

Essentials of Cloud Application Development on IBM Bluemix

Presentations Guide

Ahmed Azraq

Hala A Aziz

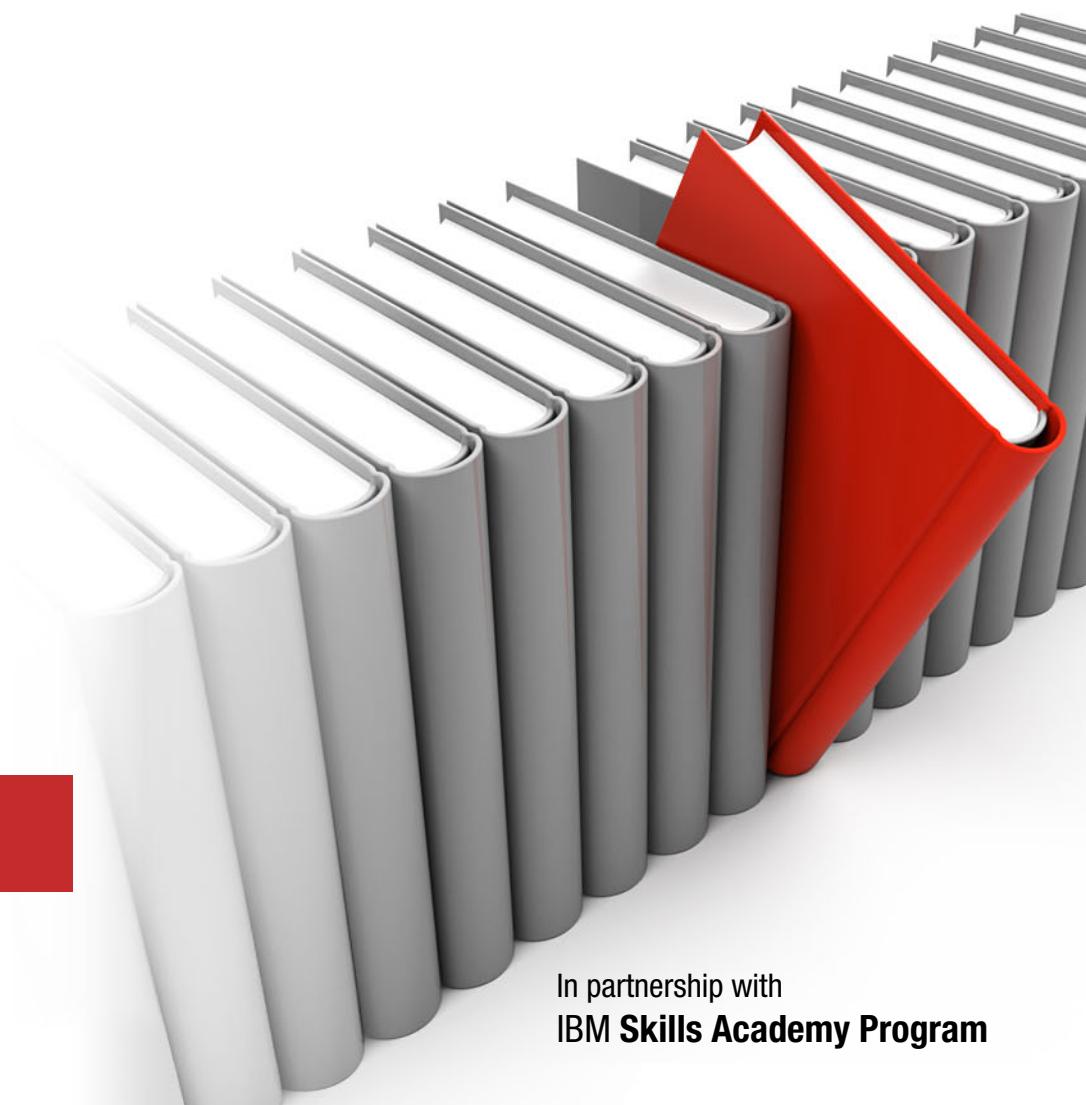
Mohamed El-Khouly

Sally Fikry

Ahmed S Hassan

Mahmoud Rasmy

Ben Smith



In partnership with
IBM Skills Academy Program



International Technical Support Organization

**Essentials of Cloud Application Development on IBM
Bluemix**

August 2017

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

Second Edition (August 2017)

This edition applies to IBM Bluemix.

© Copyright International Business Machines Corporation 2016, 2017. All rights reserved.

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

Contents

Notices	vii
Trademarks	viii
Preface	ix
Authors	x
Now you can become a published author, too!	xii
Comments welcome	xii
Stay connected to IBM Redbooks	xii
Unit 1. Introduction to cloud computing and IBM Bluemix	1
1.1 What you should be able to do	2
1.2 References	2
1.3 What is cloud computing?	3
1.4 As opposed to	4
1.5 Factors contributing to growth of cloud	5
1.6 Cloud and mobile computing are changing traditional IT	6
1.7 Cloud service models	7
1.8 Infrastructure as a service architecture	8
1.9 IBM SoftLayer: IaaS offering from IBM	9
1.10 Platform as a service (PaaS) architecture	10
1.11 Software as a service (SaaS) architecture	11
1.12 Split of provider-side and consumer-side responsibilities	12
1.13 Cloud computing: Benefits for developers	13
1.14 What is IBM Bluemix?	14
1.15 IBM Bluemix ecosystem	15
1.16 Bluemix architectural overview	16
1.17 Bluemix: Choice of runtimes	18
1.18 Bluemix: Services	19
1.19 Bluemix: Regions	20
1.20 Unit summary	21
1.21 Checkpoint questions	22
1.22 Checkpoint answers	23
Unit 2. Getting started with IBM Bluemix	25
2.1 What you should be able to do	26
2.2 References	26
2.3 Getting started: Creating an IBM Bluemix account	27
2.4 IBM Bluemix infrastructure types	28
2.5 What can you build in IBM Bluemix?	30
2.6 Cloud Foundry Apps, Containers, and Virtual Servers	31
2.7 Containers versus Virtual Servers	32
2.8 IBM Bluemix catalog: Infrastructure	34
2.9 IBM Bluemix catalog: Apps	35
2.10 IBM Bluemix catalog: Boilerplates	36
2.11 IBM Bluemix catalog: Cloud Foundry Apps	37
2.12 IBM Bluemix catalog: Containers	38
2.13 IBM Bluemix catalog: OpenWhisk	39
2.14 IBM Bluemix catalog: Mobile	40
2.15 IBM Bluemix catalog: Services	41

2.16 Creating an IBM Bluemix application (1 of 3)	42
2.17 Creating an IBM Bluemix application (2 of 3)	43
2.18 IBM Bluemix app name must be unique across domain.	44
2.19 Creating an IBM Bluemix application (3 of 3)	45
2.20 IBM Bluemix Dashboard	46
2.21 IBM Bluemix Application Details page	47
2.22 Testing applications through the application route	48
2.23 Adding an IBM Bluemix service	49
2.24 Binding a service to an application	50
2.25 IBM Bluemix environment variables	51
2.26 IBM Bluemix organizations and users (team members)	52
2.27 IBM Bluemix domains and quota	53
2.28 IBM Bluemix organizations: Spaces	54
2.29 IBM Bluemix organizations: User roles	55
2.30 Managing IBM Bluemix organizations.	56
2.31 Inviting users to an organization	57
2.32 Unit summary	58
2.33 Checkpoint questions	59
2.34 Checkpoint answers	60
Unit 3. Developing IBM Bluemix applications from a local workstation	61
3.1 What you should be able to do	62
3.2 References	62
3.3 Node.js	63
3.4 Options to develop and deploy your IBM Bluemix app	64
3.5 Overview of bx and cf command-line interfaces	65
3.6 Why use command-line tools?	66
3.7 Reviewing the documentation	67
3.8 Using the cf CLI, Step 1: Install the cf CLI	68
3.9 Using the cf CLI, Step 2: Connect to your IBM Bluemix account	69
3.10 Using the cf CLI, Step 3: Deploy your application to IBM Bluemix	70
3.11 Using the IBM Bluemix CLI, Step 1: Install the IBM Bluemix CLI	71
3.12 Using the IBM Bluemix CLI, Step 2: Connect to your IBM Bluemix account.	72
3.13 Using the IBM Bluemix CLI, Step 3: Deploy your application to IBM Bluemix	73
3.14 Test your IBM Bluemix application	74
3.15 IBM Bluemix and Eclipse	75
3.16 Step 1: Review the documentation	76
3.17 Step 2: Install the latest version of Eclipse	77
3.18 Step 3: Install Eclipse and Eclipse tools for IBM Bluemix	78
3.19 Step 4: Connect to your IBM Bluemix account	79
3.20 Step 5: Create a Node.js application project.	80
3.21 Unit summary	81
3.22 Checkpoint questions	82
3.23 Checkpoint answers	83
Unit 4. Using IBM Bluemix DevOps services	85
4.1 What you should be able to do	86
4.2 References	86
4.3 What is DevOps?	87
4.4 Benefits of DevOps	88
4.5 What is IBM Bluemix DevOps Services?	89
4.6 What services does IBM Bluemix DevOps provide?	90
4.7 Additional DevOps tool Integration	91

4.8 Setting up an IBM Bluemix DevOps Services project	92
4.9 New tab opens to create Continuous Delivery toolchains	93
4.10 Web IDE: Edit Code features	94
4.11 Web integrated development environment	95
4.12 Editing source code	96
4.13 Editor features: Code completion	97
4.14 Editor features: Run bar	98
4.15 Bluemix Live Sync features	99
4.16 Source control with a Git repository	101
4.17 Git repository overview	102
4.18 Connect a Git client to your IBM Bluemix DevOps project	103
4.19 IBM Bluemix DevOps Services: Delivery Pipeline	104
4.20 Customizing the delivery pipeline	105
4.21 Example: Default delivery pipeline	106
4.22 Configuring the Build Stage	107
4.23 Configuring Build jobs	108
4.24 Configuring Deploy jobs	109
4.25 Configuring Test job	110
4.26 Example: A successful build and deploy result	111
4.27 Issue Tracking tool	112
4.28 Creating an issue	113
4.29 Unit summary	114
4.30 Checkpoint questions	115
4.31 Checkpoint answers	116
Unit 5. REST architecture and Watson APIs	117
5.1 What you should be able to do	118
5.2 References	118
5.3 What is REST?	119
5.4 Applying REST to server-side applications	121
5.5 Example: Application model architecture for REST services	122
5.6 What is a RESTful web service?	123
5.7 Example: Sending an HTTP request to a REST service	124
5.8 Example: Receiving an HTTP response from a REST service	125
5.9 REST characteristics	126
5.10 Introduction to JSON	127
5.11 JSON data types	128
5.12 JSON data type: Objects	129
5.13 JSON data type: Arrays	130
5.14 What is Watson?	131
5.15 Watson Services in IBM Bluemix	132
5.16 Watson API Explorer	133
5.17 Example: Watson API Explorer - Natural Language Understanding (Authors)	134
5.18 Unit summary	136
5.19 Checkpoint questions	137
5.20 Checkpoint answers	138
Unit 6. Introduction to data services in IBM Bluemix	139
6.1 What you should be able to do	140
6.2 References	140
6.3 Database choices on Bluemix	141
6.4 Data services in IBM Bluemix catalog	142
6.5 Cloudant capabilities	143

6.6	Cloudant in IBM Bluemix versus Cloudant.com	145
6.7	Documents in Cloudant.	146
6.8	Getting started with Cloudant on IBM Bluemix	147
6.9	IBM Bluemix Cloudant: VCAP_SERVICES.	148
6.10	Cloudant Dashboard	149
6.11	Cloudant REST API	150
6.12	Sample database at Cloudant	151
6.13	Reading a document in Cloudant	152
6.14	View all documents	153
6.15	More Cloudant REST APIs	154
6.16	Cloudant query	155
6.17	HTTP status codes	156
6.18	Unit summary	157
6.19	Checkpoint questions	158
6.20	Checkpoint answers	159
Unit 7.	IBM Bluemix mobile backend as a service	161
7.1	What you should be able to do	162
7.2	References	162
7.3	What is mobile backend as a service (MBaaS)?	163
7.4	IBM Mobile backend services (1 of 2)	164
7.5	IBM Mobile backend services (2 of 2)	165
7.6	MBaaS architecture.	166
7.7	Push Notifications service	168
7.8	Push notification process	169
7.9	Configuring push notifications in IBM Bluemix	170
7.10	Sending manual notifications from Bluemix	171
7.11	App ID service.	172
7.12	App ID Authentication options.	173
7.13	App ID architecture	174
7.14	What is Kinetise?	176
7.15	How to use Kinetise	177
7.16	MobileFirst Services Starter Boilerplate	178
7.17	Unit summary	179
7.18	Checkpoint questions	180
7.19	Checkpoint answers	181

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered 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 <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AppScan®	IBM®	Rational Team Concert™
Bluemix®	IBM API Connect™	Redbooks®
Cloudant®	IBM MobileFirst™	Redbooks (logo)  ®
Concert™	IBM Watson®	WebSphere®
dashDB®	Jazz™	
Global Business Services®	Rational®	

The following terms are trademarks of other companies:

SoftLayer, and The Planet 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.

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.

Preface

This IBM® Redbooks® publication is based on the Presentations Guide of the course *Essentials of Cloud Application Development on IBM Bluemix* that was developed by the IBM Redbooks team in partnership with IBM Skills Academy Program.

This course is designed to teach university students the basic skills that are required to develop, deploy, and test cloud-based applications that use the IBM Bluemix® cloud services.

The primary target audience for this course is university students in undergraduate computer science and computer engineer programs with no previous experience working in cloud environments. However, anyone new to cloud computing can also benefit from this course.

After completing this course, you should be able to accomplish the following tasks:

- ▶ Define cloud computing
- ▶ Describe the factors that lead to the adoption of cloud computing
- ▶ Describe the choices that developers have when creating cloud applications
- ▶ Describe infrastructure as a service, platform as a service, and software as a service
- ▶ Describe IBM Bluemix and its architecture
- ▶ Identify the runtimes and services that IBM Bluemix offers
- ▶ Describe IBM Bluemix infrastructure types
- ▶ Create an application in IBM Bluemix
- ▶ Describe the IBM Bluemix dashboard, catalog, and documentation features
- ▶ Explain how the application route is used to test an application from the browser
- ▶ Create services in IBM Bluemix
- ▶ Describe how to bind services to an application in IBM Bluemix
- ▶ Describe the environment variables that are used with IBM Bluemix services
- ▶ Explain what are IBM Bluemix organizations, domains, spaces, and users
- ▶ Describe how to create an IBM SDK for Node.js application that runs on IBM Bluemix
- ▶ Explain how to manage your IBM Bluemix account with the Cloud Foundry CLI
- ▶ Describe how to set up and use the IBM Bluemix plug-in for Eclipse
- ▶ Describe the role of Node.js for server-side scripting
- ▶ Describe IBM Bluemix DevOps Services and the capabilities of IBM DevOps Services
- ▶ Identify the Web IDE features in IBM Bluemix DevOps
- ▶ Describe how to connect a Git repository client to Bluemix DevOps Services project
- ▶ Explain the pipeline build and deploy processes that IBM Bluemix DevOps Services use
- ▶ Describe how IBM Bluemix DevOps Services integrate with the IBM Bluemix cloud
- ▶ Describe the agile planning tools in IBM Bluemix
- ▶ Describe the characteristics of REST APIs
- ▶ Explain the advantages of the JSON data format
- ▶ Describe an example of REST APIs using Watson

- ▶ Describe the main types of data services in IBM Bluemix
- ▶ Describe the benefits of IBM Cloudant®
- ▶ Explain how Cloudant databases and documents are accessed from IBM Bluemix
- ▶ Describe how to use REST APIs to interact with Cloudant database
- ▶ Describe Bluemix mobile backend as a service (MBaaS) and the MBaaS architecture
- ▶ Describe the Push Notifications service
- ▶ Describe the App ID service
- ▶ Describe the Kinetise service
- ▶ Describe how to create Bluemix Mobile applications by using MobileFirst Services Starter Boilerplate

The workshop materials were created in June 2017. Therefore, all IBM Bluemix features that are described in this Presentations Guide and IBM Bluemix user interfaces that are used in the examples are current as of June 2017.

Note: This IBM Redbooks publication references exercises that are *not* included in this book. The exercises are available only to students who attended the course.

Authors

This book was produced by a team of specialists from around the world working with the IBM International Technical Support Organization (ITSO), and IBM Skills Academy Program.

Ahmed Azraq is a Certified IT Specialist in IBM Egypt. Since joining IBM in 2012, Ahmed worked as a Senior Cloud Developer, Technical Team Leader, and Architect in the IBM Middle East and Africa (MEA) Client Innovation Center, which is part of IBM Global Business Services® (GBS). His areas of expertise include cloud, IBM Business Process Manager, middleware integration that uses IBM Integration Bus, Java, and IBM Watson®. Ahmed has acquired several professional certifications, including Open Group IT Specialist, IBM Certified Solution Advisor - Cloud Reference Architecture, IBM Bluemix, Java EE, IBM Business Process Manager, Agile development process, and IBM Design Thinking. Ahmed also has delivered training on IBM Bluemix, DevOps, hybrid cloud Integration, Node.js, Watson APIs, and IBM WebSphere® Liberty Profile to IBM clients, IBM Business Partners, university students and professors around the world. He is the recipient of several awards, including Eminence and Excellence Award in the IBM Watson worldwide competition Cognitive Build, the IBM Service Excellence Award for showing excellent client value behaviors, and knowledge-sharing award. Ahmed also is the author of the IBM Redbooks publication *Building Cognitive Applications with IBM Watson Services: Volume 2 Conversation*, SG24-8394.

Hala A. Aziz is an IT Specialist in the Cairo Technology Development Center (CTDC) in IBM Egypt. She has more than 10 years of experience in IBM Application and Integration Middleware software, such as IBM WebSphere Application Server, IBM WebSphere Portal, IBM MobileFirst™, and IBM Endpoint Manager. Hala's expertise also includes IBM Cloud and IBM Bluemix. She worked as a consultant on eGovernment, telecom, and banking solutions for clients in Egypt, Saudi Arabia, Dubai, Oman, and Switzerland. Hala has several technical professional certifications, including Certified Application Developer for IBM Web Content Manager and IBM Worklight. Hala has published several technical articles and IBM Redbooks publications. She has also delivered IBM internal education and client enablement training workshops around the world.

Mohamed El-Khouly is a certified IT specialist in IBM Egypt. In his 12-year career with IBM, Mohamed held various roles in product development and IBM IT services. Mohamed's current focus is IBM Cloud services. Mohamed has delivered several IBM Bluemix training courses.

Sally Fikry is a Certified Expert IT Specialist in the IBM Egypt Cloud unit. Sally has over 11 years experience providing consulting services to IBM clients on middleware and hybrid cloud integration. Throughout her career, Sally participated in many consulting projects with IBM clients worldwide. She had several roles, including supporting pre-sales activities, consulting, solution design, development, and leading technical teams. She holds several technical certifications. Sally's current focus is to help IBM clients to add value to their businesses through hybrid clouds and middleware offering on cloud, such as IBM API Connect™, IBM Integration Bus, IBM MQ, and IBM Bluemix.

Ahmed S. Hassan has over 11 years experience in Information technology. He worked as software developer and Integration engineer for many projects in different industries, including electronic design automation, electronic payment, telecommunications, and travel and transportation. Ahmed is an IBM Certified Cloud Application Developer.

Mahmoud Rasmy is a Cloud Specialist in the IBM Hybrid Cloud organization, IBM Egypt.

Ben Smith is a Solution Developer in the Emerging Technology Institute in the IBM laboratory in Research Triangle Park (RTP), Raleigh, North Carolina. Ben's current focus is IBM Blockchain and open source hyperledger. He is a recipient of the IBM PhD Fellowship for 2010 and 2011 and received his Doctorate in Computer Science in July of 2012. Since joining IBM in August of 2012, Ben has integrated IBM WebSphere Liberty, the premiere web application platform from IBM, into Cloud Foundry. Ben was instrumental in standing up the original version of IBM Bluemix, the platform as a service (PaaS) IBM offering that is based on Cloud Foundry. Ben continues to help to integrate IBM WebSphere Liberty and other IBM products and solutions into this new Cloud platform. Ben has also worked with several IBM clients to stand up their own on-premises instances of IBM Bluemix, including IBM cloud services and the IBM WebSphere Liberty buildpack.

The project that produced this publication was managed by **Marcela Adan**, IBM Redbooks Project Leader, ITSO.

Thanks to the following people for their contributions to this project:

Juan Pablo Napoli
IBM Skills Academy Program Manager

Arlemi Turpault
IBM Global Markets, IBM France

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

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:
ibm.com/redbooks
- ▶ Send your comments in an email to:
redbooks@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://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>



Introduction to cloud computing and IBM Bluemix

This unit covers the following topics:

- ▶ Cloud overview
- ▶ Bluemix overview

1.1 What you should be able to do

After completion of this unit, you should be able to:

- ▶ Define cloud computing
- ▶ Describe the factors that lead to the adoption of cloud computing
- ▶ Describe the choices that are available to developers when creating cloud applications
- ▶ Describe infrastructure as a service, platform as a service, and software as a service
- ▶ Describe IBM Bluemix
- ▶ Describe the architecture of IBM Bluemix
- ▶ Identify the runtimes and services that IBM Bluemix offers

1.2 References

The following publications are useful for further research on the topic that is presented in this unit:

- ▶ What is Cloud Computing?
<https://www.ibm.com/cloud-computing/what-is-cloud-computing>
- ▶ Cloud Service Models:
 - <https://www.ibm.com/cloud-computing/iaas-paas-saas>
 - <http://www.ibm.com/developerworks/cloud/library/cl-cloudservicemodels/>
- ▶ Bluemix Runtimes:
<https://console.ng.bluemix.net/docs/cfapps/runtimes.html>
- ▶ Bluemix Services:
https://console.ng.bluemix.net/docs/services/index.html#experimental_services
- ▶ Services by Region:
https://console.ng.bluemix.net/docs/services/index.html#services_region
- ▶ What is IBM Bluemix?
<https://www.ibm.com/developerworks/cloud/library/cl-bluemixfoundry/>
- ▶ Bluemix fundamentals: 5 key advantages, from a developer's perspective:
<https://ibm.biz/BdjxBP>

1.3 What is cloud computing?

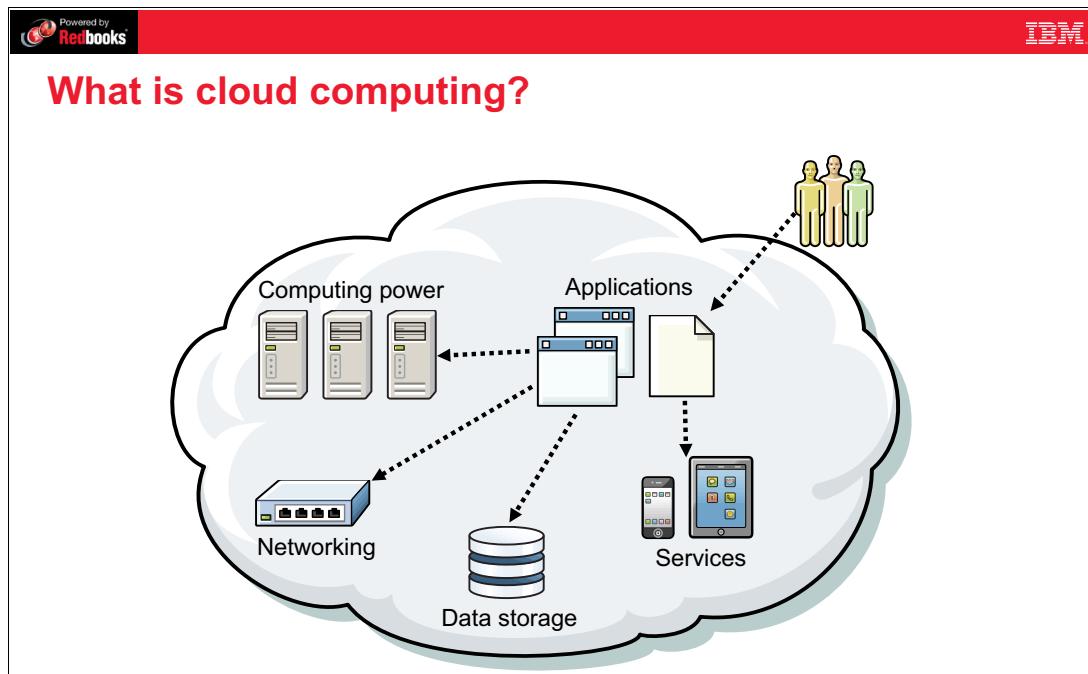


Figure 1-1 What is cloud computing?

Notes:

The term *cloud* is used as a metaphor for the internet and a virtualized set of hardware resources. The term cloud is an abstraction for the complex infrastructure it conceals. The generally accepted definition of cloud computing comes from the National Institute of Standards and Technology (NIST). The NIST definition runs to several hundred words but essentially says that:

“Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

Examples of computing resources include:

- ▶ Networks
- ▶ Servers
- ▶ Storage
- ▶ Applications
- ▶ Services

1.4 As opposed to...

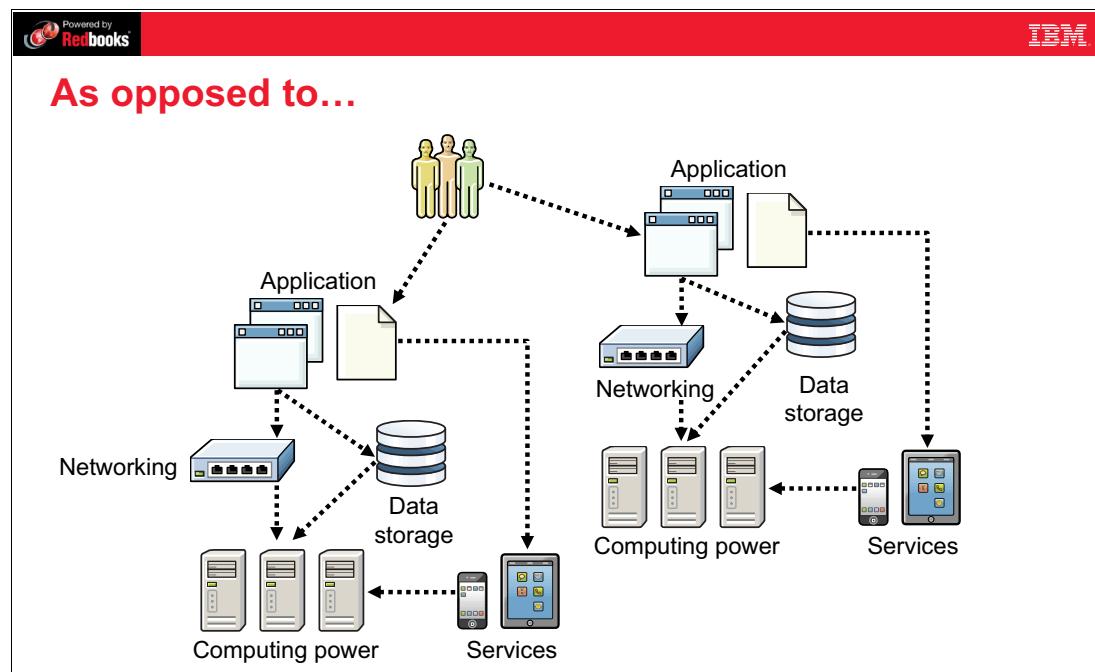


Figure 1-2 As opposed to...

Notes:

Cloud computing as a deployment model is replacing an older approach where each application that a user interacts with had its own custom built services, networking, data storage, and computing power.

The ability to reuse and repurpose hardware rapidly, and to host multiple applications and systems within a single set of hardware in an isolated fashion, are some of the main characteristics driving the adoption of cloud computing.

In the old approach, the IT staff needs to manage the entire stack, from hardware all the way to the latest software changes. This model does not scale as well as today's businesses and organizations require.

1.5 Factors contributing to growth of cloud

The slide has a red header bar with the 'Powered by Redbooks' logo on the left and the 'IBM' logo on the right. The main title 'Factors contributing to growth of cloud' is in bold red font at the top. Below it is a bulleted list of four factors:

- Applications with a short lead time to delivery
- Developers expect to have programming language options and interact with predefined services
- Modern applications must be able to scale and be managed dynamically
- Developers expect the pay-as-you-go (PAYG) utility computing billing method

Figure 1-3 Factors contributing to growth of cloud

Notes:

One factor contributing to the growth of cloud computing is that today's applications require a short time to delivery. Developers are pressured to get their product to market as soon as possible. They want to get feedback quickly, and then iterate on the idea to make the product better, faster.

Cloud makes hardware resources readily available and quick to configure, which shortens the time required for developers to show a working version of their products. Also, cloud allows the reuse of the same resources for multiple successive projects, which is more cost-efficient.

Another factor contributing to the growth of cloud computing is that developers expect to be able to use many languages and interact with predefined services. Cloud computing provides prepackaged language support, which enables the support of many more languages than the traditional do-it-yourself environment. Cloud computing can also make available shared services that provide an externally managed way of delivering frequently-used functions.

Another factor driving the adoption of cloud computing is that developers want to be able to add more resources to a specific application (*scaling up*, or *vertical scaling*), or add more duplicate instances of an application (*scaling out*, or *horizontal scaling*) to handle increased customer load. Cloud platforms provide standardized methods to scale applications.

Developers expect the pay-as-you-go utility computing billing method that cloud provides.

1.6 Cloud and mobile computing are changing traditional IT

The slide has a red header bar with the IBM logo on the right and the text 'Powered by Redbooks' on the left. The main title 'Cloud and mobile computing are changing traditional IT' is in bold red font at the top. Below it is a bulleted list of points.

- Cloud computing is a disruptive change in the IT industry:
 - New computing model, different from traditional IT computing models
 - Enables *ubiquitous computing*
 - Mobile device access
- Demand for dynamic and responsive IT infrastructure requires new:
 - Development processes
 - Application design
 - Development tools
- Virtualization and high-speed internet connectivity are the foundation for cloud computing:
 - Software-defined machine that represents a physical machine

Figure 1-4 Cloud and mobile computing are changing traditional IT

Notes:

Cloud and mobile computing are changing traditional IT.

Cloud computing is a disruptive change in the IT industry that represents a new model for the IT infrastructure that is very different from the traditional IT computing models. Cloud computing enables ubiquitous computing, where computing is made to appear anytime and everywhere, using any device, in any location, and in any format. The surge of mobile devices is greatly contributing to this model.

This new model demands a dynamic and responsive IT infrastructure due to short application lifecycles. To support this model, new development processes, application design, and development tools are required.

Virtualization and high-speed internet connectivity provided the foundation that enables cloud computing. Virtualization is key to cloud computing. It is the enabling technology that allows the creation of an intelligent abstraction layer that hides the complexity of underlying hardware or software. Virtualization provides the ability to represent physical hardware with software-defined and managed systems.

1.7 Cloud service models

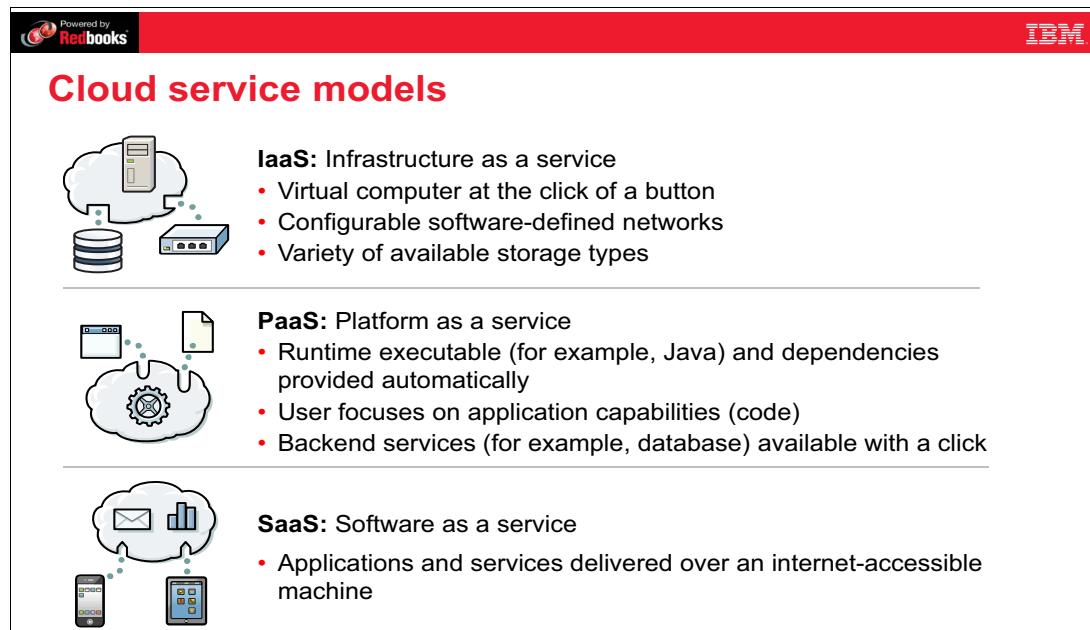


Figure 1-5 Cloud service models

Notes:

In the infrastructure as a service model (IaaS) cloud service model, a set of physical assets such as servers, network devices, and storage disks, are offered as reserved and privately accessible to consumers. The services in this model support application infrastructure. IBM SoftLayer® is an example of an IaaS cloud service model.

Platform as a service (PaaS) is a cloud service model in which application framework and runtime is a self-service, shared, virtualized entity. The goal of PaaS is to enable the developer or team to focus on the application business functions, code, and data, rather than worrying about infrastructure. IBM Bluemix is an example of the PaaS cloud service model.

Most everyday web users are familiar with software as a service (SaaS), although they might not know it. Applications in the SaaS model are provided on-demand to users through the internet, as opposed to desktop applications. Examples of SaaS applications include Salesforce.com, Google Apps, IBM Cloud for SAP Applications, and Facebook.

1.8 Infrastructure as a service architecture

Powered by
Redbooks

IBM

Infrastructure as a service (IaaS) architecture

- An infrastructure provider (IP) makes either a physical or virtual (through a *hypervisor*) computing infrastructure available “as a service.”



- Manages a pool of shared (sometimes isolated) computing resources.
- Uses virtualization to assign and dynamically resize customer resources according to their requirements.
- Customers use internet-accessible endpoints to rent computing power, memory, data storage, and networking resources.

Figure 1-6 Infrastructure as a service (IaaS) architecture

Notes:

Infrastructure as a service (IaaS) is a way of delivering cloud computing infrastructure (including servers, storage, network, and operating systems) on-demand and by using self-service tools. Rather than purchasing servers, software, data center space, or network equipment, organizations instead buy or reserve these resources through an application or API that automatically provisions or reserves the resources and makes them available.

IaaS offerings are built on top of a standardized, secure, and scalable infrastructure. Virtualizing the hardware is performed by a program known as *hypervisor*. A hypervisor manages *virtual machines* or *virtual servers*, which are multiple operating system instances that are running on a specific physical machine. Each operating system appears to have the host's processor, memory, and other resources all to itself, but in reality the hypervisor is controlling and provisioning access.

Finally, self-service is an important attribute of IaaS. Users do not have to contact support to perform common tasks, such as creating their own server and setting up basic networking.

1.9 IBM SoftLayer: IaaS offering from IBM

The screenshot shows the IBM SoftLayer interface. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below it, the main title 'IBM SoftLayer: IaaS offering from IBM' is displayed in red. The interface includes a navigation bar with 'IBM Bluemix Infrastructure', 'Search', 'Catalog', 'Support', and 'Account'. On the left, there's an 'Account Summary' section showing a current balance of '\$0.00' and an estimated next balance of '\$247.5'. Below this is an 'Order' section with icons for Devices, Storage, Network, Security, and Add-On Services. To the right, there's a 'Learn more about Bluemix!' button and a 'Scheduled Maintenance & Updates' section showing a calendar for June and July 2017.

Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Su	Mo	Tu	We	Th	Fr	Sa

Figure 1-7 IBM SoftLayer: IaaS offering from IBM

Notes:

SoftLayer is an IBM infrastructure as a service platform. IBM has more than 20 SoftLayer cloud centers, with automation and standardization across data centers, to provide a seamless global network with access to a virtualized hosting infrastructure. Many organizations favor SoftLayer because of the easy access it provides to "bare metal" servers.

In addition to virtual servers, SoftLayer offers bare metal servers, which provide the raw horsepower that some organizations require for processor-intensive and disk I/O-intensive workloads.

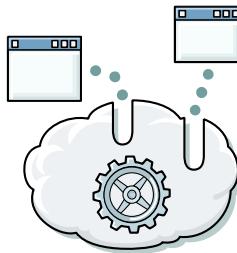
1.10 Platform as a service (PaaS) architecture

Powered by Redbooks

IBM

Platform as a service (PaaS) architecture

- Service provider (SP) supplies the runtime executables and frameworks to which new applications are automatically deployed.
- Service user is responsible for creating, updating, and maintaining the application.



- PaaS provider can provide external services, URL management, ability to scale, and built-in application monitoring.

Figure 1-8 Platform as a service (PaaS) architecture

Notes:

PaaS can be defined as a computing platform that allows the creation of cloud-based applications quickly and easily, and without the complexity of configuring the required hardware and software resources. PaaS typically entails the developer uploading the application code, or pointing to it and letting the PaaS complete the following tasks:

1. Obtain the runtime binaries and dependencies for the application.
2. Structure their application bits into the proper directory tree for containerization.
3. Provision a container (or set of containers) on which the application can run.
4. Automatically generate simple and basic networking configuration for access to the application.
5. Provide automatic and built-in monitoring of the application.
6. Allow you to update and redeploy the application with zero downtime.

PaaS typically involves sacrificing some level of fine-grained control over the application's environment to gain convenience, ease of use, and rapid deployment by using a predefined deployment process. PaaS also makes use of external services or APIs that allow rapid composition of applications by reusing pieces of infrastructure (for example, a database) that require little to no investment in setup and configuration.

PaaS also gives the developer some automatic method for scaling. For example, consider a situation where the developer wants more hardware resources dedicated to an application (*scaling up* or *vertical scaling*) or more instances of the application to handle the load (*scaling out* or *horizontal scaling*). PaaS also provides built-in application monitoring. For example, the platform sends notifications to inform developers when their application crashes.

1.11 Software as a service (SaaS) architecture

Powered by Redbooks

IBM

Software as a service (SaaS) architecture

- Service provider (SP) is responsible for creating, updating, and maintaining the runtime software and applications.



- Service user accesses the service through Internet-based interfaces.

Figure 1-9 Software as a service (SaaS) architecture

Notes:

SaaS is a delivery model that provides access to capabilities through web-based services. SaaS enables organizations to access business functionality, typically at a lower cost than paying for licensed applications. SaaS pricing is often based on a monthly fee. Because software is hosted remotely, organizations do not need to invest in new hardware to run the application. SaaS removes the need for organizations to handle installation, setup, and maintenance.

Sometimes SaaS applications are free and providers generate revenue; for example, from web ads. Alternatively, SaaS application providers generate revenue directly from the use of the service. These scenarios might sound familiar because this cloud service model is ubiquitous. If you use a tax preparation service to file your income taxes online, or use an email service to check your mail, you are familiar with this cloud service model. These types of applications are just two examples. Thousands of SaaS applications are available, and the number grows daily, primarily because of Web 2.0 technologies.

Under the SaaS model, the software provider is responsible for creating, updating, and maintaining software, including the responsibility for licensing the software. Customers usually rent the software on a per-usage basis, or buy a subscription to access it that includes a separate license for each person who uses the software.

In this model, the service user needs only to access the service and not the platform or the infrastructure on which the service is running. The service is usually accessed as a web application, or invoked by using REST or other web-based APIs.

1.12 Split of provider-side and consumer-side responsibilities

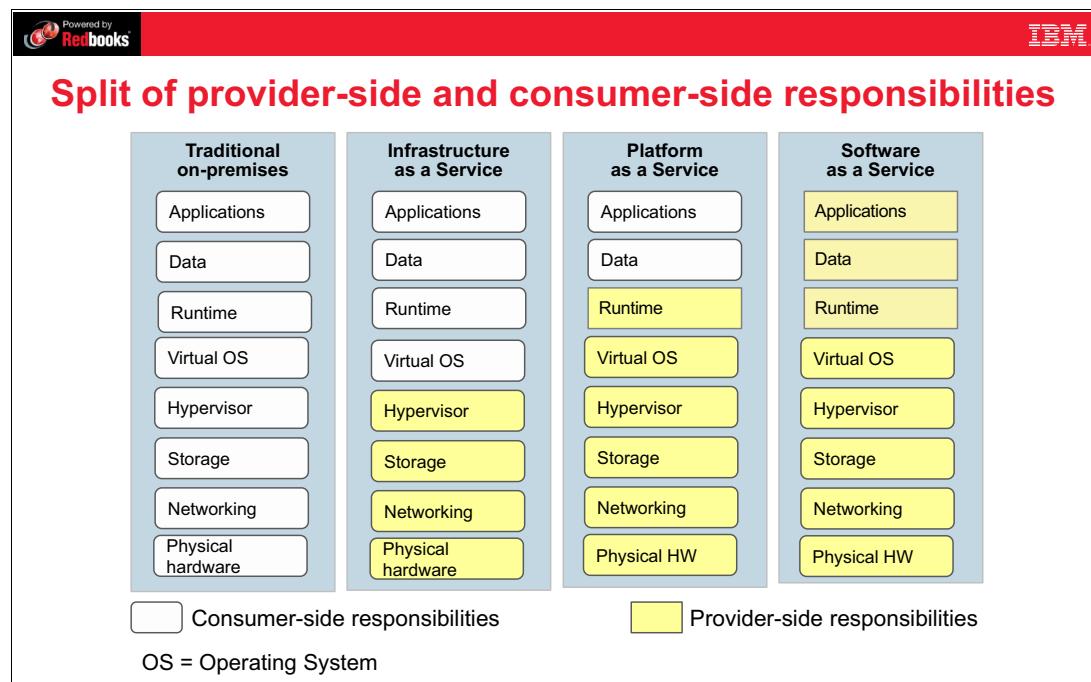


Figure 1-10 Split of provider-side and consumer-side responsibilities

Notes:

Figure 1-10 shows the split between the provider and consumer-side responsibilities when dealing with on-premises or “as a service” scenarios.

Typically, the cost reduces as you move to the right in the scenarios that are shown in Figure 1-10; however, the flexibility reduces as well. Organizations or departments within an organization make their own cost-based decision about which delivery model to use for individual applications or projects. Most enterprises end up using some combination of all of the models that are shown in Figure 1-10.

Traditional on-premises environments might not include a hypervisor. Some enterprises use their own internally-managed hypervisor to be more efficient. When they do have a hypervisor, they manage the storage, networking, and physical network on which the hypervisor runs.

1.13 Cloud computing: Benefits for developers



Cloud computing: Benefits for developers

- Readily available sandbox and production environments
 - Free trials offered with most products
 - Pre-built templates and examples help get started fast
 - Smaller learning curve to understand application lifecycle
 - Environment setup in minutes instead of days
- Choices
 - Programming languages and frameworks
 - Services
 - APIs
- Integrated development and debugging
 - The new model is to integrate development and operations teams into *devops*
 - Build engine for compilation and testing

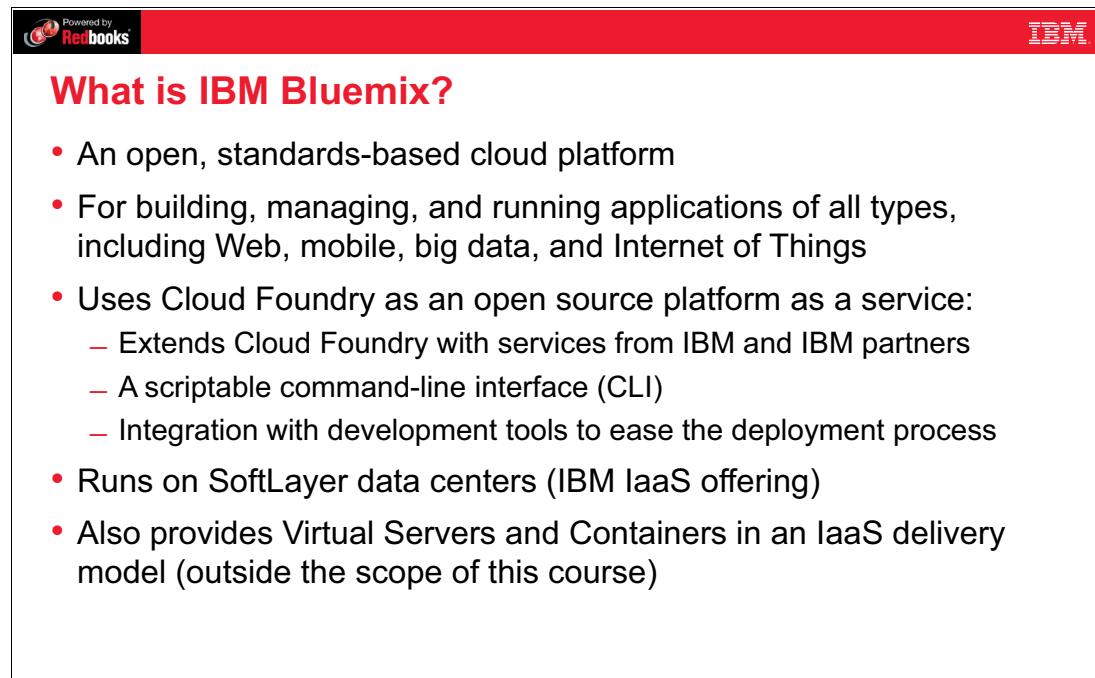
Figure 1-11 Cloud computing: Benefits for developers

Notes:

Cloud computing brings the following benefits to developers:

- ▶ Provides readily available sandbox and production environments. These environments offer the following capabilities that are attractive for developers:
 - Free trials that are offered with most products
 - Pre-built templates and examples that help developers to get started fast
 - Smaller learning curve to understand application lifecycle
 - The environment to run an application is set up in minutes instead of days
- ▶ Brings a wide range of choices to developers in the following areas:
 - Programming languages and frameworks
 - Services
 - APIs
- ▶ Facilitates integrated development, test, and debugging:
 - New model is to integrate development and operations teams into DevOps
 - Build engine for compilation and testing

1.14 What is IBM Bluemix?



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "What is IBM Bluemix?" is in red. Below it is a bulleted list of features:

- An open, standards-based cloud platform
- For building, managing, and running applications of all types, including Web, mobile, big data, and Internet of Things
- Uses Cloud Foundry as an open source platform as a service:
 - Extends Cloud Foundry with services from IBM and IBM partners
 - A scriptable command-line interface (CLI)
 - Integration with development tools to ease the deployment process
- Runs on SoftLayer data centers (IBM IaaS offering)
- Also provides Virtual Servers and Containers in an IaaS delivery model (outside the scope of this course)

Figure 1-12 What is IBM Bluemix?

