Cloud Native Development

Mangesh Patankar
– IBM Developer Advocate

Riya M Roy, Saurav Raiguru
– IBM Developer Advocate

IBM
CODE

About Me

- Developer Advocate @IBM
- 18+ years of IT Experience
- Been in IBM for 3 years
- Working as Technical Evangelist on trending technologies, solutions like Serverless, Containers, Orchestrations, Private Cloud (Cloud Native), Blockchain, IoT, More...
- Speaker in Technical Conferences by OSI, Cypher, Container Conference, More
- Meetup Organizer – Cloud Mumbai
- Twitter: @MangeshPatank

IBM CODE

Agenda

What is Cloud Native Applications?

Current State of Applications

Attributes of Cloud-Native Applications

Challenges for Developers

How those are solved using Kabanero?

Demo

Hands On

Cloud Native Applications

Cloud-native applications have a lot of advantages compared to monolithic architectures
- scalability
- elasticity

Cloud-native applications are built using **multiple, independent microservices** that are deployed in cloud environments.

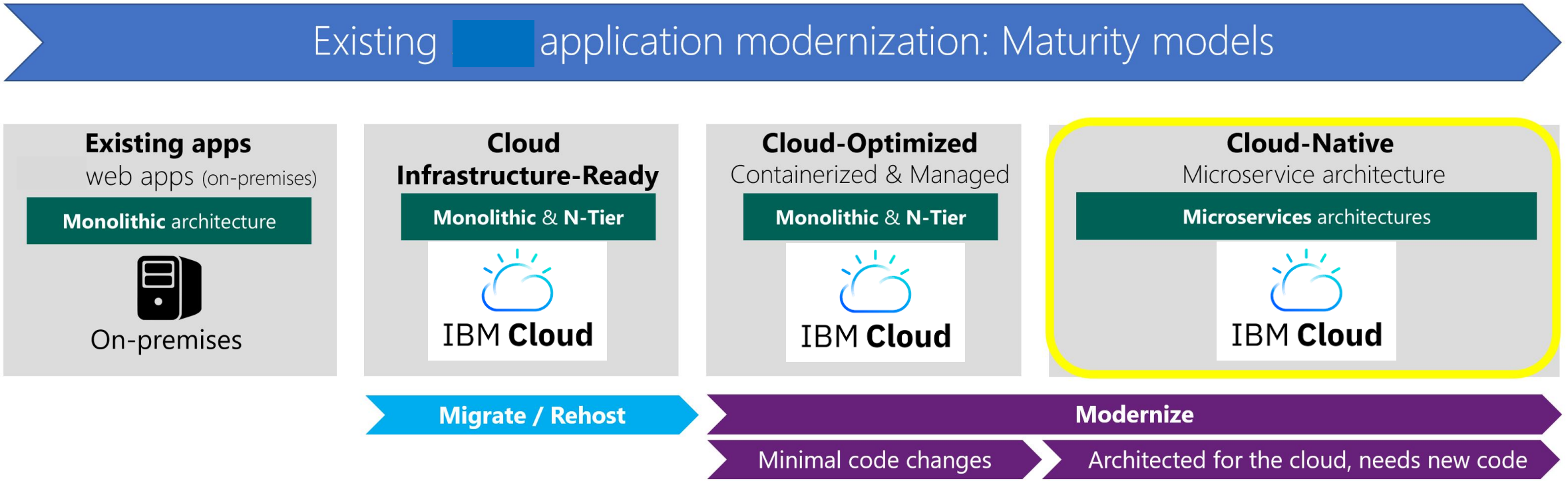
Cloud-native platforms typically also provide services that developers can use without having to worry about infrastructure.

These advantages are available in public clouds and now also in some private clouds which is important for companies with high data security and privacy requirements.

Using a **DevOps-based** continuous delivery model, software development teams can quickly, iteratively and automatically add new features to an application

Cloud-native development takes an **agile approach** to creating new applications.

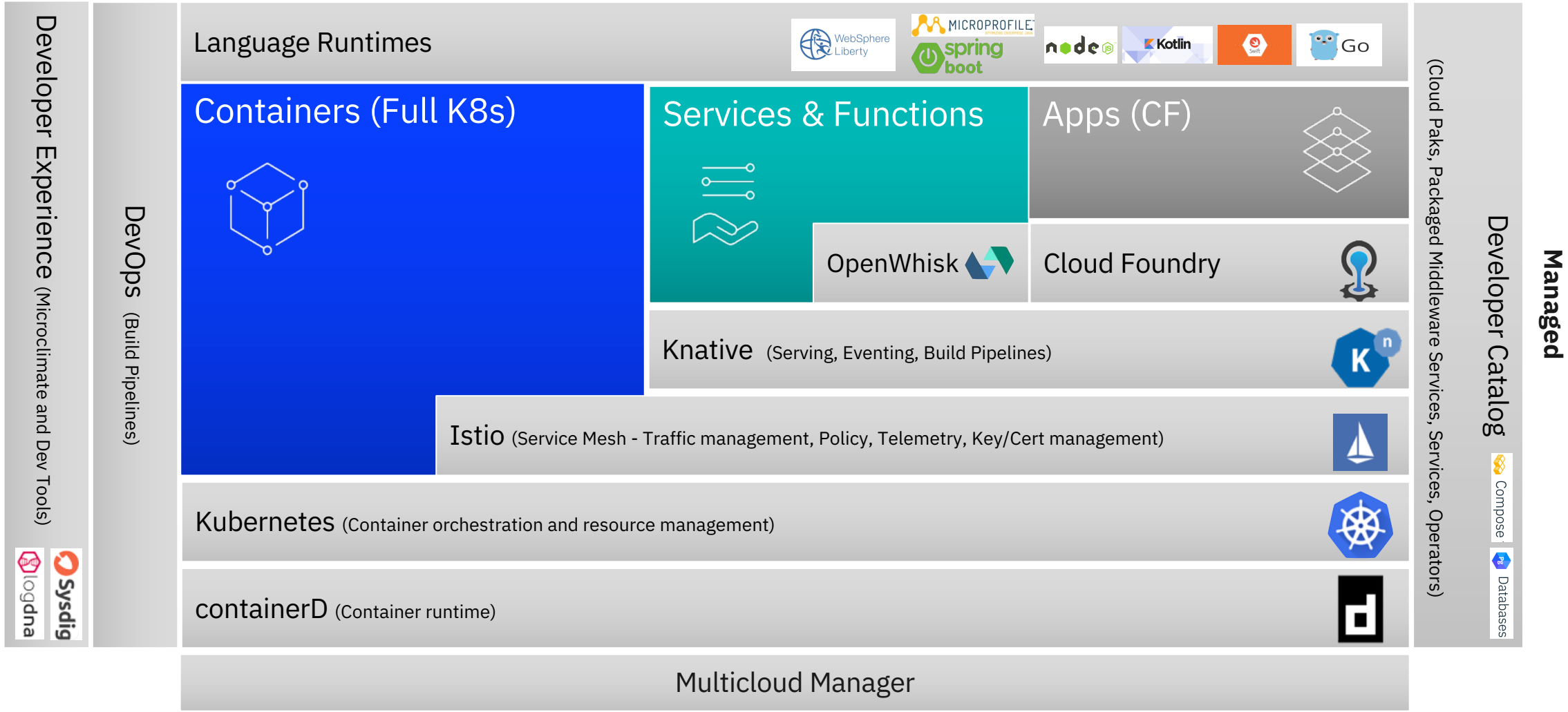
Modernization maturity level to Cloud-Optimized applications



