GUIDE SHARE EUROPE UK REGION

GSE UK Conference 2019
Dock into the Dark Side

Deploying and managing integrations across private and third-party clouds with App Connect Enterprise Containers

Aaron Gashi
IBM

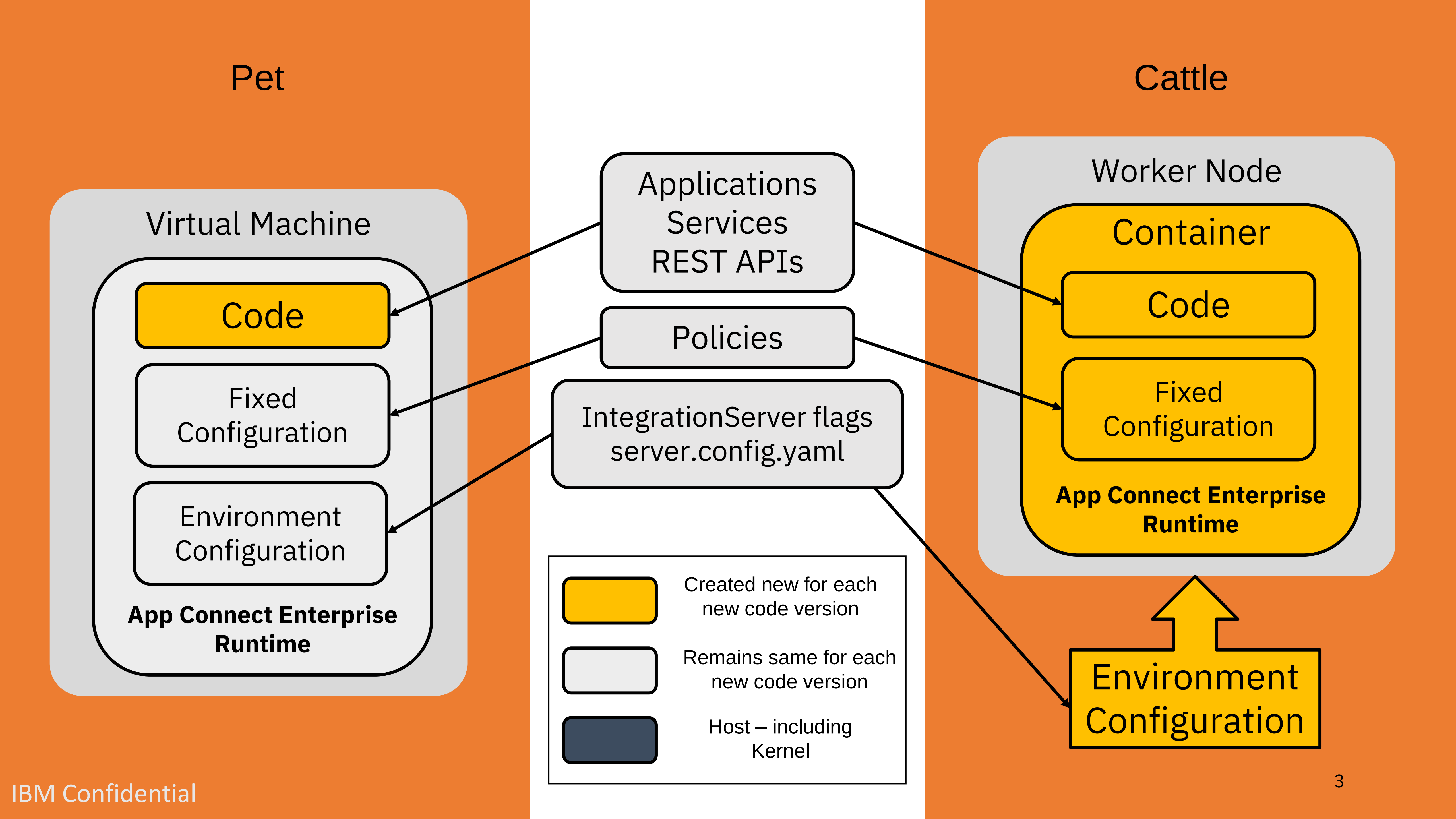
November 2019
Session JG

The 1 chart summary of ACE Certified Containers

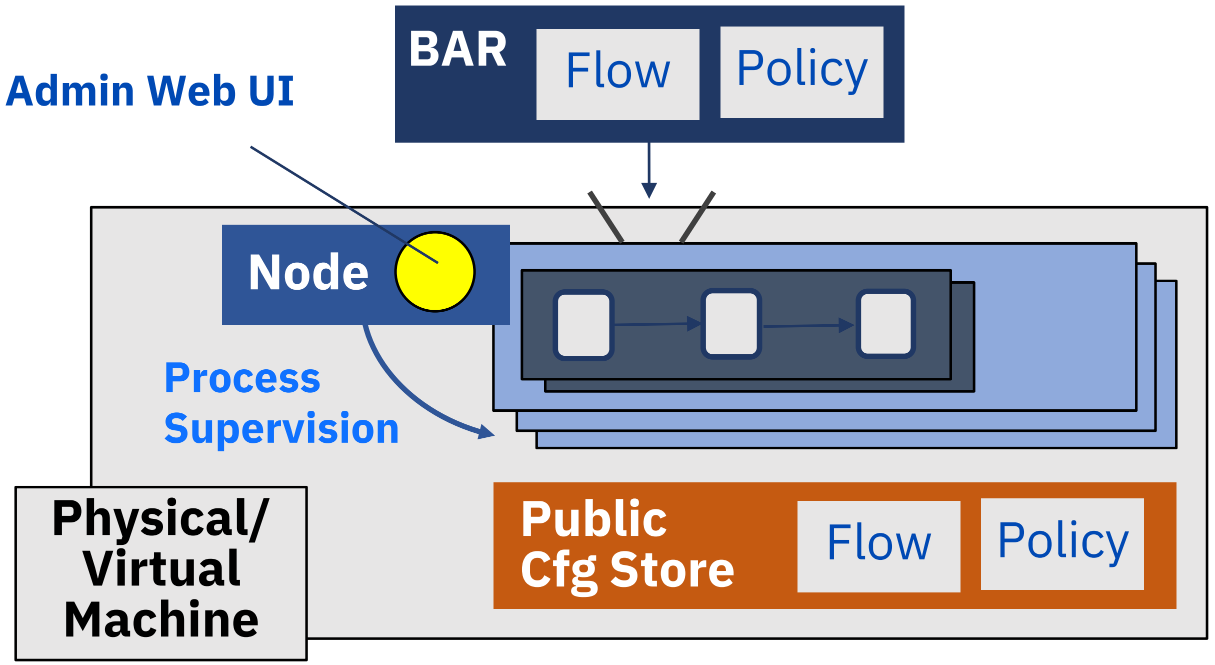
- IBM Cloud Private is a private cloud platform for developing and running workloads locally. Built on top of Kubernetes and Helm technologies as well as providing a catalog, private image repository, management console and monitoring framework.

