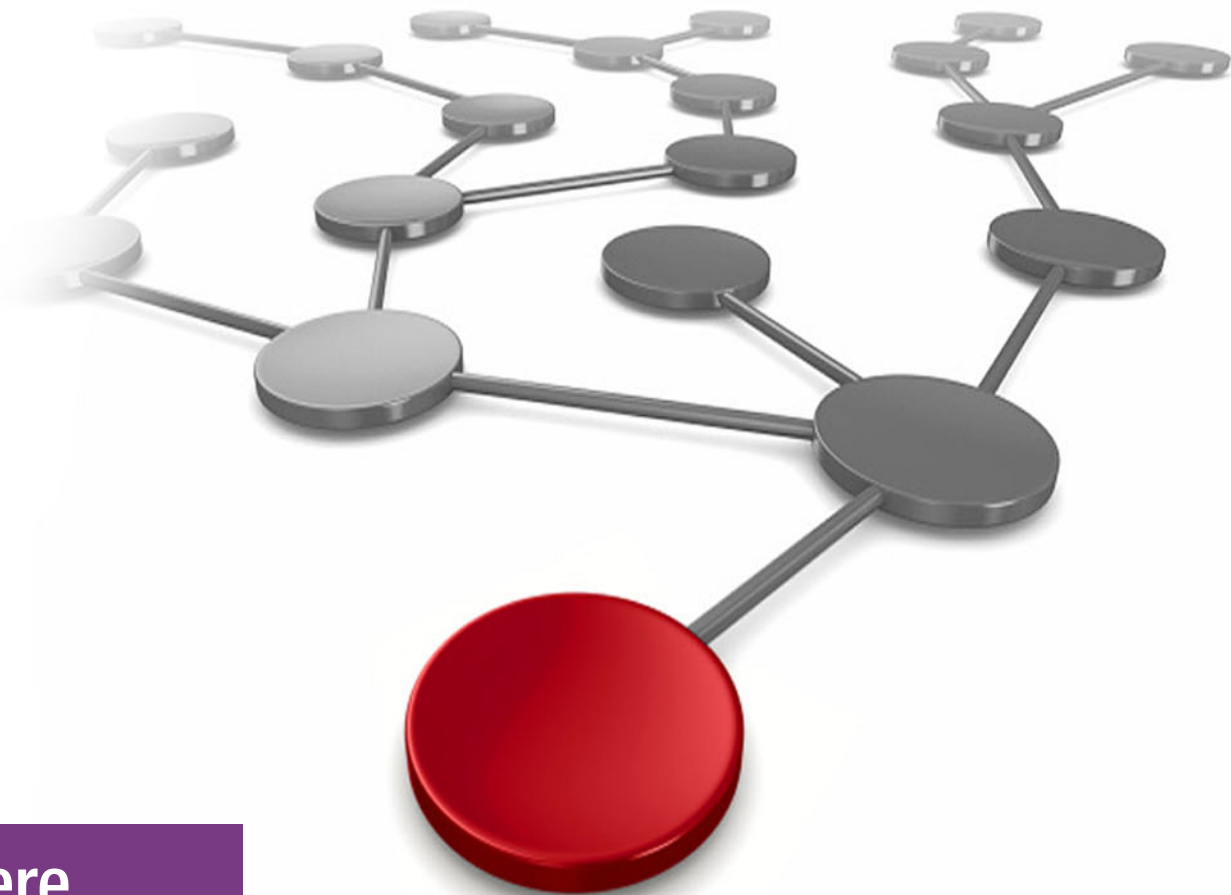


Liberty in a DevOps Continuous Delivery Environment

Sima Modir



WebSphere

Liberty in a DevOps Continuous Delivery Environment

This IBM® Redbooks® Solution Guide describes the benefits of Java development using IBM WebSphere® Application Server Liberty (Liberty) in a DevOps manner. The guide shows using a number of IBM and non-IBM tools that can be used throughout the application lifecycle management (ALM).

Liberty is a highly composable, fast to start, dynamic application server runtime environment that has the following characteristics:

- ▶ Easy configuration: Configuration is described in an XML file, which can be edited with a text editor and shared across the development team.
- ▶ Dynamic and composable runtime: Liberty loads only the applications deployed on the server, as needed.
- ▶ Quick server start time: It has been estimated that the server starts in less than 5 seconds for a basic web application.
- ▶ Low memory footprint: The server uses a less than 60 MB footprint for Java Platform, Enterprise Edition web application development.
- ▶ Extensible Liberty: Provides support for user and product extensions. System Programming Interfaces (SPIs) can be used to extend the runtime.
- ▶ Open service support: Liberty is built using the open standards Open Service Gateway Initiative (OSGi) Framework.

