Introduction to Cloud Pak for Applications

—

Luca Floris
IBM Cloud Technical Consultant

Agenda – Cloud Pak for Applications

Intoduction to Cloud Pak for Applications
Business value
Overview of Components

IBM Accelerators for Teams
What challenge is Kabanero trying to solve?
What are Kabanero's main players?
Overview of Kabanero components

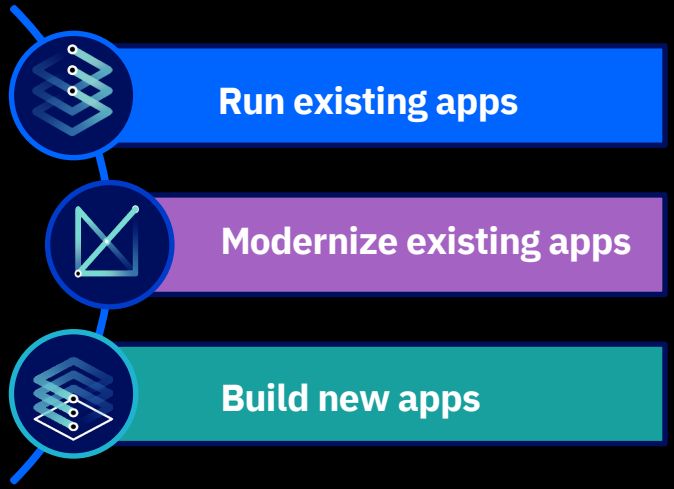
Introduction to Appsody
View of a Solution Architect
View of a Developer

Introduction to Tekton
Kabanero Collections
Pipelines with Tekton

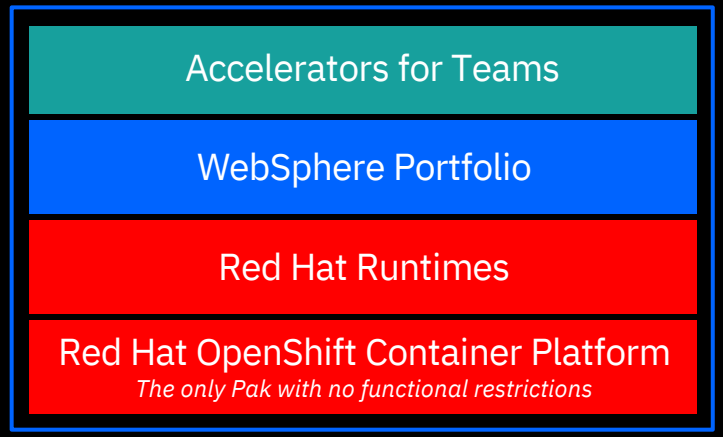
Cloud Pak for Applications V4 – Strategic Value

The ultimate flexibility across open source, platform, runtimes, and tools all in 1 place!

- Build new cloud-native apps on the broadest range of supported runtimes - ***ultimate choice***
- OpenShift Container Platform 4.X for hybrid multi-cloud - ***ultimate portability***
- "Right-size" and modernize between the various WAS editions - ***ultimate flexibility***
- Empower teams to start quickly with centralized control and increased productivity - ***ultimate control***



Ultimate open source stack for all workload needs



Cloud Pak for Applications: what you need today, what you need tomorrow

Run existing apps

WebSphere Application Server

WebSphere ND | WebSphere Base

Liberty Core | Mobile Foundation

JBoss Enterprise Application Server

Modernize existing apps

IBM Modernization & Developer Tools

Included with all components

Transformation Advisor

Application Navigator

WebSphere Migration Toolkit

Enterprise Dev tools & extensions for local IDE's

Supported when used with Cloud Pak for Applications, no charge

Build new Cloud Native apps

ACCELERATORS FOR TEAMS & ENTERPRISE GOVERNANCE

Equipping teams with content, tools, architectures and methods so they can focus on business problems from day 1

Accelerators for App Mod and Transformation	Accelerator Builder for Cloud-Native Apps	Accelerators for Innovation

CODE READY DEVELOPER TOOLS

Red Hat Runtimes

Java/Jakarta EE Open Liberty, JBoss EAP	**Java SE** OpenJDK	**Java Web** JBoss WS
Javascript node.js	**Spring** Spring Boot	**Java.next** QUARKUS
MicroProfile Open Liberty , Thorntail	**Reactive** VERT.X	**Serverless** Cloud Functions

