WebSphere Application Server Liberty Profile and Docker

David Currie

david_currie@uk.ibm.com @dcurrie

Agenda

- What is the promise of containers?
- Overview of Docker
- Docker and IBM
- WebSphere Application Server and Docker

What is Docker?

“Docker is an open-source project to easily create
lightweight,
portable,
self-sufficient
containers from any application.”

It works for me!

	David's Desktop	Ian's Laptop	Test	Staging	Data Center	Cloud VM
Web Server	?	?	?	?	?	?
App Server	?	?	?	?	?	?
Database	?	?	?	?	?	?
Messaging	?	?	?	?	?	?

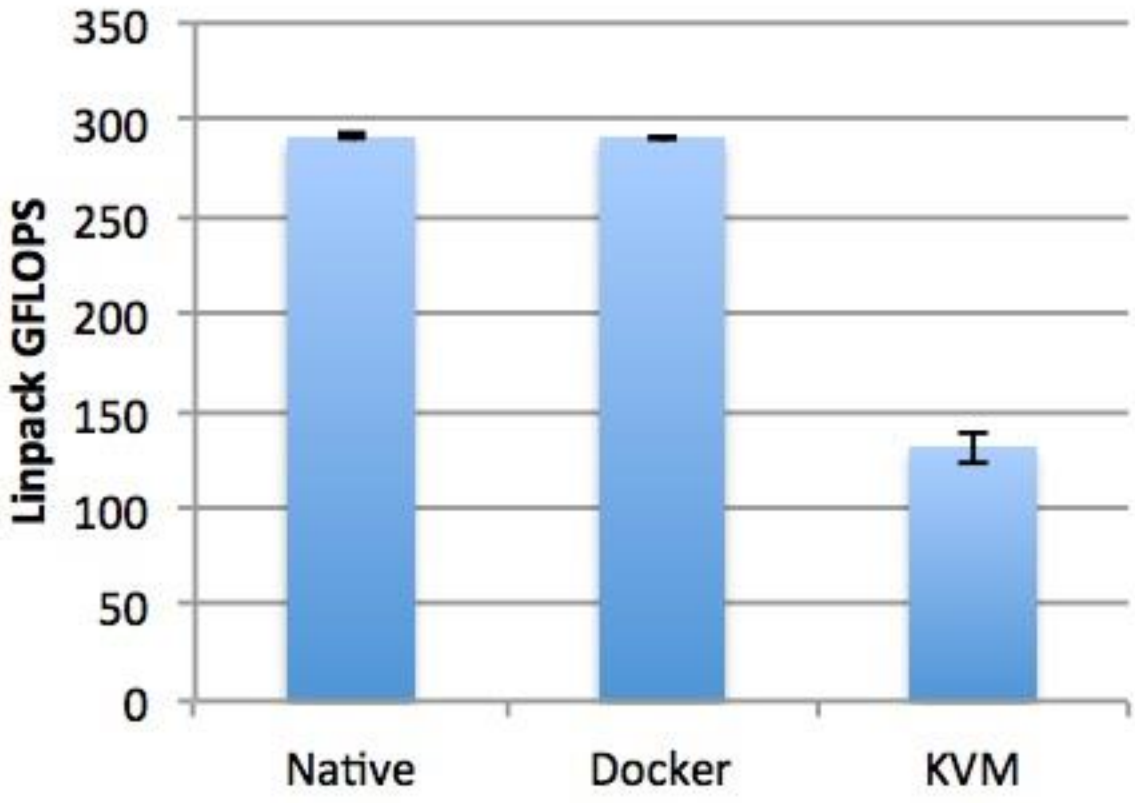
Containers Everywhere

	David's Desktop	Ian's Laptop	Test	Staging	Data Center	Cloud VM
Web Server						
App Server						
Database						
Messaging						

Speed

	Obtain within…	Manual deployment takes…	Automated deployment takes…	Starts in…
Bare Metal	Days	Hours	Minutes	Minutes
VM	Minutes	Minutes	Seconds	< Minute
Container	Seconds	Minutes	Seconds	Seconds

Near Bare-Metal Performance



http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf

Dev vs Ops

- Dev inside the container
    - Code
    - Libraries
    - Server runtime
    - Configuration

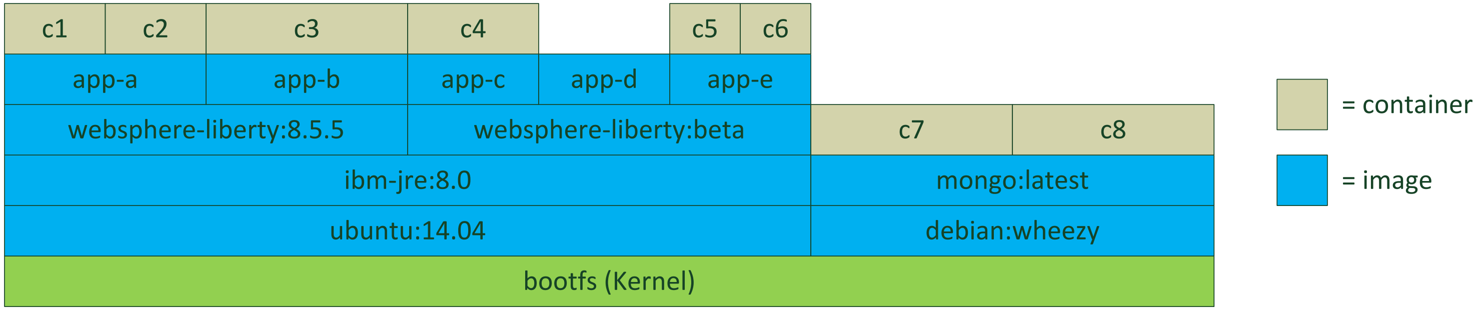
- Ops outside the container
    - Logging
    - Remote access
    - Network configuration
    - Monitoring

