

Liberty Quarterly Update

21.0.0.7-21.0.0.9

Alasdair Nottingham – Liberty Lead Architect

 @nottocode

Agenda

Open Liberty



Part 1: 20 Minute Liberty overview

Part 2: Security Best Practices

Part 3: What is new this quarter

Part 3: Q&A

20 minute overview



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>jsf-2.3</feature>
```

jsp-2.3

jsf-2.3

servlet-4.0

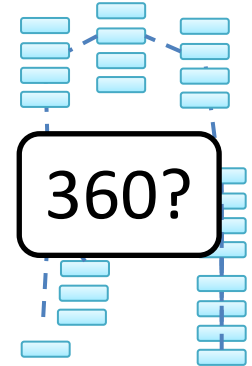
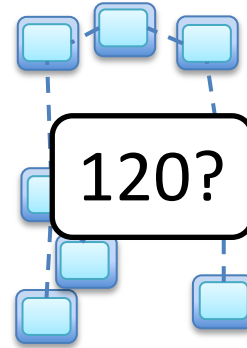
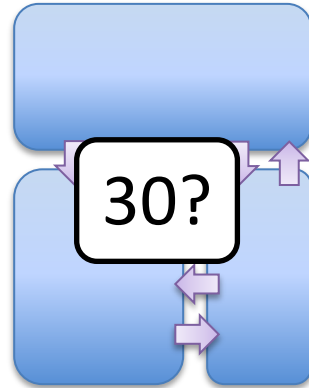
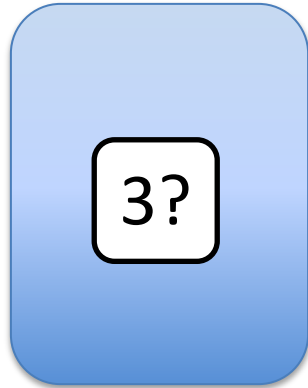
http-2.0

appmgr

Kernel



Granularity cost implications



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

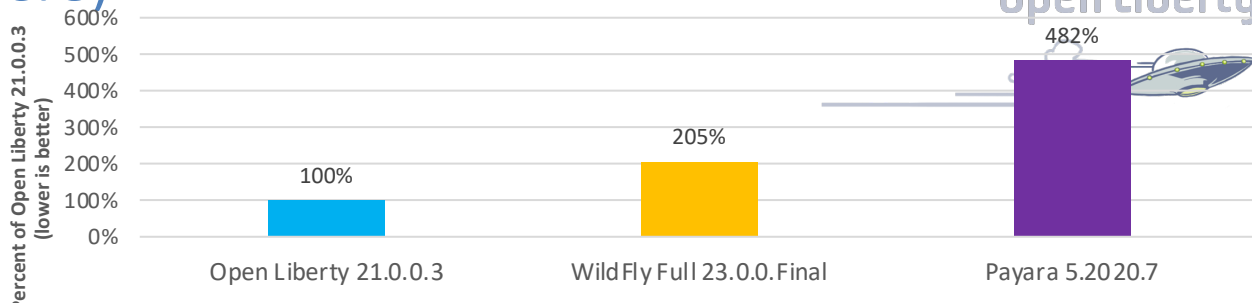
Performance (Daytrader8)

- 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)

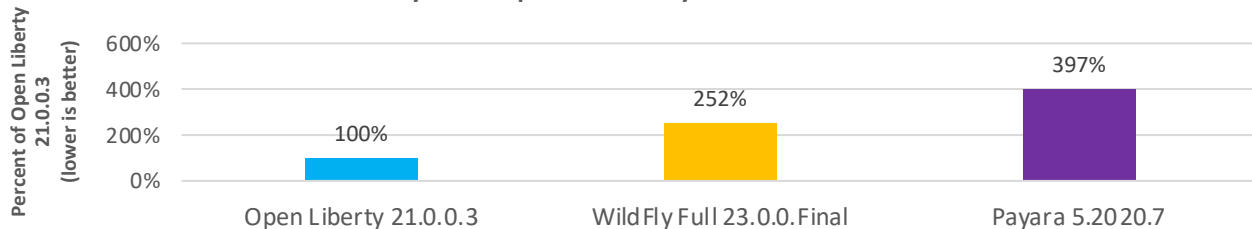
System Configuration:

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.

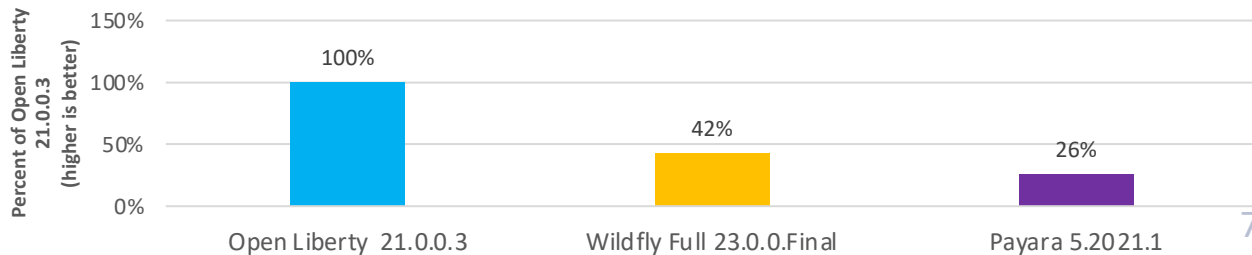
Startup - Daytrader8 - Docker



Memory Footprint - Daytrader8 - Docker



Throughput- Daytrader8 - Docker



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

Traditional Deployment

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

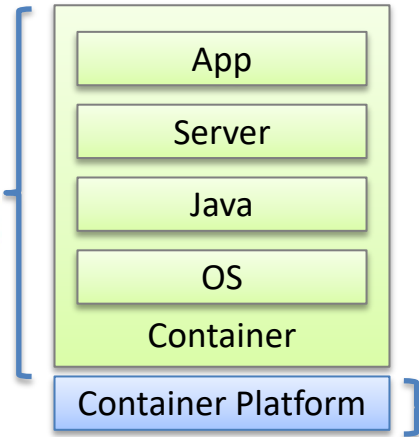
- I deploy the app
- I handle automation
- I monitor the app
- I maintain the infrastructure
- I apply important security fixes
- I plan and execute migrations




Cloud platforms shift responsibilities

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**



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

Liberty Release Cadence

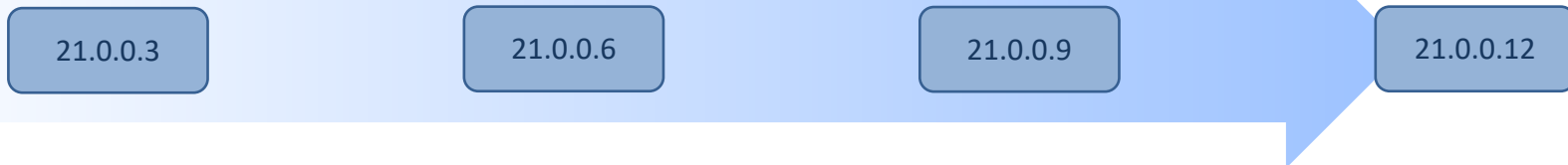
Open Liberty



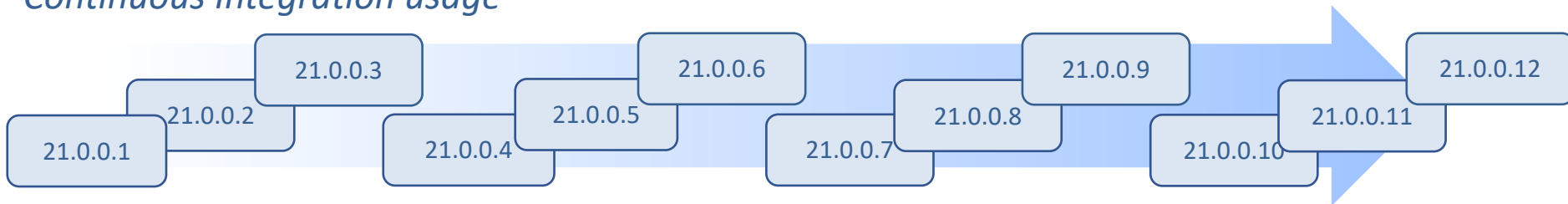
Liberty's 'zero migration' architecture makes picking up a new release simple

Skipping a release does not introduce migration work

Traditional 'fix pack' usage

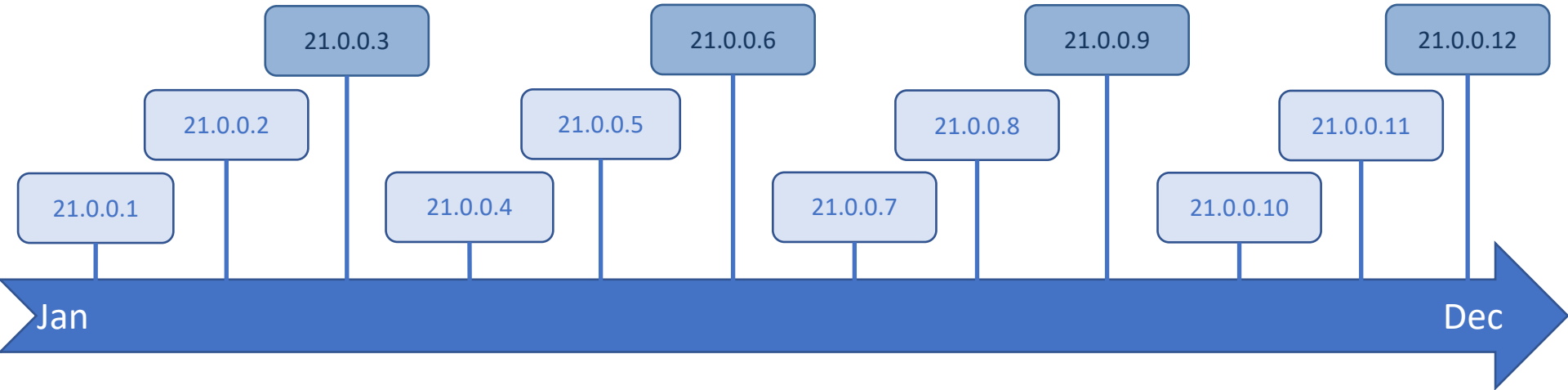


Continuous Integration usage



Liberty Release Cadence

