Deploying Integration in a real-world dev-ops environment

—

Rob Nicholson – Distinguished Engineer
Andy Garratt – Offering Manager

IBM

Agile Integration

Modernizing integration to enable **business** agility

Decentralized Ownership

People & Process

→ Accelerate agility and innovation (***development*** agility)

Fine-grained deployment

Architecture

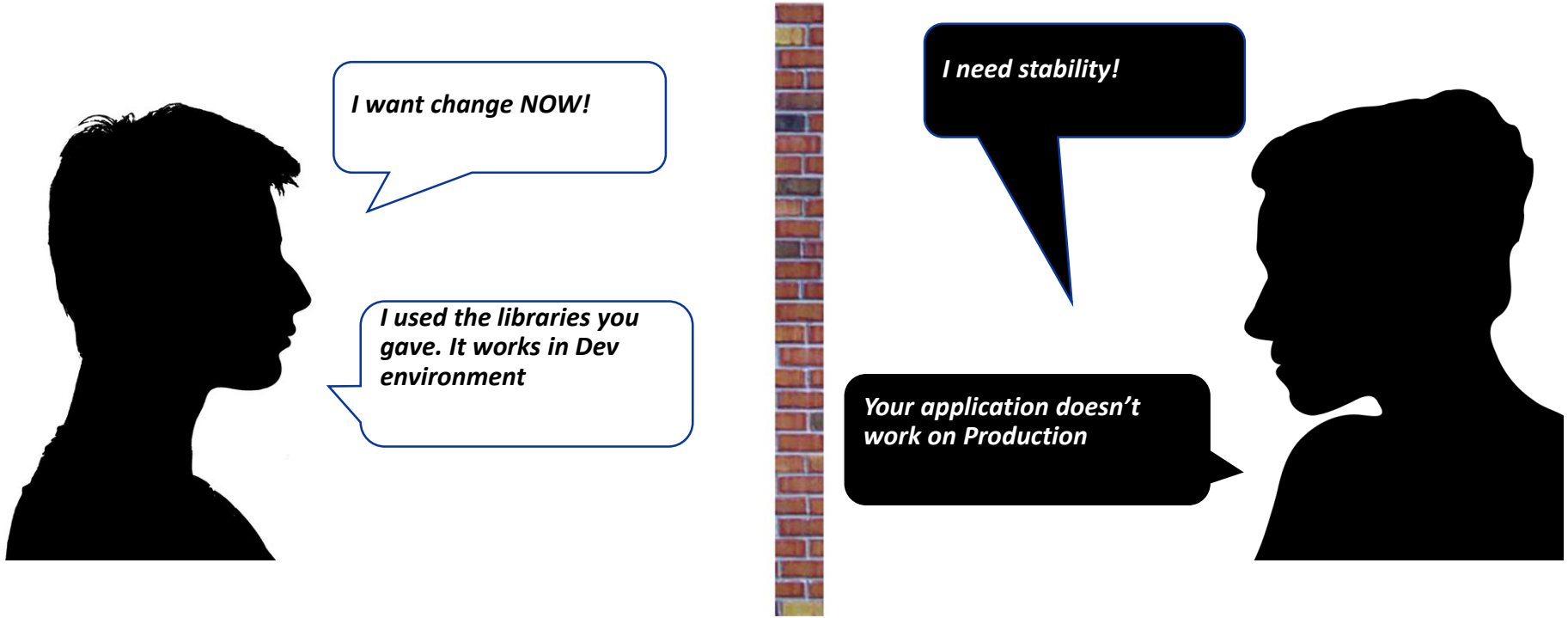
→ Improve build independence and production velocity (***deployment*** agility)

Cloud native infrastructure

Technology

→ Dynamic scalability and inherent resilience (***operational*** agility)

Why DevOps?


I want change NOW!
I used the libraries you gave. It works in Dev environment
I need stability!
Your application doesn't work on Production
DEVELOPMENT
OPERATIONS