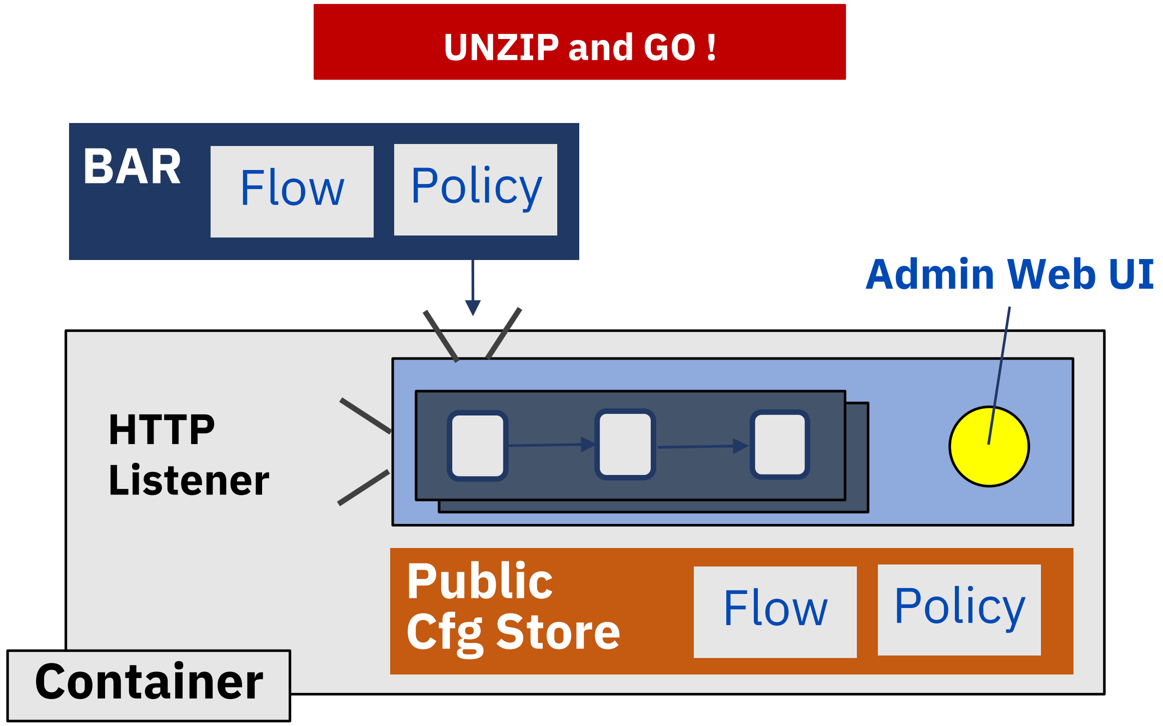
- Docker, Kubernetes and Helm work together to provide a platform for managing, packaging and orchestrating containerised workloads. For App Connect Enterprise this enables the packaging of an integration server into a standardised unit for deployment that can be promoted through a development pipeline then deployed, managed and scaled.

- The ACE Certified Container provides:
  - **Dashboard:** Management of BAR files and integration servers, no need to wait for integration server to start before deploying a BAR file
  - **Monitoring:** Message flow statistics and JVM statistics data provided to the IBM Cloud Private monitoring dashboard
  - **Logging:** Integration with the IBM Cloud Private ELK stack
  - **Configuration:** Greater control of the integration server through configuration


