

IBM Middleware Community

Cloud Native Deployment Automation with IBM Cloud Pak for Integration

Andy Garratt (andyg@uk.ibm.com)

Offering Manager, IBM Cloud Pak for Integration

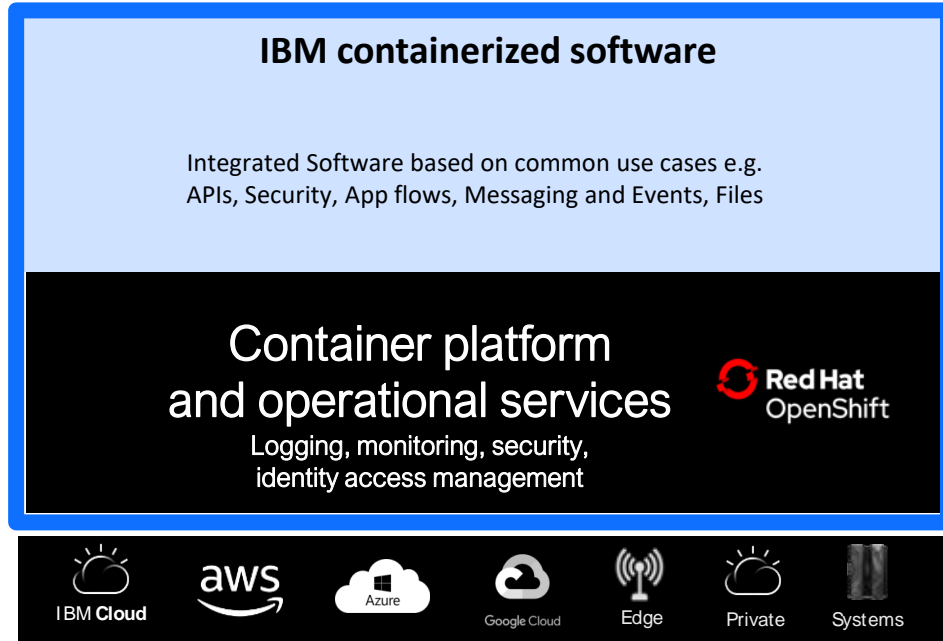
Rob Nicholson (rob_nicholson@uk.ibm.com)

Distinguished Engineer, Cloud Pak for Integration



Abstract: IBM Cloud Pak for Integration continues to deliver new enhancements that help your enterprise address pressing business challenges. This session will cover how the latest release of CP4I helps you accelerate your integrations while reducing costs and give you a technical introduction to operators. Join us as our experts explain how operators simplify the deployment and management of running CP4I on Kubernetes, and learn how operators can help you start to exploit the power of OpenShift for automated deployment and operation activities to make your CI/CD pipeline goals a reality.

Concept Of A Cloud Pak



Complete yet simple

IBM certified

Run anywhere

IBM Cloud Pak for Integration



Cloud Pak for Integration Version Updates:

