

A Flotilla of open-source tools for container-based application development and deployment

—
Erin Schnabel
@ebullientworks

Cloud Native

Microservices architecture

Containerized services

Elastic scaling

Orchestration

Orchestrating all the things...

Cloud Environment

- Dynamic infrastructure
- Flexible capacity



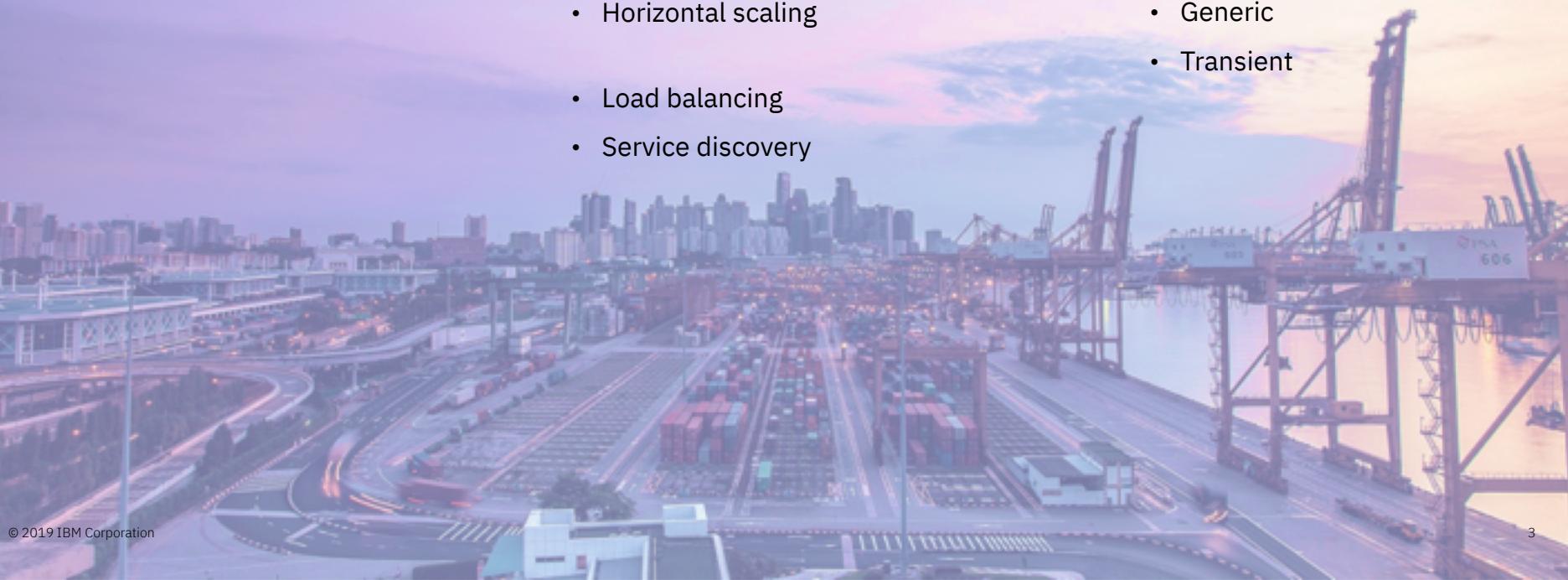
Container orchestration:

- System health (self-healing)
- Workload scheduling
- Horizontal scaling
- Load balancing
- Service discovery



