WebSphere Application Server V9 Update:
On Premise and In the Cloud

Ian Robinson: DE, WebSphere Foundation Chief Architect
@ian__robinson

Raj Suryavanshi: Offering Manager

rajsurya@us.ibm.com

IBM

Agenda

- **Digital transformation is happening: A quick overview**
- **WAS's Entry point approach to facilitate customer's Journey to Cloud**
- **Highlights of 2q2016 deliverables: Announcement: June 7, 2016, eGA: June 24, 2016**
- **WAS v9 – Highlights**
  - **Create, Connect, Optimize – brief overview**
  - **Deep dive**
- **Seamlessly upgrade to WebSphere Application Server V9**

Digital Transformation is happening….Clients are looking to:

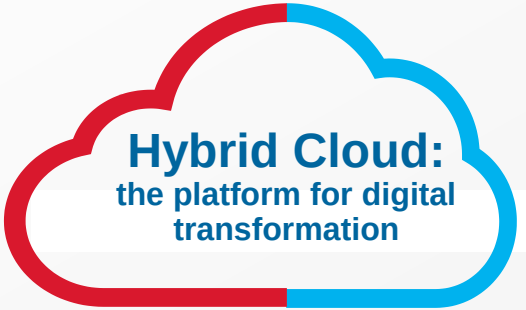
Deliver Better, Faster and Cheaper

Adopt a hybrid cloud strategy

DevTest in the cloud

Implement Microservices Architecture & leverage Docker Containers for portability

Integrate seamlessly across any platform



Continuous Delivery of Apps

Optimize the cost of their existing infrastructure

Adopt and Expand API usage

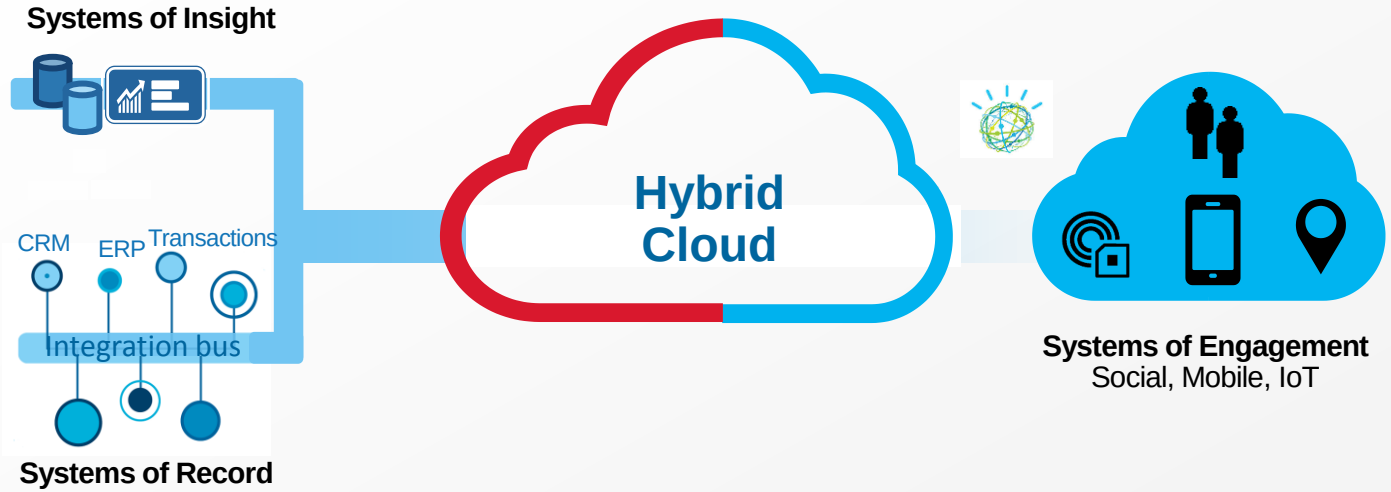
Add cognitive capabilities

Move their IT infrastructure to cloud

Enhance applications with new services

WebSphere supports the entire hybrid cloud landscape

- ✓ Integrate seamlessly across anything
- ✓ Innovation across enterprise IT and new ecosystems



WebSphere Application Server

WebSphere on Premises

docker

IBM Bluemix™

WebSphere Application Server

The cornerstone of your cloud strategy

30%+
TCO
WAS on Cloud Bluemix vs On-premises

122%
ROI
Liberty vs Open Source

$325K
annual infrastructure savings
by year 3 vs. Open Source

Industry Leading
Security
- Open ID Connect
- Secure Engineering Accreditation O-TTPS

30%
better performance
with Java 8

Intelligent
Management
45% less hardware
60% admin savings
45% less software
90% fewer outages

CREATE

Developer focused to speed delivery pipeline

- Lightweight composable runtime - perfect for microservices
- Full integration with any DevOps toolchain for continuous delivery
- Java EE7 market leadership and support for Open Source

CONNECT

Easy cloud connections for new & existing apps

- Create, expose and connect APIs
- Re-use existing apps and connect to “on or off” premises
- Deploy anywhere - on premises, in cloud or hybrid

OPTIMIZE

Smart management of the mission critical

- Leading edge cloud & mobile security
- Enterprise Management of Java & Node.js
- High availability: auto-scaling, dynamic routing, health management, diagnostics

WAS V9: 2Q16 Key Announcements

Create
new cloud apps

WAS V9 delivers robust & modern developer environment for speed & enterprise scalability

- Cloud-First Java app platform - Speed development with composable runtime and microservices architecture;
  - Full **Java EE7** certification for **both WAS Traditional and Liberty Profiles**;
  - Leverage **Spring and Spring Boot** frameworks within applications;
  - Quick start cloud native Java apps with **Liberty app accelerator** and **Game-On exemplar** – seamless deploy to IBM Bluemix
  - Leverage end-to-end **DevOps Toolchain** and **Garage Method** (best practices)
  - Portability of apps with Docker support; Deploy to IBM Container Service, Docker Data Center, and other container services
- Seamlessly manage Java and Node.js apps and APIs through common management interface

WAS V9: 2Q16 Key Announcements (cont.)

Optimize costs for app infra

Move apps anywhere, in any way, to increase speed and optimize costs

- Create a borderless environment with easy app portability regardless of architectural environment (Cloud container services, Docker, VMWare)
- Exploit WAS ND intelligent management for workload optimization and placement
- Move apps to the IBM public cloud with choice of pre-configured environ, flexible PayGo pricing models
- WAS Liberty app deployment to OpenShift and Pivotal Cloud Foundry running in IBM SoftLayer, Amazon AWS, Microsoft Azure, and IBM Bluemix
- WAS on Bluemix- Single Tenant : New option to deploy on Cloud
- Ease of use enhancements for caching to optimize user experience

WAS V9: 2Q16 Key Announcements (cont.)

Connect existing apps & data

Connect to cloud services to integrate new capabilities, improve time to mkt & lower costs

- Seamlessly connect on-premises apps to cloud services like Watson, Cloudant, dashDB. API Connect , Log Analytics (beta) - take advantage of latest technologies and extend the value of existing Java apps
- Optimize use of APIs for exposing and better monetizing traditional apps
- Leverage PayGo models and eliminate risk and complexity of managing these new services
- Leverage “API Connect Essentials” now included in WAS editions

Why Java EE7 on WebSphere for Microservices

Production ready, EE7 compliant app server for microservices strategy

- Large Developer base using Java Framework (including Spring based on Java EE)
- WebSphere designed with lightweight, composable runtime architecture
- WebSphere's architecture + Java EE7 = proven, enterprise development platform for Microservices
- Allows for easy integration with DevOps workflows and Docker container strategies
- Seamless connectivity to cloud based data and cognitive services
- WAS V9 brings traditional WAS deployments up to the same level of EE7 and Java 8 as WebSphere Liberty

Lightweight & composable

<64MB

Footprint

Ideal for microservices

Java EE™

docker

IBM Bluemix™

WebSphere®
Application Server

WebSphere Application Server Editions

1 PVU of Family Edition entitles:

1 PVU ND *or*
4 PVUs Base *or*
8 PVUs Liberty Core

OR mix & match

AND can redeploy new mix over time

REMEMBER: Devt desktops cost 0 PVU (*) for any family member

8x Liberty Core

- ✓ Web, mobile, OSGi apps (Web profile specification)
- ✓ High performance transactions
- ✓ Subset of Liberty features

4x WAS

- + Full enterprise applications for WAS traditional and Liberty
- + Extended programming models