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

Integration Nodes vs. Unzip and go

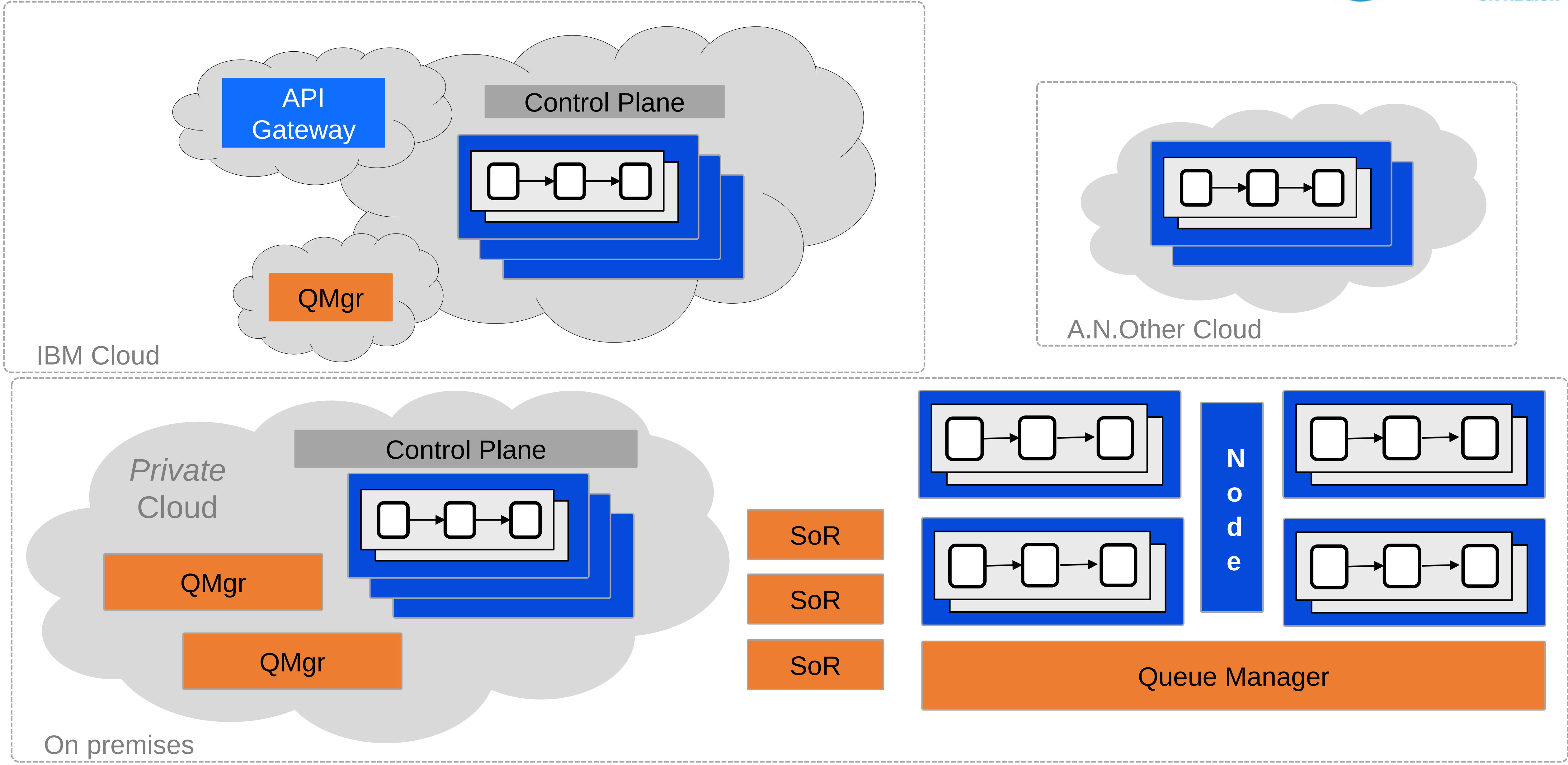




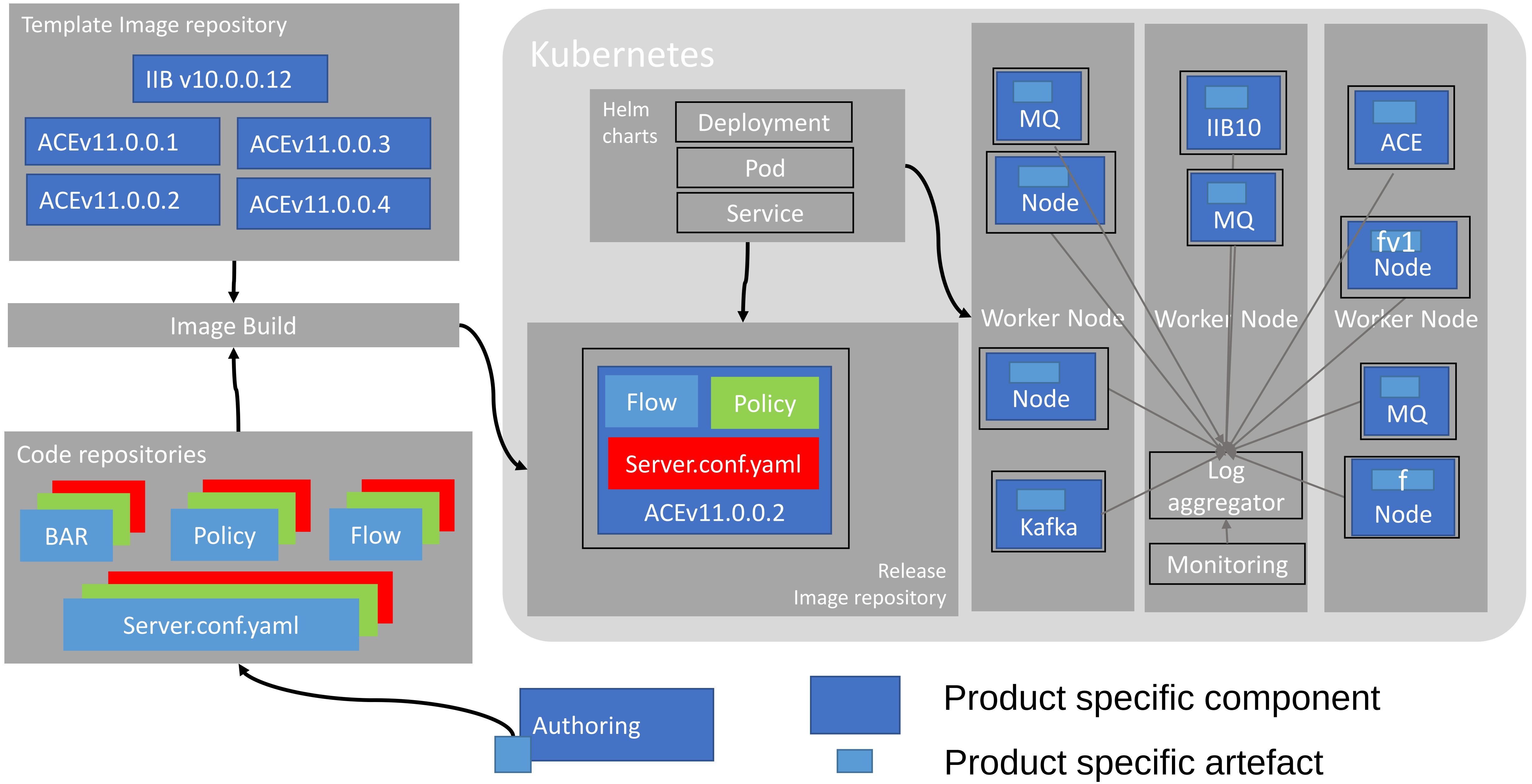
Nodes and their integration servers are long-lived.
Require dynamic operational capability using commands.

Containers can be re-started.
Configuration based on settings in a yaml file.

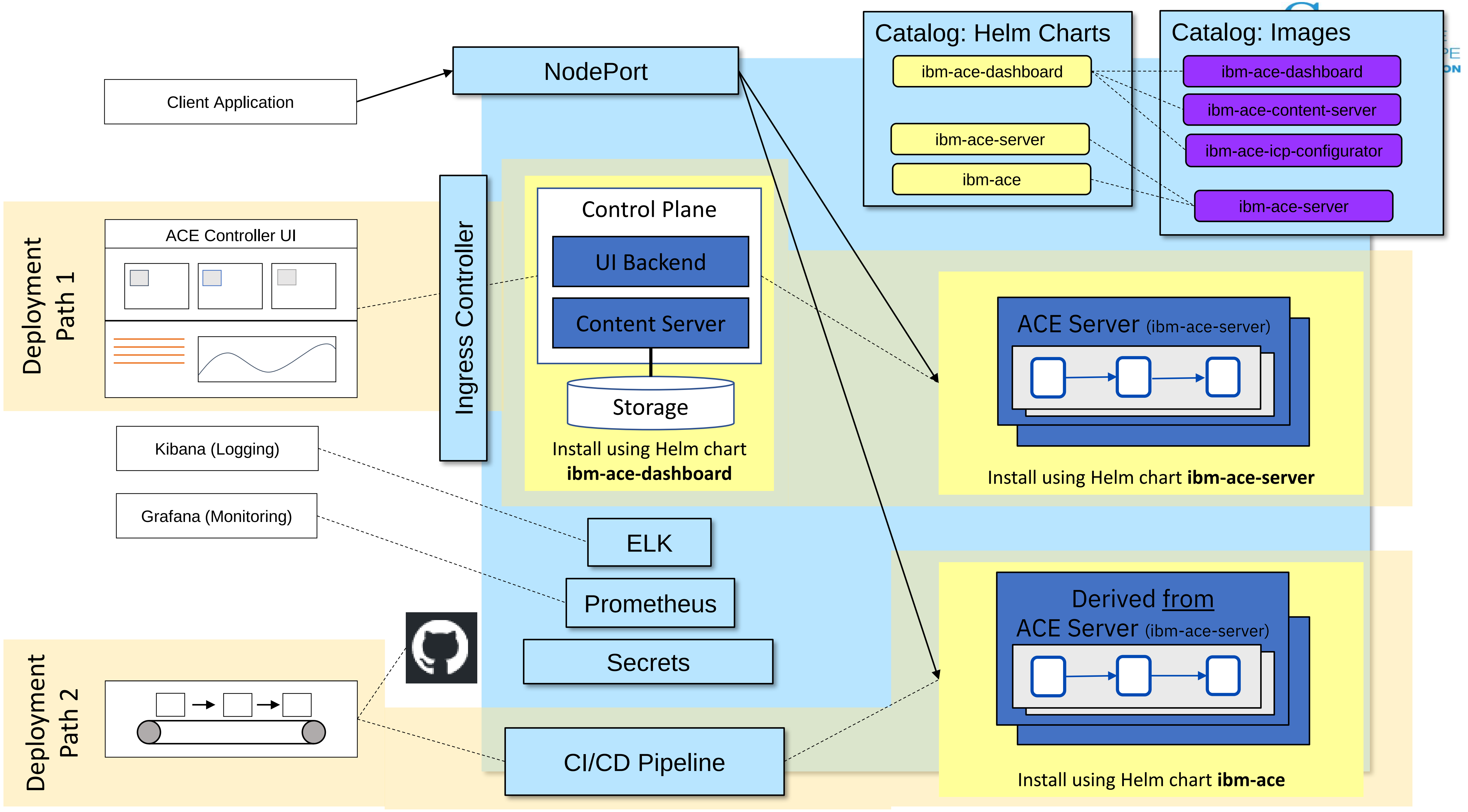
Clouds, Connections and Administrative Control



A Tangible Example



■	Product specific component
▪	Product specific artefact


Catalog: Helm Charts
ibm-ace-dashboard
ibm-ace-server
ibm-ace
Catalog: Images
ibm-ace-dashboard
ibm-ace-content-server
ibm-ace-icp-configurator
ibm-ace-server
Client Application
NodePort
Deployment Path 1
ACE Controller UI
Ingress Controller
Control Plane
UI Backend
Content Server
Storage
Install using Helm chart ibm-ace-dashboard
ACE Server (ibm-ace-server)
Install using Helm chart ibm-ace-server
Kibana (Logging)
Grafana (Monitoring)
ELK
Prometheus
Secrets
Deployment Path 2
CI/CD Pipeline
Derived from ACE Server (ibm-ace-server)
Install using Helm chart ibm-ace

Do you need a local queue manager?
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.