Attributes of Cloud-Native Applications

1. Packaged as lightweight containers
2. Developed with best-of-breed languages and frameworks
3. Designed as loosely coupled microservices
4. Centred around APIs for interaction and collaboration
5. Architected with a clean separation of stateless and stateful services
6. Isolated from server and operating system dependencies
7. Deployed on self-service, elastic, cloud infrastructure
8. Managed through agile DevOps processes
9. Automated capabilities
10. Defined, policy-driven resource allocation

Industry Cloud Native Platform



IBM CODE

Private Cloud addresses enterprise use cases

Use Case #1

Modernize and optimize existing applications

Use Case #2

Opening up enterprise data centers to work with cloud services

Use Case #3

Create new cloud-native applications

Create Cloud-Native applications

Challenge

Enterprises want to create cloud native applications, but they have security and compliance challenges that prevent them from sending their data or services outside of the firewall.

Benefits

- Speed the development of applications and shrink time to market
- Rapidly provision capacity to meet demand
- Open container technology prevents vendor lock-in; consistency with public cloud

Cloud Native

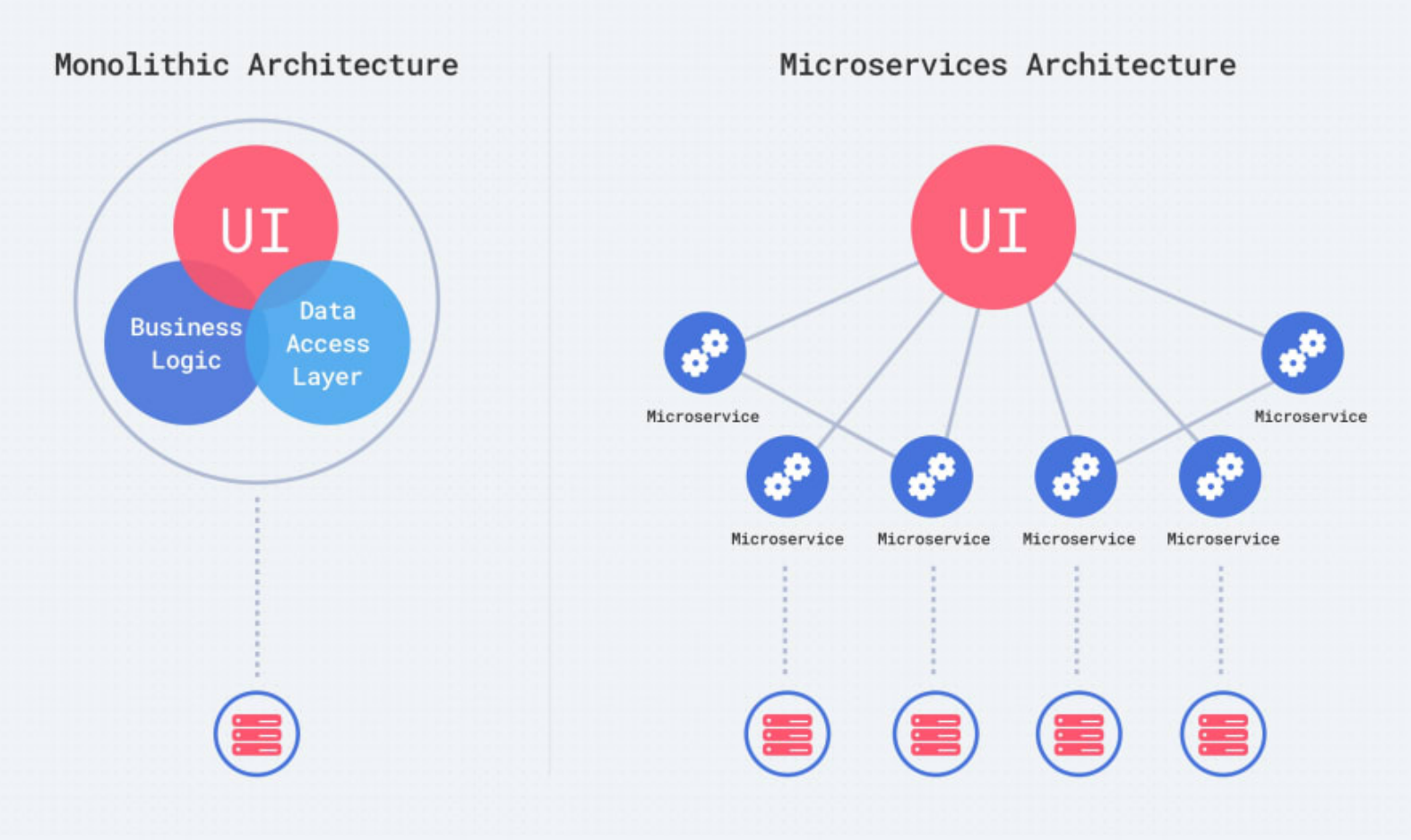
Platform
- Kubernetes
- Core services (Security, logging, monitoring, etc)
- Catalog of containerized content