DevOps is an organizational capability focused on bringing development and operation teams together. It enables them to communicate and collaborate throughout the entire delivery pipeline. DevOps capabilities span the software delivery lifecycle. An organization determines where and how to implement DevOps based on business objectives, goals, challenges, and gaps in the organization's software delivery capabilities.

The DevOps adoption paths that need to be supported by development tooling are as follows:

- ▶ Continuous delivery

Automate software delivery to non-production and production environments across the delivery pipeline consistently and frequently. This DevOps practice focuses on reducing the amount of manual labor, resource wait time, and errors.
- ▶ Continuous integration

Test software earlier in the lifecycle and continuously test it across the lifecycle. This DevOps practice supports the *Shift Left* testing approach, which involves integration of the application code by using techniques, such as service virtualization and test automation during development where defects are easier and less costly to fix.
- ▶ Continuous feedback

Monitor application usage and provide metrics data to various stakeholders, such as operations, development, testing, and line-of-business owners.

By adopting DevOps practices, an organization can realize benefits, such as:

- ▶ Improve time to market with tighter traceability from requirements planning to production
- ▶ Improve the quality of the outcome
- ▶ Reduce time to gather and analyze customer feedback
- ▶ Monitor system usage and achieve higher service levels

Figure 1 shows the ALM lifecycle phases and DevOps practices that span across these phases.

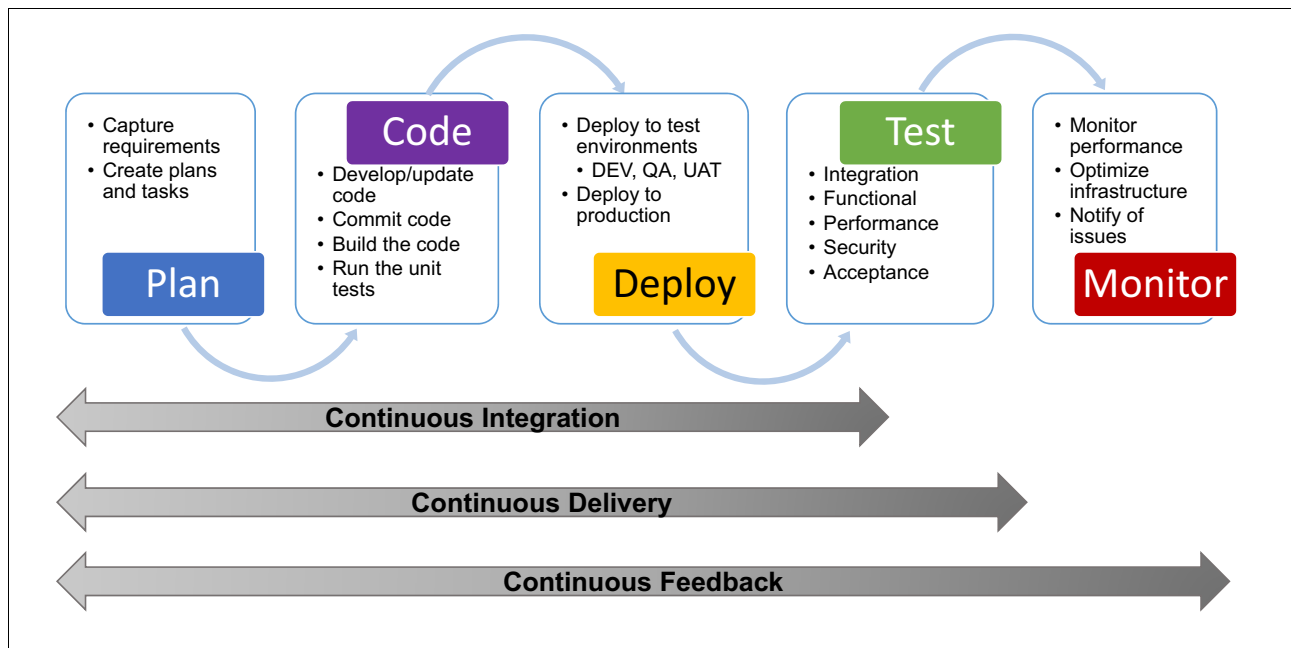


Figure 1 DevOps lifecycle

Did you know?