Containerized workloads

- *Simplified operations*
- Resource isolation
- Generic
- Transient

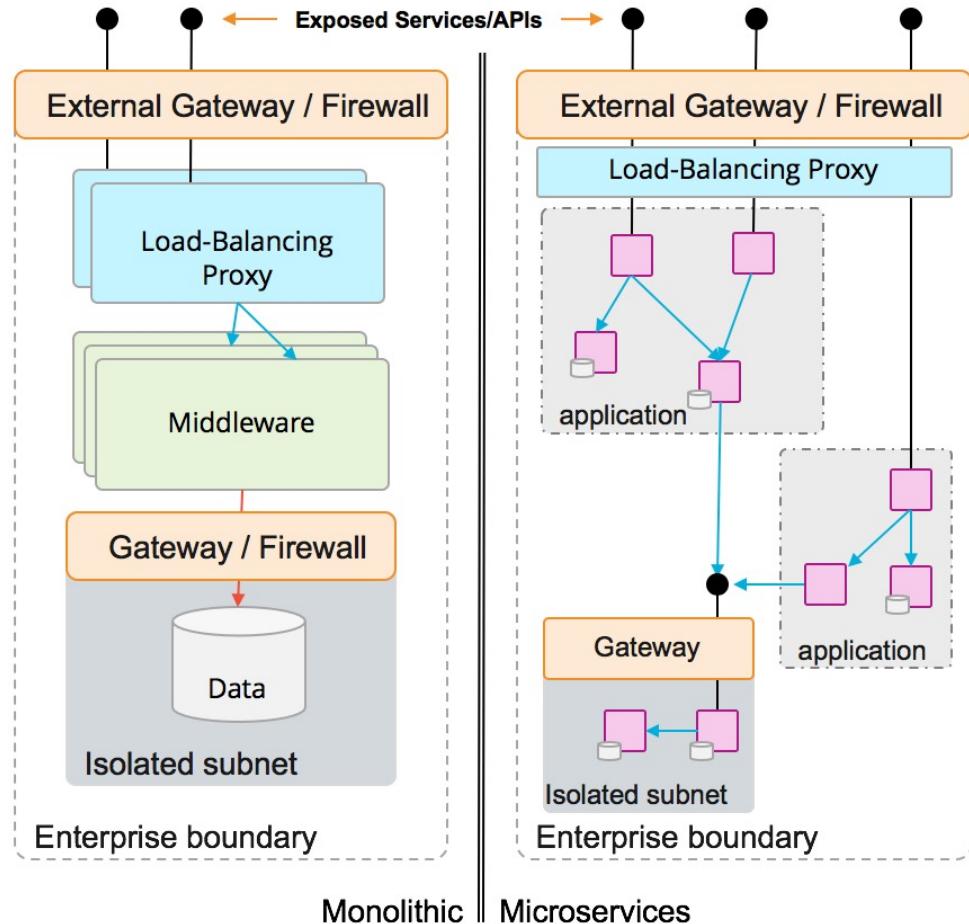


We want to move workloads into the cloud...

What is getting in our way?