2020.2.1.1

Broadest integration capabilities

Unified experience, operational efficiency & reuse

—

Deploy where needed

Container-based architecture with common enterprise services

—

Enterprise-grade

Secure, scalable, resilient



4.4 +



API Lifecycle

IBM API Connect

Unlock business data and assets as APIs

10.0.0.0



Event Streaming

IBM Event Streams

Delivers real-time Kafka event interaction

10.0.0



Application Integration

IBM App Connect

Connect your cloud and on-prem applications

11.0.0.9



High Speed Data Transfer

IBM Aspera

Super fast & secure data transport across any cloud

3.9.6



Enterprise Messaging

IBM MQ

Deliver messages reliably with enterprise-grade messaging

9.2.0



Secure Gateway

IBM DataPower Gateway

Control access to vital resources wherever they are

10.0.0.0

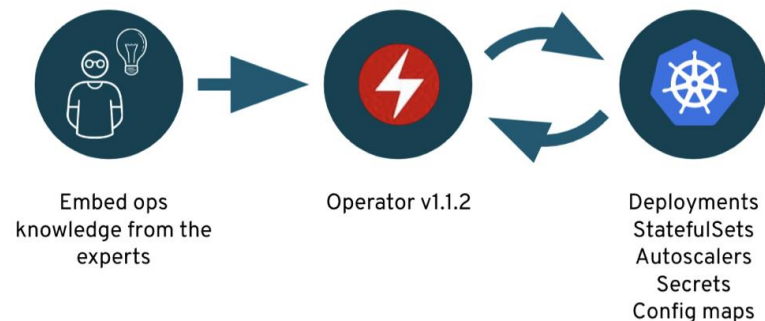
Cloud Native Deployment Automation

Operator support for all components

- ✓ An Operator is a method of packaging, deploying and managing a Kubernetes-native application
- ✓ Operators reduce the skills required to run software in Kubernetes
- ✓ Replaces Helm and makes Software Kubernetes Native – behaves as if it were part of Kubernetes
- ✓ Seamlessly use the integration capabilities of CP4I with RedHat OpenShift Container Platform
- ✓ Provides improved interoperability with CP4MCM & CP4Apps
- ✓ Multi-cluster orchestration of CP4I operators
- ✓ Event Streams operator deployment extends the Open Source Strimzi project at Cloud Native Computing Foundation

Operators significantly simplify deployment and management of CP4I on OpenShift through:

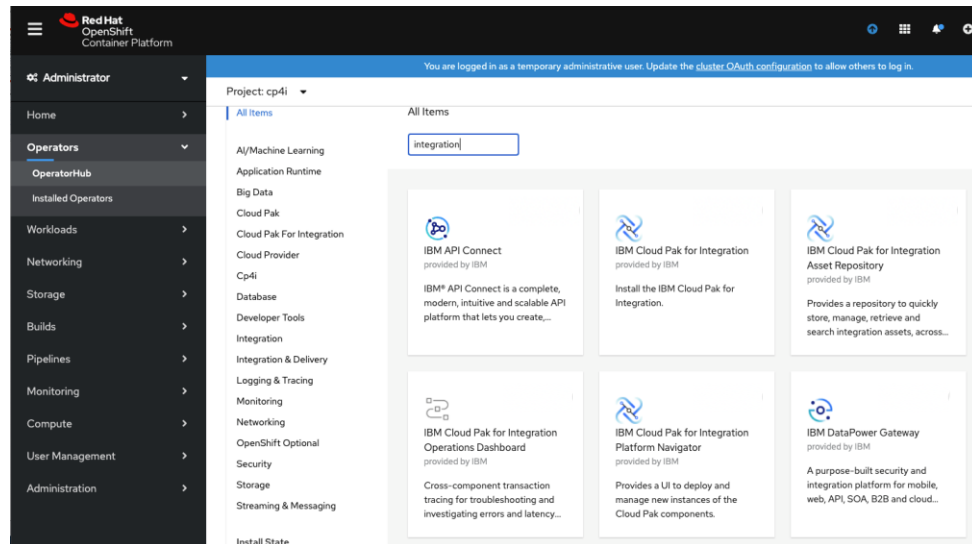
- ✓ Repeatability of installation and upgrade
- ✓ Constant health checks of every system component



CP4I Operators – Installation and update

An 'AppStore' like experience for enterprise software

- Install CP4I directly from Openshift OperatorHub Catalog
- Modular and componentised – Install only what you need on a cluster
- Updates, fixpaks and patches delivered automatically
 - No need for fix-central & Passport Advantage
- Air-gap install capability for clusters without internet connectivity.
- Installation and upgrade managed by Operator
- **Restricted Security Context** – Cloud Pak runs with Restricted SCC

