



Cloud Native Dev Tools:

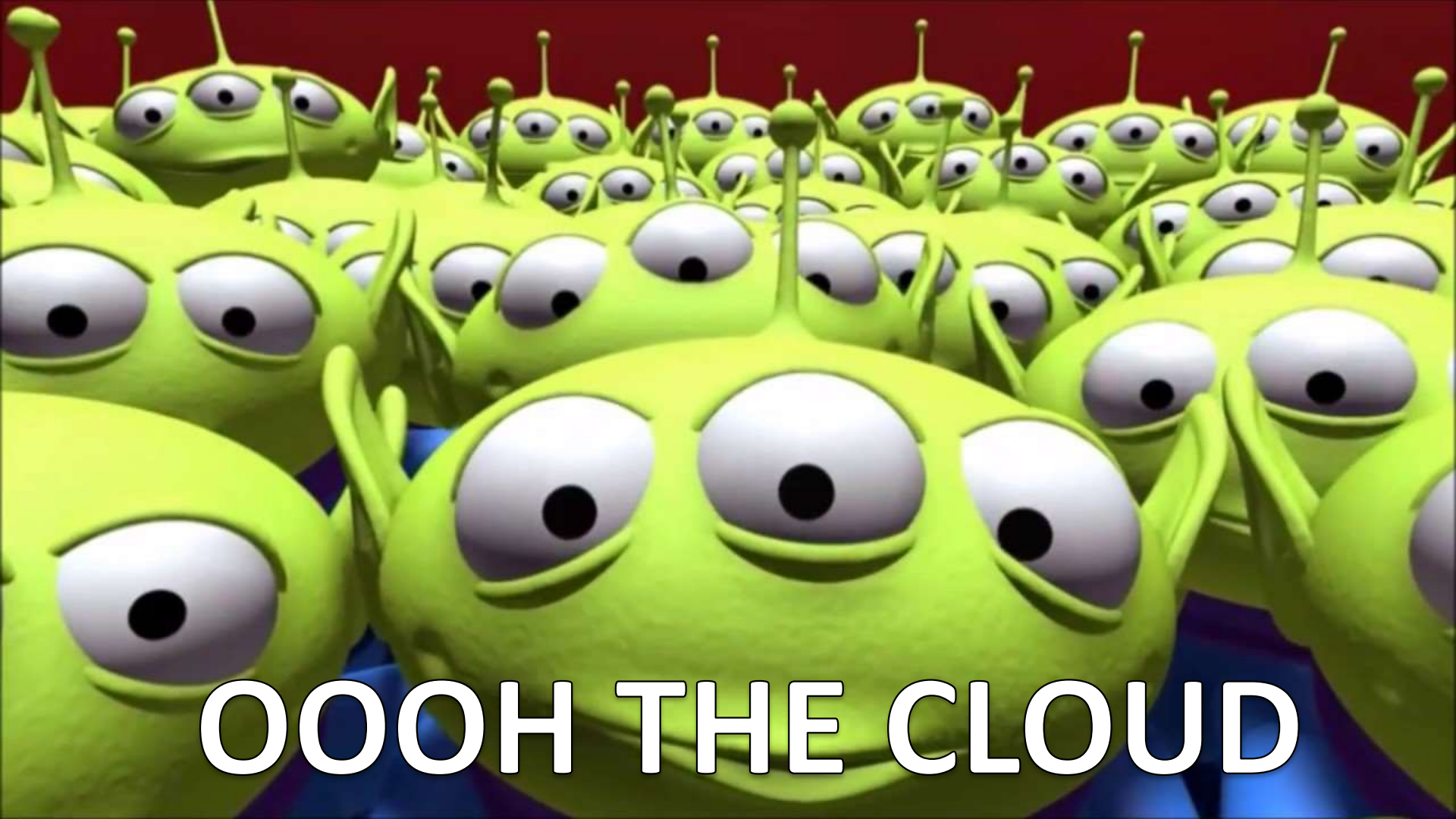
Bringing the cloud
back down to earth



GRACE JANSEN
DEVELOPER ADVOCATE,
IBM



WHY?



OOOH THE CLOUD



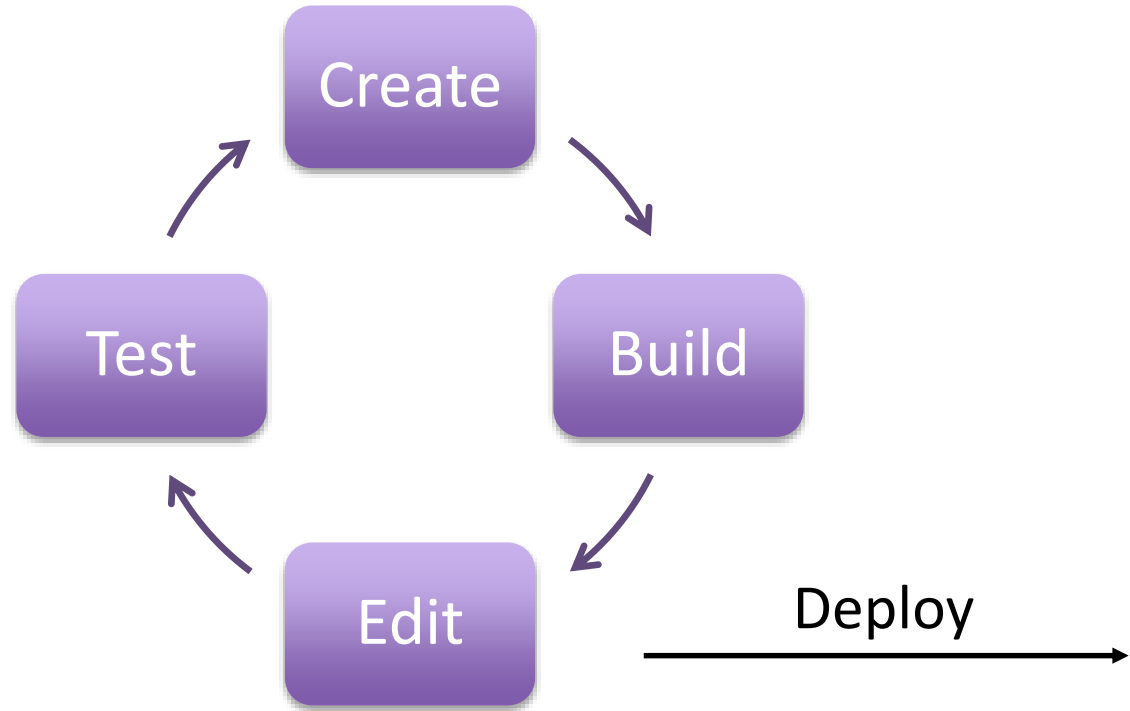
WHAT?