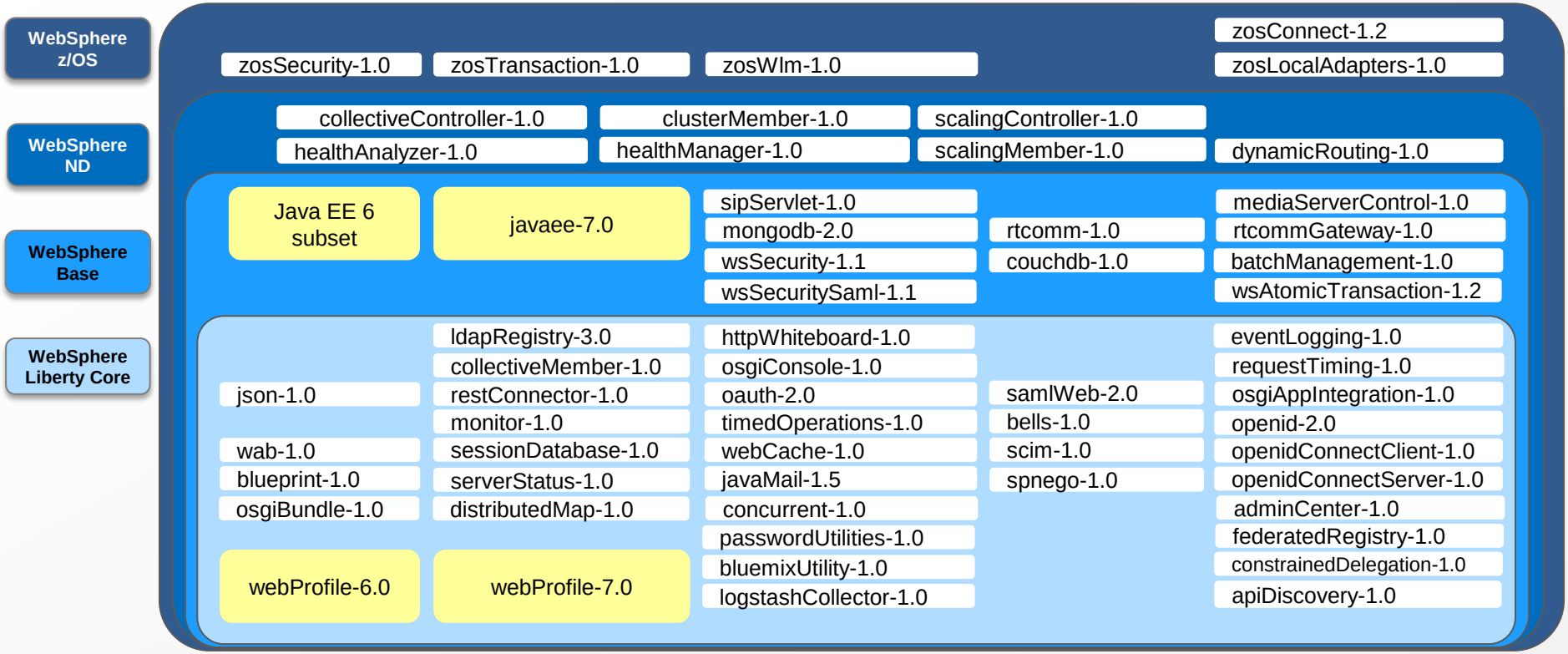
1x WAS ND

- + High availability
- + Intelligent management
- + High scalability

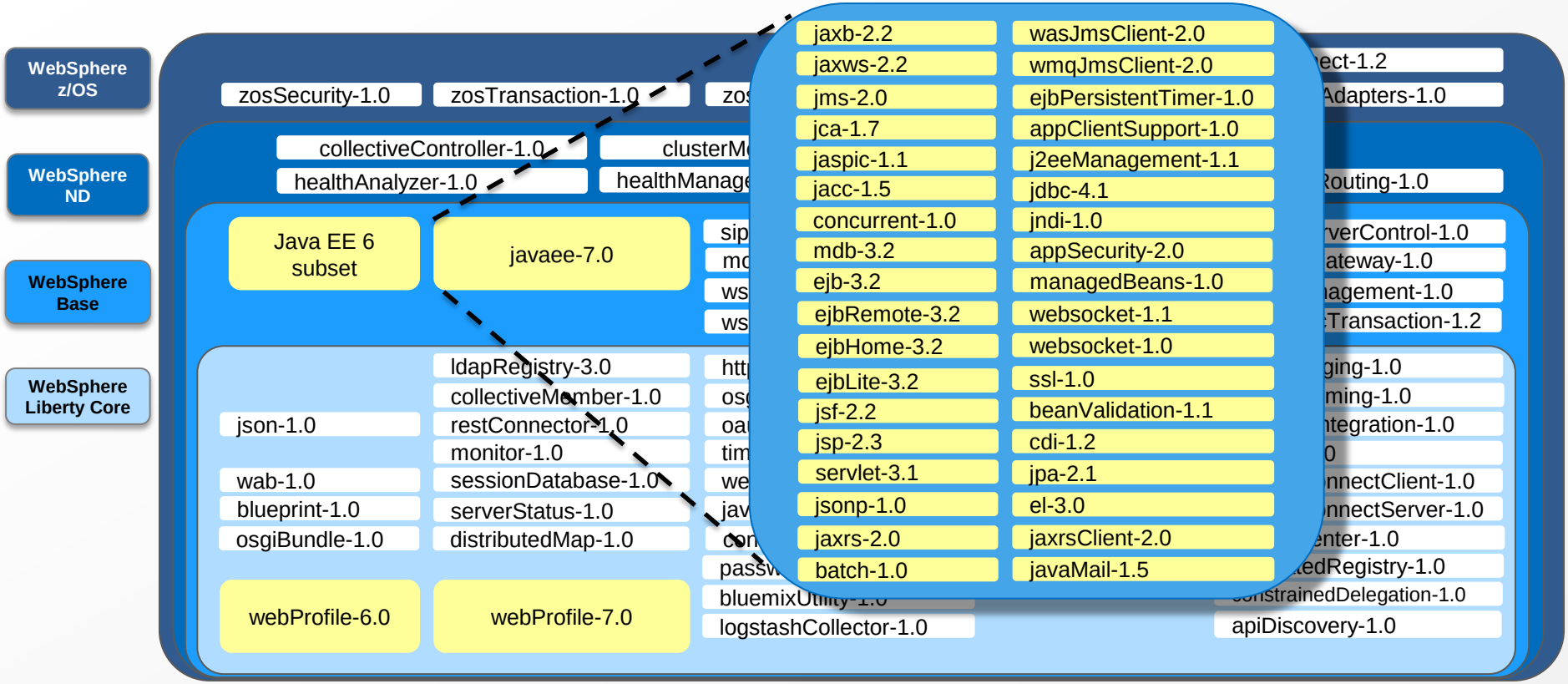
and more…

WAS Family Edition

Composable Java EE Features For Liberty



Composable Java EE Features For Liberty



Liberty Server Config: simple server ‘shaping’

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

    <!-- Enable features -->
    <featureManager>
        <feature>servlet-3.1</feature>
        <feature>jdbc-4.1</feature>
    </featureManager>

    <webApplication id="blogapp“
      location="blogapp.war" name="blogapp"/>

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

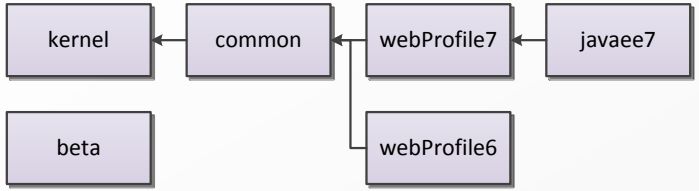
Features control which capabilities (bundles) are installed in the server

'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

Customized Docker containers for Java EE

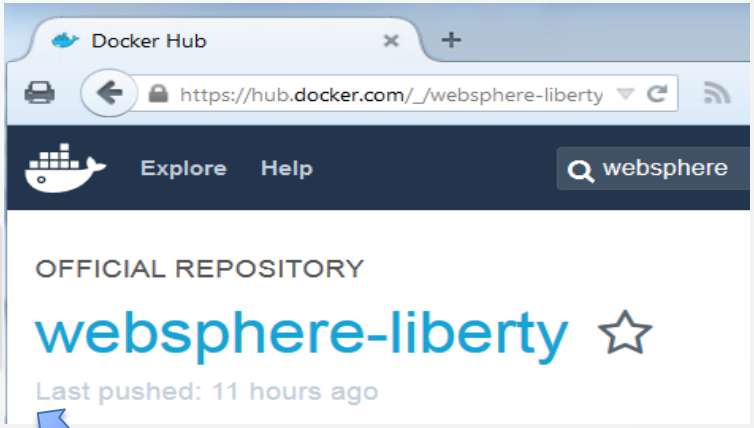
- WAS Liberty images on Docker Hub
  - WAS Liberty containers (currently V8.5.5.8):
    - Kernel, Java EE 6 Web Profile, Java EE 7 Web and Full Profile and latest Beta images
    - Docker files: https://github.com/WASdev/ci.docker



- Dockerfiles in on WASdev GitHub to:
  - Simple layer to upgrade to commercial license
  - Build your own customized image

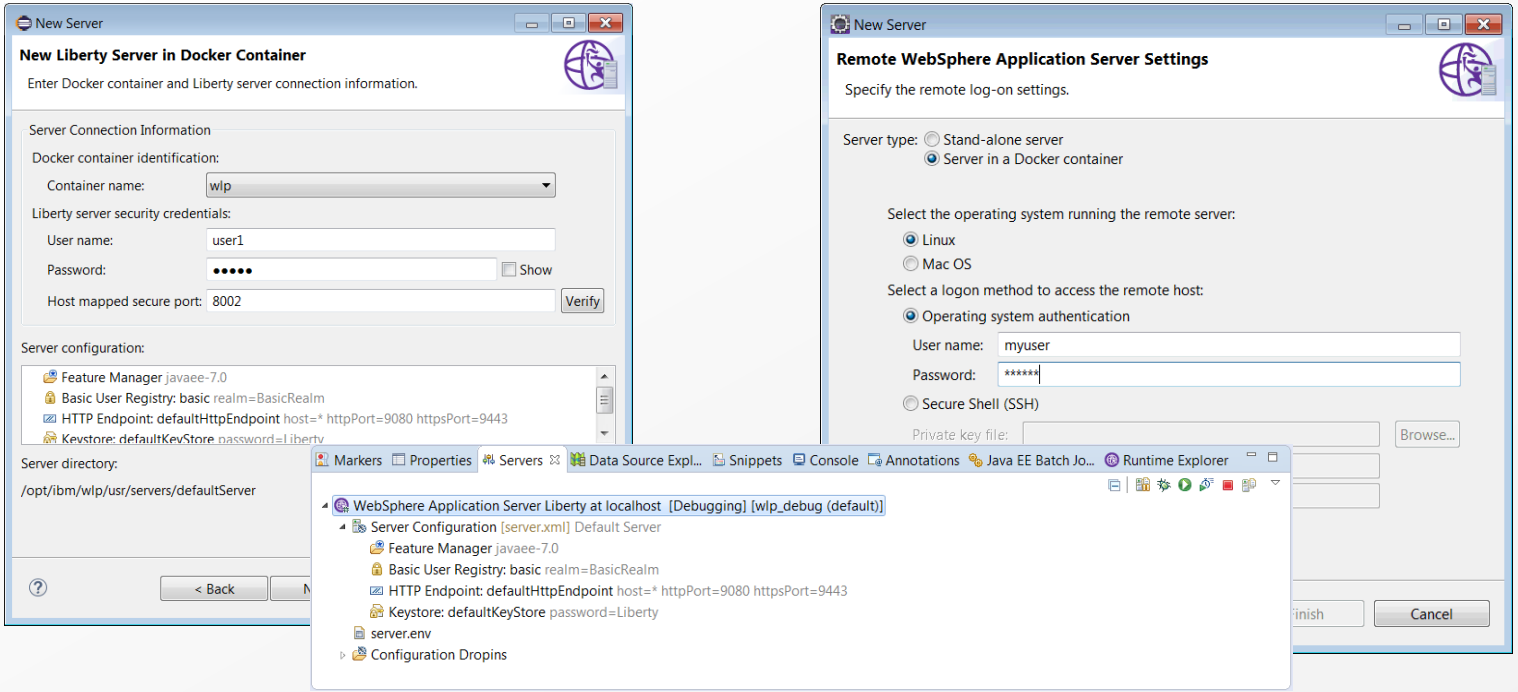
```
FROM websphere-liberty:kernel
COPY server.xml /opt/ibm/wlp/usr/servers/defaultServer/
RUN installUtility install defaultServer
```

hub.docker.com/_/websphere-liberty



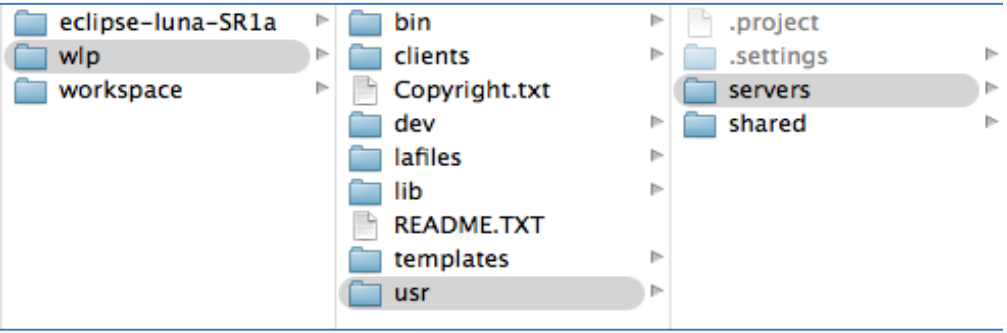
WDT Docker Support

- Easy setup an environment for developing and debugging applications with Liberty running in local or remote Docker container