WebSphere

- **Traditional WebSphere**
- **Liberty**

Distributed Data Data Grid	**Messaging** AMQ Broker	**Mobile Foundation**

Red Hat OpenShift Container Platform

Perpetual and Term licensing options available w/ no restrictions to OpenShift

Cloud Pak for Applications: what you need today, what you need tomorrow

Run existing apps

WebSphere Application Server

WebSphere ND | WebSphere Base

Liberty Core | Mobile Foundation

JBoss Enterprise Application Server

Modernize existing apps

IBM Modernization & Developer Tools

Included with all components

Transformation Advisor

Application Navigator

WebSphere Migration Toolkit

Enterprise Dev tools & extensions for local IDE's

Supported when used with Cloud Pak for Applications, no charge

Build new Cloud Native apps

ACCELERATORS FOR TEAMS & ENTERPRISE GOVERNANCE

Equipping teams with content, tools, architectures and methods so they can focus on business problems from day 1

Accelerators for App Mod and Transformation | **Accelerator Builder for Cloud-Native Apps** | **Accelerators for Innovation**

CODE READY DEVELOPER TOOLS

Red Hat Runtimes

Java/Jakarta EE Open Liberty, JBoss EAP	**Java SE** OpenJDK	**Java Web** JBoss WS
Javascript node.js	**Spring** Spring Boot	**Java.next** QUARKUS
MicroProfile Open Liberty , Thorntail	**Reactive** VERT.X	**Serverless** Cloud Functions

WebSphere

Traditional WebSphere

Liberty

Distributed Data Data Grid

Messaging AMQ Broker

Mobile Foundation

Red Hat OpenShift Container Platform

Perpetual and Term licensing options available w/ no restrictions to OpenShift

IBM Accelerators for Teams

More work and overlapping of concerns

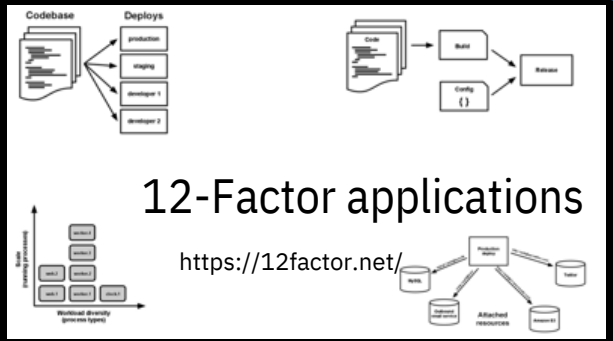
Today's skills

Jane: Lead Enterprise Developer

Champ: Solution Architect

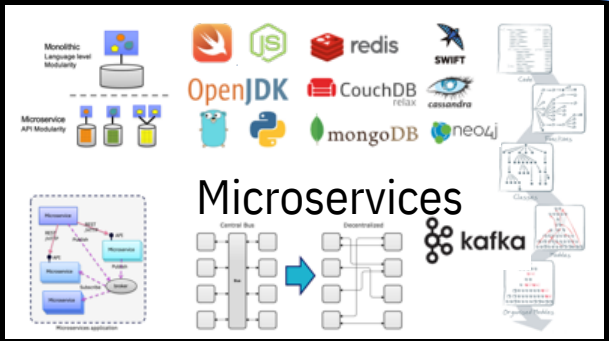
Todd: Operations Admin

Tomorrow's challenges. What is cloud-native?



Independent, autonomous, stateless, processes

Configuration sources, service references



Application as composition of polyglot services

Build, package, deploy, observe

Role expansion

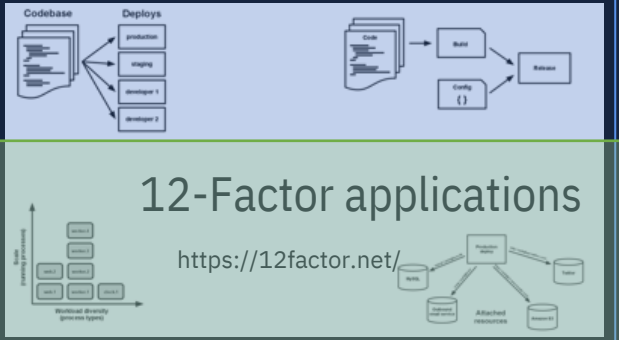
Cloud Native DevOps

Cloud-native practices cross traditional development and operational roles.