Developers?

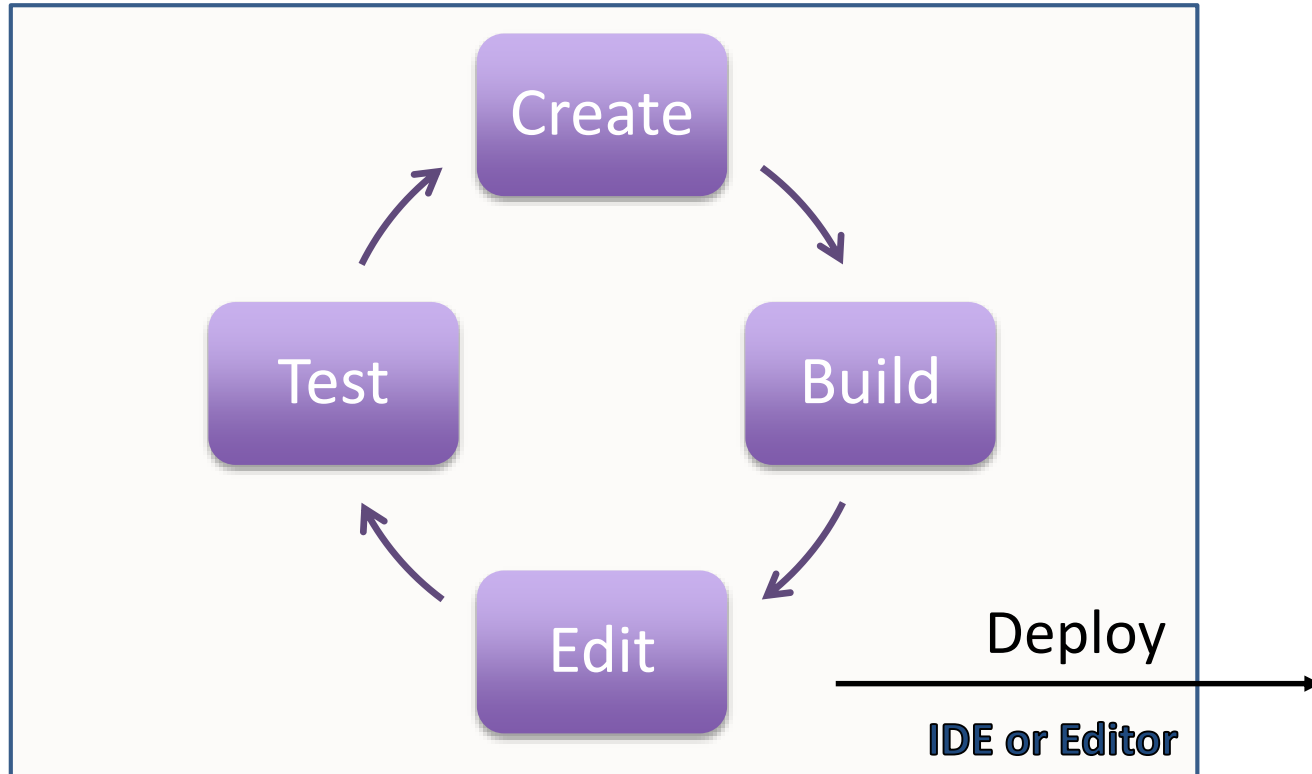
- Create a (new) project
 - Add or edit code
 - Build (and package and run)
 - Test code
 - Write tests
 - ... and repeat
 - DevOps - Deployment
-
- Coding assistance when needed
 - Choice of editors and IDEs



Iterative Development Cycle



Iterative Development Cycle





HOW?

Beam Me Up
Scotty!



Bring The Cloud Back Down To Earth



OSS Java Technologies





<https://openliberty.io>

Open Liberty™

An IBM Open Source Project

A lightweight open framework for building fast and efficient cloud-native Java microservices.

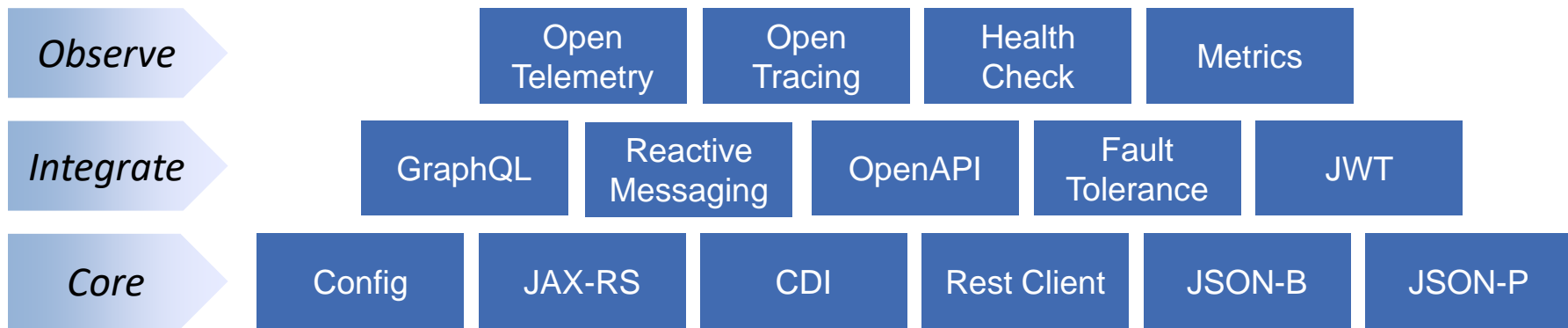
Build cloud-native apps and microservices while running only what you need. Open Liberty™ is the most flexible server runtime available to Java™ developers in this solar system.