Liberty: Simple to get started locally

1. Download desired archive from http://wasdev.net/repo (e.g. wlp-webProfile7.zip or wlp-javaee7.zip)
2. unzip <archive>
3. bin/server create myServer
4. cp myapp.war usr/servers/myServer/dropins
5. bin/server run myServer
6. http://localhost:9080/myapp



Liberty: Simple to run in any cloud

1. Clean separation of product install and user-defined configuration
2. Easy self-contained deployment of application + server/resource configuration
3. To PaaS…
4. Into a Docker container (CaaS)…
5. To IaaS (hosted or on prem)…

docker build –t myapp
wlp/usr/servers/myServer

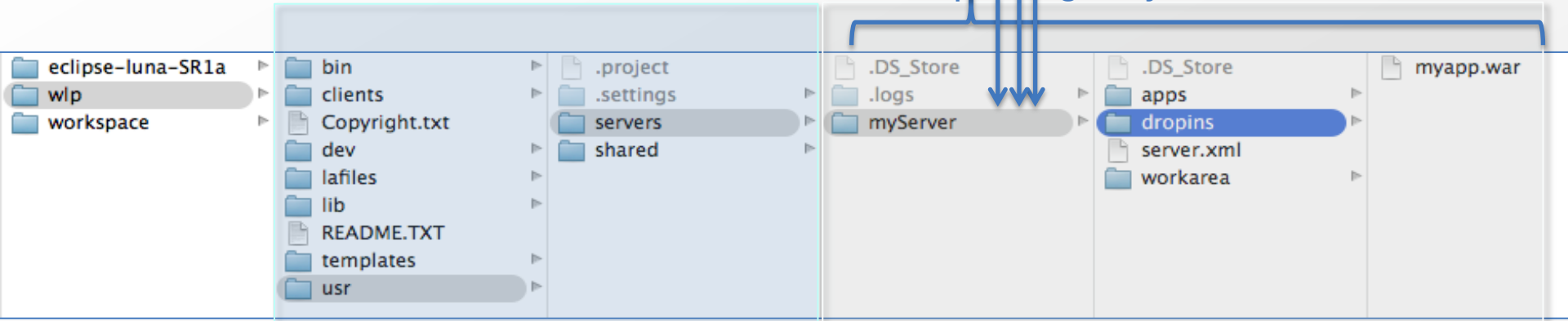
Self-contained under myServer

cf push myapp -p

dockerfile

```
FROM websphere-liberty
COPY myServer /opt/ibm/wlp/usr/servers/defaultServer
```

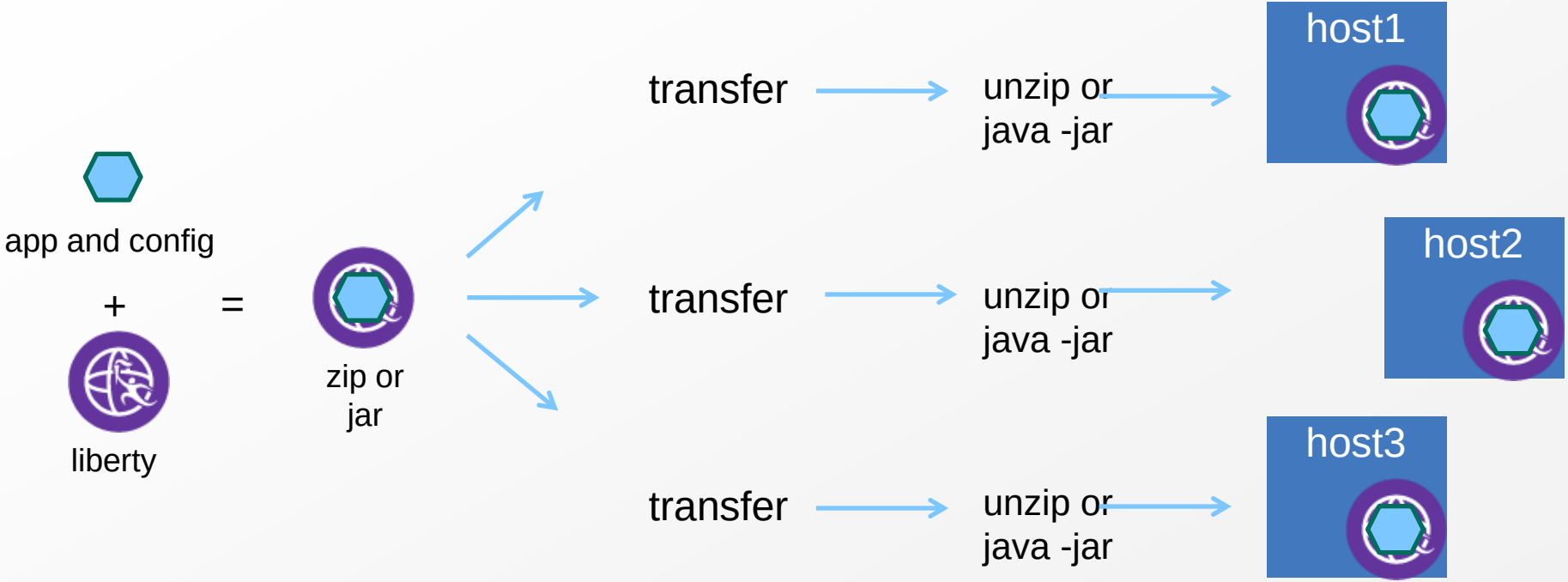
server package myServer –-include=usr



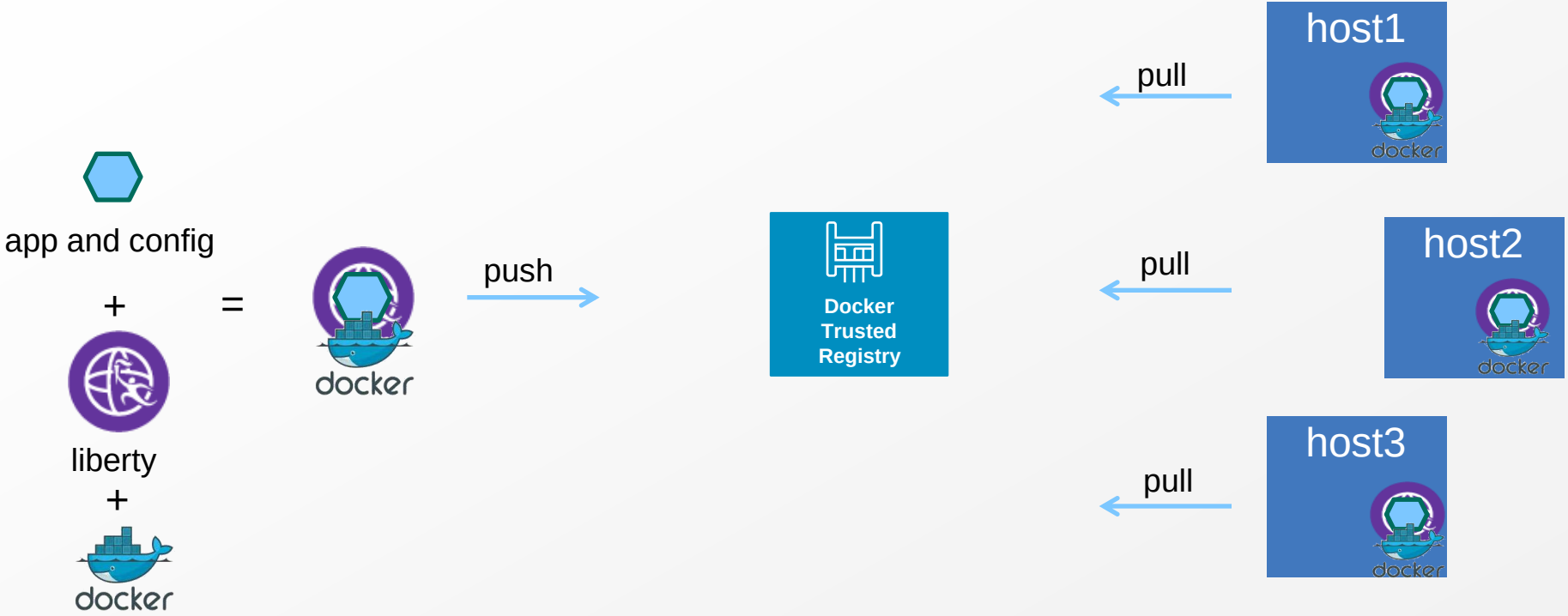
WLP Runtime Install

User Configuration and Resources

Liberty Server Package Distribution (Unmanaged)



Liberty Docker Image Distribution (Unmanaged)

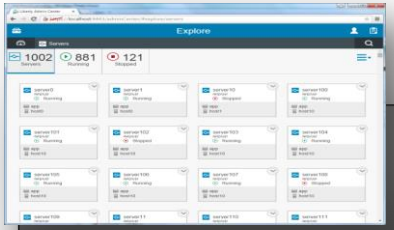


Intelligent Management with WAS ND

Collectives

Administratively defined policy

Operations target individual servers or clusters



Admin Script

HTTP/S

IHS /DP

Load balancing and session affinity/failover

collective

get dynamic topology

controllers

liberty

dynamic start/stop

liberty liberty liberty

liberty liberty liberty

liberty liberty liberty

cluster members

Dynamic Routing
Collective Controller provides routing updates to Plugin

Auto Scaling
Controller starts and provisions cluster members

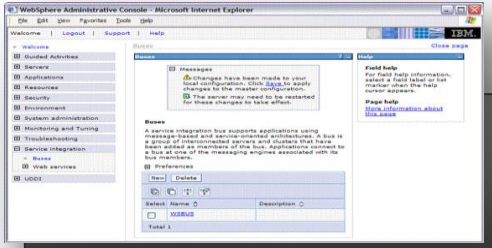
Health Management
Takes actions if policies are breached