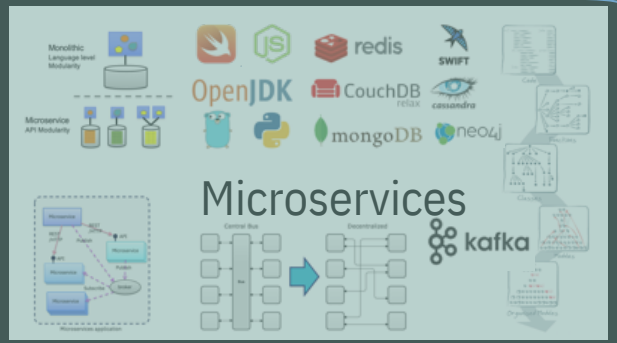
Microservices and other distributed application patterns are more complicated than monoliths, with many more moving parts.

Automated tools and shared conventions make the difference between a flexible, scalable system, and an unmanageable mess.

Development



Independent, autonomous, stateless, processes



Application as composition of polyglot services

Operations

Configuration sources, service references

Build, package, deploy, observe

100% Open Source

Accelerators for teams

based on Kabanero

A modern microservices-based framework that simplifies development, build, and continuous deploy of apps for Kubernetes and Knative

Integrated Collections

100% open source frameworks and runtimes optimized for cloud-native

- → Customizable build pipelines
- → Pre-built Kubernetes deployments
- → Lifecycle management

Integrated Developer Tools

Simplify building cloud-native apps in containers for Kubernetes and Knative

- → Extensions to industry standard IDEs
- →Templates and developer focused CLI

Integrated DevOps Toolchain

Automated end-to-end toolchain from code check-in to production Kubernetes deploy

- → Tekton and event driven DevOps
- → Completely customizable to meet company policies and choices

Kubernetes Knative Istio

Accelerators for teams

based on Kabanero™

Same open source code plus:

Versioned & Supported

Simplified Cloud Pak Install with OpenShift

Licensing / entitlement tool integration

Modernization Toolkit integration

Integrations with Other Cloud Paks including Multicloud Management

Enterprise-grade Cloud Pak Standards

Integrated Collections from an entitled and trusted source

Entitlement to Red Hat Runtimes

Collaboration!

Jane: Lead Enterprise Developer

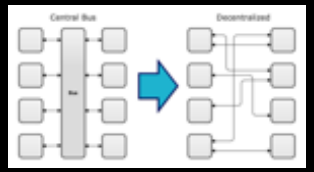
Champ: Solution Architect

Todd: Operations Admin

codewind

Visual Studio Code eclipse

Eclipse Che



Build microservices

MICROPROFILE	NODE JS	OPEN LIBERTY	LOOPBACK	SPRING BOOT	JAKARTA EE

Customised Collections

Kubernetes Containers Etcd istio Grafana Operators tekton Git

Opininated platform

Cloud Pak for Applications Components: IBM Accelerators for Teams based on Kabanero

Kabanero

https://kabanero.io/

A modern microservices-based framework for continuous deploy of apps on Kubernetes and Knative

appsody

https://appsody.dev

Pre-built stacks for popular open source runtimes and frameworks that simplifies building cloud-native apps in containers