[Get Open Liberty](#)





<https://microprofile.io/>



Open cloud-native Java APIs

JAKARTA EE

BUILDING AN OPEN SOURCE ECOSYSTEM FOR CLOUD NATIVE JAVA

Jakarta EE brings together a global community of leading developers, vendors, and enterprises to collaborate on advancing enterprise Java technologies in the cloud.



GET INFORMED

GET THE
SPECIFICATIONS



GET INVOLVED

JOIN THE
WORKING GROUP



GET STARTED

STARTER FOR JAKARTA EE
DOWNLOAD
COMPATIBLE PRODUCTS
RELEASES



Specifications

Jakarta EE Platform

Defines a platform for hosting Jakarta EE applications

Jakarta EE Web Profile

Web profile of platform for hosting Jakarta EE applications

Jakarta EE Core Profile

Core profile of platform for smaller runtimes

Jakarta Activation

Defines standard services for MIME type data access

Jakarta Annotations

Declarative style annotations for common semantic concepts

Jakarta Authentication

Low-level container SPI for authentication mechanisms

Jakarta Authorization

Low-level container SPI for authorization modules

Jakarta Batch

API plus an XML-based batch job specification language

Jakarta Bean Validation

Object level constraint declaration and validation facility

Jakarta Concurrency

API for app level concurrency that integrates with container

Jakarta Config

Standardizes externalized configuration

Jakarta Connectors

Architecture for connecting apps to enterprise info systems

Innovate with Us

[About Jakarta EE](#)

[Frequently Asked Questions](#)

[Become a Member](#)

[Stay Connected](#)

[Working Group Charter](#)

[Technical Direction](#)

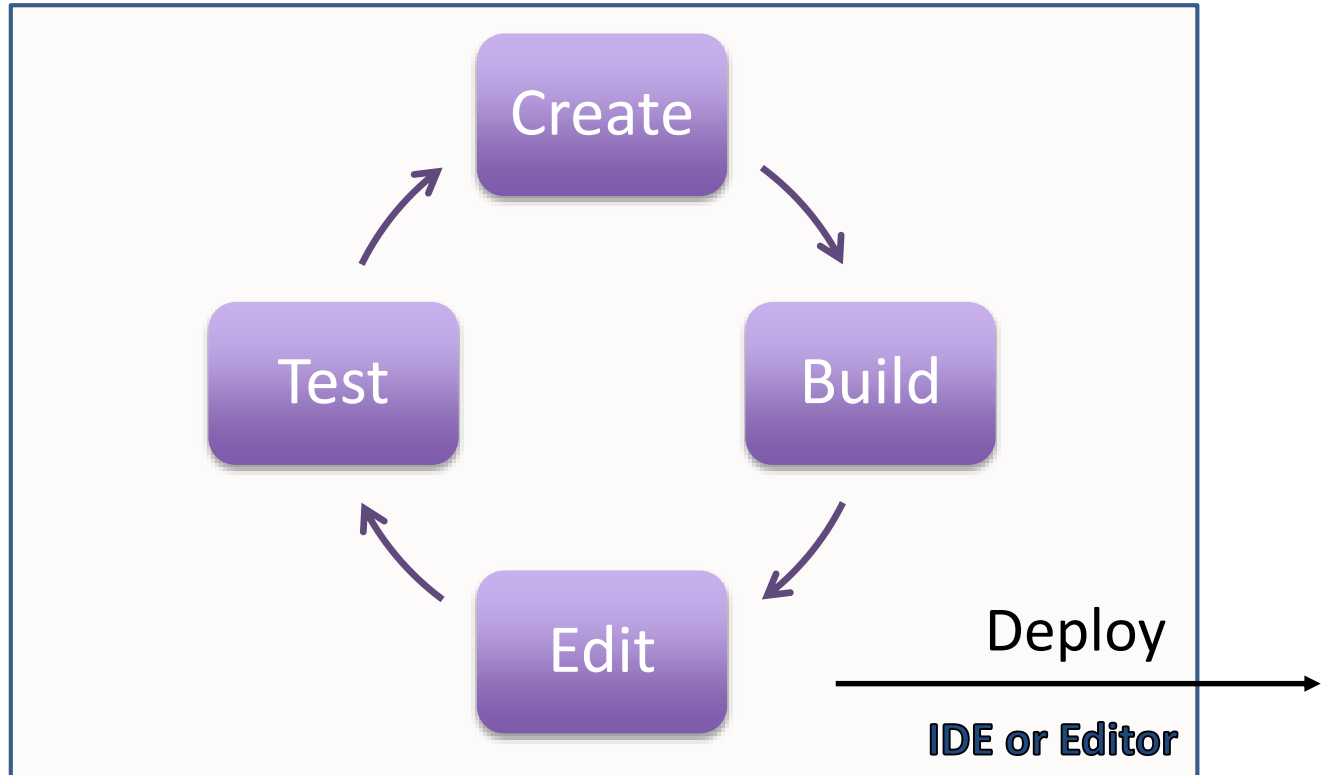
[Newsletter Archives](#)