Builds on existing technologies

- Namespaces
    - e.g. pid, net, ipc, mount
- CGroups
    - Control resources for a group of processes
    - e.g. memory, CPU
- Layered file system
    - aufs, btrfs, device mapper, overlayfs

Docker Terminology

- **Image** – layered file system where each layer references the layer below
- **Dockerfile** – build script that defines:
  - an existing image as the starting point
  - a set of instructions to augment that image (each of which results in a new layer in the file system)
  - meta-data such as the ports exposed
  - the command to execute when the image is run
- **Container** – runtime instance of an image plus a read/write layer
- **Docker Hub** – centralised repository of Docker images



c1 c2 c3 c4 c5 c6

app-a app-b app-c app-d app-e

websphere-liberty:8.5.5 websphere-liberty:beta c7 c8

ibm-jre:8.0 mongo:latest

ubuntu:14.04 debian:wheezy

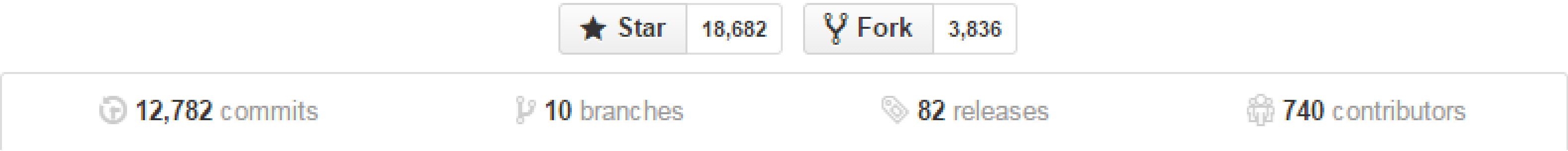
bootfs (Kernel)

▨ = container

■ = image

Docker Timeline

Jan 2013	First commit
March 2013	Docker 0.1.0 released
April 2014	Docker Governance Advisory Board announced with representation from IBM
June 2014	Docker 1.0 released
September 2014	$40 million investment round
December 2014	Docker announces Machine, Swarm and Compose
December 2014	Docker and IBM announce strategic partnership



★ Star 18,682 Fork 3,836

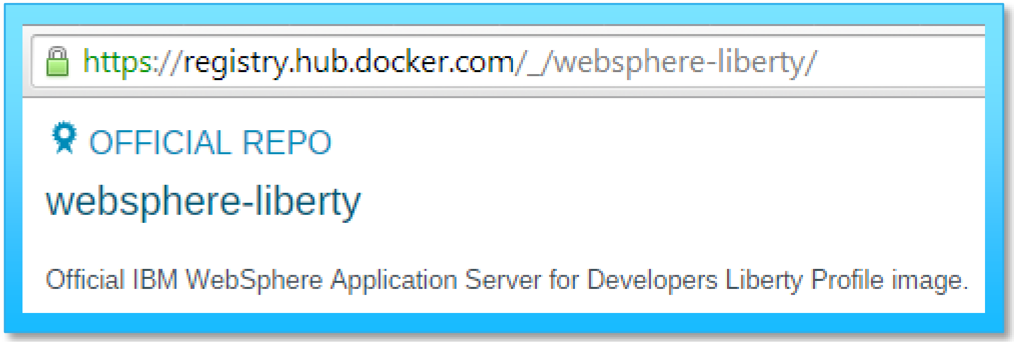
12,782 commits 10 branches 82 releases 740 contributors

Docker and IBM

- Member of the Governance Advisory Board
- Strategic partnership with Docker
  - IBM will deliver Docker Hub Enterprise (DHE) as an on-premise solution
- IBM Containers service on Bluemix
- Docker ports to Linux on Power and System z
- Docker support for Pure Patterns
- Containerised IBM software

Docker and WebSphere Application Server

- Support for WebSphere Application Server Liberty Profile and Full Profile running under Docker
- WAS Liberty images on Docker Hub for Development use
  - Latest WAS V8.5.5. Liberty driver
  - WAS Liberty V9 Beta with Java EE 7
- Dockerfiles on WASdev GitHub to:
  - Upgrade the Docker Hub image with Liberty Base or ND commercial license
  - Build your own Docker image for Liberty (Core, Base or ND)