codewind

https://www.eclipse.org/codewind/

Extensions to popular IDEs for cloud-native development with Kubernetes

TEKTON

https://tekton.dev

Kubernetes-native pipelines for CI/CD

Appsody– Main components

appsody: CLI

Containerized iterative development

run | test | debug | build | deploy

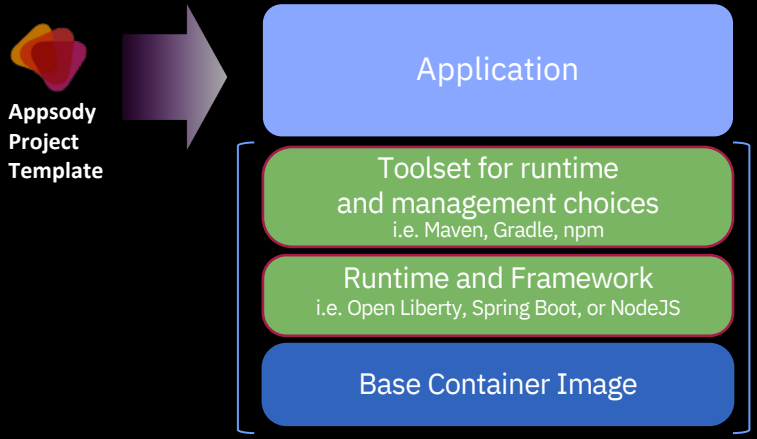
appsody: stacks

Sharable pre-built, cloud-optimized technology stacks

appsody: deploy

Production deploy to Kubernetes Server or Serverless Scaling

appsody Stack concept

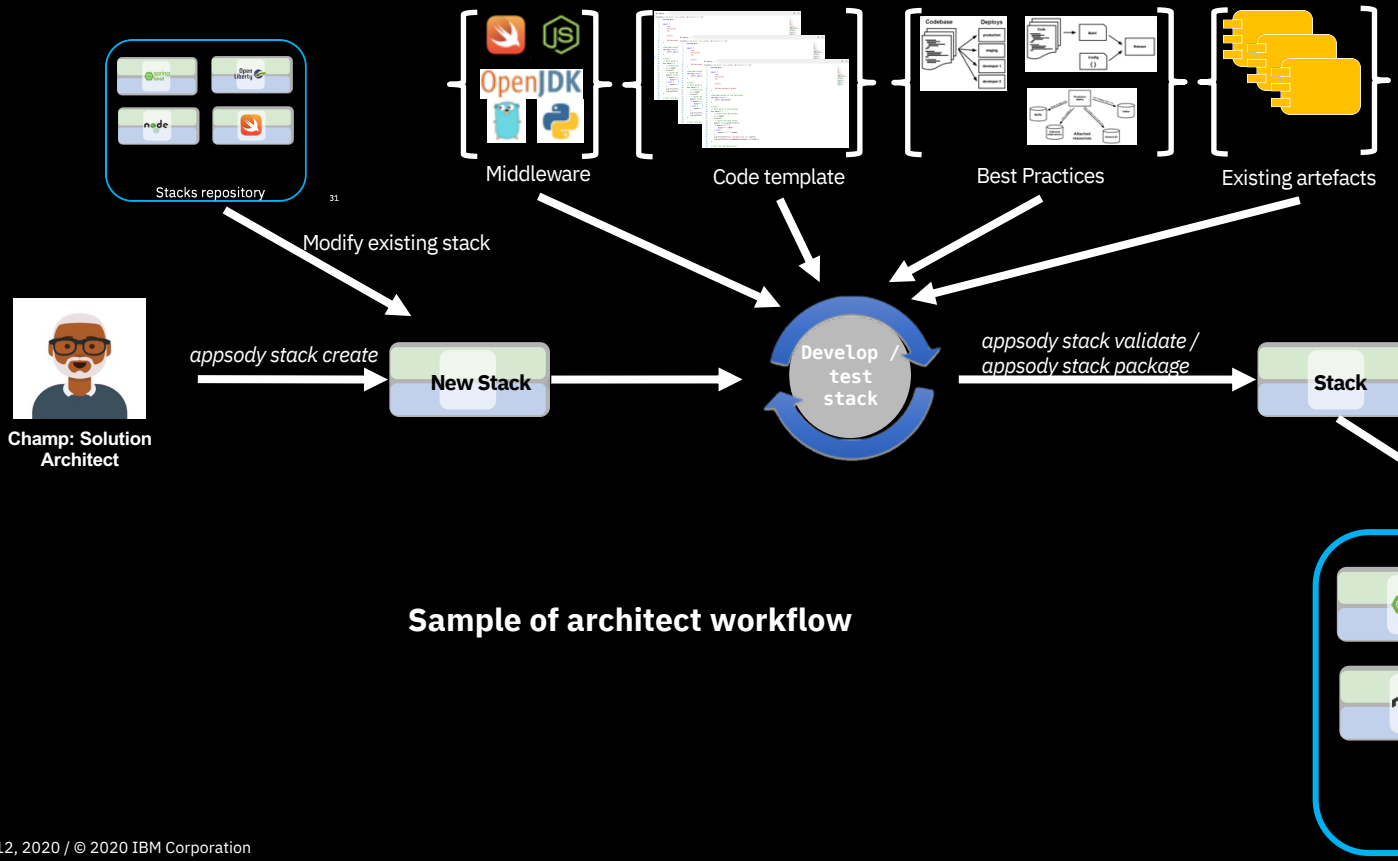


Stack
Runtimes & frameworks in pre-built container images, ready to go or customize them yourself.