Notes:

IBM Bluemix is the IBM open cloud platform that provides mobile and web developers access to IBM software for integration, security, transaction, and other key functions, and software from business partners.

Cloud Foundry is an open platform as a service offering that provides a choice of clouds, frameworks, and application services. Cloud Foundry includes a scriptable command-line interface (CLI) and integration with development tools to ease the deployment process.

IBM Bluemix is deployed on IBM SoftLayer data centers.

1.15 IBM Bluemix ecosystem

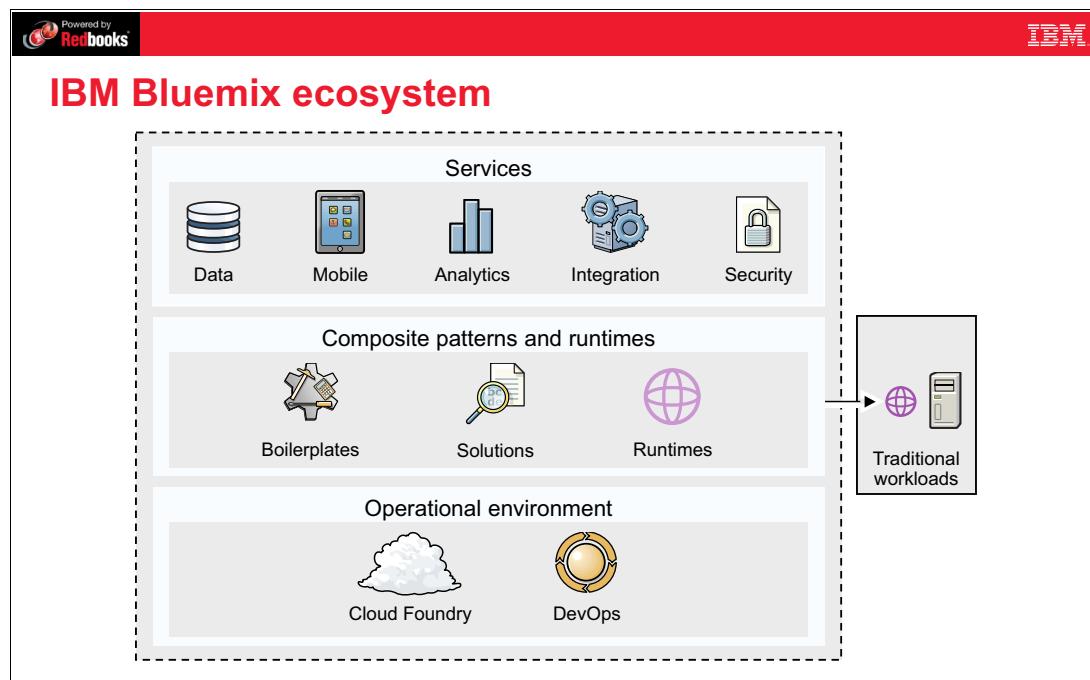


Figure 1-13 IBM Bluemix ecosystem

Notes:

The IBM Bluemix environment is an open ecosystem of services, runtimes, and boilerplates. A catalog of selectable services makes it easy for developers to bind services, such as databases, mobile support, analytics, and security, to their applications. Integration services allow applications to access traditional workloads that are running in the organization's on-premises environment.

Boilerplates are predefined, preconfigured sets of a runtime and one or more services that work together to show an example of how to compose services into a PaaS-deployed application.

The operational environment is composed of Cloud Foundry and DevOps Services. Cloud Foundry provides the monitoring, deployment, and logging tools. DevOps Services provides an online code editor, a build pipeline, and a version control system.

Solutions show how to weave these various offerings together to create your application and improve the way that you do business.

1.16 Bluemix architectural overview

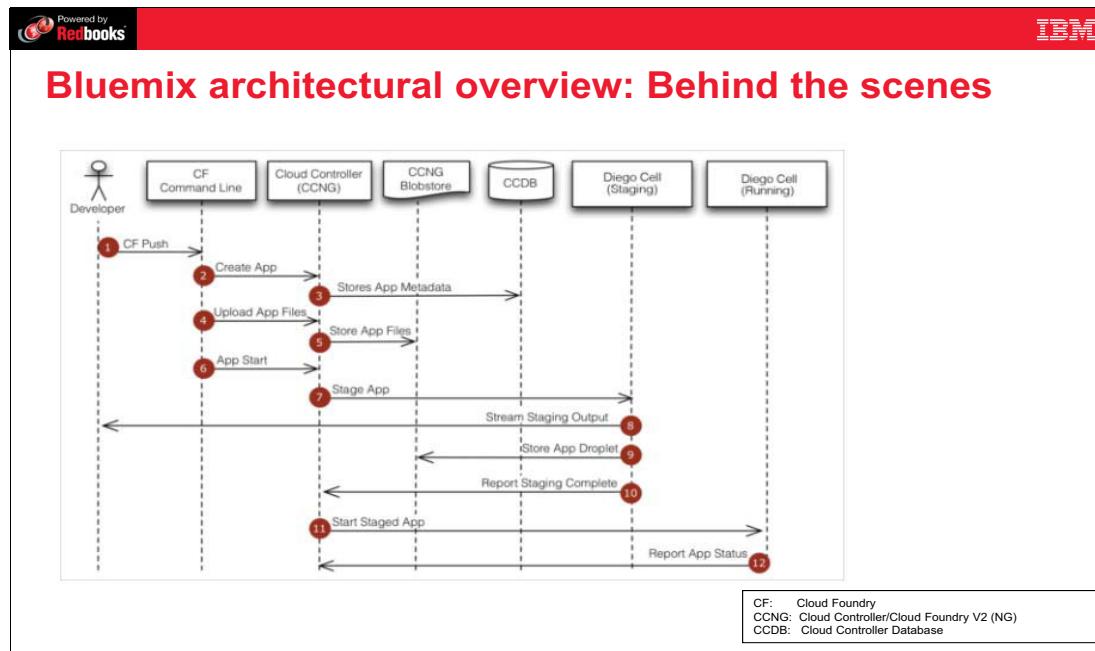


Figure 1-14 Bluemix architectural overview

Notes:

The IBM Bluemix Architecture is based on Cloud Foundry. Cloud Foundry uses the Diego architecture to manage the application life-cycle for deploying and starting the application on Cloud Foundry. The following notes correspond to the numbers that are shown in Figure 1-14:

1. As a developer, you interact with IBM Bluemix by using your web browser or the CLI.
2. When you push an application to Cloud Foundry, your requests are sent to the Cloud Controller, which is responsible for managing the lifecycle of applications. The CF CLI submits a request to the Cloud Controller to create a record for the application.
3. The Cloud Controller stores the application metadata in the Cloud Controller database (CCDB). Application metadata includes the application name, number of instances that the user specified, and the buildpack and other information about the application.
4. CF send a request to the Cloud Controller to upload all application files.
5. The Cloud Controller stores the application package in the blobstore.
6. The CF CLI issues an application start command.
7. The Cloud Controller issues a staging request to Diego, which then schedules a Cell to run the staging task. The task downloads buildpacks and, if present, buildpack cache of the app. It then uses the buildpack to build the droplet. The task uses the instructions in the buildpack to stage the application.
8. The Diego Cell streams the output of the staging process so the developer can troubleshoot application staging problems.
9. The task packages the resulting staged application into an archive called a “droplet” and the Diego Cell stores it in the blobstore. The task also uploads the buildpack cache to the blobstore for use the next time the application is staged.

- 10.The Diego cell reports to the Cloud Controller that staging is complete. Staging must complete within 15 minutes or the staging is considered failed.
- 11.Diego schedules the application as a long-running process on one or more Diego Cells.
- 12.The Diego Cells report the status of the application to the Cloud Controller and the output continues streaming to the developer.

1.17 Bluemix: Choice of runtimes

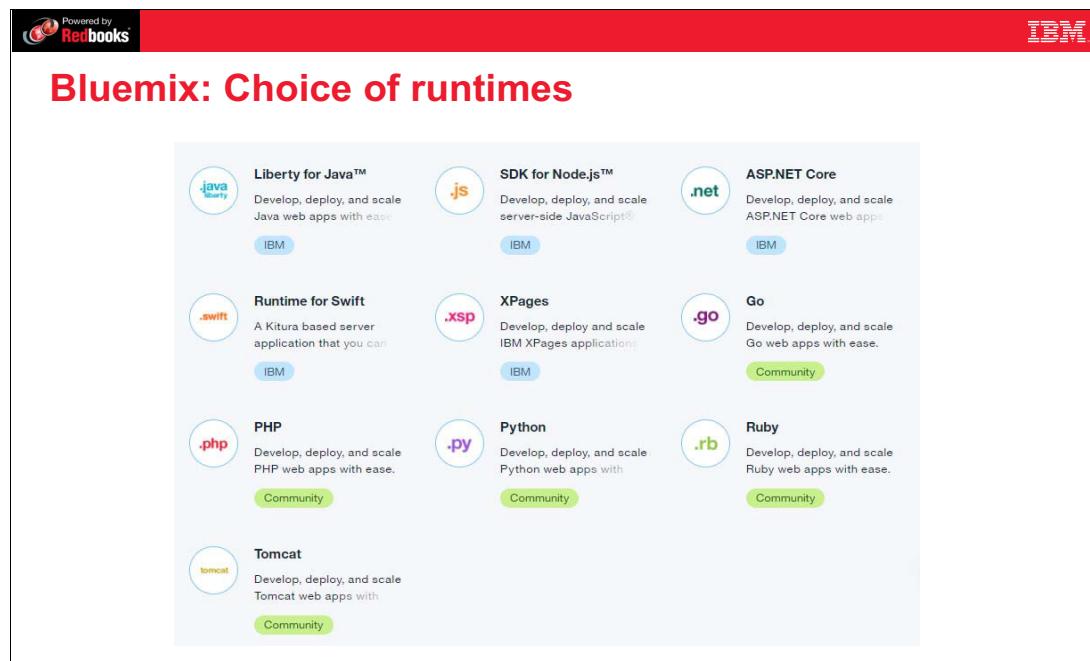


Figure 1-15 Bluemix: Choice of runtimes

Notes:

With IBM Bluemix, developers are given a choice of runtimes on which to run their applications. The IBM runtimes include Liberty for Java and the SDK for Node.js. The following runtimes are available for Java:

- ▶ Tomcat
Tomcat is an open source Java web application server.
- ▶ Liberty for Java
IBM WebSphere Liberty is another Java EE application server. Liberty can deploy any Tomcat application, but it also offers support for many more Java web features; for example, Message Beans and JMX.

Other runtimes are available and supported through the developer community forums. Runtimes are provided through the use of a buildpack, which is a set of scripts that perform the job of packaging your application and any of its dependencies into the available droplets.

Note: IBM Bluemix and Cloud Foundry support more runtimes through the *Community Buildpacks*. This open-source community has written buildpacks for virtually every language. For more information, see *Using community buildpacks*, which is available at the following website:

<https://console.ng.bluemix.net/docs/cfapps/byob.html>

1.18 Bluemix: Services

The screenshot shows the 'Bluemix: Services' catalog page. At the top left is the Redbooks logo with the text 'Powered by Redbooks'. At the top right is the IBM logo. The main title 'Bluemix: Services' is centered at the top in red. Below the title are nine service cards arranged in three rows of three. Each card includes a circular icon, the service name, a brief description, and an 'IBM' badge.

Service	Description	Status
Apache Spark	IBM Analytics for Apache Spark for Bluemix.	IBM
BigInsights for Apache Hadoop	Provision managed Apache Hadoop and Spark cluster.	IBM
BigInsights for Apache Hadoop (Subscription)	Provision managed bare metal Apache Hadoop	IBM
Cloudant NoSQL DB	Cloudant NoSQL DB is a fully managed data layer	IBM
Compose Enterprise	IBM Compose Enterprise is a service which provides	IBM
Compose for Elasticsearch	Elasticsearch combines the power of a full text search	IBM
Compose for etcd	etcd is a key/value store developers can use to hold th	IBM Beta
Compose for MongoDB	MongoDB with its powerful indexing and querying,	IBM
Compose for MySQL	MySQL is probably the most popular open source relation	IBM Beta

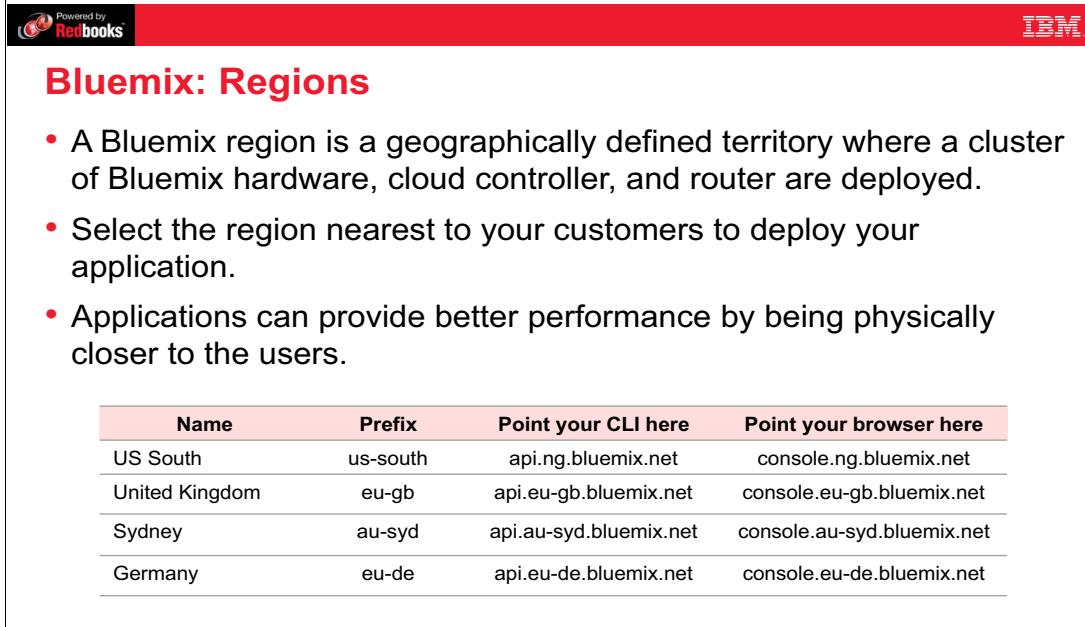
Figure 1-16 Bluemix: Services

Notes:

IBM Bluemix services provide the building blocks for delivering great applications. Too many services are available in IBM Bluemix to show in one figure. Figure 1-16 shows only the services that are available for Mobile developers. IBM Bluemix provides a broad range of pre-built services that can be used when assembling your application. The following service categories are available:

- ▶ Data & Analytics
- ▶ Watson
- ▶ Internet of Things
- ▶ APIs
- ▶ Network
- ▶ Storage
- ▶ Security
- ▶ DevOps
- ▶ Application Services
- ▶ Integrate

1.19 Bluemix: Regions



The slide is titled "Bluemix: Regions" in red. It features a bulleted list of three points about Bluemix regions. Below the list is a table showing four regions with their names, prefixes, CLI URLs, and browser URLs.

Name	Prefix	Point your CLI here	Point your browser here
US South	us-south	api.ng.bluemix.net	console.ng.bluemix.net
United Kingdom	eu-gb	api.eu-gb.bluemix.net	console.eu-gb.bluemix.net
Sydney	au-syd	api.au-syd.bluemix.net	console.au-syd.bluemix.net
Germany	eu-de	api.eu-de.bluemix.net	console.eu-de.bluemix.net

Figure 1-17 Bluemix: Regions

Notes:

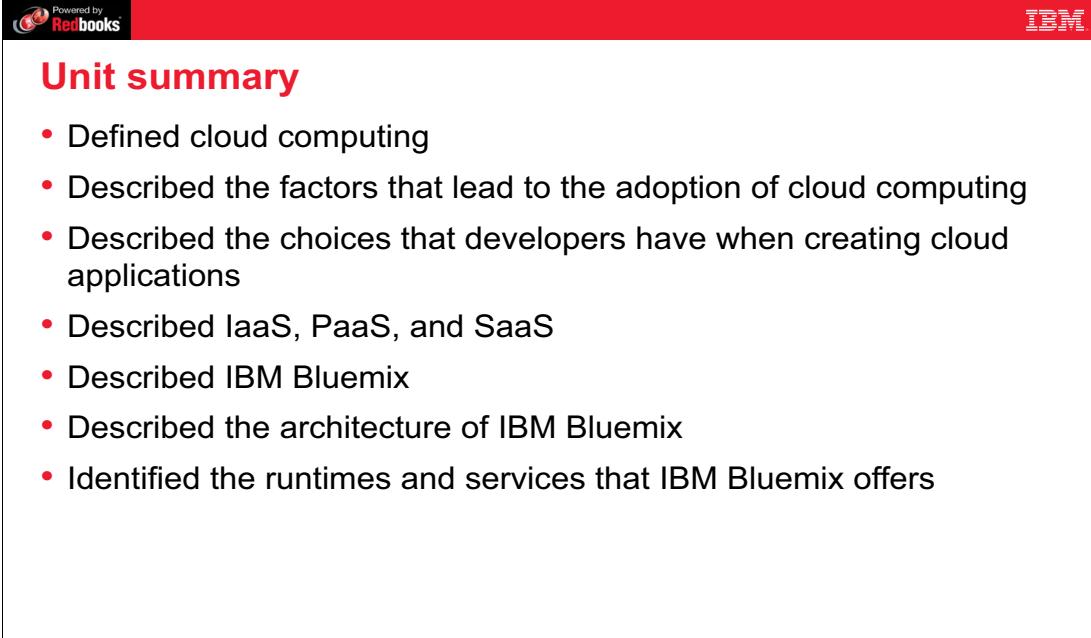
An IBM Bluemix region is a defined geographical territory to which you can deploy your applications. Select the region that is nearest to your users, and deploy your applications to this region to achieve low application latency.

Different countries with different data security requirements might require you to run your IBM Bluemix application in a different region. Running your application in multiple regions also helps make your application highly available. If your application fails in one region, it is still available from another region.

You can switch between IBM Bluemix regions easily within the IBM Bluemix web interface. You can also browse IBM Bluemix directly for each region by using the URLs that are listed in the column “Point your browser here” in Figure 1-17. You can point your command-line client to a specific IBM Bluemix region by using the URLs that are listed under the column “Point your CLI here” in Figure 1-17.

Note: Not all IBM Bluemix services are available in all regions.

1.20 Unit summary



The slide has a red header bar. On the left of the bar is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top left of this area is the title "Unit summary" in bold red font. Below the title is a bulleted list of nine items, all in black text.

- Defined cloud computing
- Described the factors that lead to the adoption of cloud computing
- Described the choices that developers have when creating cloud applications
- Described IaaS, PaaS, and SaaS
- Described IBM Bluemix
- Described the architecture of IBM Bluemix
- Identified the runtimes and services that IBM Bluemix offers

Figure 1-18 Unit summary

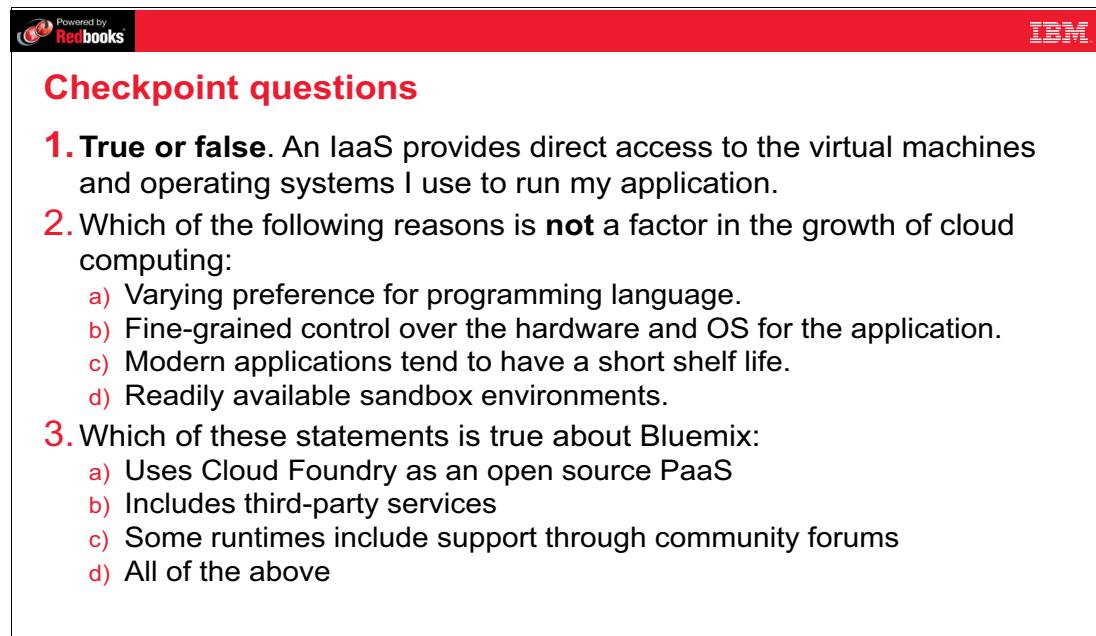
Notes:

In this unit, we defined cloud computing and described:

- ▶ The factors that lead to the adoption of cloud computing
- ▶ The choices that developers have when creating cloud applications
- ▶ IaaS, PaaS, and SaaS
- ▶ IBM Bluemix
- ▶ The architecture of IBM Bluemix

We also identified the runtimes and services that IBM Bluemix offers.

1.21 Checkpoint questions



The slide is titled "Checkpoint questions" in red text. It features the Redbooks logo and the IBM logo at the top. The content consists of three numbered questions:

- 1. True or false.** An IaaS provides direct access to the virtual machines and operating systems I use to run my application.
- 2. Which of the following reasons is **not** a factor in the growth of cloud computing:**
 - a) Varying preference for programming language.
 - b) Fine-grained control over the hardware and OS for the application.
 - c) Modern applications tend to have a short shelf life.
 - d) Readily available sandbox environments.
- 3. Which of these statements is true about Bluemix:**
 - a) Uses Cloud Foundry as an open source PaaS
 - b) Includes third-party services
 - c) Some runtimes include support through community forums
 - d) All of the above

Figure 1-19 Checkpoint questions

1.22 Checkpoint answers



Powered by
Redbooks

IBM

Checkpoint answers

1. True. Developers choose PaaS or SaaS precisely when they do not want to interact directly with operating systems and virtual hardware. IaaS allows low-level direct access to the virtual hardware.

2. b.

3. d.

Figure 1-20 Checkpoint answers



Getting started with IBM Bluemix

This unit covers the following topics:

- ▶ Understanding what you can build in IBM Bluemix
- ▶ Creating and managing an IBM Bluemix application
- ▶ IBM Bluemix users, spaces, and organizations

2.1 What you should be able to do

At the end of this unit, you should be able to describe:

- ▶ IBM Bluemix infrastructure types
- ▶ What you can build in IBM Bluemix
- ▶ How to create an application in IBM Bluemix
- ▶ The IBM Bluemix dashboard, catalog, and documentation features
- ▶ How the application route is used to test an application from the browser
- ▶ How to create services in IBM Bluemix
- ▶ How to bind services to an application in IBM Bluemix
- ▶ The environmental variables that are used with IBM Bluemix services
- ▶ IBM Bluemix organizations, domains, spaces, and users

2.2 References

For more information about the topics that are described in this unit, see this following resources:

- ▶ The IBM Bluemix documentation from the web interface provides an up-to-date reference and tutorial on building, deploying, and managing applications:
<http://bluemix.net/docs>
- ▶ IBM Bluemix overview:
<https://console.ng.bluemix.net/docs/overview/index.html>
- ▶ Join the IBM Bluemix developers community to view videos, tutorials, and forums:
<https://developer.ibm.com/bluemix/>
- ▶ Browse through a list of IBM Bluemix solutions and sample applications:
<http://bluemix.net/solutions>

2.3 Getting started: Creating an IBM Bluemix account

The screenshot shows a web page with a red header bar. On the left is the Redbooks logo, and on the right is the IBM logo. The main title 'Getting started: Creating an IBM Bluemix account' is displayed in large red text. Below the title, there is a numbered list of steps:

1. Open the IBM Bluemix page:
<http://bluemix.net>
2. Click Sign up to create a Bluemix account:
 - If you do not have an IBM ID, complete the personal information to create one.
 - Creating a Bluemix ID automatically creates an IBM ID for you.
 - There is no charge to create an IBM ID or a Bluemix trial account.

On the right side of the page, there is a form for account creation. It includes fields for Email*, First Name*, Last Name*, Company, Country or Region* (set to Afghanistan), and Phone Number*. There is also a link 'Already have a Bluemix account? Log in'.

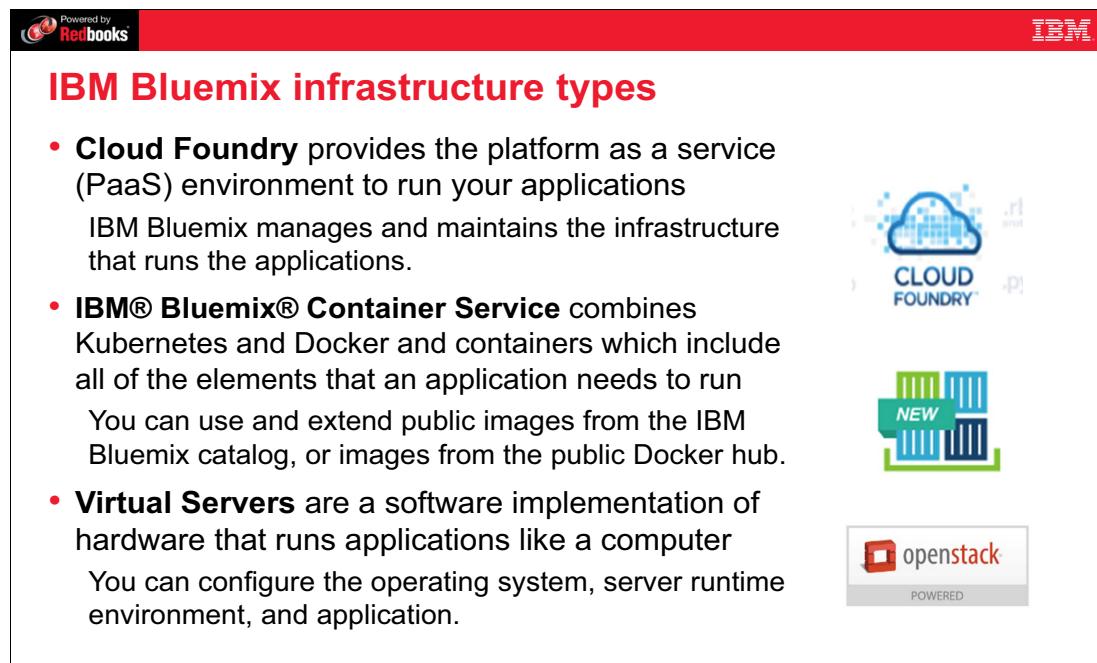
Figure 2-1 Getting started: Creating an IBM Bluemix account

Notes:

Before you work on any of the exercises in this course, you must sign up for an IBM Bluemix account. You can register for a free IBM Bluemix trial account at this website:

<https://console.bluemix.net/>

2.4 IBM Bluemix infrastructure types



The slide is titled "IBM Bluemix infrastructure types". It features three main bullet points with associated icons:

- **Cloud Foundry** provides the platform as a service (PaaS) environment to run your applications. The icon shows a blue cloud with a grid pattern.
- **IBM® Bluemix® Container Service** combines Kubernetes and Docker and containers which include all of the elements that an application needs to run. The icon shows a stack of four boxes with the word "NEW" in a green box on top.
- **Virtual Servers** are a software implementation of hardware that runs applications like a computer. The icon shows the OpenStack logo.

Figure 2-2 IBM Bluemix infrastructure types

Notes:

IBM Bluemix supports the following alternative infrastructure types:

- ▶ Cloud Foundry provides the PaaS environment to run your applications. IBM Bluemix manages and maintains the infrastructure that runs the applications.
With Cloud Foundry Apps, IBM Bluemix allows basic management capability of your application and allows it to connect to prepackaged services. Cloud Foundry boilerplates and starter apps are the simplest way to get started with IBM Bluemix.
- ▶ IBM Containers provide more fine-grained control over the computing infrastructure to run the application and services through Docker-like APIs. Docker packages software into standardized units that are called *containers*. Containers include all of the elements that an application needs to run. IBM Bluemix Container Service combines Docker and Kubernetes to deliver powerful tools, an intuitive user experience, and built-in security and isolation to automate the deployment, operation, scaling, and monitoring of containerized apps over a cluster. A Kubernetes cluster consists of one or more virtual machines.
If you want to port an image from another cloud infrastructure provider, or if you want to use a public image, consider the IBM Containers technology. Containers give you a bit more control, but still do not require you to manage an operating system or network. IBM Containers resemble the popular open source container engine Docker, but have some IBM Bluemix-specific features, such as their ability to integrate with the Cloud Foundry routers.
- ▶ Virtual Servers are software implementation of hardware that runs applications, such as a computer. You can configure the operating system, server runtime environment, and application. If you want to have control over the infrastructure down to the operating system level, consider Virtual Servers.

Most developers in the enterprise will probably use some mix of these types. The exercises in this course introduce you to the Cloud Foundry infrastructure in IBM Bluemix. As such, Containers and Virtual Servers are out of scope of this course. The focus is on Cloud Foundry Apps for the remainder of this course.

2.5 What can you build in IBM Bluemix?

The screenshot shows the IBM Bluemix dashboard. At the top, there's a navigation bar with the Redbooks logo and the IBM logo. Below the navigation bar, the main title "What can you build in IBM Bluemix?" is displayed in red. To the left of the title, there's a bulleted list describing what can be built:

- **Applications:** Programs that developers build in the IBM Bluemix Cloud Foundry environment:
 - Mobile applications
 - Web applications
- **Services:** Cloud extensions that are hosted by Bluemix:
 - Provide functionality that is ready-for-use by the application
 - Predefined services include database, messaging, push notifications for mobile apps, and elastic caching for web apps

On the right side of the dashboard, there are two sections: "All Apps (1)" and "All Services (1)".

All Apps (1)

NAME	ACTIONS
vy301-xxx-nodecloudant	(refresh) (edit) (more)

Create App +

Cloud Foundry Apps 6.375 GB/8 GB Used

All Services (1)

NAME	ACTIONS
vy301-xxx-nodecloudant-clo	(refresh) (edit) (more)

Create Service +

Services 35/40 Used

Figure 2-3 What can you build in IBM Bluemix?

Notes:

In IBM Bluemix you can build *applications*, which are the programs that developers build in the Cloud Foundry environment. You can build *mobile apps* that run outside the IBM Bluemix environment and use services to which the mobile apps are exposed. IBM Bluemix can also host application code that the developer would rather run on a back-end server in a container-based environment. *Web apps* consist of the code that is required to be run or referenced at runtime.

A *service* is a cloud extension that is hosted by IBM Bluemix. The service provides functionality that is ready-for-use by the app's running code. The predefined services that are provided by IBM Bluemix include database, messaging, push notifications for mobile apps, and elastic caching for web apps. You can create your own services in IBM Bluemix. They can be simple utilities, such as the functions you might see in a runtime library, or complex business logic that you might see in a business process modeling service or a database.

2.6 Cloud Foundry Apps, Containers, and Virtual Servers

The slide has a red header bar. On the left is the 'Powered by Redbooks' logo. On the right is the 'IBM' logo. The main title 'Cloud Foundry Apps, Containers, and Virtual Servers' is centered in red text. Below it is a bulleted list of two items.

- **Cloud Foundry Apps:** You can run code, a boilerplate, or runtime on a Cloud Foundry PaaS environment.
- **Containers and Virtual Servers** are alternative ways to manage and run applications.

Figure 2-4 Cloud Foundry Apps, Containers, and Virtual Servers

Notes:

IBM Bluemix is an implementation of IBM Open Cloud Architecture, which is based on Cloud Foundry.

Containers and Virtual Servers are alternative ways to run and manage an application. They allow more control over the underlying infrastructure and configuration, but require more work by the developer.

This course focuses on Cloud Foundry Apps.

2.7 Containers versus Virtual Servers

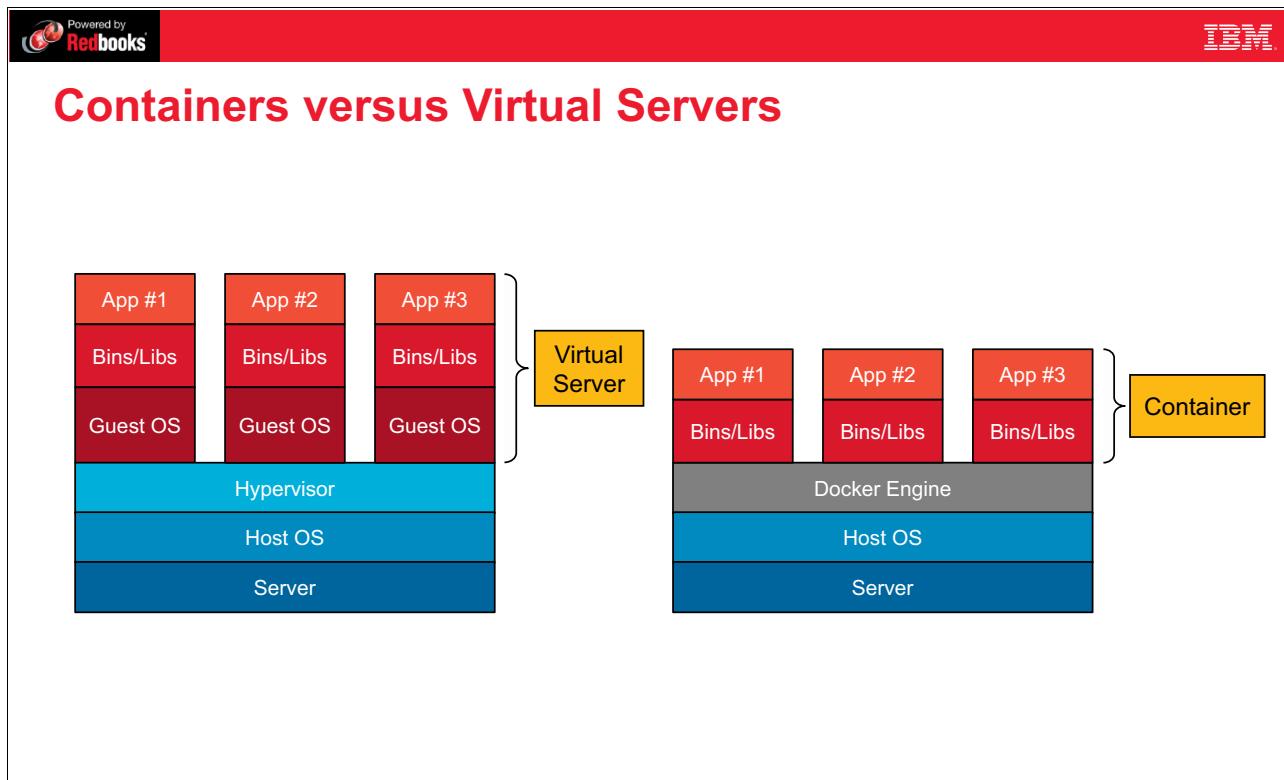


Figure 2-5 Containers versus Virtual Servers

Notes:

Containers and virtual machines (VMs) have similar objectives: To isolate an application and its dependencies into a self-contained unit that can run anywhere. They both remove the need for physical hardware, which allows for more efficient use of computing resources.

Virtual Servers (or virtual machines) are created using a hypervisor. The hypervisor virtualizes the physical hardware to create a software-defined computer that runs its own operating system. Unless special software is installed for cloud management, an operating system that is running in a VM does not have any awareness that it is running in VM. As shown in Figure 2-5, the VM packages the virtual hardware, a kernel or OS, and user space for each new VM.

Unlike a VM that provides hardware virtualization, a container provides operating system-level virtualization.

A container library, such as Docker, separates different user spaces for each container. On the surface, these spaces might resemble a VM to a user.

The main difference between containers and VMs is that containers share the operating system kernel with other containers that are running on the machine, which is not the case with VMs. Therefore, you might see a VM running Linux on a Windows server, or Windows on a Linux server.

However, you do not see a container that runs Linux on anything other than a Linux server. The containers on that Linux server are separated out user spaces on that Linux server and the hardware is not virtualized.

Data storage also is managed differently in containers. In VMs, virtual disks are created and are similar to physical disks in that after you write something to the disk, it stays there until you delete it. With containers, you can make changes to the disk and then delete your changes when done.

2.8 IBM Bluemix catalog: Infrastructure

The screenshot shows the IBM Bluemix catalog interface. On the left, a sidebar lists categories: Compute, Storage, Network, and Security. The main content area is titled "IBM Bluemix catalog: Infrastructure". It displays infrastructure services under three main categories: Compute, Storage, and Network.

- Compute:** Shows options like Bare Metal Server, Public Virtual Server (Hourly), and Public Virtual Server (Monthly).
- Storage:** Shows options like Block Storage, Cloud Object, and Content Delivery.
- Network:** Shows options like VMware Solutions, Data & Analytics, Watson, Internet of Things, APIs, Network, Storage, and Security.

Figure 2-6 IBM Bluemix catalog: Infrastructure

Notes:

You can order infrastructure services in one of the following categories:

- ▶ Compute
- ▶ Storage
- ▶ Network
- ▶ Security

2.9 IBM Bluemix catalog: Apps

The screenshot shows the IBM Bluemix Catalog: Apps interface. At the top left is the Redbooks logo with the text "Powered by Redbooks". At the top right is the IBM logo. The main title "IBM Bluemix catalog: Apps" is centered at the top. Below the title, there is a sidebar on the left with a list of categories:

- Boilerplates
- Cloud Foundry Apps
- Containers
- OpenWhisk
- Mobile

Below the sidebar, there are sections for "All Categories" and "Infrastructure". The "All Categories" section includes links for Compute, Storage, Network, Security, and Apps (which is highlighted). The "Infrastructure" section includes links for Compute, Storage, Network, Security, and Mobile.

The main content area is titled "Boilerplates" and contains several application starters:

- ASP.NET Core Cloudant Starter: Use the Cloudant NoSQL DB Service in an ASP.NET Core application.
- Internet of Things Platform Starter: Get started with IBM Watson IoT platform.
- IoT for Electronics Starter: IoT for Electronics is a integrated end-to-end...
- Java Cloudant Web Starter: Use the Cloudant NoSQL DB service with the...
- Java Workload Scheduler Web Starter: This application demonstrates how to...
- LoopBack Starter: This is a sample StrongLoop LoopBack...

Figure 2-7 IBM Bluemix catalog: Apps

Notes:

IBM Bluemix offers many options for developing applications.

2.10 IBM Bluemix catalog: Boilerplates

The screenshot shows the IBM Bluemix catalog interface. At the top, there's a red header bar with the IBM logo on the right and the Redbooks logo on the left. Below the header, the title "IBM Bluemix catalog: Boilerplates" is displayed in large red text. Underneath the title, a sub-header says "Get started with a new app, now." followed by a list of five boilerplate options:

- ASP.NET Core Cloudant Starter**: Use the Cloudant NoSQL DB Service.
- Internet of Things Platform Starter**: Get started with IBM Watson IoT platform.
- IoT for Electronics Starter**: IoT for Electronics is a integrated end-to-end solution.
- Java Cloudant Web Starter**: Use the Cloudant NoSQL DB service.
- Java Workload Scheduler Web Starter**: This application demonstrates how...
- LoopBack Starter**: This is a sample StrongLoop

- Boilerplates are packages of *templates* and *sample applications*:
 - When you create an application with boilerplates, IBM Bluemix configures the services for the application.
 - IBM Bluemix also provides you with the source code and documentation for the sample application in the boilerplate.

Figure 2-8 IBM Bluemix catalog: Boilerplates

Notes:

In IBM Bluemix, *boilerplates* are packages of templates and sample applications. When you create an app by using boilerplate, IBM Bluemix configures the services for the application. IBM Bluemix also provides you with the source code and documentation for the sample application in the boilerplate.

A boilerplate contains an app and its associated runtime environment and predefined services. You can use a boilerplate to quickly get up and running. For example, you can select the Mobile Cloud boilerplate to host mobile and web apps and accelerate development time of server-side scripts by using the mobile app template and SDK.

2.11 IBM Bluemix catalog: Cloud Foundry Apps

The screenshot shows the IBM Bluemix catalog interface for Cloud Foundry Apps. At the top, there's a red header bar with the IBM logo on the right and a 'Powered by Redbooks' logo on the left. Below the header, the title 'IBM Bluemix catalog: Cloud Foundry Apps' is displayed in large red font. A sub-header in smaller black font reads 'Deploy your app without managing underlying infrastructure.' The main content area is a grid of runtime environment cards:

- Liberty for Java™**: Develop, deploy, and scale Java. IBM runtime.
- SDK for Node.js™**: Develop, deploy, and scale serverless. IBM runtime.
- .NET Core**: Develop, deploy, and scale. IBM runtime.
- Tomcat**: Develop, deploy, and scale Tomcat. Community runtime.
- Runtime for Swift**: A Kitura based server application. IBM runtime.
- XPages**: Develop, deploy and scale IBM. IBM runtime.
- Go**: Develop, deploy, and scale Go web. Community runtime.
- Ruby**: Develop, deploy, and scale Ruby. Community runtime.
- .PHP**: Develop, deploy, and scale PHP. Community runtime.
- .Python**: Develop, deploy, and scale Python. Community runtime.

Figure 2-9 IBM Bluemix catalog: Cloud Foundry Apps

Notes:

To create your own application, select the following server runtime environment as a starting point:

- IBM packages support IBM runtime environments, such as Liberty.
- Community runtimes rely on open source and third-party packages.

A *runtime* is the set of resources that is used to run an application. IBM Bluemix provides runtime environments as containers for different types of applications. The runtime environments are integrated as buildpacks into IBM Bluemix and are automatically configured for use.

Runtimes start with a simple templated example application that you can customize to meet your needs. These runtimes do not include any services by default, but you can add and bind your own services later.

2.12 IBM Bluemix catalog: Containers

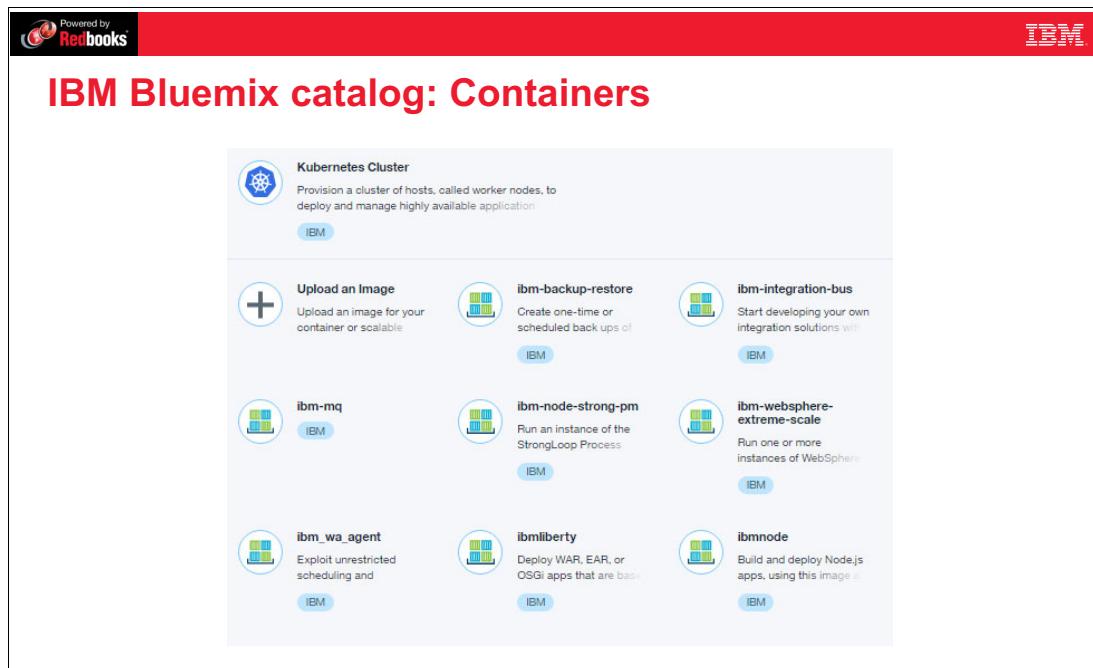


Figure 2-10 IBM Bluemix catalog: Containers

Notes:

You can create a Kubernetes cluster that consists of one or more VMs that are called *worker nodes*. Every worker node represents a compute host in which you can deploy, run, and manage containerized applications.

You can also create a container of IBM Integration Bus V10 Developer Edition, which you can use to start developing your own integration solution. Other options include IBM MQ and WebSphere eXtreme Scale Caching Server. You can also upload your container image.