Liberty is designed with the needs of developers in mind. It evolved from the WebSphere Application Server product, making it much easier for developers to use for writing web, mobile, and Internet of Things (IoT) applications. Liberty offers the following unique advantages to developers:

- ▶ Simple XML-based server profile that can be checked into the source code repository and managed as a development artifact.
- ▶ Composable, modular server runtime, which loads the application server with only components that are needed by the application. This approach results in a more compact runtime that can get started a lot quicker.
- ▶ Liberty supports DevOps testing tools to *Shift Left* integration testing to the development phase.
- ▶ Liberty Developer Tools for Eclipse are available for developers to download into the Eclipse platform.
- ▶ Small runtime footprint allows the developer to quickly change their application and deploy it to a running application server.

- ▶ Liberty is designed to be cloud friendly and supports many cloud deployment models, such as:
 - IBM SoftLayer® infrastructure as a service (IaaS).
 - IBM Bluemix™ (platform as a service (PaaS)). There is an IBM Bluemix Liberty for Java applications service on Bluemix, which is powered by the IBM WebSphere Liberty Buildpack. It provides a complete runtime environment for running Java EE 7 and OSGi applications in addition to the Liberty profile. It supports popular frameworks, such as Spring and includes the IBM Java runtime environment (JRE). Other PaaS cloud platforms, such as OpenShift and Heruko are also supported.
 - Container as a service (CaaS), a pre-built Docker image containing the Liberty Profile v8.5.5 and IBM JRE 7.1 SR1, is available on the Docker hub. CaaS allows developers to get the Liberty environment up and running in a Docker container. For more information, see the following web address:

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

Liberty continues to provide reliability, availability, serviceability (RAS) characteristics that the WebSphere Application Server product is known for. In addition, it provides the following advantages to the production team:

- ▶ Its low memory usage allows more application instances to be supported for each production server.
- ▶ The runtime can be upgraded with no migration of user configuration or application.
- ▶ Liberty supports clustering for application high availability and scalability.
- ▶ Request timing is a new Liberty feature, which automatically identifies slow and hung requests and provides detailed information to help find the root cause.
- ▶ Liberty provides support for securing the server runtime environment and web applications by using user registries, authentication, and authorization.
- ▶ Liberty provides basic implementations of logging and trace services to help identify problems in production.

Business value

Businesses are under constant pressure to bring new innovative products and services to their customers much faster. They are also finding that traditional software delivery practices with long release cycles and waterfall processes requiring long maintenance windows are no longer sufficient. There is a desire to adopt agile DevOps practices and address bottlenecks in the delivery pipeline in order to improve time to market.

DevOps practices can enable an organization to transform their software delivery process to address gaps and bottlenecks in the delivery pipeline. Every organization's journey and priority focus might be different but some of the common gaps include time consuming manual software deployments, waterfall-based release management process, lack of traceability from business requirements to test, and not being able to quickly react to application issues or customer feedback.

Liberty provides an agile platform for development and operations of Java applications. It can easily integrate and be part of the organization's DevOps tooling to implement DevOps practices and bring "continuous improvement" to their clients.

Solution overview

In this Solution Guide, we use a pragmatic approach to show how different DevOps tools along with Liberty are used during each phase of the ALM, and highlight advantages of using Liberty for Java development.

The products that we focused on include: IBM Rational® Team Concert™, IBM Rational Quality Manager, IBM Rational Test Virtualization Server (RTVS), IBM Rational Test Workbench (RTWB), IBM UrbanCode™ Deploy, and Jenkins.

Figure 2 shows DevOps tooling that is used during each phase of the ALM.