Open Liberty



	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

2021

M	T	W	T	F	S	S	M	T	W	T	F	S	S
				GA 1							GA 2		

Release to
maven central
DockerHub
IBM Container Registry

Release to
Fix Central
z/OS
Installation Manager

2022

M	T	W	T	F	S	S	M	T	W	T	F	S	S
								GA					

Poll



What would be your preferred release approach?

- a) Keep it to Fridays with separate dates
- b) Aligning to Tuesdays/Wednesdays is my preference
- c) I don't mind either way

Zero Migration

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



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.

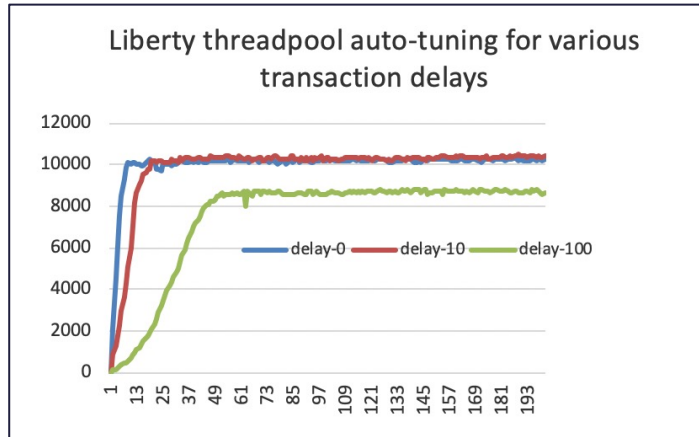
Henrik Lundström, WAS Systems Administrator,
Handelsbanken (Sweden)



43,800x
improvement

Kubernetes optimized

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**



Open Liberty Operator
provided by IBM

Deploy and manage
applications running on Open
Liberty



Kubernetes Optimized

A man with a beard and short brown hair, wearing a grey and red plaid sweater, is smiling and looking at a white smartphone. He is sitting at a desk with a laptop. The background is a bright, modern office with a woman in a dark jacket and purple accents visible in the distance.

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

WebSphere Technology Owner
Large health provider



Developer experience

IDEs

Dev Mode

Repositories

The Central Repository
docker hub

Build


Maven™
Gradle

APIs

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA
Java EE™
JAKARTA EE
Spring Boot®

Testing

microshed testing
JUnit
Arquillian

 **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

The image shows a development environment with three main components:

- IDE Explorer:** Shows a project structure with folders like `.gradle`, `.vscode`, `build`, `src`, `main`, `liberty/config`, and `webapp`. The `server.xml` file is selected.
- server.xml:** Contains the following XML configuration:

```
1 <server description="Sample Liberty server">
2   <featureManager>
3     <feature>jaxrs-2.1</feature>
4     <feature>jsonp-1.1</feature>
5     <feature>cdi-2.0</feature>
6     <feature>mpMetrics-2.0</feature>
7     <feature>mpConfig-1.3</feature>
8   </featureManager>
9
10  <webApplication location="{artifactId}.
11
12  <mpMetrics authentication="false"/>
13
14  <!-- tag::logging[] -->
15  <logging traceSpecification="com.ibm.ws.
16  <!-- end::logging[] -->
17
18  <httpEndpoint host="*" httpPort="{defal
19  httpsPort="{default.https.port}" ic
```
- Terminal:** Shows the output of the application starting in dev mode:

```
[INFO] *****
[INFO] * Liberty is running in dev mode.
[INFO] * To run tests on demand, press Enter.
[INFO] * To rebuild the Docker image and restart the container, type 'r' and press Enter.
[INFO] * To stop the server and quit dev mode, press Ctrl-C or type 'q' and press Enter.
[INFO] *
[INFO] * Liberty container port information:
[INFO] * Internal container HTTP port [ 9080 ] is mapped to Docker host port [ 9080 ]
[INFO] * Internal container HTTPS port [ 9443 ] is mapped to Docker host port [ 9443 ]
[INFO] * Liberty debug port mapped to Docker host port: [ 7777 ]
[INFO] *
[INFO] * Docker network information:
[INFO] * Container name: [ liberty-dev ]
[INFO] * IP address [ 172.17.0.2 ] on Docker network [ bridge ]
[INFO] *****
[INFO] Source compilation was successful.
[INFO] Tests compilation was successful.
[INFO] [AUDIT ] CwMKT0017I: Web application removed (default_host): http://c1bf2d4d704a:9080/
[INFO] [AUDIT ] CwMKZ0009I: The application demo-devmode-maven has stopped successfully.
[INFO] [AUDIT ] CwMKT0016I: Web application available (default_host): http://c1bf2d4d704a:9080/
[INFO] [AUDIT ] CwMKZ0003I: The application demo-devmode-maven updated in 1.157 seconds.
```

Error 404: java.io.FileNotFoundException: SRVE0190E: File not found: /health

How to get Support



WebSphere



z/OS
ND
Base
Core



WebSphere Hybrid Edition

IBM Integrated Application Runtimes

Java:

- WebSphere
- Liberty
- MicroProfile
- Jakarta EE
- OpenJ9



Cloud Foundry Migration Runtime

Transformation Advisor

Mono2micro



Red Hat OpenShift



Security Best Practices

Gary Picher – WebSphere Security Architect

Agenda

- Intrusions and vulnerabilities
 - Types of intrusions – Operating System, Network, Application
- Operating System
 - File system protections, Installation Manager, Server Administration, z/OS specific info, Windows
- Network
 - Firewalls, web servers, and topology
 - Transport level security/SSL – encryption/ciphers
 - Use of Crypto, LDAP, LTPA, other items
- Application
 - Best practices for application construction and configuration hardening

Intrusions and vulnerabilities

What are the things we're protecting against?



August 25th, 2021

Types of intrusions/vulnerabilities

- Operating system – users with machine/local access attempting to cause damage and extract sensitive information.
- Network – subversion of network protocols – unauthorized looking at traffic, altering traffic (ie: re-play & meet in the middle/MITM attacks).
- Application – external users running applications that derive or inherit privileges that they are not authorized to (ie: inheriting the identity of the server).

Operating System

How to minimize risk of internal OS users causing harm?



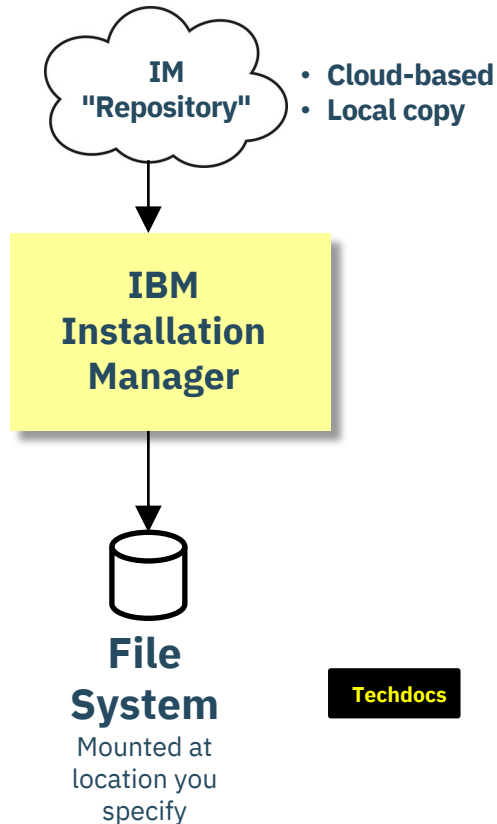
August 25th, 2021

First ... stay on current Liberty maintenance!

- With Liberty, staying current is considerably less obtrusive than with WAS in the past.
 - Fix packs ship every 4 weeks and follow the following convention:
 - 21.0.0.1 (first fix pack of 2021), 21.0.0.2, ..., 21.0.0.12 (last fix pack of 2021)
 - For applications and configuration, Liberty fix packs are upward compatible
 - Liberty is continuous delivery - with a zero-migration policy as you move to recent fix packs.
 - For example : use of the servlet-3.0 feature can continue even after servlet-3.1 and servlet-4.0 are available. There is no forced upgrade to newer features as Liberty fix packs are made available.
 - Critical to remain current to be best positioned for critical security fixes (CVEs and PSIRTs). Pay attention to security bulletin updates (CVEs):

<http://www-01.ibm.com/support/docview.wss?uid=swg21984533>

Installation Manager (IM)



Use "Group Mode"

- "Admin Mode" requires ID that runs IM be superuser (uid=0) ❌
- "User Mode" implies only that ID can run IM ✅
- "Group" mode allows any ID connected to the IM group to run IM ✅✅
- Use something *other than* default IMGROUP and IMADMIN

Use a "Service Zone" Concept

- It's a general good practice (provides greater flexibility)
- It allows IM install target to be R/W while users access R/O copy

"Copy out" and mount Read-Only

- Consider `chmod -R 750` on copy to set "other" to "none"
- Mount copy as R/O

Techdocs

<http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP102554> **IM Guide**
<http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/TD106391> **Sample Installation Jobs**
<http://www.ibm.com/support/techdocs/atmastr.nsf/WebIndex/TD106392> **WAS V9.0 Sample Install Jobs**

Windows installs

- Installation manager group mode does NOT APPLY for Windows installs.
 - IM admin mode on Windows and Liberty will install into C:\Program Files path as a default. All users have read/execute to this dir. **Best practices for Windows :**
- Set up admin user(s) for the install and a separate non-admin IDs for servers to run under. For example:

User	Group	Use this ID to ...
admin01	WASUser WASAdmin	<ul style="list-style-type: none">• Install liberty using IM user mode or archive install.• Add/remove features via IM or installutility.• Performs installation customizations such as update etc\server.env• Create server and update server configuration
serv01, serv02	WASUser	Start and stop server

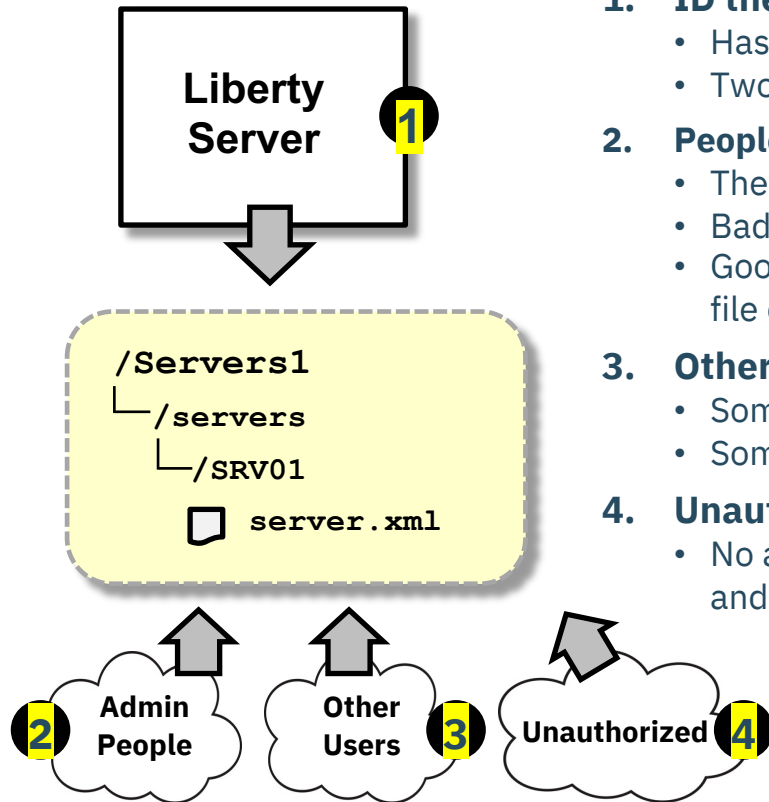
Validate the install

- Make sure to issue `./wlp_install_dir/bin/productInfo validate` after the install is to validate it (uses MD5 checksum file). Should see messages like these:

```
./productInfo validate
Start product validation...
Validating feature: appSecurity-2.0... PASS!
Validating feature: beanValidation-1.1... PASS!
Validating feature: bluemixUtility-1.0... PASS!
Validating feature: cdi-1.2... PASS!
Validating feature: collectiveMember-1.0... PASS!
Validating feature: distributedMap-1.0... PASS!
Validating feature: ejbLite-3.2... PASS!
Validating feature: el-3.0... PASS!
Validating feature: federatedRegistry-1.0... PASS!
Validating feature: jaxrs-2.0... PASS!
Validating feature: jaxrsClient-2.0... PASS!
Validating feature: jdbc-4.1... PASS!
Validating feature: jndi-1.0... PASS!
Validating feature: jpa-2.1... PASS!
Validating feature: jpaContainer-2.1... PASS!
Validating feature: jsf-2.2... PASS!
Validating feature: json-1.0... PASS!
```