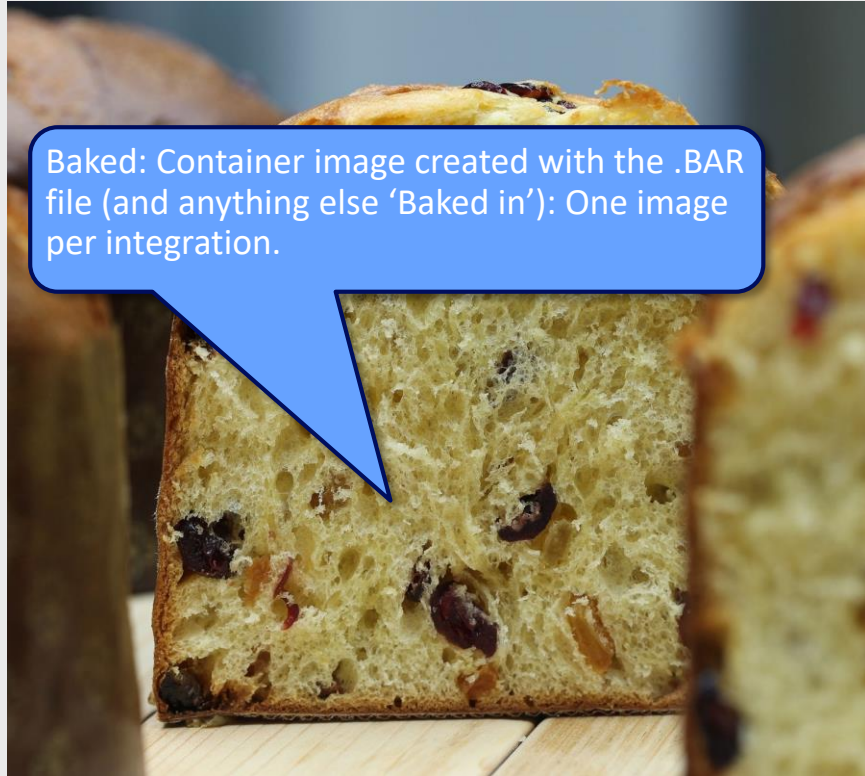


Demo

“Baking vs Frying”



“Baking vs Frying”



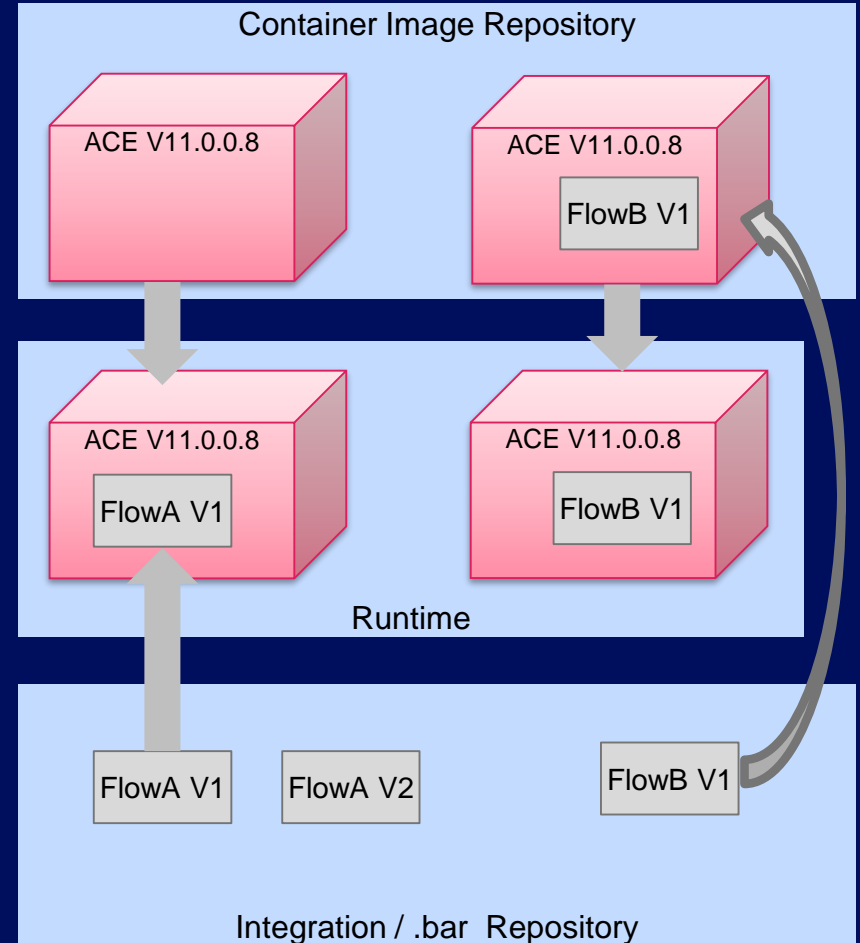
Immutability==Consistency

Do you deploy an App Connect container and then deploy flows at runtime?