IBM Enterprise Software
- Microservice Builder
- WebSphere Liberty
- IBM SDK for node.js
- Cloud Automation Manager

Optional Add-ons

Cloud Foundry

Cloud Native Development Vs Traditional Development



IBM CODE

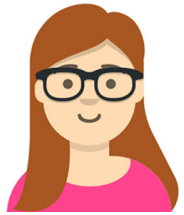
Cloud Native Development Vs Traditional Development

	Traditional App (Monolithic)	**Cloud Native (Microservices)**
Services	Coarse-grained services	Lightweight fine-grained services
Application Server or Centralized middleware platform	Application server or centralized middleware platform for deployment	Don't Require
Packaging	All services are deployed on the central middleware platform and hence they all share the same compute and storage	Services are autonomous at runtime. Every service has it's own compute and storage and is not shared between services
Tools for Development	IDE, are integrated with opensource and few vendor specific environments	These are getting developed to provide E2E environment. Fewer problems to address than complete application. Doing code review and QA is simpler
Deployment	Single point of failure during deployment and, if everything goes wrong, you could break your entire project.	Deploy each microservice independently, - orchestration tools, use CI/CD pipelines If something goes wrong, you will only break one small microservice

Helping Developers With Cloud Transition

Traditional skills

Java EE



Jane: Lead Enterprise Developer

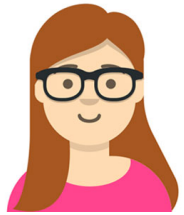
Champ: Solution Architect

Helping Developers With The Cloud Transition

Kubernetes can be complex with overwhelming choices for developers

Traditional skills

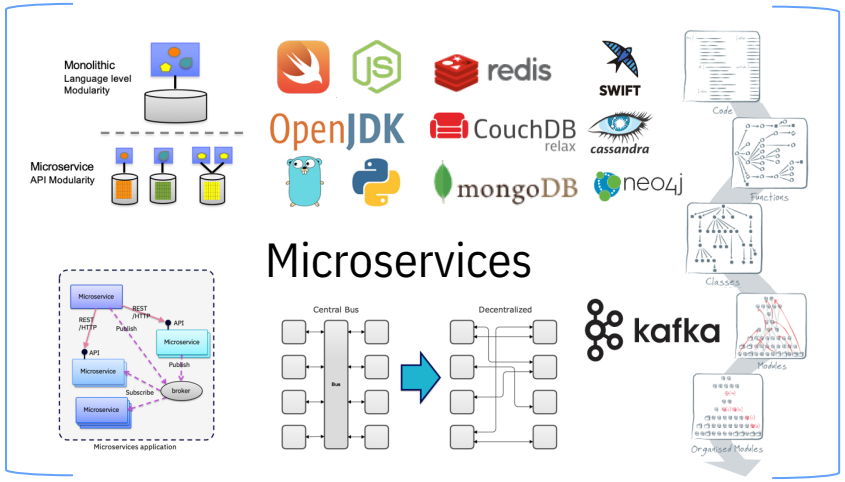
Java EE



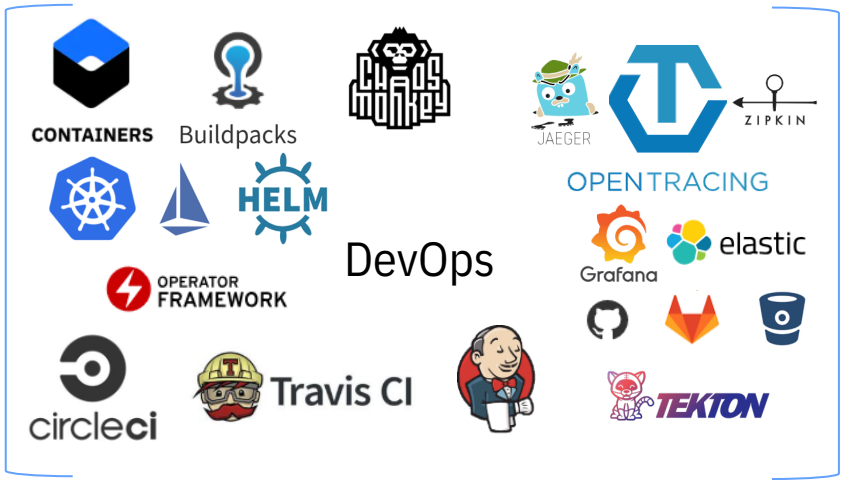
Jane: Lead Enterprise Developer

Champ: Solution Architect

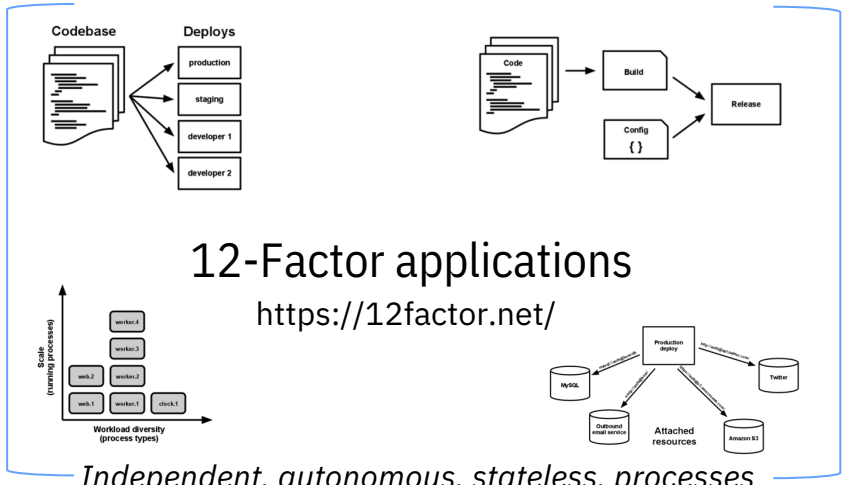
Today's challenges. Overwhelming freedom?



