

IBM Training

Red Hat OpenShift Application Development

Luca Floris, Grzegor Smolko
IBM Cloud Technical Consultant

Agenda

Day 4

DevOps: Continuous Delivery

- DevOps and DevSecOps
- WebSphere Liberty on OpenShift
- Lab 13: Set up a CI/CD pipeline on OpenShift using Jenkins to deploy a simple web application
- Transformation Advisor - Migrating WebSphere Applications to OpenShift

Microservices Architecture

- Microservices application architecture
- Developing microservices
- Twelve factor applications
- Refactoring monolith applications into microservices
- Lab 14 Build and deploy a polyglot microservices application on OpenShift

WebSphere Liberty Intro

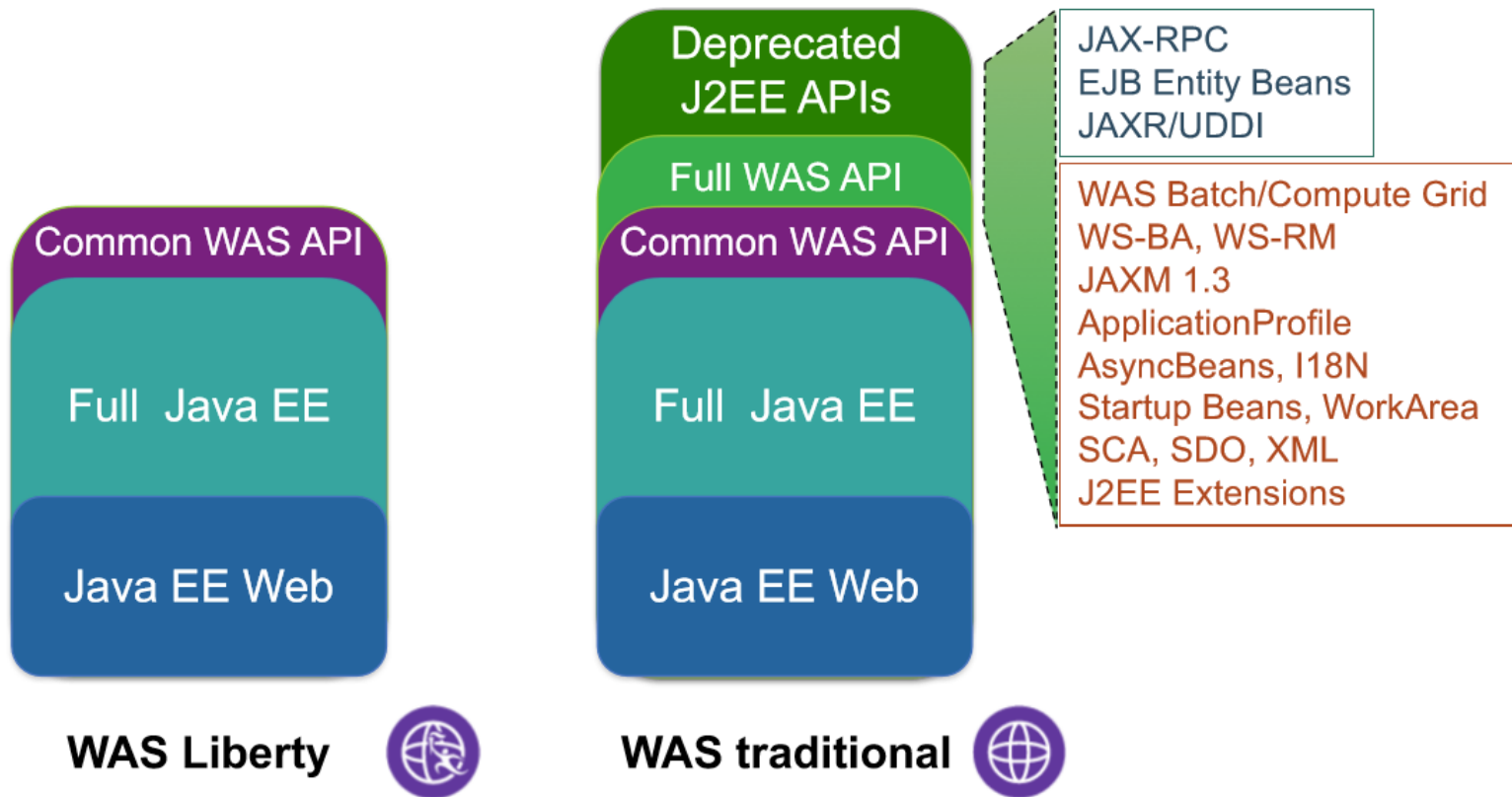
Why Choose Liberty