Or build a container with all of the flows ready-built into it?

Which makes DEV look more like PROD?

Which is simpler to deploy?



“Baking” App Connect Images is not that hard . .

The screenshot shows the GitHub interface for the repository `andygblue / cp4i-deployment-samples`, which is forked from `IBM/cp4i-deployment-samples`. The repository has 0 stars and 11 forks. The `Code` tab is selected, showing the `cp4i-deployment-samples / DrivewayDentDeletion / ace-acme / Dockerfile` file. The file is 3 lines (3 sloc) and 133 Bytes. The commit history shows a commit by `DanRoseus` with the message "Use correct base image for ACE" on 17 Jun. The Dockerfile content is as follows:

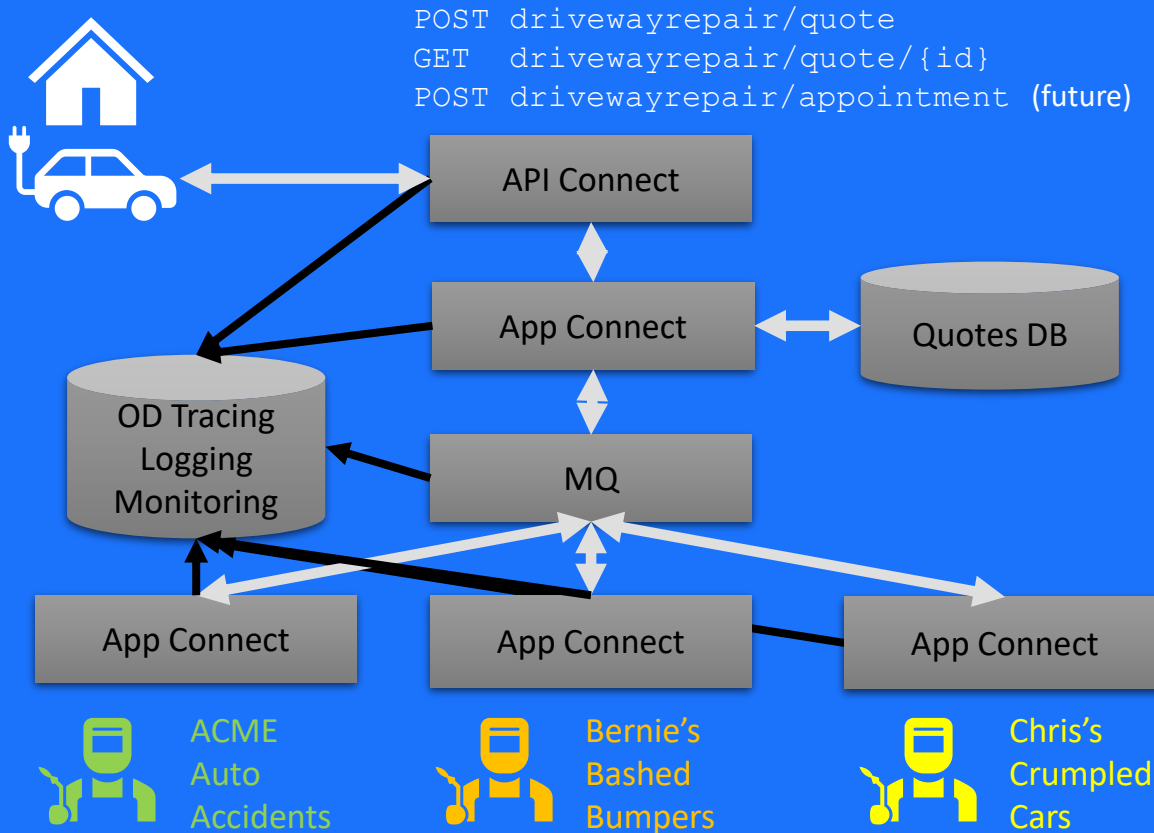
```
1 FROM cp.icr.io/cp/icp4i/ace/ibm-ace-mqclient-server-prod:11.0.0.8-r1-amd64
2 COPY Acme.bar /home/aceuser/bars/
3 RUN ace_compileBars.sh
```