2.13 IBM Bluemix catalog: OpenWhisk

The image shows the 'IBM Bluemix catalog: OpenWhisk' page. At the top left is the Redbooks logo with the text 'Powered by Redbooks'. At the top right is the IBM logo. The main title 'IBM Bluemix catalog: OpenWhisk' is in red. Below the title is a description: 'IBM Bluemix OpenWhisk is a Function-as-a-Service (FaaS) platform that executes functions in response to incoming events and costs *nothing* when not in use.' Three icons represent features: 1. 'Cost-Effective Computing' with a satellite dish and a gear. 2. 'Automatically Scale' with a satellite dish and a series of gears. 3. 'Easy Integration' with various cloud and integration icons.

Figure 2-11 IBM Bluemix catalog: OpenWhisk

Notes:

By using OpenWhisk, you can focus on writing code and building great solutions, without the hassle of setting up, configuring, or maintaining servers. You pay for only the time your code actually runs, which means no excess capacity or idle time.

Consider the following OpenWhisk basic concepts:

- ▶ Actions: A piece of code that you develop that performs one specific task. You can create the action in various runtimes, including Node.js, Python, and Swift 3.
- ▶ Triggers: Triggering the actions can be through REST APIs, or based on certain events, such as Alarm-Based Trigger, IBM Cloudant Change, and Messaging.
- ▶ Sequences: You create the flow of actions to complete one function.
- ▶ Rules: Rules link triggers to actions or sequences.

2.14 IBM Bluemix catalog: Mobile

The screenshot shows the IBM Bluemix catalog interface for mobile development. At the top, there's a red header bar with the IBM logo on the right and the Redbooks logo on the left. Below the header, the title "IBM Bluemix catalog: Mobile" is displayed in large red text. A sub-header in black text reads "IBM Bluemix catalog includes several capabilities for mobile development". The main content area is a grid of service cards:

- App ID**: Adds authentication to your apps, and host user profile information. (IBM)
- Mobile Analytics**: For mobile app developers and business stakeholders. (IBM)
- Mobile Foundation**: Comprehensive Mobile Backend for your digital apps. (IBM)
- Push Notifications**: Scalable and reliable Push Notifications service for mobile. (IBM)
- Kinetise**: Rapid development of mobile apps. With SOURCE CODE. (Third Party)
- Testdroid Cloud**: Mobile testing cloud service. (Third Party)
- Twilio**: Build apps that communicate. Integrate voice, messaging and more. (Third Party)

Figure 2-12 IBM Bluemix catalog: Mobile

Notes:

The capabilities include rapid development of mobile application by using a web editor to create a mobile application, test the mobile application by using cloud services, and gain insights into how your mobile application is being used.

2.15 IBM Bluemix catalog: Services

The screenshot shows the IBM Bluemix catalog: Services page. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below the header, the title "IBM Bluemix catalog: Services" is displayed in red. The main content area contains six service cards arranged in two rows of three. Each card includes an icon, the service name, a brief description, and an "IBM" button.

Service	Description	Provider
Conversation	Add a natural language interface to your application.	IBM
Discovery	Add a cognitive search and content analytics engine to	IBM
Document Conversion	Converts a HTML, PDF, or Microsoft Word™ document	IBM
Language Translator	Translate text from one language to another for spec	IBM
Natural Language Classifier	Natural Language Classifier performs natural language	IBM
Natural Language Understanding	Analyze text to extract meta-data from content such as	IBM

- Services are extensions to the cloud environment that IBM Bluemix hosts and manages
- The predefined services that are provided by IBM Bluemix include NoSQL and SQL databases, the ability to send push notifications to your mobile application, and automated language translation

Figure 2-13 IBM Bluemix catalog: Services

Notes:

Services are extensions to the cloud environment that IBM Bluemix hosts and manages. The predefined services that are provided by IBM Bluemix include NoSQL and SQL databases, the ability to send push notifications to your mobile app, and automated language translation.

You can add services to your IBM Bluemix application from the IBM Bluemix catalog. Services provide a predefined endpoint that you can access from your application to use the predefined functionality of that service.

The infrastructure for services are managed by IBM Bluemix, and your app needs to focus on the provided endpoint only. You can bind more than one app to a service to share services between your apps.

Many IBM Bluemix services are available and more are being added. Figure 2-13 shows a sample of the services that are available for IBM Watson.

2.16 Creating an IBM Bluemix application (1 of 3)

The screenshot shows the IBM Bluemix application catalog interface. At the top, there's a red header bar with the IBM logo on the right and a "Create App" button with a plus sign on the left. Below the header, the title "Creating an IBM Bluemix application (1 of 3)" is displayed in red. The main content area contains a numbered list and several application type cards:

1. To create an IBM Bluemix application, click **Create App** from the dashboard.
2. From the Catalog, choose your application type.

The application types listed are:

- Liberty for Java™**: Develop, deploy, and scale Java web apps with ease. IBM
- SDK for Node.js™**: Develop, deploy, and scale server-side JavaScript® apps. IBM
- .NET Core**: Develop, deploy, and scale ASP.NET Core web apps with ease. IBM
- Runtime for Swift**: A Kitura based server application that you can use. IBM
- XPages**: Develop, deploy and scale IBM XPages applications with ease. IBM
- Go**: Develop, deploy, and scale Go web apps with ease. Community

Figure 2-14 Creating an IBM Bluemix application (1 of 3)

Notes:

From the catalog, you can choose one of the following options for the application type:

- ▶ Boilerplates
- ▶ Cloud Foundry Apps
- ▶ Containers
- ▶ OpenWhisk
- ▶ Mobile

2.17 Creating an IBM Bluemix application (2 of 3)

The screenshot shows the 'Create a Cloud Foundry App' page for the IBM Bluemix Node.js SDK. At the top, there's a red banner with the Redbooks logo and the text 'Creating an IBM Bluemix application (2 of 3)'. Below the banner, a sub-header says '3. Enter your App name and Host name and choose Pricing Plan.' The main form has fields for 'App name' (with placeholder 'Enter a unique name') and 'Host name' (with placeholder 'Enter a unique name' and a dropdown for 'Domain' set to 'mybluemix.net'). To the left, there's a sidebar for the 'SDK for Node.js™' with details like 'Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.' Below the sidebar are buttons for 'View Docs', 'VERSION 3.x', 'TYPE Application', and 'REGION US South'. On the right, a 'Pricing Plans' section shows a table with one row:

PLAN	FEATURES	PRICING
<input checked="" type="checkbox"/> Default	Run one or more apps free for 30 days (375 GB-hours free).	\$0.0749 USD/GB-Hour

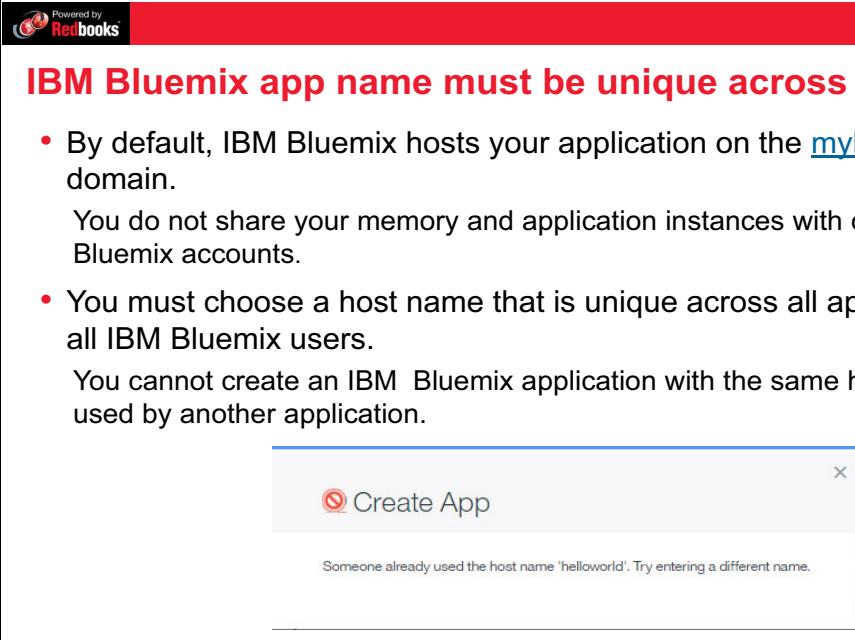
A note below the table states: 'This is a service plan for the IBM Bluemix Platform runtime.' At the bottom of the page are 'Terms' and 'IBM' buttons.

Figure 2-15 Creating an IBM Bluemix application (2 of 3)

Notes:

Enter your application name, hostname, and select the pricing plan for this application.

2.18 IBM Bluemix app name must be unique across domain



The screenshot shows a 'Create App' dialog box. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title 'IBM Bluemix app name must be unique across domain' is displayed in red. The main content area contains two bullet points: one about the default host name and another about choosing a unique host name. Under the second point, there's a note that you cannot create an application with a host name already used by another. Below this note, a modal dialog box is open, showing a red exclamation mark icon and the text 'Create App'. Inside the modal, a message says 'Someone already used the host name 'helloworld''. Try entering a different name.'

Figure 2-16 IBM Bluemix application name must be unique across domain

Notes:

By default, IBM Bluemix hosts your application on the `mybluemix.net` domain. You do not share your memory and application instances with other IBM Bluemix accounts. You must choose a host name that is unique across all applications from all IBM Bluemix users. You cannot create an IBM Bluemix application with the same host name that is used by another application. The exercises in this course manage this issue by having each team use a randomly-generated key.

Your app's name is different than its host name and must be unique only within your organization. You can have the same name as another user if you do not have the same host name as another user.

By default, IBM Bluemix sets your hostname and your app name to be the same. The app name is for your reference to use only in scripts, commands to the CLI, and to find your app within the IBM Bluemix UI. The application's host name becomes part of the application's route, which is how users access your app over the internet.

2.19 Creating an IBM Bluemix application (3 of 3)

The screenshot shows a step titled "Creating an IBM Bluemix application (3 of 3)". Step 4, "Select your type of development style.", is highlighted in red. Three options are listed:

- Eclipse tools for Bluemix**: Develop, integrate, and push apps to Bluemix using Eclipse. It includes an icon of the Eclipse logo.
- Command line interface**: Run your code locally. Manually push to Bluemix. It includes an icon of a terminal window with a greater-than sign.
- Continuous delivery**: Build, test, and deliver by using DevOps practices. It includes an icon of a person in a gear.

Figure 2-17 Creating an IBM Bluemix application (3 of 3)

Notes:

Next, select your development style. You can perform the following tasks:

- ▶ Download the Eclipse Tools for Bluemix, which allows you to use the Bluemix APIs from the Eclipse integrated development environment.
- ▶ Edit your app in your preferred text editor and use the Cloud Foundry command line to deploy.
- ▶ Use Bluemix DevOps Services to deploy your application with a completely web-based approach.

2.20 IBM Bluemix Dashboard

The screenshot shows the IBM Bluemix Dashboard interface. At the top, there's a red header bar with the IBM logo on the right and a "Powered by Redbooks" badge on the left. Below the header, the title "IBM Bluemix Dashboard" is displayed in red. To the right of the title, there's a "Create App" button with a plus sign. The main content area is divided into two sections: "All Apps (1)" and "All Services (1)".

All Apps (1)

NAME	STATE	ACTIONS
vy301-xxx-nodejs	Running	

All Services (1)

NAME	ACTIONS
Conversation	

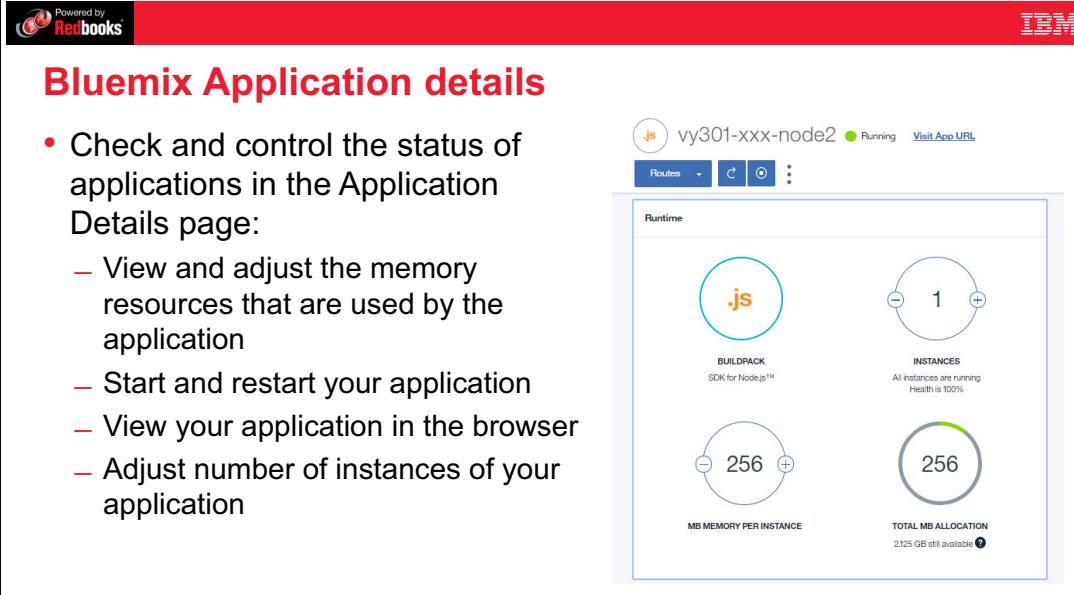
Figure 2-18 IBM Bluemix Dashboard

Notes:

IBM Bluemix Dashboard shows you the amount of memory that you are using in your organization and the number of services that you are using in your space.

If you find that your organization is running low on memory, you can stop an application without deleting it and start it again later when needed. However, if you are running out of available services, you must delete some services to free up space.

2.21 IBM Bluemix Application Details page



The screenshot shows the 'Bluemix Application details' page for an application named 'vy301-xxx-node2'. The application is listed as 'Running' with a green status indicator. Below the application name, there is a list of features:

- Check and control the status of applications in the Application Details page:
 - View and adjust the memory resources that are used by the application
 - Start and restart your application
 - View your application in the browser
 - Adjust number of instances of your application

The right side of the screen displays a 'Runtime' section with the following data:

Buildpack	Instances	Memory Allocation
SDK for Node.js™	1	256 MB
	All instances are running Health is 100%	TOTAL MB ALLOCATION 2125 GB still available

Figure 2-19 IBM Bluemix Application Details page

Notes:

To open the application details, click the application name in the IBM Bluemix Dashboard.

2.22 Testing applications through the application route

The screenshot shows the IBM Bluemix dashboard with a card for an application named "vy301-xxx-nodecloud...". The card indicates the application is "Running". To the right of the application name is a callout bubble with the text "Click to open the app URL in the browser" and a red arrow pointing to the "Open URL" icon (a browser icon) on the card.

- The *application route* is the entry point for users into your application in IBM Bluemix. You can host a REST service, web application, or web page through this route.
- To view the application route, click the **Open URL** icon:
 - You can access the application route for a running application only.
 - The application route must be unique across all applications in the domain.

Figure 2-20 Testing applications through the application route

Notes:

The application route is the entry point for users into your application in IBM Bluemix. You can host a REST service, web application, or web page through this route.

To view the application route, click the **Open URL** icon. You can access the application route for a running application only. The application route must be unique across all applications in the domain.

2.23 Adding an IBM Bluemix service

The screenshot shows the IBM Bluemix Catalog interface. On the left, there is a sidebar with instructions for adding a service:

- From the Dashboard, click **Create Service**.
- Select a service from the Catalog.
- Enter a Service name and a Pricing Plan for the service.

Below these steps, it says: "You can choose to bind the service to an application or leave it unbound."

The main catalog page displays the "ClearDB Managed MySQL Database" service. It includes a brief description, a "Service name:" input field containing "ClearDB Managed MySQL Database-s3", and a "Pricing Plans" section. The "Pricing Plans" table has three columns: PLAN, FEATURES, and PRICING. One plan is listed:

PLAN	FEATURES	PRICING
✓ CB5	Price: FREE! DB size: up to 10 MB Connections: 5 I/O Performance: Very Low	Free

At the bottom right of the catalog page is a blue "Create" button.

Figure 2-21 Add an IBM Bluemix service

Notes:

You can add an IBM Bluemix service to your application by clicking **Create Service** and then, selecting a service from the IBM Bluemix Catalog. You must then enter a name for the service and select a pricing plan for your service. Plans allow you to choose different “sizes” for the service, such as how much power you put behind it. For example, with some services you can choose if you share hardware or use dedicated hardware.

You can bind the service to your application from this page, or leave the service unbound. If you leave the service unbound, IBM Bluemix creates and provides credentials for accessing the service, which you can access from within the service dashboard. If you bind the service to an application, the credentials are in the application’s environment variables.

2.24 Binding a service to an application

The screenshot shows the IBM Bluemix Cloud Foundry Apps interface. At the top, there's a red banner with the text "Binding a service to an application". Below the banner, on the left, is a sidebar with links: Getting started, Overview, Runtime, **Connections**, Logs, Monitoring, and API Management. On the right, under the heading "Connect existing service", there's a list of services. One service, "ClearDB Managed MySQL Data", is shown with a "Select compatible service to connect to vy301-xxx-nodecloudant" message and a "Connect" button. The top navigation bar includes "Ahmed Azraq's Account | US South : aazraq@eg.ibm.com : cloud_app_dev", "IBM Bluemix", "Cloud Foundry Apps", "Catalog", "Support", and "Manage". The IBM logo is in the top right corner.

Figure 2-22 Bind a service to an application

Notes:

To bind the service to your application, go to your application details page and select **Connect new** or **Connect existing**:

- ▶ By using the Connect new option, you can create an instance of the service from the Bluemix Catalog, then bind it.
- ▶ The Connect Existing option allows you to choose one of the services that was created earlier to bind it to your application. IBM Bluemix lists the services that you created in your space that are compatible with your service and runtime environment. Some services are specific to certain runtimes, such as a Java Runtime monitor, and might not be applicable or indeed usable from your application.

When you bind a service to an application, IBM Bluemix creates a set of credentials to access the service and places the connection information for this service in the application environment variable, which is called VCAP_SERVICES.

2.25 IBM Bluemix environment variables

The screenshot shows the IBM Bluemix interface. At the top, there's a navigation bar with the Redbooks logo, the text "Powered by Redbooks", and the IBM logo. Below the navigation bar, the title "IBM Bluemix environment variables" is displayed in red. On the left, there's a sidebar with links: "Getting started", "Overview", "Runtime" (which is selected), "Connections", "Logs", "Monitoring", and "API Management". The main content area shows a table of environment variables. The first row is highlighted in yellow and contains the variable name "VCAP_SERVICES" and its value, which is a JSON object:

```
{ "cloudantNoSQLDB": [ { "credentials": { "username": "77b8dcec-a2be-4f07-ba3b-1dc993690b5e-bluemix", "password": "c93e1f6f97d5f1c090ca69bc72b7baa072b760d232aa", "host": "77b8dcec-a2be-4f07-ba3b-1dc993690b5e-bluemix.cloudant.com" } } ] }
```

Figure 2-23 IBM Bluemix environment variables

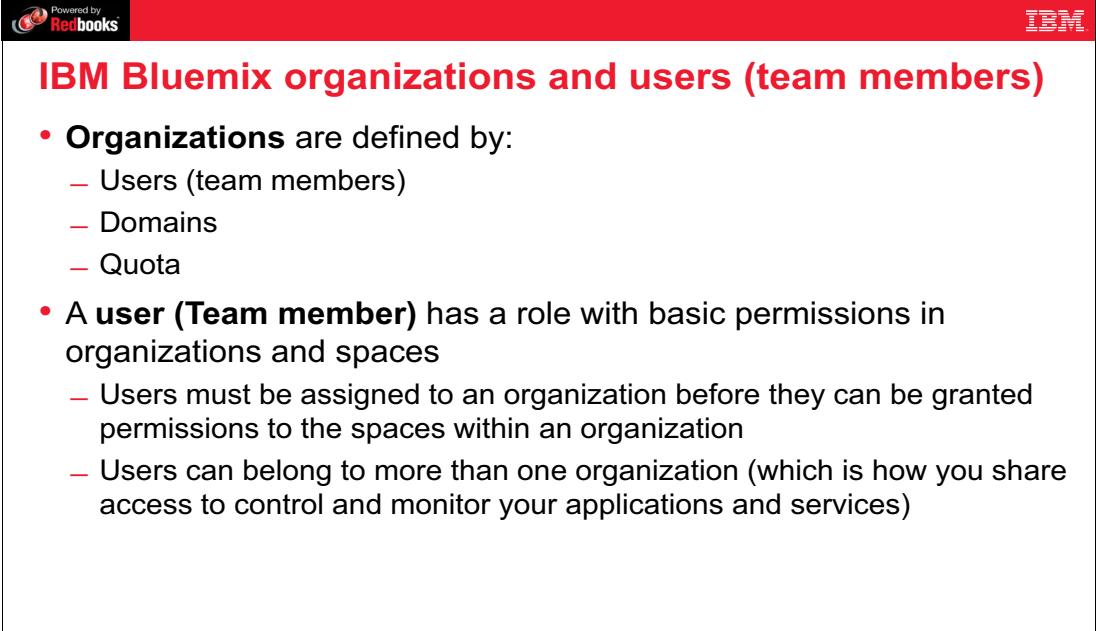
Notes:

Environment variables contain the environment information of a deployed application on IBM Bluemix, which automatically populates the environment variable VCAP_SERVICES with the services that you bind to your IBM Bluemix application.

With the User Defined environment variables, you can set configuration settings without hard-coding the values in your application. For example, you can save the web service endpoint, user name, and password for a cloud-based database as a “user-defined” environment variable.

You can access the Environment Variables of the application from the Application Details page by clicking **Runtime** in the left navigation bar.

2.26 IBM Bluemix organizations and users (team members)



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "IBM Bluemix organizations and users (team members)" is centered in red text. Below the title is a bulleted list of points.

- **Organizations** are defined by:
 - Users (team members)
 - Domains
 - Quota
- A **user (Team member)** has a role with basic permissions in organizations and spaces
 - Users must be assigned to an organization before they can be granted permissions to the spaces within an organization
 - Users can belong to more than one organization (which is how you share access to control and monitor your applications and services)

Figure 2-24 IBM Bluemix organizations and users (team members)

Notes:

Organizations, users, and spaces are the building blocks for organizing resources in the IBM Bluemix environment. The organization is the main organizational unit for IBM Bluemix.

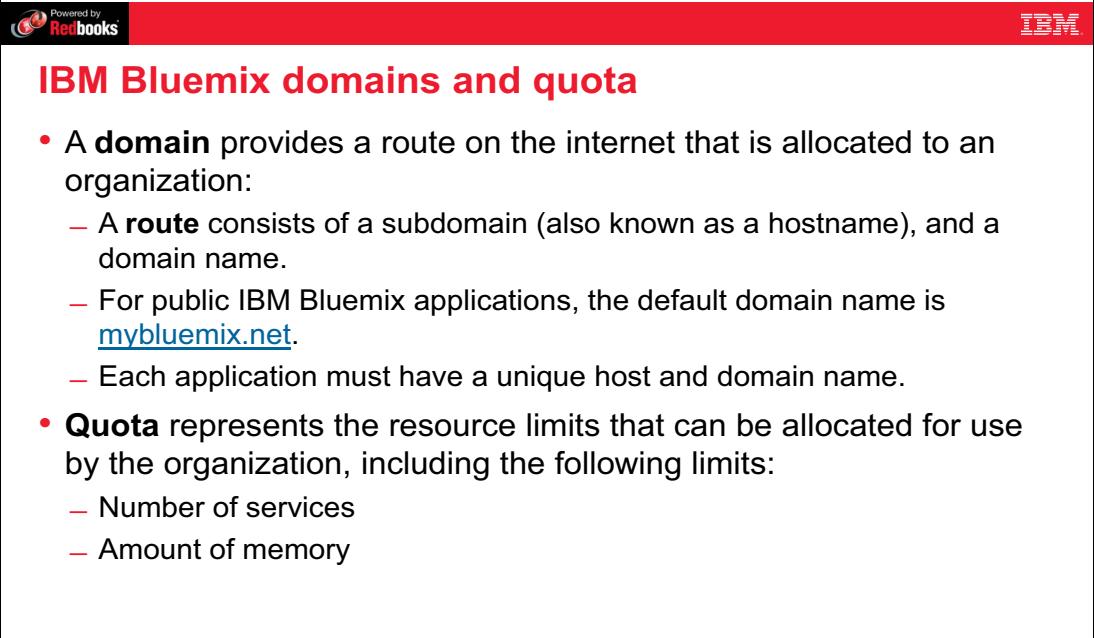
Organizations are defined by the following items:

- ▶ Users or team members
- ▶ Domains
- ▶ Quota

In IBM Bluemix, you can use organizations to enable collaboration among team members and to facilitate the logical grouping of project resources.

A user or team member has a role with basic permissions in organizations and spaces. All users must belong to at least one organization. Users can belong to more than one organization, which is how you can share control over applications and service instances.

2.27 IBM Bluemix domains and quota



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "IBM Bluemix domains and quota" is in bold red text. Below it is a bulleted list:

- A **domain** provides a route on the internet that is allocated to an organization:
 - A **route** consists of a subdomain (also known as a hostname), and a domain name.
 - For public IBM Bluemix applications, the default domain name is [mybluemix.net](#).
 - Each application must have a unique host and domain name.
- **Quota** represents the resource limits that can be allocated for use by the organization, including the following limits:
 - Number of services
 - Amount of memory

Figure 2-25 IBM Bluemix domains and quota

Notes:

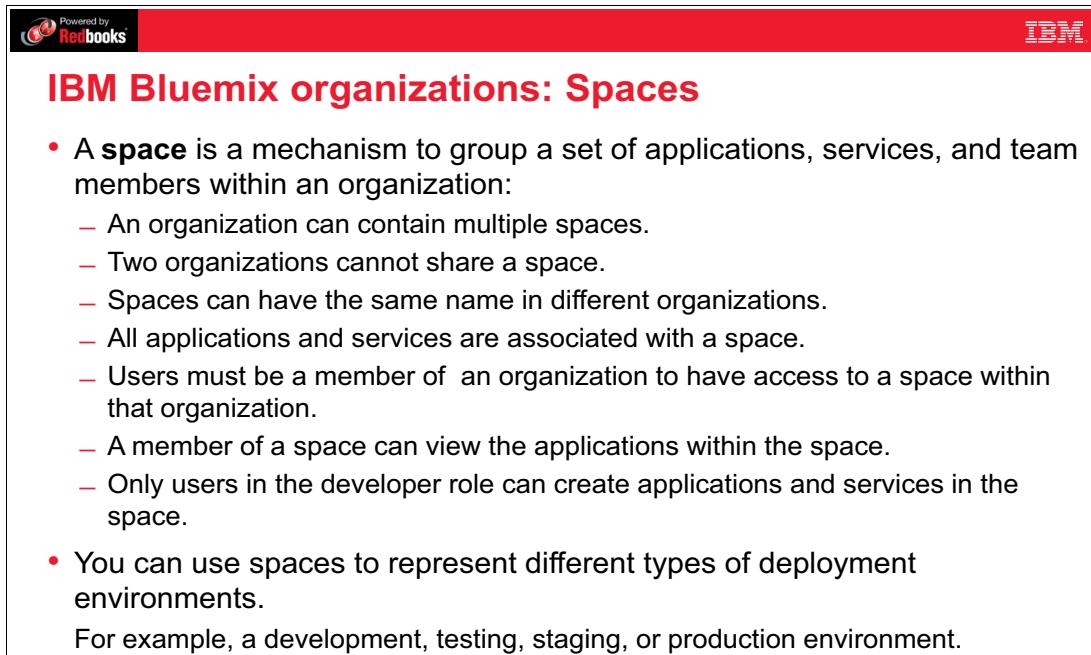
Domains provide the route on the internet that is allocated to the organization. Consider the following points:

- ▶ A route has a sub-domain and a domain.
- ▶ A sub-domain is the hostname, which is typically the application name.
- ▶ A domain might be a system domain or a custom domain that you registered for your application.
- ▶ The default domain name is mybluemix.net.
- ▶ The domain and the route determine how users interact with your IBM Bluemix applications over the network.

Quota represents the resource limits for the organization, including the number of services and the amount of memory that can be allocated for use by the organization. Consider the following points:

- ▶ Quotas are assigned when organizations are created.
- ▶ Any application or service in a space of the organization contributes to the use of the quota.
- ▶ With the Pay-As-You-Go or subscription plans, you can adjust your quota for Cloud Foundry applications and containers as the needs of your organization change.

2.28 IBM Bluemix organizations: Spaces



The slide has a red header bar with the Redbooks logo on the left and the IBM logo on the right. The main title "IBM Bluemix organizations: Spaces" is in bold red text. Below the title is a bulleted list of points about spaces in IBM Bluemix. At the bottom of the list, there is a note in blue text.

- A **space** is a mechanism to group a set of applications, services, and team members within an organization:
 - An organization can contain multiple spaces.
 - Two organizations cannot share a space.
 - Spaces can have the same name in different organizations.
 - All applications and services are associated with a space.
 - Users must be a member of an organization to have access to a space within that organization.
 - A member of a space can view the applications within the space.
 - Only users in the developer role can create applications and services in the space.
- You can use spaces to represent different types of deployment environments.
For example, a development, testing, staging, or production environment.

Figure 2-26 IBM Bluemix organizations: Spaces

Notes:

Spaces in IBM Bluemix are the next level down from organizations. Consider the following points:

- ▶ A space is a mechanism to group a set of applications, services, and team members within an organization.
- ▶ An organization can have multiple spaces, but these spaces must have unique names within an organization.
- ▶ Two organizations can have their own spaces with the same name, but two organizations cannot share the same space.
- ▶ You must belong to an organization to belong to one of its spaces.
- ▶ You must be a developer in the space in which an application or service exists to interact with that application or service as a IBM Bluemix user.
- ▶ IBM Bluemix users typically use spaces to delineate different types of deployment environments, such as development, testing, staging and production.

2.29 IBM Bluemix organizations: User roles

Role	Permissions and tasks
Organization managers	<ul style="list-style-type: none">Create or delete spaces within the organizationInvite users to the organization and manage usersManage domains of the organization
Billing managers	View (read-only) runtime and service usage information for the organization
Organization auditors	View application and service content in the organization
Space managers	<ul style="list-style-type: none">Add users to the space and manage usersEnable features for the space
Space developers	<ul style="list-style-type: none">Create, delete, and manage applications and services within the spaceHave access to logs within the space
Space auditors	Have read-only access to settings, logs, applications, and services

Figure 2-27 IBM Bluemix organizations: User roles

Notes:

Users have the following roles in spaces and organizations:

- The organization manager controls who has access to the organization.
- The billing manager can view usage information for the organization.
- The auditor can view application and service content in the organization.
- Space manager can control who has access to the space.
- Space developer can create, delete, and manage apps and services within the space.
- Space auditors have read only access to settings, logs, apps, and services.

2.30 Managing IBM Bluemix organizations

The screenshot shows the IBM Bluemix Account interface. On the left, there's a sidebar with options like Profile, Notifications, AHMED AZRAQ'S ACCOUNT (selected), Usage Dashboard, Billing, Manage Organizations (highlighted in blue), and several IBM accounts. The main area is titled 'Managing IBM Bluemix organizations' and shows a 'Select region' dropdown set to 'US South'. Below it is a table with three tabs: USERS (selected), DOMAINS, and QUOTA. The USERS tab displays two rows of user information:

USER NAME	MANAGER	BILLING MANAGER	AUDITOR
aazraq@eg.ibm.com Account owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
bart.bogaert@be.ib...	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 2-28 Managing IBM Bluemix organizations

Notes:

Organization managers control who has which type of access to the organization (Manager, Billing Manager, or Auditor).

The organization's managers can invite users to the organization and assign them the various roles.

2.31 Inviting users to an organization

The screenshot shows the IBM Bluemix Identity & Access Management interface. At the top, there's a red banner with the Redbooks logo and the word "Powered by". To the right of the banner is the "IBM" logo. Below the banner, the title "Inviting users to an organization" is displayed in red, followed by the subtitle "The organization's managers can invite users to the organization." In the top navigation bar, there are links for "Catalog", "Support", and "Manage". The main menu on the left is titled "Identity & Access" and includes "Users" (which is selected) and "Bluemix API Keys". The main content area has a heading "Invite users to 'Ahmed Azraq's Account'" and a note "All fields are required.". It contains a section titled "Users" with the sub-instruction "Users can be invited individually or in multiples. All users invited together will be granted the same access." Below this is a form field labeled "Email address or existing IBMid" with the placeholder "Enter email or IBMid." At the bottom of the form, there's a link "Do you have a large team? [Invite multiple users.](#)".

Figure 2-29 Inviting users to an organization

Notes:

The organization's managers can invite users to their organization by clicking **Manage** in the top toolbar and then, clicking **Account** → **Users** → **Invite users**.

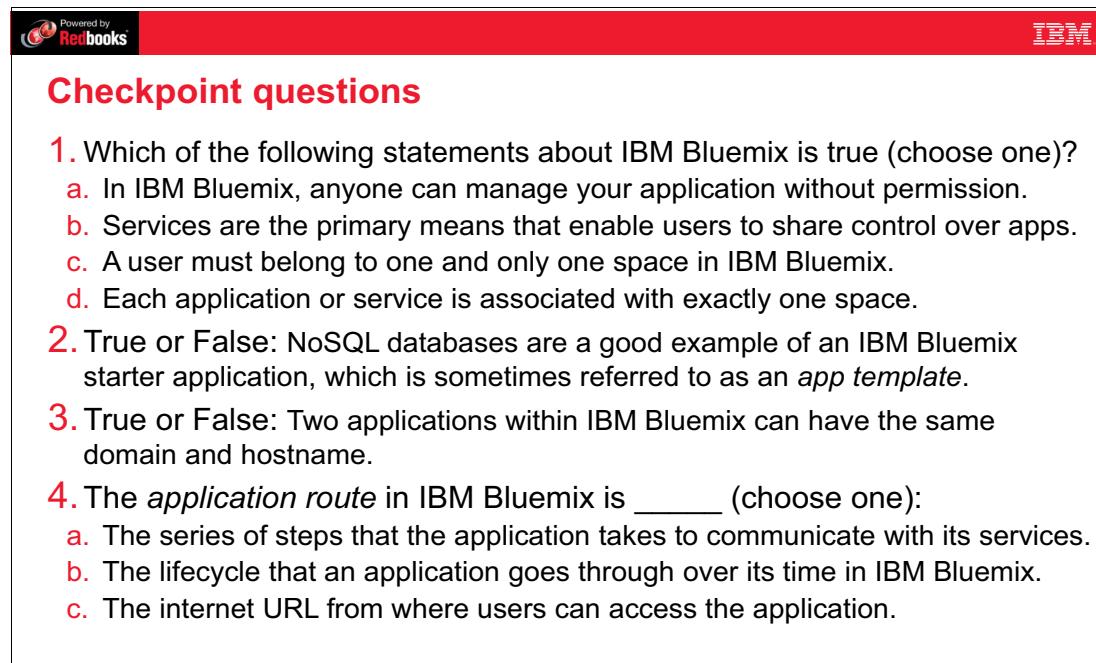
2.32 Unit summary

The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top left of this area is the title "Unit summary" in bold red font. Below it is a paragraph: "By completing this unit, you can now describe:". A bulleted list follows:

- IBM Bluemix infrastructure types
- What you can build in IBM Bluemix
- How to create an application in IBM Bluemix
- The IBM Bluemix dashboard, catalog, and documentation features
- How the application route is used to test an application from the browser
- How to create services in IBM Bluemix
- How to bind services to an application in IBM Bluemix
- The environmental variables that are used with IBM Bluemix services
- IBM Bluemix organizations, domains, spaces, and users

Figure 2-30 Unit summary

2.33 Checkpoint questions

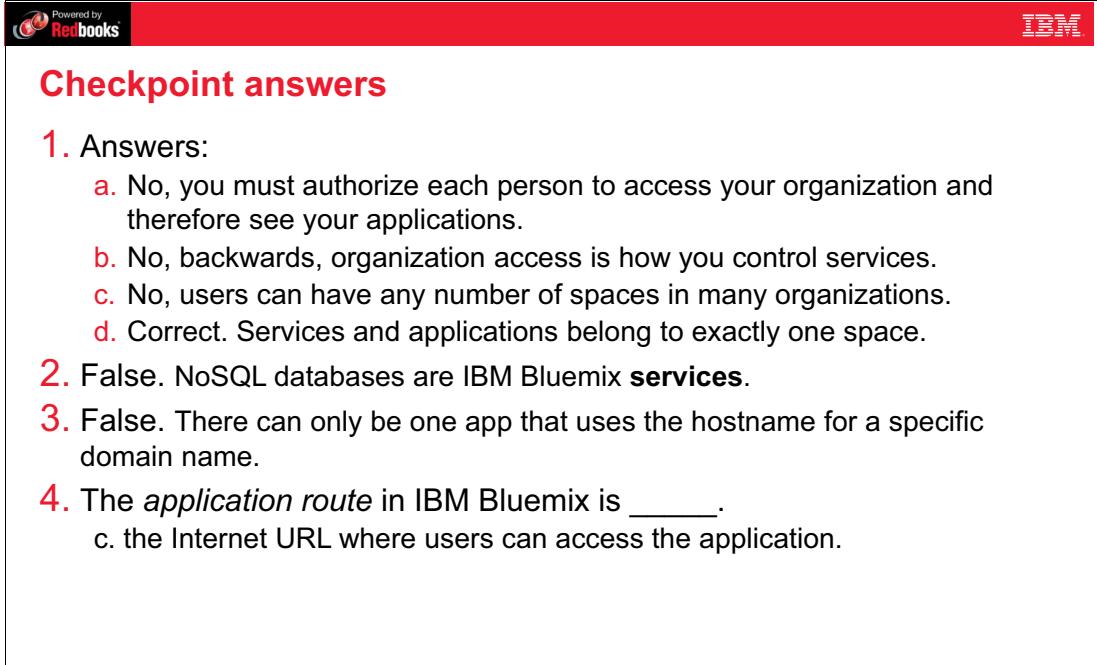


The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a red header that reads "Checkpoint questions". Below this, there is a numbered list of four questions:

1. Which of the following statements about IBM Bluemix is true (choose one)?
 - a. In IBM Bluemix, anyone can manage your application without permission.
 - b. Services are the primary means that enable users to share control over apps.
 - c. A user must belong to one and only one space in IBM Bluemix.
 - d. Each application or service is associated with exactly one space.
2. True or False: NoSQL databases are a good example of an IBM Bluemix starter application, which is sometimes referred to as an *app template*.
3. True or False: Two applications within IBM Bluemix can have the same domain and hostname.
4. The *application route* in IBM Bluemix is _____ (choose one):
 - a. The series of steps that the application takes to communicate with its services.
 - b. The lifecycle that an application goes through over its time in IBM Bluemix.
 - c. The internet URL from where users can access the application.

Figure 2-31 Checkpoint questions

2.34 Checkpoint answers



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo.

Checkpoint answers

1. Answers:
 - a. No, you must authorize each person to access your organization and therefore see your applications.
 - b. No, backwards, organization access is how you control services.
 - c. No, users can have any number of spaces in many organizations.
 - d. Correct. Services and applications belong to exactly one space.
2. False. NoSQL databases are IBM Bluemix **services**.
3. False. There can only be one app that uses the hostname for a specific domain name.
4. The *application route* in IBM Bluemix is _____.
 - c. the Internet URL where users can access the application.

Figure 2-32 Checkpoint answers



Developing IBM Bluemix applications from a local workstation

This unit covers the following topics:

- ▶ Review: Creating an IBM Bluemix Node.js application from the IBM Bluemix web interface
- ▶ Build an application with the command-line interface
- ▶ Build an application with the IBM Eclipse Tools for IBM Bluemix

3.1 What you should be able to do

After completing this unit, you should be able to:

- ▶ Describe how to create an IBM SDK for Node.js application that runs on IBM Bluemix.
- ▶ Describe the features in IBM Bluemix that help you to set up a cooperative workstation environment.
- ▶ Explain how to manage your IBM Bluemix account with the Cloud Foundry CLI.
- ▶ Describe how to set up and use the IBM Bluemix plug-in for Eclipse.
- ▶ Describe the role of Node.js for server-side scripting.

3.2 References

The following publications are useful for further research on the topic presented in this unit:

- ▶ The Eclipse project:
<http://www.eclipse.org/>
- ▶ Most recent cf installer from the Git project:
<https://github.com/cloudfoundry/cli/releases>
- ▶ *Deploying your app with the command line interface* documentation:
https://www.ng.bluemix.net/docs/starters/install_cli.html
- ▶ *Developing with Eclipse tools* documentation:
https://www.ng.bluemix.net/docs/starters/deploy_eclipsetools.html

3.3 Node.js

Node.js

- Node.js is an open-source language that runs on V8.
- V8 is an open-source engine that was developed by Google for the Google Chrome browser.
- Developers use JavaScript for client-side functionality all the time. Node.js is the server component in the same language.
- Although developers can still use JavaScript for browser functionality in frameworks, such as angularJS, dojo, and jQuery, they can now use Node.js in the same part of the architecture where they use Java, Perl, C++, Python, and Ruby.
- Node.js is used in production by companies, such as Uber, Yahoo!, LinkedIn, GoDaddy, eBay, and PayPal.
- Node.js is event-driven and uses asynchronous, non-blocking I/O.

Figure 3-1 Node.js

Notes:

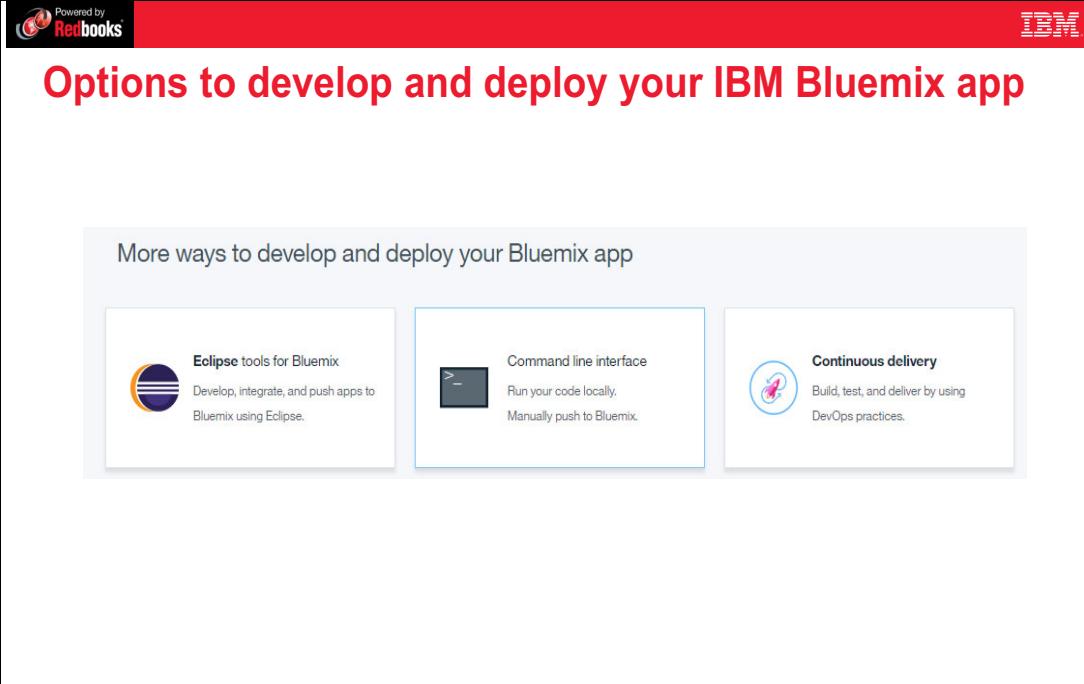
Node.js is open-source language that runs on V8. V8 is an open source engine developed by Google for the Google Chrome browser.

Developers use JavaScript for client-side functionality often. Node.js is the server component in the same language.

Although developers can still use JavaScript for browser functionality in frameworks, such as angularJS, dojo, and jQuery, they can now use Node.js in the same components of the architecture where they use Java, Perl, C++, Python, and Ruby.

Node.js is used in production by companies, such as Uber, Yahoo!, LinkedIn, GoDaddy, eBay, and PayPal. It is event-driven and uses asynchronous, non-blocking I/O.

3.4 Options to develop and deploy your IBM Bluemix app



The diagram is titled "Options to develop and deploy your IBM Bluemix app". It features a red header bar with the Redbooks logo and the IBM logo. Below the title, there is a section titled "More ways to develop and deploy your Bluemix app". This section contains three boxes: "Eclipse tools for Bluemix" (with an Eclipse icon), "Command line interface" (with a terminal icon), and "Continuous delivery" (with a gear icon). Each box includes a brief description of its function.

Option	Description
Eclipse tools for Bluemix	Develop, integrate, and push apps to Bluemix using Eclipse.
Command line interface	Run your code locally. Manually push to Bluemix.
Continuous delivery	Build, test, and deliver by using DevOps practices.

Figure 3-2 Options to develop and deploy your IBM Bluemix app

Notes:

The following options are available to develop and deploy your IBM Bluemix application:

- ▶ Eclipse
- ▶ Command-line interface (CLI):
 - Cloud Foundry CLI: The Cloud Foundry command-line interface (cf CLI) provides a set of commands for managing your apps. You can deal with IBM Bluemix or any platform that's based on Cloud Foundry using the CF CLI.
 - Bluemix CLI: The Bluemix CLI (bx CLI) provides a set of commands for users to interact with IBM Bluemix. Some IBM Bluemix commands are wrappers of existing cf commands, while others provide extended capabilities for IBM Bluemix users.
- ▶ Continuous Delivery using DevOps Practices

This unit covers developing and deploying Bluemix applications by using bx CLI and Eclipse.

For more information about the use of DevOps services, see Chapter 4, “Using IBM Bluemix DevOps services” on page 85.