Open Liberty

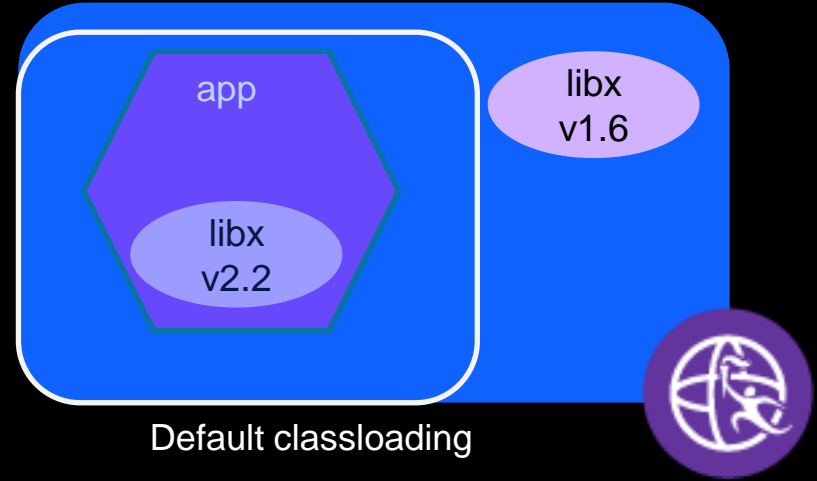
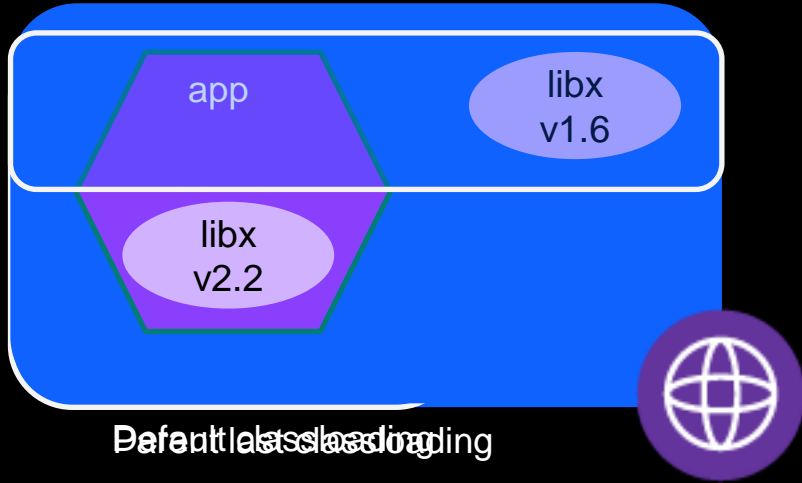


- Cloud Ready
 - Container optimized
 - Designed for dev/ops
 - Small disk footprint
 - Efficient memory usage
 - Fast startup
 - High throughput
 - Self-Tuned Thread Pool
- Developer friendly
 - Just enough application server
 - Fast inner loop with dev mode
 - Support for industry standard dev tools
 - Jakarta EE, Java EE, MicroProfile APIs
 - Zero Migration
 - Easy install and setup

Liberty and WAS API differences



Class Visibility

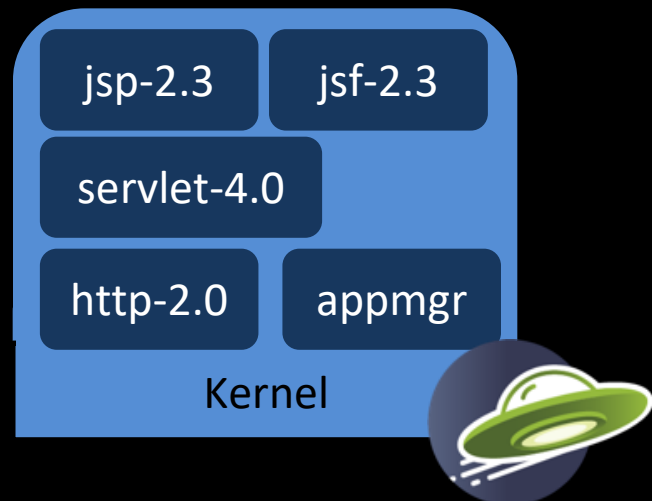


Just Enough Application Server

You control which features are loaded into each server instance



```
<feature>jsf-2.3</feature>
```