Monolith VS. Microservice



Cloud adoption challenges

Technology change

People: Culture & roles

Control & governance

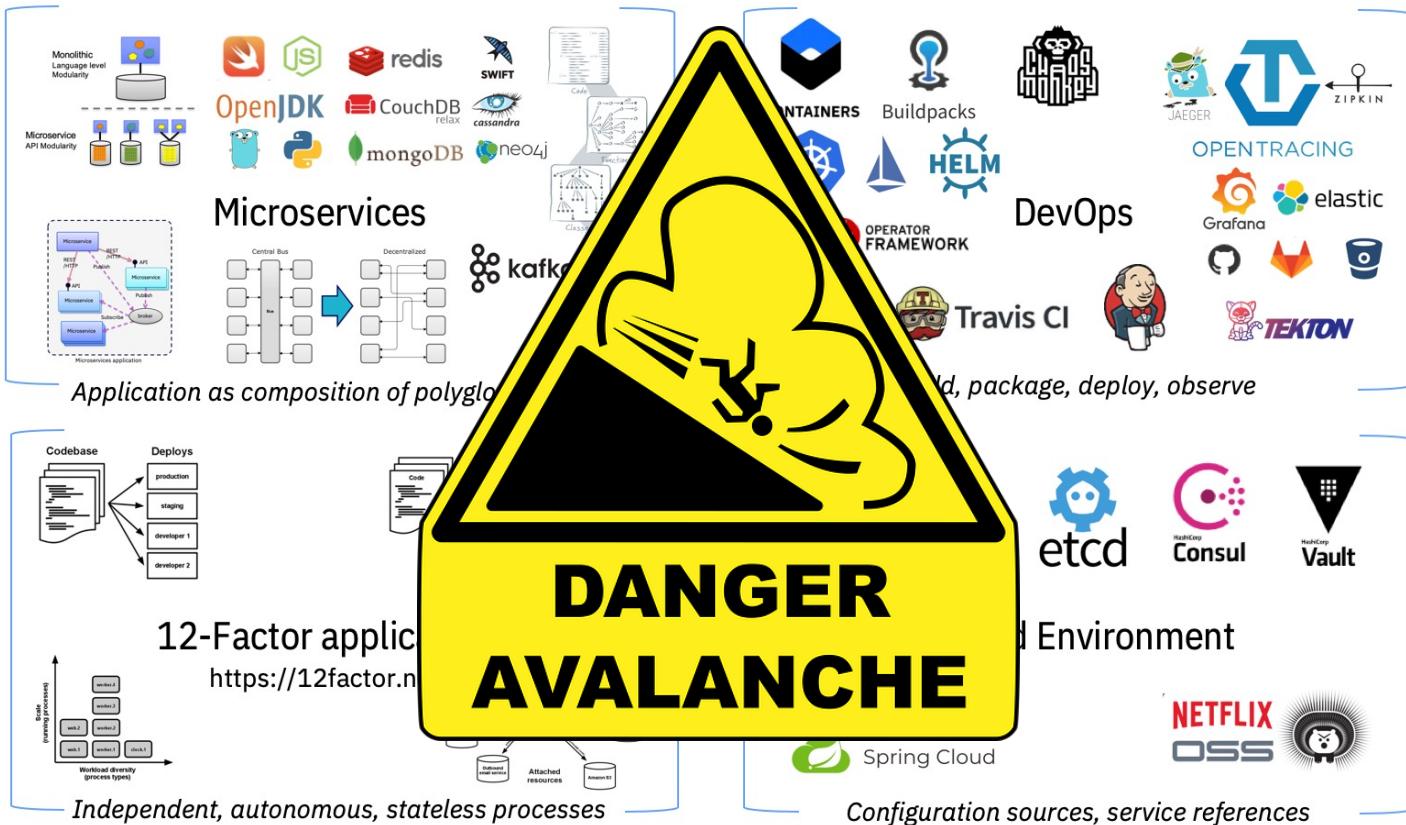
Costs

Security

Challenge: the skill gap

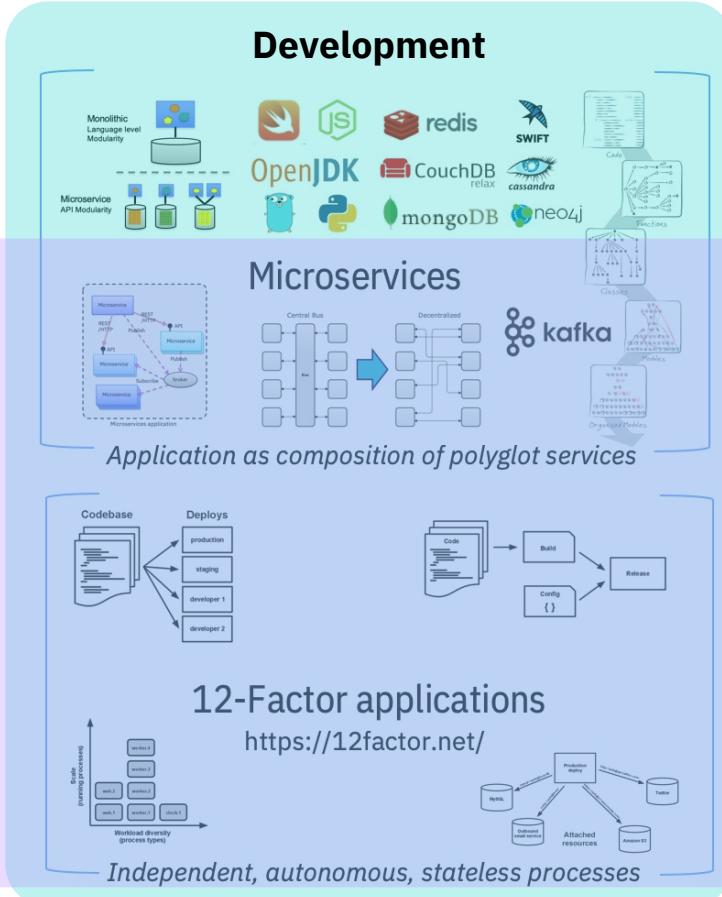


Java EE

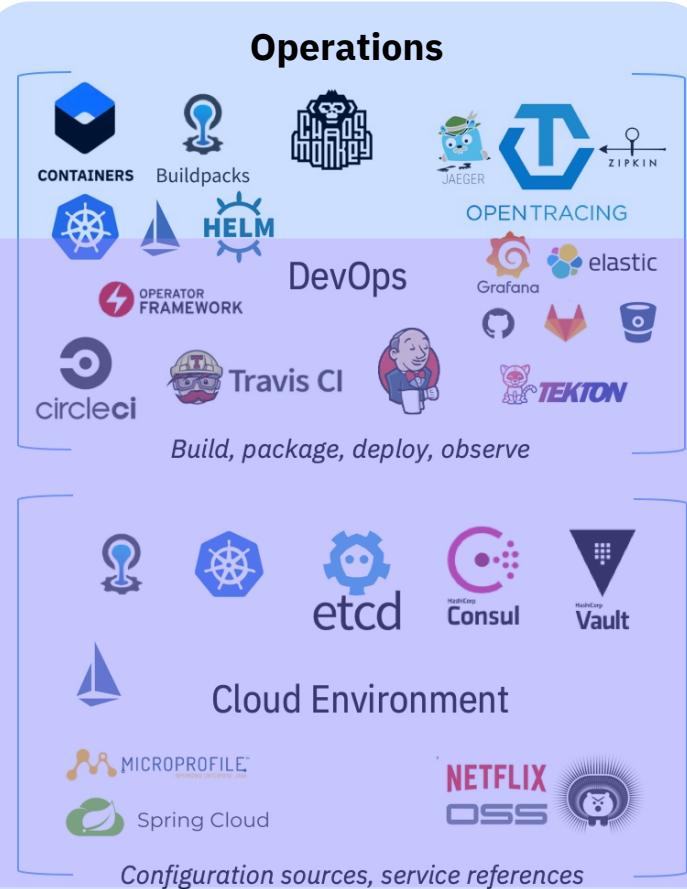


Challenge: role expansion

Cloud Native DevOps



Operations



Challenge: Control & Governance

Changing regulatory requirements

Corporate standards & required behaviors

Coding conventions

- Security practices
- Resilience / Fault tolerance practices
- Monitoring conventions
 - Distributed trace
 - Preferred metrics collection (push or pull)