A callout box highlights the Dockerfile content:

```
3 lines (3 sloc) | 133 Bytes

1 FROM cp.icr.io/cp/icp4i/ace/ibm-ace-mqclient-server-prod:11.0.0.8-r1-amd64
2 COPY Acme.bar /home/aceuser/bars/
3 RUN ace_compileBars.sh
```

What about deployment consistency and automation?



↔ API / Calls
→ Tracing Data

“Scatter gather pattern”

Or “Compare the Market/Meerkat”
Insurance style:



Or Booking flight/hotels– any kind
of online competitive marketplace.



Explore

Events

Operators >

Workloads >

Networking >

Storage >

Builds >

Pipelines >

Pipelines

Pipeline Runs

Pipeline Resources

Tasks

Task Runs

Cluster Tasks

Monitoring >

Compute >

User Management >

Project: all projects

Pipeline Runs

Dev Preview

Filter by name...

1 Complete

0 Running

0 Failed

0 Cancelled

Select All Filters

1 Item

Name

Namespace

Task Status

Started

Duration

Status

PLR main-pipelinerun-xrgmv

NS driveway-dent-deletion



Jun 25, 8:57 am

8m 3s

Succeeded



Pipelines allow us to automate deployments. Predictable. Repeatable.



Explore

Events

Operators >

Workloads >

Networking >

Storage >

Builds >

Pipelines ▾

Pipelines

Pipeline Runs

Pipeline Resources

Tasks

Task Runs

Cluster Tasks

Monitoring >

Compute >

User Management >

Project: all projects ▾

Pipeline Runs

Dev Preview

Filter by name...

1 Complete

0 Running

0 Failed

0 Cancelled

Select All Filters

1 Item

Name ↑

Namespace ↑

Task Status ↑

Started ↓

Duration

Status ↑

PLR main-pipelinerun-xrgmv

NS driveway-dent-deletion



🕒 Jun 25, 8:57 am

8m 3s

✅ Succeeded



Pipelines allow us to automate deployments. Predictable. Repeatable.



Explore

Events

Operators >

Workloads >

Networking >

Storage >

Builds >

Pipelines >

Pipelines

Pipeline Runs

Pipeline Resources

Tasks

Task Runs

Cluster Tasks

Monitoring >

Compute >

User Management >

Project: all projects ▾

Pipeline Runs

Filter by name...

1 Complete

0 Running

0 Failed

0 Cancelled

[Select All Filters](#)