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

OFFICIAL REPO

websphere-liberty

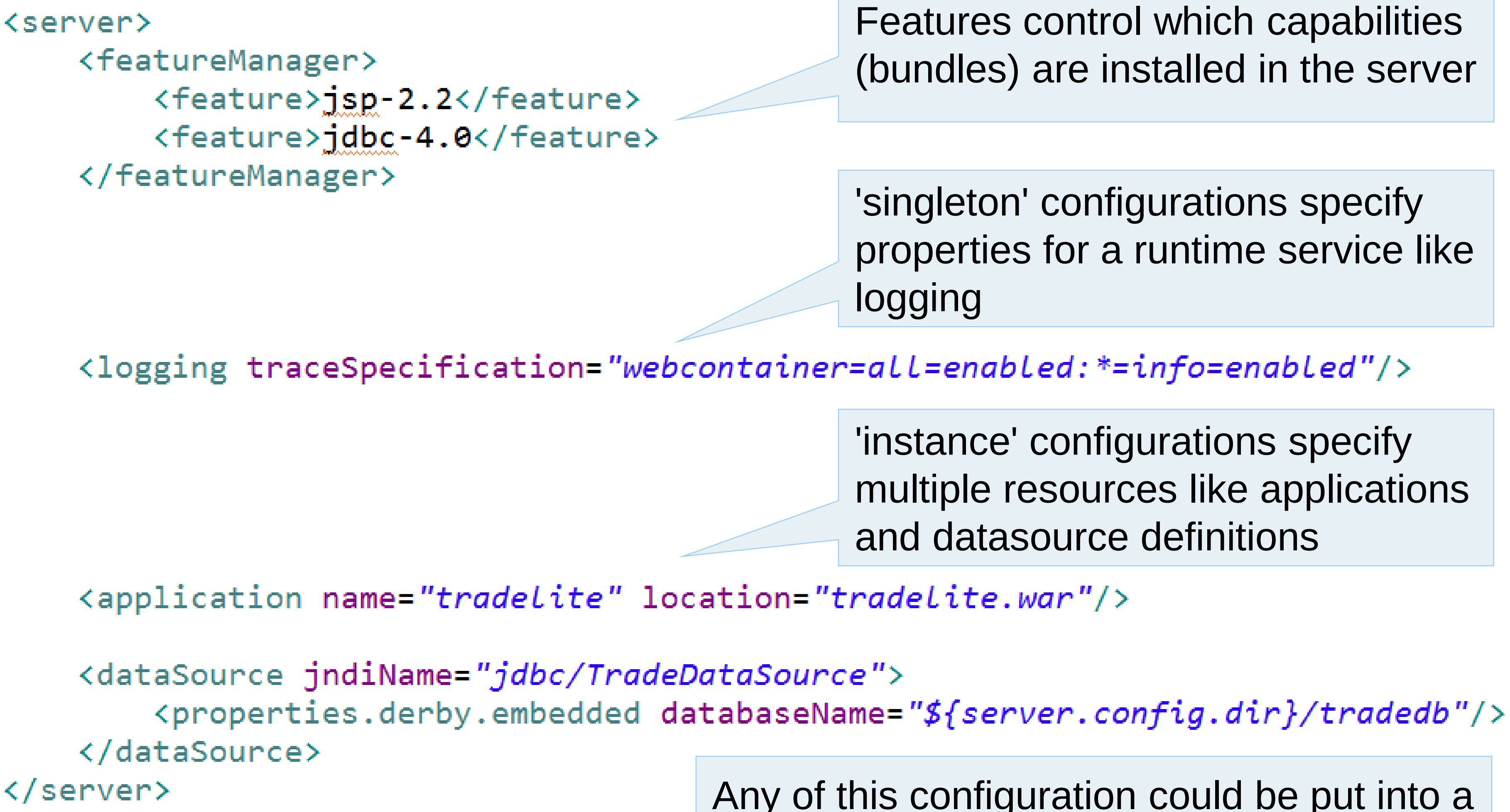
Official IBM WebSphere Application Server for Developers Liberty Profile image.

What is the WebSphere Liberty Profile?

For rapid development and light-weight production deployment with high scalability

- Web Profile certified (Liberty Core Edition)
- Small footprint (< **54MB**), quick startup (< **3 sec**)
- Developer-first design of simple, shareable XML configuration
- Dynamic runtime and configuration
- Unzip install and deploy
- Fidelity to WebSphere Application Server Full Profile
- Monitoring and management through Admin Center or scripting
- Install new features from repository with no server restart
- Lightweight collective management scales to 10,000 servers

Simple Server Configuration



```
<server>
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>jdbc-4.0</feature>
    </featureManager>

    <logging traceSpecification="webcontainer=all=enabled:*=info=enabled"/>

    <application name="tradelite" location="tradelite.war"/>

    <dataSource jndiName="jdbc/TradeDataSource">
        <properties.derby.embedded databaseName="${server.config.dir}/tradedb"/>
    </dataSource>
</server>
```

Features control which capabilities (bundles) are installed in the server

'singleton' configurations specify properties for a runtime service like logging

'instance' configurations specify multiple resources like applications and datasource definitions

Any of this configuration could be put into a separate xml file and 'included' in this 'master' configuration file

WebSphere Liberty Repository:

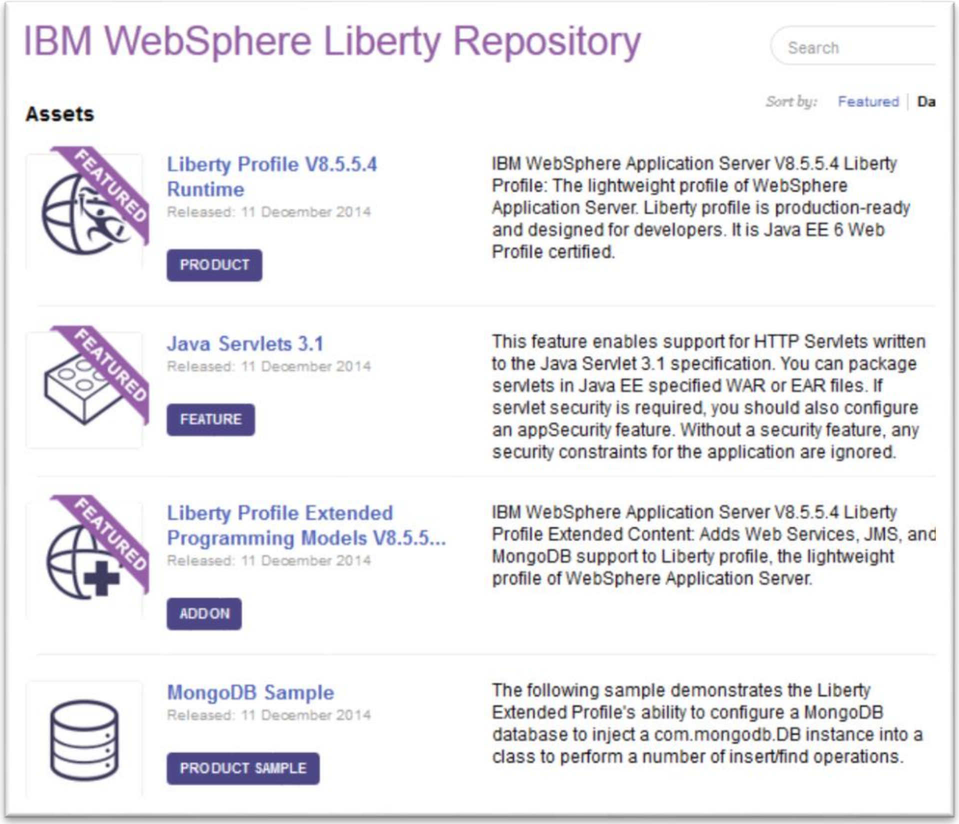
Key to Continuous Delivery of Business Value