```
Validating feature: jsonp-1.0... PASS!
Validating feature: jsp-2.3... PASS!
Validating feature: ldapRegistry-3.0... PASS!
Validating feature: localConnector-1.0... PASS!
Validating feature: managedBeans-1.0... PASS!
Validating feature: microProfile-1.0... PASS!
Validating feature: monitor-1.0... PASS!
Validating feature: requestTiming-1.0... PASS!
Validating feature: restConnector-1.0... PASS!
Validating feature: restConnector-2.0... PASS!
Validating feature: servlet-3.1... PASS!
Validating feature: sessionDatabase-1.0... PASS!
Validating feature: ssl-1.0... PASS!
Validating feature: transportSecurity-1.0... PASS!
Validating feature: webCache-1.0... PASS!
Validating feature: webProfile-7.0... PASS!
Validating feature: websocket-1.1... PASS!
Product validation completed successfully.
```

Who wants (or needs) access to configuration?



1. ID the Liberty Server runs with

- Has a need to READ its configuration (and WRITE to logs, etc)
- Two options: server ID owns files, or is a separate ID from file owner

2. People Responsible for Administering the Server(s)

- They have a need to both READ and WRITE
- Bad practice: sharing login ID/password
- Good practice: Principle of least privilege- use sudo and `/etc/sudoers` file or for z/OS use SAF SURROGAT to switch to file owning ID

3. Other People Related to the Server's Activities

- Some have READ only ... logs, etc.
- Some may need limited WRITE ... then use "include" processing

4. Unauthorized People

- No access at all; key is insuring these fall into 'other' permission bit, and that is marked as '0'.

At the heart of this is a discussion of the UNIX file owner, group, and other permissions

Using securityUtility to encrypt passwords

- Best practice : use encryption for passwords instead of base64 or xor encoding
- **securityUtility** – located in <wlp_install_dir>/wlp/bin
 - Usage: securityUtility {encode|createSSLCertificate|createLTPAKeys|help} [options]
 - For encryption, use encode --key=encryption_key
 - Specifies the key to be used when encoding using AES encryption. This string is hashed to produce an encryption key that is used to encrypt and decrypt the password. The key can be provided to the server by defining the variable **wlp.password.encryption.key** whose value is the key. If this option is not provided, a default key is used.

```
./securityUtility encode --encoding=aes --key=myKey passW0rd  
{aes}AH00aXdiVD96u4oMRhoKeYH3U7aDqtFXTuHFBS098W1b
```

- Also supports 1-way hash encoding – for passwords in server.xml with basicRegistry
 - For hash, use encode --encoding=hash

```
./securityUtility encode --encoding=hash XXXXXXXX  
{hash}ATAAAAAlcqTmHn5qZahAAAAAIMjzy+hP8YFaIO6LiCreVe4etRLUS9a25eVuYtx6WKiv
```

https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/rwlp_command_securityutil.html

Liberty User registries

- User Registry Options with Liberty



Basic (and quickStartSecurity)

- The user ID and password values are maintained in the server.xml (or an include file)
- Adequate for initial validation and *some* testing, but **not for readiness testing, QA, or production use.**



LDAP

- Liberty access an LDAP server
- Commonly used when the user-population is large and dynamic
- **Best practice:** maintain bind password in a separate "include" file, not in server.xml



SAF (z/OS local)

- The user ID and password values are maintained in SAF (zosSecurity-1.0 feature)
- Very secure and very well-suited for production
- Can be an issue if the user-population is very large – there are options or filter/map LDAP to SAF



Federated Registries

- Multiple registries are employed: LDAP and SAF typically
- Often involves "distributed identity mapping" -- mapping an LDAP user to a SAF user



Custom

- Use the UserRegistry class to implement a custom registry
- Should be thoroughly reviewed before used in anything other than development and test

Network

How to secure remote access to Liberty and minimize risk of threats over the network?

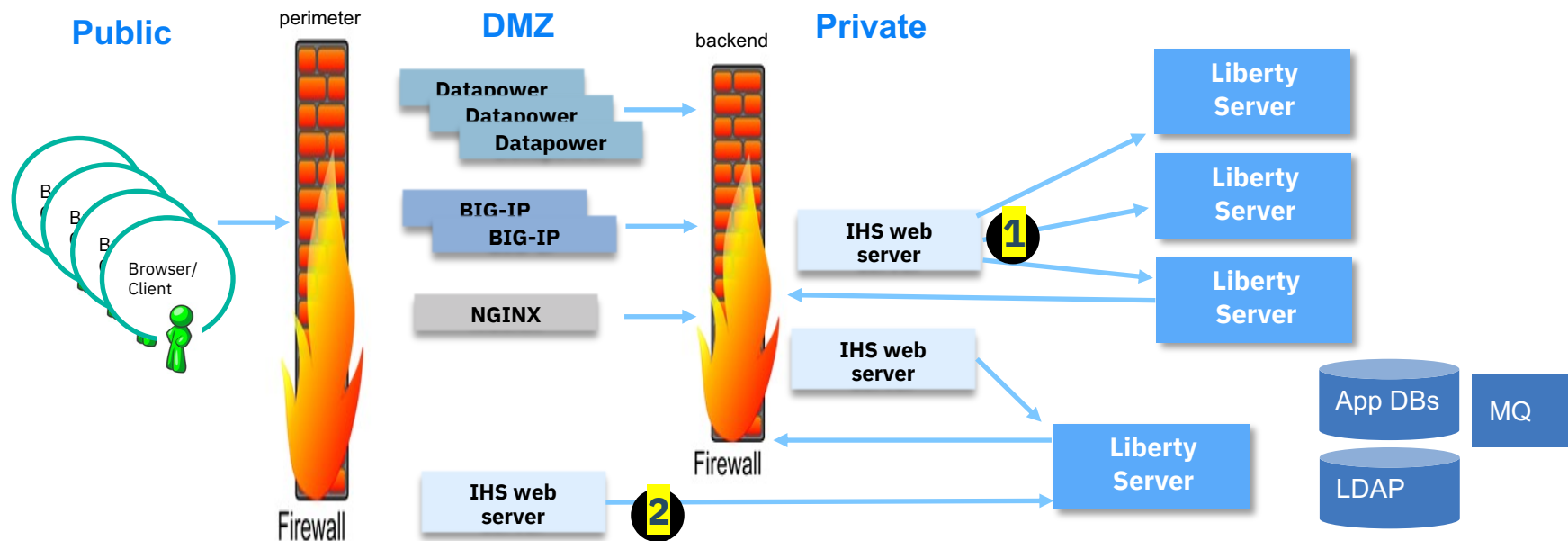


August 25th, 2021

Production environments belong inside a Firewall

- Shield production Liberty servers using fire wall technology – do not run them in the de-militarized zone (DMZ). Follow general networking best practices.
- The following devices are among the many choices that provide protection :
 1. Data Power
 2. BIG-IP F5, etc.
 3. Web servers
 - IHS Apache, NGINX, etc.

Typical topology



Some basic questions ...

1. How do I ensure my proxies/web servers can only connect to specific Liberty servers?
2. What are the implications for running the web server in the DMZ?

Production environments securing connectivity

- Several aspects for hardening connectivity in to Liberty.
 1. For protection at the transport level when coming in over a proxy (ie: web server): arrange at the network level for the required Liberty servers to be inaccessible to all but trusted proxies (web servers). This can be achieved using the **SSL tunnel trick** – as described here :

<https://community.ibm.com/community/user/wasdevops/blogs/james-mulvey1/2021/05/20/twas-security-hardening>

2. When using mutual authentication, in conjunction with a proxy (ie: web server) consider using the httpDispatcher parameter '**trustedHeaderOrigin**' option. This ensures only those IP origins listed will be permitted to login/authenticate.