Optimized for the target platform
- Open Tracing
- Metrics/Monitoring
- Logging
- Security and Auth
- … etc

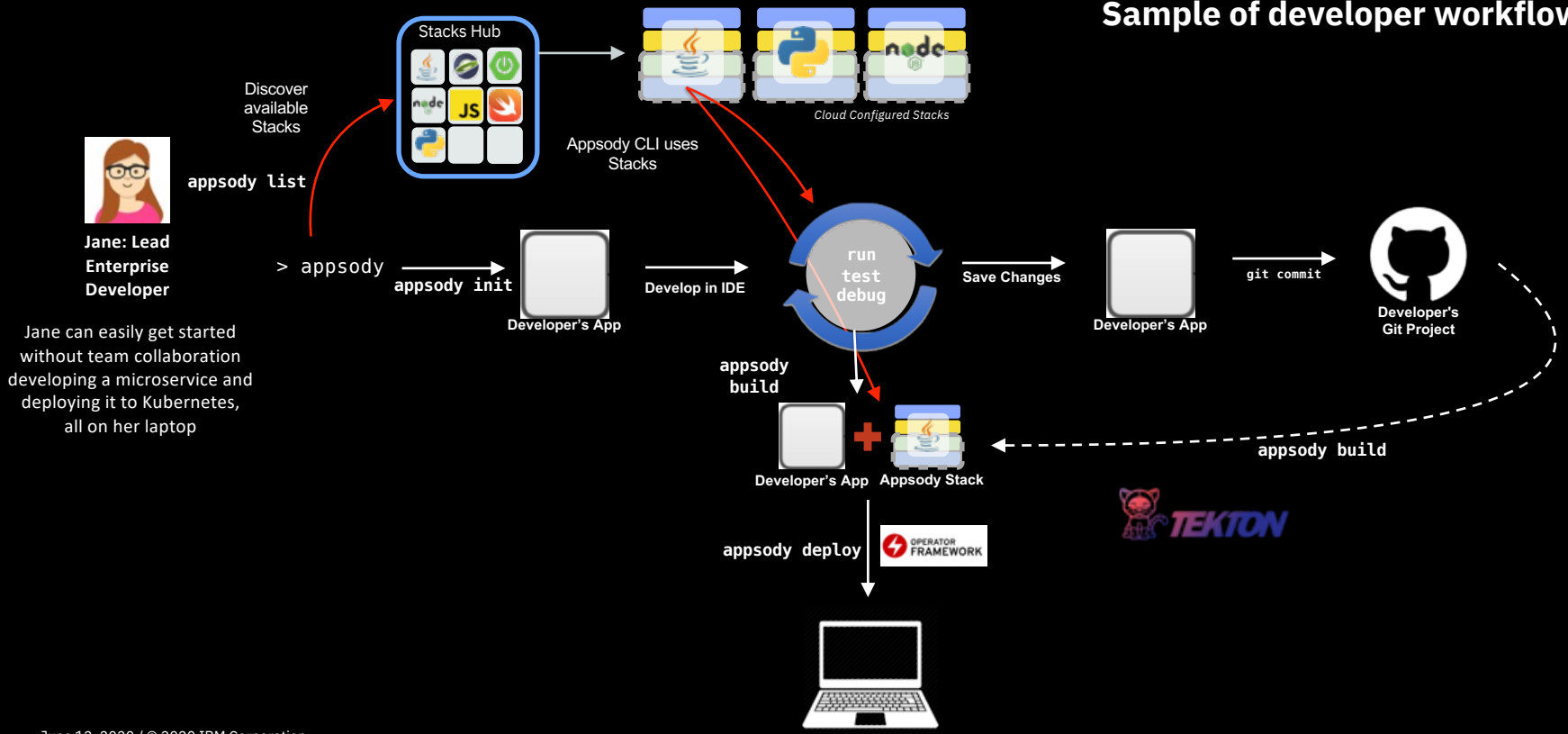
Target cloud platform

Create a Collection



Sample of architect workflow

Create a micro service application



Developer Tools

Simplified Development in Containers

Kabanero Introducing CI/CD

Introduces “Collections” which include everything you need to create an app or microservice in a single container image, along with enterprise-grade deployment & integrated continuous delivery pipelines.
Stacks and pipelines are customizable to meet enterprise and platform needs.