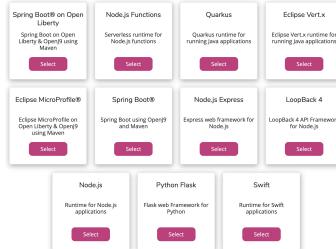




appsody



> appsody



appsody: CLI

Containerized iterative development
run | test | debug | build | deploy

appsody: stacks

Sharable pre-built, cloud-optimized
technology stacks

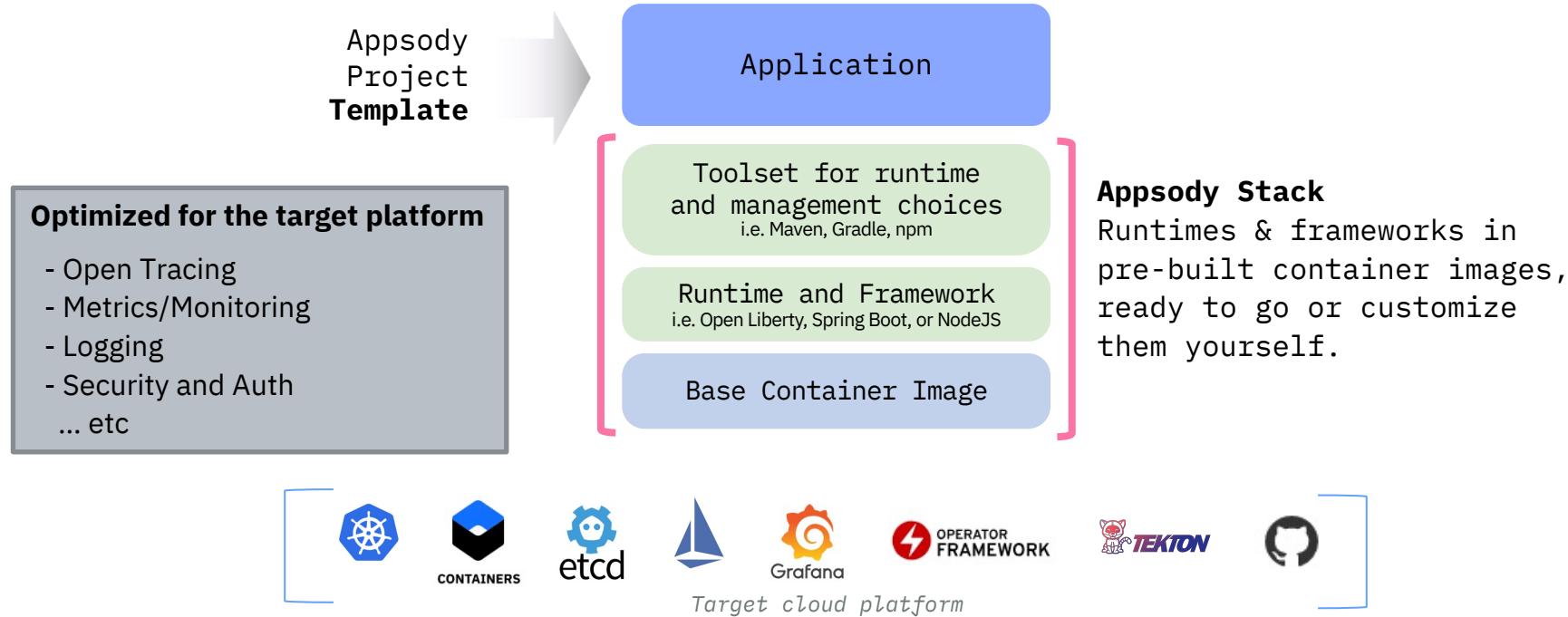
appsody: deploy

Production deploy to Kubernetes
Server or Serverless Scaling



appsody

Stack concept





appsody

Stack value

An appsody stack is more than a docker image in a registry.

A stack provides:

- (1) organizational best-practices for **coding**.

Templates for getting started

Stacks for ongoing (painless) enforcement

- (2) optimization of **development** and **deployment** behavior

Stack maintainer streamlines stack behavior (run, test, debug, build, deploy),

- simplify/streamline development process
- standardize deployments (reusable tasks)