1 Item

Name ↑

Namespace ↑

Task Status ↑

Started ↓

Duration

Status ↑

PLR main-pipelinerun-xrgmv

NS driveway-dent-deletion



🕒 Jun 25, 8:57 am

8m 3s

✅ Succeeded



2020.1.1 (March) = OpenShift 4.3

2020.2.1 (June) = OpenShift 4.4+



Stateful Sets

Secrets

Config Maps

Cron Jobs

Jobs

Daemon Sets

Replica Sets

Replication Controllers

Horizontal Pod Autoscalers

Networking

Storage

Builds

Pipelines

Pipelines

Pipeline Runs

Pipeline Resources

Tasks

Task Runs

Project: driveway-dent-deletion

Tasks

Create Task

Name ↑

Namespace ↑

T build-push-ace-acme-docker-image

NS driveway-dent-deletion

T build-push-ace-api-docker-image

NS driveway-dent-deletion

T build-push-ace-bernie-docker-image

NS driveway-dent-deletion

T build-push-ace-chris-docker-image

NS driveway-dent-deletion

T build-push-mq-docker-image

NS driveway-dent-deletion

T helm-deploy-task-ace-acme

NS driveway-dent-deletion

T helm-deploy-task-ace-api

NS driveway-dent-deletion

T helm-deploy-task-ace-bernie

NS driveway-dent-deletion

T helm-deploy-task-ace-chris

NS driveway-dent-deletion

T helm-deploy-task-mq

NS driveway-dent-deletion

Pipelines are made up of tasks – tasks can be reused.



Explore

Events

Operators

Workloads

Networking

Storage

Builds

Pipelines

Pipelines

Pipeline Runs

Pipeline Resources

Tasks

Task Runs

Cluster Tasks

Monitoring

Compute

Tools on 2020 / © 2020 IBM Corporation

User Management

Project: driveway-dent-deletion

Pipeline Runs > Pipeline Run Details

PLR main-pipelinerun-xrgmv ✓ Succeeded

Dev Preview

Actions

Overview **YAML** Logs

Pipeline Run Overview



Name

main-pipelinerun-xrgmv

Namespace

NS driveway-dent-deletion

Labels

A pipeline run is all of the tasks needed – can deploy the entire 'solution'



James-Kirk Merge pull request #31 from IBM/zero-downtime

..

| | | |
|----------------------------|---|-------------|
| ace-acme | Revert to V1 of the bars | 16 days ago |
| ace-api | Updating with dan's corrected api bar | 16 days ago |
| ace-bernie | Revert to V1 of the bars | 16 days ago |
| ace-chris | Revert to V1 of the bars | 16 days ago |
| ace-src | Add the ACE source | 15 days ago |
| media | Improve Driveway Dent Deletion readme | 29 days ago |
| mq | Update to latest bars and fix mq dockerfile | last month |
| cicd-webhook-triggers.yaml | Use 1 replica for ace-api until fixed to work with >1 | 15 days ago |
| continuous-load.sh | Updating param docs with defaults | 16 days ago |
| prereqs.sh | Remove echo | 24 days ago |
| readme.md | Update readme with webhook type | 24 days ago |
| repositories.yaml | Fix helm installs to use old helm | last month |
| run-pipeline-script.sh | Add pipeline files: https://github.ibm.com/cp4i/icp4i-sagan/issues/775 | last month |

Check out github for samples for this (WIP) – the 'Car Crash' demo is on public git too



Replica Sets

Replication Controllers

Horizontal Pod Autoscalers

Networking

Storage

Persistent Volumes

Persistent Volume Claims

Storage Classes

Builds

Build Configs

Builds

Image Streams

Project: ace ▼

Image Streams

Create Image Stream

Name ↑

Namespace ↓

IS ace-ddd-acme

NS ace

IS ace-ddd-api

NS ace

IS ace-ddd-bernie

NS ace

IS ace-ddd-chris

NS ace

Image streams can be used to store 'baked' container images

Jobs

Daemon Sets

Replica Sets

Replication Controllers

Horizontal Pod Autoscalers

Networking

Storage

Persistent Volumes

Persistent Volume Claims

Storage Classes

Builds

Build Configs

Builds

Image Streams

Project: ace ▼

[Image Streams](#) > Image Stream Details

 **ace-ddd-acme**

Overview

YAML

History

Jun 25, 9:04 am

 **ace-ddd-acme:latest-amd64**

from image-registry.openshift-image-registry.svc:5000/ace/ace-ddd-acme

sha256:95bd37b893321e3d24fd1f3c4a7443e5d3319eb39d64af930273d1c1bb05688a

Keeps the 'history'
and tags versions –
can also use 'latest'

Demo

Thank you

Andy Garratt
Offering Manager, IBM Cloud Pak for Integration

andyg@uk.ibm.com

<https://www.ibm.com/cloud/cloud-pak-for-integration>

© Copyright IBM Corporation 2020. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).