Intelligent Management with WAS ND

Cells

Administratively defined policy

Operations target individual servers or clusters



Admin Script

HTTP/S

IHS /DP

Load balancing and session affinity/failover

get dynamic topology

DMGR

dynamic start/stop

cell

WAS trad

Node agent

cluster members

Dynamic Routing
Collective Controller provides routing updates to Plugin

Auto Scaling
Controller starts and provisions cluster members

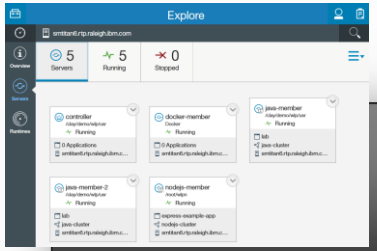
Health Management
Takes actions if policies are breached

ND Intelligent Management: Also Supports Node.js and Docker

Collectives

Administratively defined policy

Added deploy rules and REST API on controller for Liberty Docker and Node.js



Admin Script

HTTP/S

IHS /DP

Load balancing and session affinity/failover

get dynamic topology

controllers

liberty

Docker Trusted Registry

dynamic start/stop

collective

liberty liberty liberty

Node.js Node.js Node.js

liberty liberty liberty

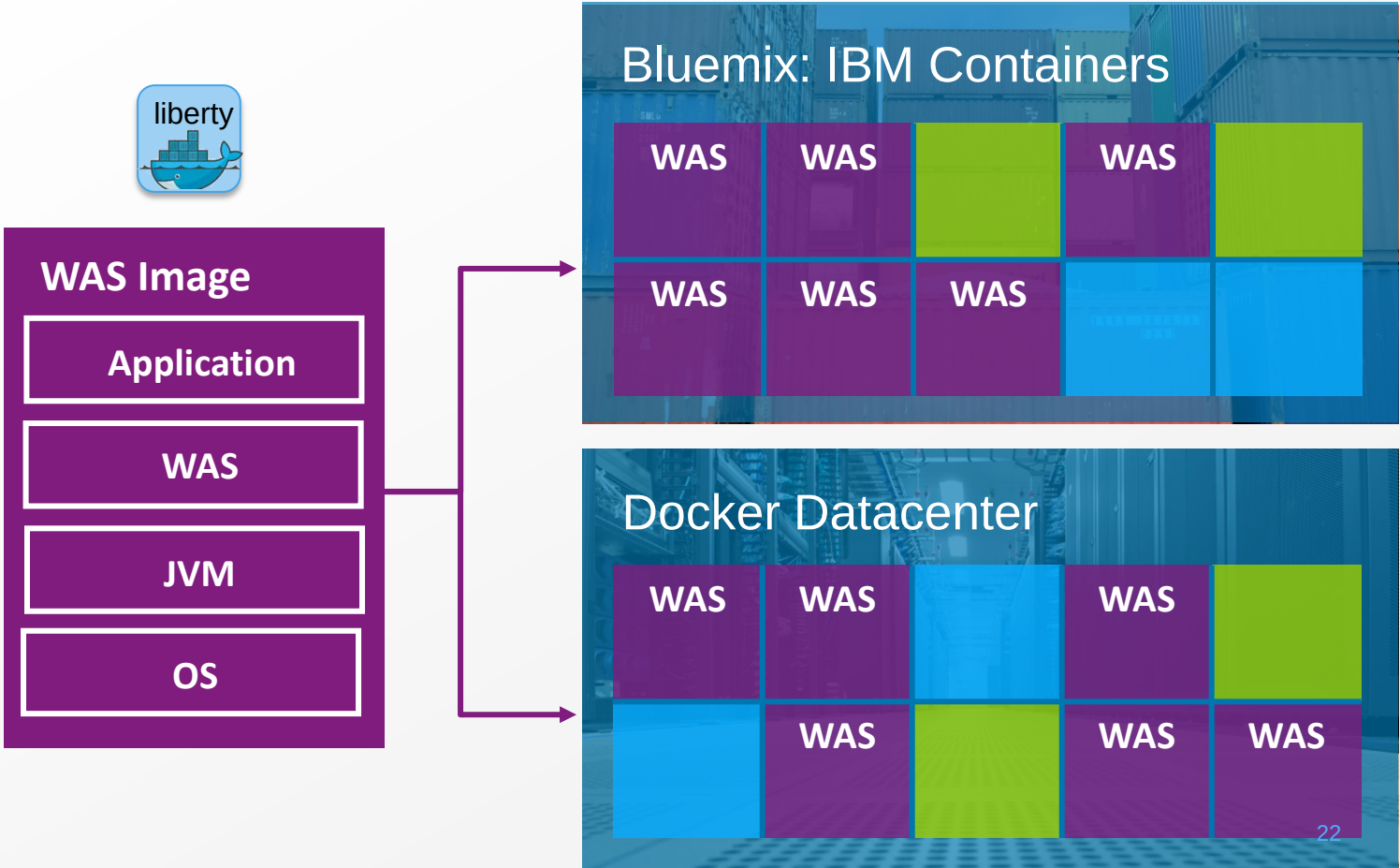
cluster members

Dynamic Routing
Collective Controller provides routing updates to Plugin

Auto Scaling
Controller starts and provisions cluster members (Node.js coming soon)

Health Management
Takes actions if policies are breached

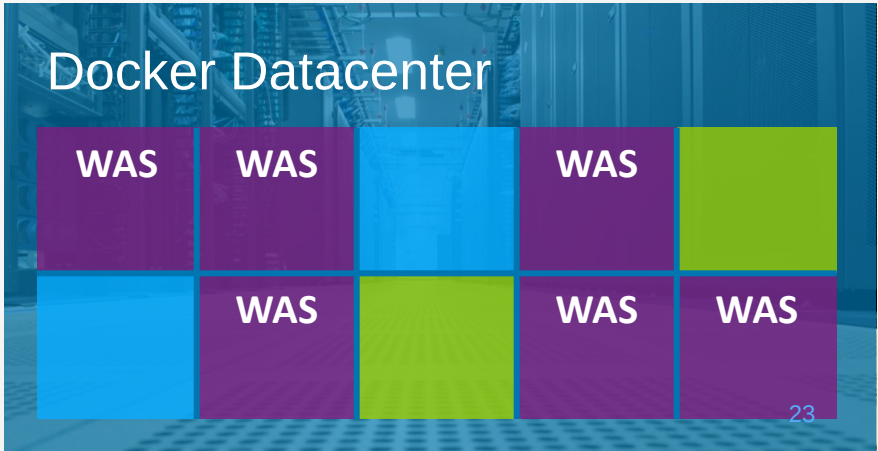
Integration with Managed Docker Deployments



Managed Docker Container Deployments

Heterogeneous Container lifecycle and operational management

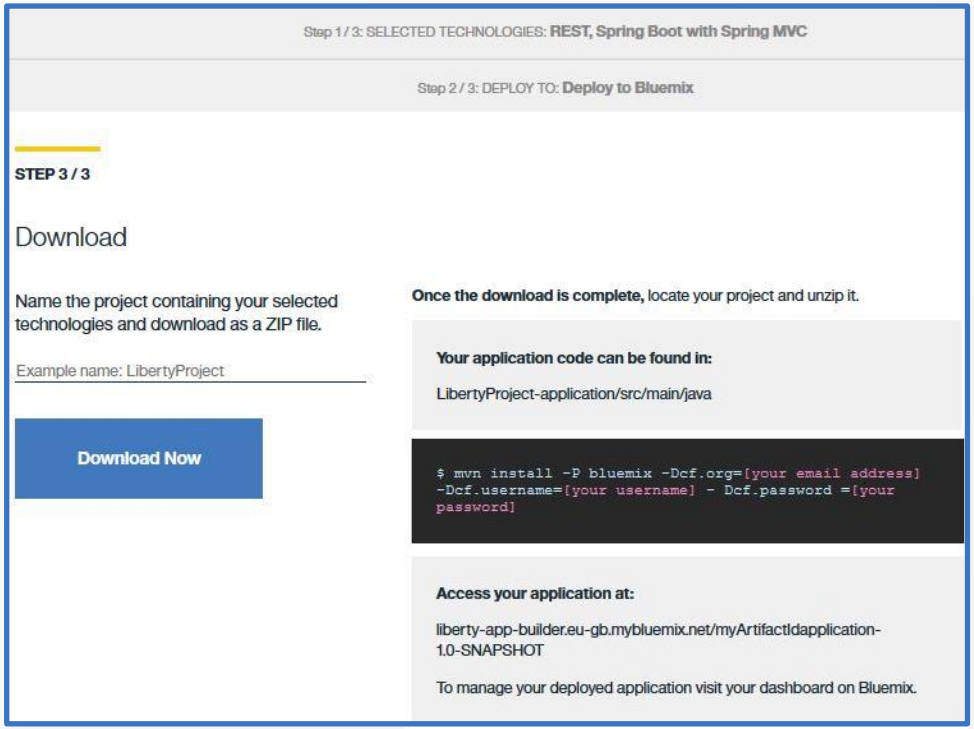
- IBM Containers provide a managed runtime environment for Docker in Bluemix
- On-prem solutions also include Docker Datacenter, from Docker Inc, **which IBM also supports as a reseller.**
- Both provide secure Docker registry, the ability to deploy and manage multiple-container **Compose** topologies and load-balancing across container clusters using Docker **Swarm**
- The choice is transparent to the WebSphere application container
- Standalone containers or Collectives with built-in WAS intelligent management.