3.5 Overview of bx and cf command-line interfaces



Overview of bx and cf command-line interfaces

- Cloud Foundry is an open-source platform as a service (PaaS) that provides a range of cloud environments, frameworks, and application services.
- IBM Bluemix implements IBM Open Cloud Architecture, which is based on Cloud Foundry.
- Bluemix (bx) command-line interface and Cloud Foundry (cf) command-line interface provide a simple way to manage your application on Bluemix from your workstation.
- Use the *bx* or *cf cli* for the following tasks:
 - Log in to the IBM Bluemix environment
 - Push an application to your IBM Bluemix account
 - Manage domains, routes, organizations, and spaces

© Copyright IBM Corporation 2017

Figure 3-3 Overview of bx and cf command-line interfaces

Notes:

Cloud Foundry is an open-source platform as a service (PaaS). Cloud Foundry provides an open-source CLI that is available for multiple platforms.

You can use the CLI through bx CLI or cf CLI to log in to IBM Bluemix and deploy an application to IBM Bluemix, and to manage domains, routes, organizations, and spaces.

3.6 Why use command-line tools?

Powered by
Redbooks

IBM

Why use command-line tools?

- You want to deploy, update, and manage your IBM Bluemix environment with a fast, minimal tool.
- You want to automate the management and deployment process with shell scripts.
- You want to use specific tools for each project to provide some level of separation.

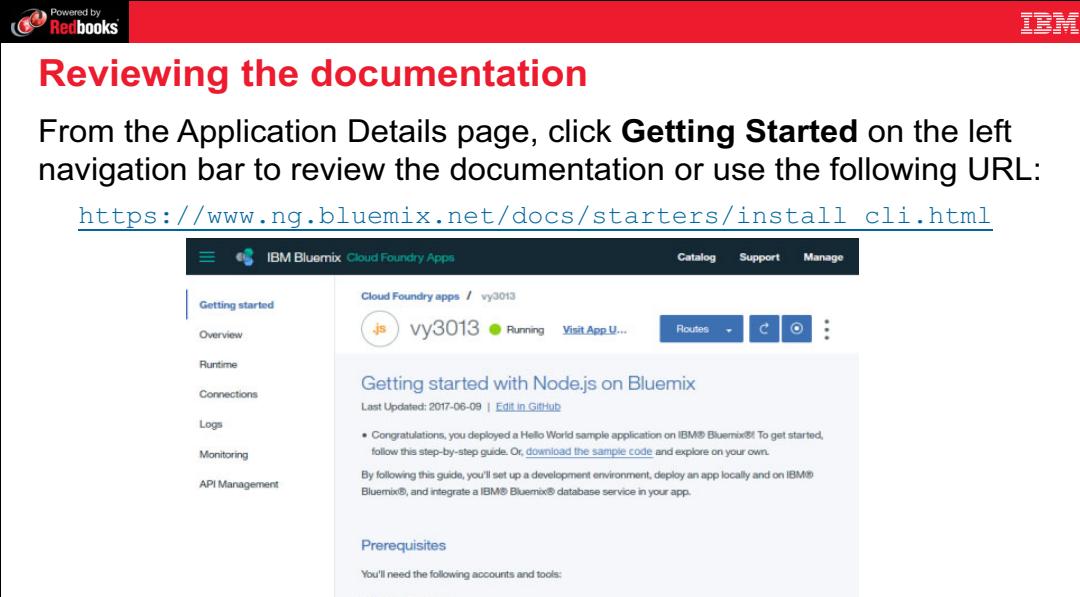
Figure 3-4 Why use command-line tools?

Notes:

IBM Bluemix integrates into the Eclipse IDE. With this available GUI, why would you want to use command-line tools? You might want to use a tool that is quick to install and has a minimal footprint on your development workstation.

Also, you might want to automate the management and deployment process with shell scripts, or you want to use different tools for different projects to provide some project-level separation.

3.7 Reviewing the documentation



The screenshot shows the IBM Bluemix Cloud Foundry Apps interface. At the top, there's a navigation bar with 'IBM' on the right and a 'Powered by Redbooks' logo on the left. Below the bar, the title 'Reviewing the documentation' is displayed in red. A main content area contains a heading 'From the Application Details page, click **Getting Started** on the left navigation bar to review the documentation or use the following URL:' followed by a blue link: https://www.ng.bluemix.net/docs/starters/install_cli.html. To the left of this text is a sidebar with a 'Getting started' section containing links for Overview, Runtime, Connections, Logs, Monitoring, and API Management. The main content area also shows a 'Cloud Foundry apps / vy3013' section with a 'js' icon, the app name 'vy3013', a green 'Running' status indicator, and a 'Visit App U...' button. Below this, a 'Getting started with Node.js on Bluemix' guide is shown, last updated on 2017-06-09. It includes a note about congratulations on deploying a Hello World sample application and provides a link to download the sample code. The guide also mentions setting up a development environment, deploying locally and on Bluemix, and integrating a database service. A 'Prerequisites' section lists the need for a Bluemix account.

Figure 3-5 Reviewing the documentation

Notes:

You can review the documentation for the command-line tool from Getting Started in Application Details. From this page, you can also download the bx CLI by clicking **Download Bluemix Command Line Interface**. The bx CLI is based on the cf CLI.

To work with IBM Bluemix by using the CLI, you can use bx CLI or cf CLI. The next steps show how to download and use the command-line interface to manage IBM Bluemix applications.

3.8 Using the cf CLI, Step 1: Install the cf CLI

**Using the cf CLI:
Step 1: Install the cf CLI**

- Download the most recent cf installer from the Git project:
<https://github.com/cloudfoundry/ci/releases>
- Install the cf tool to your local workstation
- Verify the cf tool by entering `cf --version` to check that the tool works successfully in your environment

Installers

- Debian [64 bit / 32 bit \(deb\)](#)
- Redhat [64 bit / 32 bit \(rpm\)](#)
- Mac OS X [64 bit \(pkg\)](#)
- Windows [64 bit / 32 bit \(zip\)](#)

Binaries

- Linux [64 bit / 32 bit \(tgz\)](#)
- Mac OS X [64 bit \(tgz\)](#)
- Windows [64 bit / 32 bit \(zip\)](#)

Figure 3-6 Step 1: Install the cf CLI

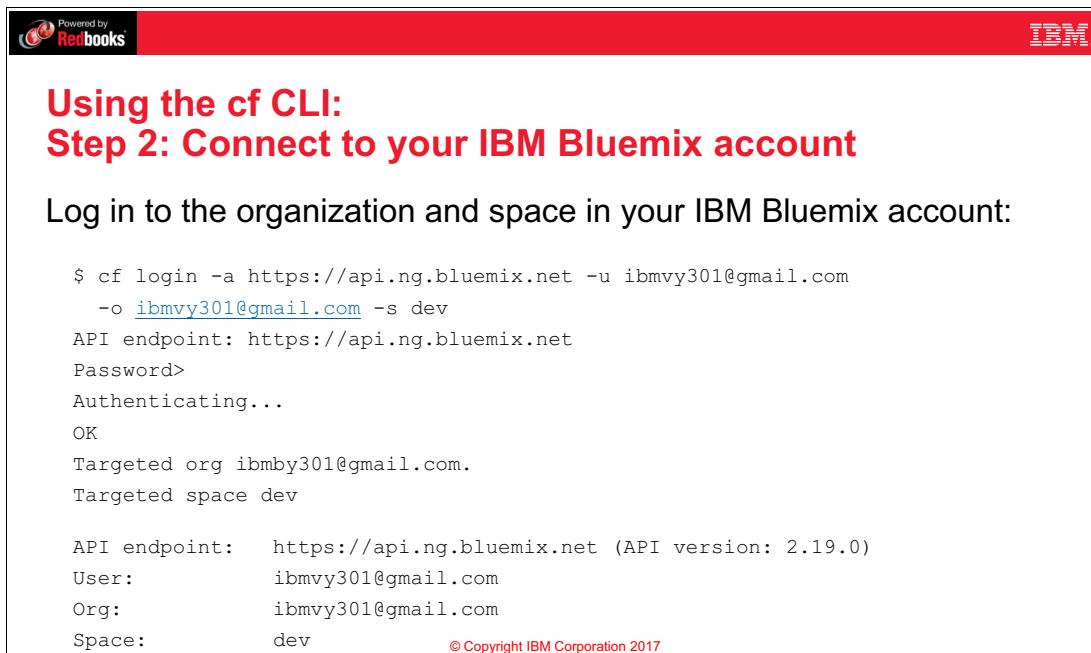
Notes:

Download the most recent cf installer from the Git project, which includes a list of installers and binaries downloadable for most systems. Generally, download an installer because they are faster and easier to use than the raw binaries.

After you download and run the installer, open a command prompt on your system and run the command line tool by entering `cf --version`.

More information about this process is included in the exercises that accompany this unit.

3.9 Using the cf CLI, Step 2: Connect to your IBM Bluemix account



The screenshot shows a terminal window with a red header bar. On the left of the header is the Redbooks logo, and on the right is the IBM logo. The main content area has a white background. At the top, it says "Using the cf CLI:" in red, followed by "Step 2: Connect to your IBM Bluemix account" in red. Below that, the text "Log in to the organization and space in your IBM Bluemix account:" is displayed in black. Then, a command-line session is shown:

```
$ cf login -a https://api.ng.bluemix.net -u ibmvy301@gmail.com
-o ibmvy301@gmail.com -s dev
API endpoint: https://api.ng.bluemix.net
Password>
Authenticating...
OK
Targeted org ibmvy301@gmail.com.
Targeted space dev

API endpoint: https://api.ng.bluemix.net (API version: 2.19.0)
User: ibmvy301@gmail.com
Org: ibmvy301@gmail.com
Space: dev
© Copyright IBM Corporation 2017
```

Figure 3-7 Step 2: Connect to your IBM Bluemix account

Notes:

Log in to IBM Bluemix by using the CLI. The login command requires the following parameters:

- ▶ Cloud Foundry API endpoint
- ▶ Username
- ▶ Password
- ▶ Organization name
- ▶ Space name

The exercises provide instructions for how to log in by using this method. You can use the command-line client with other Cloud Foundry regions by pointing to other API endpoints; for example, api.eu-gb.bluemix.net.

This course uses the US South region. You can use the flags (dash a, dash u, and so on) to specify the parameters at the command line. If you do not specify one of the parameters, the command-line utility prompts you to enter it later. In the example that is shown in Figure 3-7, the password was not entered, so the command-line utility prompts the user to enter the password.

If you are using a federated ID, use cf login –sso. A URL is provided for you. Open this URL in a browser; then, login and copy your one-time passcode and paste it in the CLI to authenticate.

3.10 Using the cf CLI, Step 3: Deploy your application to IBM Bluemix

Using the cf CLI: Step 3: Deploy your application to IBM Bluemix

- The **push** command uploads, deploys, and starts the application in your Bluemix space.
- If you push to an application, the cf command replaces the current application on the server.

```
$ cf push ibm301-xxx-nodesample  
Using manifest file /Users/ibm301/bluemix/manifest.yml  
  
Updating app ibm301-xxx-nodesample in org ibmvy301@gmail.com / space dev as  
ibm301@gmail.com  
OK  
Using route ibm301-xxx-nodesample.mybluemix.net  
Uploading ibm301-xxx-nodesample...  
Uploading app files from: /users/ibm301/bluemix  
Done uploading  
OK
```

© Copyright IBM Corporation 2017

The manifest includes stored deployment settings.

Write down the route to your application. Test your application with this entry point.

Figure 3-8 Step 3: Deploy your application to IBM Bluemix

Notes:

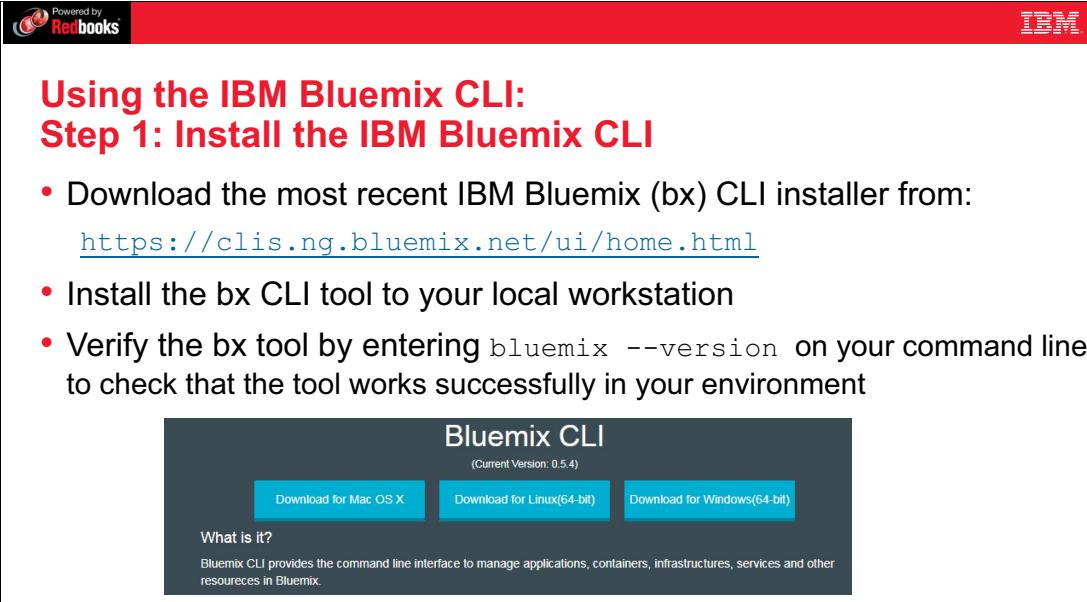
The **push** command uploads, deploys, and starts the application in your IBM Bluemix space.

When you are ready to deploy your application to IBM Bluemix, navigate to your application directory; then, run the **cf push** command. “Pushing your app” is the terminology that is used to indicate that you are going to deploy the latest changes of your app to IBM Bluemix.

Remember that when you push your application, it overwrites the current contents of your application in your IBM Bluemix account.

Figure 3-8 shows the command-line utility accessing a manifest file. You can control various aspects of the deployment, such as the amount of memory an application receives and the route of the application, by specifying them as flags in the command. However, you can also store these settings in a manifest YML file to make the command you need shorter to enter in later iterations of your application.

3.11 Using the IBM Bluemix CLI, Step 1: Install the IBM Bluemix CLI



The screenshot shows a web page titled "Using the IBM Bluemix CLI: Step 1: Install the IBM Bluemix CLI". The page has a red header with the IBM logo and a "Powered by Redbooks" badge. Below the header, there is a section titled "Using the IBM Bluemix CLI: Step 1: Install the IBM Bluemix CLI". A bulleted list provides instructions:

- Download the most recent IBM Bluemix (bx) CLI installer from:
<https://clis.ng.bluemix.net/ui/home.html>
- Install the bx CLI tool to your local workstation
- Verify the bx tool by entering `bluemix --version` on your command line to check that the tool works successfully in your environment

Below the instructions is a dark blue box containing the "Bluemix CLI" logo and download links for Mac OS X, Linux(64-bit), and Windows(64-bit). The text "What is it?" and a brief description of the CLI's function are also present.

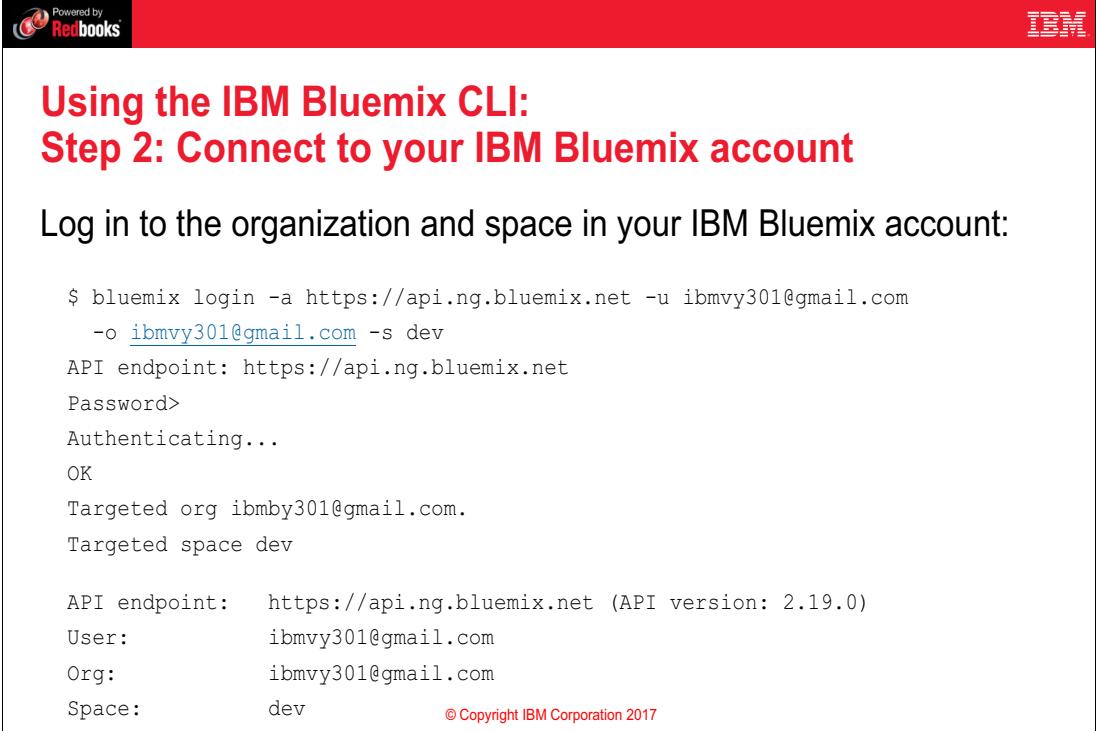
Figure 3-9 Step 1: Install the IBM Bluemix CLI

Notes:

Download the most recent IBM Bluemix Command-line Interface installer, which includes a list of downloadable installers for Mac OS X, Linux, and Windows. After you download and run the installer, open a command prompt on your system and run the command line tool by entering `bluemix --version`.

More information about this process is included in the exercises that accompany this unit.

3.12 Using the IBM Bluemix CLI, Step 2: Connect to your IBM Bluemix account



The screenshot shows a web-based IBM Bluemix CLI interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. The main content area has a white background with a red title bar containing the text "Using the IBM Bluemix CLI: Step 2: Connect to your IBM Bluemix account". Below this, a large heading says "Log in to the organization and space in your IBM Bluemix account:". Underneath, there's a code block showing the terminal command and its output:

```
$ bluemix login -a https://api.ng.bluemix.net -u ibmvy301@gmail.com  
-o ibmvy301@gmail.com -s dev  
API endpoint: https://api.ng.bluemix.net  
Password>  
Authenticating...  
OK  
Targeted org ibmby301@gmail.com.  
Targeted space dev  
  
API endpoint: https://api.ng.bluemix.net (API version: 2.19.0)  
User: ibmvy301@gmail.com  
Org: ibmvy301@gmail.com  
Space: dev
```

© Copyright IBM Corporation 2017

Figure 3-10 Step 2: Connect to your IBM Bluemix account

Notes:

Log in to IBM Bluemix by using the CLI. The login command requires the same parameters that are used in the cf CLI.

If you are using a federated ID, use `bluemix login -sso`. A URL is supplied for you. Open this URL in a browser; then, log in and paste your one-time passcode into the CLI to authenticate.

3.13 Using the IBM Bluemix CLI, Step 3: Deploy your application to IBM Bluemix

The screenshot shows a terminal window with the following content:

```
$ bluemix app push ibm301-xxx-nodesample
Using manifest file /Users/ibm301/bluemix/manifest.yml
Updating app ibm301-xxx-nodesample in org ibmvy301@gmail.com / space dev as
ibm301@gmail.com
OK
Using route ibm301-xxx-nodesample.mybluemix.net
Uploading ibm301-xxx-nodesample...
Uploading app files from: /users/ibm301/bluemix
Done uploading
OK
© Copyright IBM Corporation 2017
```

Annotations on the right side of the terminal output:

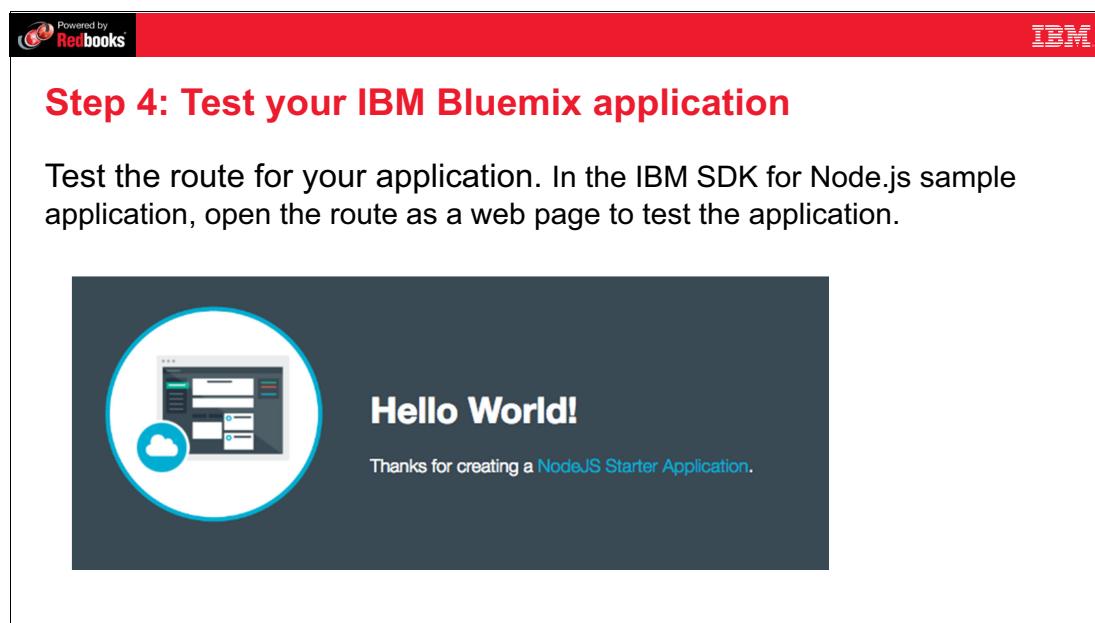
- A callout box points to the line "Using manifest file /Users/ibm301/bluemix/manifest.yml" with the text: "The manifest includes stored deployment settings."
- A callout box points to the line "Using route ibm301-xxx-nodesample.mybluemix.net" with the text: "Write down the route to your application. Test your application with this entry point."

Figure 3-11 Step 3: Deploy your application to IBM Bluemix

Notes:

The use of the `bluemix app push` command uploads, deploys, and starts the application in your IBM Bluemix space.

3.14 Test your IBM Bluemix application



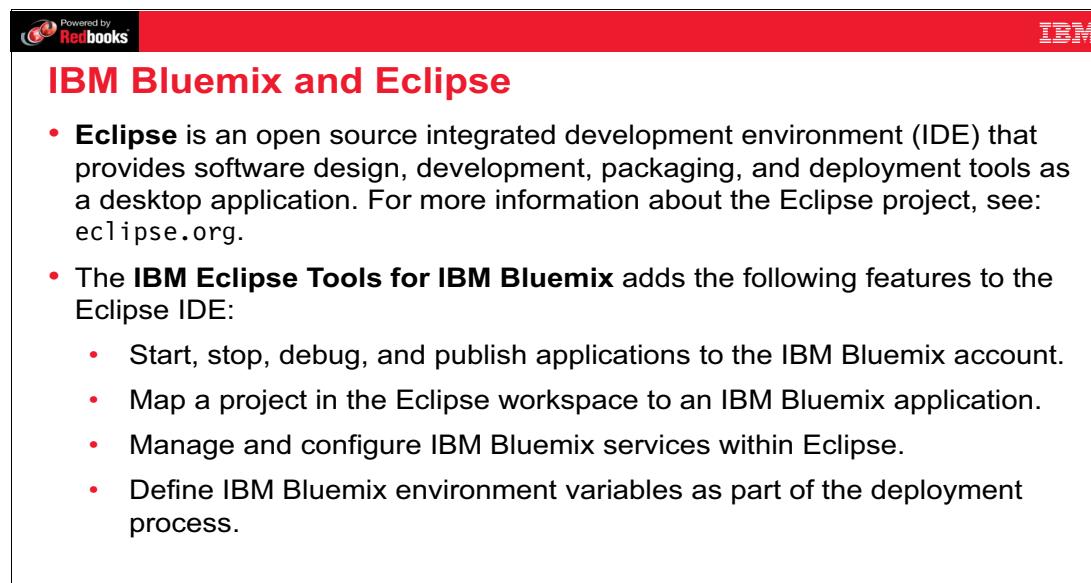
The screenshot shows a web browser window with a red header bar. On the left of the header is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a dark blue background. At the top, it says "Step 4: Test your IBM Bluemix application" in red. Below that, in white text, it says "Hello World!". Underneath "Hello World!", it says "Thanks for creating a NodeJS Starter Application." To the left of the text, there is a circular icon containing a stylized computer monitor with a cloud icon below it.

Figure 3-12 Test your IBM Bluemix application

Notes:

After the command-line utility finishes pushing your application, browse to the route name you wrote down in the previous step to test whether the application is running.

3.15 IBM Bluemix and Eclipse



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "IBM Bluemix and Eclipse" is in bold red text. Below it is a bulleted list:

- **Eclipse** is an open source integrated development environment (IDE) that provides software design, development, packaging, and deployment tools as a desktop application. For more information about the Eclipse project, see: eclipse.org.
- The **IBM Eclipse Tools for IBM Bluemix** adds the following features to the Eclipse IDE:
 - Start, stop, debug, and publish applications to the IBM Bluemix account.
 - Map a project in the Eclipse workspace to an IBM Bluemix application.
 - Manage and configure IBM Bluemix services within Eclipse.
 - Define IBM Bluemix environment variables as part of the deployment process.

Figure 3-13 IBM Bluemix and Eclipse

Notes:

Eclipse is a popular open source integrated development environment (IDE) that provides development tools in a desktop application. For more information about Eclipse, see this website:

<http://www.eclipse.org/>

IBM Eclipse tools for IBM Bluemix is an open-source plugin for interacting with IBM Bluemix to manage and deploy your applications and services from within the Eclipse interface.

3.16 Step 1: Review the documentation

The screenshot shows a web page titled "Step 1: Review the documentation". The page is part of the IBM Bluemix documentation, as indicated by the "Powered by Redbooks" logo and the "IBM" logo in the top right corner. The main title is "Step 1: Review the documentation". Below it, a sub-section title is "From the Application Details page, click **Getting Started** on the left navigation bar to review the documentation “Developing with Eclipse Tools” or use the following URL:". A blue hyperlink URL is provided: <https://console.bluemix.net/docs/manageapps/eclipsetools/eclipsetools.html#eclipsetools>. The main content area has a light gray background and contains the following text:

Developing with Eclipse Tools
Last Updated: 2017-01-12 | [Edit in GitHub](#)

IBM® Eclipse Tools for Bluemix® provides lightweight tools in Eclipse for rapid development and integration of applications with IBM® Bluemix® or Cloud Foundry Clouds.

① If you don't already have Eclipse, install Eclipse Neon for Java EE Developers (4.6.1).

② Click and hold the following button to drag and drop it into the Eclipse toolbar, and then follow the prompts to install IBM Eclipse Tools for Bluemix:

[Install](#)

For step-by-step directions in deploying apps, see [Deploying apps with IBM Eclipse Tools for Bluemix](#).

Figure 3-14 Step 1: Review the documentation

Notes:

To start developing with the IBM Eclipse Tools for IBM Bluemix plugin, Eclipse for Java EE Developers (Neon or any newer versions) must be installed. The Java Development Kit must be installed first.

You can then browse to the URL that is shown in Figure 3-14 to review the documentation for Eclipse tools.

3.17 Step 2: Install the latest version of Eclipse

The screenshot shows a web page with a red header bar. On the left is the Redbooks logo, and on the right is the IBM logo. The main title 'Step 2: Install the latest version of Eclipse' is displayed in large red text. Below it, a sub-instruction reads: 'Download and install the latest version of Eclipse IDE for Java EE Developers:'. A blue link 'https://www.eclipse.org/downloads/packages/' is provided. The central part of the page features a download card for 'Eclipse IDE for Java EE Developers'. It includes a small icon of a gear with 'Java EE IDE' text, the package name, its size (304 MB), and the number of downloads (1,300,459 Times). To the right, there are three download links: 'Windows 32-bit 64-bit', 'Mac Cocoa 64-bit', and 'Linux 32-bit 64-bit'.

Figure 3-15 Step 2: Install the latest version of Eclipse

Notes:

Eclipse Tools for IBM Bluemix works with Eclipse Neon or a newer version.

3.18 Step 3: Install Eclipse and Eclipse tools for IBM Bluemix

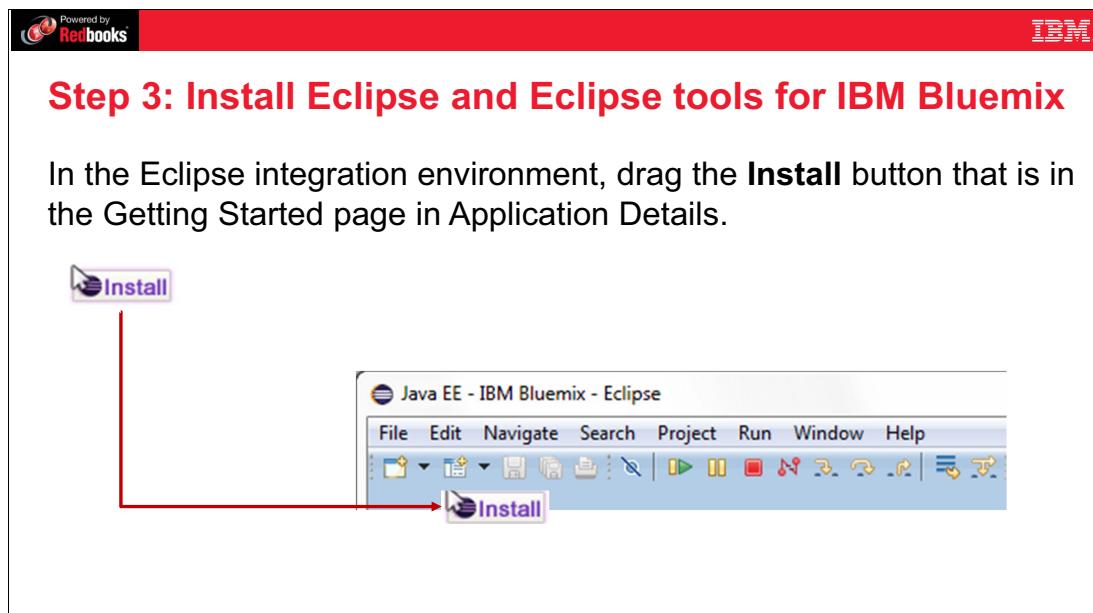


Figure 3-16 Step 3: Install Eclipse and Eclipse tools for IBM Bluemix

Notes:

The simplest way to install the IBM Eclipse Tools for IBM Bluemix is to drag the **Install** button from the documentation on top of your Eclipse toolbar, as shown in Figure 3-16.

You can also install IBM Eclipse Tools for IBM Bluemix by completing the following steps:

1. Start Eclipse on your workstation and click **Help** → **Eclipse Marketplace**. Then, search for Bluemix.
2. Select **IBM Eclipse Tools for Bluemix** and then, click **Install**.
3. Some features are selected for you by default. Click **Confirm**.
4. Accept the license agreement and click **Finish**.

3.19 Step 4: Connect to your IBM Bluemix account

The screenshot shows the Eclipse IDE interface. At the top, there's a red header bar with the text "Powered by Redbooks" and the IBM logo. Below the header, the title "Step 4: Connect to your IBM Bluemix account" is displayed in large red font. Underneath the title, there are two bullet points:

- In the Eclipse integrated development environment, create a server connection for your IBM Bluemix account.
- Define an IBM Bluemix server connection and enter the IBM ID and password for your account.

Below the bullet points is a screenshot of the Eclipse "Servers" view. It shows a message: "No servers are available. Click this link to create a new server...".

At the bottom of the screenshot is a window titled "New Server" for "IBM Bluemix Account". It contains fields for "Email" (warren@ca.ibm.com), "Password" (redacted), and "URL" (IBM Bluemix - https://api.ng.bluemix.net). There are buttons for "Validate Account", "Register Account...", and "Sign Up".

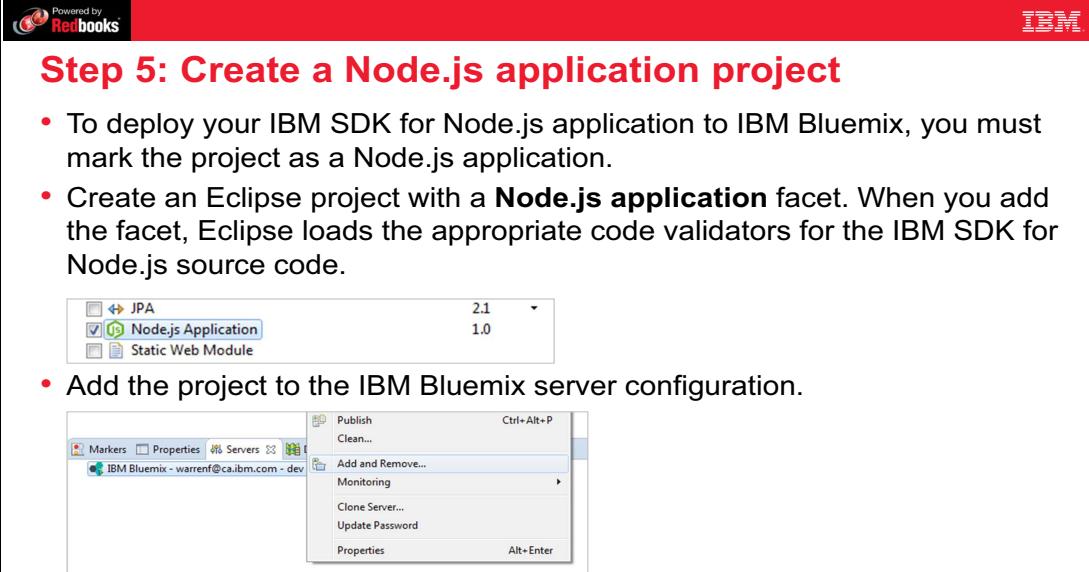
Figure 3-17 Step 4: Connect to your IBM Bluemix account

Notes:

After you successfully install the IBM Bluemix tools for Eclipse plugin, you can set up a server object within Eclipse that links to your IBM Bluemix account.

Create a server object; then, enter the IBM ID and password that you use to log in to your IBM Bluemix account.

3.20 Step 5: Create a Node.js application project



The screenshot shows a window titled "Step 5: Create a Node.js application project". It contains two main sections: a list of steps and a screenshot of the Eclipse IDE interface.

- To deploy your IBM SDK for Node.js application to IBM Bluemix, you must mark the project as a Node.js application.
- Create an Eclipse project with a **Node.js application** facet. When you add the facet, Eclipse loads the appropriate code validators for the IBM SDK for Node.js source code.

In the screenshot of the Eclipse interface, a context menu is open over a server named "IBM Bluemix - warrenf@ca.ibm.com - dev". The "Add and Remove..." option is highlighted in the "Servers" submenu.

Figure 3-18 Step 5: Create a Node.js application project

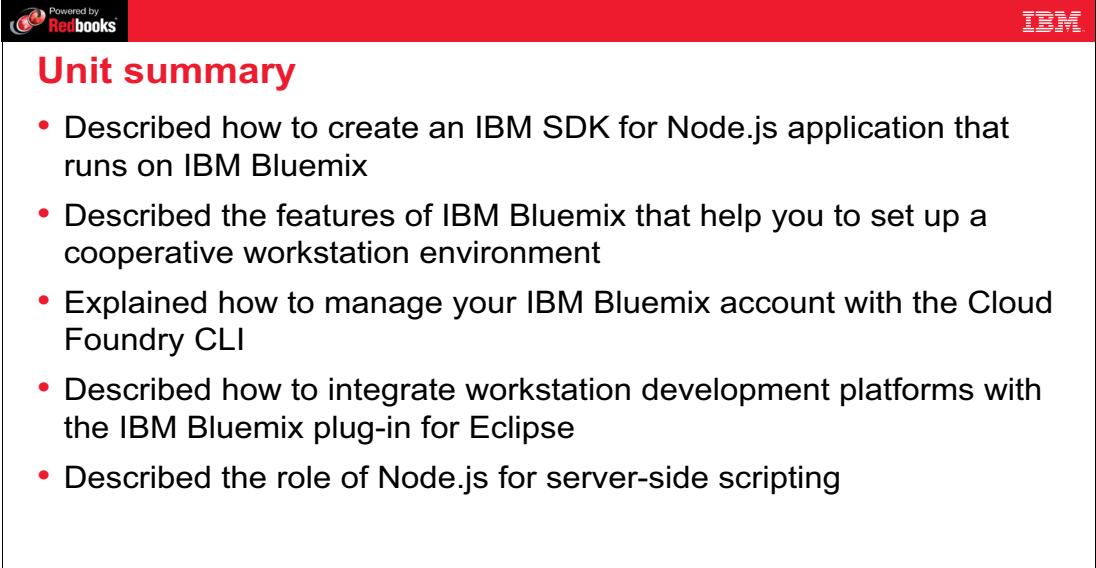
Notes:

You can add only projects for server runtime environments that IBM Bluemix supports. For example, you must mark your Eclipse project as a Node.js application to deploy it for the IBM SDK for Node.js runtime. You “mark” your project by using Eclipse project facets.

To view or modify the facets for your project, select the Properties for your project and select **Project Facets**. After you add the Node.js facet to your application, it appears in Eclipse as an item that you can deploy to IBM Bluemix.

To deploy an application to IBM Bluemix, right-click the Bluemix server and click **Add and Remove**. Then, add the application to the server. Alternatively, you can drop the application to the server.

3.21 Unit summary

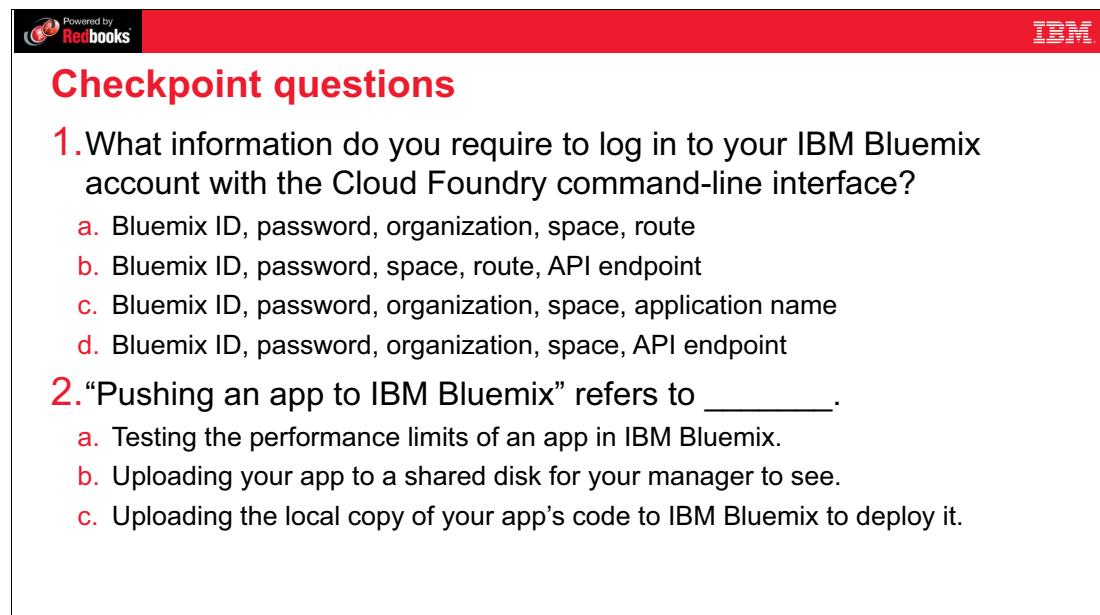


Unit summary

- Described how to create an IBM SDK for Node.js application that runs on IBM Bluemix
- Described the features of IBM Bluemix that help you to set up a cooperative workstation environment
- Explained how to manage your IBM Bluemix account with the Cloud Foundry CLI
- Described how to integrate workstation development platforms with the IBM Bluemix plug-in for Eclipse
- Described the role of Node.js for server-side scripting

Figure 3-19 Unit summary

3.22 Checkpoint questions



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a red title bar at the top with the text "Checkpoint questions" in white.

Checkpoint questions

- 1.** What information do you require to log in to your IBM Bluemix account with the Cloud Foundry command-line interface?
 - a. Bluemix ID, password, organization, space, route
 - b. Bluemix ID, password, space, route, API endpoint
 - c. Bluemix ID, password, organization, space, application name
 - d. Bluemix ID, password, organization, space, API endpoint
- 2.** “Pushing an app to IBM Bluemix” refers to _____.
 - a. Testing the performance limits of an app in IBM Bluemix.
 - b. Uploading your app to a shared disk for your manager to see.
 - c. Uploading the local copy of your app’s code to IBM Bluemix to deploy it.

Figure 3-20 Checkpoint questions

3.23 Checkpoint answers

The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a red title bar at the top with the text "Checkpoint answers" in white.

Checkpoint answers

- 1.d.** To successfully log in, you must enter your username, password, IBM Bluemix organization, space, and API endpoint. Your username is the IBM ID or email address with which you created your IBM Bluemix account.
- 2.c.** Uploading the local copy of your app's code to IBM Bluemix to deploy it.

Figure 3-21 Checkpoint answers



Using IBM Bluemix DevOps services

This unit covers the following topics:

- ▶ IBM Bluemix DevOps overview
- ▶ Web IDE (Edit Code)
- ▶ Source Code Management (SCM)
- ▶ Automated Delivery Pipeline (Build & Deploy)
- ▶ Agile Planning Tools (Track & Plan)

4.1 What you should be able to do

After completion of this unit, you should be able to:

- ▶ Describe DevOps.
- ▶ Describe IBM Bluemix DevOps Services.
- ▶ Describe the capabilities of IBM DevOps Services.
- ▶ Identify the Web IDE features in IBM Bluemix DevOps.
- ▶ Describe how to connect the Git repository client to your IBM Bluemix DevOps Services project.
- ▶ Explain the pipeline build and deploy processes that IBM Bluemix DevOps Services use.
- ▶ Describe how IBM Bluemix DevOps Services integrate with the IBM Bluemix cloud.
- ▶ Describe the agile planning tools in IBM Bluemix.

4.2 References

The following publications are useful for further research on the topic presented in this unit:

- ▶ Documentation about Agile:
 - <http://www.agilenutshell.com/>
 - <http://agilemanifesto.org/>
- ▶ Open source Git project:
<https://www.git-scm.com/>
- ▶ Overview of IBM Bluemix DevOps Services:
<https://www.youtube.com/watch?v=VJesera9jR0>
- ▶ Tutorials/Documentation for Bluemix DevOps Services:
 - <https://www.ibm.com/cloud-computing/bluemix/devops/>
 - <https://hub.jazz.net/tutorials/>
 - <https://console.ng.bluemix.net/docs/develop/bluemixlive.html>

4.3 What is DevOps?

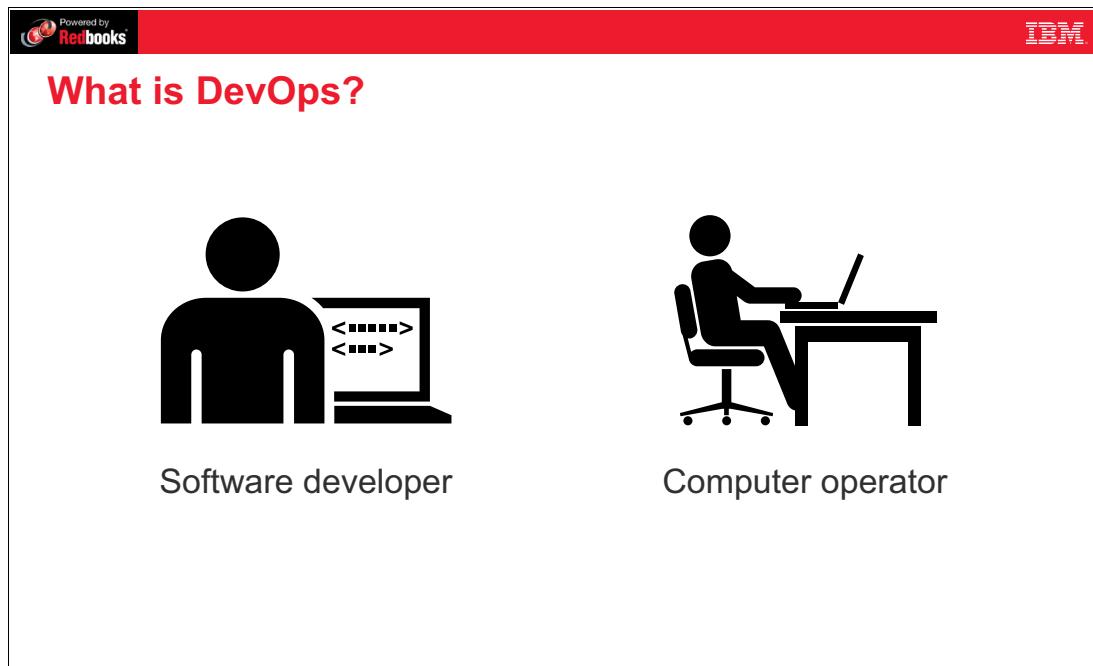


Figure 4-1 What is DevOps?

Notes:

Before DevOps, the following teams were responsible for code delivery:

- ▶ Development team: This team designed the code, delivered new features, fixed bugs, and tested the code.
- ▶ Operations team: This team deployed the code to the different environments, maintained the uptime on production, and diagnosed failures.