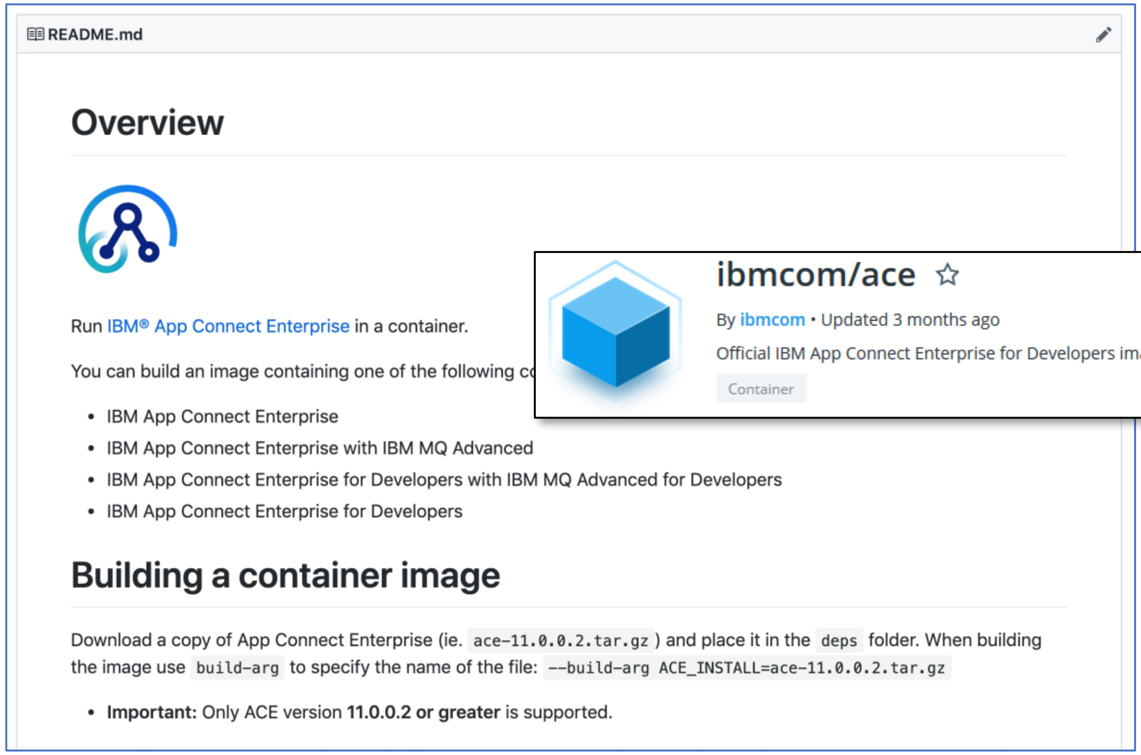
- **VOLUMES:** MQ holds its message data on persistent volumes, and specific servers need access to specific volumes within the MQ topology. If IBM App Connect Enterprise has a local MQ server, it becomes locked into this topology. This makes it more complex to elastically add new servers to handle demand 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.

IBM Certified Containers and IBM Cloud Paks

Container delivery models / Capabilities & Value	Ad hoc containers Client takes software binaries, Creates their own containers	IBM provided containers Client receives IBM Software in the form of container(s)	Certified IBM Cloud Paks on IBM Cloud Private Easy, Enterprise grade, Fully supported
IBM Software supported	Depends on product	Yes	Yes
Full stack support by IBM (Base OS, software, deployment on cloud platform)	No	No	Yes
Vulnerability Scanned (Manages image vulnerabilities)	Scan yourself	Yes	Yes
Orchestrated for Production (Built for Kubernetes by product experts)	None	None	Yes
Management and Operations	Roll your own	Roll your own	Built-in
License Metering Integration	Do it yourself	Do it yourself	Yes
Lifecycle Management	Manage it yourself	Manage it yourself	Tested upgrade & rollback

TRY!

THEN BUY!


README.md

Overview

Run IBM® App Connect Enterprise in a container.

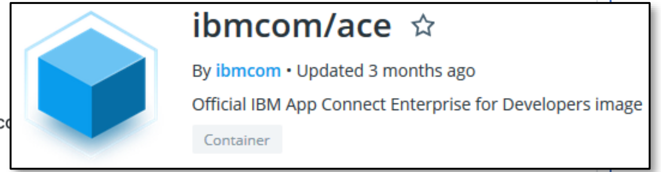
You can build an image containing one of the following c

- IBM App Connect Enterprise
- IBM App Connect Enterprise with IBM MQ Advanced
- IBM App Connect Enterprise for Developers with IBM MQ Advanced for Developers
- IBM App Connect Enterprise for Developers

Building a container image

Download a copy of App Connect Enterprise (ie. `ace-11.0.0.2.tar.gz`) and place it in the `deps` folder. When building the image use `build-arg` to specify the name of the file: `--build-arg ACE_INSTALL=ace-11.0.0.2.tar.gz`

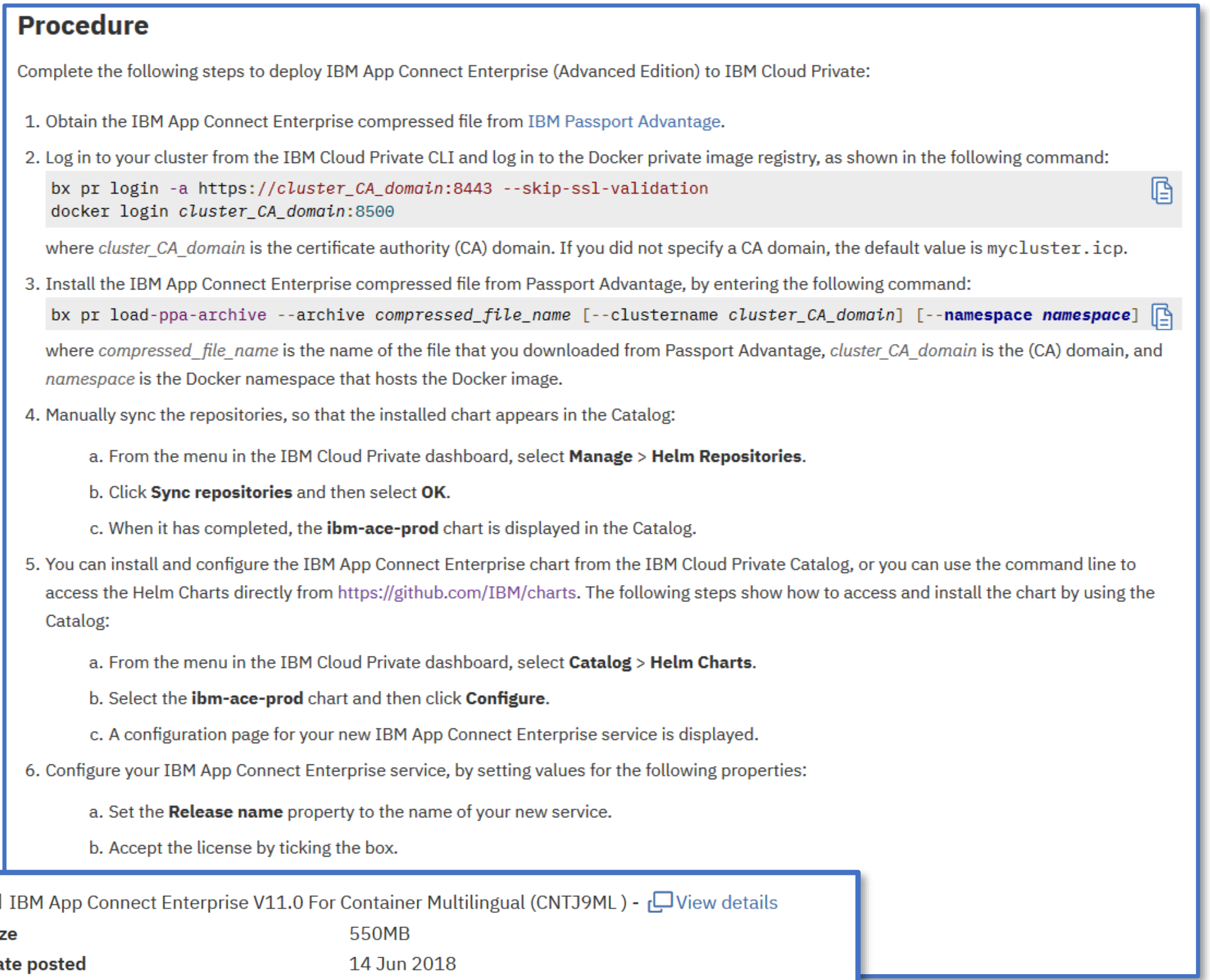
- **Important:** Only ACE version **11.0.0.2 or greater** is supported.