```
<httpDispatcher trustedHeaderOrigin="10.20.30.40, 10.20.50.60"/>
```

https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.liberty.autogen.base.doc/ae/rwlp_c_onfig_httpDispatcher.html

3. When using mutual authentication and users log in directly to a Liberty server without a proxy/web server, you should set the following in the server.xml:

```
<webcontainer trusted="false"/> or
```

```
<httpDispatcher trustedHeaderOrigin="none"/>
```

For more details refer to 'web authentication trust risk' in this article:

<https://community.ibm.com/community/user/wasdevops/blogs/james-mulvey1/2021/05/20/twas-security-hardening>

Transport Level Security (TLS/SSL) is a must

- Only use Certificate Authority signed certificates (including product-created and your own organization's CAs) for public-facing production servers (no-self signed certs)
- Lock down TCP ports – only expose SSL enabled ports
 - Non SSL ports should be disabled where possible. This requires careful consideration of the server.xml file since Liberty features may automatically start non SSL ports.
 - Disable non-SSL ports by setting them to -1 (server.xml config):

```
<httpEndpoint id="defaultHttpEndpoint" httpPort="-1" httpsPort="9445" />  
<iiopEndpoint id="defaultIiopEndpoint" iiopPort="-1"> <iiopsOptions iiopsPort="9402"  
sslRef="defaultSSLConfig"/> </iiopEndpoint>  
<wasJmsEndpoint id="InboundJmsEndpoint" wasJmsPort="-1" wasJmsSSLPort="7288">  
</wasJmsEndpoint>
```
 - For each new fix pack review the new features and ensure hardening steps are applied.

See : http://www.ibm.com/support/knowledgecenter/SSAW57_liberty/com.ibm.websphere.wlp.nd.doc/ae/rwlp_portnums.html

Transport Level Security (TLS/SSL) is a must (2)

- Use feature `transportSecurity-1.0` over `ssl-1.0`

- `transportSecurity-1.0` is superset of `ssl-1.0`, adding outbound SSL configuration
- `<sslDefault>` `sslRef` attribute references a Liberty managed SSL configuration for inbound requests,
 - Points to the server's SSL trust and key files.
 - Doing so will result in any Liberty feature opening up new ports using the Liberty provided certificates and not an application (or JDK) provided one.
- With `transportSecurity-1.0` `<sslDefault>` **`outboundSSLRef`** new parameter references an application managed SSL configuration for *outbound calls*. This can now be configured as:

```
<sslDefault outboundSSLRef="defaultSSLConfig" sslRef="libertyConnectionConfig" />
```

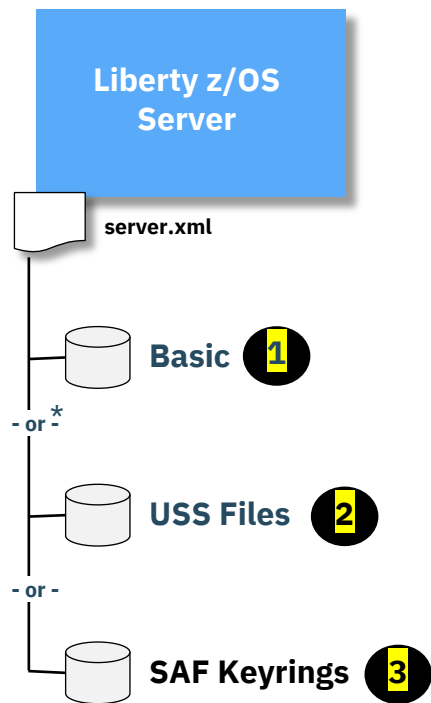
- Once a basic SSL configuration is set up, for outbound calling applications, consider an additional SSL configuration to permit fine grained management of outbound SSL calls. For example:

```
<ssl id="alternateSSLSettings" keyStoreRef="alternateKeyStore"
  trustStoreRef="alternateTrustStore" >
  <outboundConnection host="server99.ibm.com" port="8020" />
  <outboundConnection host="server99.ibm.com" port="9020" />
</ssl>
```

Note: for outbound calls, the trust store should be a trusted global CA (ie: your org's global CA)

In this example, when application code makes an outbound call to the 'server99.ibm.com' on either port 8020 or 9020, the alternateSSLSettings are used. This is done automatically (providing the application obtains its SSL Socket factory from `javax.net.ssl.SSLSocketFactory.getDefault()`).

z/OS : SSL/TLS considerations



* Combinations within a given Liberty z/OS server is possible.

1. Basic Key and Trust Store

- Simple one-line addition to the server.xml
`<keyStore id="defaultKeyStore" password="Liberty"/>`
- Satisfies *basic* requirements for TLS, but good only for initial validation. Not good for testing (self-signed certificate), and certainly *not* for production.

2. File-based Key and Trust Store

- Same mechanism as used on distributed platforms (keytool or ikeyman)
- Can be used for testing and production
- File password in server.xml can be encoded. SAF keyrings eliminate need for passwords

3. SAF-based Keyrings

- The server.xml file points to SAF as its key and trust store
- Use SAF keyrings to hold digital certificates and signer (CA) certificates
- No passwords in server.xml
- Access to SAF keyrings protected by SAF IRR.DIGTCERT.* profiles
- **Best practice: use z/OS facilities when on z/OS**

Secure links to LDAP servers

- Make sure Liberty links to LDAP servers is secure/encrypted.

```
<featureManager>
  <feature>transportSecurity-1.0</feature>
</featureManager>
```

```
<ssl id="ldapSSLConfig" keyStoreRef="ldapTrustStore" trustStoreRef="ldapTrustStore"/>
```

```
<keyStore id="ldapTrustStore" location="ldapTrustStore.jks" type="JKS"
  password="{aes}BQUFbm1sa2o=" />
```

Important: Make sure to encrypt the password and keep this config in an included/protected file from the server.xml

```
<ldapRegistry id="IBMDirectoryServerLDAP" realm="SampleLdapIDSRealm"
  host="host.domain.com" port="636" ignoreCase="true" baseDN="o=domain,c=us"
  ldapType="IBM Tivoli Directory Server" idsFilters="myidsfilters"
  sslEnabled="true"
  sslRef="ldapSSLConfig" />
```

Securing Cookies

- By default, cookies are sent by the browser over HTTPS or HTTP
 - Cookies support the attribute of “secure” which tells the browser to send the cookie over HTTPS only
- The cookie that carries HTTP session management information (JSESSIONID) should be protected
 - Set option in server.xml so session cookies are only sent over SSL (default is false):

```
<httpSession cookieSecure="true"/>
```

- The cookie that carries the LTPA Token (ltpaToken2) should be protected
 - If stolen, a third-party intruder can act as that user until the token expires
 - Set option in server.xml so LTPA cookies are only sent over SSL (default is false):

```
<webAppSecurity ssoRequiresSSL='true' />
```

Disable automated config and application updates

- By default, each server contains a monitored application directory called dropins/configDropins.
- When an application is placed in the dropins directory, the server automatically deploys and starts the application.
- When config updates are made to the server.xml or the configDropins directory, the server automatically deploys the config changes.
- Explicitly configured applications are recommended for production environments. Dropins monitoring should be disabled using:

```
<applicationMonitor updateTrigger="mbean" dropinsEnabled="false"/>
```

- Configuration updates should be carefully controlled in production environments. To disable automatic updates for config:

```
<config updateTrigger= "mbean"/>
```

https://www.ibm.com/support/knowledgecenter/en/SSEQTP_8.5.5/com.ibm.websphere.wlp.doc/ae/twlp_setup_dyn_upd.html

Other recommendations for production

- Configure applications to run on a virtual host instead of the default host for the server, to keep Liberty admin traffic separate from application requests.

https://www.ibm.com/support/knowledgecenter/en/SSEQTP_8.5.5/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts_ovr.html

- Disable default welcome page. The welcome page is enabled by default and accessing the root context “/” will display the Liberty homepage. For production, this can be disabled by setting `enableWelcomePage="false"` in the `server.xml` (default is `true`).