Liberty App Accelerator for Microservice creation

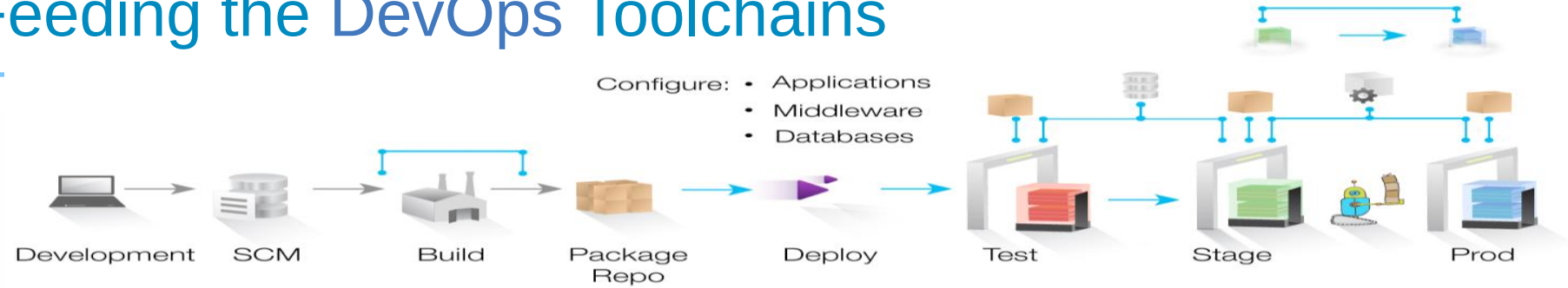
http://wasdev.net/accelerate

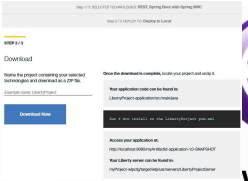
- Get a Java-based microservice that uses Liberty built and deployed in < 10min
- Provides configuration for your application
- Provides code snippets for your application
- Possible deployment options:
  - Local
  - Bluemix



Also See: http://wasdev.net/microservices

Feeding the DevOps Toolchains





Liberty App Accelerator

WDT

Git

Jenkins

Docker Trusted Registry

WAS Liberty

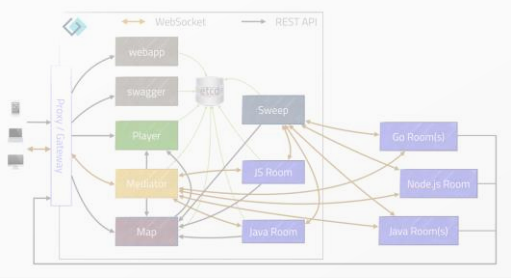
maven

Configuration Management

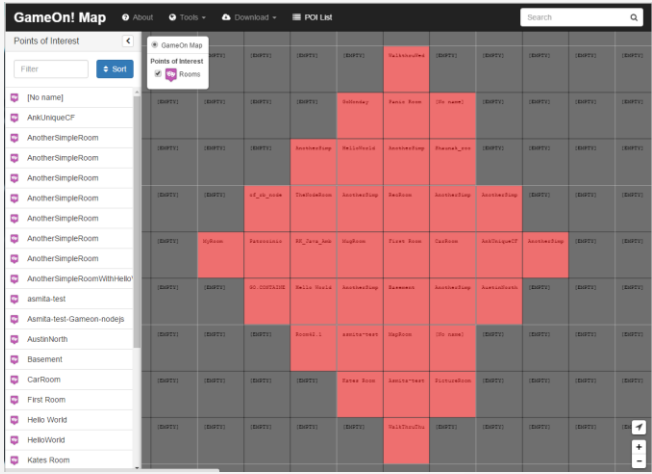
urban{code}

A microservices adventure

- Exemplar microservices application
  - Ployglot
    - Core services built using Liberty and Java EE7
    - Deployed using Cloud Foundry, Container, and Event-driven runtimes

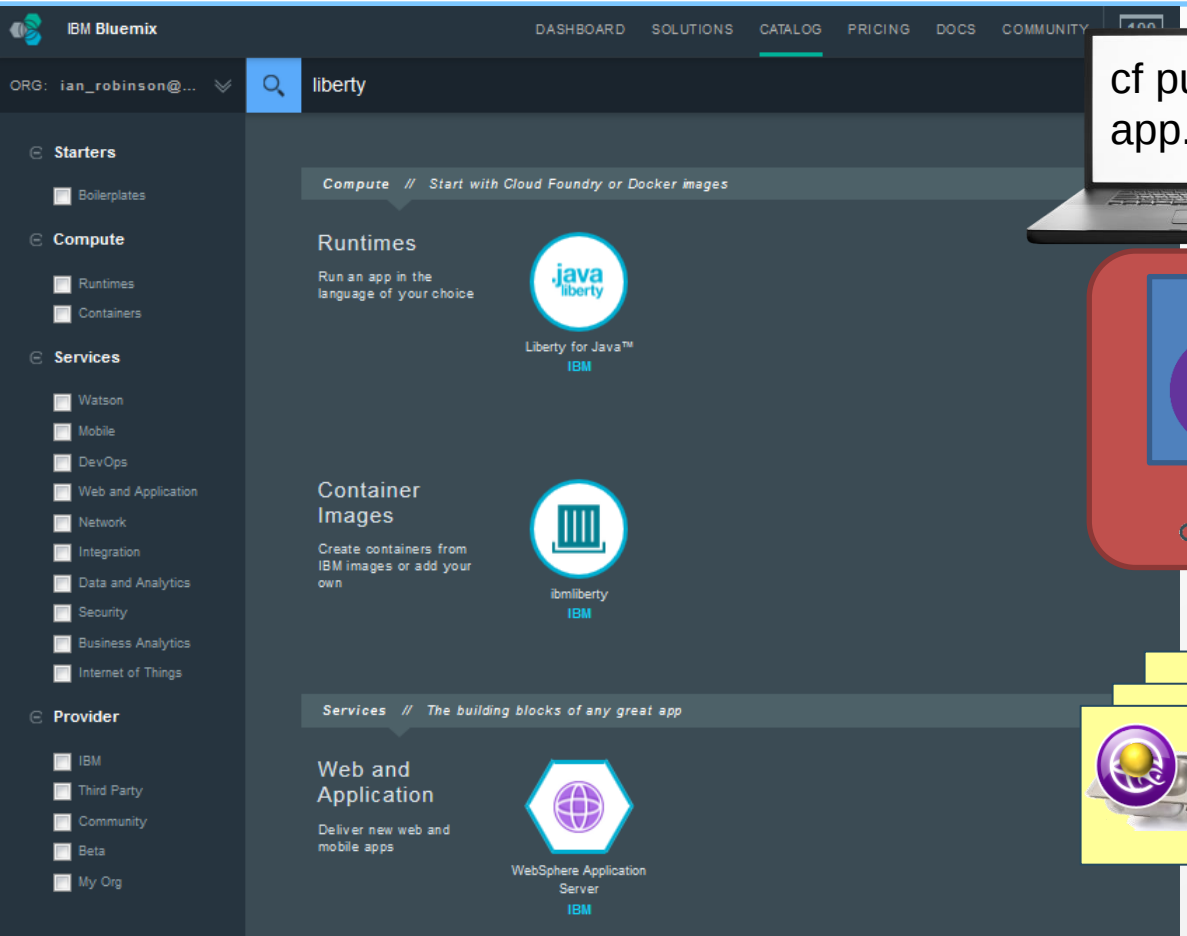


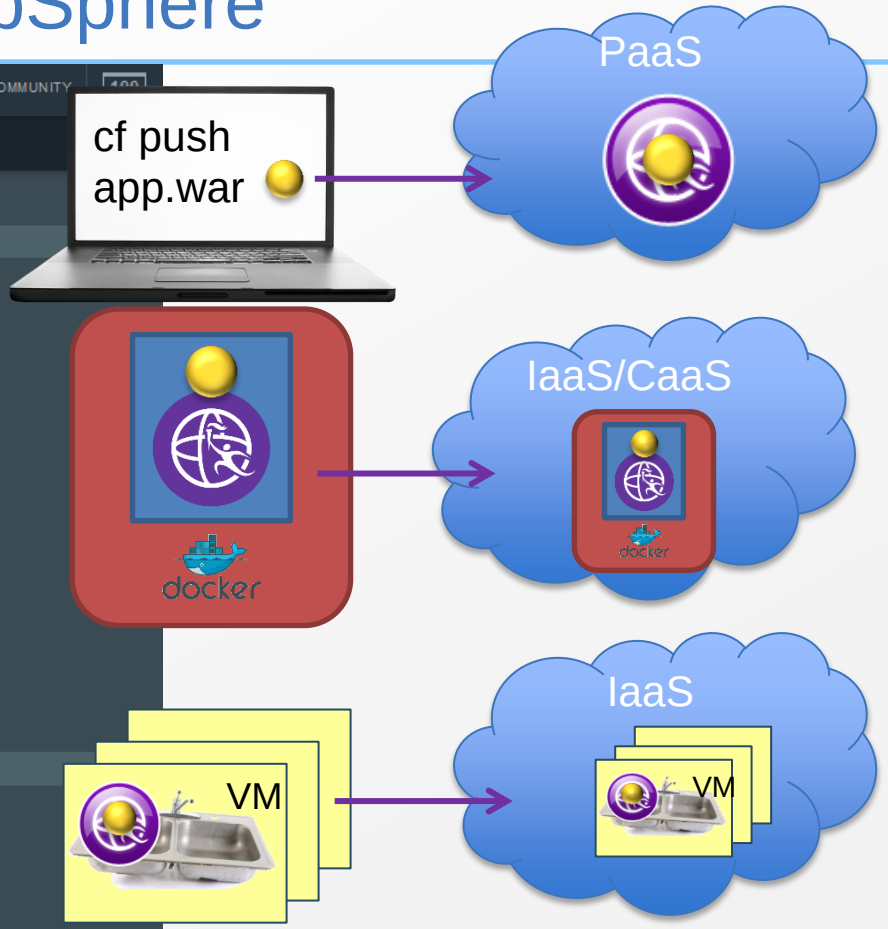
  - Parity between development and test
    - Lightweight runtimes and Docker containers support production-like local development
  - Ease of integration and extension
    - Use Java EE7 and Liberty to create extensions for the game
    - JAX-RS for Swagger defined REST APIs
    - WebSockets and Rx-Java for asynchronous messaging.
  - Open source and extensible
    - An “adventure game” where each room is a microservice.
    - Anyone can add a new microservice room to the game instance on Bluemix
    - Rooms scored on availability and adherence to the Room API
    - Rooms with the highest score move towards the middle of the map.