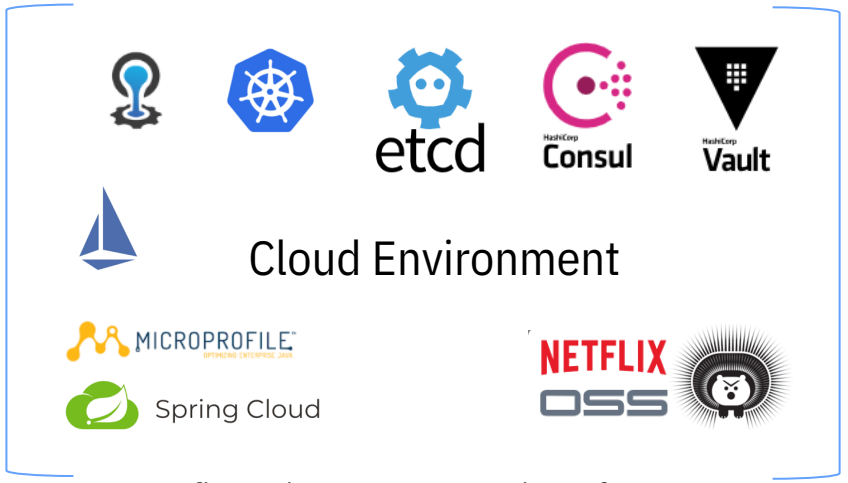
Application as composition of polyglot services



Build, package, deploy, observe



Independent, autonomous, stateless, processes



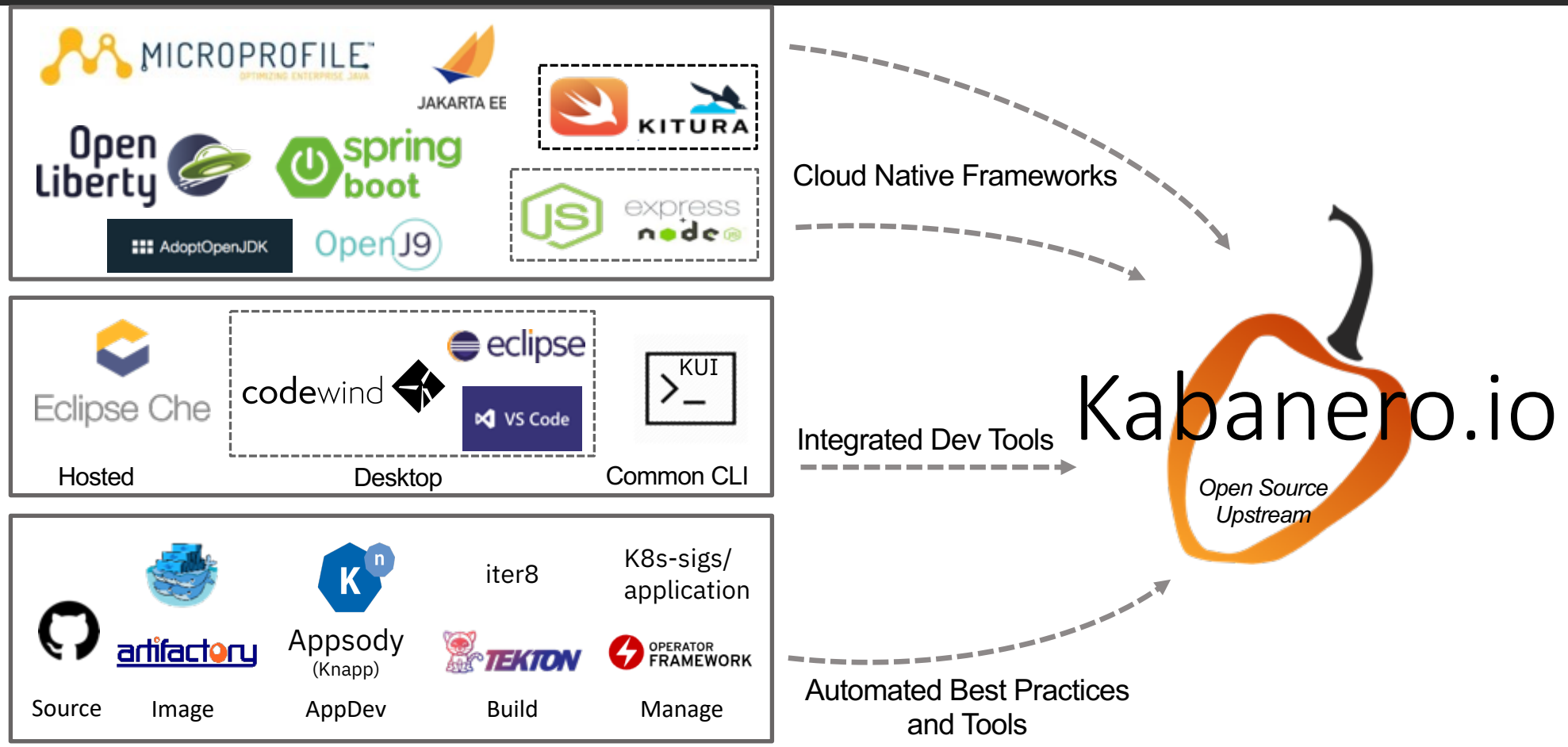
Configuration sources, service references

Cloud Native Development : Tooling Requirement

- Simplifies the development of applications that run on Kubernetes
- End-to-end solution that enables you to architect, build, deploy, and manage the lifecycle of Kubernetes-based applications, faster
- Architects and operations can include their company's standards for aspects like security and build pipelines customized for their developers use.

Introducing “Kabanero” to simplify Modernization/Cloud Native Kubernetes

Built for developers, scaled for the enterprise



IBM CODE

Modernize your applications, operations, and culture

Kabanero

Kabanero is focused on simplifying the task of architecting, developing, deploying, and managing cloud-native apps, using tailored application stacks and tightly integrated tooling that works in harmony with open source.

Enable developers to architect, build, deploy, and manage the lifecycle of Kubernetes-based applications, faster

Kabanero brings together architecture, development, and operations through the use of integrated collections and application stacks.

Kabanero is built on:

- Appsody
- Eclipse Codewind
- Knative
- Istio
- Tekton

- Eclipse Che

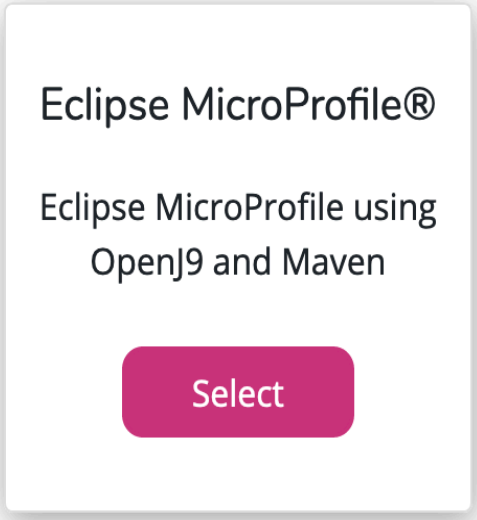
Appsody

Appsody provides Kabanero with simplicity in the creation of cloud-native applications in containers.