Key Objectives of a DevOps pipeline

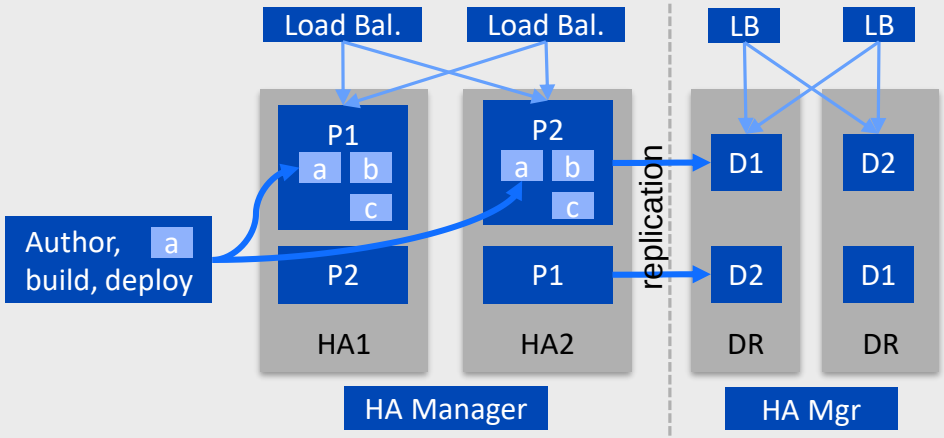
- Continuous Integration - Developers ‘safely’ integrate and test as they go
- Continuous Deployment – The artifacts they create progress rapidly into production ‘safely’.
- Good automated test coverage + control over manual ‘what if’ tests.
- The system running in production is exactly the same as the system that is tested.
  - Avoidance of all persistent environmental ‘state’.
  - Everything is under version control, including the environment → gitops.
  - Infrastructure as code.
- Declarative deployment: Declare the way the world shall be. Kubernetes makes it so.

Traditional vs Cloud native deployment

Product component Product artefact

Traditional

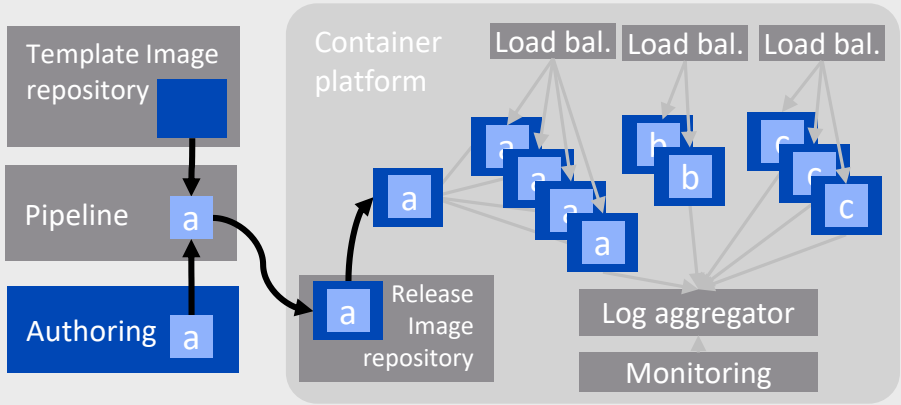
Artifacts are delivered into long-lived product ‘runtimes’.



Cloud-native

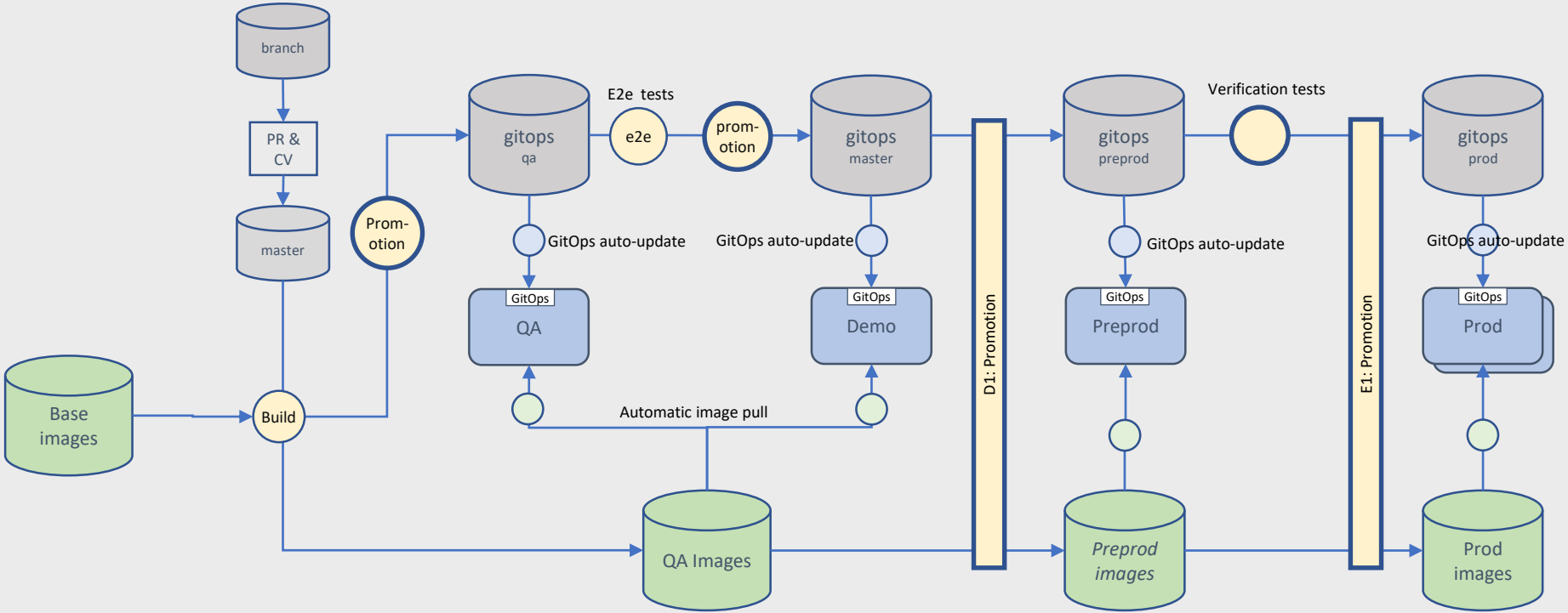
Complete control of the environment.