- https://game-on.org/

IBM Bluemix Cloud and WebSphere





WAS as a Service

WebSphere Application Server

IBM

PUBLISH DATE
12/18/2015

TYPE
Service

VIEW DOCS

Pick a plan — Monthly prices shown are for country or region: United Kingdom

Plan	Features	
WAS Liberty Core Plan	WebSphere Application Server Liberty Core. Red Hat Linux Guest.	£0.25 GBP/Hour
✓ WAS Base Plan	WebSphere Application Server Classic Base. Red Hat Linux Guest.	£0.32 GBP/Hour

```
ejbdeploy.sh            pluginCfgMerge.sh       wrd.sh
enableProvisioning.sh   pluginMerge.sh          wsadmin.sh
endptEnabler.sh         pmt.sh                  ws_ant.sh
eventbucket.sh          postinstall.sh          wsappid.sh
eventcatalog.sh         PropFilePasswordEncoder.sh  WSBatchPackager.sh
eventemit.sh            queryCertificate.sh     wsdb2gen.sh
eventpurgepool.sh       registerwstp.sh         wsdbgen.sh
eventpurge.sh           removeNode.sh           wsdeploy.sh
eventquery.sh           renameNode.sh           WSDL2Java.sh
eventUpgradeDB2.sh      requestCertificate.sh   wsenhancer.sh
eventUpgradeDB2Z05.sh   ResponseFilePasswordEncoder.sh  wsgen.sh
eventUpgradeOracle.sh   restoreConfig.sh        WSGrid.sh
executeXPath.sh         restoreJobManager.sh    wsimport.sh
executeXQuery.sh        retrieveSigners.sh      wsjpaversion.sh
executeXSLT.sh          revokeCertificate.sh    wsmapping.sh
Extracter.sh            runConfigActions.sh     wsreversemapping.sh
findEJBTimers.sh        schemagen.sh            wsschema.sh
genHisteryReport.sh     [illegible]             wsve_encodePassword.sh
GenPluginCfg.sh         serverStatus.sh         xdaSetupCmdLine.sh
genVersionReport.sh     setupCmdLine.sh         xd_hadmgrAdd.sh
hadmgrAdd.sh            showlog.sh              xd_hadmgrRemove.sh
hadmgrRemove.sh         sibDBUpgrade.sh         xjc.sh
-bash-4.1$ pwd
/opt/IBM/WebSphere/Profiles/DefaultAppSrv01/bin
-bash-4.1$ ./serverStatus.sh -all
ADMU0116I: Tool information is being logged in file
           /opt/IBM/WebSphere/Profiles/DefaultAppSrv01/logs/serverStatus.log
ADMU0128I: Starting tool with the DefaultAppSrv01 profile
ADMU0503I: Retrieving server status for all servers
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: server1
ADMU0508I: The Application Server "server1" is STARTED
-bash-4.1$
```

Linux

9 GBP/Hour

WebSphere Application Server — Learn

Plan Name : WAS Base Plan

Creation Time: Feb 5, 2016 4:20:05 PM (UTC)
Lease Expires: Feb 19, 2016 4:20:05 PM (UTC)
Lease Renewal Available: Feb 12, 2016 4:20:05 PM (UTC)

Download VPN Configuration

Admin Console: http://159.122.233.228:9060/ibm/console

Admin Username: wsadmin
Admin Password: Show Password

Application Hosts/Nodes

Classic Profile WebSphere

OS Distribution: RHEL 6.6 X64 Show Terms

Specification:
1 Virtual CPU
2048 MB Memory
12.5 GB Hard Disk Space

Creation Information:
iFix Info:
8550-WS-WAS-IFPI52103_85500020151112_0822

Guest Info:
tWAS Base (RHEL 6.6, WebSphere 8557 and JDK 7.0) Nov

Show Password

Private SSH Key: Show Key Save Key

Key Store Password: Show Password

WebSphere software

WebSphere Integrated Solutions Console

User ID:
Password:
Log in

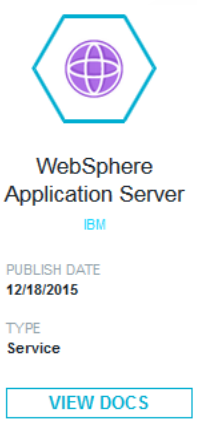
IBM

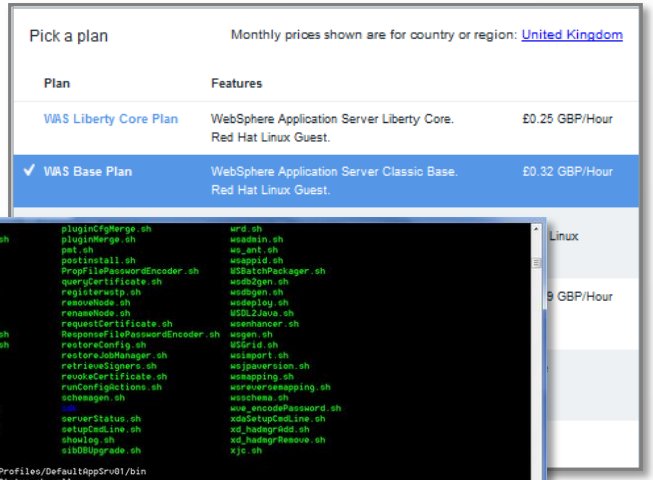
Licensed Materials - Property of IBM (c) Copyright IBM Corp. 1997, 2011 All Rights Reserved. IBM, the IBM logo, ibm.com and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information.

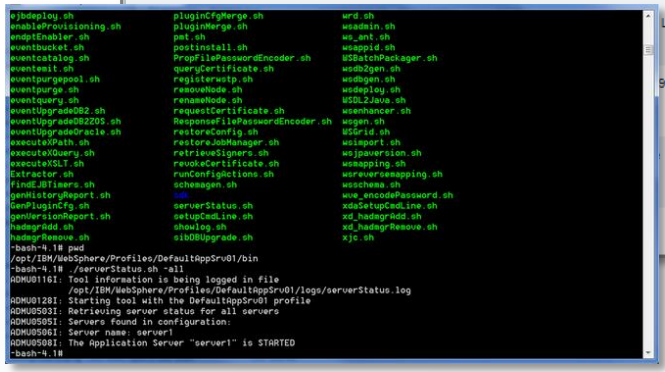
```
 S - 2GB RAM,  1vCP,  12.5 GB disk - WAS ND -  $1.14/hr
 M - 4GB RAM,  2vCP,  25   GB disk - WAS ND -  $2.28/hr
 L - 8GB RAM,  4vCP,  50   GB disk - WAS ND -  $4.56/hr
XL -16GB RAM,  8vCP, 100   GB disk - WAS ND -  $9.12/hr
XXL -32GB RAM, 16vCP, 200  GB disk - WAS ND - $18.24/hr
```

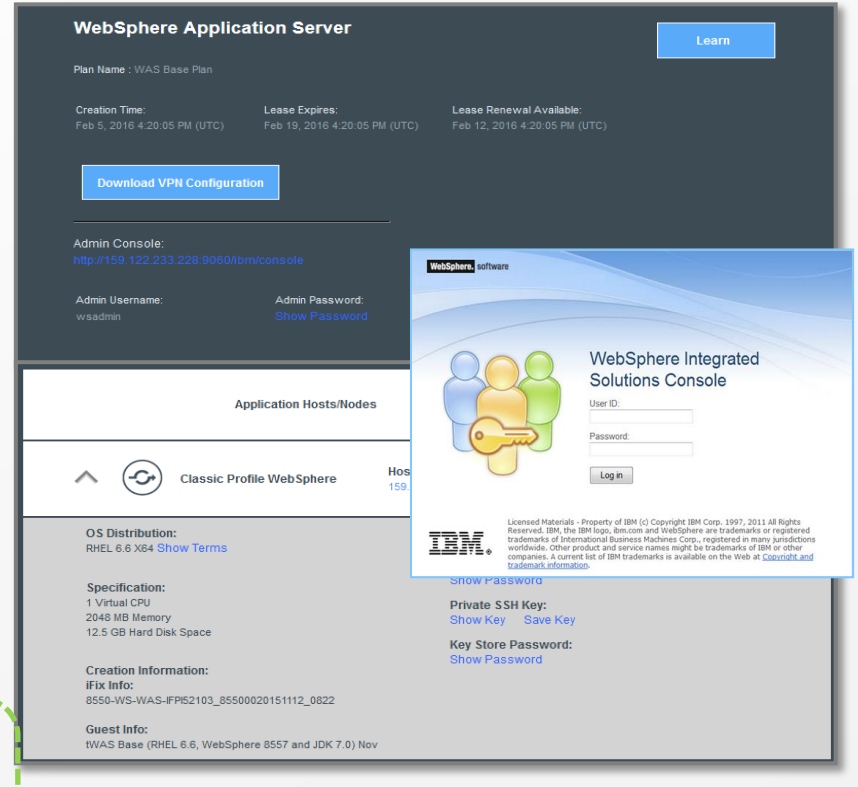
Variable VM sizes

WAS as a Service – Single Tenant Option









Deploy WAS for Bluemix VM on *private bare-metal hardware* in which the physical cores, memory, and storage is isolated to one customer

Intelligent workload placement, VPN, firewall, DirectLink high speed connection up to 10Gps, & Softlayer VLAN spanning to expand to additional customer solutions

Deployment flexibility on additional cloud environments

Elaine
Systems Administrator

"I want to run Liberty in a non-IBM PaaS on-premise, in Amazon and Azure. I want all the components from IBM to be fully supported and I want to use subcapacity pricing."