www.wasdev.net

The Liberty Repository holds all of the content that you can install on the Liberty Kernel to make your own customized app server container.

- Repo hosts IBM code as well as open source integration samples for popular open source pkgs
- Enables our continuous delivery
- Serves as the base for dynamically built Buildpacks for Bluemix

Liberty Kernel



IBM WebSphere Liberty Repository

Search

Assets

Sort by: Featured | Da

FEATURED

Liberty Profile V8.5.5.4 Runtime

Released: 11 December 2014

PRODUCT

IBM WebSphere Application Server V8.5.5.4 Liberty Profile: The lightweight profile of WebSphere Application Server. Liberty profile is production-ready and designed for developers. It is Java EE 6 Web Profile certified.

FEATURED

Java Servlets 3.1

Released: 11 December 2014

FEATURE

This feature enables support for HTTP Servlets written to the Java Servlet 3.1 specification. You can package servlets in Java EE specified WAR or EAR files. If servlet security is required, you should also configure an appSecurity feature. Without a security feature, any security constraints for the application are ignored.

FEATURED

Liberty Profile Extended Programming Models V8.5.5...

Released: 11 December 2014

ADDON

IBM WebSphere Application Server V8.5.5.4 Liberty Profile Extended Content: Adds Web Services, JMS, and MongoDB support to Liberty profile, the lightweight profile of WebSphere Application Server.

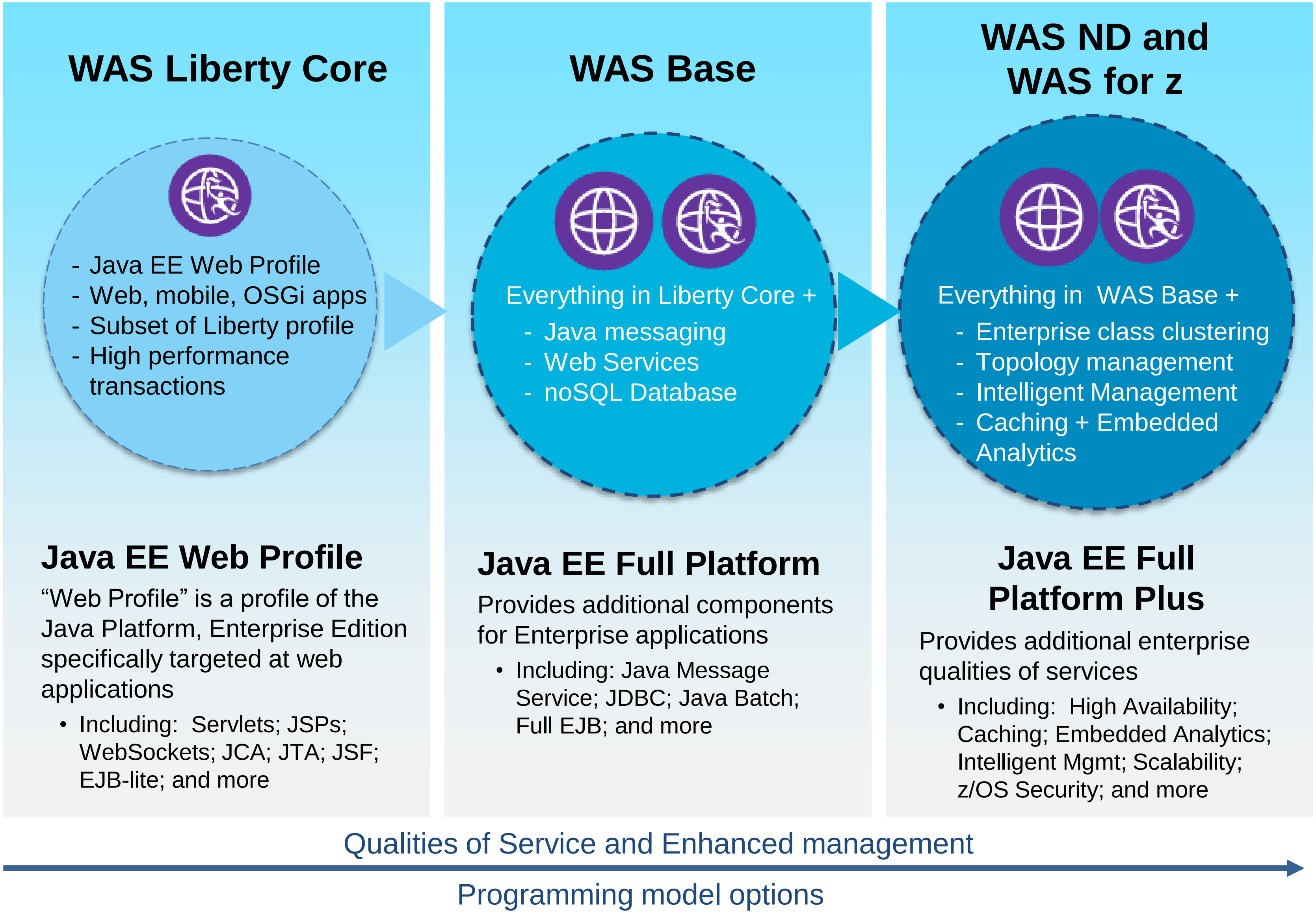
MongoDB Sample

Released: 11 December 2014

PRODUCT SAMPLE

The following sample demonstrates the Liberty Extended Profile's ability to configure a MongoDB database to inject a com.mongodb.DB instance into a class to perform a number of insert/find operations.