```
<httpDispatcher enableWelcomePage="false" />
```

- Remove server headers. Liberty server headers are enabled by default. Setting `removeServerHeader="true"` will remove server implementation information from HTTP headers (and also disables the default Liberty profile welcome page).

```
<httpOptions removeServerHeader="true"/>
```

- Consider disabling the *X-Powered-By* header if you do not want to reveal which server you are running. Use this custom property to disable the *X-Powered-By* header, which prevents the header from being sent on the HTTP response (default value is `false`).

```
<webContainer disableXPoweredBy="true"/>
```

Other recommendations for production (cont)

- Disable HTTP session overflow. Applications that use in-memory sessions should be restricted with the number of sessions that can be created to prevent a denial-of-service attack whereby the attacker continually generates new sessions until all memory in the JVM is exhausted.

```
<httpSession allowOverflow="false" maxInMemorySessionCount="1000" alwaysEncodeUrl="true"
cookieSecure="true" cookieHttpOnly=true/>
```

- Secure access to Liberty JMX connector for remote admin services in the Web Server plugin. The plugin should have the following entries removed/commented out.

```
<!-- <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId" Name="/ibm/api/*"/>
<Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionId" Name="/IBMJMXConnectorREST/*"/> -->
```

- LTPA keys should be replaced regularly – with Liberty this needs to be done manually.
 - Delete the ltpa.keys file and allow the server to regenerate them
 - Then copy the new keys file to file systems for other servers that share the keys

Applications

Application construction and configuration hardening



August 25th, 2021

JEE Role based authorization and Liberty



Who Are You?

This is **authentication**. We looked at that earlier. Once a user successfully authenticates, the server knows who the user is.

What Are You Allowed To Do?

This is **authorization**. It is a function of the application. An application may or may not define different "roles," but if roles are defined, then the server gets involved to help the application determine if a user is a member of the defined role.



Application WAR File web.xml

```
<servlet>
  <servlet-name>myHello</servlet-name>
  <servlet-class>HelloServlet</servlet-class>
  <security-role>
    <role-name>MyRole</role-name>
  </security-role>
</servlet>
```

role

Liberty server.xml

```
<feature>appSecurity-2.0</feature>
<application
  id="myServlet" name="myServletWAR" type="war"
  location="/<path>/myServletWAR.war" >
  <application-bnd>
    <security-role name="MyRole">
      <group name="Operators" />
    </security-role>
  </application-bnd>
</application>
```

role

group

If the authenticated user is a member of this group, then they have access to that role.

If no application-bnd in server.xml (no explicit authorization) the default behavior is to use the group name for the role.

Understanding Java EE roles and priorities

- Java EE security relies on roles and mappings to users and groups in a registry. Annotations and the web.xml are used to define the roles and the ibm-application-bnd.xml* and the server.xml files are used to bind the roles to the users or groups.
- For conflicts in the role mapping for a servlet between an application's web.xml file and annotations in the application, the configuration in the web.xml takes precedence (as defined by the servlet specification). If there is a conflict between the ibm-application-bnd.xml* and server.xml files, then the configuration in the server.xml file takes precedence.
- When the role mapping binding information for a protected application is not provided, the group name and role name need to be the same. For example, if the role name is 'manager', then a user who belongs to a 'manager' group has access to that resource. This applies only when no application bind information is specified for the application (in the server.xml or the application binding file).

Application based preventative measures

- Never set Web server document root to WAR
 - WAR files contain application code and lots of sensitive information. Only some of that information is Web-servable content, and so it is inappropriate to set the Web server document root to the WAR root. If you do this, the Web server will serve up all the content of the WAR file without interpretation, resulting in code, raw JSPs, and more being served up to your users.
- Do not serve servlets by class name
 - Servlets can be served by class name or via a normal URL alias. Typically, applications choose the latter. That is, developers define a precise mapping from each URL to each servlet class in the web.xml.
 - Liberty does let you serve servlets by class name. Instead of defining a mapping for each servlet, a single generic URL (such as /servlet) serves all servlets. The component of the path after the base is assumed to be the class name for the servlet. For example, "/servlet/com.ibm.sample.MyServlet" refers to the servlet class "com.ibm.sample.MyServlet."
 - Serving servlets by class name is accomplished by setting the **serveServletsByClassnameEnabled** property to true in the ibm-web-ext.xmi file. **Do not enable this feature.** It makes it possible for anyone that knows the class name of any servlet to invoke it directly.

Application based preventative measures (cont)

- Carefully verify that every servlet alias is secure

- Servlets are secured by URL. Each URL that is to be secured must be specified in the web.xml file describing the application. If a servlet has more than one alias (that is, multiple URLs access the same servlet class) or there are many servlets, it is easy to accidentally forget to secure an alias. Be cautious. Since Liberty secures URLs and not the underlying classes, **if just one servlet URL is insecure, an intruder might be able to bypass security**. To alleviate this, use wildcards to secure servlets wherever possible. If that is not appropriate, carefully double check your web.xml file before deployment.
- Make sure that when you specify authorization constraints for servlets that you **either list no methods, or very carefully list (likely in multiple constraints) ALL the methods** for a servlet (GET, POST, PUT, HEAD, and so on).

Best practice ('safe' example below as default). "Default to deny" design. New end-points added to the web application are unreachable by default. If a new servlet is added, you will have to add a security constraint for the new URL in order for anyone to use it.

Unsafe

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>myservlet</web-resource-name>
    <url-pattern>/myservlet</url-pattern>
    <http-method>GET</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>arole</role-name>
  </auth-constraint>
</security-constraint>
```

Safe

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>DefaultDeny</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>NoAccess</role-name>
  </auth-constraint>
</security-constraint>
... or for Servlet 3.1 - use: <deny-uncovered-http-methods/>
or <http-method-omission> elements
```

Application based preventative measures (cont)

- Do not place sensitive information in the WAR root
 - WAR files contain servable content. The Web container will serve any files found in the root of the WAR file. This is fine as long as you place only servable content in the root. Thus, **you should never place content that shouldn't be shown to users in the root of the WAR.** For example, don't put property files, class files, or other important information there. If you must place such information in the WAR file, place it within the WEB-INF directory, as permitted by the servlet specification. Information there is never served by the Web container.
- Define a default error handler
 - When errors occur in a Web application -- or even before the application dispatch (for example, if Liberty can't find the target servlet) -- an error message is displayed to the user. By default, the app server will display a raw exception stack dump of the error. Not only is this incredibly unfriendly to the user, it also reveals information about the application (names of classes and methods are in the stack information). The exception message text is also displayed, which could contain sensitive information.
 - It is best to ensure that users never see a raw error message by defining a default error page that displays whenever an unhandled exception occurs. This page can be a user friendly error message rather than a stack trace. The default error page is defined in ibm-web-ext.xmi using the defaultErrorPage attribute.

<https://blogs.oracle.com/arungupta/totd-136:-default-error-page-using-servlets-30-improved-productivity-using-java-ee-6>

Application based preventative measures (cont)

- Consider disabling file serving and directory browsing
 - You can further limit the risk of serving inappropriate content by disabling file serving and directory browsing in your Web applications. Set in `ibm-web-ext.xmi` (the default is false):

```
<enable-directory-browsing value="false"/>
```
 - Of course, if the WAR contains servable static content, file serving will have to be enabled. There is rarely reason to enable directory browsing.
- Ensure applications specify transport guarantee of confidential in `web.xml` (or `ibm-web-ext.xmi`):