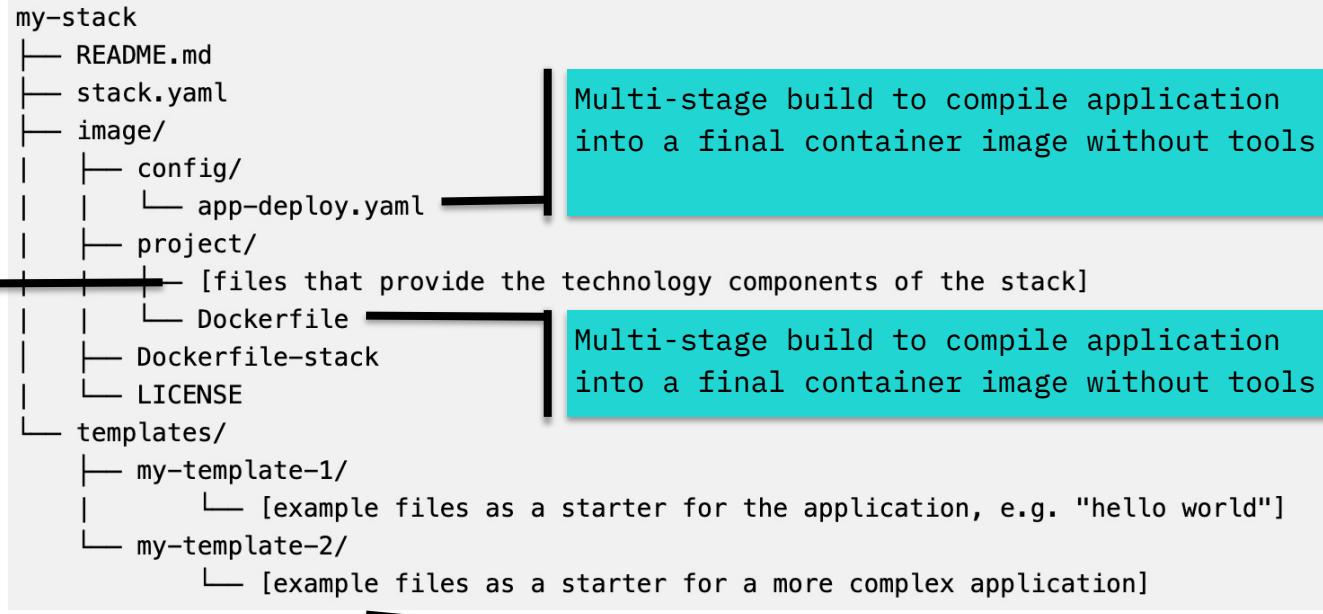


Two images:

- Development (stack)
- Deployment

Parent pom.xml
- spring actuators
- DevTools plugin

build scripts
- validation
- consistency
- optimize behavior



Customized templates for common patterns



appsody

Developer workflow

> **appsody list**



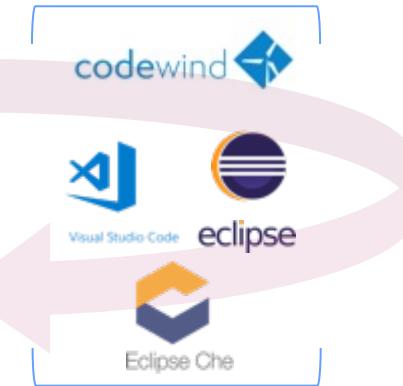
> **appsody init**



Stack + template



application



> **appsody run**



> **appsody debug**

> **appsody test**



> **appsody build**



TEKTON

OPERATOR
FRAMEWORK

> **appsody deploy**





appsody

Stack evolution

> appsody list



> appsody init



> appsody run

> appsody debug

> appsody test

> appsody build

> appsody deploy

Evolution of Appsody stacks

Updated stack published (versioned)

Updated stack is automatically picked up in development and build processes that use the appsody CLI





appsody

CLI workflow

```
erebor{elh} ~/codeone $
```

Eclipse Codewind

Extensions to industry standard IDEs to support
cloud-native development for Kubernetes

Rapidly create, test, debug, and extend
applications running in containers



Codewind's Developer Tools

Create projects from extensible list of templates

Project info – container ports, endpoints, etc.

Container build & runtime status

Console output directly in the IDE

Shell into running containers

Debugging

...

The screenshot shows a dark-themed interface for managing a project named 'appsody-node-express'. At the top right is a blue 'Build' button. Below it, the project details are listed:

Type:	Appsody
Language:	Nodejs
Project ID:	374300e0-d0ca-11e9-9e29-cf9664898a12
Container ID:	60652b15d9ec6dacdb490ef78e9e55e5
Location on Disk:	/Users/deboer/codewind-workspace/appsody-node-express
Auto build:	<input checked="" type="checkbox"/>
Application Status: Running	
Build Status:	Build Succeeded
Last Image Build:	9/6/2019 at 4:46:55 PM
Last Build:	9/6/2019 at 4:46:55 PM
Exposed App Port:	32784
Internal App Port:	3000
Application Endpoint:	http://127.0.0.1:32784/
Exposed Debug Port:	Not debugging
Internal Debug Port:	9229

At the bottom are two buttons: 'Remove project' (red) and 'Disable project' (blue).



EXPLORER

∨ OPEN EDITORS

∨ NO FOLDER OPENED

You have not yet opened a folder.

Open Folder



∨ CODEWIND

∨ Codewind

∨ Projects (Local)

🚫 No projects (Click here to ...)