[Meeting Minutes](#)

[Trademark Guidelines](#)

[Specification Process](#)

Iterative Development Cycle





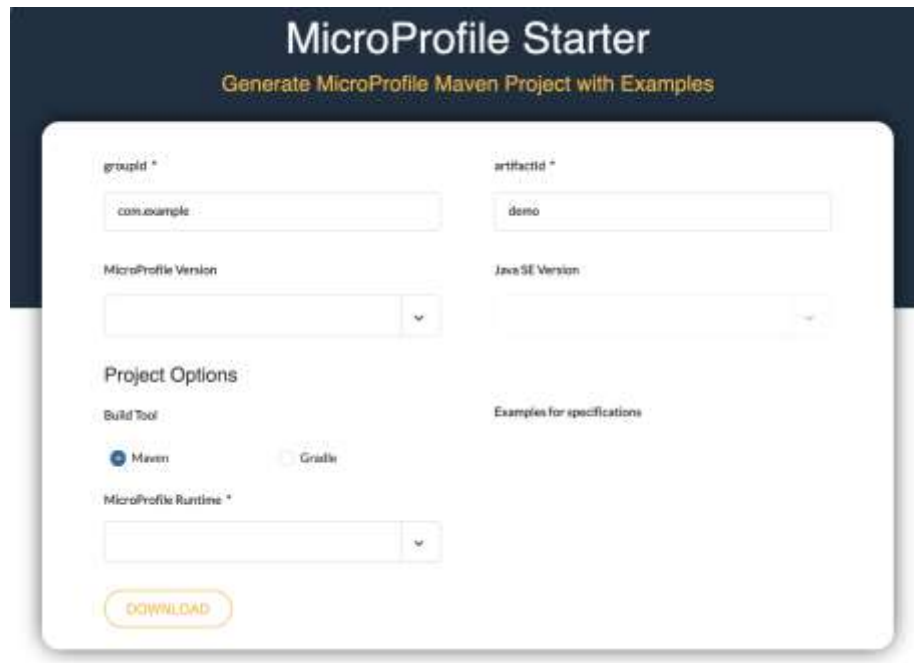
CREATE

Starters



Give me a starter project or template for my MicroProfile application

- start.microprofile.io
 - VS Code extension
 - IntelliJ plugin



The screenshot shows the 'MicroProfile Starter' web application. The title is 'MicroProfile Starter' with a subtitle 'Generate MicroProfile Maven Project with Examples'. The form contains the following fields and options:

- groupid ***: Text input field containing 'com.example'.
- artifactid ***: Text input field containing 'demo'.
- MicroProfile Version**: Dropdown menu.
- Java SE Version**: Dropdown menu.
- Project Options**: Section header.
- Build Tool**: Radio buttons for 'Maven' (selected) and 'Gradle'.
- MicroProfile Runtime ***: Dropdown menu.
- Examples for specifications**: Text input field.
- DOWNLOAD**: A yellow button at the bottom.

Starters

Give me a starter project or template for my Jakarta EE application

- start.jakarta.ee
- Runtime-specific starters/generators/templates
 - Open Liberty Starter
 - start.openliberty.io



The Jakarta EE Tutorial is a comprehensive guide to developing enterprise applications for the Jakarta EE Platform.

Try it out!

Download and unzip Eclipse GlassFish

```
[glassfish-install-dir]$ ./bin/asadmin start-domain
```

Get started with Open Liberty

Whether trying or updating Open Liberty, we've got you covered!

Create a starter application

Select the development tools that you prefer to use, then generate a package to start developing your application.

Group: Artifact:

Build tool: Maven Gradle

Java SE Version: Java EE/Jakarta EE Version: MicroProfile Version:

BUILD



Dev Mode



What if I don't have to build, package, deploy and start my application manually during development?

- Liberty Maven/Gradle Plugin
- Dev mode for hot deploy
 - Continuous/hot testing
 - Container support
 - Debugging

```
<project>
  ...
  <build>
    <plugins>
      <!-- Enable liberty-maven-plugin -->
      <plugin>
        <groupId>io.openliberty.tools</groupId>
        <artifactId>liberty-maven-plugin</artifactId>
        <version>3.6.1</version>
        <!-- Specify configuration, executions for libe
        ...
      </plugin>
    </plugins>
  </build>
  ...
</project>
```

```
mvn liberty:dev
```

```
mvn liberty:devc
```

WAD - Watch and Deploy

Open Liberty



AdamBien / wad Public

Watch 16 Fork 25 Star 109

Code Issues Pull requests Actions Projects Security Insights

master 2 branches 11 tags

Go to file Add file Code

AdamBien Merge pull request #12 from franden/generic_war_name ✓ a8e7c4f on Jul 17, 2020 121 commits

_layouts	link changed, contrast increased	4 years ago
assets/css	header changed	4 years ago
src	the name of WAR file will be determined dynamically, according to ...	2 years ago
.gitignore	release script excluded	4 years ago
CNAME	Create CNAME	4 years ago
LICENSE	Initial commit	4 years ago
README.md	Installation section added	3 years ago
pom.xml	version updated	3 years ago

README.md

watch and deploy (WAD)

WAD watches changes in `src/main/java` folder, builds the project using the `pom.xml` and deploys the ThinWAR into the configured folder.