WebSphere Application Server 2015



WAS Liberty Core

- Java EE Web Profile
- Web, mobile, OSGi apps
- Subset of Liberty profile
- High performance transactions

Java EE Web Profile

“Web Profile” is a profile of the Java Platform, Enterprise Edition specifically targeted at web applications

- Including: Servlets; JSPs; WebSockets; JCA; JTA; JSF; EJB-lite; and more

WAS Base

Everything in Liberty Core +

- Java messaging
- Web Services
- noSQL Database

Java EE Full Platform

Provides additional components for Enterprise applications

- Including: Java Message Service; JDBC; Java Batch; Full EJB; and more

WAS ND and WAS for z

Everything in WAS Base +

- Enterprise class clustering
- Topology management
- Intelligent Management
- Caching + Embedded Analytics

Java EE Full Platform Plus

Provides additional enterprise qualities of services

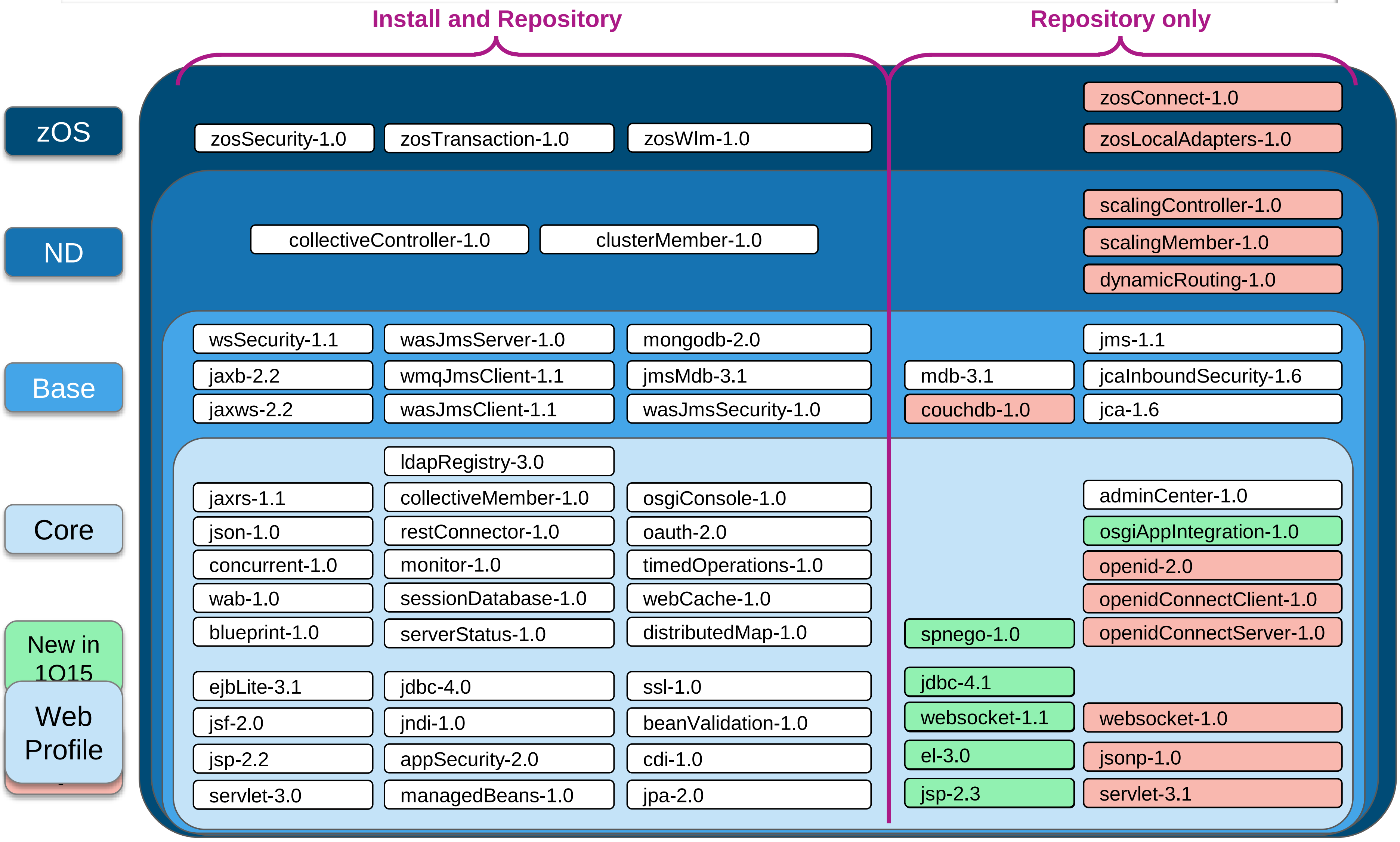
- Including: High Availability; Caching; Embedded Analytics; Intelligent Mgmt; Scalability; z/OS Security; and more