Appsody provides

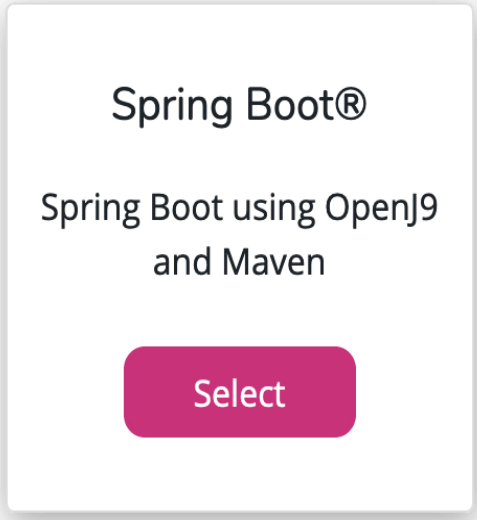
- pre-configured container images (stacks) and
- project templates for a growing set of popular open source runtimes and frameworks,
- providing a foundation on which to build applications for Kubernetes and Knative deployments.

Appsody : Application Stacks



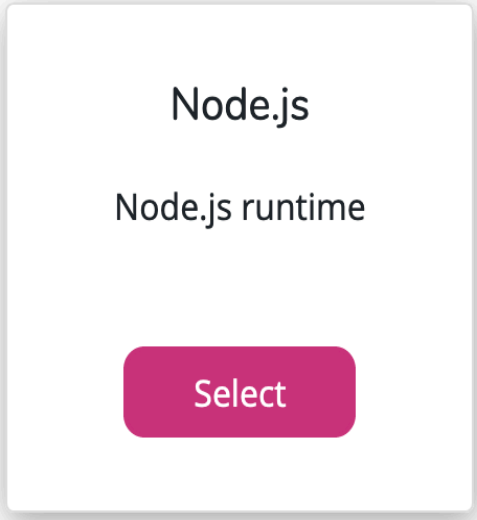
Deliver microservices based upon the Eclipse MicroProfile specifications.

Build and run high-quality applications with Open Liberty runtime, which includes OpenJDK with container-optimizations in OpenJ9.



Microservices based upon the Spring framework provided by Pivotal.

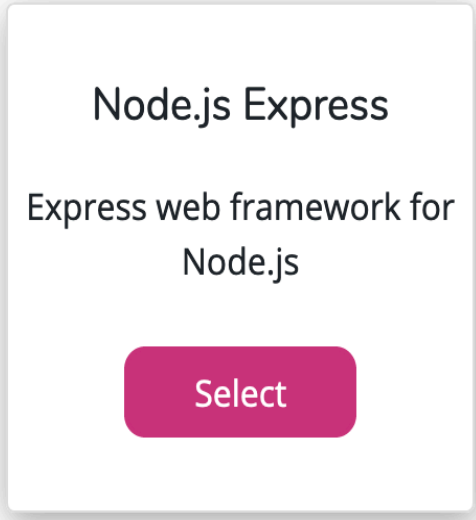
Open Liberty runtime provides support for Spring applications with OpenJDK and container-optimizations in OpenJ9.
.



JS based runtime, not limited to browser

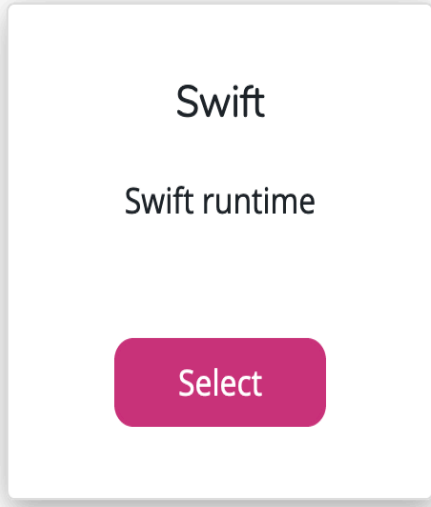
Single-threaded nature, Node.js enables the creation of non-blocking and event-driven servers.

Fits well in web-based frontends and backend API services for microservices-based applications..



JS based Web Framework.

Fits well in web-based frontends to handle routing on web based applications.
.



Swift is a very popular language for mobile application developers that develop for Apple Platforms.

Having the language on the server allows Apple Platform developers to produce the back-end portion of the mobile app as a microservice.

IBM CODE

Range of Developer Frameworks

- J2EE
  - For Java developers that was to continue with what they're familiar with