Installation

About

Watch and Deploy - deploying ThinWARs without IDE plugins

wad.sh

java utility deployment watch tomcat speed jetty wildfly jbase jboss payara tomee microprofile wad openliberty sparksee fireware

Readme MIT license 109 stars 16 watching 25 forks

Releases 11

substitutor Latest on Oct 25, 2019 + 10 releases

Barbanas

EDIT

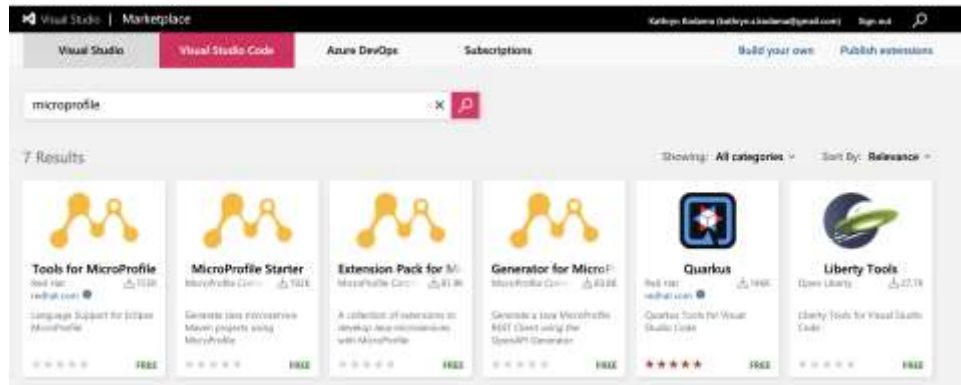


IDE or Editor Integration



Working with MicroProfile API and runtimes inside my favorite editor or IDE

- Eclipse
 - IntelliJ
 - VS Code
 - NetBeans ...
-
- IDE integration for runtime lifecycle management
 - Custom runtime plugins or extensions



Coding Assistance in Editor/IDE



Can I get help with working with MicroProfile APIs inside my favorite editor or IDE?

- Language Server for Eclipse MicroProfile incubator project at Eclipse Foundation
- LSP4MP-based VS Code extension, Tools for MicroProfile from Red Hat

Eclipse LSP4MP - Language Server for Eclipse MicroProfile

build passing License EPL 2.0

A language server implementation based on the [Language Server Protocol for Eclipse MicroProfile](#). This Language Server for Eclipse MicroProfile (LSP4MP) provides core language support capabilities (such as code complete, diagnostics, quick fixes) to enable developers to quickly and easily develop applications using MicroProfile APIs.

This project contains:

- MicroProfile Language Server
- MicroProfile JDT LS Extensions

You can build all projects at once by running the `buildAll.sh` script (`buildAll.bat` on Windows).

Contribute at

<https://github.com/eclipse/lsp4mp/>

Coding Assistance in Editor/IDE

Can I get help with working with Jakarta EE APIs inside my favorite editor or IDE?

- Language Server for Jakarta EE (LSP4Jakarta) incubator project under the Eclipse Foundation



Contribute at

<https://github.com/eclipse/lsp4jakarta>

Code Generators

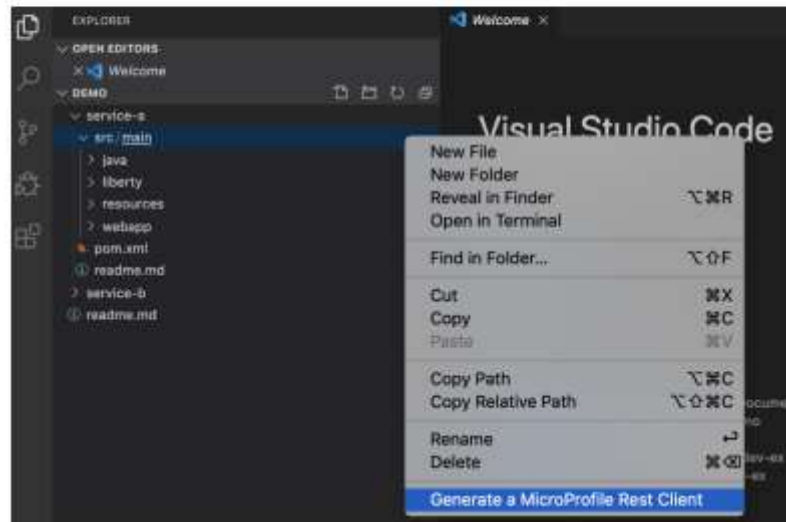
Any additional help with generating (boilerplate) code?

- MicroProfile Rest Client from OpenAPI docs
 - CLI via [OpenAPI Tools](#)
 - VS Code extension
- JAX-RS stubs generation in OpenAPI Tools

VS Code MicroProfile Rest Client Generator Extension

Visual Studio Marketplace v0.1.1 build: passing

The MicroProfile REST Client Generator Extension provides support for generating a [MicroProfile REST Client](#) interface template from an [OpenAPI](#) document. This extension calls the `openapi-generator` to generate `codeLs` and `apis` folders. The code for this extension is hosted under the [MicroShed](#) organization on [GitHub](#). Learn more about MicroProfile REST Client on [GitHub](#).



TEST



Automated Testing



- *The Usual Suspects*
 - JUnit
 - Arquillian ...
- Leveraging containers
 - Testcontainers
 - MicroShed Testing

```
@MicroShedTest
public class JaxrsJsonTest {

    @Container
    public static ApplicationContainer app = new ApplicationContainer()
        .withAppContextRoot("/myservice")
        .withReadinessPath("/myservice/app/people");

    @RESTClient
    public static PersonService personSvc;

    @Test
    public void testCreatePerson() {
        Long createId = personSvc.createPerson("Hank", 42);
        assertNotNull(createId);
    }
}
```