Qualities of Service and Enhanced management

Programming model options

A Truly Composable and Dynamic Server

Based on configurable features



Install and Repository

Repository only

zOS

ND

Base

Core

New in 1Q15

Web Profile

zosSecurity-1.0
zosTransaction-1.0
zosWlm-1.0
zosConnect-1.0
zosLocalAdapters-1.0

collectiveController-1.0
clusterMember-1.0
scalingController-1.0
scalingMember-1.0
dynamicRouting-1.0

wsSecurity-1.1
wasJmsServer-1.0
mongodb-2.0
jms-1.1
jaxb-2.2
wmqJmsClient-1.1
jmsMdb-3.1
mdb-3.1
jcaInboundSecurity-1.6
jaxws-2.2
wasJmsClient-1.1
wasJmsSecurity-1.0
couchdb-1.0
jca-1.6

ldapRegistry-3.0
jaxrs-1.1
collectiveMember-1.0
osgiConsole-1.0
adminCenter-1.0
json-1.0
restConnector-1.0
oauth-2.0
osgiAppIntegration-1.0
concurrent-1.0
monitor-1.0
timedOperations-1.0
openid-2.0
wab-1.0
sessionDatabase-1.0
webCache-1.0
openidConnectClient-1.0
blueprint-1.0
serverStatus-1.0
distributedMap-1.0
spnego-1.0
openidConnectServer-1.0

ejbLite-3.1
jdbc-4.0
ssl-1.0
jdbc-4.1
jsf-2.0
jndi-1.0
beanValidation-1.0
websocket-1.1
websocket-1.0
jsp-2.2
appSecurity-2.0
cdi-1.0
el-3.0
jsonp-1.0
servlet-3.0
managedBeans-1.0
jpa-2.0
jsp-2.3
servlet-3.1

Docker Quick Start

- Linux – run natively e.g. on Ubuntu
  - sudo apt-get install docker.io
  - curl -sSL https://get.docker.com/ubuntu/ | sudo sh

- Windows/Mac – run in VM under VirtualBox
  - http://boot2docker.io/

- Docker Machine (https://github.com/docker/machine)
  - docker machine -d virtual-box dev
  - docker machine -d openstack … test
  - docker machine -d softlayer … prod

Running a Liberty server under Docker

```
$ docker run -e LICENSE=accept websphere-liberty

Unable to find image 'websphere-liberty:latest' locally
websphere-liberty:latest: The image you are pulling has been verified
511136ea3c5a: Pull complete
53f858aaaf03: Pull complete
837339b91538: Pull complete
…
74cc1db7db68: Pull complete
2d01e6a9e5c9: Pull complete
da20ffbb1921: Pull complete
Status: Downloaded newer image for websphere-liberty:latest
Launching defaultServer (WebSphere Application Server 8.5.5.4/wlp-1.0.7.cl50420141211-1039) on IBM J9 VM, version pxa6470_27sr1fp1-20140708_01 (SR1 FP1) (en_US)
[AUDIT   ] CWWKE0001I: The server defaultServer has been launched.
…
```

Useful Docker commands

- Run in the background with a port exposed on the host
  - docker **run** -e LICENSE=accept -p 80:9080 -d --name wlp websphere-liberty
  - docker **logs** --tail=all -f wlp

- Or allow Docker to allocate ports if running multiple containers
  - C=$(docker **run** -e LICENSE=accept -P -d websphere-liberty)
  - docker **port** $C 9080
  - docker **logs** --tail=all -f $C

- And when you’re done
  - docker **stop** $C
  - docker **rm** $C

Mounting a Local Volume for Development

```
docker run -d -e LICENSE=accept -p 9080:9080
  -v /tmp/app.war:/opt/ibm/wlp/usr/servers/defaultServer/dropins/app.war
  websphere-liberty
```

- Updates from the host are now reflected in the container
- It really is a mount – not a copy of the file
- Fine for development, but to ensure we can recreate across environments we don’t want to be dependent on configuration of the host

Deploying an Application: Option 1

- Build a layer on top of the image containing the application

- Dockerfile

```
FROM websphere-liberty
ADD app.war /opt/ibm/wlp/usr/servers/defaultServer/dropins/
ENV LICENSE accept
```

- docker **build** -t app .
- docker **run** -d -p 80:9080 -p 443:9443 app

- Modifying the application requires rebuilding and redeploying just the application layer

Deploying an Application: Option 2

- Put the application in a separate data volume container

- Dockerfile

```
FROM websphere-liberty
ADD app.war /opt/ibm/wlp/usr/servers/defaultServer/dropins/
ENV LICENSE accept
```

- docker **build** -t appimage .
- docker **run** -v /opt/ibm/wlp/usr/servers/defaultServer/dropins --name app appimage true

- docker **run** -d -e LICENSE=accept -p 80:9080 --volumes-from app websphere-liberty

- Modifying the application requires rebuilding and redeploying just the application container AND it doesn’t need to be rebuilt if the application server container is updated

Data Volumes for Logging

- Update Liberty config to log to directory based on hostname
  - server.xml

    ```
    …<logging logDirectory="/logs/${env.HOSTNAME}" />…
    ```

  - Dockerfile

    ```
    FROM websphere-liberty
    ADD server.xml /opt/ibm/wlp/usr/servers/defaultServer/
    ENV LICENSE accept
    ```

  - docker build -t wlp-log .
- Create a data volume container for logs
  - docker run --name logs -v /logs wlp-log true
- Run one or more Liberty instances
  - docker run -d --volumes-from logs wlp-log
- Access consolidated log files
  - docker run --volumes-from logs -v `pwd`:/backup ubuntu sh -c 'tar –czf /backup/backup.tar.gz /logs'

Adding Liberty Features

- The websphere-liberty image on Docker Hub only contains the contents of the runtime install JAR
- Additional features can be added via featureManager and the online repository
- Example to create an image with the MongoDB feature
  - Dockerfile

```
FROM websphere-liberty
RUN featureManager install --acceptLicense mongodb-2.0
```

  - docker build -t websphere-liberty:mongodb .