- Springboot/Tomcat
  - Used for >50% of Java microservices written today

- Microprofile
  - Alternative, open Java microservice framework

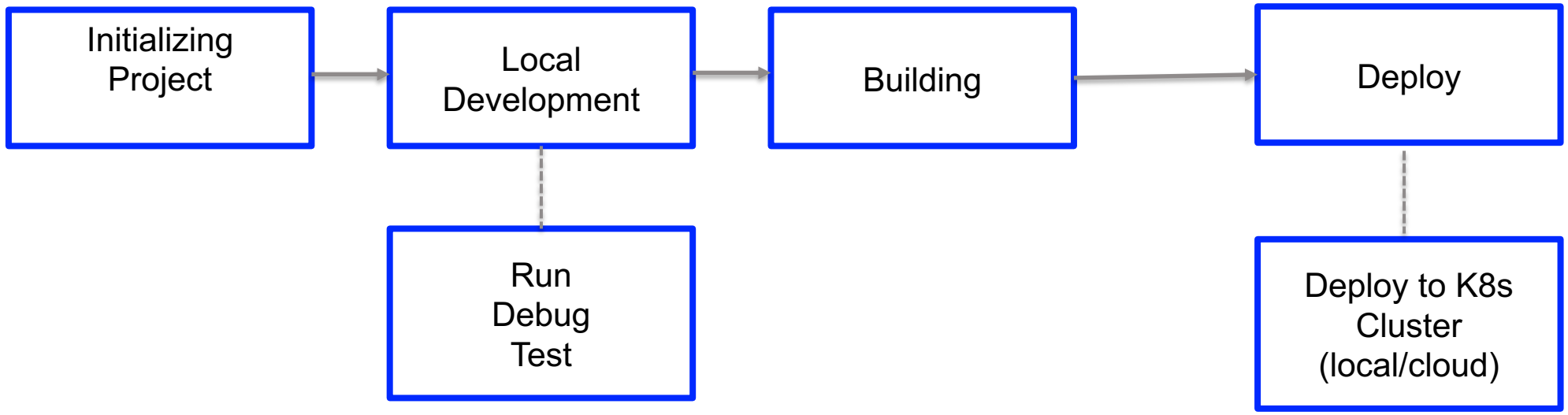
- Express
  - For node developers

- Kitura
  - For Swift developers creating backend iOS mobile applications

IBM CODE

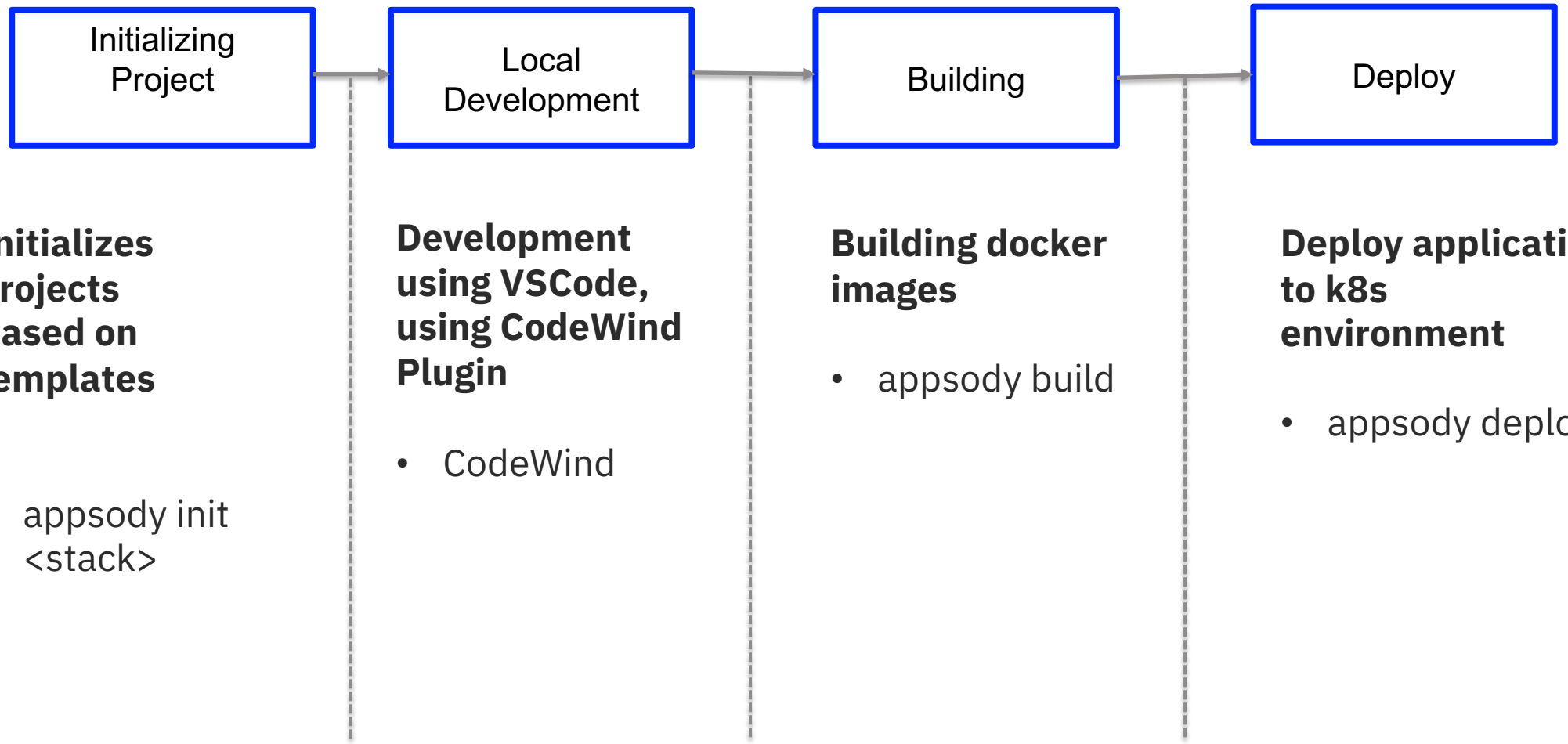
Appsody

Pre-req : Installed Appsody, based on your OS.



IBM CODE

Appsody


Initializing Project
Local Development
Building
Deploy
Initializes projects Based on templates
• appsody init <stack>
Development using VSCode, using CodeWind Plugin
• CodeWind
Building docker images
• appsody build
Deploy application to k8s environment
• appsody deploy

Why Appsody ?

CLI

Intuitive and powerful. The Appsody CLI allows you to connect to a Hub, pull down a stack, and create, build, test and deploy your application.

Stacks

Create, modify and share technology stacks with inbuilt cloud native capabilities, such as health checks, monitoring and OpenAPI descriptions.

Hub

A central repository of available stacks, enabling a single point of control for applications built from these foundations.

Built on open source

CodeWind

Codewind provides Kabanero with IDE integration and extensions to popular IDEs like VS Code, Eclipse, and Eclipse Che (with more planned).

This enables developers to use a workflow and IDE they already know.

You can rapidly iterate, debug, and performance test apps inside containers, just like when they run in production.