Periodic Table of Liberty (20.0.0.6)

zOS

ND

Base

Core












Open
Liberty

New in
4Q19

New in
3Q19

New in
2Q20

New in
1Q20

batchSMFLogging-1.0		zosLocalAdapters-1.0		zosTransaction-1.0		zosSecurity-1.0	
collectiveController-1.0		zosRequestLogging-1.0		zosWlm-1.0			
dynamicRouting-1.0		healthManager-1.0		scalingController-1.0			
clusterMember-1.0		healthAnalyzer-1.0		scalingMember-1.0		Security	
cloudant-1.0		sipServlet-1.0		batchManagement-1.0		Operations	
javaee-7.0				wsAtomicTransaction-1.2			
javaee-8.0							
jakartaee-8.0						passwordUtilities-1.0	
						wsSecurity-1.1	
						wsSecuritySaml-1.0	
bells-1.0		mpContextPropagation-1.0		adminCenter-1.0		constrainedDelegation-1.0	
concurrent-1.0		mpGraphQL-1.0		collectiveMember-1.0		federatedRepository-1.0	
javaMail-1.6		mpReactiveMessaging-1.0		distributedMap-1.0		jwt-1.0	
jaxb-2.2		mpReactiveStreams-1.0		eventLogging-1.0		jwtSso-1.0	
jdbc-4.3		opentracing-1.3		logstashCollector-1.0		sessionDatabase-1.0	
jpaContainer-2.2		osgiConsole-1.0		monitor-1.0		webCache-1.0	
jsfContainer-2.3		springBoot-2.0		openapi-3.1			
json-1.0		webProfile-7.0		requestTiming-1.0			
jsonbContainer-1.0		webProfile-8.0		usageMetering-1.0			
jsonpContainer-1.1				restConnector-2.0			
microProfile-3.3				sessionCache-1.0			
		APIs				audit-1.0	
						ldapRegistry-3.0	
						oauth-2.0	
						openid-2.0	
						openidConnectClient-1.0	
						openidConnectServer-1.0	
						samlWeb-2.0	
						scim-1.0	
						socialLogin-1.0	
						spnego-1.0	
						transportSecurity-1.0	

Periodic Table of Liberty

ZOS

ND

Base

Core

Open
Liberty

New in
3Q19

New in
2Q19

New in
1Q19

New in
4Q18

batchSMFLogging-1.0

collectiveController-1.0

clusterMember-1.0

dynamicRouting-1.0

cloudant-1.0

javaee-7.0

javaee-8.0

sipServlet-1.0

bells-1.0

concurrent-1.0

javaMail-1.6

jaxb-2.2

jdbc-4.3

jpaContainer-2.2

jsfContainer-2.3

json-1.0

jsonbContainer-1.0

jsonpContainer-1.1

microProfile-3.0



mpContextPropagation-1.0

mpReactiveMessaging-1.0

mpReactiveStreams-1.0

opentracing-1.3

osgiConsole-1.0

springBoot-2.0

webProfile-7.0

webProfile-8.0

APIs

appClientSupport-1.0

appSecurityClient-1.0

batch-1.0

concurrent-1.0

ejb-3.2

ejbHome-3.2

ejbPersistentTimer-3.2

ejbRemote-3.2

j2eeManagement-1.1

javaMail-1.6

jacc-1.5

jaxb-2.2

jaxws-2.2

jca-1.7

jms-2.0

managedBeans-1.0

mdb-3.2

wasJmsClient-2.0

webProfile-8.0

wmqJmsClient-2.0

appSecurity-3.0

beanValidation-2.0

cdi-2.0

ejbLite-3.2

el-3.0

jaspic-1.1

jaxrs-2.1

jaxrsClient-2.1

jdbc-4.2

jndi-1.0

jpa-2.2

jsonp-1.0

jsonp-1.1

jsf-2.3

jsp-2.3

managedBeans-1.0

servlet-4.0

ssl-1.0

websocket-1.1

cdi-2.0

jaxrs-2.1

jsonp-1.1

jsonp-1.0

mpConfig-1.3

mpFaultTolerance-2.0

mpMetrics-2.0

mpJwt-1.1

mpOpenAPI-1.1

mpOpenTracing-1.3

mpRestClient-1.3

mpHealth-2.0

Liberty Server Configuration

simple but powerful and flexible

```
<server description="new server">
```

```
<!-- Enable features -->
```

```
<featureManager>
```

```
  <feature>servlet-4.0</feature>
```

```
  <feature>jdbc-4.2</feature>
```

```
</featureManager>
```

```
<httpEndpoint id="defaultHttpEndpoint"
```

```
  host="${env.COMPUTERNAME}"
```

```
  httpPort="${httpAdminPort}"
```

```
  httpsPort="${httpsAdminPort}" />
```

```
<webApplication id="blogapp"
```

```
  location="blogapp.war"/>
```

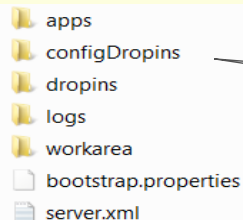
```
<include location="${shared.config.dir}/datasource.xml"/>
```

```
</server>
```

You define **which features** you want to use and in **which version**. Features can be **activated on the fly**.

Use **environment variables** or **properties** to **share and re-use** server configurations.

Use **includes** to split the configuration into useful parts for **split of responsibility** and **re-use**.