Container Linking

- Linking provides a mechanism to make containers aware of one another
- For example, start a container running MongoDB:
    - docker run -d --name mongodb dockerfile/mongodb
- Then link it to another container with the alias ‘db’
    - docker run -d --link mongodb:db -p 80:9080 app
- /etc/hosts will be updated in the second container to resolve ‘db’ to the correct IP address for the first container
- Configuration in this container can therefore make use of this alias host name
    - server.xml

```
<mongo libraryRef="lib" hostNames="db" ports="27017"/>
```

Cross-host dependencies

- Container linking only works on a single host
- Two options for dependencies across hosts:
  1. Ambassador pattern
     - Link to a local container that acts as a port forwarder to a remote service
     - Enables changing the backend service by only restarting the ambassador
  2. Environment variables
     - Pass the details of the remote service as environment variables
     - For example:
       - docker run -e DB=192.0.2.0 -d app
       - server.xml

```
<mongo libraryRef="lib" hostNames="${env.DB}" ports="27017"/>
```

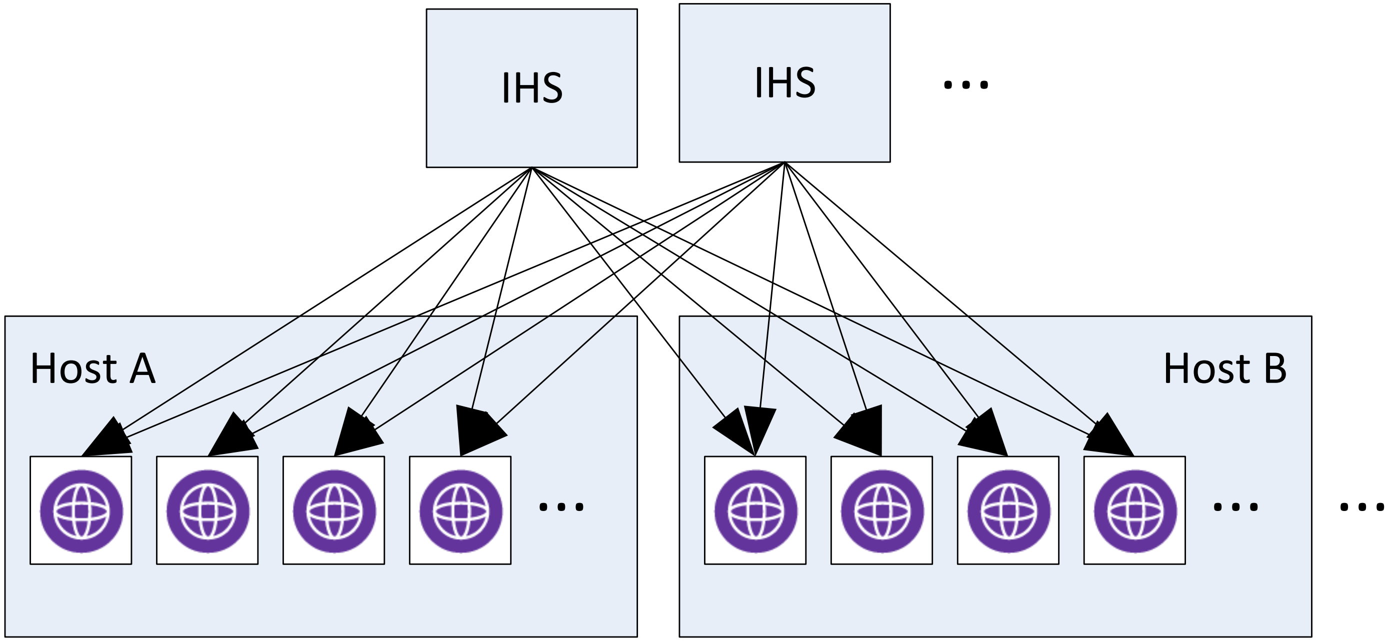
Obtaining a Production Licensed Image

- Obtain JRE and Liberty binaries from Passport Advantage/developerWorks and build your own Docker image
- Upgrade the Liberty image from Docker Hub using executable license JARs obtained from Passport Advantage
- Dockerfiles available at https://github.com/WASdev/ci.docker