The term *DevOps* comes from merging the words *development* and *operations*. With DevOps, both teams work together to support the software lifecycle, from code design to deployment to production. This collaboration includes automating all of the tasks.

The developer only has to develop code, and can then rely on the DevOps platform to do the rest. For example, the developer can automate the build process, code testing, and deployment to the different environments.

4.4 Benefits of DevOps

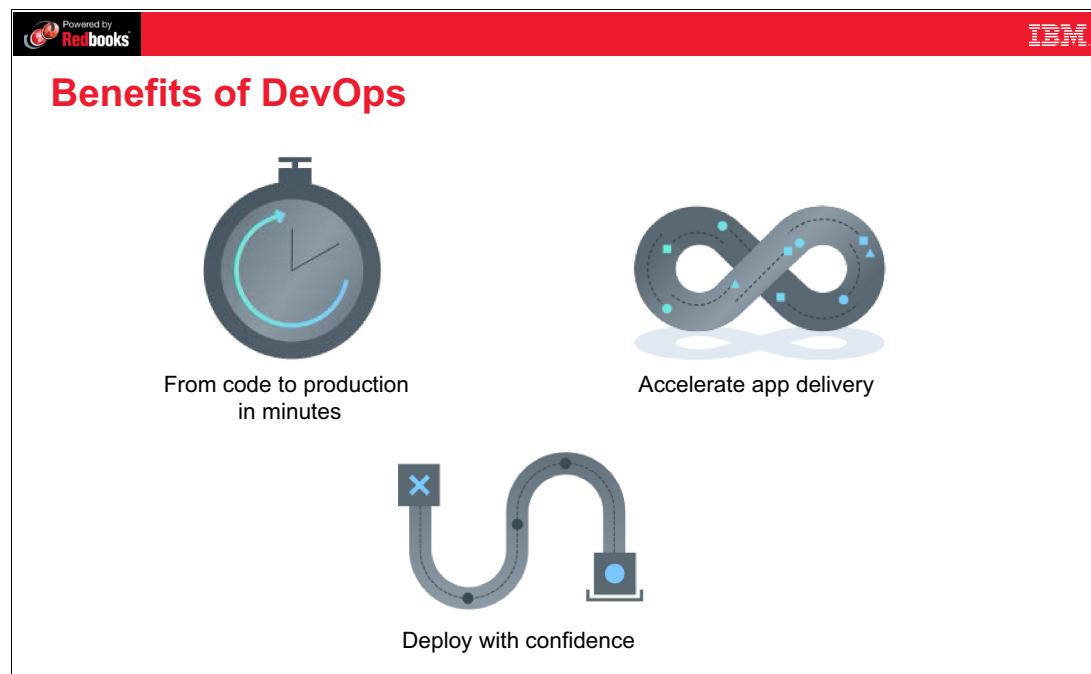


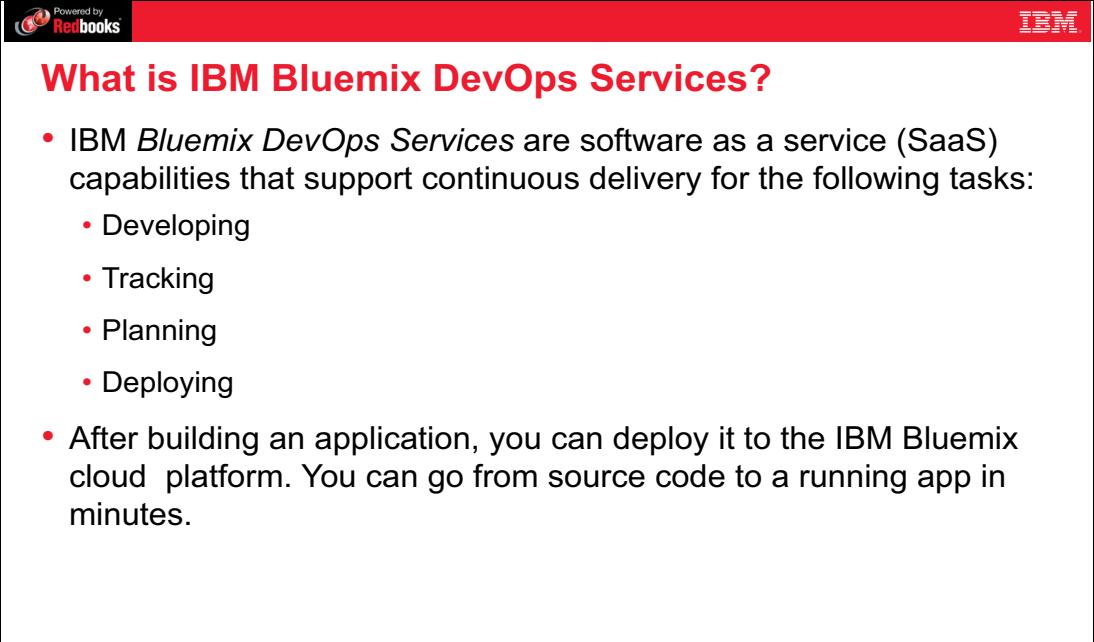
Figure 4-2 Benefits of DevOps

Notes:

DevOps provides the following benefits, among others:

- ▶ From code to production in minutes: Work on your own or use the collaboration tools to work with a team. In minutes, you can go from source code to a running app.
- ▶ Accelerate app delivery: Innovate like a startup, and scale for the enterprise. You can host an open source project, run a hackathon, or start a skunkworks project. Plan anything, even monthly meetings. DevOps code is continuously delivered and integrated into the main shared repository.
- ▶ Deploy with confidence: Automatically deploy your projects whenever a project member pushes code to your repository. Simply deploy files as they are pushed, or you can configure more advanced build options.

4.5 What is IBM Bluemix DevOps Services?



The slide has a red header bar with the Redbooks logo on the left and the IBM logo on the right. The main title 'What is IBM Bluemix DevOps Services?' is in bold red text. Below it is a bulleted list of features.

What is IBM Bluemix DevOps Services?

- IBM *Bluemix DevOps Services* are software as a service (SaaS) capabilities that support continuous delivery for the following tasks:
 - Developing
 - Tracking
 - Planning
 - Deploying
- After building an application, you can deploy it to the IBM Bluemix cloud platform. You can go from source code to a running app in minutes.

Figure 4-3 What is IBM Bluemix DevOps Services?

Notes:

IBM Bluemix DevOps Services are software as a service (SaaS) capabilities that support continuous delivery for:

- ▶ Developing
- ▶ Tracking
- ▶ Planning
- ▶ Deploying

With IBM Bluemix DevOps Services, you can develop, track, plan, and deploy software in one place.

IBM Bluemix DevOps Services complement the IBM Bluemix cloud platform. Your organization can produce working applications in minutes, starting from an IBM Bluemix boilerplate or sample application.

4.6 What services does IBM Bluemix DevOps provide?

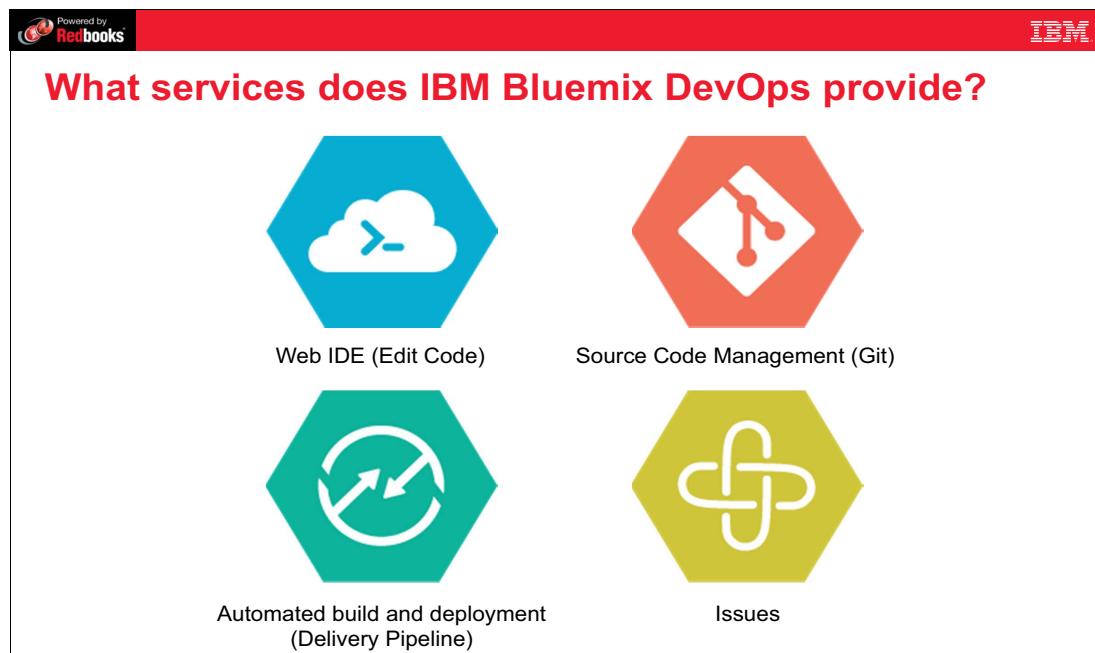


Figure 4-4 What services does Bluemix DevOps provide?

Notes:

IBM Bluemix DevOps provides a comprehensive set of features for the operations and software development teams. As part of the default IBM Bluemix DevOps toolchains, the following tools are available:

- ▶ **Web IDE:** You can develop your code (for example, Node.js, Java, or any other code) in this web-based integrated development environment (Web IDE). No other software is necessary, other than your web browser. The Web IDE saves your current work in a cloud-based file directory, which is known as the *local repository*. IBM Bluemix provides Eclipse Orion as the Web IDE. Although the use of the Web IDE works well, you can still use a desktop IDE, such as Eclipse with DevOps services.
- ▶ **Source control system (SCM):** The local repository is a copy of your latest edits before you submit your work to a SCM. IBM Bluemix DevOps creates a Git repository as a change management system. The Git repository, which is known as the *remote repository*, is hosted on the IBM Bluemix DevOps Services infrastructure.
- ▶ **IBM Bluemix Delivery Pipeline:** This service automates the process of building and deploying your code as an IBM Bluemix application. You can also configure the build, deploy, or test scripts within the Web IDE.
- ▶ **Issue Tracking:** This tool tracks the progress of your development and IBM Bluemix DevOps services.

4.7 Additional DevOps tool Integration

The page is titled "Additional DevOps tool Integration". It features a grid of nine tool icons, each with a brief description and an "IBM" or "Third Party" label.

Tool	Description	Provider
Github Enterprise Whitewater	Store and manage your code and engage in social coding.	IBM
Jenkins	Build, deploy, and automate any project.	Third Party
JIRA	Manage projects and track issues.	Third Party
Nexus	Store build artifacts in your Nexus repository.	Third Party
PagerDuty	When major issues arise, send alerts and get alerts.	Third Party
Rational Team Concert	Manage your team and resources.	IBM
Sauce Labs	Automate continuous integration testing for your	Third Party
Slack	Coordinate and collaborate on your project.	Third Party
SonarQube	Inspect the quality of your code.	Third Party
UrbanCode Deploy	Gain insights into deployments that are done by	IBM Beta
Other Tool	Create a custom integration with any tool.	

Figure 4-5 Additional DevOps tool Integration

Notes:

In addition to the four DevOps core elements of toolchains that are shown in Figure 4-4 on page 90, you can integrate Slack, Jira, Jenkins, and more, as shown in Figure 4-5.

4.8 Setting up an IBM Bluemix DevOps Services project

The screenshot shows a web interface for setting up a DevOps services project. At the top, there's a red header bar with the 'Powered by Redbooks' logo on the left and the 'IBM' logo on the right. Below the header, the main title 'Setting up an IBM Bluemix DevOps services project' is displayed in large red text. A sub-instruction follows in black text: 'To create an IBM Bluemix DevOps services project, complete these steps:'. Three numbered steps are listed in red: 1. In the Bluemix Dashboard, select your application. 2. Click **Overview** in the left navigation bar. 3. In the Continuous delivery panel, click **Enable**. Below these instructions is a light gray rectangular box containing a 'Continuous delivery' section. It displays the message 'Continuous delivery is not enabled for this app.' and a descriptive text: 'Enable continuous delivery to automate builds, tests, and deployments through the Delivery Pipeline, GitHub, and more.' At the bottom of this section is a blue 'Enable' button.

Figure 4-6 Setting up a Bluemix DevOps Services project

Notes:

After you enable continuous delivery for the project, a new tab opens to configure Continuous Delivery Toolchains.

4.9 New tab opens to create Continuous Delivery toolchains

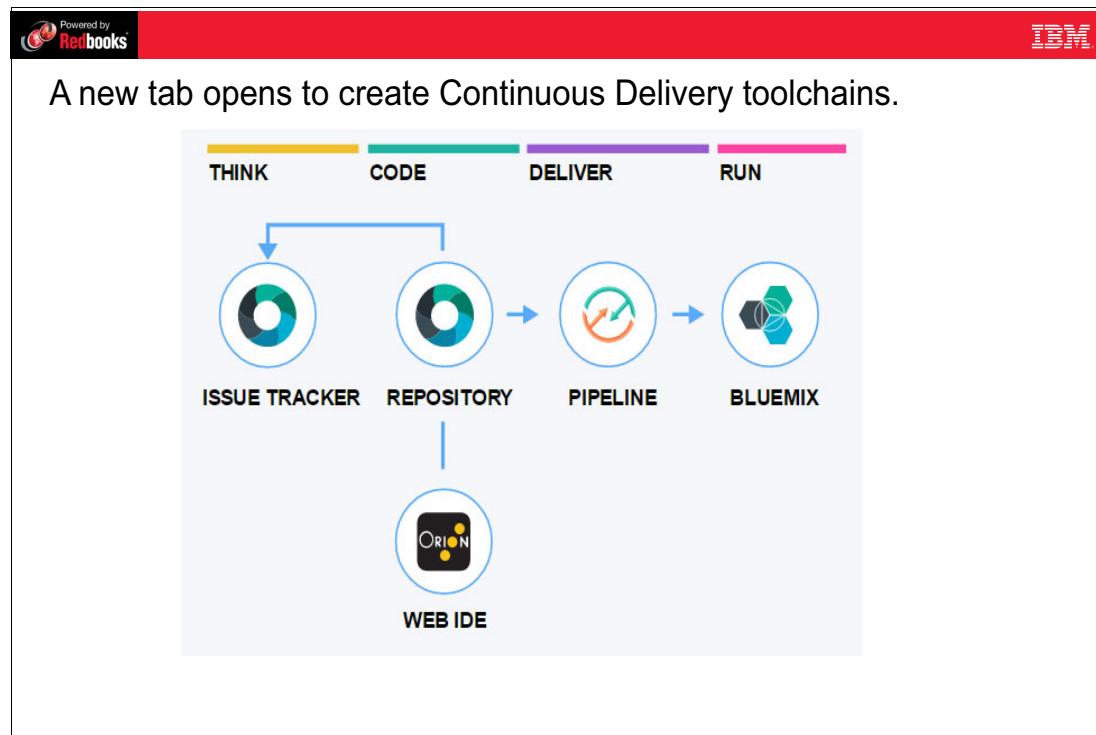


Figure 4-7 Creating Continuous Delivery chains

Notes:

A new window opens that shows the following main stages for the Continuous Delivery toolchains:

- **THINK:** This stage is for planning for the application by creating bugs, tasks, or ideas by using the Issue Tracking tool.
- **CODE:** This stage is for the implementation of the application by providing a GIT repository as source code management system, and a Web IDE (Eclipse Orion) to edit your code online. In the repository, you can specify whether to clone a repository or start from scratch by selecting **New** in the repository type.
- **DELIVER:** This stage is for configuring the delivery pipeline. It allows you to specify automatic build, deployment, and testing of your code after a developer pushes a new code to the Git repository.
- **RUN:** The output of this stage is to run the application in the IBM Bluemix environment.

4.10 Web IDE: Edit Code features

The screenshot shows a web-based interface for the IBM Bluemix DevOps platform. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "Web IDE: Edit Code features" is displayed in bold red text. To the right of the title is a graphic element featuring a blue circle containing a black square with the word "ORION" and three yellow dots. Below this graphic, the text "WEB IDE" is written in bold black capital letters.

- Web IDE:
 - Eclipse Orion
 - No installations, just start coding now
- Bluemix Live Sync:
 - Live Edit for Node.js applications
 - Debug for Node.js applications
 - Desktop Sync

Figure 4-8 Web IDE: Edit Code features

Notes:

The Edit Code features in IBM Bluemix DevOps provide the following capabilities:

- ▶ Web IDE capabilities:
 - Uses Eclipse Orion Web IDE.
 - No installations, just code now.
 - The Edit Code feature provides a workspace to develop source code and configuration files.
 - It provides a fully featured environment for writing your application code by using your web browser.
 - It provides rich code completion capabilities for CSS, HTML, and JavaScript.
 - You can deploy, stop, and run applications from the Run bar.
 - You can also view the logs from the Run bar.
- ▶ Bluemix Live Sync features:
 - Live Edit (currently available for Node.js applications only): Allows you to make changes to your application from the Web IDE without the need to redeploy it.
 - Debug (currently available for Node.js applications only): When a Node.js application is in Live Edit mode, you can shell into it and debug it. You can edit code dynamically, insert breakpoints, step through code, restart the runtime, and more by using the Node Inspector debugger.
 - Desktop Sync: You can synchronize any desktop directory tree with a cloud-based project workspace similar to the way Dropbox works.

4.11 Web integrated development environment

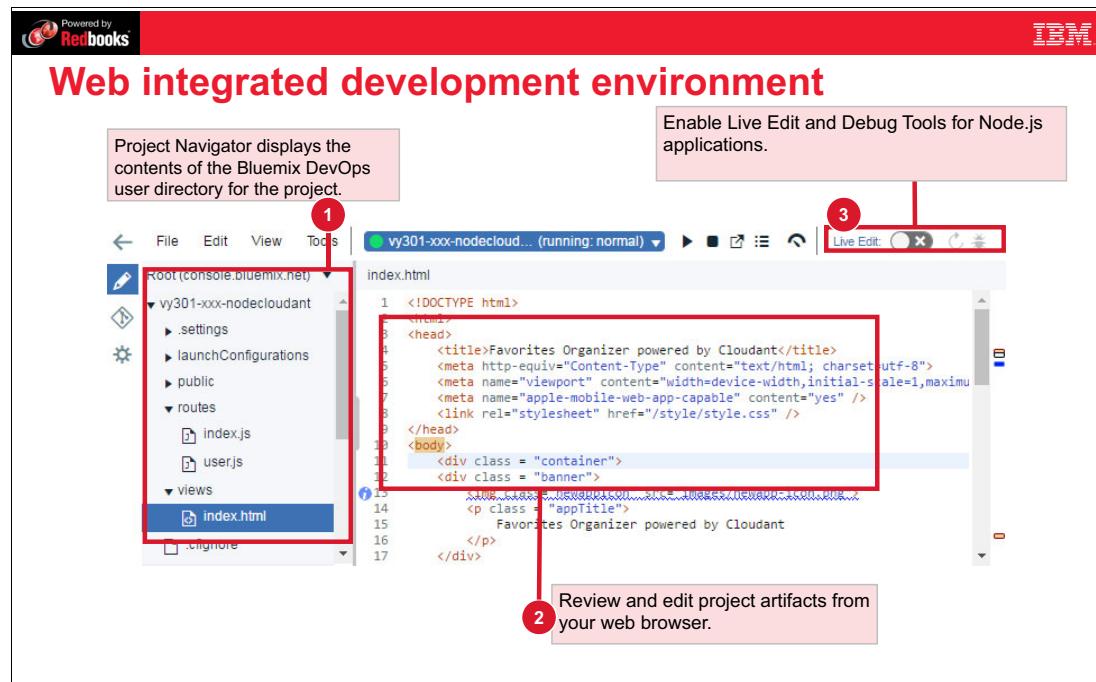


Figure 4-9 Web integrated development environment: Overview

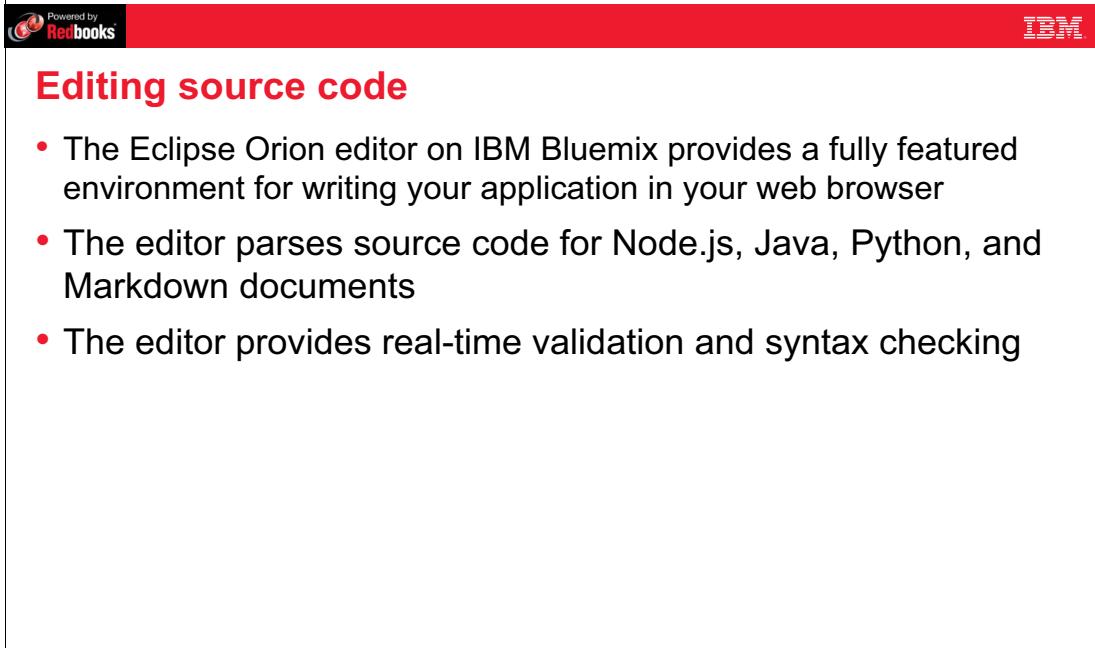
Notes:

The Edit Code perspective for the Web IDE in Bluemix DevOps is shown in Figure 4-9.

When you select a file, the editor displays the contents on the right side of the page. You can edit source code, configuration files, and other artifacts directly within your web browser.

This view is for your local workspace on IBM Bluemix DevOps. To commit the code changes, switch to the GIT view that is described next.

4.12 Editing source code



The screenshot shows a web-based development environment. At the top, there's a red header bar with the "Powered by Redbooks" logo on the left and the "IBM" logo on the right. Below the header, the main content area has a white background. On the left side of the content area, there's a sidebar or navigation panel. The main content title is "Editing source code" in bold red text. To the right of the title is a bulleted list of features:

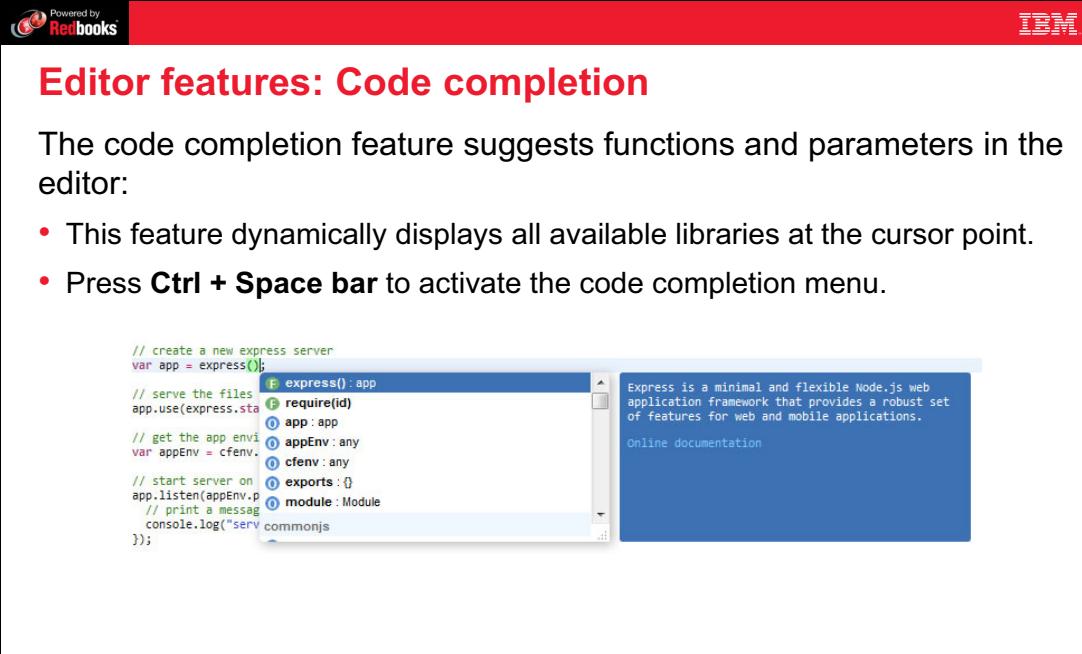
- The Eclipse Orion editor on IBM Bluemix provides a fully featured environment for writing your application in your web browser
- The editor parses source code for Node.js, Java, Python, and Markdown documents
- The editor provides real-time validation and syntax checking

Figure 4-10 *Editing source code in Bluemix DevOps Services*

Notes:

When you open an IBM SDK for Node.js script file, the editor provides real-time validation and syntax checking of the source code. It uses tools, such as JSHint, which is a JavaScript code quality tool that helps to detect errors and potential problems in the code. A preview window flags warnings and errors on the right side of the page.

4.13 Editor features: Code completion



The screenshot shows a code editor window with the following code:

```
// create a new express server
var app = express();
// serve the files
app.use(express.static('public'));
// get the app environment variable
var appEnv = cfenv.get('app');
// start server on port
app.listen(appEnv.port, function() {
  // print a message to the log
  console.log(`Server running on port ${appEnv.port}`);
});
```

A code completion dropdown menu is open at the cursor position, showing suggestions for the word 'express'. The suggestion 'express(): app' is highlighted. To the right of the dropdown, a tooltip provides information about the Express module:

Express is a minimalist and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Online documentation

Figure 4-11 Editor features: Code completion

Notes:

To run the code completion shortcut, place your cursor within the editor and press **Ctrl + Space bar**. This feature displays the libraries that are available at the cursor point, including third-party modules that you imported in the script. For example, as shown in Figure 4-11, the code completion feature displays functions and templates for the Express web application framework for Node.js because this point is the cursor point.

You can use templates to add blocks of code for common tasks, such as error handling or object creation.

4.14 Editor features: Run bar

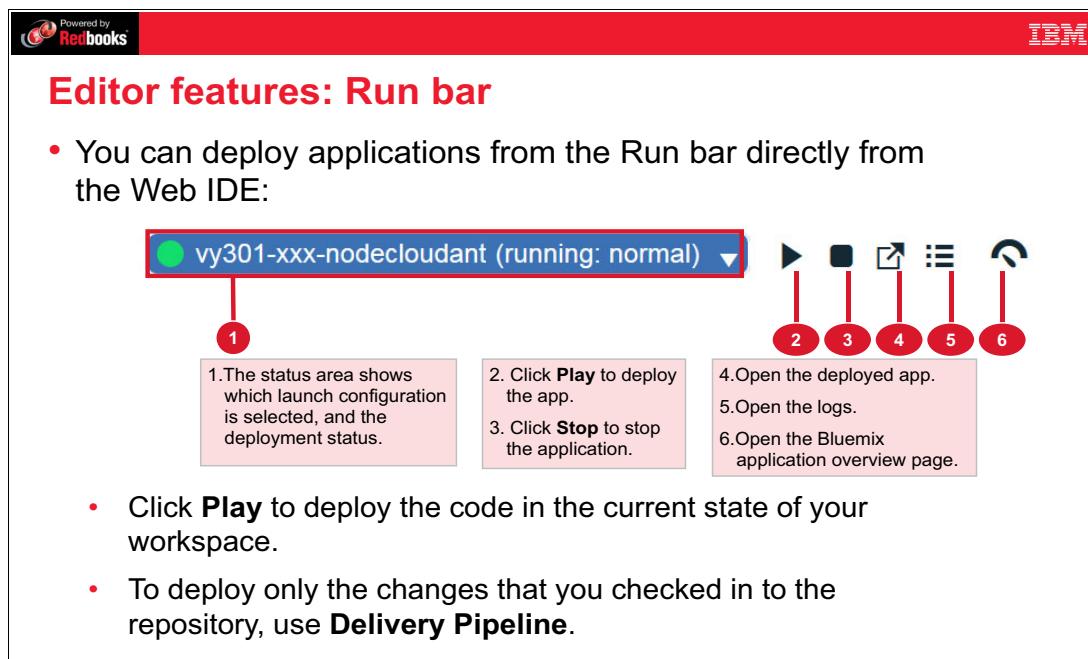


Figure 4-12 Editor features: Run bar

Notes:

You can quickly build and deploy your application to a test environment on IBM Bluemix without committing your code to the Git repository. The following features are highlighted in Figure 4-12:

1. The status area displays the launch configuration the run bar uses during the build and deploy task.
2. Click the **Play** icon from the Run bar to build and deploy the code in your user directory to your IBM Bluemix account.
3. Click the **Stop** icon to stop the application.
4. Click the **Open Deployed App** icon to open the application route.
5. Click the **Open Logs** icon to open application logs.
6. Click the **Access** icon to access the dashboard from the same bar.

4.15 Bluemix Live Sync features

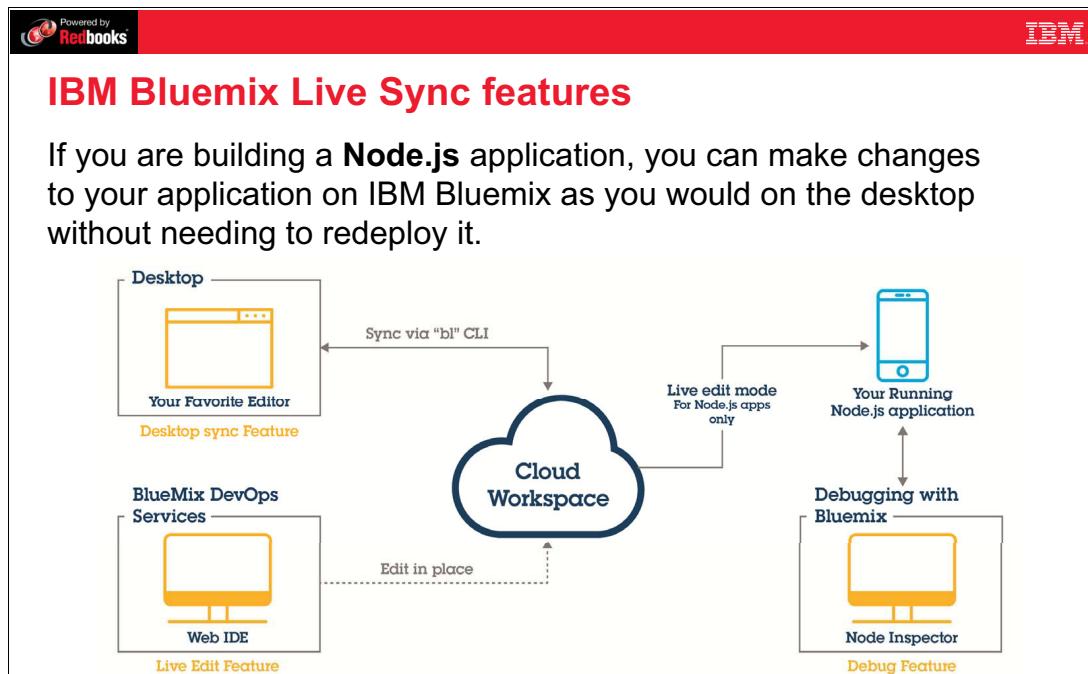


Figure 4-13 Bluemix Live Sync features

Notes:

If you are building a Node.js application, you can use IBM Bluemix Live Sync to quickly update the application instance on IBM Bluemix and develop as you do on the desktop without redeploying.

When you make a change, you can see that change in your running IBM Bluemix application immediately (without the need to recompile and redeploy). In addition, you can access your application from the browser on the desktop or a mobile device.

IBM Bluemix Live Sync works from the command line and in the Web IDE. You can debug applications that are written in Node.js by using Bluemix Live Sync.

IBM Bluemix Live Sync consists of the following features:

- ▶ Desktop Sync

You can synchronize any desktop directory tree with a cloud-based project workspace similar to the way Dropbox works. The Web IDE directly edits the same cloud-based workspace, so both stay in sync.

Desktop Sync works for any kind of application. To use Desktop Sync, you must download and install the BL command-line interface, which is the IBM Bluemix Live Sync command line. Desktop sync works with all types of applications.

- ▶ Live Edit (only for Node.js applications)

You can make changes to a Node.js application that is running in IBM Bluemix and test them in your browser immediately. Any changes that you make in a synchronized desktop directory or in the Web IDE are propagated to the application's file system instantly.

- ▶ Debug (only for Node.js applications)

When a Node.js application is in Live Edit mode, you can debug it on the Web IDE. You can edit code dynamically, insert breakpoints, step through code, restart the runtime, and more. You must use a Chrome browser to use the Debug feature.

4.16 Source control with a Git repository

The screenshot shows a section titled "Source control with a Git repository". It includes a bulleted list of options for managing source code:

- IBM Bluemix DevOps toolchain gives you many options for source control:
 - When you enable continuous delivery for a project, a Git repository is created to manage your source code.
 - Alternatively, you can use other source code management repositories, such as GitHub and IBM Rational Team Concert.
- When you develop your application online, you can enter common Git commands through the Web IDE.
- If you develop your application in your own workstation, use a Git client to synchronize your own workspace and push your changes to the IBM Bluemix DevOps Git repository.
- For more information, see the open source Git project:
<http://www.git-scm.com/>

Figure 4-14 Source control with a Git repository

Notes:

By default, enabling Continuous Delivery for a project creates DevOps toolchains for your project. The Toolchains feature includes a Git repository that is based on GitLab. Git is an open source change management system.

If your organization uses IBM Rational® Team Concert™, you can set up an IBM Jazz™ SCM environment to manage your code. You can also point your IBM Bluemix DevOps services project to an external, third-party Git repository provider, such as GitHub.

The Git repository perspective in the IBM Bluemix DevOps services web IDE supports common Git commands to manage your code. You can also develop your application on your own workstation and commit your changes to the Git repository with a standard Git client.

For more information, see the open-source Git project at this website:

<http://www.git-scm.com>

4.17 Git repository overview

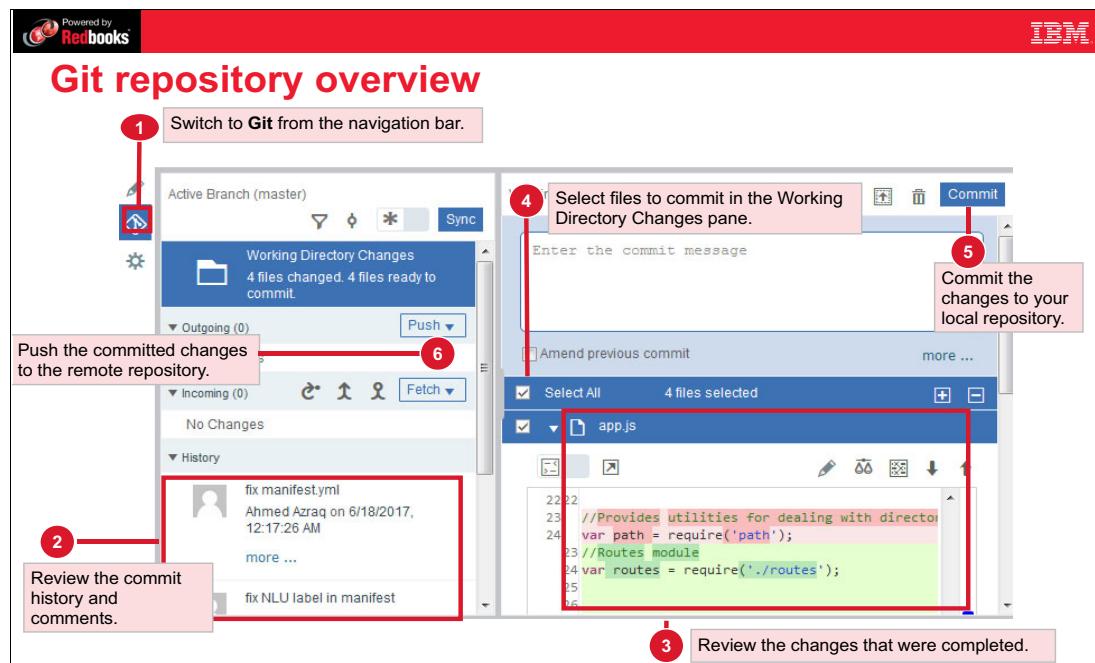


Figure 4-15 Git repository overview

Notes:

The numbers that are shown in Figure 4-15 correspond to the following steps:

1. Switch to the Git perspective from the navigation bar on the left side of the web IDE.
2. You can review the commit history with the time/date for each commit. You can perform actions for each commit, such as view the files committed, and revert changes that were introduced by any commit.
3. The Working Directory Changes pane detects any updated files in the IBM Bluemix DevOps user directory. Click a file to review the changes that were made in that file.
4. Select the files that you want to commit and enter a descriptive comment about the change.
5. Click **Commit** to commit the changes to your local repository.
6. The Outgoing pane lists the files that you want to commit to the remote repository. View the outgoing changes; then, click **Push** to push the committed changes to the remote repository.

If another user updated the files in the remote repository, the Incoming pane lists the updated files.

4.18 Connect a Git client to your IBM Bluemix DevOps project

Powered by
Redbooks

IBM

Connect a Git client to your IBM Bluemix DevOps project

- If you want to develop your application on your own workstation, use a Git client to save your changes to the Bluemix DevOps Git repository.
- Issue the **git clone** command with the Git repository address to retrieve a copy of the source code in the remote repository:
`git clone https://hub.jazz.net/git/warrenf/warrenf-nodesample`
- Issue the **status** command to check whether your local copy is synchronized with the remote repository:
`git status`
- Issue the **commit** and a **push** command to push changes from the local repository to the remote repository:
`git add -A`
`git commit -m 'Adding project description.'`
`git push`

Figure 4-16 Connect a Git client to your IBM Bluemix DevOps project

Notes:

You can still develop your application on your own local workstation. Start by retrieving a copy of the IBM Bluemix application source code by using the **git clone** command. After you have a copy of the source code, use a text editor or integrated development environment to write and test your application.

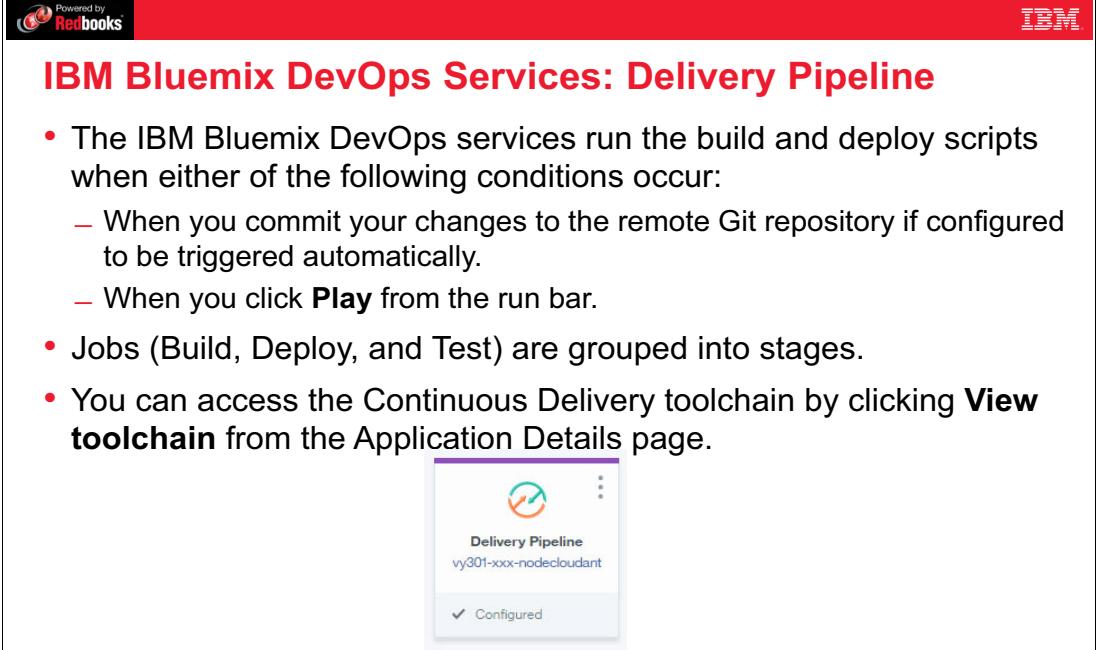
To check whether another developer on your team updated the source code, run the **git status** command.

When you want to save your updated source code, run the **git add** command to stage the needed files to be committed.

Then, run the **git commit** command and save a message for the history log to commit all of the added files to your local repository.

Finally, run **git push** to send your committed changes to the remote repository.

4.19 IBM Bluemix DevOps Services: Delivery Pipeline



The screenshot shows a red header bar with the IBM logo on the right and the Redbooks logo on the left. Below the header, the title "IBM Bluemix DevOps Services: Delivery Pipeline" is displayed in red. The main content area contains a bulleted list of features:

- The IBM Bluemix DevOps services run the build and deploy scripts when either of the following conditions occur:
 - When you commit your changes to the remote Git repository if configured to be triggered automatically.
 - When you click **Play** from the run bar.
- Jobs (Build, Deploy, and Test) are grouped into stages.
- You can access the Continuous Delivery toolchain by clicking **View toolchain** from the Application Details page.

Below the list, there is a small screenshot of a delivery pipeline card. The card has a purple header bar with a circular icon containing a checkmark. The main body of the card displays the text "Delivery Pipeline" and "vy301-xxx-nodecloudant". At the bottom, there is a button labeled "Configured" with a checkmark icon.

Figure 4-17 Bluemix DevOps Services: Delivery Pipeline

Notes:

By default, IBM Bluemix DevOps services automatically runs the build and deploy tasks when you commit changes to the Git repository. If you want to quickly test your code without committing your changes, click **Play** from the Run bar.

4.20 Customizing the delivery pipeline

The screenshot shows a web-based interface for managing delivery pipelines. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "Customizing the delivery pipeline" is displayed in bold red text. To the right of the title is a list of bullet points describing customization options. Further down is a "Deploy Stage" configuration panel with tabs for INPUT, JOBS (which is selected), and ENVIRONMENT PROPERTIES. The JOBS tab shows two stages: "Deploy" and "Test". The "Test" stage is highlighted with a blue circle around its icon and has a checkmark inside a box. Below the configuration panel is a caption.

Customizing the delivery pipeline

- When you commit changes to the Git repository, IBM Bluemix DevOps pushes out the changes to your IBM Bluemix application. Edit the delivery pipeline to customize the deployment tasks that run when you commit your changes.
- You can customize the delivery pipeline in multiple stages and multiple jobs within a stage:
 - Create multiple stages that deploy code to the testing, staging, and production environments.
 - Add jobs within a stage to run automated tests of your code.

Deploy Stage

INPUT JOBS ENVIRONMENT PROPERTIES

Deploy Stage

Deploy Test ADD JOB

Figure 4-18 Customizing the delivery pipeline

Notes:

You can configure each stage to have one or more job (Build, Deploy, and Test). You can also configure a stage to have more than one job of the same type. For example, you can have a stage that features one Build job, one Deploy job, and two Test jobs.

4.21 Example: Default delivery pipeline

The screenshot shows the IBM Bluemix DevOps interface with two stages: Build Stage and Deploy Stage.

Build Stage:

- LAST INPUT: Last commit by Ahmed Azraq 23h ago ([fix manifest.yml](#))
- JOB: Build Passed 23h ago
- LAST EXECUTION RESULT: Build 3

Deploy Stage:

- LAST INPUT: Stage: Build Stage / Job: Build Build 3
- JOB: Deploy Passed 23h ago
- LAST EXECUTION RESULT: vy102-XXX-express ([vy102-XXX-express.mybluemix.net](#))

A pink callout box on the right side lists three points:

1. The default delivery pipeline takes the files that are pushed to the remote Git repository.
2. Build Stage by default contains a Build job that builds the code artifact.
3. Deploy stage by default contains a Deploy job that deploys the built application to your Bluemix space.

Figure 4-19 Example: Default delivery pipeline

Notes:

The default build script is simple: It takes the files that are pushed to the remote Git repository and triggers IBM Bluemix to build your code in the server runtime. For example, IBM Bluemix downloads the IBM SDK for Node.js modules that you imported into your script.

You can customize the settings for the server runtime through the `manifest.yml` file.

After the build stage completes successfully, the deploy stage runs. IBM Bluemix DevOps deploys the built files to your IBM Bluemix space.