- apps
- configDropins
- dropins
- logs
- workarea
- bootstrap.properties
- server.xml

Use the **configDropins** directory to **override** configurations

Security

- Security by default
 - No remotely accessible ports

```
<httpEndpoint id="defaultHttpEndpoint" host="*" />
```

- Enable admin, enable security

```
<feature>restConnector-1.0</feature>  
<quickStartSecurity userName="admin"  
    userPassword="{aes}adSDwijgnb==" />
```

- Enable ssl using ssl-1.0

```
<feature>ssl-1.0</feature>  
<keyStore password="{aes}adSDwijgnb==" />
```

App Security

- Feature to enable
- Configure security role bindings in server.xml

```
<feature>appSecurity-2.0</feature>
<webApplication location="myweb.war">
  <application-bnd>
    <security-role name="user">
      <group name="myGroup"/>
    </security-role>
  </application-bnd>
</webApplication>
```

- Configure registry

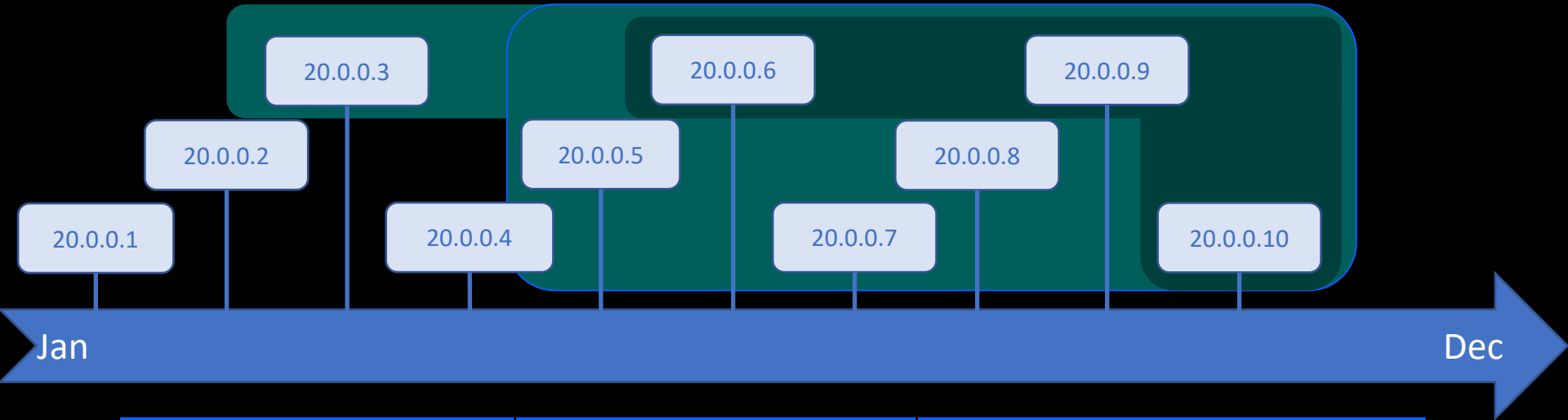
```
<feature>ldapRegistry-3.0</feature>
```

Liberty Zero Migration

- Zero config migration
 - Write once, run forever
- Zero app migration
 - No behavior changes in existing features
 - New behaviors in new features
- Choose your Java
 - Java 14, 11, 8
 - AdoptOpenJDK
 - IBM
 - OpenJDK
 - Oracle



Liberty Release Cadence Example



	All CD releases	CD releases ending .3 .6 .9 .12
Support Provided	5 years	5 years
iFixes	24 weeks	2 years
Proactive Security iFixes	Most recent	Most recent 2