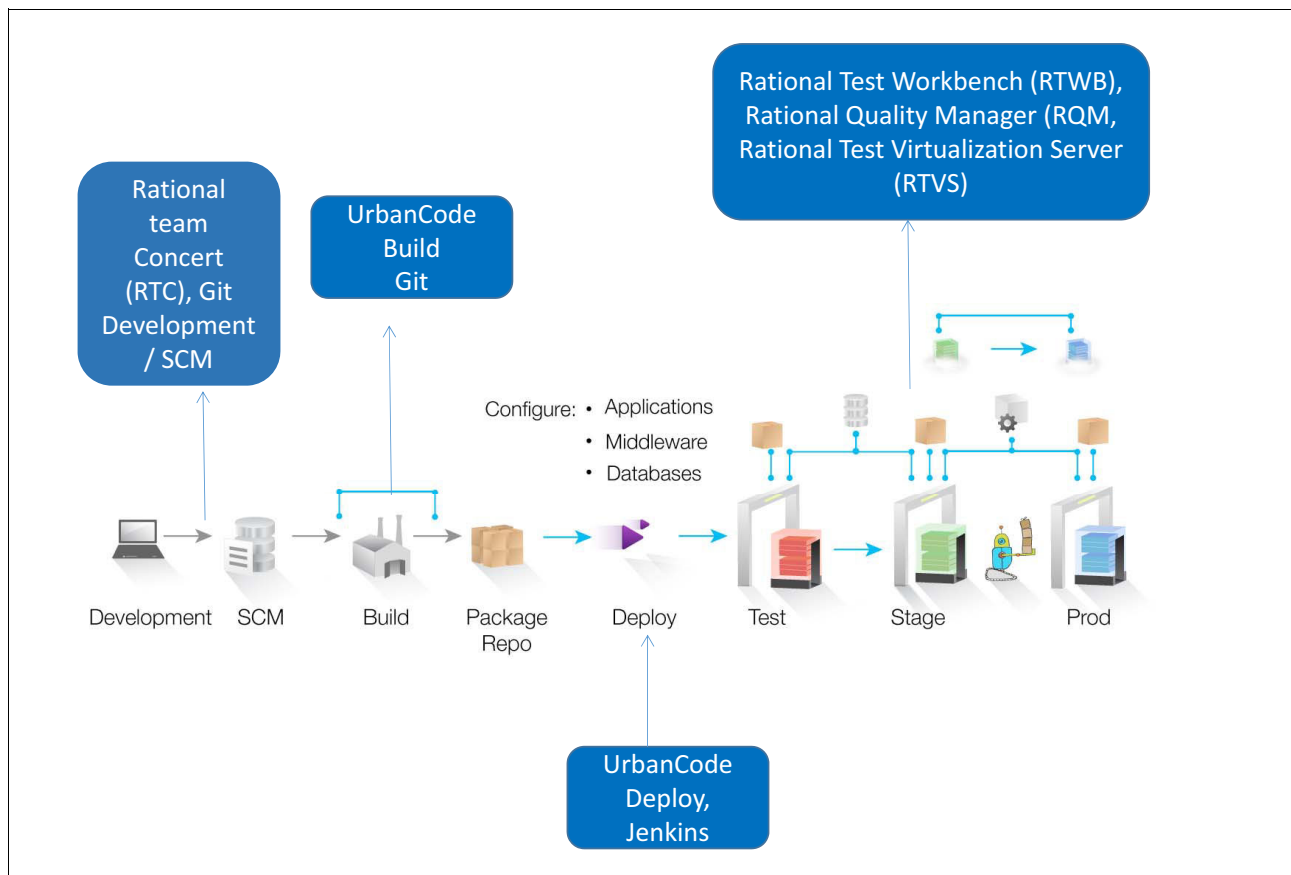


Figure 2 DevOps tooling

Planning with Rational Team Concert and Rational Quality Manager

The planning phase is a crucial stage of ALM that often gets undermined. In this phase, a number of key activities must occur to set the development on the correct track, such as:

- Requirements management

During this stage, business and IT stakeholders are involved. Key functional and non-functional requirements for the application are identified, which later form the basis of the application solution architecture and design. A proper requirements management tool, such as IBM Rational DOORS NG, can provide linkage with other artifacts in later stages of ALM. For example, a business requirement can get linked to functional use cases during development or test cases during testing and provide end-to-end traceability.

- **Project planning**

The planning component provides tools to assist with the planning and execution of the overall product backlog, releases, iterations, and the daily developer activities. IBM RTC can be used to build a high-level release plan, iteration, or individual project plans with specific activities. It provides a real-time dashboard that shows transparency and metrics on status of project activities to all stakeholders.

- **Test planning**

Test planning is about defining test cases, metrics, and the resources that are required for testing. IBM Rational Quality Manager, which is similar to Rational Team Concert, is built on the IBM Jazz™ platform. It provides a collaborative, role-based environment for test planning, tracking, and metrics reporting.