4.22 Configuring the Build Stage

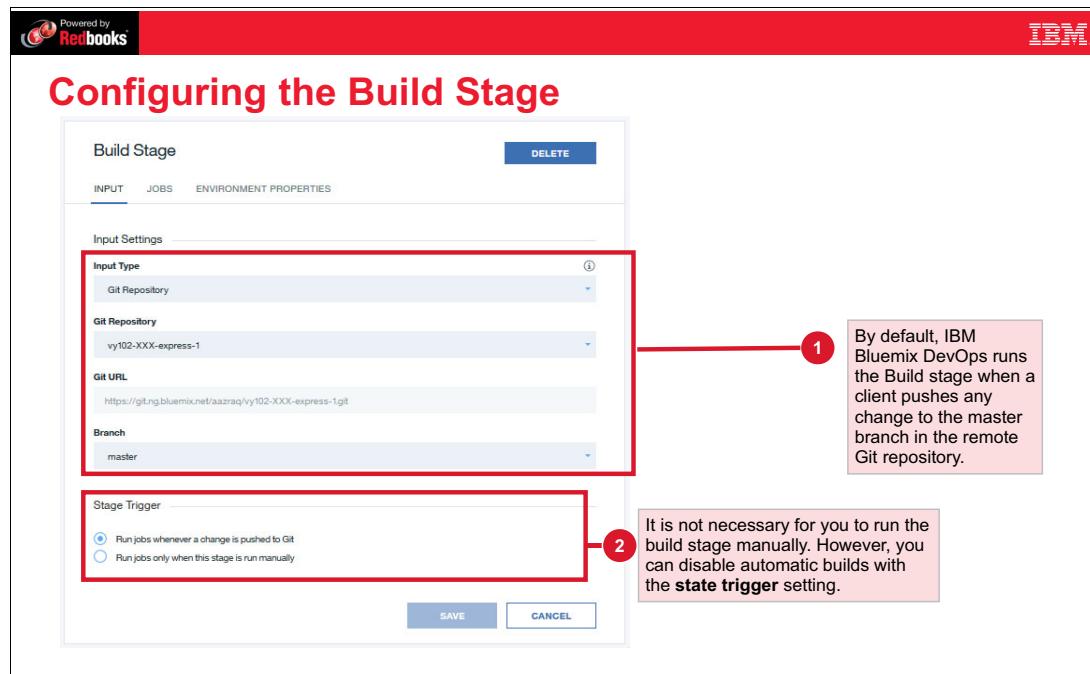


Figure 4-20 Configuring the Build Stage

Notes:

By default, IBM Bluemix DevOps runs the Build Stage when a client pushes any change to the master branch in the remote Git repository.

If you do not want to automatically push your changes to your IBM Bluemix account, change the State trigger setting to **Run jobs only when this stage is run manually**.

4.23 Configuring Build jobs

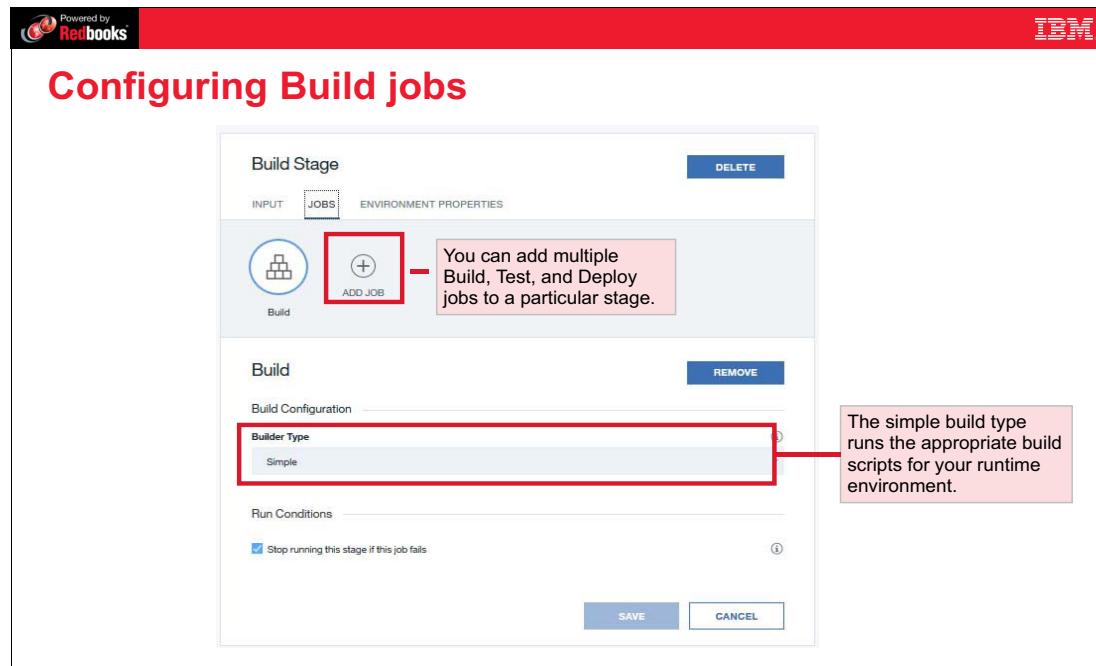


Figure 4-21 Configuring build jobs

Notes:

You can extend the features of the build stage by adding jobs. Although IBM Bluemix DevOps names this stage configuration the Build Stage, you can add build, test, or deploy jobs.

As of this writing, Builder Type supports the following types:

- ▶ Simple
- ▶ Ant
- ▶ Gradle
- ▶ Grunt
- ▶ IBM Container Service
- ▶ IBM Globalization Pipeline
- ▶ Maven
- ▶ npm: For Node.js projects
- ▶ Shell Script

The simple build type runs the appropriate build scripts for your runtime environment. For example, the IBM SDK for Node.js runtime resolves modules that your application requires.

4.24 Configuring Deploy jobs

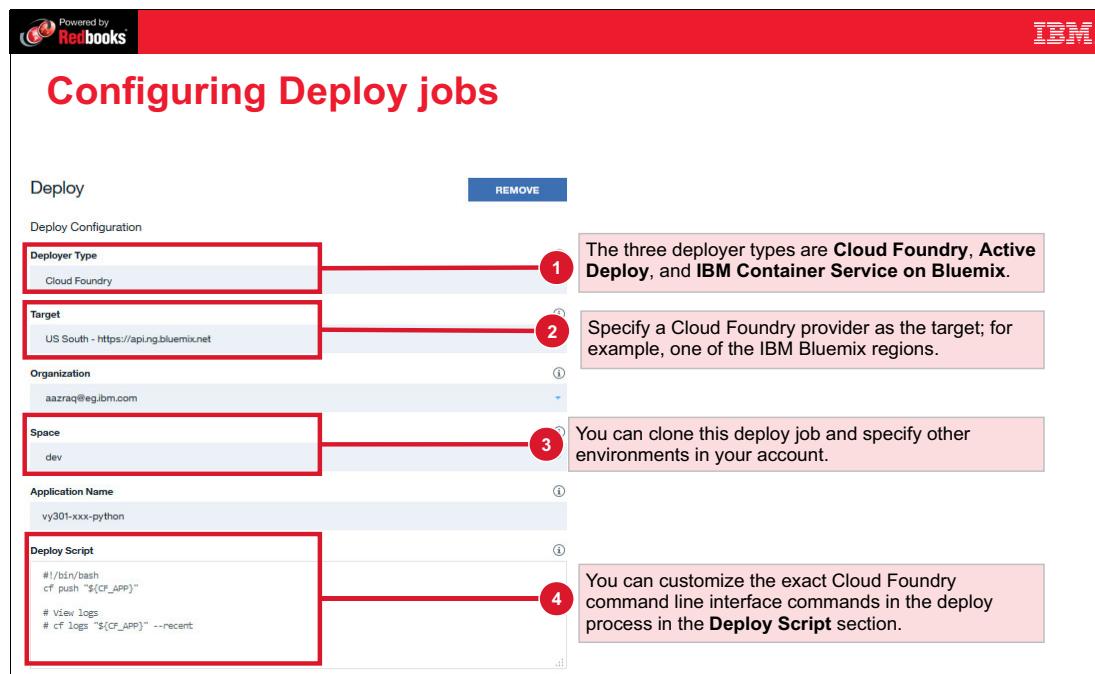


Figure 4-22 Configuring Deploy jobs

Notes:

In this example, you create a deploy task that pushes the application to a specific space on IBM Bluemix.

The following Deployer Types are available:

- ▶ Cloud Foundry: Deploys applications to Cloud Foundry servers, such as IBM Bluemix. This type is the default deployer type for the deploy job.
- ▶ Active Deploy: You can update running applications with no downtime by using the IBM Active Deploy service in your pipeline. The Active Deploy - Begin job includes a script that starts the deployment process to add a new instance of your new application until both versions of your application are live in production. Active Deploy - Complete ends that deployment process and removes that extra instance of the application if the test phase was successful.
- ▶ IBM Containers on Bluemix: Provides a set of default deployment patterns for containers on the IBM Containers service. Containers are typically built by using the Docker Builder.

In this example, your application is pushed to the development space in your organization. You can also publish your application to your staging and production spaces.

You can customize the Cloud Foundry command-line interface (cf CLI) commands in the deploy process in the Deploy Script section. The default deploy action is equivalent to running `cf push` from the CLI. You can add custom shell script commands as well (`${CF_APP}` as shown in Figure 4-22 refers to the application name).

4.25 Configuring Test job

The screenshot shows the 'Configuring Test job' interface in IBM Bluemix. At the top, there's a navigation bar with the Redbooks logo and the IBM logo. Below the navigation bar, the title 'Configuring Test job' is displayed. The main area is divided into two sections: 'Deploy Stage' and 'Test'. In the 'Deploy Stage' section, there are two circular icons: one for 'Deploy' (with an upward arrow) and one for 'Test' (with a checkmark). Between them is a right-pointing arrow. Below these icons are the labels 'Deploy' and 'Test'. To the right of the 'Test' icon is a blue '+' button labeled 'ADD JOB'. Above the 'Test' section is a blue 'DELETE' button. The 'Test' section has a 'REMOVE' button at the top right. Under the 'Test' section, there are two configuration fields: 'Tester Type' (set to 'Simple') and 'Test Command' (containing the text '#!/bin/bash # invoke tests here').

Figure 4-23 Configuring Test job

Notes:

After the Deploy job is completed, you can add and configure a Test job to test your deployed application.

The following Tester types are available:

- ▶ Simple
- ▶ Advanced Tester
- ▶ IBM AppScan® Dynamic Analyzer
- ▶ IBM Security Static Analyzer
- ▶ IBM Vulnerability Adviser

4.26 Example: A successful build and deploy result

The screenshot shows the IBM Bluemix DevOps interface with two main sections: 'Build Stage' and 'Deploy Stage'.
In the 'Build Stage' section:

- A green header bar says 'STAGE PASSED'.
- The 'LAST INPUT' section shows a commit by 'Ahmed Azraq' from 'git://github.com/ahmedazraq/HelloWorld' made 23 hours ago.
- The 'JOBS' section shows a 'Build' job that passed 23 hours ago, indicated by a green checkmark and the text 'Passed 23h ago'.
- The 'LAST EXECUTION RESULT' section shows a 'Build 3' entry with a green checkmark and the text 'vy102-XXX-express vyt02-XXX-express.mybluemix.net'.

Annotations with red circles and arrows:

- Annotation 1: Points to the 'Build' job status in the 'JOBS' section of the Build Stage.
- Annotation 2: Points to the 'Deploy' job status in the 'JOBS' section of the Deploy Stage.
- Annotation 3: Points to the 'vy102-XXX-express' entry in the 'LAST EXECUTION RESULT' section of the Deploy Stage.

In the 'Deploy Stage' section:

- A green header bar says 'STAGE PASSED'.
- The 'LAST INPUT' section shows the 'Stage: Build Stage / Job: Build'.
- The 'JOBS' section shows a 'Deploy' job that passed 23 hours ago, indicated by a green checkmark and the text 'Passed 23h ago'.
- The 'LAST EXECUTION RESULT' section shows a 'Build 3' entry with a green checkmark and the text 'vy102-XXX-express vyt02-XXX-express.mybluemix.net'.

Figure 4-24 Example: A successful build and deploy result

Notes:

The default settings for the delivery pipeline are the following stages:

- ▶ Build Stage:
 - Input: This stage is triggered whenever a change is pushed to Git.
 - Jobs: Simple Builder Type.
- ▶ Deploy Stage:
 - Input: This stage is triggered whenever the Build Stage is completed successfully. It takes as input the build artifacts that were produced from the Build Stage.
 - Jobs: Deploys the application to IBM Bluemix.

In this example, all of the jobs in the Build Stage and Deploy Stage were completed successfully, as shown in the JOBS pane of the Build Stage and Deploy Stage.

The LAST EXECUTION RESULT pane shows that IBM Bluemix DevOps successfully deployed the application to your IBM Bluemix account. It also shows that the application is running on IBM Bluemix.

You can also check the build logs in the JOBS pane in the Build Stage column, the deploy logs in the JOBS pane in the Deploy Stage column, and the runtime logs from LAST EXECUTION RESULT pane in Deploy Stage column.

You can also clone the Deploy Stage and deploy the application to any number of spaces in your IBM Bluemix account by clicking the settings wheel icon at the top of Deploy Stage and selecting **Clone Stage**.

4.27 Issue Tracking tool

The screenshot shows the 'Issue Tracking tool' section of the IBM DevOps toolchain. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below the header, the title 'Issue Tracking tool' is displayed in large red font. A sub-header 'DevOps toolchains includes a tool to track issues' follows. On the left, there's a text block explaining the purpose of the Issue Tracker and a 'New issue' button. To the right, there's a sidebar with a 'Issues' icon and the identifier 'vy301-xxx-nodecloudant'. A checkmark indicates it is 'Configured'. Below the sidebar is a colorful illustration of a rocket launching from a platform, with a magnifying glass focusing on a target, symbolizing tracking and resolution.

Figure 4-25 Issue Tracking tool

Notes:

The Issue Tracking tool (*Issue Tracker* as shown Figure 4-25) is used to track items (issues) that require resolution or improvements in a project.

Issues can be bugs, tasks, or ideas to be discussed. Also, issues are searchable and filterable.

4.28 Creating an issue

The screenshot shows a web-based form titled "Creating an issue". At the top, there's a red header bar with the "Powered by Redbooks" logo on the left and the "IBM" logo on the right. Below the header, the main title "Creating an issue" is displayed in red. A sub-instruction "To create a bug or task, click **New issue**" is shown in bold black text. The form itself has a white background and is titled "New Issue". It contains several input fields:

- Title:** A text input field with a placeholder "Add description templates to help your contributors communicate effectively!" below it.
- Description:** A rich-text editor with "Write" and "Preview" tabs. It includes a toolbar with icons for bold, italic, underline, etc., and a text area with placeholder "Write a comment or drag your files here...". Below the text area, it says "Markdown and slash commands are supported".
- Confidentiality:** A checkbox labeled "This issue is confidential and should only be visible to team members with at least Reporter access."
- Assignee:** A dropdown menu set to "Assignee". To its right are buttons for "Assign to me" and "Select due date".
- Milestone:** A dropdown menu set to "Milestone".
- Labels:** A dropdown menu set to "Labels".

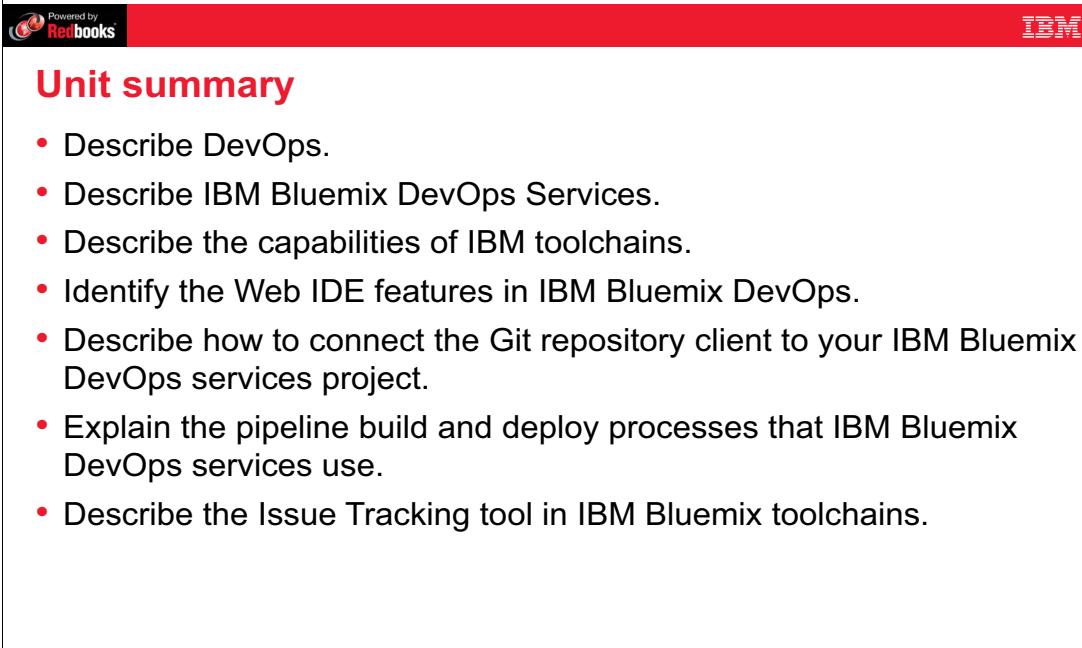
At the bottom of the form are two buttons: a green "Submit issue" button on the left and a "Cancel" button on the right.

Figure 4-26 Creating an issue

Notes:

Complete the form that is shown in Figure 4-26. Then, click **Submit issue** to create the bug or task.

4.29 Unit summary



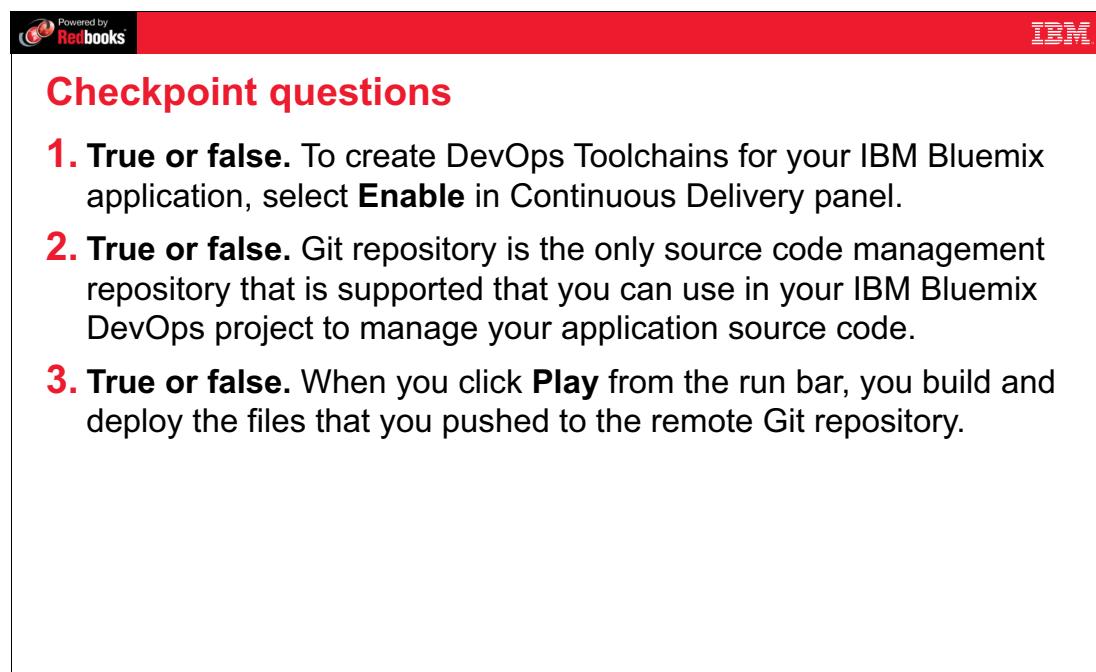
The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a black border.

Unit summary

- Describe DevOps.
- Describe IBM Bluemix DevOps Services.
- Describe the capabilities of IBM toolchains.
- Identify the Web IDE features in IBM Bluemix DevOps.
- Describe how to connect the Git repository client to your IBM Bluemix DevOps services project.
- Explain the pipeline build and deploy processes that IBM Bluemix DevOps services use.
- Describe the Issue Tracking tool in IBM Bluemix toolchains.

Figure 4-27 Unit summary

4.30 Checkpoint questions



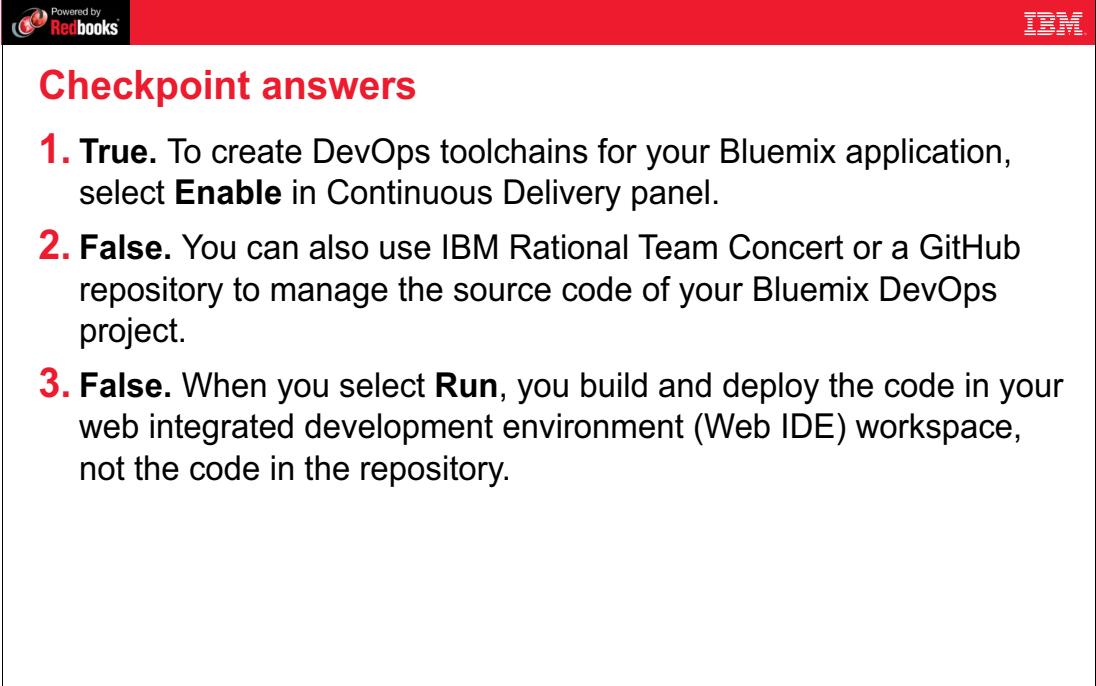
The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. The title "Checkpoint questions" is in bold red font. Below it is a numbered list of three questions.

Checkpoint questions

- 1. True or false.** To create DevOps Toolchains for your IBM Bluemix application, select **Enable** in Continuous Delivery panel.
- 2. True or false.** Git repository is the only source code management repository that is supported that you can use in your IBM Bluemix DevOps project to manage your application source code.
- 3. True or false.** When you click **Play** from the run bar, you build and deploy the files that you pushed to the remote Git repository.

Figure 4-28 Checkpoint questions

4.31 Checkpoint answers



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a red title "Checkpoint answers". Below the title is a numbered list of three items.

- 1. True.** To create DevOps toolchains for your Bluemix application, select **Enable** in Continuous Delivery panel.
- 2. False.** You can also use IBM Rational Team Concert or a GitHub repository to manage the source code of your Bluemix DevOps project.
- 3. False.** When you select **Run**, you build and deploy the code in your web integrated development environment (Web IDE) workspace, not the code in the repository.

Figure 4-29 Checkpoint questions



REST architecture and Watson APIs

This unit covers the following topics:

- ▶ Representational State Transfer (REST)
- ▶ JavaScript Object Notation (JSON)
- ▶ Example: Using REST APIs with Watson

5.1 What you should be able to do

After completion of this unit, you should be able to:

- ▶ Describe the characteristics of REST APIs.
- ▶ Explain the advantages of the JSON data format.
- ▶ Describe an example of REST APIs using Watson.

5.2 References

The following publications are useful for further research on the topic presented in this unit:

- ▶ REST APIs:
 - ECMA-262 ECMAScript language standard, third edition:
<http://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - RESTful Web services: The basics at IBM developerWorks:
<https://www.ibm.com/developerworks/library/ws-restful/>
- ▶ Watch the following videos to learn more about Watson:
 - IBM CEO Ginni Rometty describes a new era in technology and business:
<https://www.youtube.com/watch?v=bMLYKhiZCVI>
 - Scientists at IBM Research have collaborated with 20th Century Fox to create the first-ever cognitive movie trailer for the movie Morgan using Watson:
<https://www.youtube.com/watch?v=gJEzuYnaiw>
 - Watson and the Jeopardy! Challenge:
<https://www.youtube.com/watch?v=P18EdAKuC1U>

5.3 What is REST?

Powered by
Redbooks

IBM

What is REST?

- Representational State Transfer (REST) is an architecture style for building resources on the World Wide Web.
- HTML documents, images, and script files are web resources.
- REST can retrieve, update, or delete a resource by using HTTP methods such as:
 - GET
 - POST
 - DELETE
- A Uniform Resource Identifier (URI) identifies the resource to retrieve or update. The URI describes the network location of the resource.



Figure 5-1 What is REST?

Notes:

Representational State Transfer (REST) is an architecture style for building resources on the web. Examples of resources for website include HTML documents, images, and script files.

To retrieve or update a resource, perform an action through HTTP methods. To identify which resource to retrieve or update, REST uses a Uniform Resource Identifier (URI) to describe the network location of the resource.

REST provides the following HTTP methods:

- ▶ GET
- ▶ POST
- ▶ DELETE
- ▶ PUT
- ▶ OPTIONS
- ▶ HEAD
- ▶ TRACE
- ▶ CONNECT

The GET method is used to retrieve information from the server. When you use your browser to navigate to any URI, you use the GET method to get the HTML of that website. The query string that contains the parameters that are needed for the request are sent in the URL by placing a question mark (?) at the end of the URI, and then, writing the parameters.

Each parameter is represented as a name-value pair. The parameters are separated by an ampersand (&). The URI for a GET request can be formatted as shown in the following examples:

`http://example.com/personDetail?firstName=Ahmed&age=28`

or

`http://example.com/personDetail/Ahmed/28.`

The POST method is used to post data to the server. In this case, the parameters are posted in the body of the request, not in the URI.

The DELETE method is used to delete a resource from the server.

5.4 Applying REST to server-side applications

The slide has a red header bar. On the left is the Redbooks logo with the text 'Powered by Redbooks'. On the right is the IBM logo. The main title 'Applying REST to server-side applications' is centered in large red font. Below it is a bulleted list of points.

- In a more general sense, web resources represent a source of information:
 - HTML documents define the structure of a web page.
 - CSS documents define the presentation of a web page.
 - Image files provide a visual representation of information.
- With REST services, you expose your server applications as a web resource:
 - A REST service is an entry point to application on the server.
 - HTTP method verbs are used to call a REST service.
 - A URI specifies the REST service to call. The URI describes the network location of the server application resource.

Figure 5-2 Applying REST to server-side applications

Notes:

In a more general sense, web resources represent a source of information. For example, HTML documents define the structure of a web page. Cascading Style Sheet (CSS) documents define the presentation of a web page, and image files provide a visual representation of information. With REST services, you treat server applications as web resources.

A REST service is now an entry point to an application on the server. It provides information from the server application. To call a REST service, use HTTP method verbs, such as GET, PUT, and POST. To specify which REST service to call, use a URI to describe the location of the resource on the server.

5.5 Example: Application model architecture for REST services

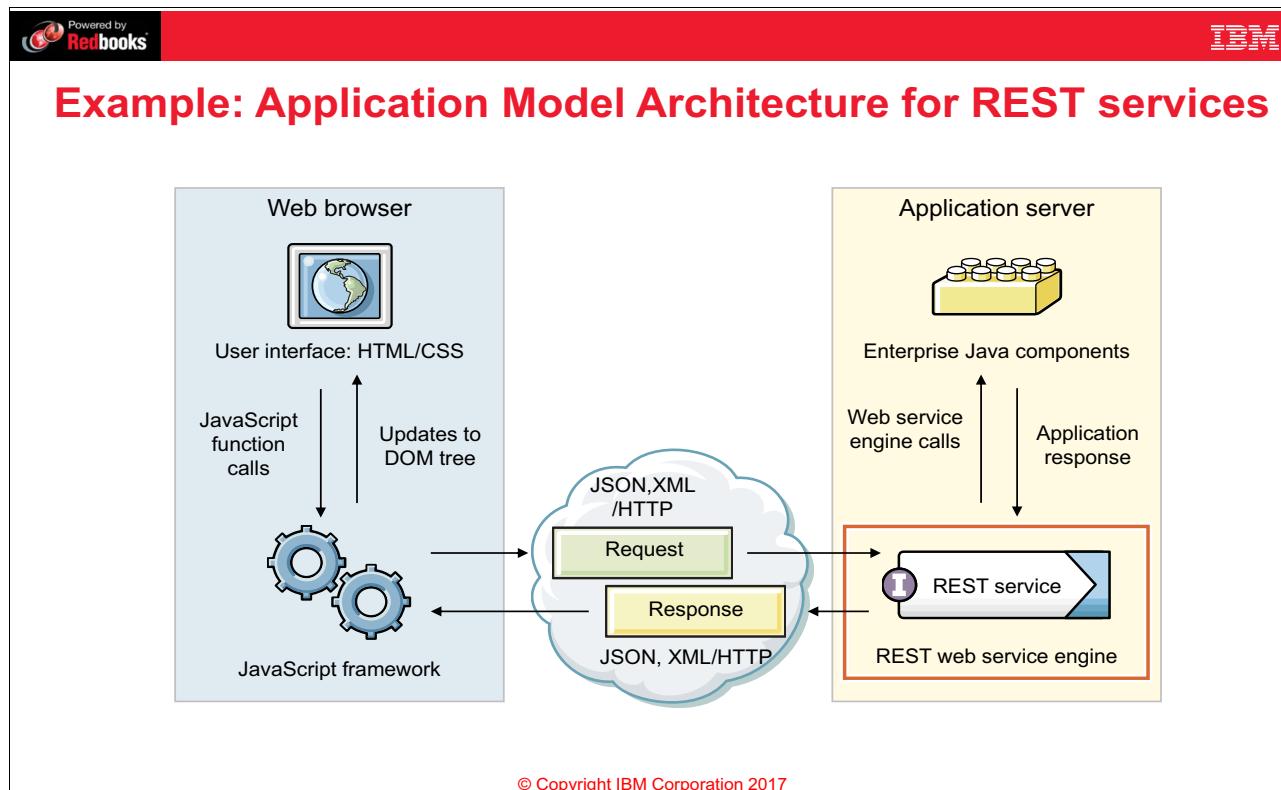


Figure 5-3 Example: Application model architecture for REST services

Notes:

In this example, Enterprise Java components represent the server-side application while the client-side of the application is based on JavaScript. The server-side application exposes a list of services as REST APIs. The client-side application calls these REST APIs by using one of the HTTP methods. The request and the response can be JSON or XML over HTTP Protocol.

You do not have to code the server side in Java and the client in JavaScript. Although many people use Groovy, Rails, Python, and other languages, they all can still use REST.

5.6 What is a RESTful web service?



What is a RESTful web service?

A **RESTful web service**, or **REST service**, is a web service that follows the principles of REST:

- A web server hosts web resources: Applications and sources of information. Consider an example service, the IBM stock service contains information about the current stock price.
- **Identifiers** uniquely reference web resources. The resource path `/stock/IBM/` represents the IBM stock resource on the server
- The client uses HTTP methods as a uniform interface to interact with resources. To retrieve the current IBM stock price, send a `GET` operation to `/stock/IBM`.

Figure 5-4 What is a RESTful web service?

Notes:

A web service is a service exposed over the web to perform a certain function, such as `getStockPrice` for a company. A RESTful web service, or REST service, is a web service that follows the principles of REST.

A web server hosts web resources, such as applications and sources of information; for example, the IBM stock resource that contains information about the current stock price.

Identifiers uniquely references web resources. The resource path `/stock/IBM/` represents the IBM stock resource on the server. The client uses HTTP methods as a uniform interface to interact with resources. In our example, a `GET` operation is sent on `/stock/IBM` to retrieve the current IBM stock price.

5.7 Example: Sending an HTTP request to a REST service

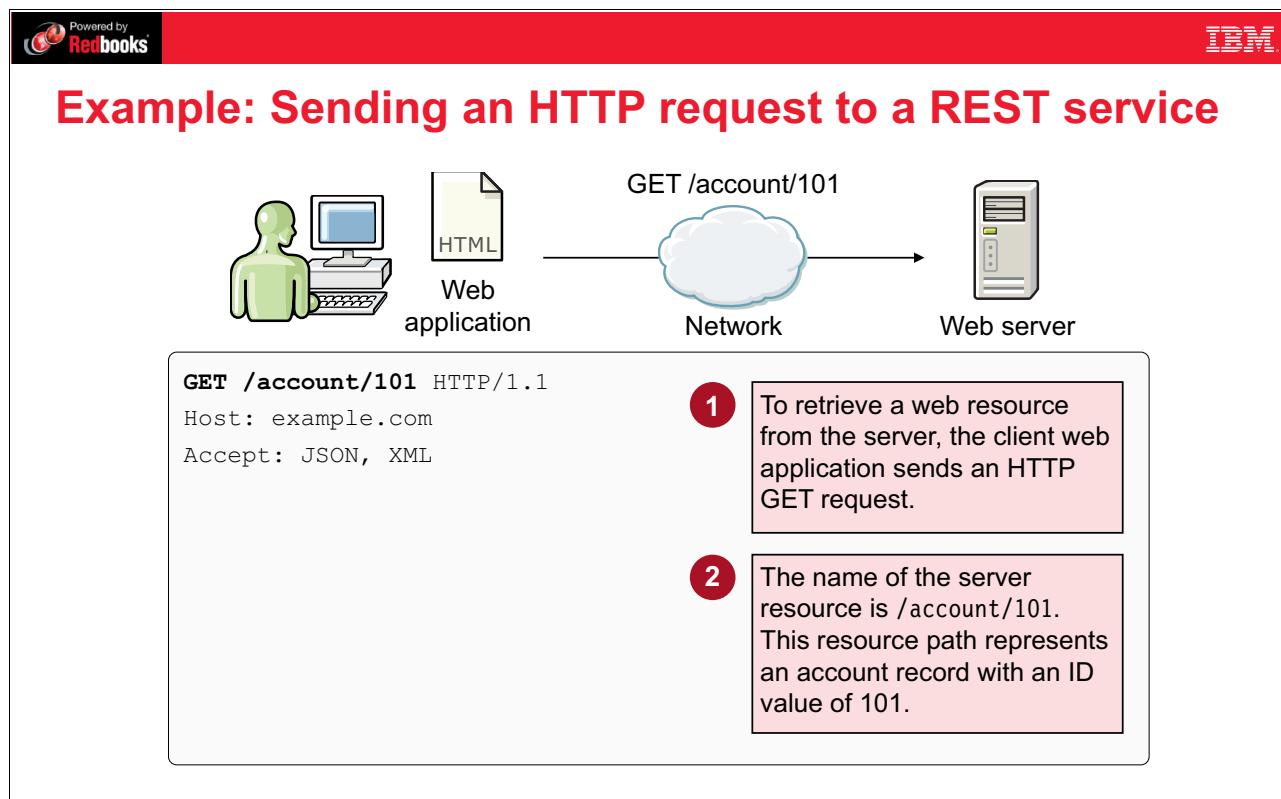


Figure 5-5 Example: Sending an HTTP request to a REST service

Notes:

In this example, a client application running in the web browser sends an HTTP GET request for the resource on the server. Notice that the procedure for calling a REST service is the same as making a request for a web page by using an HTTP GET request. When you navigate to a URL in your browser, your browser automatically sends a GET request to retrieve the requested page.

The name of the server resource is /account/101. This resource path represents an account record with an ID value of 101.

5.8 Example: Receiving an HTTP response from a REST service

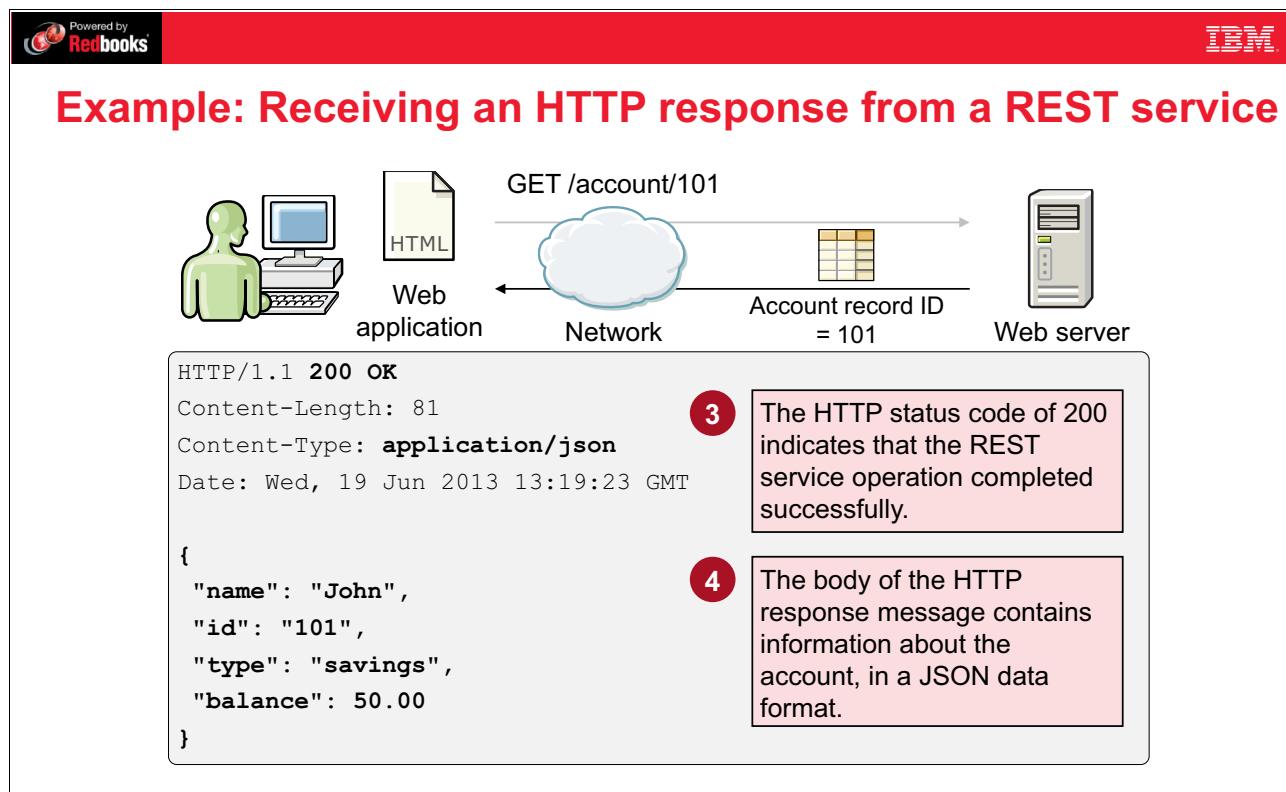


Figure 5-6 Example: Receiving an HTTP response from a REST service

Notes:

The REST service running on the web server receives the HTTP GET request. It fulfills the request by returning an HTTP response message with information about the account in the message body. In the response message, the REST service writes the protocol type/version, HTTP status code, Content-Length, Content-Type, Date, and Response Body.

In this example, the protocol type/version is HTTP/1.1 and the HTTP status code of 200 indicates that the REST service operation completed successfully. A human-readable description of the status code (which is OK) appears after the code.

Content-Length contains the length of the response of message. In this example, it is 81 characters.

Content-Type describes the data type of the response. In this example, it is JSON.

Response Body is a JSON object that contains four name-value pairs, which contains the values of the keys name, ID, type, and balance.

5.9 REST characteristics

Powered by  Redbooks

IBM

REST characteristics

- REST is an architecture style for designing network applications. REST is an architecture, not a product.
- REST provides a simple approach for building services for client/server interactions that are based on web resources.
- REST services follow standard protocols such as HTTP.
- REST services tend to use lightweight data models, such as JSON. XML is also supported.
- REST services are a popular way for applications to interact with server-side applications.



Figure 5-7 REST characteristics

Notes:

REST has the following characteristics:

- ▶ REST is a simple way of building services for client/server interactions, which are built on web resources.
- ▶ REST is an architecture, not a product. You build services that follow the REST architectural style.
- ▶ REST services follow standard web protocols, such as HTTP. There is a misconception that REST can work solely over the HTTP protocol, but this idea is not entirely true. Although the most common scenarios for the use of REST is over the HTTP protocol, REST can be used over other transfer protocols, such as SMTP.
- ▶ REST services tend to use lightweight data models, such as JSON. It is used also for XML.
- ▶ REST services are a popular way for applications to interact with server-side applications.

5.10 Introduction to JSON



Powered by
Redbooks

IBM

Introduction to JSON

- **JavaScript Object Notation (JSON)** is a text format for structured data
 - Its syntax is derived from the object literals of JavaScript, according to the ECMA-262 ECMAScript language standard, third edition:
<http://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - JSON is a platform-neutral and language-neutral data format
- The main design goal of JSON is to provide a minimal, portable, textual data interchange format.
- JSON is not a markup language. Unlike XML, it does not use descriptive tags to encapsulate the data <title></title>.
- JSON is built on two structures:
 - Name-Value Pairs
 - List of Values

Figure 5-8 Introduction to JSON

Notes:

JavaScript Object Notation (JSON) is a text format for structured data. Its syntax is derived from the object literals of JavaScript, according to the ECMA-262 ECMAScript language standard, third edition, which is scripting language standard.

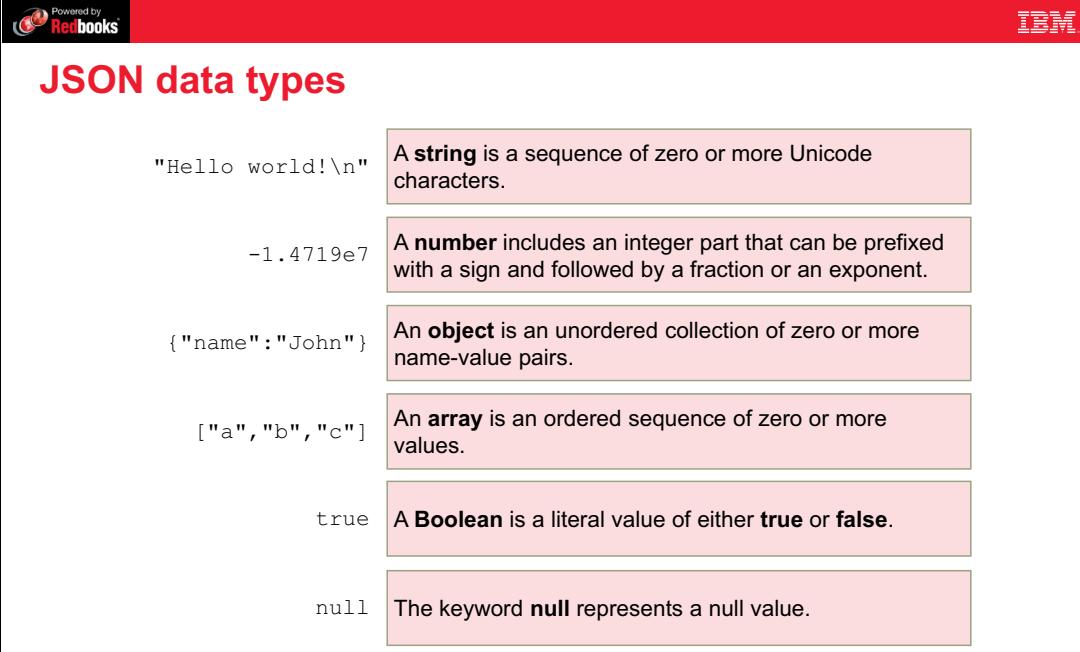
JSON is a platform-neutral and language-neutral data format.

The main design goal of JSON is to provide a minimal, portable, textual data interchange format.

JSON is not a markup language. Unlike XML, it does not use descriptive tags to encapsulate its data. For example, XML is a markup language because it uses tags, such as <title></title>, to declare the title of the page; JSON is not a markup language.

JSON is built on two structures: A collection of name-value pairs known as *objects* and a list of values known as *arrays*.

5.11 JSON data types



The diagram is titled "JSON data types" and is presented in a grid format. It shows six examples of JSON values on the left, each with a corresponding definition on the right. The examples are: "Hello world!\n" (String), -1.4719e7 (Number), {"name": "John"} (Object), ["a", "b", "c"] (Array), true (Boolean), and null (Null). The definitions are: A string is a sequence of zero or more Unicode characters; A number includes an integer part that can be prefixed with a sign and followed by a fraction or an exponent; An object is an unordered collection of zero or more name-value pairs; An array is an ordered sequence of zero or more values; A Boolean is a literal value of either true or false; The keyword null represents a null value.

Value	Description
"Hello world!\n"	A string is a sequence of zero or more Unicode characters.
-1.4719e7	A number includes an integer part that can be prefixed with a sign and followed by a fraction or an exponent.
{"name": "John"}	An object is an unordered collection of zero or more name-value pairs.
["a", "b", "c"]	An array is an ordered sequence of zero or more values.
true	A Boolean is a literal value of either true or false .
null	The keyword null represents a null value.

Figure 5-9 JSON data types

Notes:

JSON features the following data types:

- ▶ A string is a sequence of zero or more Unicode characters.
- ▶ A number includes an integer part and a fraction. Numbers can be prefixed by a positive or negative sign. It can also include an exponent.
- ▶ There are two data types to hold a group of values:
 - An object: An unordered collection of zero or more name-value pairs. Objects are denoted by curly brackets, which means that the order is not guaranteed in JSON objects. For example, if you send a request { "name": "John", "preferredColor": "Blue"}, it is not always guaranteed that the receiver receives them in the same order.
 - An array: An ordered sequence of zero or more values. Use square brackets to denote arrays. Order is guaranteed in JSON arrays.
- ▶ A Boolean is a literal value of true or false.
- ▶ The keyword null represents a null value.

JSON values must be an object, array, number, or string, or one of the three literal names (false, true, or null). JSON does not support the JavaScript keyword “undefined”. Use null or another set value to represent an undefined value.