CodeWind

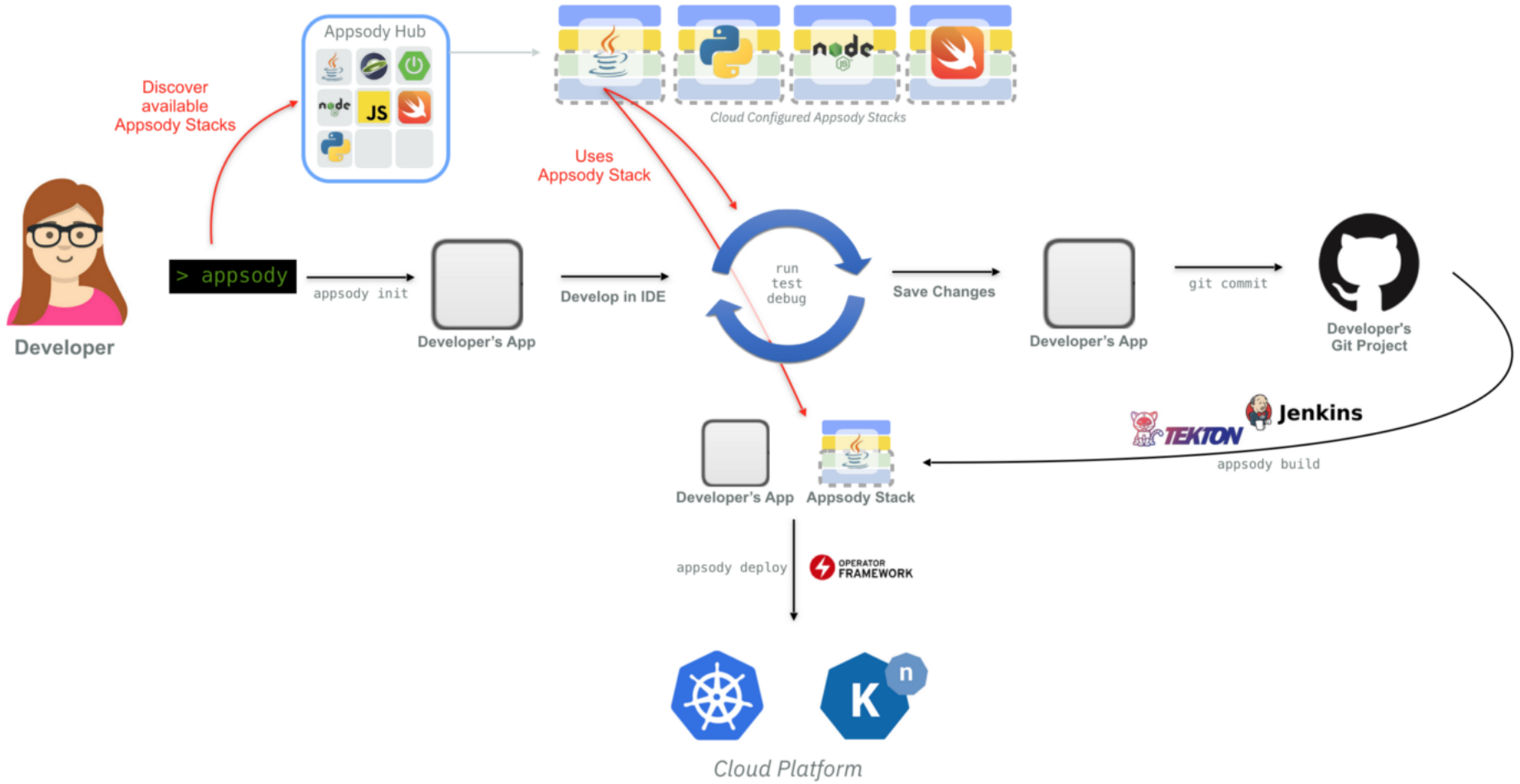
- **Lightweight** container development
- **Develop in containers** without feeling like you are developing in containers.
  Quick, iterative development and fast feedback directly in your IDE.
- **Leverage existing knowledge**
  Code in the tools you use today, for the runtimes and languages you use today.
- **Kubernetes** gently
  You don't need to be an expert in Kubernetes to deploy your apps!
  Get started quickly, and learn as you go.

IBM CODE

Demo :