Devops for the cloud native world.



Example DevOps pipeline.

Declarative deployment



Approaches to pipeline

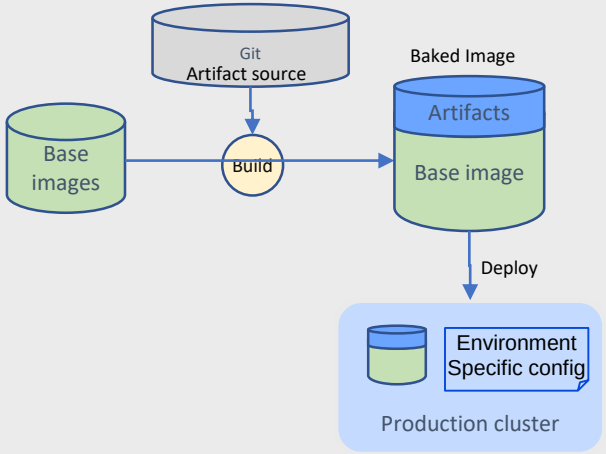
Baked Image Approach

(As per previous slide)

- Base Images contain product runtimes
- Images are extended, baking integration artifacts in.

PROs
- Immutable images deployed across pipeline.

CONS
- Requires a more sophisticated pipeline.

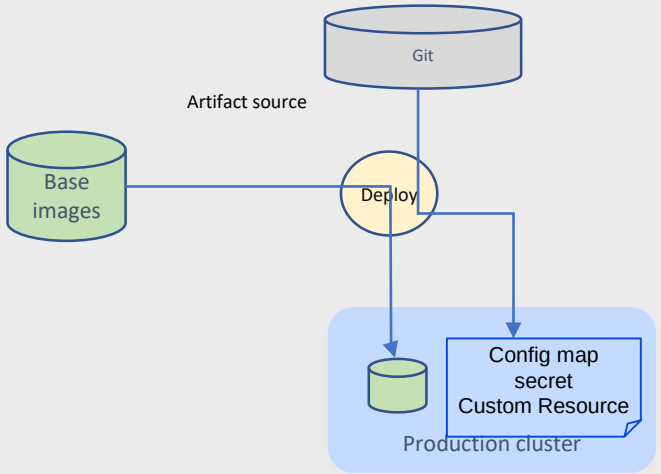


Fried Approach

- Base Images contain product runtimes, deployed unchanged.
- Integration artifacts deployed as configuration.

PROs
- Much simpler pipeline

CONS
- Not suitable for large/complex objects (libraries)