5.12 JSON data type: Objects



JSON data type: Objects

An unordered collection of key/value pairs:

- Curly brackets ({}) hold object declarations
- Colons separate object keys and values
- Commas separate each key-value pair
- Keys are strings
- Values can be any JSON data type
- Objects can be nested.

```
{  
    "name": {  
        "first": "John",  
        "last": "Smith"  
    },  
    "id": 101,  
    "email": "john.smith@example.com"  
}
```

Figure 5-10 JSON data type: Objects

Notes:

JSON Object is an unordered collection of key/value pairs with the following characteristics:

- ▶ Curly brackets ({}) hold object declarations
- ▶ Colons separate object keys and values
- ▶ Commas separate each key-value pair
- ▶ Keys are strings
- ▶ Values can be any JSON data type
- ▶ Objects can be nested

In our example, the JSON object has three fields: Name, ID, and email. The name field is another JSON object with two fields: First and last.

5.13 JSON data type: Arrays



JSON data type: Arrays

An ordered sequence of values with these characteristics:

- Arrays must begin and end with square brackets ([])
- Commas separate array values
- Arrays can be nested

```
[ "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
  "Friday", "Saturday"]

[
  [0, -1, 0],
  [{"one":1}, 0, "hello"],
  [0, , 1]
]
```

Figure 5-11 JSON data type: Arrays

Notes:

JSON Array is an ordered sequence of values with the following characteristics:

- ▶ Arrays must begin and end with square brackets ([])
- ▶ Commas separate array values
- ▶ Arrays can be nested to represent multidimensional arrays

Two examples are shown in Figure 5-11. The first example is an array of seven string values. The second example is a multi-dimensional array. Notice that the array can hold a mix of JSON data types.

JSON must start with an object or an array at the top level.

5.14 What is Watson?

Powered by
Redbooks

What is Watson?

IBM Watson is a technology platform that uses natural language processing and machine learning to reveal insights from large amounts of unstructured data. Today, 80% of all data is unstructured.*

OUTTHINK

- *Source: <https://www.ibm.com/watson/what-is-watson.html>

© Copyright IBM Corporation 2017

Figure 5-12 What is Watson?

Notes:

IBM Watson is a technology platform that uses natural language processing and machine learning to reveal insights from large amounts of unstructured data.

Watson analyzes unstructured data. As of this writing, 80% of all data is unstructured; that is, not structured in machine-readable format. Unstructured data includes news articles, research reports, social media posts, and enterprise system data.

For more information, see the video *IBM Watson How it Works*, which is available on YouTube:

https://www.youtube.com/watch?v=_Xcmh1LQB9I

5.15 Watson Services in IBM Bluemix

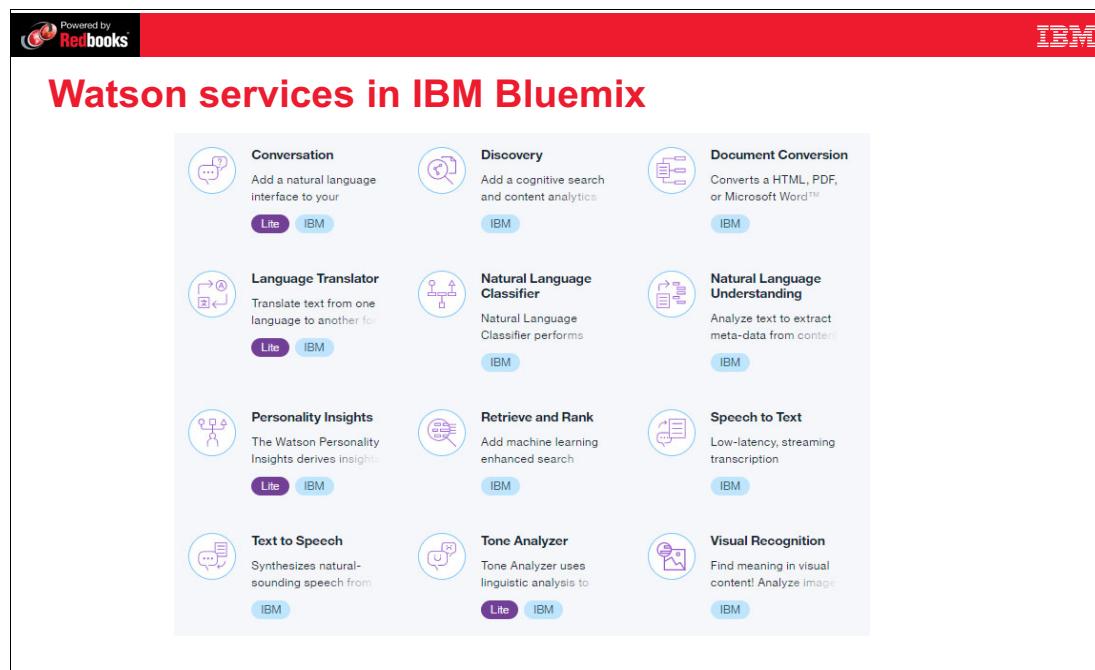


Figure 5-13 Watson services in IBM Bluemix

Notes:

The IBM Bluemix catalog includes the following Watson services (new services are introduced periodically):

- ▶ **Natural Language Understanding**

This service analyzes unstructured text content. You can use it to extract semantic metadata from content, such as information about people, places, companies, topics, facts, relationships, authors, and languages.

- ▶ **Language Translation**

This service translates text from one language to another.

- ▶ **Personality Insights**

This service extracts and analyses a spectrum of personality attributes to help discover actionable insights about people and entities, and in return guides users to highly personalized interactions. It takes a text as an input and outputs the analysis of the personality according to the input text. For an interactive demonstration of this service, see the following website:

<https://personality-insights-livedemo.mybluemix.net/>

- ▶ **Speech to text**

This service converts speech into written words. As of this writing, English, Japanese, Arabic, Mandarin, Portuguese, Spanish, and French are supported.

5.16 Watson API Explorer

The screenshot shows the Watson API Explorer interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the main title "Watson API Explorer" is displayed in large red font. To the right of the title, there are two buttons: "Watson Developer Cloud" and "Menu". The main content area has a purple background and contains the text "Watson API Explorer" and "A collection of Swagger documentation for the Watson APIs.". On the right side of the content area, there's a sidebar titled "Language" which lists several Watson services with their corresponding icons and names: AlchemyLanguage, Conversation, Dialog (Deprecated), Document Conversion, Language Translator, and Natural Language Classifier.

Figure 5-14 Watson API Explorer

Notes:

Watson APIs are exposed as REST APIs. You can interact with each API by using one of the HTTP methods, such as GET and POST.

Watson API Explorer is a public portal that contains documentation for the different Watson REST APIs. It also allows the user to call these REST APIs from the portal directly.

Watson API Explorer is based on Swagger, which is used to document REST APIs. For more information about Watson API Explorer, see the following website:

<https://watson-api-explorer.mybluemix.net/>

5.17 Example: Watson API Explorer - Natural Language Understanding (Authors)

The screenshot shows the Watson API Explorer interface. On the left, under 'Request', there is a table of parameters:

Parameter	Value	Description
version	2017-02-27	The release date of the version of the API you want to use. Specify dates in YYYY-MM-DD format.
text	<input type="text"/>	Text to analyze
html	<input type="text"/>	HTML to analyze
url	http://www.reuters.com/art	Public webpage to analyze
features	<input type="text" value="sentiment,metadata,relations,semantic_roles"/>	Comma separated list of analysis features

On the right, under 'Response Body', is a JSON object representing the analysis results:

```
{ "retrieved_url": "http://www.reuters.com/article/us-africa-japan-idUSKCN19P0JU", "metadata": { "title": "Japan pledges $30 billion for Africa over next three years", "publication_date": "2016-08-27T00:00:00", "image": "", "feeds": [ { "link": "http://www.reuters.com/tools/rss" } ], "authors": [ { "name": "George Obulutsa" } ] }, "language": "en" }
```

Figure 5-15 Example: Watson API Explorer - Natural Language Understanding (Authors)

Notes:

The Watson Natural Language Understanding service is used to analyze various features of text content. For more information, see the video *Watson Natural Language Understanding Service Overview*, which is available on YouTube:

<https://www.youtube.com/watch?v=J5I1Kj7H8T8>

In this example, the Analyze REST API is called from the Natural Language Understanding service to extract the author of an article in Reuters.

The Resource Path, which is the path of the exposed REST resource on the server, is /v1/analyze. The HTTP verb that is used is GET.

The following parameters are used in the example:

- ▶ Version: Watson services is available in different releases. The latest release of Watson Natural Language Understanding as of the time of this writing is 2017-02-27.
- ▶ Url: The URL of the article from which to extract the author name.
- ▶ Features: Natural Language Understanding provides several features for analyzing the text. This example shows the use of the metadata feature, which gets author information, publication date, and the title of your text or HTML content.

After the parameters are entered, click **Try it out!**. Watson API Explorer builds the Request URI, as shown in the following example:

```
https://watson-api-explorer.mybluemix.net/natural-language-understanding/api/v1/analyze?version=2017-02-27&url=http%3A%2F%2Fwww.reuters.com%2Farticle%2Fus-africa-ja-pan-idUSKCN112077&features=metadata&return_analyzed_text=false&clean=true&fallback_to_raw=true&concepts.limit=8&emotion.document=true&entities.limit=50&entities.emotion=false&entities.sentiment=false&keywords.limit=50&keywords.emotion=false&keywords.sentiment=false&relations.model=en-news&semantic_roles.limit=50&semantic_roles.entities=false&semantic_roles.keywords=false&sentiment.document=true
```

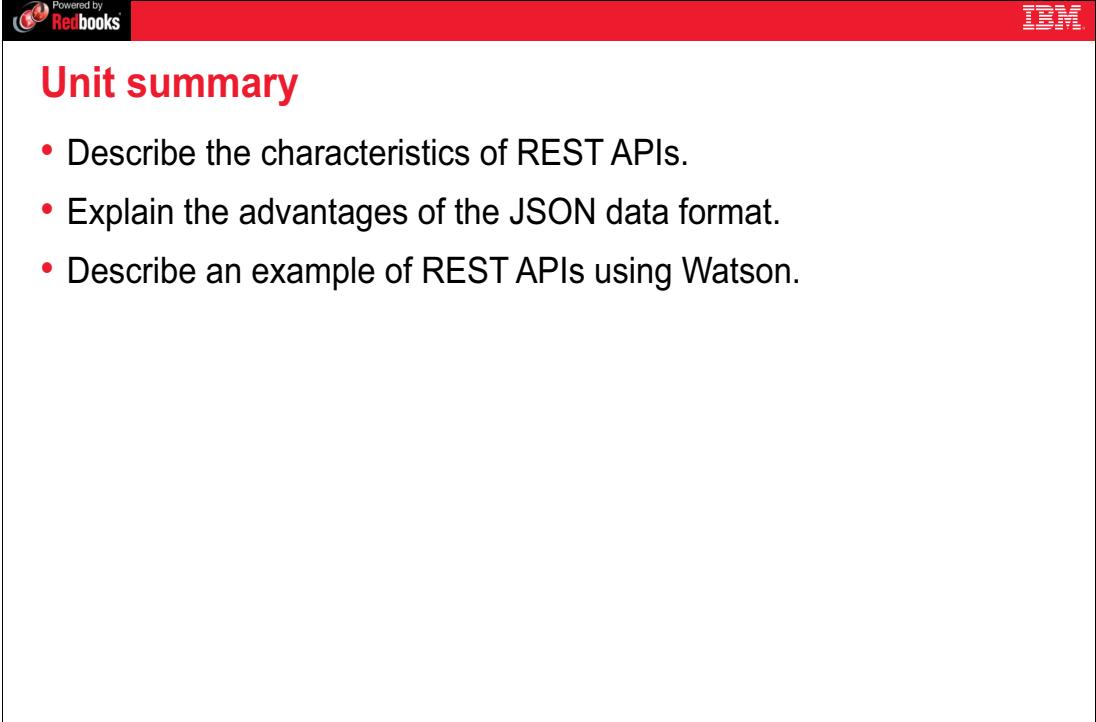
The service URL is

```
https://watson-api-explorer.mybluemix.net/natural-language-understanding/api/
```

The resource path is /v1/analyze, and all the parameters are passed in the query string of the URL because it is a GET method. Also, because it is a GET method, you can try it directly from your browser.

The JSON response is a JSON object with a collection of name-value pairs with keys retrieved_url, metadata, and language. Metadata is another nested JSON object that contains title, publication_date, and authors. Authors includes the key names, which contain a JSON array. This JSON array contains the names of the authors of the article.

5.18 Unit summary



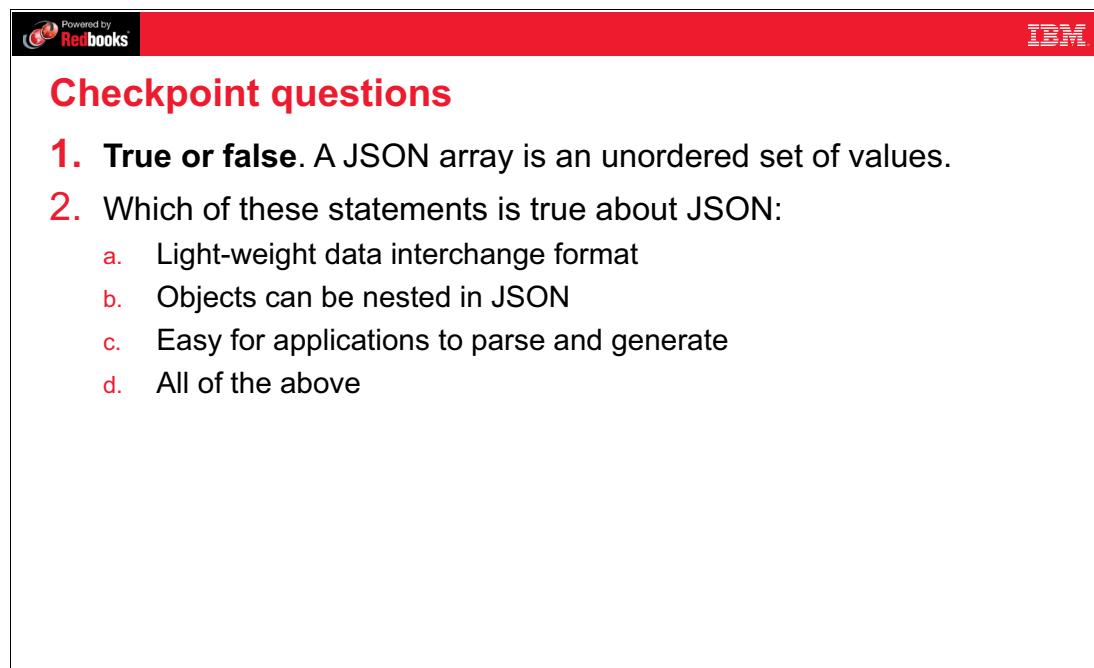
The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo.

Unit summary

- Describe the characteristics of REST APIs.
- Explain the advantages of the JSON data format.
- Describe an example of REST APIs using Watson.

Figure 5-16 Unit summary

5.19 Checkpoint questions



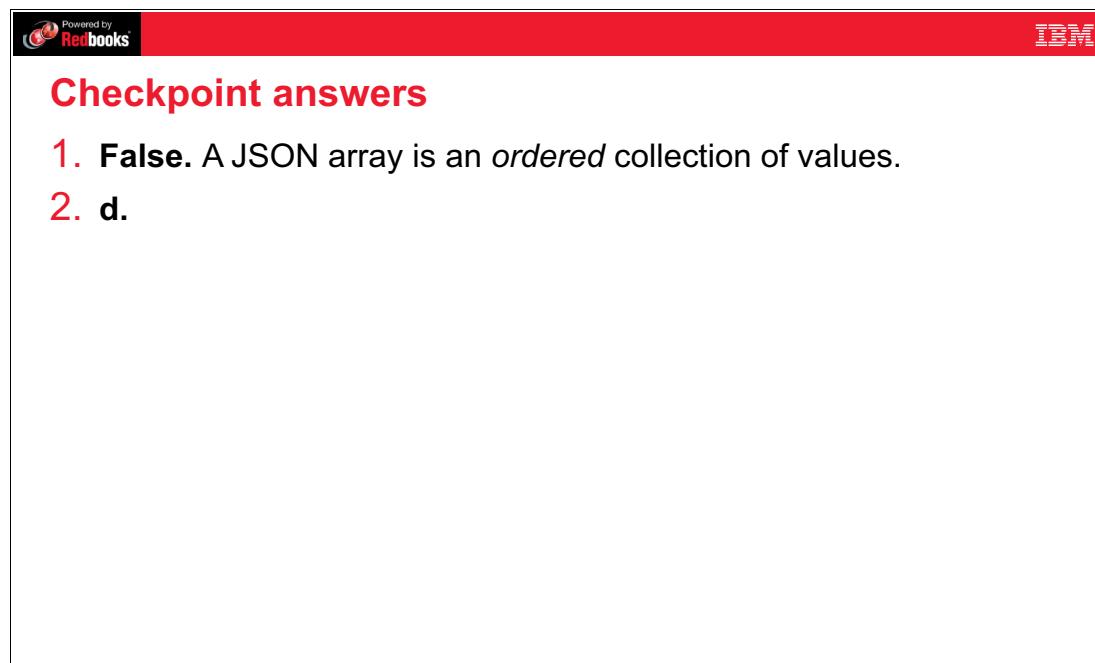
The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a red title bar at the top with the text "Checkpoint questions" in white.

Checkpoint questions

- 1. True or false.** A JSON array is an unordered set of values.
- 2. Which of these statements is true about JSON:**
 - a. Light-weight data interchange format
 - b. Objects can be nested in JSON
 - c. Easy for applications to parse and generate
 - d. All of the above

Figure 5-17 Checkpoint questions

5.20 Checkpoint answers

A screenshot of a presentation slide titled "Checkpoint answers". The slide has a red header bar with the "Powered by Redbooks" logo on the left and the "IBM" logo on the right. The main content area contains two numbered items: "1. False. A JSON array is an *ordered* collection of values." and "2. d.". The background of the slide is white.

Checkpoint answers

1. **False.** A JSON array is an *ordered* collection of values.
2. **d.**

Figure 5-18 Checkpoint answers



Introduction to data services in IBM Bluemix

This unit covers the following topics:

- ▶ Database services in IBM Bluemix
- ▶ IBM Cloudant

6.1 What you should be able to do

After completing this unit, you should be able to:

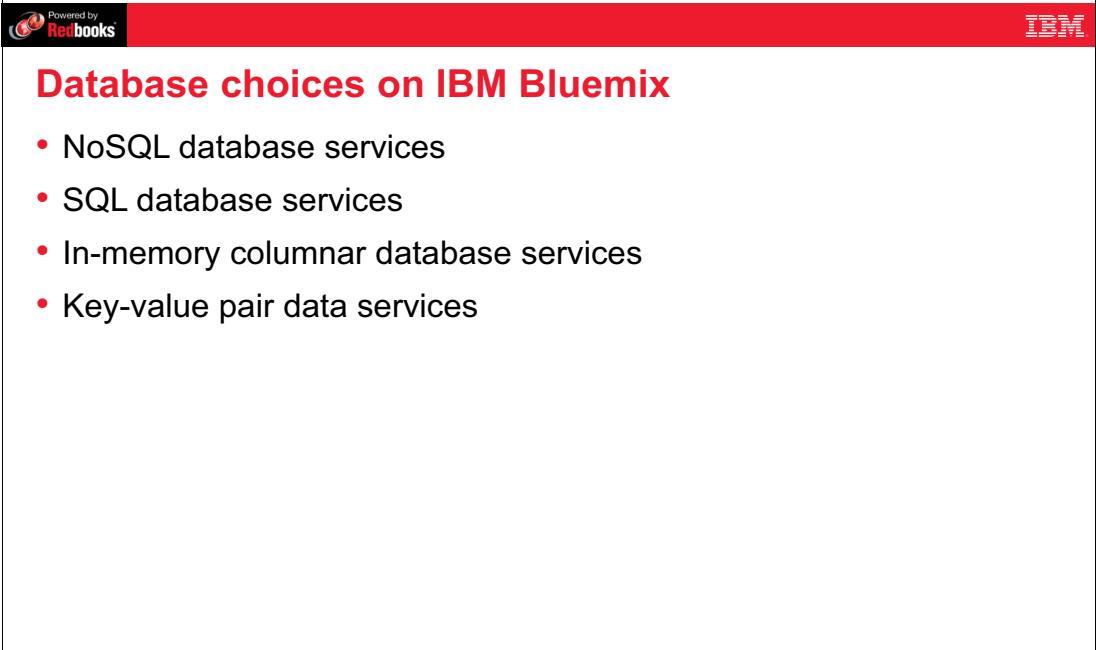
- ▶ Describe the main types of data services in IBM Bluemix.
- ▶ Describe the benefits of IBM Cloudant.
- ▶ Explain how Cloudant databases and documents are accessed from IBM Bluemix.
- ▶ Describe how to use REST APIs to interact with Cloudant database.

6.2 References

The following publications are useful for further research on the topic presented in this unit:

- ▶ Cloudant.com documentation:
<https://docs.cloudant.com/index.html>
- ▶ IBM Cloud Data Services documentation:
<https://developer.ibm.com/clouddataservices/>
- ▶ IBM DeveloperWorks articles and resources:
<https://www.ibm.com/developerworks>

6.3 Database choices on Bluemix

A slide titled "Database choices on IBM Bluemix". The slide is powered by Redbooks and features the IBM logo. It lists four types of database services: NoSQL database services, SQL database services, In-memory columnar database services, and Key-value pair data services.

Powered by
Redbooks

IBM

Database choices on IBM Bluemix

- NoSQL database services
- SQL database services
- In-memory columnar database services
- Key-value pair data services

Figure 6-1 Database choices on IBM Bluemix

Notes:

IBM Bluemix provides the main types of databases and the following database services for each type:

- ▶ NoSQL databases have the advantage of storing objects in the same format that is expected by applications. Data is transferred and stored in standard formats, such as JSON, or XML. To perform a more complex query on a NoSQL database, you typically must create a view. A view is a self-maintained index that the database management system provides for quickly traversing the data set.
- ▶ SQL Database service provides databases that are based on relational tables. They are used extensively to store data in enterprise computing. SQL allows you to write queries that combine rows from multiple related tables. Each table has a record with various attributes (much like an object).
- ▶ Columnar database is the database that stores data in form of columns instead of rows; for example, to model the following information:
 - ID:1, Name: Ahmed, Age: 29, Weight: 65
 - ID:2, Name: Ben, Age: 34, Weight: 70
 - ID:3, Name: John, Age: 32, Weight: 73
- ▶ Each column is stored in a database record:
 - Ahmed:1, Ben: 2, John:3
 - 29:1, 34:2, 32:3
 - 65:1, 70:2, 73:3
- ▶ Key-Value Pair Data Services allow efficient storage of key-value pair data.

6.4 Data services in IBM Bluemix catalog

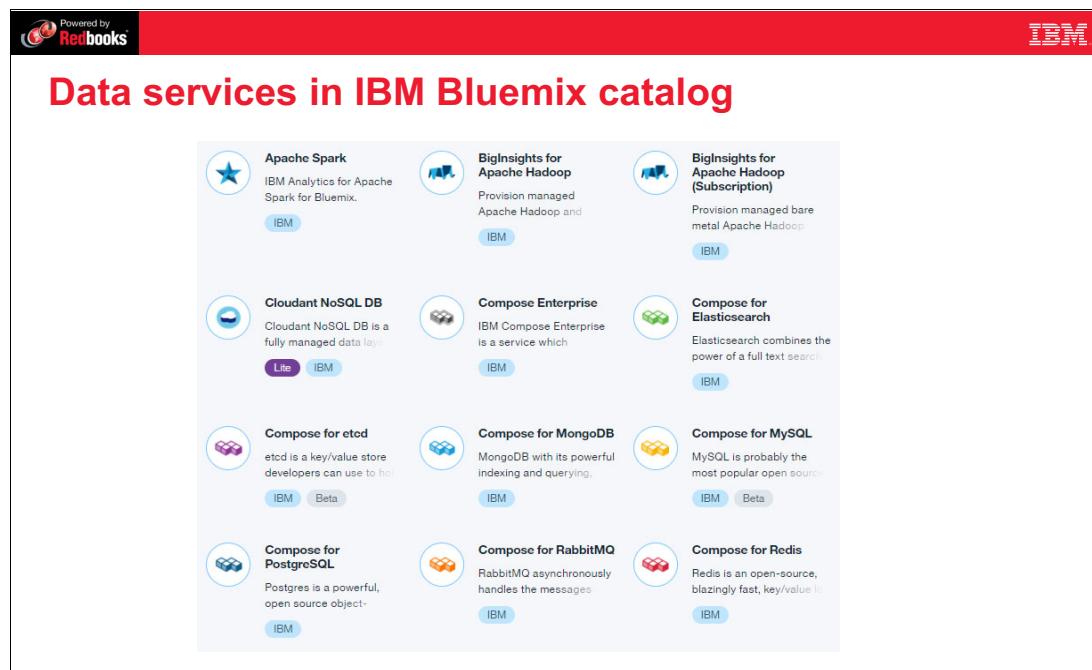


Figure 6-2 Data services in IBM Bluemix catalog

Notes:

IBM Bluemix offers the following database management systems that are supported by IBM and other third-party vendors to use with cloud computing:

- ▶ NoSQL database services, such as Cloudant NoSQL DB and Compose for MongoDB.
- ▶ SQL database services, such as Db2 hosted, Compose for PostgreSQL, ElephantSQL, and ClearDB Managed MySQL Database.
- ▶ In-memory columnar database services, such as IBM dashDB® for Analytics.
- ▶ Key-value pair data services, such as Redis Cloud, which is an open-source, fast key/value low maintenance store.

The database services in the catalog are offered by IBM or third parties.

6.5 Cloudant capabilities



IBM

Cloudant capabilities

- Database as a service: Provision and scale according to your requirements
- Data is stored as documents in JSON format: Schema-less NoSQL format
- Simple API: REST-based
- Cloudant Search
- Cloudant Geo
- Offline-First mobile web apps capabilities
- Synchronization feature for disconnected Android and Apple apps
- Client libraries for developing your own application

Figure 6-3 Cloudant capabilities

Notes:

IBM acquired Boston-based cloud database startup Cloudant in 2014.

IBM Cloudant is a NoSQL database as a service (DBaaS) that is optimized for handling heavy workloads of concurrent reads and writes in the cloud. These workloads are typical for large, fast-growing web and mobile apps. It is built to scale globally, run non-stop, and handle various data types, such as JSON, full-text, and geospatial. Cloudant is designed to ensure that the flow of data between an application and its database remains uninterrupted and performs to the users' satisfaction. The data replication technology also allows developers to put data closer to where their applications need it most.

Cloudant frees developers from worrying about managing the database and enables them to focus on the application. Cloudant eliminates the risk, cost, and distractions of database scalability, which enables you to regain valuable time and your applications to scale larger and remain consistently available to users worldwide.

Data is stored and sent in JSON format, and the data documents are accessed with a simple REST-based HTTP method. Anything that is encoded into JSON can be stored as a document.

The NoSQL databases feature the following query capabilities that are well-suited for the data format used in the applications:

- ▶ Cloudant Search is powered by Apache Lucene, the most popular open-source search library that is designed to index and search JSON documents. By drawing on the speed and simplicity of Lucene, the Cloudant service provides a familiar way to add search to apps.

- ▶ Cloudant Geo is an easy GeoJSON storage with built-in spatial querying and map visualization. GeoJSON is a format based on JSON that supports the encoding of geographic data structures.
- ▶ Offline First Capabilities let users synchronize data with their mobile devices, allowing them to work in disconnected mode. Offline-first applications built with Cloudant Sync provide a better, faster user experience offline and online, by storing and accessing data locally and then synchronizing this data with the cloud database when an internet connection is available.

Language-specific libraries for Cloudant are available, which are wrappers that help you work with a simple API.

For more information, see this website:

<https://Cloudant.com>

6.6 Cloudant in IBM Bluemix versus Cloudant.com

Cloudant in IBM Bluemix versus Cloudant.com

- At cloudant.com, you have a choice of the following settings:
 - IaaS platform: SoftLayer, Rackspace, and Azure
 - Data location:

By
you agree

Data Location

Moonshine - SoftLayer, US E ▾

Moonshine - SoftLayer, US East

Jenever - SoftLayer, Amsterdam

Baijiu - SoftLayer, Hong Kong

Sling - SoftLayer, South-East Asia

Malort - Rackspace, US-Chicago

Mead - Rackspace, London

Porter - SoftLayer, London

Lagoon 2 (Beta) - Azure, US West

- When you create a Bluemix Cloudant service, the following processes occur:
 - An account with name and password are generated for you on the SoftLayer Cloudant cluster for the specific Bluemix region.
 - All credentials details are stored in VCAP_SERVICES.

Figure 6-4 Cloudant in IBM Bluemix versus Cloudant.com

Notes:

You can sign up to use the Cloudant database at the following website or use the Cloudant data management services that are provided by IBM Bluemix:

<https://cloudant.com/>

When you sign up, users are offered a choice of the underlying infrastructure as a service, including SoftLayer, Rackspace, and Azure.

When you create an IBM Bluemix Cloudant service, an account is generated for you on the SoftLayer Cloudant cluster for the specific IBM Bluemix region that you are signed in with IBM Bluemix.

6.7 Documents in Cloudant

The screenshot shows a presentation slide with a red header bar. On the left of the header is the Redbooks logo with the text 'Powered by Redbooks'. On the right is the IBM logo. The main content area has a white background and a title 'Documents in Cloudant' in bold red font. Below the title is a bulleted list of five items:

- Documents are JSON objects.
- Cloudant documents are containers for the data.
- All documents have the following unique mandatory fields:
 - Unique `_id`
 - `_rev`
- In addition to the two mandatory fields, documents can contain any other content that is expressed in the JSON format.

Figure 6-5 *Documents in Cloudant*

Notes:

Cloudant documents are containers for the data, and the documents are JSON objects. All documents in Cloudant must contain the following unique fields:

- ▶ An identifier `_id` field serves as the document key. It can be created by the application or generated automatically by Cloudant.
- ▶ A revision number `_rev` field is automatically generated and used internally by the Cloudant database as a revision number. A revision number is added to your documents by the server when you insert or modify them. You must specify the latest `_rev` when updating a document or your request fails. It also helps avoid conflicting data states.

6.8 Getting started with Cloudant on IBM Bluemix

The screenshot shows a web page titled "Getting started with Cloudant on IBM Bluemix". At the top left is the Redbooks logo, and at the top right is the IBM logo. The main heading is "Getting started with Cloudant on IBM Bluemix". Below it, a sub-heading reads: "The following starter boilerplate applications on IBM Bluemix use Cloudant:" followed by a bulleted list:

- ASP.NET Core Cloudant Starter
- Java Cloudant Web Starter
- Node.js Cloudant DB Web Starter

Below the list are three cards, each representing a different starter application:

- ASP.NET Core Cloudant Starter**: A card with a blue icon showing two people, the text "ASP.NET Core Cloudant Starter", and the description "Use the Cloudant NoSQL DB Service in an IBM".
- Java Cloudant Web Starter**: A card with a blue icon showing a person, the text "Java Cloudant Web Starter", and the description "Use the Cloudant NoSQL DB service with the IBM".
- Node.js Cloudant DB Web Starter**: A card with a blue icon showing a person, the text "Node.js Cloudant DB Web Starter", and the description "Use the Cloudant NoSQL DB service with the IBM". It also includes a "Lite" button.

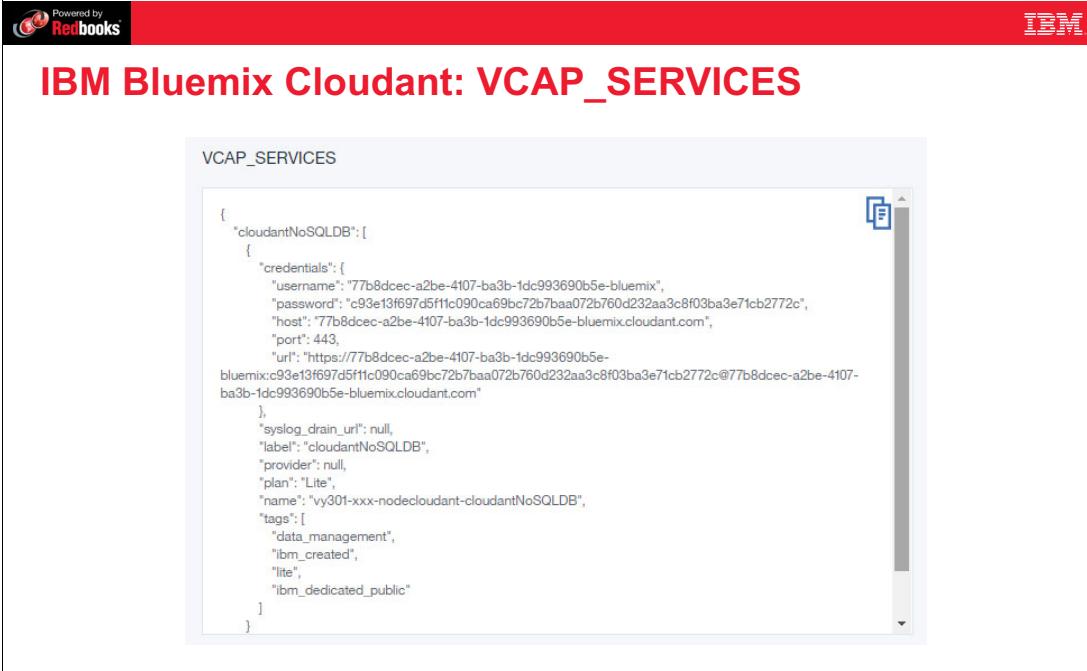
Figure 6-6 Getting started with Cloudant on IBM Bluemix

Notes:

IBM Bluemix provides boilerplate starter applications for Cloudant with ASP.NET, Java, and Node.js. The Node.js Cloudant DB Web Starter is used in one of the exercises in this unit.

Cloudant also features rich documentation.

6.9 IBM Bluemix Cloudant: VCAP_SERVICES



The screenshot shows a web-based interface for managing services in IBM Bluemix. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "IBM Bluemix Cloudant: VCAP_SERVICES" is displayed in red. Underneath the title, the heading "VCAP_SERVICES" is shown in blue. The main content area contains a JSON object representing service credentials:

```
{  
  "cloudantNoSQLDB": [  
    {  
      "credentials": {  
        "username": "77b8dcec-a2be-4f07-ba3b-1dc993690b5e-bluemix",  
        "password": "c93e13f697d5f1fc090ca69bc72b7baa072b760d232aa3c8f03ba3e71cb2772c",  
        "host": "77b8dcec-a2be-4f07-ba3b-1dc993690b5e-bluemix.cloudant.com",  
        "port": 443,  
        "url": "https://77b8dcec-a2be-4f07-ba3b-1dc993690b5e-bluemix:c93e13f697d5f1fc090ca69bc72b7baa072b760d232aa3c8f03ba3e71cb2772c@77b8dcec-a2be-4f07-ba3b-1dc993690b5e-bluemix.cloudant.com"  
      },  
      "syslog_drain_url": null,  
      "label": "cloudantNoSQLDB",  
      "provider": null,  
      "plan": "Lite",  
      "name": "vy301-xxx-nodecloudant-cloudantNoSQLDB",  
      "tags": [  
        "data_management",  
        "ibm_created",  
        "lite",  
        "ibm_dedicated_public"  
      ]  
    }  
  ]  
}
```

Figure 6-7 IBM Bluemix Cloudant: VCAP_SERVICES

Notes:

When you create an application in IBM Bluemix that uses the Cloudant database, or when you add the Cloudant service to an application, IBM Bluemix adds the credentials of the Cloudant database to the VCAP_SERVICES environment variable. This variable includes the credentials, such as the URL, host, port number, username, and password for accessing the Cloudant database. It is easy to parse this variable into your app and extract the information that you need for a connection.

You are automatically assigned all of the credentials that you need to access the Cloudant database with the Cloudant Dashboard user interface.

6.10 Cloudant Dashboard

The screenshot shows the Cloudant Dashboard interface. At the top, there's a red header bar with the IBM logo on the right and a 'Powered by Redbooks' badge on the left. Below the header is a main title 'Cloudant Dashboard'. On the left side, there's a vertical navigation menu with the following items: Monitoring (selected), Databases (highlighted in orange), Replication, Analytics, Active Tasks, Account, Support, Documentation, and IBM Cloudant. The 'Databases' section contains a table titled 'Your Databases' with one entry: 'my_sample_db'. The table has columns for Name, Size, # of Docs, and Actions. The 'Actions' column for 'my_sample_db' includes icons for replication, locking, and deleting. At the bottom of the dashboard, it says 'Showing 1–1 of 1 databases.' and has a page number indicator '1'.

Figure 6-8 Cloudant Dashboard

Notes:

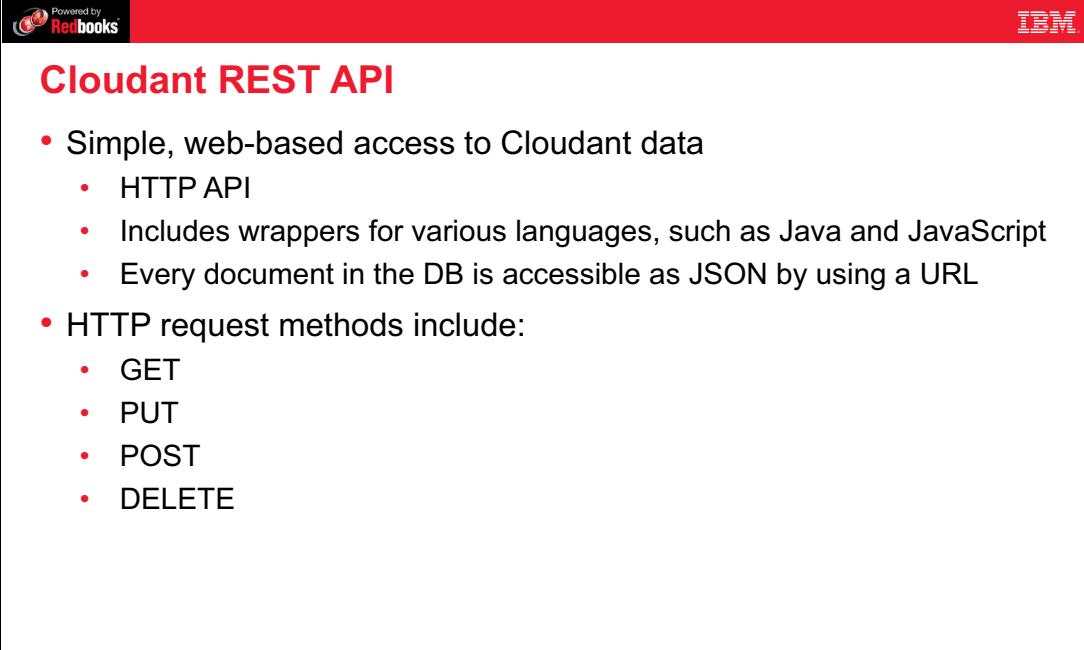
Cloudant Dashboard is a cloud-based web interface that makes it easy to develop, administer, and monitor your databases. You can perform many tasks, such as view and manage Cloudant databases, view documents, create documents, create and run queries, and manage the permissions to the database.

You can display the contents of a Cloudant document in IBM Bluemix by selecting the database. Then, select **All Documents** to display the list of documents. You can edit each of the documents in the list to display or modify the document contents.

The Cloudant Dashboard user interface is similar whether you use the following website or IBM Bluemix:

<https://cloudant.com/>

6.11 Cloudant REST API



The slide is titled "Cloudant REST API" in red text at the top center. In the top left corner, there is a "Powered by Redbooks" logo with a small globe icon. In the top right corner, the IBM logo is present. The main content area contains a bulleted list describing the Cloudant REST API:

- Simple, web-based access to Cloudant data
 - HTTP API
 - Includes wrappers for various languages, such as Java and JavaScript
 - Every document in the DB is accessible as JSON by using a URL
- HTTP request methods include:
 - GET
 - PUT
 - POST
 - DELETE

Figure 6-9 Cloudant REST API

Notes:

Cloudant uses a REST API to provide simple, web-based access to data in the Cloudant data store. The REST API is a programmatic way of accessing the data from your applications. It provides several REST access methods for data read, add, update, and delete functions.

The following HTTP Request methods can be used:

- ▶ GET: Request a specific JSON document
- ▶ PUT: Create databases and documents
- ▶ POST: Set values, and create documents
- ▶ DELETE: Delete a specific document

6.12 Sample database at Cloudant

The screenshot shows the Cloudant Sample Database dashboard. At the top, there's a red header bar with the Redbooks logo and the IBM logo. The main title is "Sample database at Cloudant". Below the title, a message states: "The Cloudant sample database that is created by the boilerplate is named my_sample_db:". Below this message is a URL: [https://\\$USERNAME.cloudant.com/my_sample_db/](https://$USERNAME.cloudant.com/my_sample_db/). The main content area is titled "Cloudant users and API keys with permissions on my_sample_db". It lists two entries: "77b8dcec-a2be-4107-ba3b-1dc993690b5e-bluemix" and "Unauthenticated connections". A table below shows permissions for each entry across four roles: _admin, _reader, _writer, and _replicator. For the first entry, all four checkboxes are checked. For the second entry, none are checked. At the bottom, there's a "Grant database permissions to:" input field and a "Grant Permissions" button.

Figure 6-10 Sample database at Cloudant

Notes:

The sample database that was created by Node.js Cloudant DB Web Starter Boilerplate is named `my_sample_db`.

You can get more information about the database by calling this REST API
`https://$USERNAME.cloudant.com/my_sample_db/` where `$USERNAME` is the username of Cloudant DB that you saw earlier in the Bluemix VCAP_SERVICES environment variables.

The permissions that are assigned to the creator of the database as seen in the Cloudant Dashboard UI are shown in Figure 6-10.

6.13 Reading a document in Cloudant

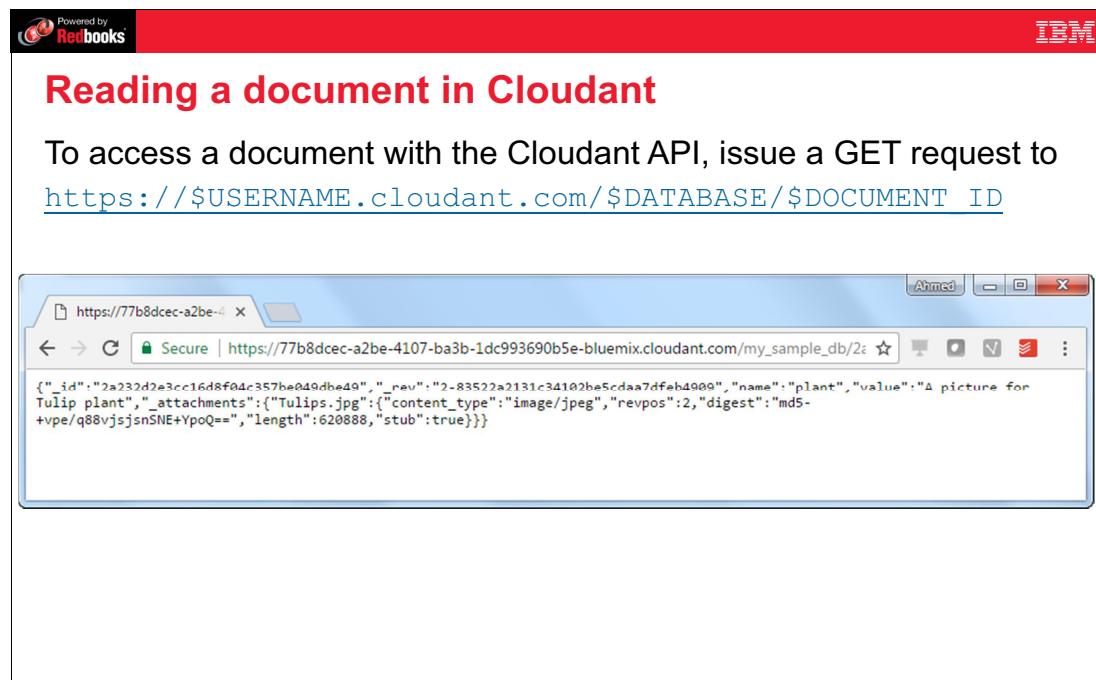


Figure 6-11 Reading a document in Cloudant

Notes:

Accessing a document with Cloudant Dashboard is shown in Figure 6-11. From your application, you can access the document through REST APIs.

To access a document with the Cloudant RESTful API, append the document ID to the URL of the database. The URL used to access this document in the Cloudant sample database is [https://\\$USERNAME.cloudant.com/\\$DATABASE/\\$DOCUMENT_ID](https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID), which is accessed by using a **GET HTTP REST request command** if you have sufficient permissions to access the DB. The `_id` is a unique key that is used when reading a document in the Cloudant database.

The sample document in the Cloudant Dashboard that is shown in Figure 6-11 includes three fields in addition to the mandatory fields `_id` and `_rev`: `name`, `company`, and `age`.

6.14 View all documents

The screenshot shows the IBM Cloudant Dashboard interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "View all documents" is displayed in red. A sub-instruction follows: "To view all documents at a database, issue a GET request to:" followed by a URL: [https://\\$USERNAME.cloudant.com/\\$DATABASE/_all_docs?include_docs=true](https://$USERNAME.cloudant.com/$DATABASE/_all_docs?include_docs=true). The main content area contains a table with two rows of data. The first row has an ID column with "id: \"c972849492809abc33c9ea8313b1a07f\"", a Rev column with "1", and a Body column with a truncated JSON object. The second row has an ID column with "id: \"c972849492809abc33c9ea8313b1a07f\"", a Rev column with "2", and a Body column with another truncated JSON object. To the right of the table, there's a "Document ID" dropdown, "Options" button, and an "API" button with a dropdown menu open. The API menu shows the URL https://2d02e29c-5104-466a-831d-2e07dd3a89ce.bluemix.cloudant.com/my_sample_db/_all_docs?include_docs=true, a "Copy URL" button, and a "View JSON" button.