Appsody Project Template

Appsody Stacks

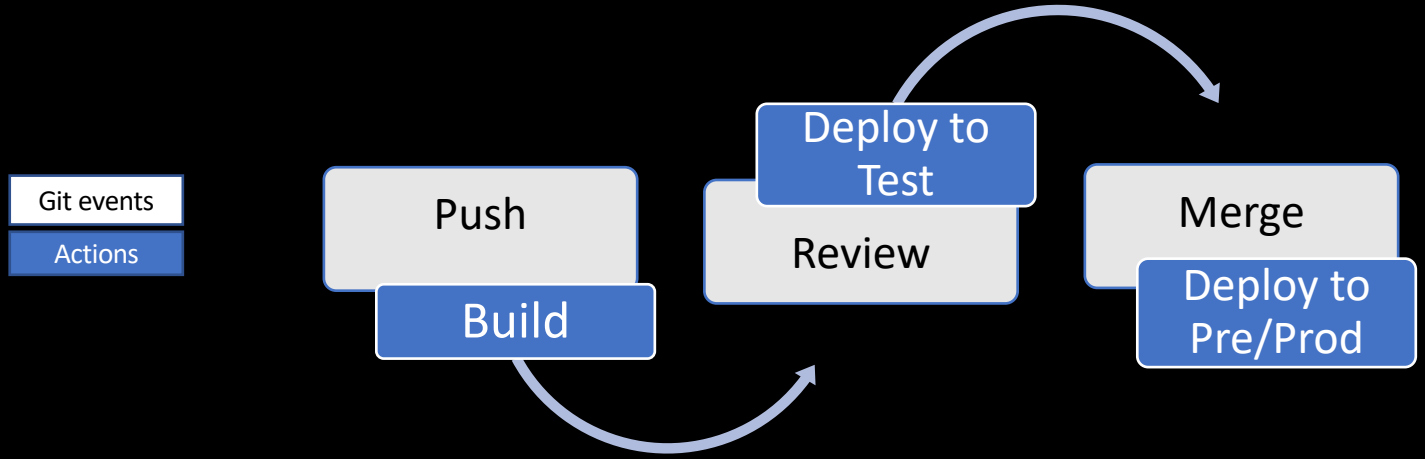
- Pre-built container image ready for dev
- Pre-built container image ready for prod

Each Container image includes:

- ✓ Defined toolset for the runtime
- ✓ Frameworks (one or more)
- ✓ Data gathering tools
- ✓ Health end points
- ✓ Logging configuration

Kubernetes-native Build/CD pipelines

Enterprise-grade deployment using a Kubernetes operator with choice of serverless with Knative or standard Kubernetes resources