For a Liberty development project during the planning phase, a project manager can define a project plan and the testing team can use Rational Quality Manager to define a set of functional test cases. For this exercise, we did not include integration of the requirements management tool, such as IBM Rational DOORS® Next Generation, but looked at Rational Team Concert for planning and defined test cases in Rational Quality Manager. If the development team plans to use agile development methodology, Liberty is suited to an iterative development approach.

Coding and build with Rational Team Concert and Jenkins

Rational Team Concert provides a collaborative environment for development. A developer can view assigned tasks from the Rational Team Concert dashboard and view the status of the overall project.

For Liberty development, developers install Liberty tools in Rational Team Concert Eclipse client and start coding on their notebooks. Developers run JUnit for unit testing in Rational Team Concert. After unit testing is successfully completed, developers mark the work item complete. Rational Team Concert maintains a code repository of all code artifacts.

The Jenkins tool is used for creating a build. This is an alternative tool that can be integrated with IBM tools as part of the DevOps tooling. The Rational Team Concert plug-in can be installed in Jenkins.

The build flow in Rational Team Concert creates a request for a build, which links to appropriate code artifacts from the Rational Team Concert code repository. After Jenkins receives the build request from Rational Team Concert, it creates an application build.

Deployment with UrbanCode Deploy and Jenkins

A plug-in for UrbanCode Deploy (UCD) needs to be installed in Jenkins. Jenkins, in addition to creating a build, also publishes the build to UrbanCode Deploy.

UCD provides a framework to define applications, environments, components, and processes that are necessary for deployment.

UCD components for a Liberty application include an application server, application server configuration, and code (WAR or EAR file).

UCD resources are deployment target environments. For example, a deployment could include two target environments: a QA/Test server and a Production server with one UCD agent running at each target environment.

A UCD process is the workflow for installing and deploying a Liberty application in a target environment.

Testing with Rational Quality Manager, RTWB, and RTVS

In Rational Quality Manager, test cases for different test activities get defined, such as system and integration testing. These test cases are required to validate that the application being tested works correctly. There is a built-in integration between Rational Team Concert and Rational Quality Manager.

The RTWB solutions provide a test automation solution for mobile applications, regression testing, integration technologies, and performance and scalability testing. It can help to significantly reduce test cycle times by moving integration testing in the development lifecycle.

The RTVS is used for creating, maintaining, publishing, and running message-based stubs, database stubs, and rich virtualized applications.

IBM Rational Integration Test (part of the Rational Test Automation tool) can be used to record and create database stubs or web service or WebSphere MQ transaction stubs. The IBM DB2® database stub shows how a virtual stub can simulate a database and store data persistently.

Solution architecture

Figure 3 on page 7 shows the solution architecture for a Liberty DevOps pipeline solution.