Creating a Static Topology

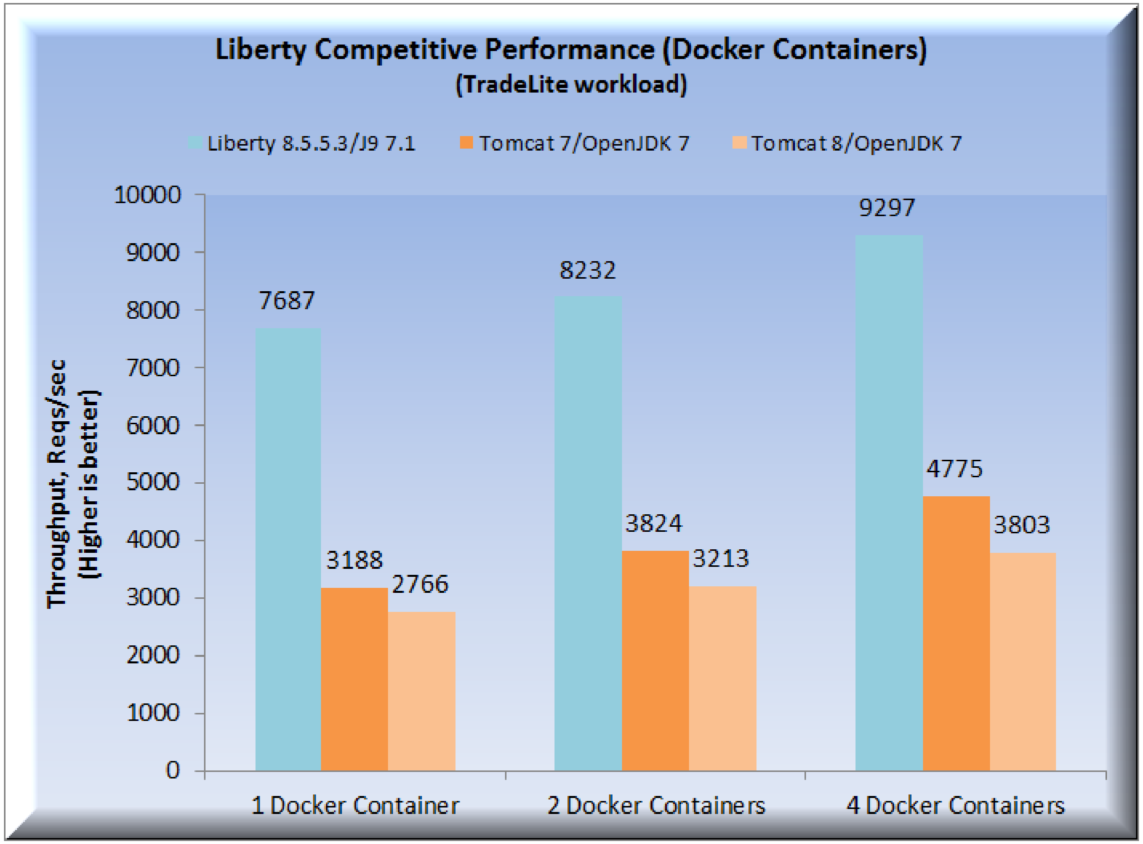
1Q

- Documentation and scripts to assist in the construction of a static topology with IHS and Liberty Profile under Docker



Performance – Liberty vs Tomcat

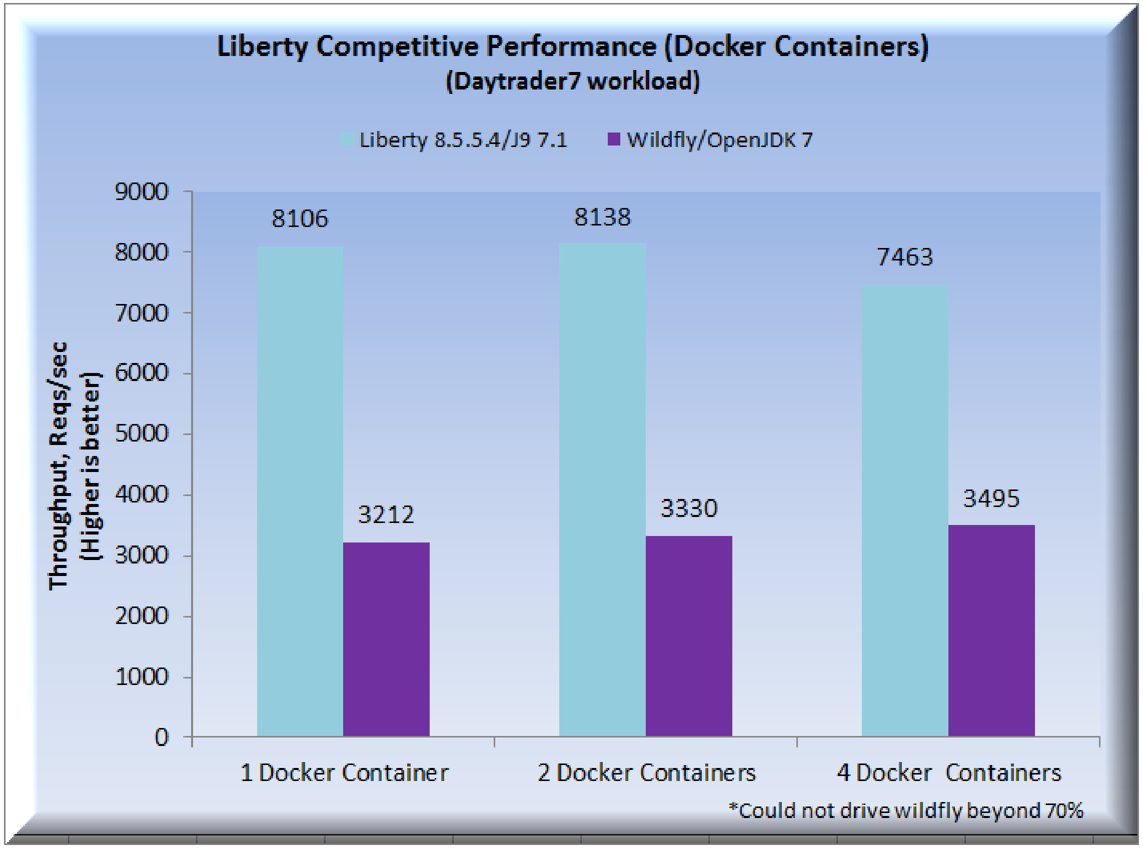
Liberty performs **2.4x** better than Tomcat 8 with a 4 Docker Container topology



Single host – 4 CPUs assigned across 1, 2 or 4 containers

Performance – Liberty vs Wildfly

Liberty outperforms Wildfly by greater than **x2** running Docker topologies



DayTrader 7 workload contains few features of Java EE 7 like Servlet 3.1, Web Sockets

Questions?

Summary

- What is the promise of containers?
- Overview of Docker
- Docker and IBM
- WebSphere Application Server and Docker

Thank You

Notices and Disclaimers

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and Disclaimers (con't)

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM, the IBM logo, ibm.com, Bluemix, Blueworks Live, CICS, Clearcase, DOORS®, Enterprise Document Management System™, Global Business Services ®, Global Technology Services ®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, SoDA, SPSS, StoredIQ, Tivoli®, Trusteer®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.