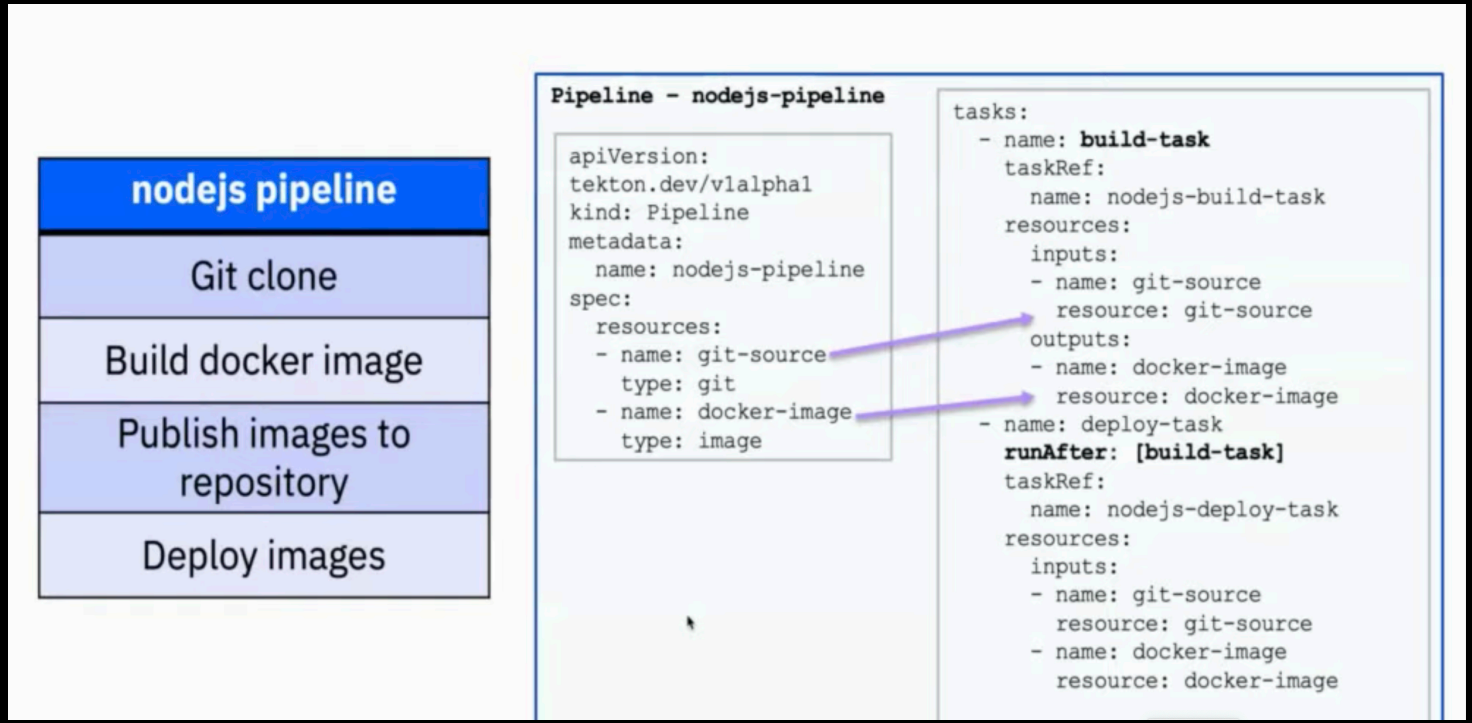
Pipelines: Task 1 → Task 2 → ... → Task N

Types of Tasks:

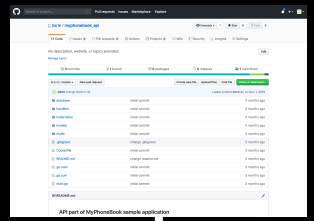
- Appsody build
- Container vulnerability scan
- Container signature
- Container publish (Tagging)
- Application scan
- Acceptance test harness
- Appsody deploy (operator-based)
- Razee pipeline promotion / cross cluster deploy

Customizable in a Collection by Champ (Solution Architect)

Demystifying a simple Tekton pipeline



Deploy an application with Tekton pipelines



Pipeline resource

Test Application

Image registry

Red Hat OpenShift Container Platform

Pipeline resource

Build

Get code

Build app

Test

Create instance

Run test

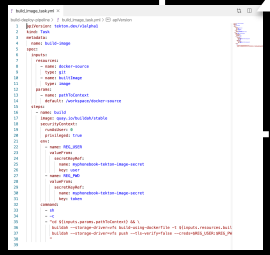
Publish

Build container

Publish container

Deploy

Deploy App



Pipeline

Shared PV

Questions ?

Demo – Building an Appsody Application

Lab – Building an Appsody Application and a Tekton Pipeline

Visit https://github.com/lfloris/openshift-bootcamp/tree/master for lab materials

Go to Lab 13

Goals

Deploy an application using Appsody, then implement a Tekton pipeline

Cloud Pak for Applications Installation

Export your credentials

```
export ENTITLED_REGISTRY=cp.icr.io
export ENTITLED_REGISTRY_USER=cp
export ENTITLED_REGISTRY_KEY=<apikey>
```

Extract configuration files

```
docker run -v $PWD/data:/data:z -u 0 \
           -e LICENSE=accept \
           "$ENTITLED_REGISTRY/cp/icpa/icpa-installer:4.2.0" cp -r "data/*" /data
```

Install CP4Apps

```
docker run -v ~/.kube:/root/.kube:z -u 0 -t \
           -v $PWD/data:/installer/data:z \
           -e LICENSE=accept \
           -e ENTITLED_REGISTRY -e ENTITLED_REGISTRY_USER -e ENTITLED_REGISTRY_KEY \
           "$ENTITLED_REGISTRY/cp/icpa/icpa-installer:4.2.0" install
```

Thank you

.