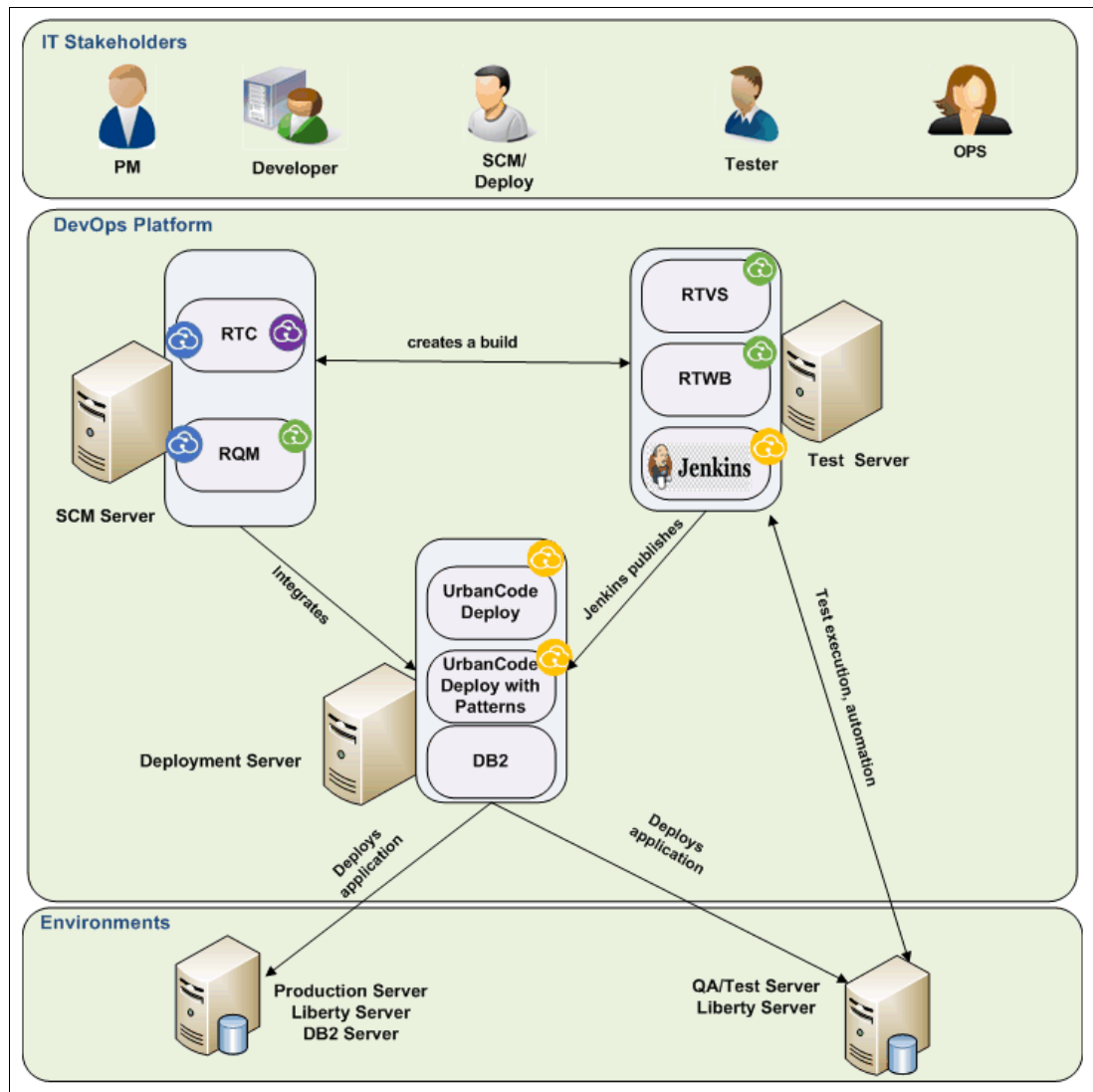


Figure 3 Liberty DevOps solution architecture

The solution architecture includes the following components:

► Target Environments

– QA Server:

An instance of Liberty Server version 8.5.5 is installed on the QA Server. In a given project, likely multiple QA environments are used for different stages of testing.

– Production Server:

An instance of Liberty Server version 8.5.5 and DB2 version 10.5 software are installed on the production server.

Because this is a fairly simple application, developers can use their notebooks as the development environment. Thus, a development server is not included in the solution. But in a typical project, it is necessary to also provision a development environment that is shared by the developers.

► DevOps Tooling Platform

DevOps toolset was installed on the following three servers:

– SCM Server

IBM Rational Team Concert™ version 5.0.2 and Rational Quality Manager version 5.0.2. software.

– Test Server

Rational Test Virtualization Service (RTVS) version 8.7.0, Rational Test Workbench (RTWB), and open source Jenkins version 1.6.0.9.2 software.

– Deployment Server

UrbanCode Deploy version 6.1.1 and UrbanCode Deploy with Patterns version 6.1.1. DB2 version 10.5 as well as Rational License Server (RLS) version 8.1.4. DB2 and RLS are used by DevOps tools.

DevOps flow of Liberty application

In this example, the iterative flow for taking Liberty application changes through the delivery pipeline is as follows (see Figure 3 on page 7):