1. AppSody
2. CodeWind

Developer Flow



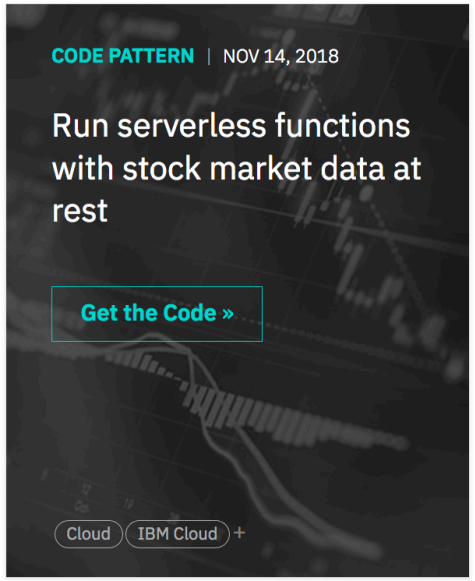
Hands On:**Appsody**

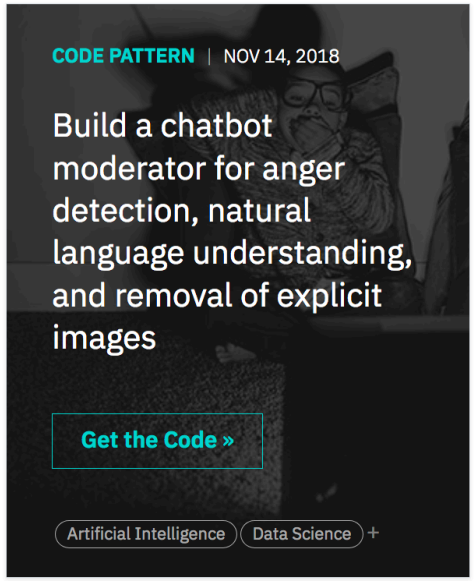
http://bit.ly/cloudnativews

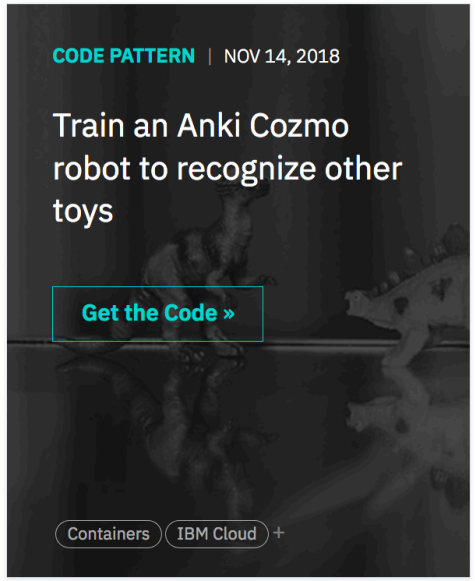
IBM CODE

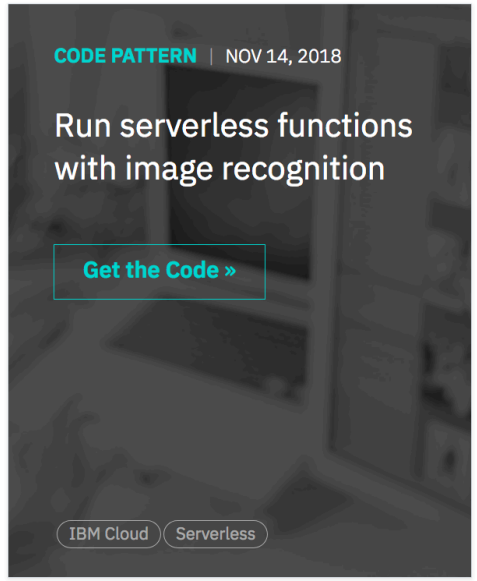
Code Patterns

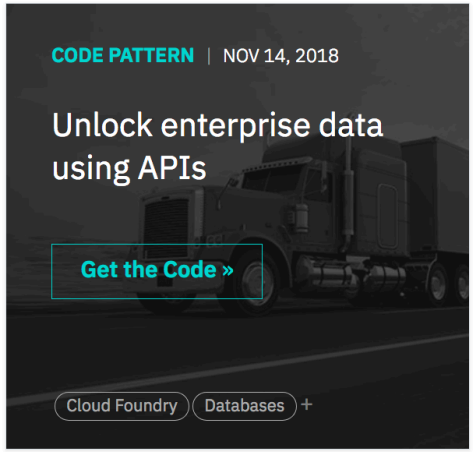
https://developer.ibm.com/technologies/containers/

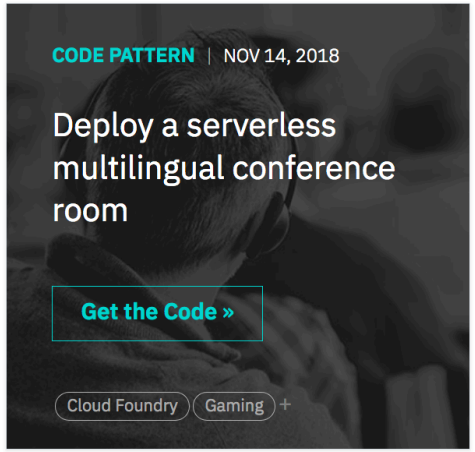


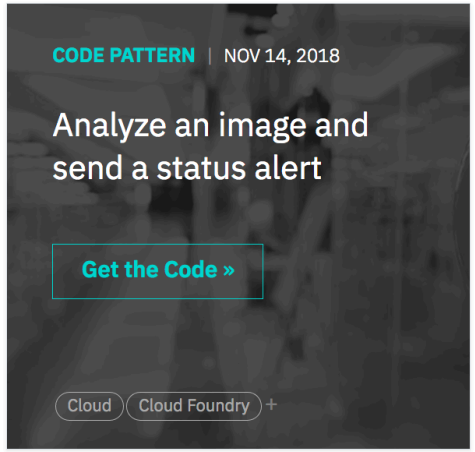


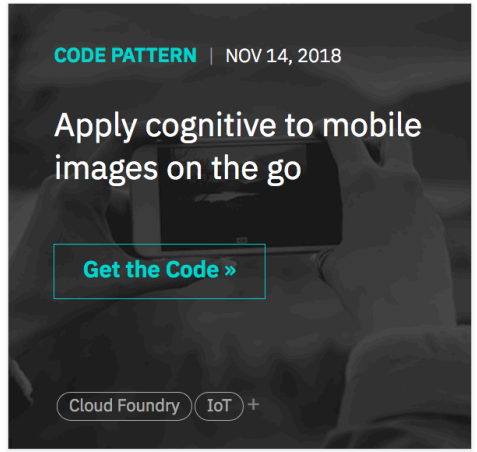










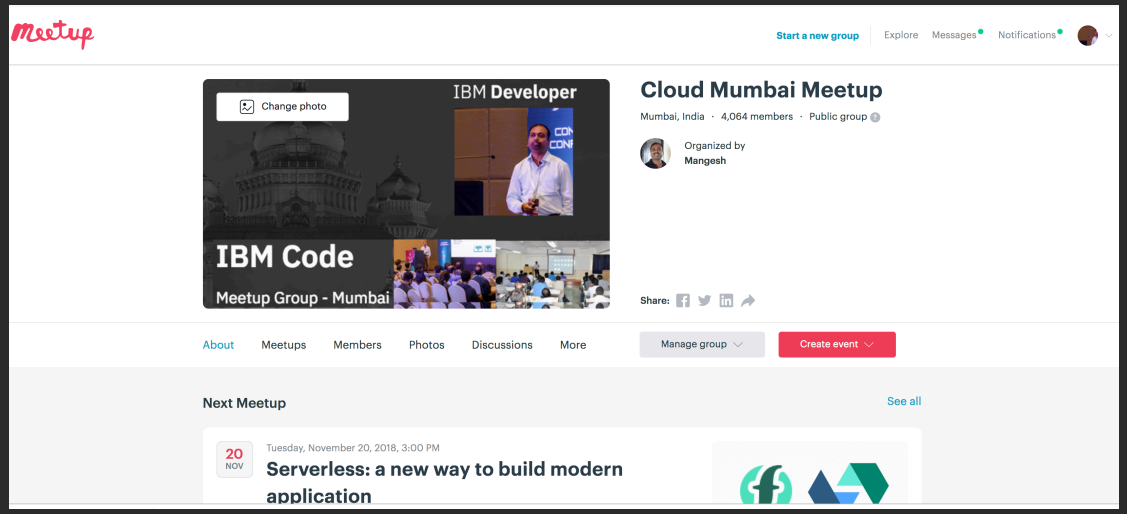


IBM CODE

IBM Cloud Registration

Register on IBM Cloud

- bit.ly/ibmcloudsign



@MangeshPatank

/mdpatankar

in/mangesh-patankar-1961a019

mapatank@in.ibm.com

IBM CODE