Build

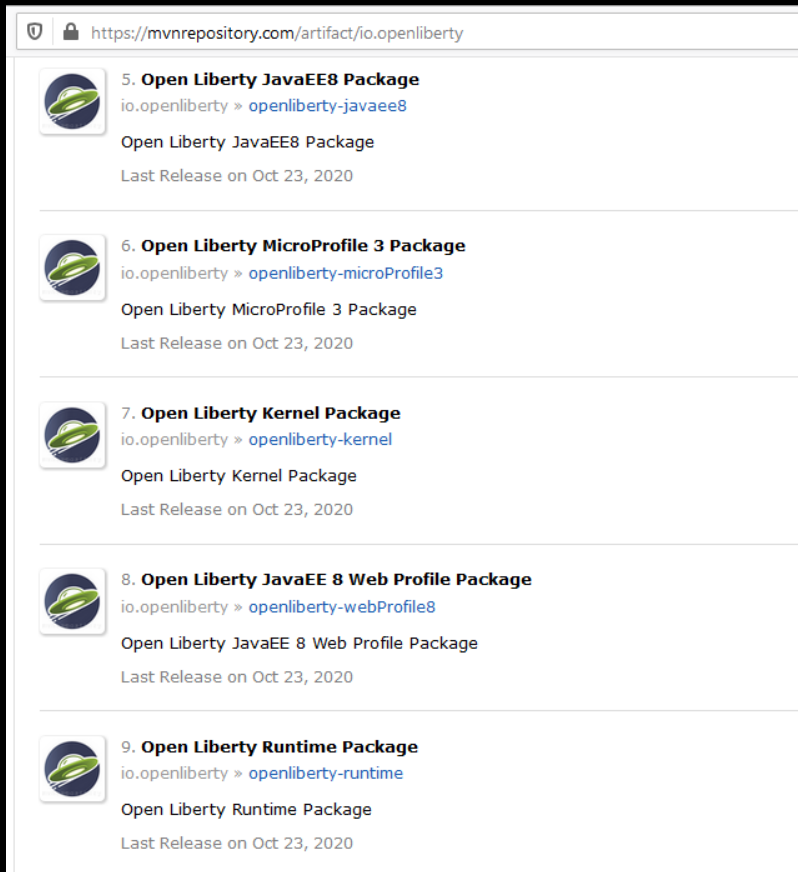
Integrate into build environment

APIs provided in Maven Central

Plugins for Maven and Gradle

Dev mode allows you to develop with any text editor providing hot reload and deployment, on demand testing, and debugger support

```
<plugin>
  <groupId>io.openliberty.tools</groupId>
  <artifactId>liberty-maven-plugin</artifactId>
  <version>3.2</version>
  <configuration>
    <serverName>guideServer</serverName>
  </configuration>
</plugin>
```



The screenshot shows a web browser window with the URL <https://mvnrepository.com/artifact/io.openliberty>. The page displays a list of five Open Liberty packages, each with a green and blue logo icon. The packages are:

- 5. **Open Liberty JavaEE8 Package**
io.openliberty » [openliberty-javaee8](#)
Open Liberty JavaEE8 Package
Last Release on Oct 23, 2020
- 6. **Open Liberty MicroProfile 3 Package**
io.openliberty » [openliberty-microProfile3](#)
Open Liberty MicroProfile 3 Package
Last Release on Oct 23, 2020
- 7. **Open Liberty Kernel Package**
io.openliberty » [openliberty-kernel](#)
Open Liberty Kernel Package
Last Release on Oct 23, 2020
- 8. **Open Liberty JavaEE 8 Web Profile Package**
io.openliberty » [openliberty-webProfile8](#)
Open Liberty JavaEE 8 Web Profile Package
Last Release on Oct 23, 2020
- 9. **Open Liberty Runtime Package**
io.openliberty » [openliberty-runtime](#)
Open Liberty Runtime Package
Last Release on Oct 23, 2020

Customized Docker containers

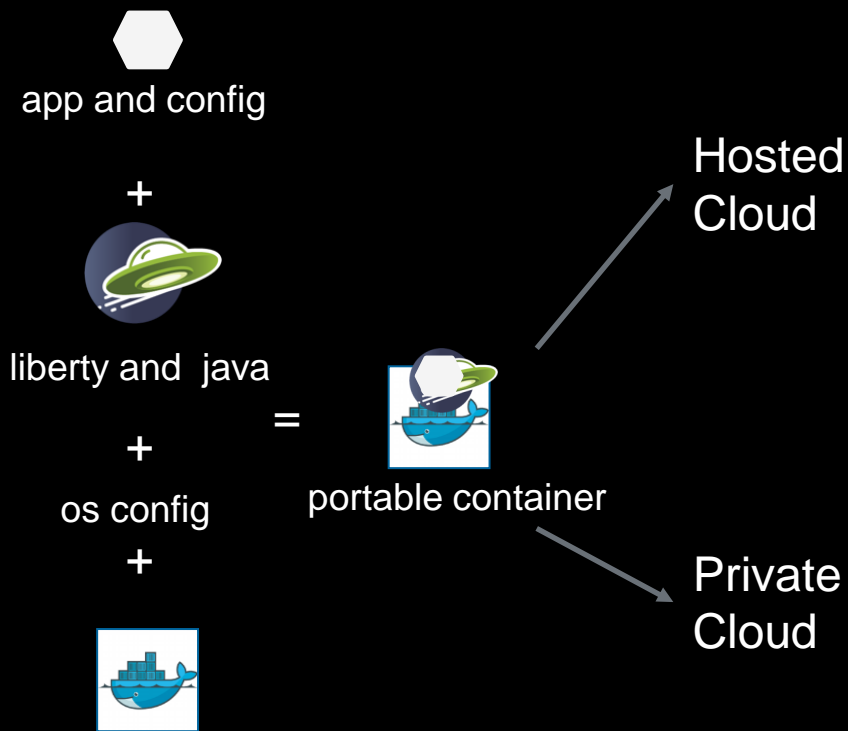
Liberty images on Docker Hub

- Liberty containers:
 - Kernel, webProfile, full Profile and latest Beta images,
 - on Ubuntu or UBI
 - with java8 or java11
 - Docker files:
<https://github.com/WASdev/ci.docker>
<https://github.com/OpenLiberty/ci.docker>

```
# FROM openliberty/open-liberty:kernel-java8-openj9-ubi
FROM websphere-liberty:kernel
COPY server.xml /config/
COPY myapp.war /config/apps/
RUN configure.sh
```