1. The project or release manager creates a project or release plan in Rational Team Concert, which includes development-specific work items.
2. The development team uses Rational Team Concert for collaboration. Developer views are assigned project activities. They use the Liberty developer tool for Eclipse and start coding.
3. Each developer performs unit testing by using Junit with Rational Team Concert.
4. Once the developer is happy with the code, they check their code into the Rational Team Concert code repository.
5. The developer runs the whole set of Junit test cases in Rational Quality Manager and arranges for a code review.
6. SCM SME uses Rational Team Concert to send build creation requests to Jenkins and identify code components that should be included in that Build.
7. Jenkins creates a Build with a link to code artifacts that reside in Rational Team Concert and publishes the Build to the deployment server.
8. SCM SME creates a deployment process and snapshot in UrbanCode Deploy, which includes the latest Build created by Jenkins and deploys the snapshot to QA server.
9. The tester runs system and integration test cases defined in Rational Quality Manager and creates any defects in Rational Team Concert.
10. The tester runs test automation and test virtualization test cases.
11. If there are any defects that are generated during any of these testings, defects are created in Rational Team Concert and get assigned to the developer.
12. Developer picks up defects in Rational Team Concert and adds fixes to code. This cycle continues until all defects are addressed and wanted metrics are achieved.
13. A business decision is made to deploy that release of code to production. The Ops team uses the deployment server to deploy the code snapshot that has gone through the testing cycle to the production server.
14. Ops uses monitoring tools to track usage and provide metrics to stakeholders.

Usage scenarios

The application use case that is used is called *LibertyCloudTrader*. It is an application that simulates an online stock trading system. The program allows users to perform the following functions:

- ▶ Log in
- ▶ View their portfolio
- ▶ Look up stock quotes
- ▶ Buy or sell shares

The LibertyCloudTrader application is based on the CloudTrader application, which is developed by using Liberty Profile Server. The application is built primarily with Java servlets, JavaServer Pages (JSP), and beans. It makes a SOAP web service call to an external DayTrade application to retrieve stock quotes. It simulates trade transactions by populating a local DB2 database. LibertyCloudTrader has a Java Database Connectivity (JDBC) connection to the DB2 database.

The application can be accessed either through an HTTP web interface or a Representational State Transfer (REST) application programming interface (API), which is added to show a mobile interface.

The application can be accessed either through an HTTP web interface or a REST API, which is added to show a mobile interface.

Figure 4 shows a system context diagram for a LibertyCloudTrader application.

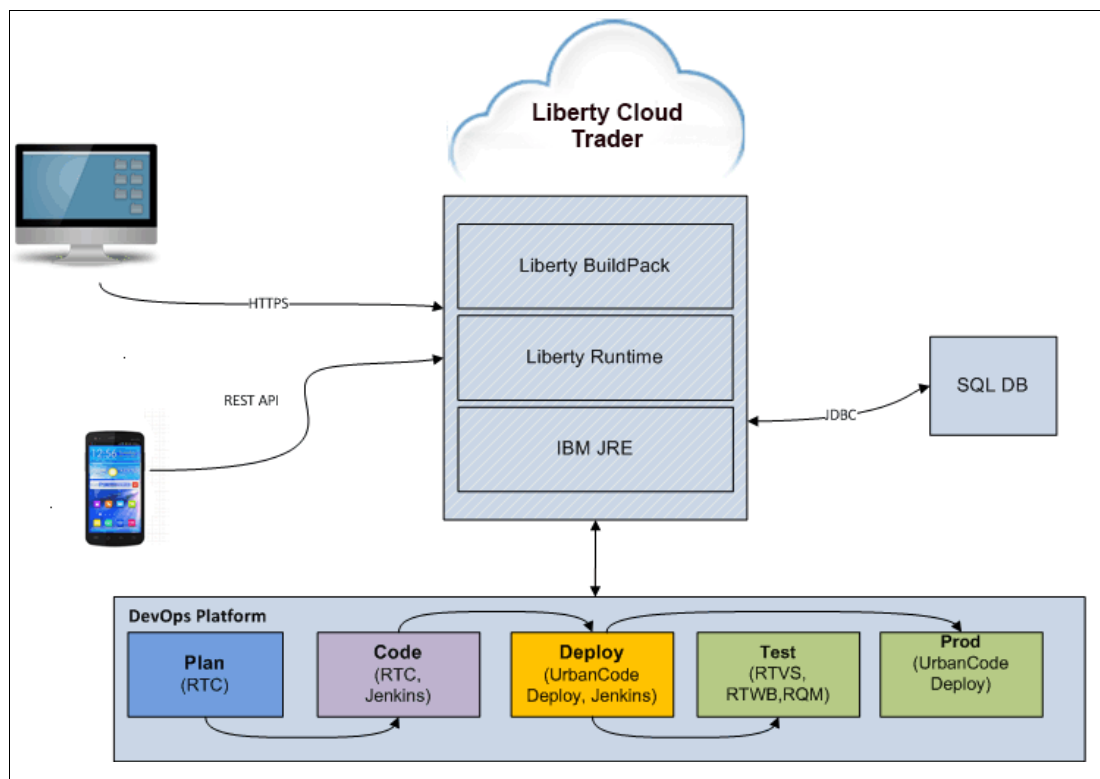


Figure 4 LibertyCloudTrader system context diagram

Integration

The DevOps toolset can be different from one organization to the next, but it is important to ensure that selected DevOps tools can be integrated to provide traceability and ease of flow of applications through the pipeline.