<https://microshed.github.io>

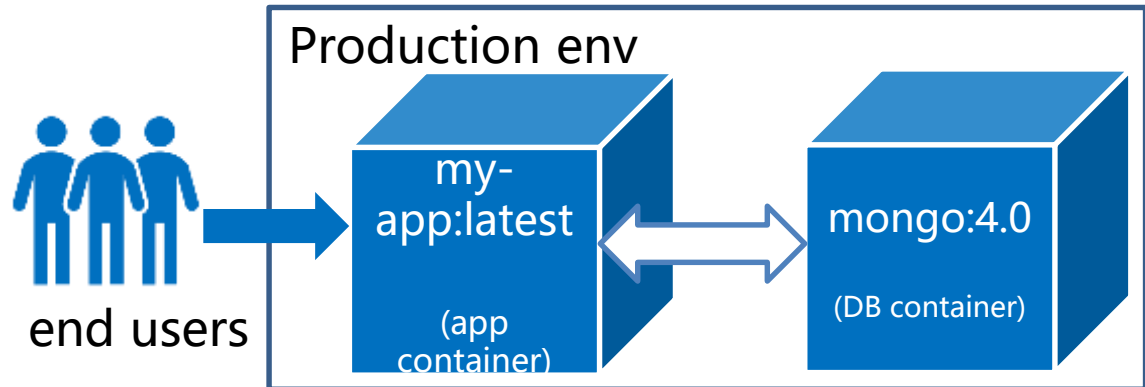
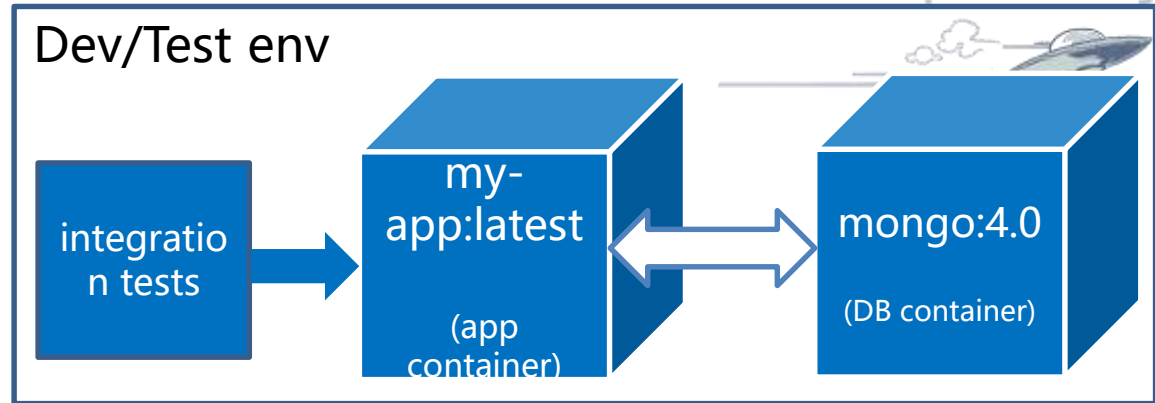
Testcontainers

Integration tests that are easy to setup, write, and run

Test your apps the same way they run in production

Tests are portable to any compatible implementation:

- Liberty
- Wildfly
- Payara
- TomEE
- etc...



DEPLOY



Buildpacks.io

Open Liberty



Provide framework and runtime support for applications

Transforms your application source code into container images that can run on any cloud

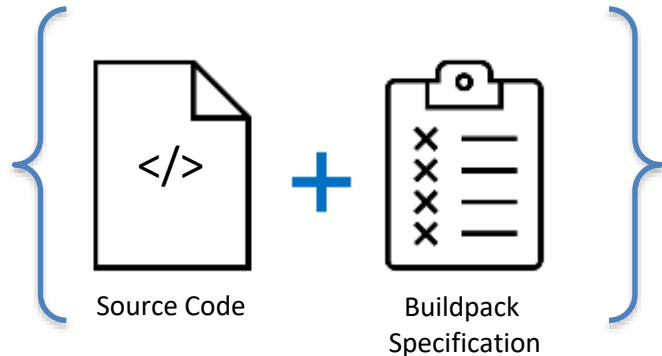
Avoid the need to manage Dockerfiles



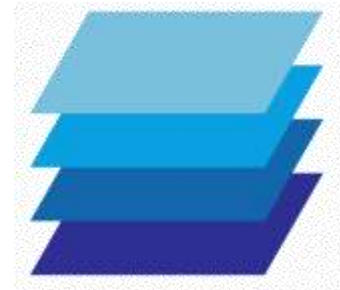
Buildpacks.io



CLOUD NATIVE
COMPUTING FOUNDATION



pack CLI



OCI Image

Feature comparison



Comparison

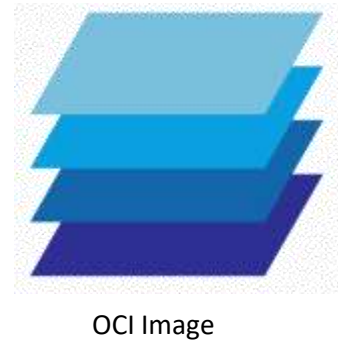
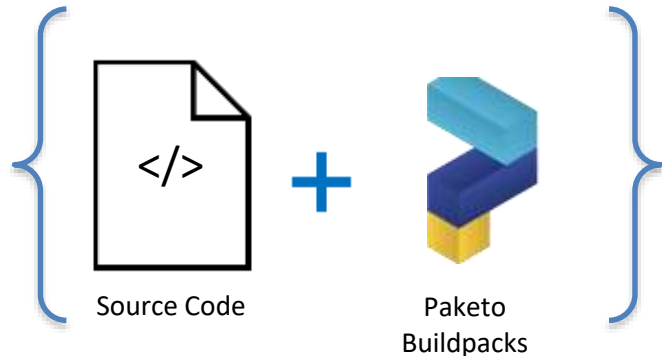
	 Cloud Native Buildpacks	 Dockerfile	 source-to-image (s2i)	 Jib	 ko
Advanced Caching	Yes	No	Yes	No	No
Auto-detection	Yes	No	Yes	Yes	Yes
Bill-of-Materials	Yes	No	No	No	No
Modular / Pluggable	Yes	No	No	N/A [†]	N/A [†]
Multi-language	Yes	Yes	Yes	No	No
Multi-process	Yes	No	No	No	No
Minimal app image	Yes	Yes [*]	Yes [‡]	Yes	Yes
Rebasing	Yes	No	No	No	No
Reproducibility	Yes	No	No	Yes	Yes
Reusability	Yes	No	Yes	N/A [†]	N/A [†]