Liberty in Containers



IBM Cloud Kubernetes Service

Azure Kubernetes Service

Google Kubernetes Engine

Amazon Elastic Kubernetes Service

Jelastic

Red Hat Open Shift Container Platform

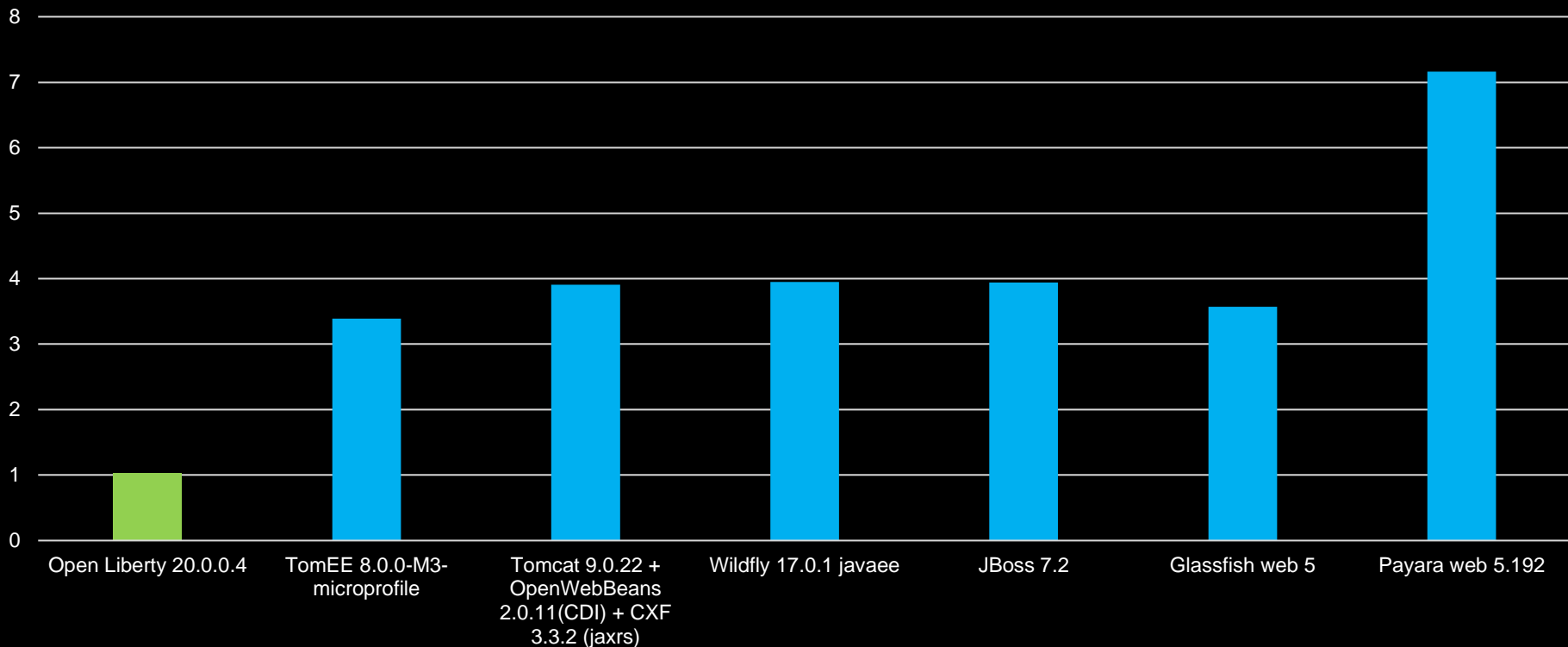
Pivotal Kubernetes Service

Pivotal Cloud Foundry

FROM open-liberty
COPY myapp.war /config/dropins/myapp.war

Open Liberty startup time comparison (using OpenJ9 JVM)

PingPerf application startup time with OpenJ9 Shared Classes Cache (in seconds)



Open Liberty starts faster than other app servers, OpenJ9 starts faster than HotSpot