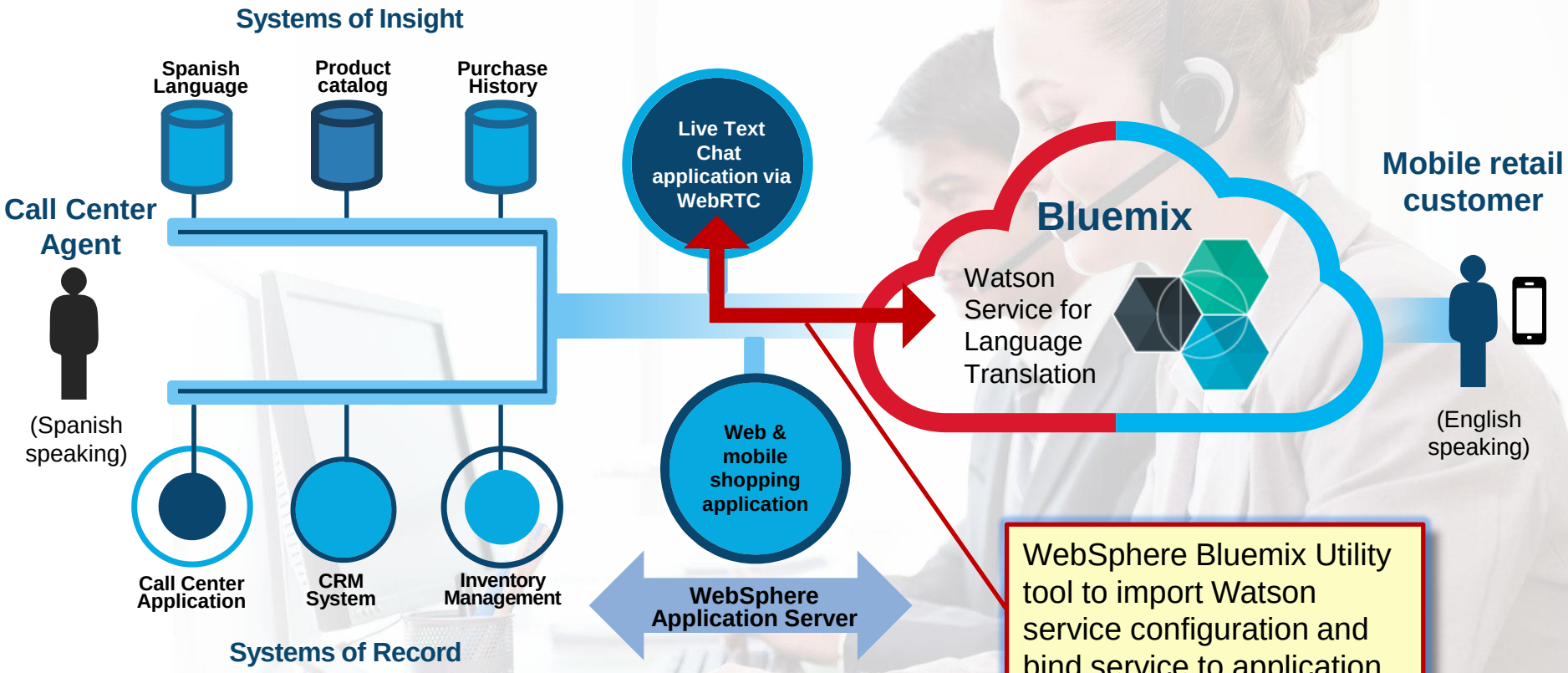
Solution Highlights:

1. A user can deploy Liberty apps in a Docker container to AWS or Azure.
2. A user can deploy Liberty apps to OpenShift and Pivotal Cloud Foundry running locally and can apply sub-capacity charging through suitable configuration.
3. A user can deploy Liberty apps to OpenShift and Pivotal Cloud Foundry running in Softlayer, AWS or Azure, and can apply sub-capacity charging through suitable configuration.

Bringing efficiencies of cloud-services to on-premises deployments

Enhance Applications with Bluemix Cloud Services

Retail example: WAS integration with Bluemix Watson Services for Cognitive Apps

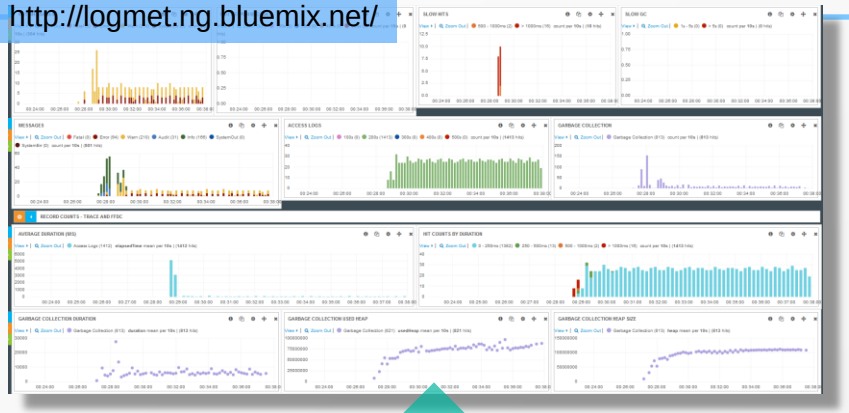


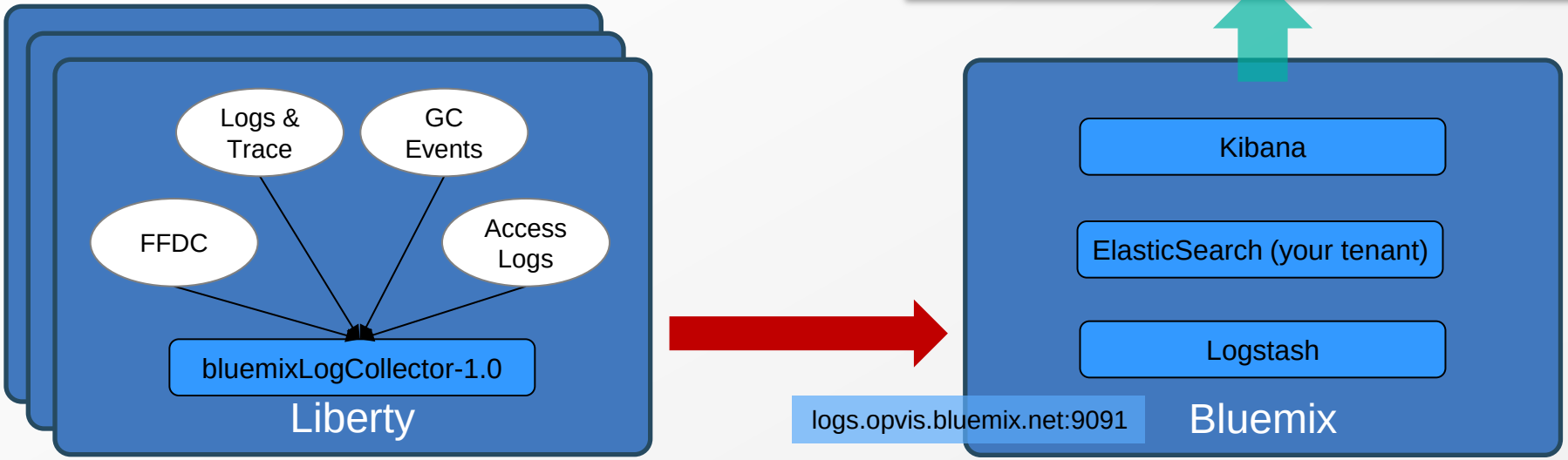
Log Analytics with Bluemix

BETA

- **Log analytics for WAS Liberty hosted for you in Bluemix**
- View consolidated data from servers across your hybrid environment in a single dashboard
- Based on popular ELK technology
- Start monitoring your Liberty servers in minutes
- Index up to 1GB of data per day for free, with 7 days of data retention

http://logmet.ng.bluemix.net/

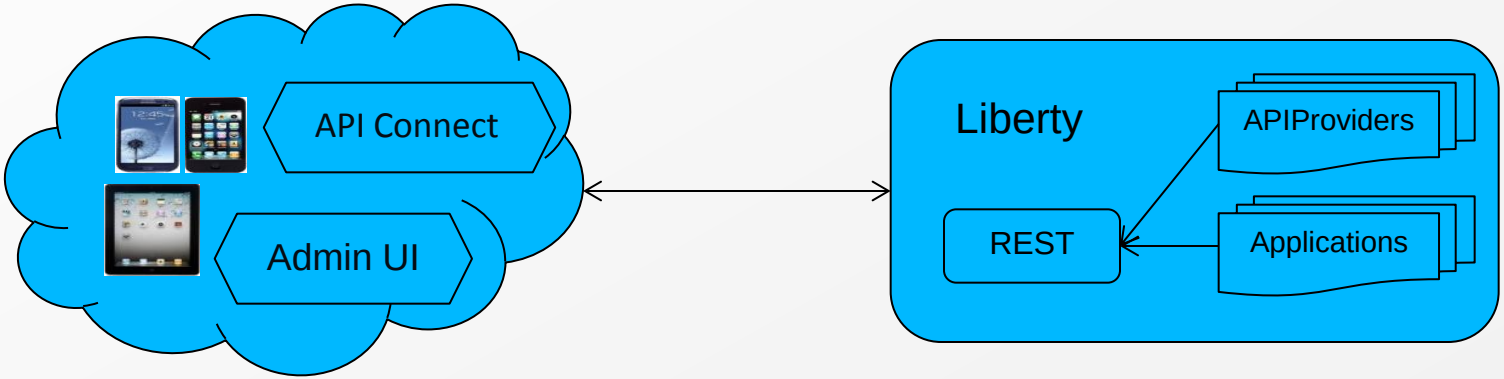




On-ramp to API Economy

API Discovery with Swagger Doc

- New apiDiscovery-1.0 feature makes API from all applications discoverable via a single RESTful endpoint: /ibm/api/docs
  - APIs described using Swagger 2.0 / Open API
  - Query parameter allows filtering based on context root
  - Supports both JSON and YAML (input and output)
  - Specialized extension, /ibm/api/docs/apiconnect, for native integration with API Connect



API Discovery Scenarios

- Liberty supports any path toward API discovery:

Top-Down (Documentation before Code)

- Embed an existing Swagger doc inside an application
- Swagger can be created manually or through an editor (e.g. API Connect, or Swagger's online editor)

Bottom-Up (Code before Documentation)

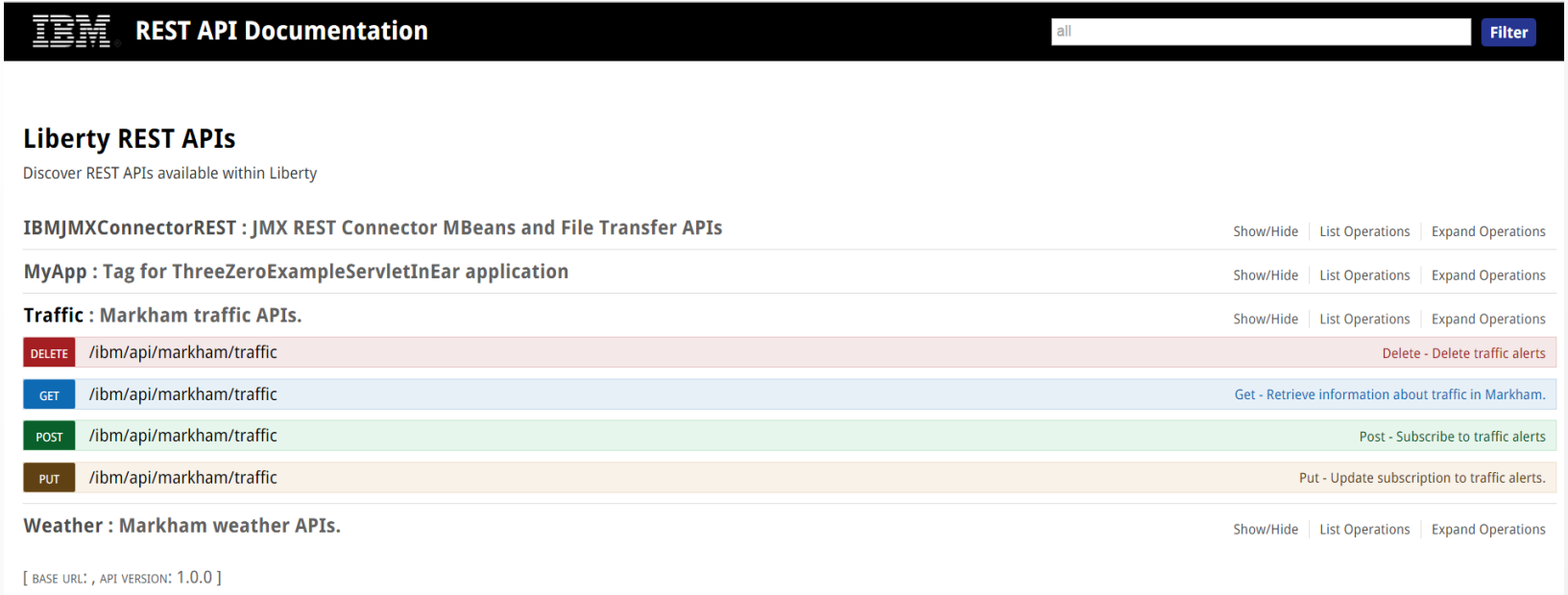
- Swagger doc is generated by the runtime based on annotations in the application (JAX-RS, Swagger, or both)

Mixed

- Existing Swagger doc merged with generated Swagger from annotations
- E.g. Apps that have some JAX-RS and some non-JAX-RS services

API Discovery User Interface

- Based on the Open Source Swagger UI
- Available at /ibm/api/explorer



Ready for WebSphere Application Server V9?