ibmcom/ace ☆

By ibmcom • Updated 3 months ago

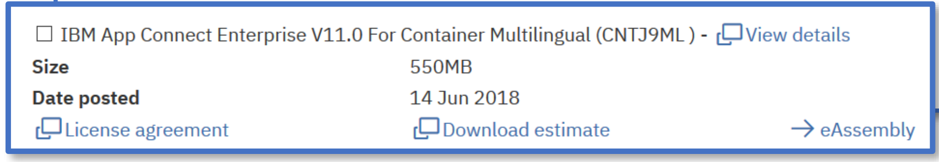
Official IBM App Connect Enterprise for Developers image

Container


Procedure

Complete the following steps to deploy IBM App Connect Enterprise (Advanced Edition) to IBM Cloud Private:

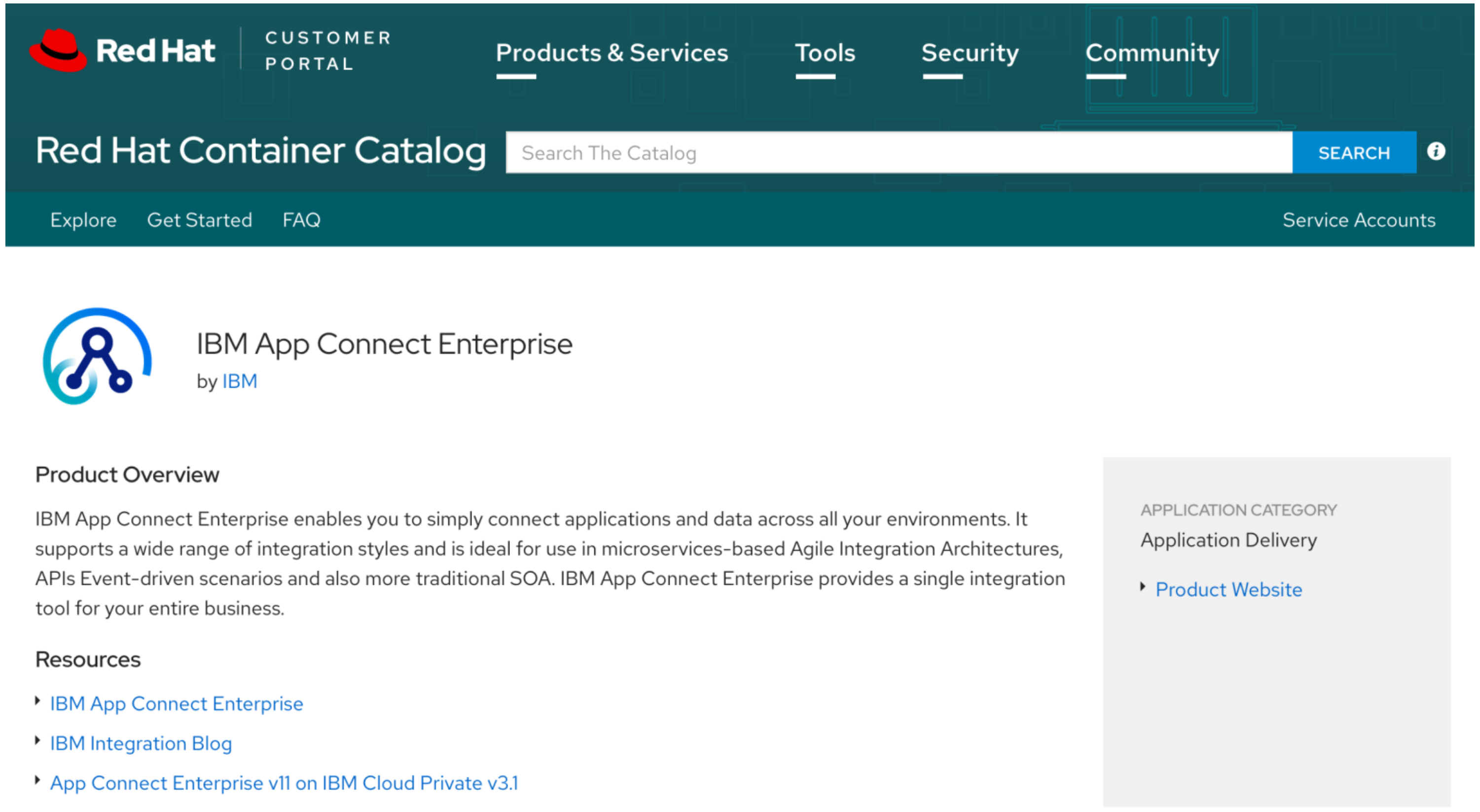
1. Obtain the IBM App Connect Enterprise compressed file from IBM Passport Advantage.
2. Log in to your cluster from the IBM Cloud Private CLI and log in to the Docker private image registry, as shown in the following command:
```
bx pr login -a https://cluster_CA_domain:8443 --skip-ssl-validation
docker login cluster_CA_domain:8500
```
   where *cluster_CA_domain* is the certificate authority (CA) domain. If you did not specify a CA domain, the default value is `mycluster.icp`.
3. Install the IBM App Connect Enterprise compressed file from Passport Advantage, by entering the following command:
```
bx pr load-ppa-archive --archive compressed_file_name [--clustername cluster_CA_domain] [--namespace namespace]
```
   where *compressed_file_name* is the name of the file that you downloaded from Passport Advantage, *cluster_CA_domain* is the (CA) domain, and *namespace* is the Docker namespace that hosts the Docker image.
4. Manually sync the repositories, so that the installed chart appears in the Catalog:
   a. From the menu in the IBM Cloud Private dashboard, select **Manage** > **Helm Repositories**.
   b. Click **Sync repositories** and then select **OK**.
   c. When it has completed, the **ibm-ace-prod** chart is displayed in the Catalog.
5. You can install and configure the IBM App Connect Enterprise chart from the IBM Cloud Private Catalog, or you can use the command line to access the Helm Charts directly from https://github.com/IBM/charts. The following steps show how to access and install the chart by using the Catalog:
   a. From the menu in the IBM Cloud Private dashboard, select **Catalog** > **Helm Charts**.
   b. Select the **ibm-ace-prod** chart and then click **Configure**.
   c. A configuration page for your new IBM App Connect Enterprise service is displayed.
6. Configure your IBM App Connect Enterprise service, by setting values for the following properties:
   a. Set the **Release name** property to the name of your new service.
   b. Accept the license by ticking the box.


☐ IBM App Connect Enterprise V11.0 For Container Multilingual (CNTJ9ML) - View details

Size	550MB
Date posted	14 Jun 2018

License agreement Download estimate → eAssembly

ACE and Red Hat OpenShift



IBM App Connect Enterprise
by IBM

Product Overview

IBM App Connect Enterprise enables you to simply connect applications and data across all your environments. It supports a wide range of integration styles and is ideal for use in microservices-based Agile Integration Architectures, APIs Event-driven scenarios and also more traditional SOA. IBM App Connect Enterprise provides a single integration tool for your entire business.