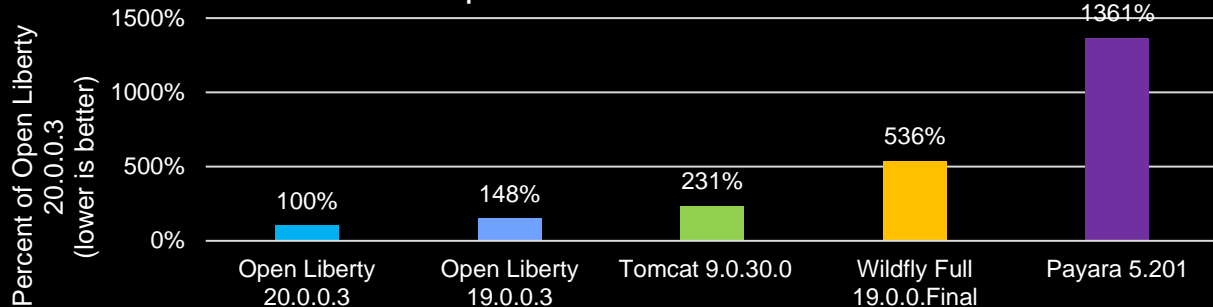
EE7 Performance (Trade7)

- Comparisons used each application server's Docker image
- Liberty outperforms others on all metrics for EE7 performance (startup time less than half, throughput and memory footprint are much better as well)

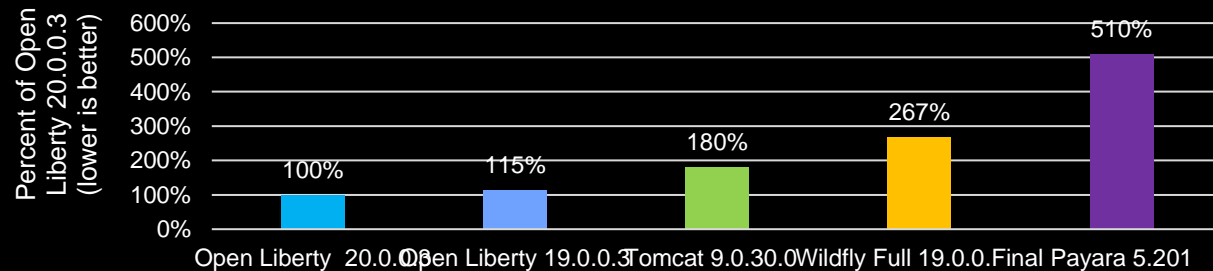
System Configuration:

SUT: LinTel – SLES 12.3, Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 4 physical cores, 64GB RAM. JDK version distributed with the docker images used for each server instance.

Startup Time - Trade7 - Docker



Memory Footprint - Trade7 - Docker

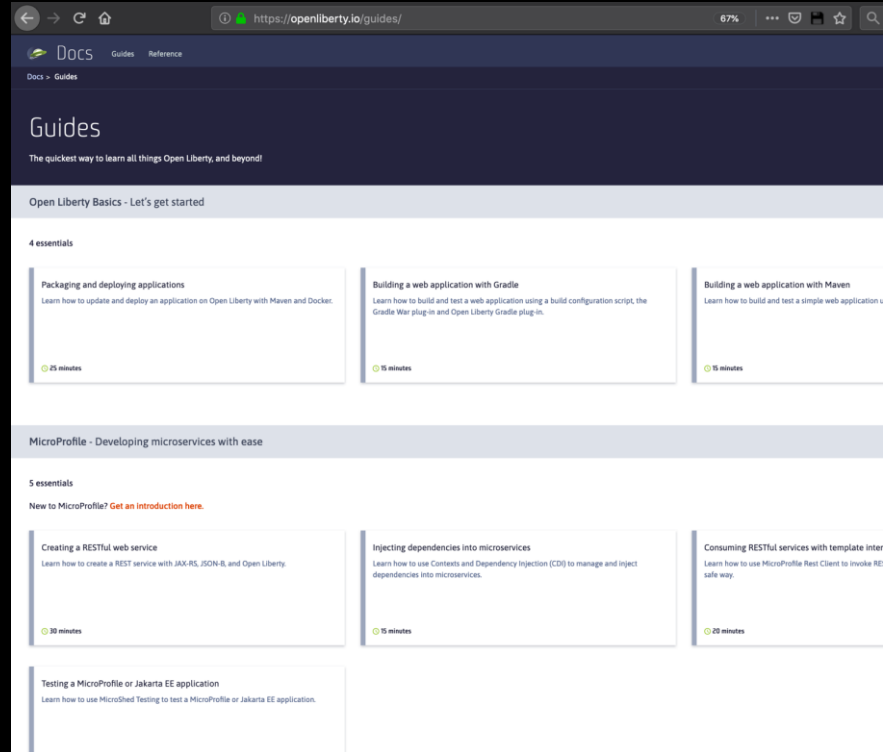


Throughput - Trade7 - Docker



Open Liberty Guides

- Hands-on learning in ~20 minutes
- 44 guides
 - MicroProfile & Jakarta EE
 - Open Shift, Docker, Kubernetes Istio
- Latest Guides
 - *Deploying microservices to an OKD cluster using Minishift*
 - *Deploying microservices to Google Cloud Platform*



<https://openliberty.io/guides>

WebSphere on OpenShift

Liberty Operator

Deploy and manage applications running on Open Liberty into [OKD](#) or [OpenShift](#) clusters

Perform Day-2 operations such as gathering traces and dumps

<https://github.com/OpenLiberty/open-liberty-operator/blob/master/doc/user-guide.adoc>

OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. Install Operators on your clusters to provide optional add-ons and shared services to your cluster, providing a self-service experience.