Upgrading to WebSphere v9

Migrate today to take advantage of v9 capabilities

- Intelligent management
- API Lifecycle Management
- Integration with cloud services
- Improved security & availability

Helping the move to traditional WebSphere V9:

- Planning: WebSphere Migration Discovery Tool
- App and Config Migration: WebSphere Migration Toolkit
  - Reduces time with configuration migrations & eliminates errors
- Config modelling and transformation: UrbanCode Deploy
  - Parameterize/modify and version-control WAS config

Hands-on Support

- Migration Team engagement
- No charge 1 or 2 Day Migration Workshop
- Proof of Concept

Version to Version Upgrade Waiver

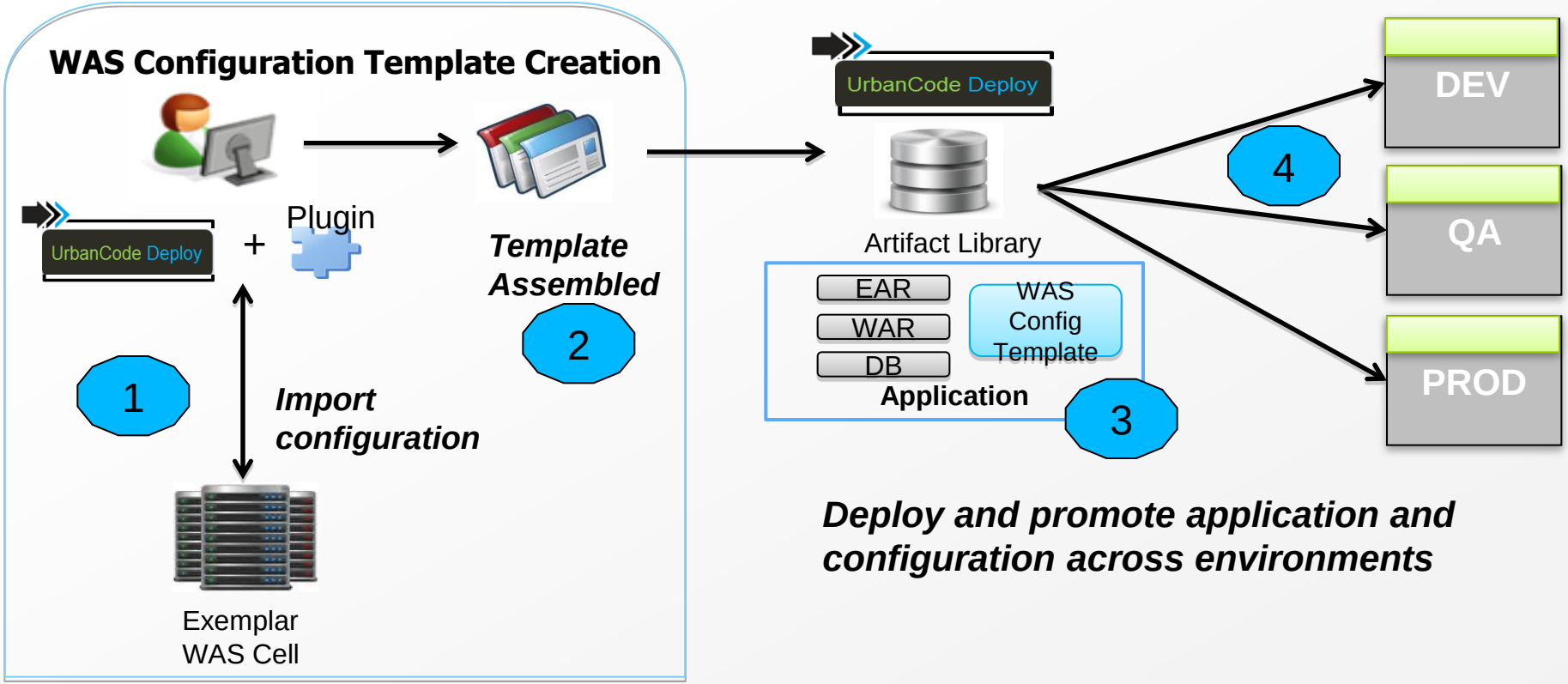
- No charge dual entitlement during the upgrade

50%

Less time to upgrade

Cut your typical upgrade time in half with the **WebSphere Application Server Migration Toolkit**

IBM UrbanCode Deploy – WAS Configure Plugin



Deploy and promote application and configuration across environments

Eliminate Future Migration Costs

New Liberty features, and config, augment rather than replace old ones

Zero migration for unchanged apps on WAS Liberty, regardless of Java EE version

unzip wlp-javaee7-8.5.5.9.zip

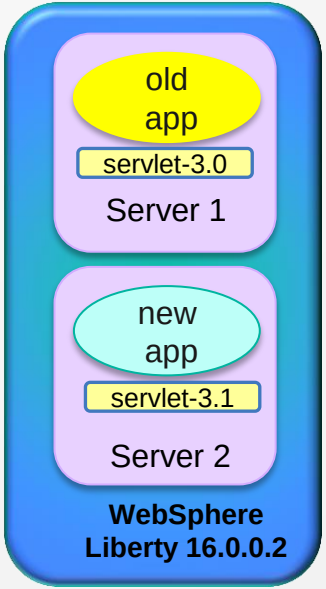
8.5.5.9

~~9.0.0.0~~
16.0.0.2

unzip wlp-javaee7-8.next.zip

WLP_USER_DIR

Your configuration, applications, resources

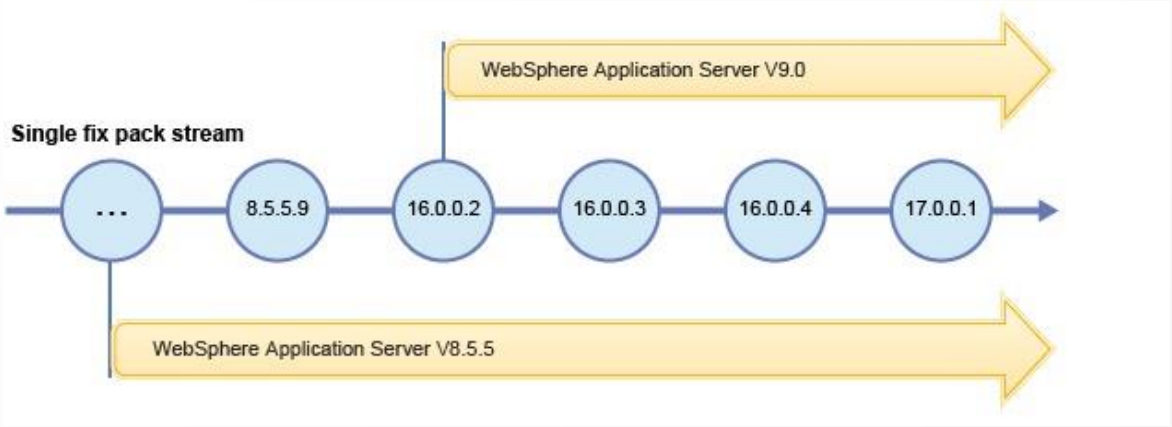


Point to your existing JRE

Within supported software levels. Java 6 will soon be out of service.
Each features has a minimum level of Java

What's With the 16.0.0.2?

Liberty Continuous Delivery



WAS V9 introduces a new 9.0.0.0 service stream for traditional WAS fixpacks.
WAS Liberty follows a continuous delivery model with a single service stream – it makes no sense to persist with a major version for Liberty fixpacks in a 'versionless' single delivery stream.
To properly reflect this we are changing Liberty fixpack numbering to:
Y.R.M.F: year.release.modlevel.fixpack
eg 16.0.0.4 would be the fourth fix pack in 2016

The next fixpack after 8.5.5.9 is the second fixpack of 2016 and is numbered 16.0.0.2.
*Its just a number - **there is no migration required at all** to the first V9 release of Liberty: 16.0.0.2.*

Quick links for Additional Information

WebSphere Application Server v9 new capabilities on YouTube

- Intelligent management
- API Lifecycle Management
- Integration with cloud services
- Improved security & availability

WebSphere Migration Tools

- WebSphere Migration Discovery Tool
- WebSphere Migration Toolkit
- Urban Code Deploy

UrbanCode Deploy WAS plugin

- Automatically migrates configurations to newer versions of WAS on premises or IBM Bluemix

Accelerate Microservices Creation

- WebSphere Liberty App Accelerator http://wasdev.net/microservices

WebSphere Connect

ibm.com/cloud-computing/connect/websphere

Thank You

Recent Customer requirements delivered

Create and vote on Requests For Enhancement

- https://www.ibm.com/developerworks/rfe/execute?use_case=submitRfe

Update jython level for wsadmin (most voted RFE!)
Liberty quiesce server options
Liberty server maintenance mode
DB2 client reroute
websocket ++
Automatic EAR/WAR expansion
Option to package microservices as a standalone JAR (in addition to zip)
Standalone Web server plugin configuration merge tool for Liberty
OSGi apps access to shared libraries
Liberty web services policy flexible configuration
zos Liberty Request logging to SMF 120-11 records