Resources

- IBM App Connect Enterprise
- IBM Integration Blog
- App Connect Enterprise v11 on IBM Cloud Private v3.1

APPLICATION CATEGORY
Application Delivery

- Product Website

https://access.redhat.com/containers/#/product/b9b058da22f02c21

July 2019
- Docker images updated to include ACEv11.0.0.5 and support IBM Cloud Private 3.2.0.1906
- Docker images now based on Red Hat Universal Base Image (UBI)

May 2019
- Docker images updated to include ACEv11.0.0.4 and support IBM Cloud Private 3.1.2

April 2019
- ACEv11 Docker images (RHEL) Part ID: CC0FNML. Red Hat Certified and available in the Red Hat Container Catalog

Jan 2019
- Docker images updated to include ACEv11.0.0.3

Dec 2018
- Instructions for building your own ACEv11 for IBM Cloud Private on Red Hat OpenShift
- Instructions for building your own ACEv11 on Red Hat OpenShift

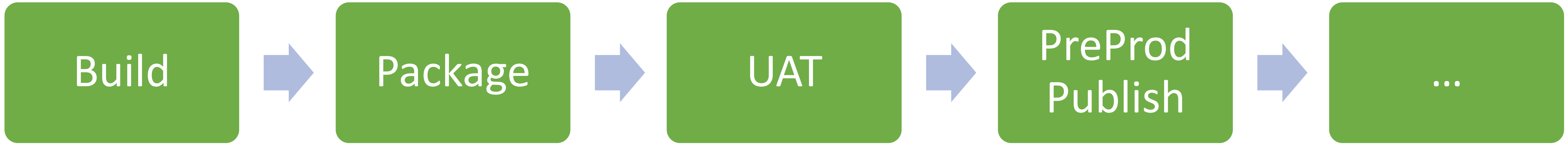
Nov 2018
- ACEv11 Docker images (Ubuntu) and Helm charts for use on IBM Cloud Private 3.1

App Connect Enterprise Certified Containers links:

- Developer Helm Charts:
  - https://github.ibm.com/IBMPrivateCloud/charts/tree/master/stable/ibm-ace-dashboard-dev
  - https://github.ibm.com/IBMPrivateCloud/charts/tree/master/stable/ibm-ace-server-dev
- Server Images source
  - https://github.com/ot4i/ace-docker
- Images on Dockerhub (Developer Edition)
  - ibmcom/**ace**
  - ibmcom/**ace-mq**
  - ibmcom/**ace-mqclient**
  - ibmcom/**ace-dashboard**
  - ibmcom/**ace-content-server**
  - ibmcom/**ace-icp-configurator**
  - Ibmcom/**ace-designer-flows**

CI/CD Pipelines and ACE v11

Pipeline overview

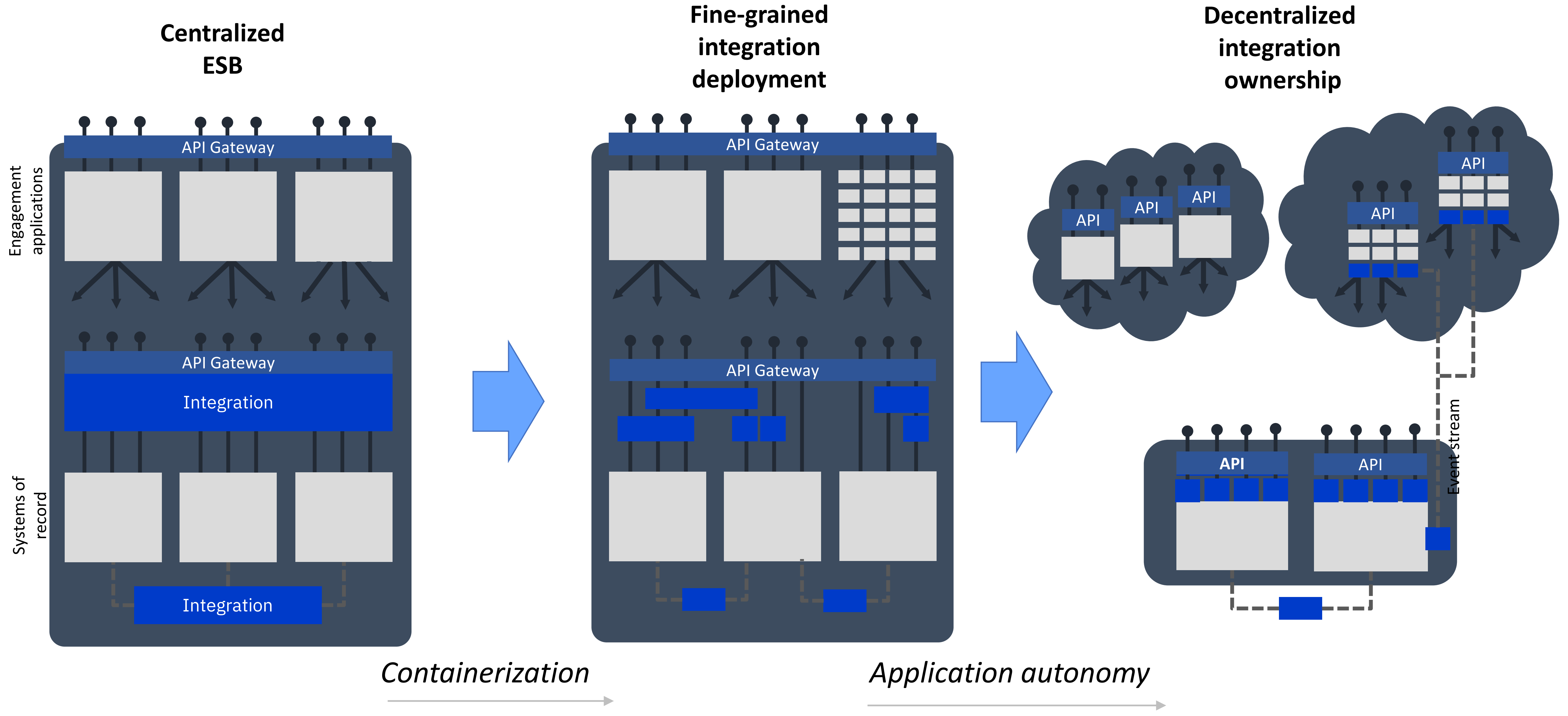


- Pipelines of one sort or another exist in most IT shops
  - Some more automated than others!
- Manual trigger or automatic
- Inputs could be source or whole BAR files
- Destination: static integration node? HA node? Container?

CI and CD

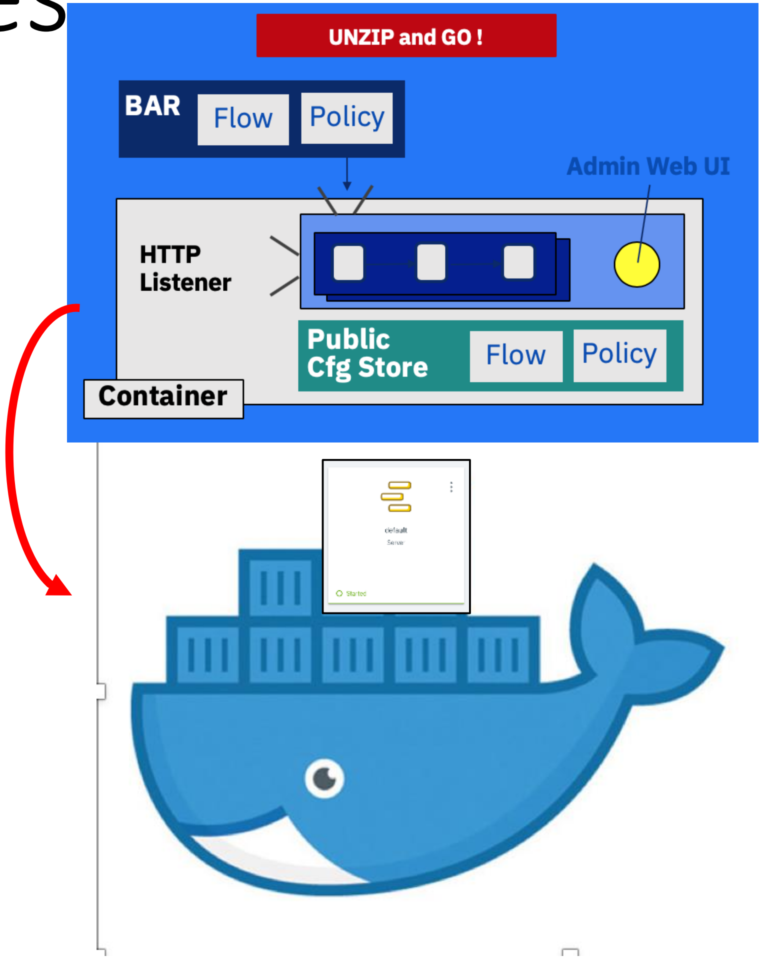
- Continuous Integration
  - Automatically build and test whenever new code is available
  - Tests show whether or not the code has broken anything
- Continuous Delivery
  - Pushing to production (delivering to customers) from a continuous stream
  - No waiting for fixed release cycles
  - Once code is decided to be ready, it is pushed to production
- Continuous Deployment (extension)
  - Automated production pushes – no explicit decision by humans
  - Only a failed test stops the code going live

Evolution to **agile integration architecture** – detail view



Jenkins, Docker, and Kubernetes

- Jenkins for builds
- Docker for application isolation and filesystem layering
- Kubernetes for deploying and managing the results.