For Liberty development, you can use a preintegrated set of DevOps tools, such as Rational tool suite or leverage other DevOps tools including open source tools. Liberty can be used effectively with many of these DevOps tools at each phase of the ALM to allow an easy flow of the application through the pipeline.

For more information, see the Liberty WebSphere Application Server DevOps page on IBM developerWorks:

<https://developer.ibm.com/wasdev/docs/category/devops>

Supported platforms

WebSphere Application Server is supported on IBM AIX®, HP-UX, IBM i, Linux, Solaris, Microsoft Windows, and IBM z/OS®. MAC OS X is also supported for development. Complete information about the supported platforms and hardware can be found at the following site:

<http://www.ibm.com/software/products/us/en/appserv-was>

Ordering information

The following product is only available through IBM Passport Advantage®. Ordering information can be found in the software announcement for:

WebSphere Liberty Profile Application Server V8.5.5 Software Announcement:

<http://ibm.co/13tRFV2>

Related information

For more information, see the following documents:

- ▶ IBM Redbooks publication: *Liberty and DevOps, continuous delivery and deployment environment*, SG24-8286:
<http://www.redbooks.ibm.com/abstracts/sg248286.html>
- ▶ IBM Redbooks publication: *IBM WebSphere Application Server Liberty Profile Guide for Developers*, SG24-8076
<http://www.redbooks.ibm.com/abstracts/sg248076.html>
- ▶ WebSphere Application Server V8.5.5:
<http://www.ibm.com/support/docview.wss?uid=swg24034969>
- ▶ IBM UrbanCode Deploy:
<http://www.ibm.com/software/products/en/ucdep>

- ▶ Rational Quality Manager 5.02
<http://www.ibm.com/software/products/en/ratiquallmana>
- ▶ Rational Team Concert 5.02:
<http://www.ibm.com/software/products/en/rtc>
- ▶ Rational Test Workbench
<http://www.ibm.com/software/products/en/rtw>
- ▶ IBM Offering Information page (announcement letters and sales manuals):
http://www.ibm.com/common/ssi/index.wss?request_locale=en

On this page, enter one of the following products: WebSphere Application Server V8.5.5, IBM DB2 10.5, UrbanCode Deploy 6.1.1, Rational License Key Server 8.1.4, Rational Team Concert 5.02. Select the information type, and then click **Search**. On the next page, narrow your search results by geography and language.

Authors

This Solution Guide was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.

Sima Modir is a Senior IT Architect with extensive experience in sales and delivery. She is currently a Client Technical Lead in Sales and Distribution. She is responsible for architecture and delivery of complex cross-brand IBM solutions. Sima has worked on various roles including project management, architecture, development, and testing. Sima has over 20 years of client-facing experience. Her industry experience includes telecommunications, media, retail, energy, and utilities. Sima is currently focused on implementing a DevOps continuous delivery and tooling solution.

This project was led by Margaret Ticknor, a Redbooks Project Leader in the Raleigh Center. She primarily leads projects about WebSphere products and IBM PureApplication® System. Before joining the ITSO, Margaret worked as an IT specialist in Endicott, NY. Margaret attended the Computer Science program at State University of New York at Binghamton.

Thanks to Ann Lund, IBM Redbooks Residency Administrator, for her support of this project.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<https://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document, REDP-5269-00, was created or updated on October 14, 2015.


Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
Bluemix™
Concert™
DB2®
DOORS®
IBM®

IBM UrbanCode™
Jazz™
Passport Advantage®
PureApplication®
Rational®
Rational Team Concert™

Redbooks®
Redbooks (logo) ®
WebSphere®
z/OS®

The following terms are trademarks of other companies:

SoftLayer, and SoftLayer device are trademarks or registered trademarks of SoftLayer, Inc., an IBM Company.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.



REDP-5269-00

ISBN 0738454575

Printed in U.S.A.

Get connected