```
<user-data-constraint>  
  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  
</user-data-constraint>
```

Application design and implementation

Advanced security hardening in WebSphere Application Server V7, V8 and V8.5, Part 2

<https://community.ibm.com/community/user/wasdevops/blogs/james-mulvey1/2021/05/20/twas-security-hardening>

- Use WebSphere Application Server security to secure applications
 - Do not rely on HTTP session for user identity
 - Secure every layer of the application
 - Validate all user input
 - Write secure applications
 - Store information securely
 - Be sensitive to auditing and tracing
 - Avoid the use of basic authentication for browser clients
 - Avoid widget jacking: Use SSL if you build your browser UI using GETs from third party sites
- **Highly recommend studying this material and incorporating it into your application design**

Best practice: Run IBM AppScan or a similar product on your applications as early as possible to identify vulnerabilities. <https://www.ibm.com/security/application-security/appscan>

Recent Updates



Periodic Table of Liberty (21.0.0.9)



zOS

ND

Base

Core

Open Liberty

New in 4Q20

New in 3Q21

New in 2Q21

New in 1Q21

batchSMFLogging-1.0		zosLocalAdapters-1.0		zosTransaction-1.0		zosSecurity-1.0	
collectiveController-1.0		dynamicRouting-1.0		healthManager-1.0		scalingController-1.0	
clusterMember-1.0		healthAnalyzer-1.0		scalingMember-1.0		Security	
cloudant-1.0	heritageAPIs-1.0	batchManagement-1.0	Operations		acmeCA-1.0		
javaee-7.0	sipServlet-1.1	wsAtomicTransaction-1.2			passwordUtilities-1.0		
javaee-8.0					wsSecurity-1.1		
jakartaee-8.0					wsSecuritySaml-1.0		
bells-1.0	microProfile-4.1	adminCenter-1.0	audit-1.0	ldapRegistry-3.0			
concurrent-1.0	mpContextPropagation-1.2	collectiveMember-1.0	constrainedDelegation-1.0	oauth-2.0			
grpc-1.0	mpGraphQL-1.0	distributedMap-1.0	federatedRepository-1.0	openid-2.0			
javaMail-1.6	mpReactiveMessaging-1.0	eventLogging-1.0	jwt-1.0	openidConnectClient-1.0			
jaxb-2.2	mpReactiveStreams-1.0	logstashCollector-1.0	jwtSso-1.0	openidConnectServer-1.0			
jdbc-4.3	opentracing-1.3	monitor-1.0	sessionDatabase-1.0	samlWeb-2.0			
jpaContainer-2.2	osgiConsole-1.0	openapi-3.1	webCache-1.0	scim-1.0			
jsfContainer-2.3	springBoot-2.0	requestTiming-1.0		socialLogin-1.0			
json-1.0	webProfile-7.0	usageMetering-1.0		spnego-1.0			
jsonbContainer-1.0	webProfile-8.0	restConnector-2.0		transportSecurity-1.0			
jsonpContainer-1.1	APIs	sessionCache-1.0					

Focus areas



Developer Experience

APIs

Foundation

Orchestration

Security

Liberty Last Quarter Review

Open Liberty



Security

Dev Exp

- Liberty starter
- Run guides in cloud

Foundation

- Server Endpoint Control Mbean isActive
- Images published to IBM Container Registry

API

- MicroProfile 4.1
- MicroProfile Health 3.1

Orchestration

- BASIC trace format for messages.log
- Tx Peer Recovery

Liberty Starter



<https://start.openliberty.io>

Create a starter application

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

Group

com.demo

Artifact

app-name

Build Tool Maven Gradle

Java SE Version 11

Java EE/Jakarta EE Version 8

MicroProfile Version 4.0

Generate Project

app-name

- Dockerfile
- mvnw
- mvnw.cmd
- pom.xml
- src/main
 - liberty/config/
 - server.xml
 - java/com/demo/rest
 - RestApplication.java

Container Images in ICR.io



- Container images are now also available via IBM Container Registry
- No authentication required
- Download directly from IBM
- No rate limits

```
FROM docker.io/openliberty/open-liberty:kernel-java11-openj9-ubi
```



```
FROM icr.io/appcafe/open-liberty:kernel-java11-openj9-ubi
```

Container Images in ICR.io



- Container images are now also available via IBM Container Registry
- No authentication required
- Download directly from IBM
- No rate limits

```
FROM docker.io/ibmcom/websphere-liberty:kernel-java11-openj9-ubi
```



```
FROM icr.io/appcafe/websphere-liberty:kernel-java11-openj9-ubi
```

Container Images in icr.io

Open Liberty



icr.io/appcafe/open-liberty

kernel-java11-openj9-ubi

kernel-java8-openj9-ubi

kernel-java8-ibmjava-ubi

full-java11-openj9-ubi

full-java8-openj9-ubi

full-java8-ibmjava-ubi



Container Registry

docker.io/openliberty/open-liberty

kernel-java11-openj9-ubi

kernel-java8-openj9-ubi

kernel-java8-ibmjava-ubi

full-java11-openj9-ubi

full-java8-openj9-ubi

full-java8-ibmjava-ubi



Docker Hub

Container Images in icr.io

Open Liberty



icr.io/appcafe/websphere-liberty

kernel-java11-openj9-ubi

kernel-java8-openj9-ubi

kernel-java8-ibmjava-ubi

full-java11-openj9-ubi

full-java8-openj9-ubi

full-java8-ibmjava-ubi



Container Registry

docker.io/ibmcom/websphere-liberty

kernel-java11-openj9-ubi

kernel-java8-openj9-ubi

kernel-java8-ibmjava-ubi

full-java11-openj9-ubi

full-java8-openj9-ubi

full-java8-ibmjava-ubi



Docker Hub

MicroProfile Health 3.1

Open Liberty



Liveness

- Restart on failure

Readiness

- Do not route traffic on failure

Startup

- Do not route traffic or restart until timeout or pass