	 Cloud Native Buildpacks	 Dockerfile	 source-to-image (s2i)	 Jib	 ko
Integrations	<ul style="list-style-type: none"> Azure CircleCI GitLab Google Heroku Spring Boot Tekton ... 	<ul style="list-style-type: none"> Amazon ECS Azure CircleCI GitLab Google Tekton ... 	<ul style="list-style-type: none"> OpenShift 	<ul style="list-style-type: none"> Gradle Maven 	
Governance	CNCF	Docker	Red Hat	Google	Google
Best for Building—	<ul style="list-style-type: none"> Applications 	<ul style="list-style-type: none"> Applications Base Images 	<ul style="list-style-type: none"> Applications 	<ul style="list-style-type: none"> Applications 	<ul style="list-style-type: none"> Applications

Paketo Buildpacks

Implementation of the Cloud Native Buildpack interface specification

Multiple languages



Liberty Buildpack

paketo-buildpacks/
liberty

A Cloud Native Buildpack that provides the Open Liberty and WebSphere Liberty runtimes

Rx 7 Contributors 2 Issues 8 Stars 5 Forks



Let's get started!

The following examples use the [Open Liberty starter](#) application as the application source. Download the Open Liberty starter application:

```
git clone https://github.com/openliberty/guide-getting-started.git
cd guide-getting-started/finish
```

Set a default builder:

```
pack config default-builder gcr.io/paketo-buildpacks/builder:base
```

Build the application on Liberty with Jakarta EE Platform 9.1 features and IBM Semeru OpenJ9:

```
pack build --env BP_JAVA_APP_SERVER=liberty --env BP_LIBERTY_PROFILE=jakartaee9 \
--buildpack paketo-buildpacks/eclipse-openj9 --buildpack paketo-buildpacks/java myapp
```



Your application is now transformed into an OCI image!

<https://paketo.io/docs/reference/builders-reference/>

<https://openliberty.io/blog/2022/04/01/cloud-native-liberty-buildpack.html>

gcr.io/paketo-buildpacks/liberty

TELEPRESENCE



What is Telepresence?

Open Liberty



- Fast, local development for Kubernetes and OpenShift Microservices
- Benefits:
 - Accelerate Inner Dev Loop
 - Shift Testing Left
 - Use Your Existing Workflow



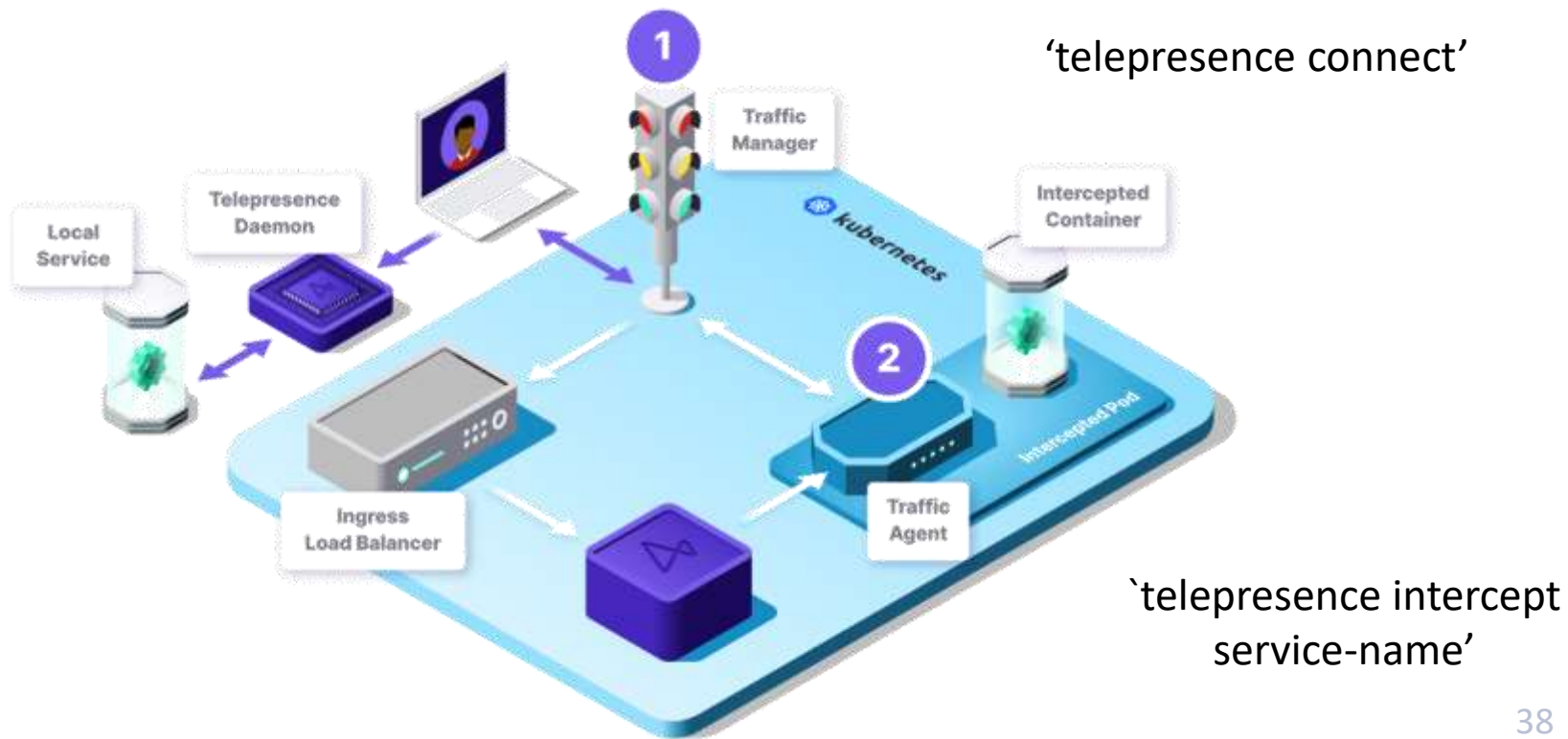
 **CLOUD NATIVE**
COMPUTING FOUNDATION

```
$> kubectl get deployments
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
hello-world  1/1    1           1          1m

$> telepresence --swap-deployment hello-world --expose 9080

# in a new terminal
$> mvn liberty:dev
```