Show All Commands ⌘ P

Open File or Folder ⌘ O

Open Recent ⌘ R

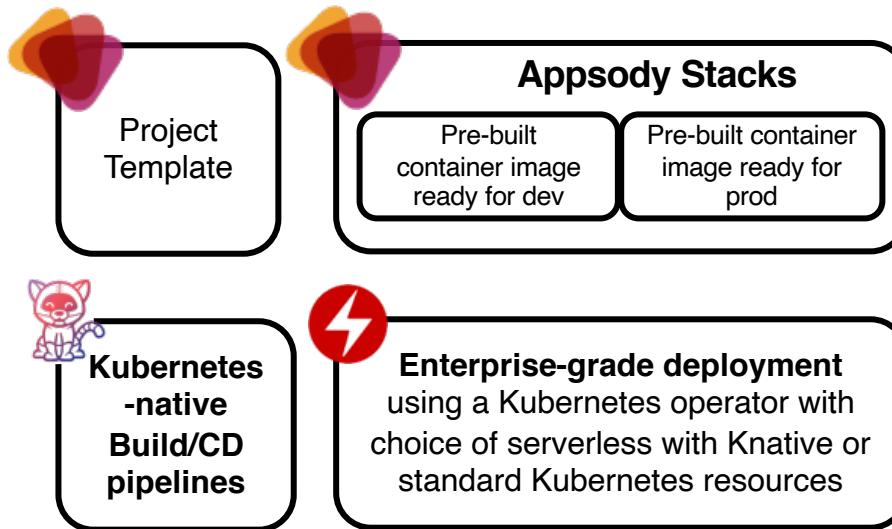
New Untitled File ⌘ N

> OUTLINE

> MAVEN PROJECTS



Everything you need to create an app or microservice
Customized to meet your enterprise platform needs



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

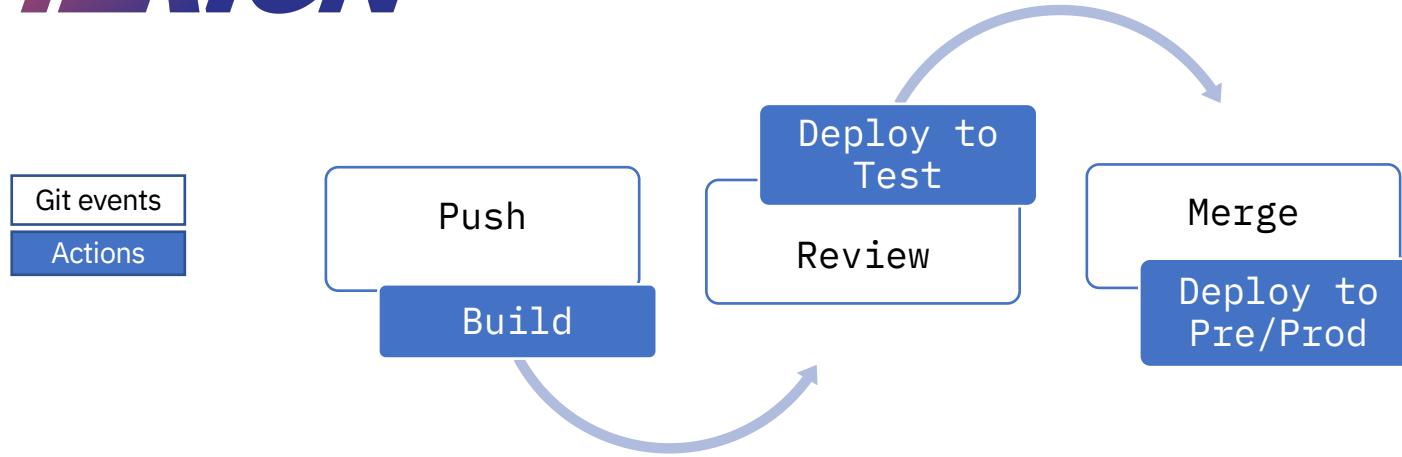


DevOps is critical for a successful microservices-based deployment.

Kabanero supplies an integrated open source CI/CD (continuous integration / continuous delivery) infrastructure based on the open source project Tekton



TEKTON



Pipelines



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



A multi-cluster continuous delivery tool for Kubernetes

Automate the rollout process of Kubernetes resources across multiple clusters, environments, and cloud providers, and gain insight into what applications and versions run in your cluster.

Thank you!



appsody

<https://appsody.dev>



TEKTON

<https://tekton.dev>