```
@Startup
@ApplicationScoped
public class StartupCheck {
    public HealthCheckResponse call() {
        // Compute whether started and return
    }
}
```

WebSphere traditional BASIC logs

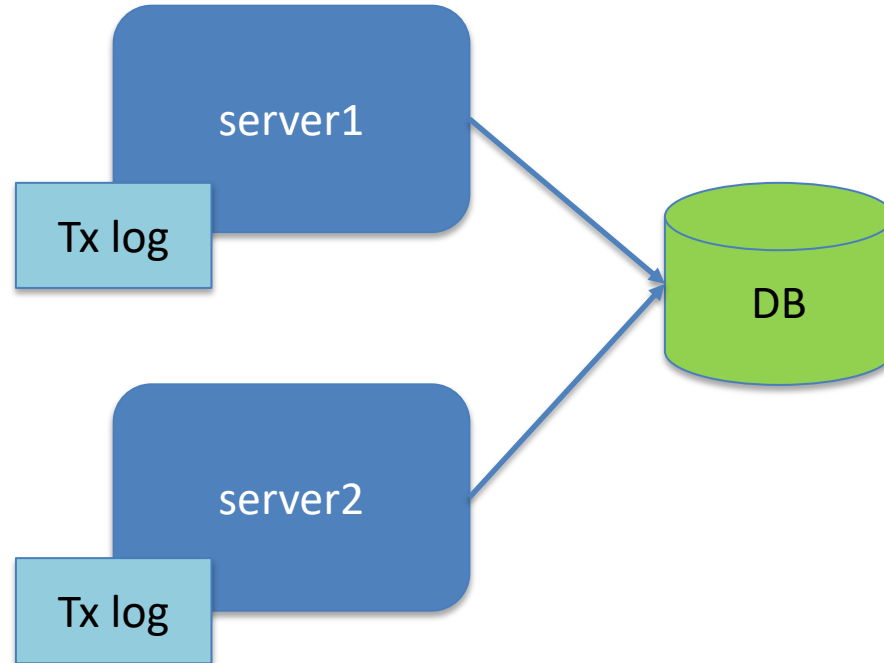


- WebSphere Liberty can now write tWAS BASIC formatted logs
- Allows the use of existing log parsers
- Format is called TBASIC

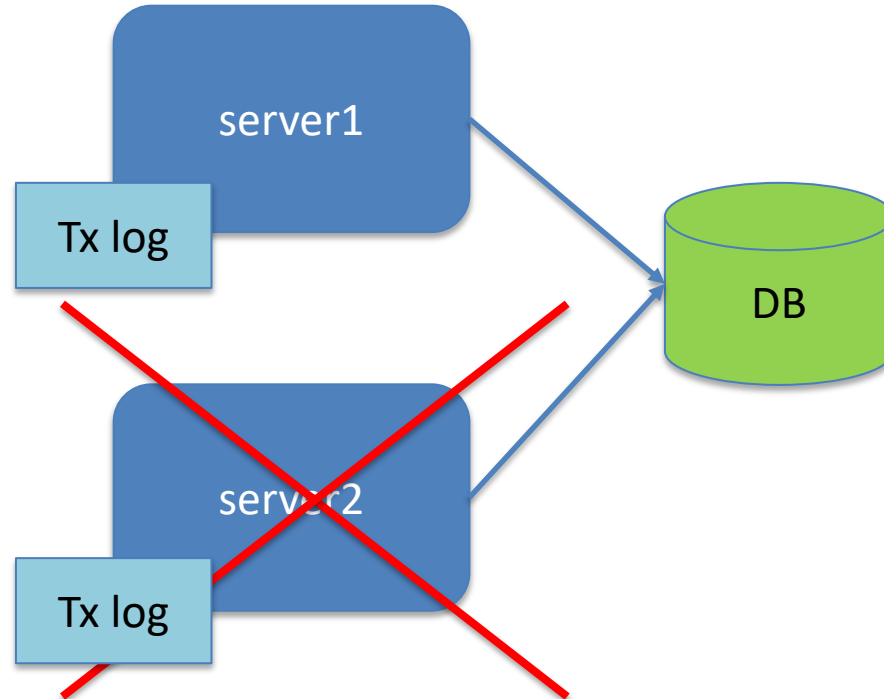
server.xml	bootstrap.properties	Env Var	Values
consoleFormat	com.ibm.ws.logging.console.format	WLP_LOGGING_CONSOLE_FORMAT	dev, simple, tbasic, json
messageFormat	com.ibm.ws.logging.message.format	WLP_LOGGING_MESSAGE_FORMAT	simple, tbasic, json

```
[24/03/21 15:04:10:331 EDT] 00000001 FrameworkMana A CWWKE0001I: The server defaultServer has been launched.
[24/03/21 15:04:11:338 EDT] 00000001 FrameworkMana I CWWKE0002I: The kernel started after 1.177 seconds
[24/03/21 15:04:11:465 EDT] 0000003e FeatureManage I CWWKF0007I: Feature update started.
[24/03/21 15:04:11:635 EDT] 00000033 DropinMonitor A CWWKZ0058I: Monitoring dropins for applications.
```

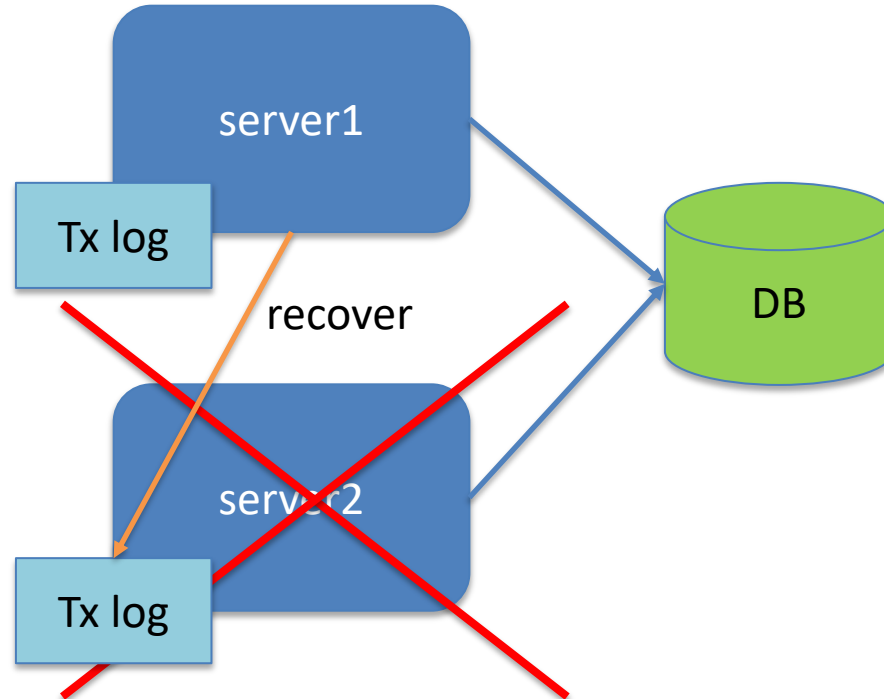

Transactional Peer Recovery



Transactional Peer Recovery



Transactional Peer Recovery



Transactional Peer Recovery



- Transaction log must be visible to recovering server
 - Store in shared file system
(In Kubernetes requires RWX Persistent Volume)
 - Store in Database
- Configure recovery group and server identity

```
<server>  
  <transaction  
    recoveryGroup="myApp"  
    serverIdentity="${HOSTNAME}${wlp.server.name}" />  
</server>
```

Labs, Questions

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)



Liberty Quarterly Update_20.0.0.4-6.final.pdf



LibertyPoT_20.0.0.6_WIN.zip V2



LibertyPoT_20.0.0.6_MAC.zip V2



LibertyPoT_20.0.0.6_LINUX.zip V2



labs_n_presentations_only.zip V2



LibertyResourceList.pdf V2

charts only

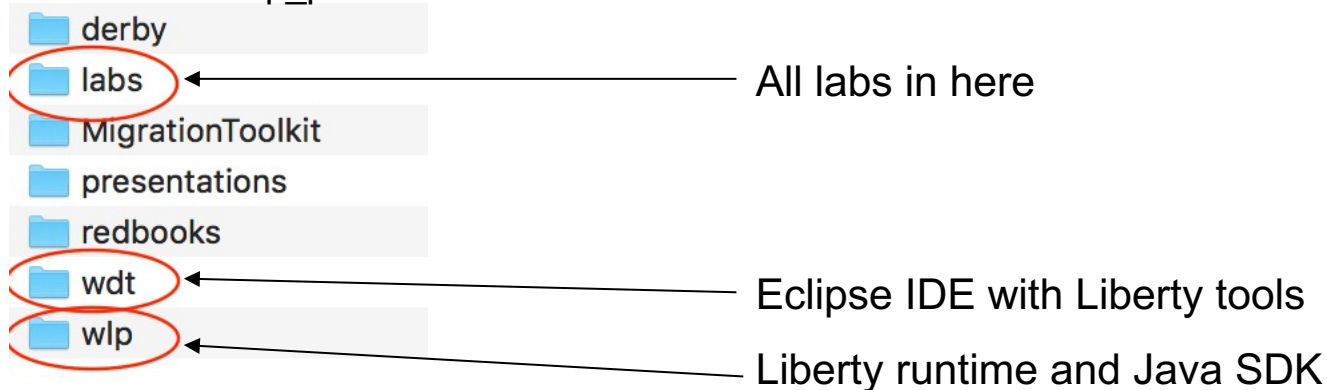
charts & lab instructions
only

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](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://hub.docker.com/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

Open Liberty



Liberty 21.0.0.7-9 Update

~~Session#1: Sep 22, 2021 from 1-3pm ET - <http://ibm.biz/Liberty-Sep22>~~

Session#2: Sep 29, 2021 from 9-11am ET - <http://ibm.biz/Liberty-Sep29>

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>



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