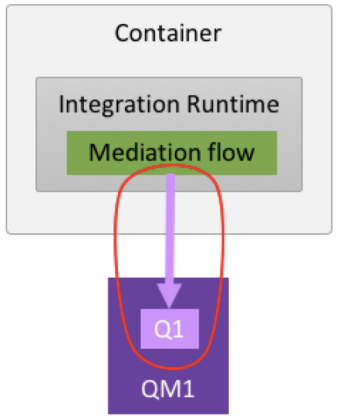
- **START-UP:** Start-up times of the containers are much faster as only one operating system process is started – that of the integration engine. This improves agility by reducing test cycle time, and improves resilience and elastic scalability by being able to introduce new runtimes into a cluster more rapidly.

- **VOLUMES:** MQ holds its message data on persistent volumes, and specific servers need access to specific volumes within the MQ topology. If IBM App Connect Enterprise has a local MQ server, it becomes locked into this topology. This makes it more complex to elastically add new servers to handle demand dynamically. For those using Kubernetes it may result in a StatefulSet rather than the more straightforward ReplicaSet. Once again, this makes it harder to take advantage of the cost benefits of elastic cloud infrastructure.

Copyright IBM 2018

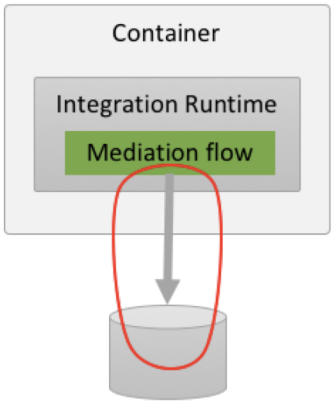
ACE Transactions and Queue Managers (1)

a) Single transaction with an MQ queue



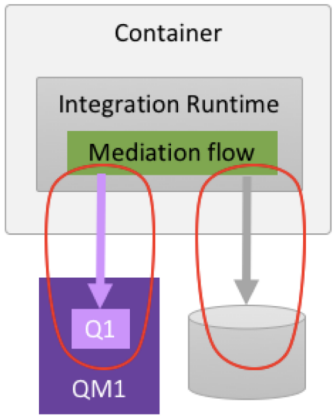
- Single phase commit sufficient
- No local queue manager required

b) Single transaction with a database



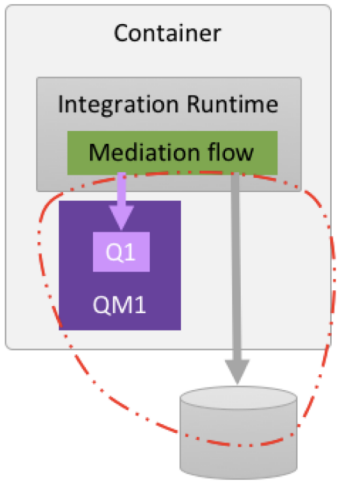
- Single phase commit sufficient
- No local queue manager required

c) Separate transactions to a database and a queue



- Single phase commit sufficient
- No local queue manager required

d) Combined transaction to a database and a queue

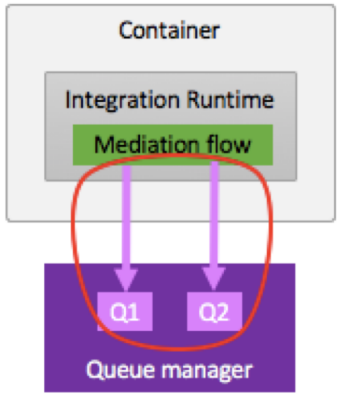


- Two phase commit required
- Local queue manager required to co-ordinate.

Copyright IBM 2018

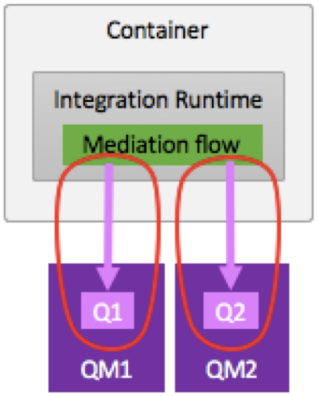
ACE Transactions and Queue Managers (2)

e) One transaction updating two queues in the same queue manager.



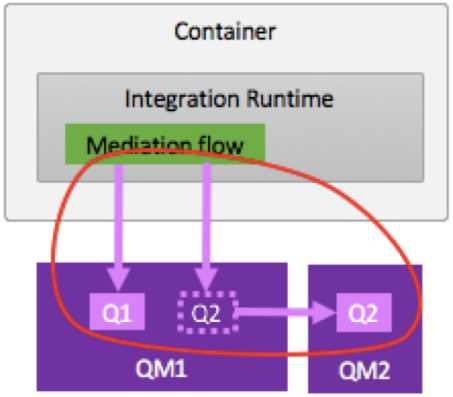
- Single phase commit sufficient
- No local queue manager required

f) Two separate transactions to two queues in different queue managers



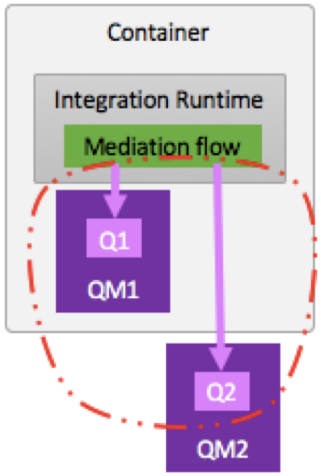
- Single phase commit sufficient
- No local queue manager required

g) An update to two queues appearing on the same queue manager by virtue of a remote queue definition



- Single phase commit sufficient
- No local queue manager required
- Co-ordinated implicitly by QM1

h) A coordinated transaction to two queues in two different queue managers.



- Two phase commit required
- Local queue manager required to co--ordinate

Copyright IBM 2018

In Summary…

Copyright IBM 2018

Summary

- Integration Architectures are Evolving
- ACE V11 is following this Evolution
- Container Orchestration
- ACE and Containers
- ACE and MQ

Copyright IBM 2018

Questions & Answers

Copyright IBM 2018