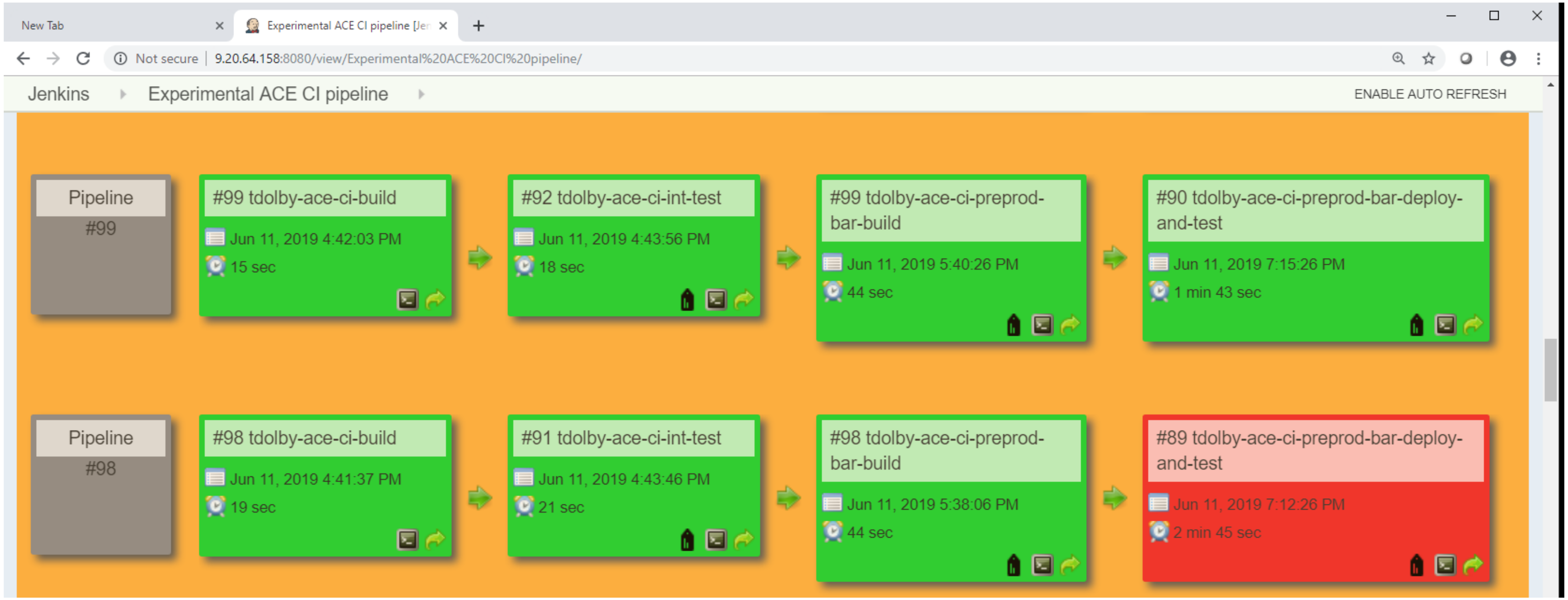
ACE v11 Docker layers

- Install in one layer, based on underlying operating system
- App in small layer at the end, minimizing per-application overhead
  - Huge saving when multiple applications deployed
  - Install is shared.
- Credentials usually added in at runtime
- Test new fixpacks by changing the FROM line

ACE v11 in a pipeline today

- Pipeline structures often (usually) involve multiple products
  - Might need to create MQ queues, database definitions, etc
- mqsicreatebar, mqsipackagebar, mqsibar
- Policy file creation

Demo ACE v11 pipeline using Jenkins



New Tab | Experimental ACE CI pipeline [Jen

Not secure | 9.20.64.158:8080/view/Experimental%20ACE%20CI%20pipeline/

Jenkins ▸ Experimental ACE CI pipeline ▸ ENABLE AUTO REFRESH

Pipeline	Build	Int test	Preprod bar build	Preprod bar deploy and test
Pipeline #99	#99 tdolby-ace-ci-build Jun 11, 2019 4:42:03 PM 15 sec	#92 tdolby-ace-ci-int-test Jun 11, 2019 4:43:56 PM 18 sec	#99 tdolby-ace-ci-preprod-bar-build Jun 11, 2019 5:40:26 PM 44 sec	#90 tdolby-ace-ci-preprod-bar-deploy-and-test Jun 11, 2019 7:15:26 PM 1 min 43 sec
Pipeline #98	#98 tdolby-ace-ci-build Jun 11, 2019 4:41:37 PM 19 sec	#91 tdolby-ace-ci-int-test Jun 11, 2019 4:43:46 PM 21 sec	#98 tdolby-ace-ci-preprod-bar-build Jun 11, 2019 5:38:06 PM 44 sec	#89 tdolby-ace-ci-preprod-bar-deploy-and-test Jun 11, 2019 7:12:26 PM 2 min 45 sec

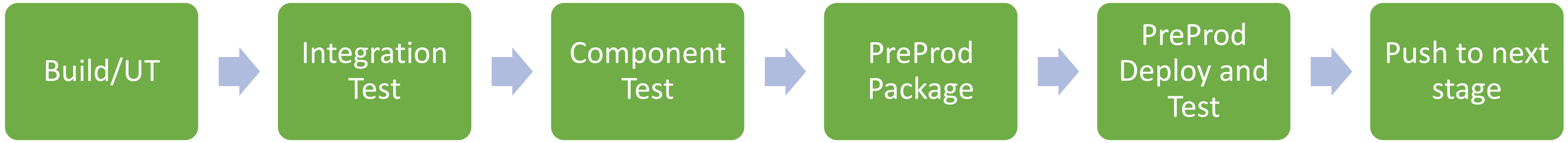
AKS and Windows Containers

- The “unzip and go” architecture does not restrict ACE v11 to Linux
  - Underlying implementation is platform-neutral
- Azure offers Windows Container support, and ACE v11 runs in Windows Containers (Server 2016 and 2019)
  - .Net nodes allow CLR support
- Windows Container support not limited to AKS
- IBM MQ does not yet support Window Containers

ACE v11 versus IIB v10

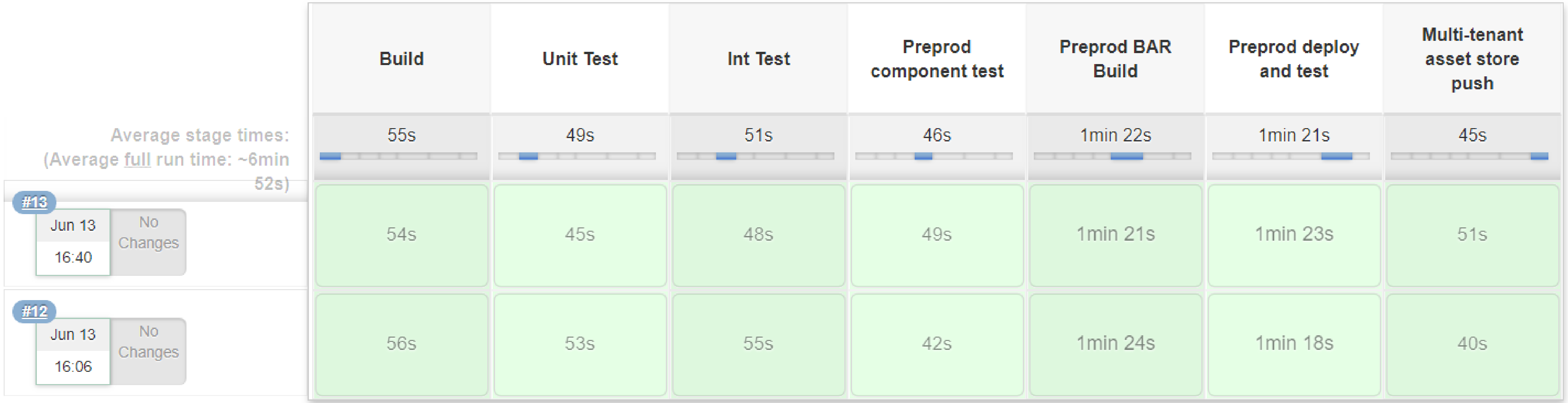
- ACE v11 allows policy, unzip and go, etc
- IIB v10 requires the use of integration nodes
    - Commands needed for some admin operations
    - More processes
    - Slower startup
- IIBoC used IIB v10 codebase in cloud containers, but not in a way that was supported for customers
    - Complicated and fragile