How does it work?



The background is a dark, starry space scene. It features several hand-drawn style celestial bodies: a ringed planet like Saturn in the upper left, a comet with a long tail in the upper right, and various smaller planets and stars scattered throughout. The overall aesthetic is minimalist and illustrative.

TRY IT YOURSELF

Open Liberty Guides

<https://openliberty.io/guides/>

The screenshot shows the Open Liberty Guides website. At the top, there is a navigation bar with links for "Get Started", "Guides", "Docs", "Support", and "Blog". The main heading is "Guides" with the subtitle "The quickest way to learn all things Open Liberty, and beyond!". A search box labeled "Filter guides" is visible in the top right. The content is organized into sections: "DEVELOP (37 guides)", "BUILD AND TEST (10 guides)", and "DEPLOY (19 guides)". The "DEVELOP" section is expanded to show "Developing your cloud-native application", which includes "Getting started" and "RESTful service". Under "Getting started", there are two guides: "Getting started with Open Liberty" (25 minutes) and "Injecting dependencies into microservices" (15 minutes). Under "RESTful service", there are four guides: "Creating a RESTful web service", "Consuming RESTful services with template interfaces", "Consuming a RESTful web service", and "Documenting RESTful APIs".

DEVELOP (37 guides)

- Getting started
- RESTful service
- Reactive service
- Configuration
- Fault tolerance
- Observability
- Security
- Persistence
- Client side

DEVELOPING YOUR CLOUD-NATIVE APPLICATION

Getting started

- Getting started with Open Liberty**
Learn how to develop a Java application on Open Liberty with Maven and Docker.
25 minutes
- Injecting dependencies into microservices**
Learn how to use Contexts and Dependency Injection (CDI) to manage and inject dependencies into microservices.
15 minutes

RESTful service

- Creating a RESTful web service**
Learn how to create a REST service with JAX-RS, JSON-B, and Open Liberty.
- Consuming RESTful services with template interfaces**
Learn how to use MicroProfile Rest Client to invoke RESTful services over HTTP in a type-
- Consuming a RESTful web service**
Explore how to access a single RESTful web service and consume its resources in Java using JSON-B and JSON-P.
- Documenting RESTful APIs**
Explore how to document and filter RESTful APIs from code or static files by using MicroProfile OpenAPI.

BUILD AND TEST (10 guides)

- Build
- Test
- Containerize

DEPLOY (19 guides)

- Kubernetes
- Cloud deployment

Summary

<i>Developer Tools</i>	MicroProfile - Common Tools	Jakarta EE - Common Tools	Runtime Tooling (Open Liberty)
<i>Starters</i>	start.microprofile.io MP Starter IDE Plugins	start.jakarta.ee	Open Liberty Starter
<i>Build & Run</i>	N/A	N/A	Liberty Maven & Gradle Build Plugins Dev mode
<i>IDE/Editor Integration</i>	Eclipse LSP4MP LSP4MP IDE Plugin(s) MP Extension Pack for VS Code Code	Eclipse LSP4Jakarta	<i>Liberty Tools for Eclipse, IntelliJ IntelliJ and VS Code (tech previews)</i> <i>Dev mode for any editor</i>
<i>Code Generators</i>	OpenAPI Tools MP Rest Client Generator	N/A	N/A
<i>Automated Testing</i>	JUnit, Arquillian Testcontainers, MicroShed Testing	JUnit, Arquillian, Testcontainers,	Hot testing
<i>Deployment</i>	N/A	N/A	Paketo Buildpacks, Dev Mode

Resource Links



- Liberty Tools for Eclipse: <https://github.com/OpenLiberty/liberty-tools-eclipse>
- Open Liberty Starter: <https://start.openliberty.io/>
- Liberty Tools for VS Code:
<https://marketplace.visualstudio.com/items?itemName=Open-Liberty.liberty-dev-vscode-ext>
- Liberty Tools for IntelliJ: <https://plugins.jetbrains.com/plugin/14856-liberty-tools>
- VS Code Extension Pack for MicroProfile:
<https://marketplace.visualstudio.com/items?itemName=MicroProfile-Community.vscode-microprofile-pack>

Resource Links



- TestContainers: <https://www.testcontainers.org/>
- Paketo Buildpack for Liberty: gcr.io/paketo-buildpacks/liberty
- Telepresence Examples:
 - <https://github.com/sdaschner/liberty-dev-experience>
 - <https://heidloff.net/article/debugging-microservices-Kubernetes/>
- Open Liberty Guides:
 - Jakarta EE: <https://openliberty.io/guides/?search=jakarta%20ee&key=tag>
 - MicroProfile: <https://openliberty.io/guides/?search=microprofile&key=tag>
- Open Liberty Deep Dive: <https://openliberty.io/guides/liberty-deep-dive.html>



Thank You

@gracejansen27