Baked versus Fried for Integration runtimes

Baked Image Approach

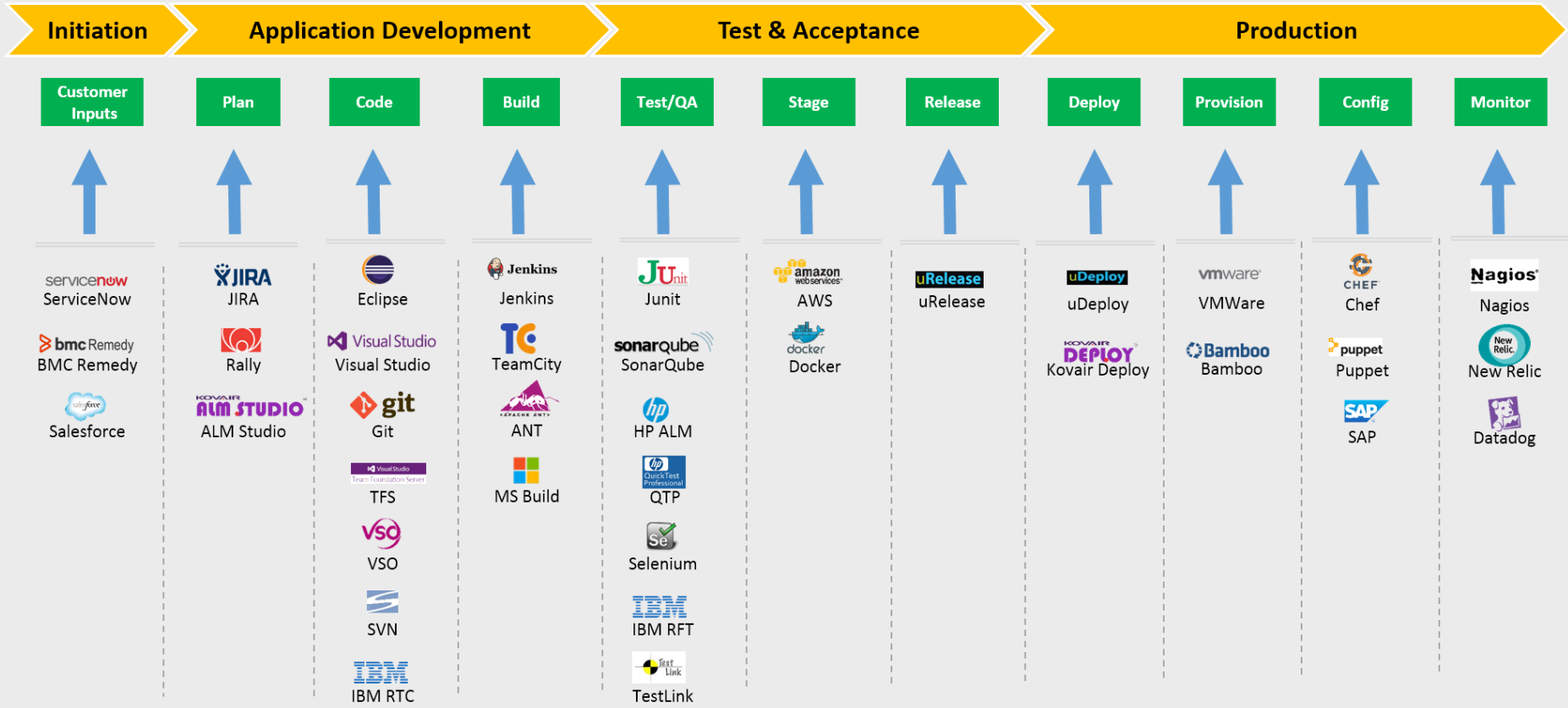
- Extend CP4I base images deploy with Helm
- ACE
  - Bake BAR files into image
- MQ
  - Bake MQSC & .ini into the image.
- Datapower
  - Bake configuration into the image.

Fried Approach

- ACE
  - Bar file can be served from URL.
    - Object store/dashboard server.
  - Configuration placed into config maps and secrets.
- MQ
  - MQSC and Ini files info config map.
- Datapower
  - Place configuration into a config map.