CI/CD Pipelines with testing



- Automated testing rather than manual
  - Repeatable
- Note that the final deploy target of this pipeline could be integration nodes or containers
  - Testing in the early stages unrelated to the final target.

ACE v11 pipeline with more testing (future)



	Build	Unit Test	Int Test	Preprod component test	Preprod BAR Build	Preprod deploy and test	Multi-tenant asset store push
Average stage times: (Average full run time: ~6min 52s)	55s	49s	51s	46s	1min 22s	1min 21s	45s
#13 Jun 13 16:40 No Changes	54s	45s	48s	49s	1min 21s	1min 23s	51s
#12 Jun 13 16:06 No Changes	56s	53s	55s	42s	1min 24s	1min 18s	40s

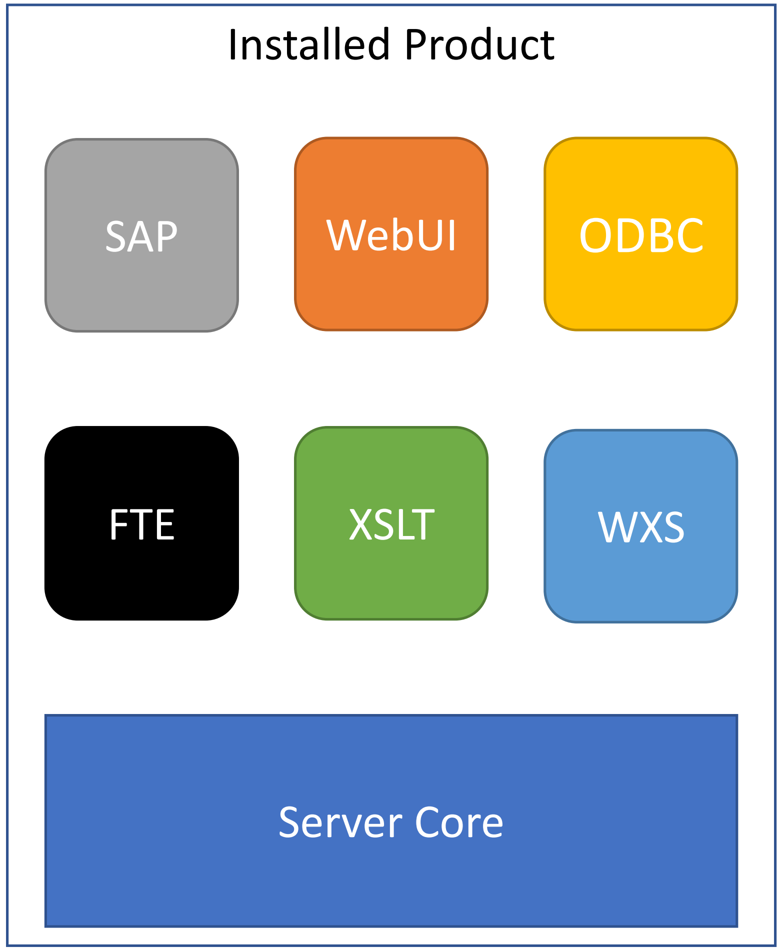
- Unit test and component test used internally already
- TDD before check-in of flows, maps, etc

Lightweight integration engine?

- ACE v11 disk footprint
  - 1.15GB with Ubuntu 18.04 but without the toolkit
- ACE v11 runtime memory consumption
  - Server itself fits in all but the smallest containers on public clouds
  - Applications may require much more
- Servers run threads and maintain in-memory configuration for all available applications and flows

Componentized ACE v11 runtime

- Not all solutions require the complete ACE v11 runtime package.
  - Web UI not always needed in containers
  - Adapters not always used
  - JRE needed, but not always JDK
  - …
- Create a container image with only the required components.
  - Saves space and memory consumption
- Work in progress!

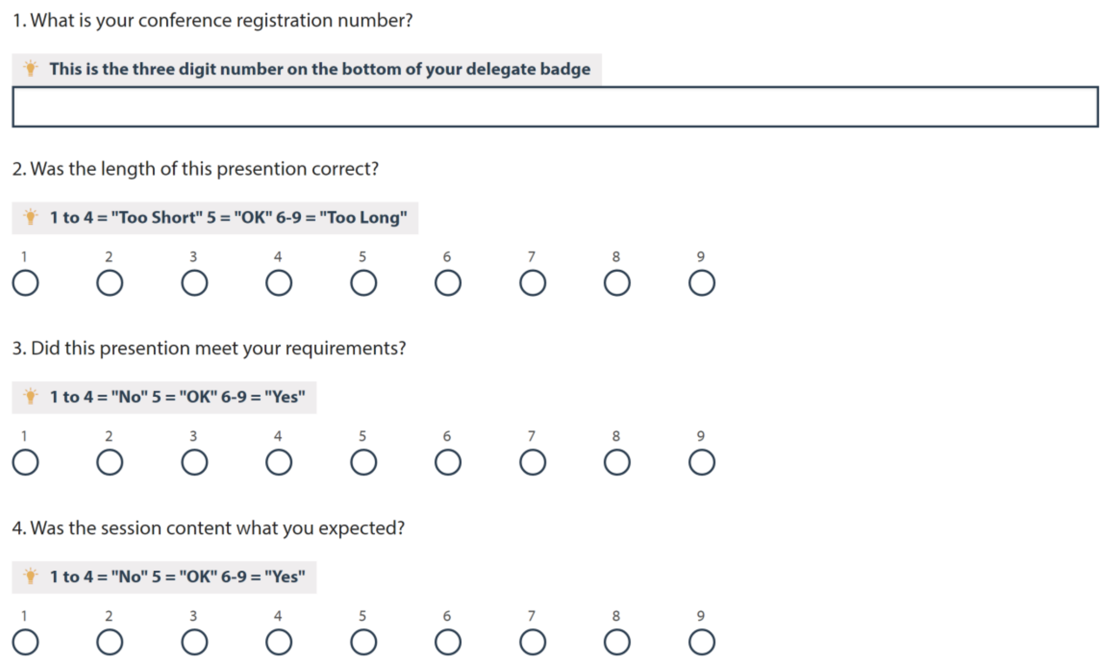


ACE v11 using Alpine base image

- Alpine docker base image designed for lightweight containers
  - Much smaller than Ubuntu or RedHat UBI
  - Does not use GNU C or C++ runtime libraries due to size constraints
- Used by IBM Java and OpenJDK for base images
- Lab will demonstrate size advantage when used in conjunction with customised runtime components.
- Fits in the free tier of IBM Container Registry (512MB quota)

Please submit your session feedback!

- Do it online at http://conferences.gse.org.uk/2019/feedback/jg
- This session is JG



1. What is your conference registration number?

This is the three digit number on the bottom of your delegate badge

2. Was the length of this presention correct?

1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○

3. Did this presention meet your requirements?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○

4. Was the session content what you expected?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○