All Items

All Items

AI/Machine Learning

Application Runtime

Big Data

Cloud Provider

Database

liberty



Operator Backed

Open Liberty Application

provided by IBM

Configuration for the deployment of an Open Liberty application



Operator Backed

Open Liberty Dump

provided by IBM

Day-2 operation for generating server dumps



Operator Backed

Open Liberty Trace

provided by IBM

Day-2 operation for gathering server traces



Open Liberty Operator

provided by IBM

Deploy and manage applications...



Installed

Operator key features

- Routing - expose your application to external users via a single toggle.
- High Availability - run multiple instances of your application for high availability (static/autoscaling)
- Persistence - enable persistence for your application by specifying storage requirements.
- Serviceability - easily use a single storage for serviceability related operations, such as gathering server traces or dumps.
- Service Binding - easily bind to available services in your cluster.
- Knative - deploy your serverless application on Knative using a single toggle.

Liberty Operator - Application

Operator container with the controller is deployed into a Pod and listens for incoming resources with

Kind: OpenLibertyApplication.

Creating an OpenLibertyApplication custom resource (CR) triggers the Open Liberty Operator to create, update or delete Kubernetes resources needed by the application to run on your cluster.

```
apiVersion: openliberty.io/v1beta1
kind: OpenLibertyApplication
metadata:
  name: my-liberty-app
spec:
  applicationImage: quay.io/my-repo/my-app:1.0
  service:
    type: ClusterIP
    port: 9080
  expose: true
  storage:
    size: 2Gi
    mountPath: "/logs"
```


Liberty Operator - Request server dump

Application needs to have storage for serviceability already configured

CR must be created in the same namespace as the Pod to operate on.

Dump file name will be added to OpenLibertyDump CR status and file will be stored in serviceability folder using format such as
/serviceability/NAMESPACE/POD_NAME/TIMESTAMP.zip

Once the dump has started, the CR can not be re-used to take more dumps. A new CR needs to be created for each server dump.

```
apiVersion: openliberty.io/v1beta1
kind: OpenLibertyDump
metadata:
  name: example-dump
spec:
  podName: Specify_Pod_Name_Here
  include:
    - thread
    - heap
```

To see the status of all dump operations in the current namespace run:

```
oc get oldump -o wide
```

Liberty Operator - Request server traces

Application needs to have storage for serviceability already configured

CR must be created in the same namespace as the Pod to operate on.

Generated trace files, along with *messages.log* files, will be in the folder using format */serviceability/NAMESPACE/POD_NAME/*

To stop the trace:

- Set the disable parameter to true
- Or delete the CR

```
apiVersion: openliberty.io/v1beta1
kind: OpenLibertyTrace
metadata:
  name: example-trace
spec:
  podName: Specify_Pod_Name_Here
  traceSpecification: "*=info:com.ibm.ws.webcontainer*=all"
  maxFileSize: 20
  maxFiles: 5
```

To see the status of all trace operations in the current namespace run:

```
oc get oltrace -o wide
```

Resources

- Open Liberty: <https://www.openliberty.io/>
- Eclipse MicroProfile: <https://microprofile.io/>
- Jakarta EE: <https://jakarta.ee/>
- Liberty advantage: <https://www.ibm.com/downloads/cas/NVY3KY4E>
- Open Liberty Guides: <https://openliberty.io/guides>
- Why Liberty is the best Java runtime for the Cloud <https://developer.ibm.com/wasdev/docs/liberty-profile-best-java-runtime-cloud/>
- WebSphere Application Server V8.5 Administration and Configuration Guide for Liberty Profile (Redbook)
<http://www.redbooks.ibm.com/abstracts/sg248170.html?Open>
- Liberty videos: https://www.ibm.com/support/knowledgecenter/SSAW57_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/covr_media.html
- Java support dates <http://www.ibm.com/developerworks/java/jdk/lifecycle>
- Single Stream Continuous Delivery <https://www-01.ibm.com/support/docview.wss?uid=ibm10869798>
- WebSphere Migration Knowledge Collection: Planning and Resources <https://www-01.ibm.com/support/docview.wss?uid=swg27008724>

Questions/Discussions?