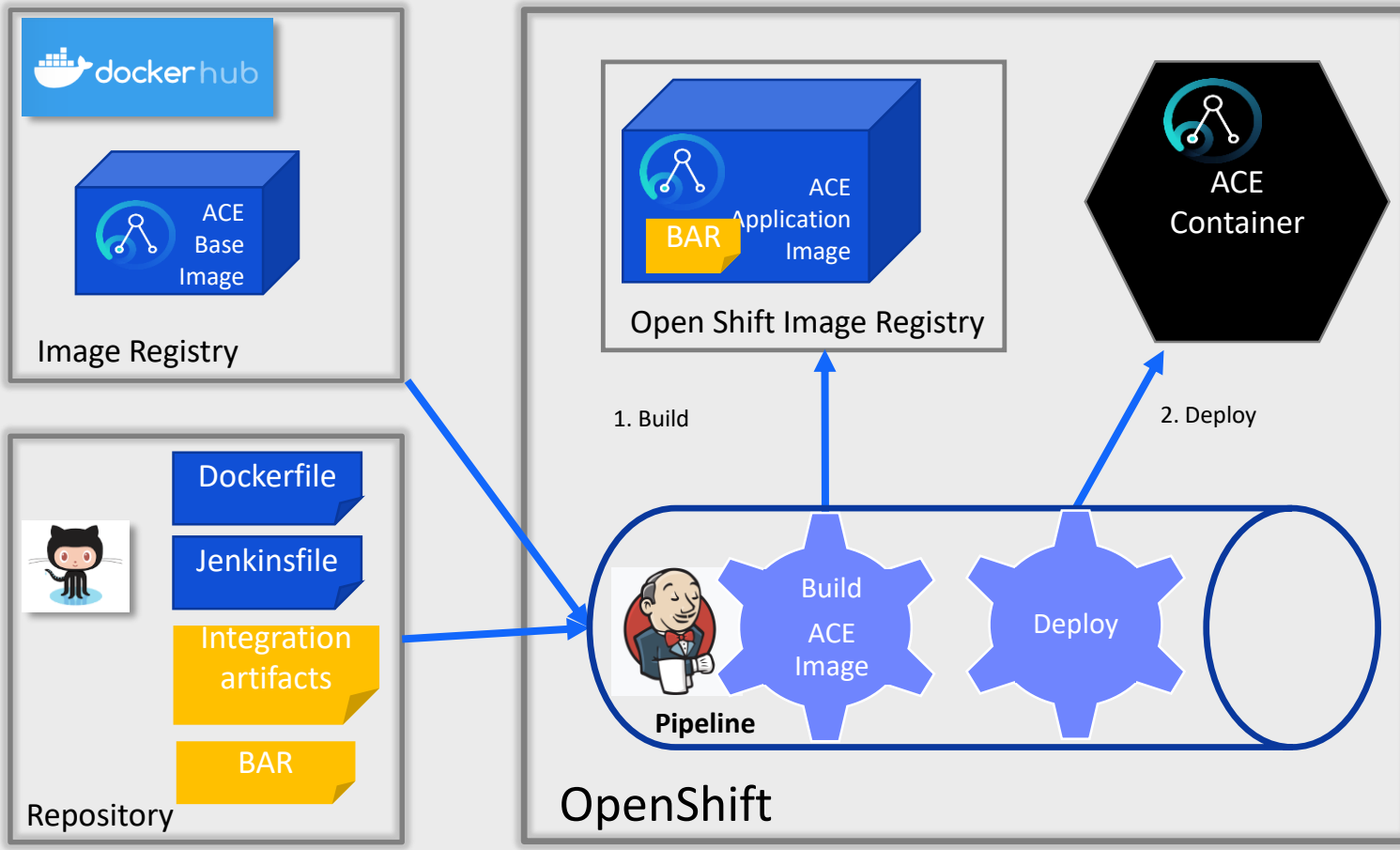
In the future, Kubernetes Custom Resources combine the best of both approaches

DevOps Tools Landscape



Demo - Cloud Native DevOps

Build and deploy your ACE applications with Open Shift embedded CI/CD Pipeline



References.

[Create your integration application on OpenShift using Jenkins pipeline]

Looking to explore Red Hat OpenShift streamlined CI/CD workflows to run your ACE container natively on Red Hat OpenShift? In this post we show how...

→ Continue reading

[IBM ACE v11 Continuous Integration-*Maven*-Jenkins - IBM Integration]

How to build an ACE v11 (App Connect Enterprise v11) project and deploy the bar file to target the Integration Server using Maven and Jenkins.

→ Continue reading

[IIB (v9 & v10) Continuous Integration-*Maven*-Jenkins - IBM Integration]

In this article I will explain how to build an IIB (IBM Integration Bus v9 & v10) project and deploy the bar file to target...

→ Continue reading

[Creating a custom integration node setup on a virtual machine using Chef - IBM Integration]

Learn how to build a virtual machine that has a custom integration node setup, using

[Integration Development to Micro Services Principles on OpenShift – Part 3]

Introduction Modern platforms, DevOps tooling and agile approaches have accelerated the rate at which organizations can bring new applications and business function to bare. At...

→ Continue reading

[Integration Development to Micro Services Principles on OpenShift – Part 2]

Introduction Modern platforms, DevOps tooling and agile approaches have accelerated the rate at which organizations can bring new applications and business function to bare. At...

→ Continue reading

[Integration Development to Micro Services Principles on OpenShift – Part 1]

Introduction Modern platforms, DevOps tooling and agile approaches have accelerated the rate at which organizations can bring new applications and business function to bare. At...

→ Continue reading

[An approach to build DevOps pipeline for ACE on Cloud Pak for Integration]

We had published a recipe in developerWorks to automate the build and deployment of ACE projects on Cloud Pak for Integration. In this blog, we...

→ Continue reading

Thank You

Rob Nicholson rob_Nicholson@uk.ibm.com

Andy Garratt andy.garratt@uk.ibm.com