Figure 6-12 View all Documents

Notes:

Cloudant includes an index called `_all_docs` with which you can use a URL to list all the documents in the database. You can pass to it an optional parameter called `include_docs` to return the contents of the documents, not just the `_id` and `_rev`.

As shown in Figure 6-12, you can click the API at any stage in Cloudant Dashboard to view the generated REST API for the page that the user is viewing in the Cloudant Dashboard.

6.15 More Cloudant REST APIs



More Cloudant REST APIs

- Create a document
POST `https://$USERNAME.cloudant.com/$DATABASE`
with the document's JSON content.
- Update a document
PUT `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID`
with the updated document JSON content.
- Delete a document
DELETE `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID?rev=$REV`

Figure 6-13 More Cloudant REST APIs

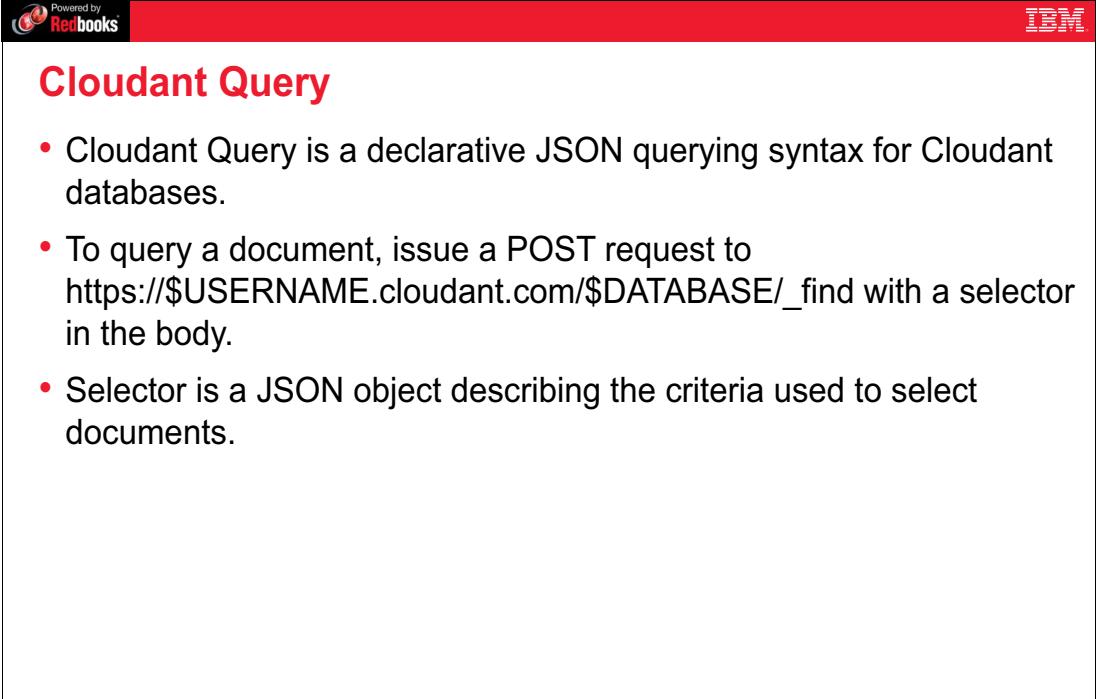
Notes:

To create a document, send a POST request to `https://$USERNAME.cloudant.com/$DATABASE` with the document's JSON content in the request body.

To update (or create) a document, send a PUT request to `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID` with the updated JSON content *and* the latest `_rev` value (not needed for creating documents) in the request body.

To delete a document, send a DELETE request to `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID?rev=$REV` where `$REV` is the document's latest `_rev`.

6.16 Cloudant query



The slide is titled "Cloudant Query" in red text. It features a bulleted list of four items describing Cloudant Query:

- Cloudant Query is a declarative JSON querying syntax for Cloudant databases.
- To query a document, issue a POST request to [https://\\$USERNAME.cloudant.com/\\$DATABASE/_find](https://$USERNAME.cloudant.com/$DATABASE/_find) with a selector in the body.
- Selector is a JSON object describing the criteria used to select documents.

Figure 6-14 Cloudant query

Notes:

Before you query for a specific field, it is suggested that an index is created for each field in the selector to optimize query performance.

6.17 HTTP status codes



Powered by
Redbooks

IBM

HTTP status codes

- Status and errors in Cloudant are reported by using a combination of the following data:
 - HTTP status code
 - Corresponding data in the body of the response data
- Example status codes:
 - 200 – OK
 - 201 – Created
 - 400 – Bad request
 - 401 – Unauthorized
 - 404 – Not Found
- Example detail that is supplied in JSON format, following a 404 status code:

```
{  
  "error": "not_found",  
  "reason": "missing"  
}
```

Figure 6-15 HTTP status codes

Notes:

Cloudant uses HTTP status codes that are returned in HTTP response headers.

More information might also be included in the response body area for the message.

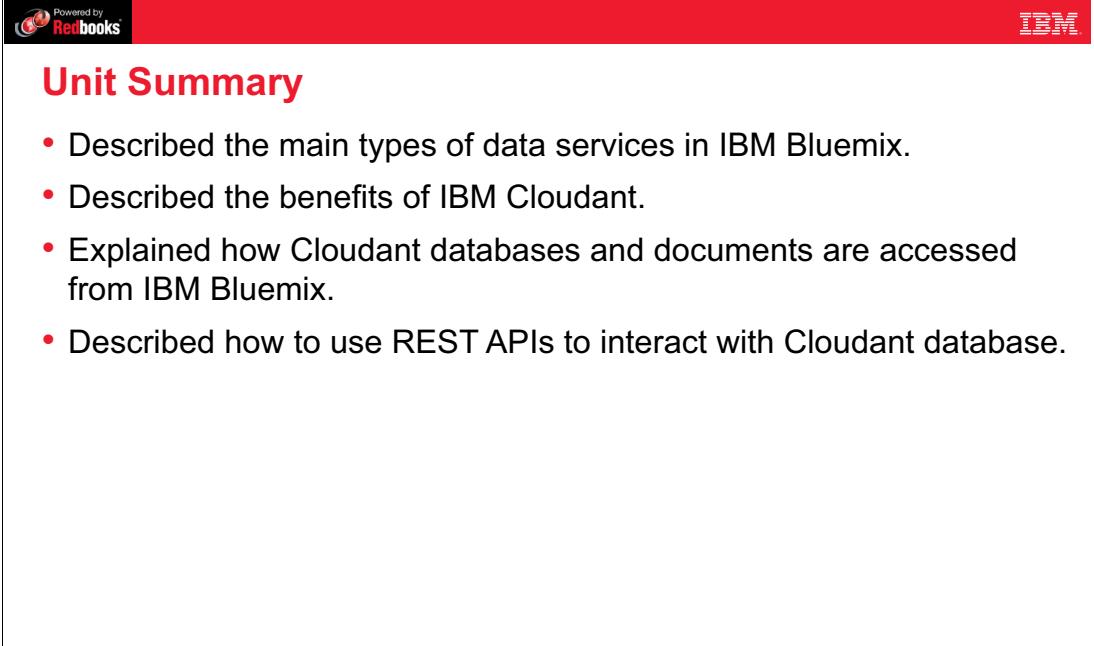
The following example status codes adhere to the widely-accepted status codes for HTTP:

- ▶ 200 – OK
- ▶ 201 – Created
- ▶ 400 – Bad request
- ▶ 401 – Unauthorized
- ▶ 404 – Not Found

For example, if you try to use `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID` to retrieve a document that does not exist in the database, Cloudant responds with status code 404 in the header, and more information about the error is returned in the response as JSON, as shown in Figure 6-15.

The language-specific libraries often include error handling for these various cases.

6.18 Unit summary

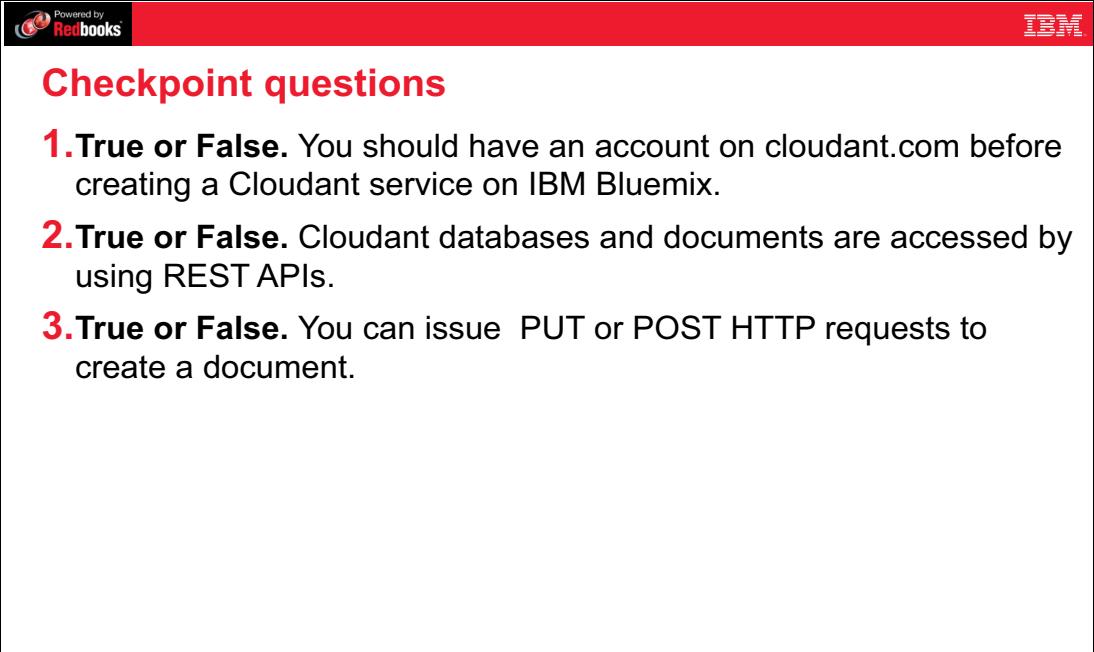


The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a title "Unit Summary" in bold red font. Below the title is a bulleted list of four items.

- Described the main types of data services in IBM Bluemix.
- Described the benefits of IBM Cloudant.
- Explained how Cloudant databases and documents are accessed from IBM Bluemix.
- Described how to use REST APIs to interact with Cloudant database.

Figure 6-16 Unit summary

6.19 Checkpoint questions



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a red title bar at the top with the text "Checkpoint questions" in white. Below this, there is a numbered list of three true/false questions.

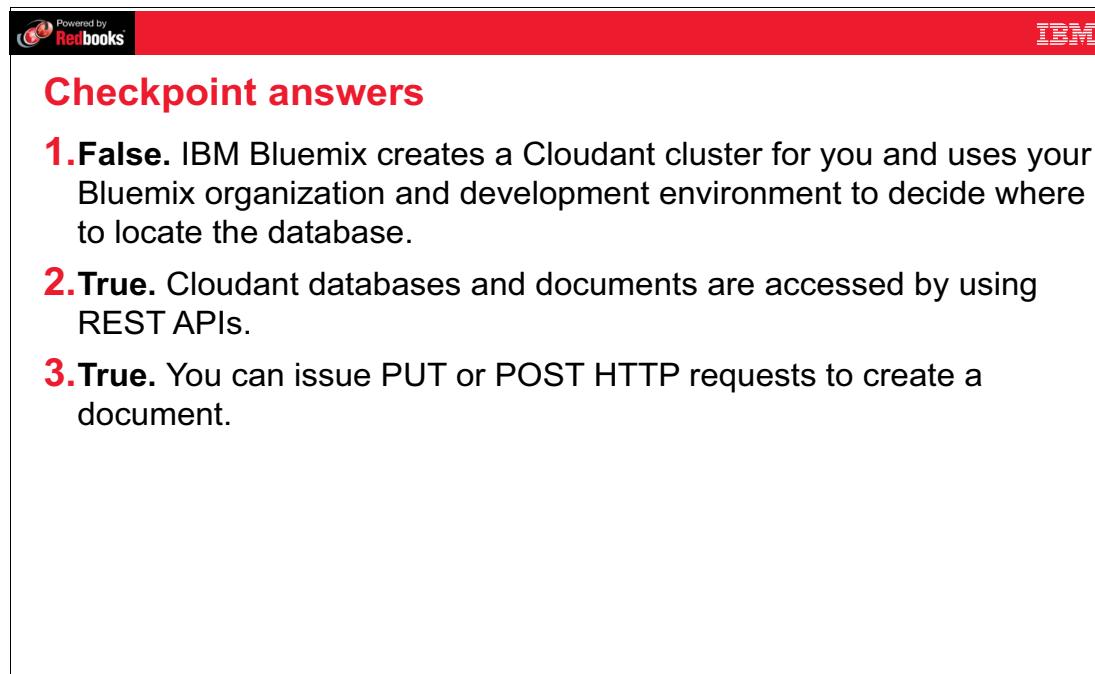
1. True or False. You should have an account on cloudant.com before creating a Cloudant service on IBM Bluemix.

2. True or False. Cloudant databases and documents are accessed by using REST APIs.

3. True or False. You can issue PUT or POST HTTP requests to create a document.

Figure 6-17 Checkpoint questions

6.20 Checkpoint answers



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a black border.

Checkpoint answers

- 1. False.** IBM Bluemix creates a Cloudant cluster for you and uses your Bluemix organization and development environment to decide where to locate the database.
- 2. True.** Cloudant databases and documents are accessed by using REST APIs.
- 3. True.** You can issue PUT or POST HTTP requests to create a document.

Figure 6-18 Checkpoint answers



IBM Bluemix mobile backend as a service

This unit covers the following topics:

- ▶ Introduction to mobile backend as a service (MBaaS)
- ▶ How does MBaaS work?
- ▶ Push Notifications service
- ▶ App ID service
- ▶ Kinetise service
- ▶ MobileFirst Services Starter Boilerplate

7.1 What you should be able to do

After completing this unit, you should be able to describe:

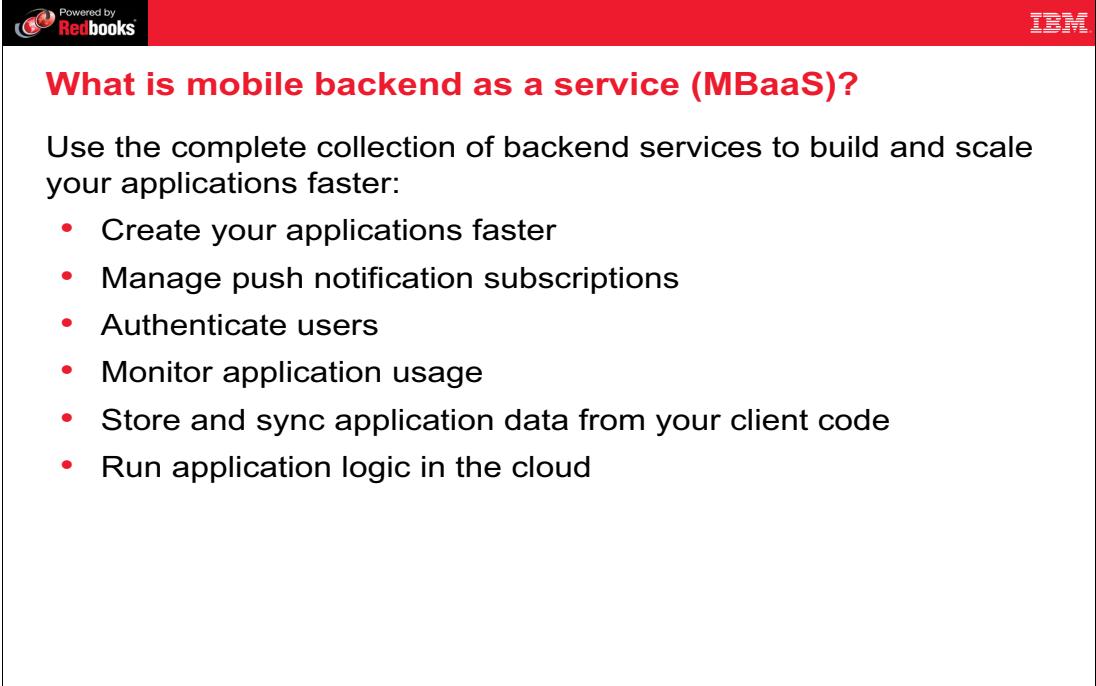
- ▶ IBM Bluemix mobile backend as a service (MBaaS)
- ▶ The MBaaS architecture
- ▶ The Push Notifications service
- ▶ The App ID service
- ▶ The Kinetise service
- ▶ How to create IBM Bluemix Mobile applications by using MobileFirst Services Starter Boilerplate

7.2 References

The following publications are useful for further research on the topic presented in this unit:

- ▶ Mobile Backend Services on Bluemix:
<https://console.ng.bluemix.net/docs/mobile/index.html>
- ▶ App ID:
 - <https://console.ng.bluemix.net/docs/services/appid/index.html>
 - <https://ibm.biz/BdjxpN>
- ▶ Push Notifications:
<https://console.ng.bluemix.net/docs/services/mobilepush/index.html>
- ▶ Kinetise:
 - <https://console.ng.bluemix.net/catalog/services/kinetise>
 - <https://helpcenter.kinetise.com/backend-developers>
 - https://bluemix.kinetise.com/bluemix_tutorial.html
- ▶ How to extend an Android application by using the IBM Push Notifications service on Bluemix:
<https://www.ibm.com/developerworks/library/mo-android-push-app/>

7.3 What is mobile backend as a service (MBaaS)?



The slide has a red header bar. On the left is the "Powered by Redbooks" logo, and on the right is the "IBM" logo. The main content area has a white background with a red border. The title "What is mobile backend as a service (MBaaS)?" is at the top in red. Below it is a sub-headline in black: "Use the complete collection of backend services to build and scale your applications faster:". A bulleted list follows:

- Create your applications faster
- Manage push notification subscriptions
- Authenticate users
- Monitor application usage
- Store and sync application data from your client code
- Run application logic in the cloud

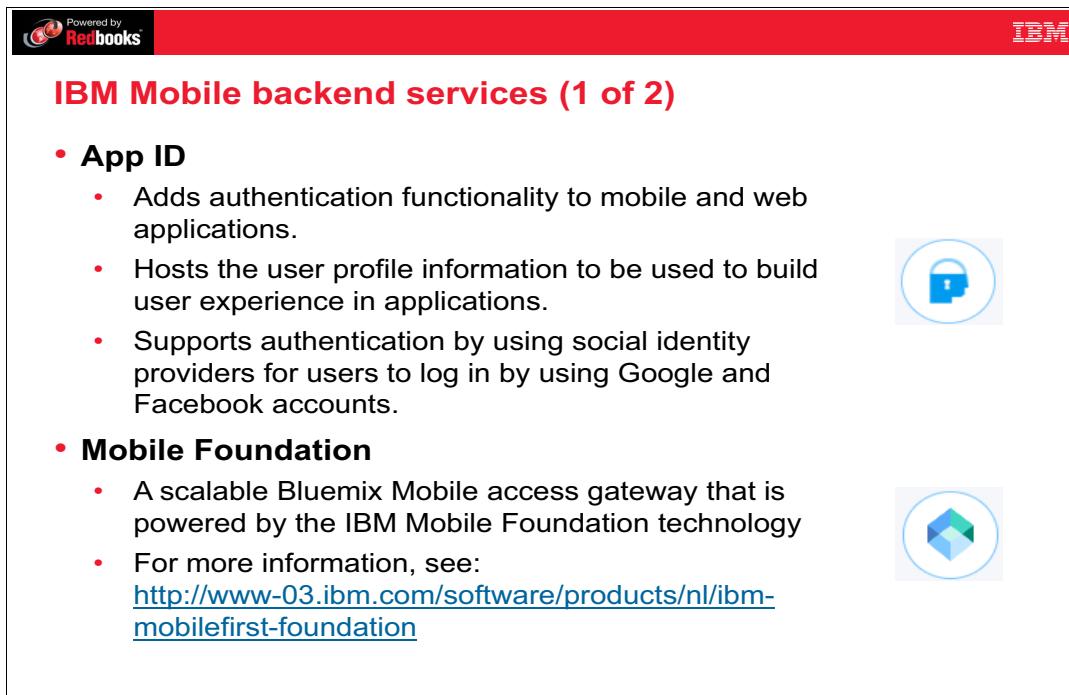
Figure 7-1 What is mobile backend as a service (MBaaS)?

Notes:

With IBM Bluemix mobile backend as a service (MBaaS), you can incorporate pre-built, managed, and scalable cloud services into your mobile applications without relying on IT involvement. By using the complete set of mobile backend services, you can complete the following tasks:

- ▶ Create your apps faster
- ▶ Manage push notification subscriptions
- ▶ Authenticate users
- ▶ Monitor app usage
- ▶ Store and sync app data from your client code
- ▶ Run app logic in the cloud

7.4 IBM Mobile backend services (1 of 2)



The slide is titled "IBM Mobile backend services (1 of 2)". It features a red header bar with the Redbooks logo and the IBM logo. The main content area contains two sections: "App ID" and "Mobile Foundation". Each section has a bulleted list of features and a corresponding circular icon.

- **App ID**
 - Adds authentication functionality to mobile and web applications.
 - Hosts the user profile information to be used to build user experience in applications.
 - Supports authentication by using social identity providers for users to log in by using Google and Facebook accounts.
- **Mobile Foundation**
 - A scalable Bluemix Mobile access gateway that is powered by the IBM Mobile Foundation technology
 - For more information, see:
<http://www-03.ibm.com/software/products/nl/ibm-mobilefirst-foundation>

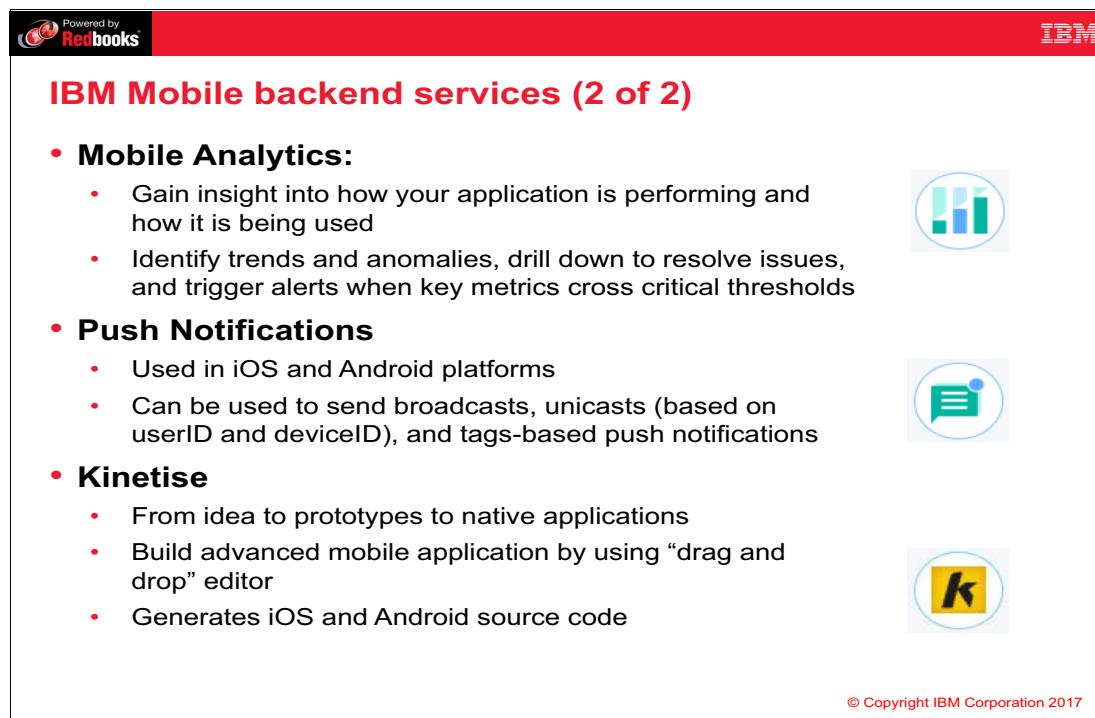
Figure 7-2 IBM Mobile backend services (1 of 2)

Notes:

The following IBM mobile services are provided on IBM Bluemix:

- ▶ App ID
 - Use the IBM Bluemix App ID service to add authentication to your mobile and web app. App ID supports authentication by using social identity providers, so that users can log in with their Facebook and Google accounts. You can also store user profile information that you can use to build engaging experiences.
- ▶ Mobile Foundation
 - The Mobile Foundation service is a scalable IBM Bluemix Mobile access gateway that is powered by the market-leading IBM Mobile Foundation technology. The service offers a comprehensive set of mobile backend capabilities, such as, App lifecycle, Push, Analytics, Feature Toggle, Security and Authentication, and offline synch. It also offers native and Cordova SDKs.

7.5 IBM Mobile backend services (2 of 2)



The slide is titled "IBM Mobile backend services (2 of 2)". It features three main sections: "Mobile Analytics" (with a bar chart icon), "Push Notifications" (with a speech bubble icon), and "Kinetise" (with a stylized 'k' icon). Each section contains a bulleted list of features.

- **Mobile Analytics:**
 - Gain insight into how your application is performing and how it is being used
 - Identify trends and anomalies, drill down to resolve issues, and trigger alerts when key metrics cross critical thresholds
- **Push Notifications**
 - Used in iOS and Android platforms
 - Can be used to send broadcasts, unicasts (based on userID and deviceID), and tags-based push notifications
- **Kinetise**
 - From idea to prototypes to native applications
 - Build advanced mobile application by using “drag and drop” editor
 - Generates iOS and Android source code

© Copyright IBM Corporation 2017

Figure 7-3 IBM Mobile backend services (2 of 2)

Notes:

► Mobile Analytics

Use IBM Mobile Analytics for Bluemix to gain insight into how your app is performing and how it is being used. Monitor performance and usage of all your applications from your desktop or tablet without creating queries or engaging a data analyst. Quickly identify trends and anomalies, drill down to resolve issues, and trigger alerts when key metrics cross critical thresholds.

► Push Notifications

Use the IBM Push Notifications for Bluemix service to send and manage mobile push notifications that are targeted to iOS and Android platforms. This service manages the mapping of your application users to their devices, device platform, and handles dispatching push notifications to the devices. By using this service, you can send broadcasts, unicasts (based on userID and deviceID), and tags (or topics) based on push notifications to your mobile application users.

► Kinetise

This service enables the rapid development of mobile apps, from idea to prototypes to native apps. Build advanced mobile app by using “drag and drop” editor. It also generates iOS and Android source code.

7.6 MBaaS architecture

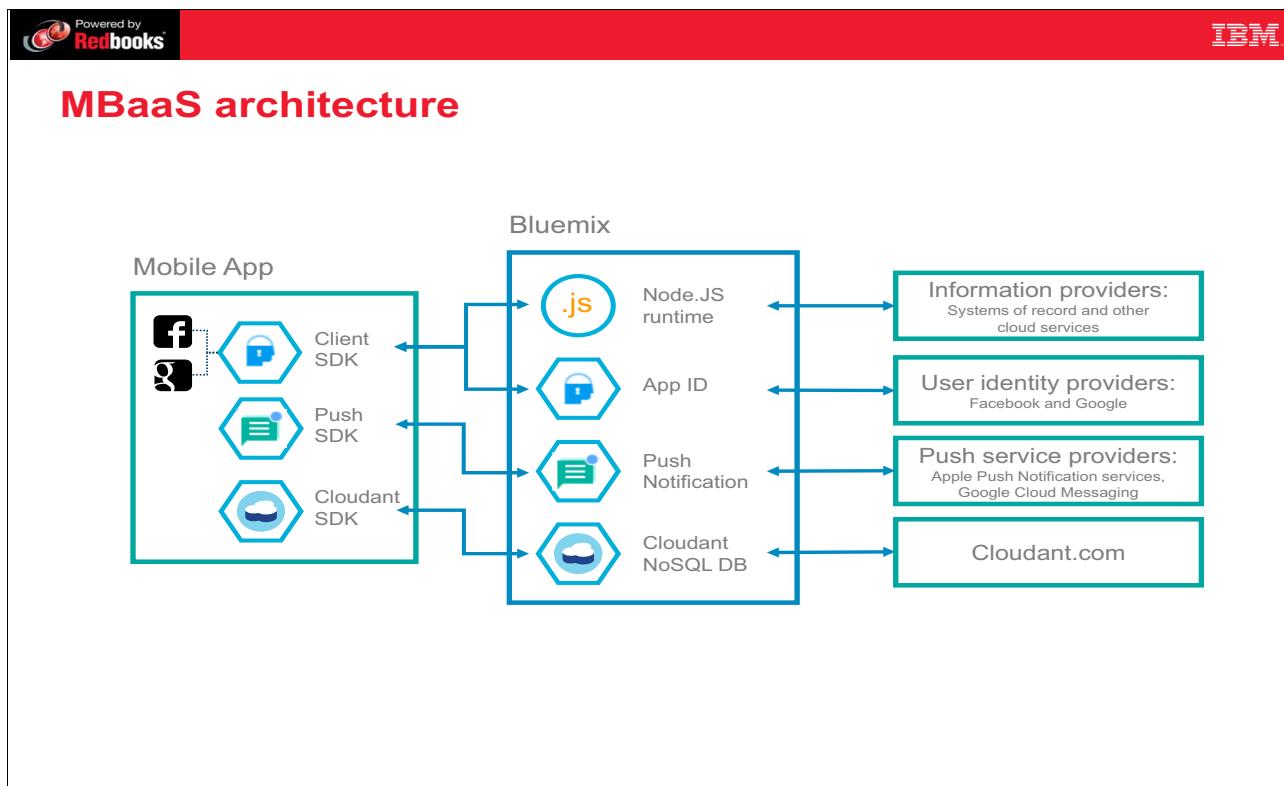


Figure 7-4 MBaaS architecture

Notes:

The structure of the APIs that use Node.js runtime and some of the backend services that are provided by IBM Bluemix is shown in Figure 7-4. Also shown is how the different components interact with each other in the mobile environment.

The Mobile App component, which is the front-end component, provides the following services:

- ▶ Client SDK is used to interact with the server-side components (Node.js runtime and App ID) to provide clients with Google and Facebook authentication.
- ▶ Push SDK to interact with the device-specific push service providers (iOS and Android only).
- ▶ Cloudant SDK to implement data persistence with the Cloudant NoSQL database service. It also provides offline data and backend synchronization.

IBM Bluemix component contains the mobile backend services attached to Node.js runtime:

- ▶ Node.js runtime interacts with external information providers as systems of record and other cloud services.
- ▶ App ID service interacts with the user identity providers, such as Google and Facebook accounts.

- ▶ Push Notifications service is configured to access Apple Push Notifications service (APNS) or Google Cloud Messaging (GCM) to send push notifications to mobile devices.
- ▶ Cloudant NoSQL DB service connects to the Cloudant.com data center to access and manage the database.

7.7 Push Notifications service

The screenshot shows a web page titled "Push Notifications service". At the top left is the Redbooks logo with the text "Powered by Redbooks". At the top right is the IBM logo. The main content area has a red header bar with the title. Below the header, there is a bulleted list of features:

- Push Notifications service provides a unified platform to send and manage mobile push notifications that are targeted to iOS and Android mobile platforms.
- Push Notifications service provides the following features:
 - Manages the mapping of your application users to their devices and device platform
 - Handles dispatching push notifications to devices
 - Sends:
 - Broadcasts
 - Unicasts (based on device ID)
 - Tag-based (or topics)
 - Platform-based to your mobile application users

Figure 7-5 Push Notifications service

Notes:

You can use the IBM Bluemix Push Notifications service to send push notifications with relevant content to the right people at the right place and time. Push notifications provide a programmable mechanism to send data to mobile devices. The mobile application logic determines how to handle push notifications.

The service provides a unified platform to send and manage mobile push notifications that are targeted to iOS and Android mobile platforms. The service provides the following features:

- ▶ Manages the mapping of your application users to their devices and device platform
- ▶ Handles dispatching push notifications to devices
- ▶ Sends broadcasts, unicasts, (based on deviceID), tag-based, and platform-based as push notifications to your mobile application users

7.8 Push notification process

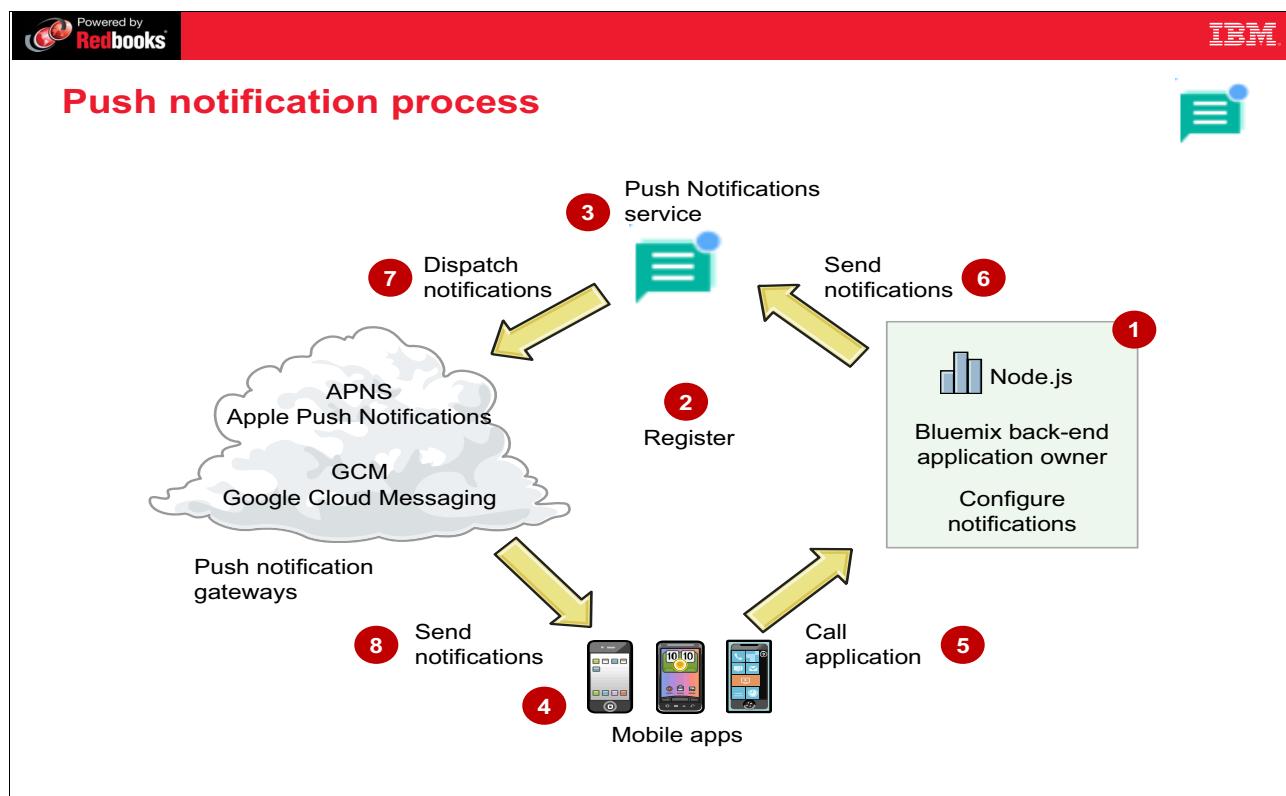


Figure 7-6 Push notification process

Notes:

As shown in Figure 7-6, a simplified push notification process consists of the following steps:

1. Create an IBM Bluemix application that uses the Push Notification service.
2. The application owner registers with Apple Push Notification Service (or APNS) for Apple devices and with Google Cloud Messaging, now migrated to Google Firebase Cloud Messaging, for Android devices.
3. Configure Push notifications in IBM Bluemix. Here, the credentials from the notification provider are used.
4. Create the mobile application by using IBM Bluemix or outside IBM Bluemix (Android studio or Xcode), that uses the Push Notifications service and APIs.
5. The mobile app calls the IBM Bluemix backend application.
6. The IBM Bluemix application uses the IBM Push API to send and dispatch notifications to the push notification gateway.
7. The push notification gateway sends notifications to the mobile devices.
8. You can send notifications to your app through both providers from a single interface by using the IBM Bluemix Push Notifications service.

7.9 Configuring push notifications in IBM Bluemix

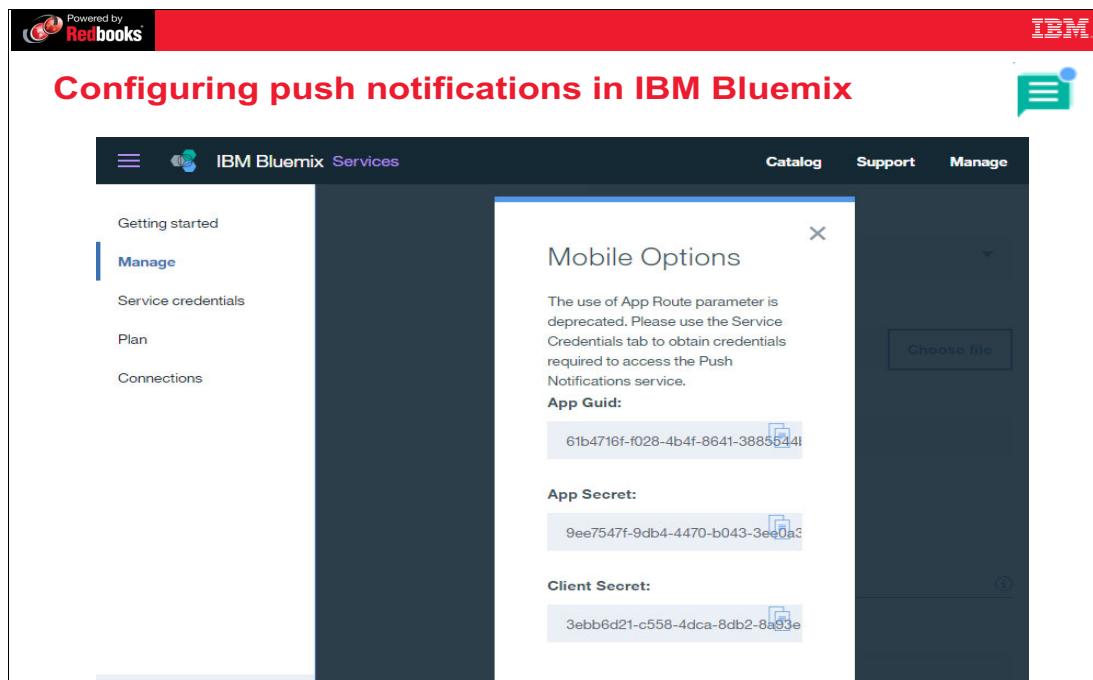


Figure 7-7 Configure push notifications in Bluemix

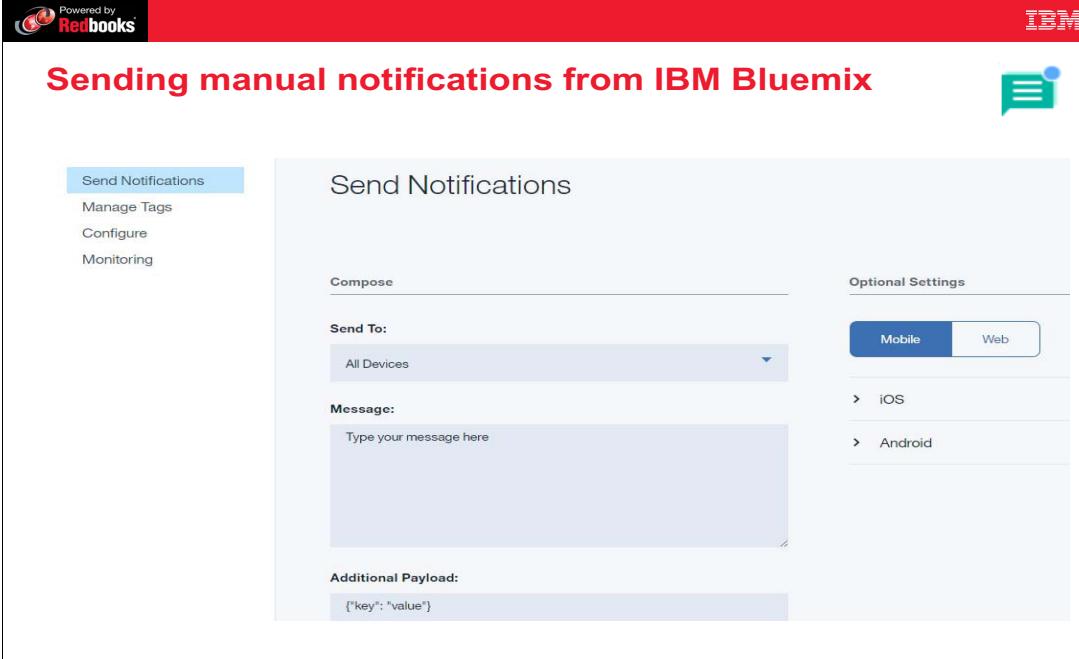
Notes:

After the application owner registers to use the Google Cloud Messaging or Apple Push Notifications APIs, you can use the credentials that are provided by the push notification provider to configure push notifications in IBM Bluemix.

The use of the App Guid, App Secret, and Client Secret (related to the Push Notification Service) to configure calling the service in the application is shown in Figure 7-7.

You can push your notification to mobile or web-based browsers, such as Google.

7.10 Sending manual notifications from Bluemix



The screenshot shows the 'Send Notifications' interface in the IBM Bluemix mobile backend service. The top navigation bar includes the Redbooks logo and the IBM logo. On the left, there's a sidebar with links for 'Send Notifications' (which is active and highlighted in blue), 'Manage Tags', 'Configure', and 'Monitoring'. The main content area has a title 'Send Notifications' and a 'Compose' input field. Below it, the 'Send To:' dropdown is set to 'All Devices'. The 'Message:' area contains a placeholder 'Type your message here'. To the right, there's an 'Optional Settings' section with tabs for 'Mobile' (selected) and 'Web', and sub-options for 'iOS' and 'Android'. At the bottom, there's an 'Additional Payload:' field containing the JSON object '{\"key\": \"value\"}'.

Figure 7-8 Sending manual notifications from IBM Bluemix

Notes:

You can send messages as a push notification to all devices that are subscribed to the push notification or to one of the following specific groups:

- ▶ Device by Tag: A Tag is a topic that a group of users is following, which is related to the business domain of the mobile application
- ▶ Device ID
- ▶ User ID
- ▶ Android devices
- ▶ IOS devices
- ▶ Web notifications
- ▶ Chrome apps and extensions
- ▶ Chrome browser
- ▶ Firefox browser
- ▶ Safari

You can also use the IBM Bluemix Push Notification Service Rest API to send push messages.

7.11 App ID service

The screenshot shows a web page titled "App ID service". At the top left is the "Powered by Redbooks" logo, and at the top right is the "IBM" logo. The main content area has a red header bar. Below the header, the title "App ID service" is displayed in red. To the right of the title is a blue icon of a padlock with a keyhole. The main content is a bulleted list:

- Adds authentication to your mobile or web application
- Hosts user profile information that you can use to build engaging user experiences
- You can configure client authentication so that users can log in to the application by using their Google or Facebook accounts
- App ID features the following components:
 - **Dashboard**
Used to configure various authentication types and see activity logs.
 - **Client SDK**
Build the mobile or web application to use the service to implement user authentication.
 - **Server SDK**
Used to protect resources that are hosted on IBM Bluemix.

Figure 7-9 App ID service

Notes:

You can add authentication to your mobile application by using the IBM App ID Bluemix service. You can configure client authentication and identity providers so that users can log in to the application by using their Google or Facebook accounts. You can also use the App ID service to protect Node.js and Swift applications that are hosted on IBM Bluemix.

You can also host user profile information that you can use to build engaging user experiences.

App ID features the following components:

- ▶ Dashboard: Download “Getting started” samples, see activity logs, configure various authentication types and identity providers.
- ▶ Client SDK: Build mobile and web applications that use the service to implement user authentication. The following platforms are supported:
 - iOS 9+, MacOS 10.11.5, Xcode 8.2
 - Android API 25 or higher, Java 8.x, Android SDK tools 25.2.5 or higher, Android SDK Platform Tools 25.0.3 or higher, and Android Build Tools version 25.0.2
- ▶ Server SDK
Protect resources that are hosted on IBM Bluemix. At the time of this writing, Node.js and Swift runtimes are supported.

7.12 App ID Authentication options

The screenshot shows a mobile application interface. At the top, there is a navigation bar with the Redbooks logo and the IBM logo. Below the navigation bar, the title 'App ID Authentication options' is displayed in red. To the right of the title is a blue circular icon with a white lock symbol. The main content area contains two sections: 'Facebook' and 'Google'. Each section has a bulleted list of instructions.

- **Facebook**
 - User logs in by using the Facebook credentials
 - You must create an application on the Facebook Developer Portal at <https://developers.facebook.com/docs/apps/register> to get the Facebook App ID
- **Google**
 - User logs in by using the Google+ credentials
 - You must create a project in Google Developer Console at <https://console.developers.google.com/apis/library> to get the Google Client ID

Figure 7-10 App ID Authentication options

Notes:

You can use the following types of authentication in your mobile application:

► Facebook

Users log in to the mobile application by using their Facebook credentials.

To start using Facebook as identity provider, you must create an application in the Facebook Developer Portal to get a Facebook Application ID, which is a unique identifier to let Facebook know which application is attempting to connect.

► Google

Users log in to the mobile application by using their Google credentials.

To start using Google as an identity provider, create a project in the Google Developer Console to obtain a Google Client ID. The Google Client ID is a unique identifier for your application used by Google authentication, and is needed for setting up the IBM Bluemix application.

7.13 App ID architecture

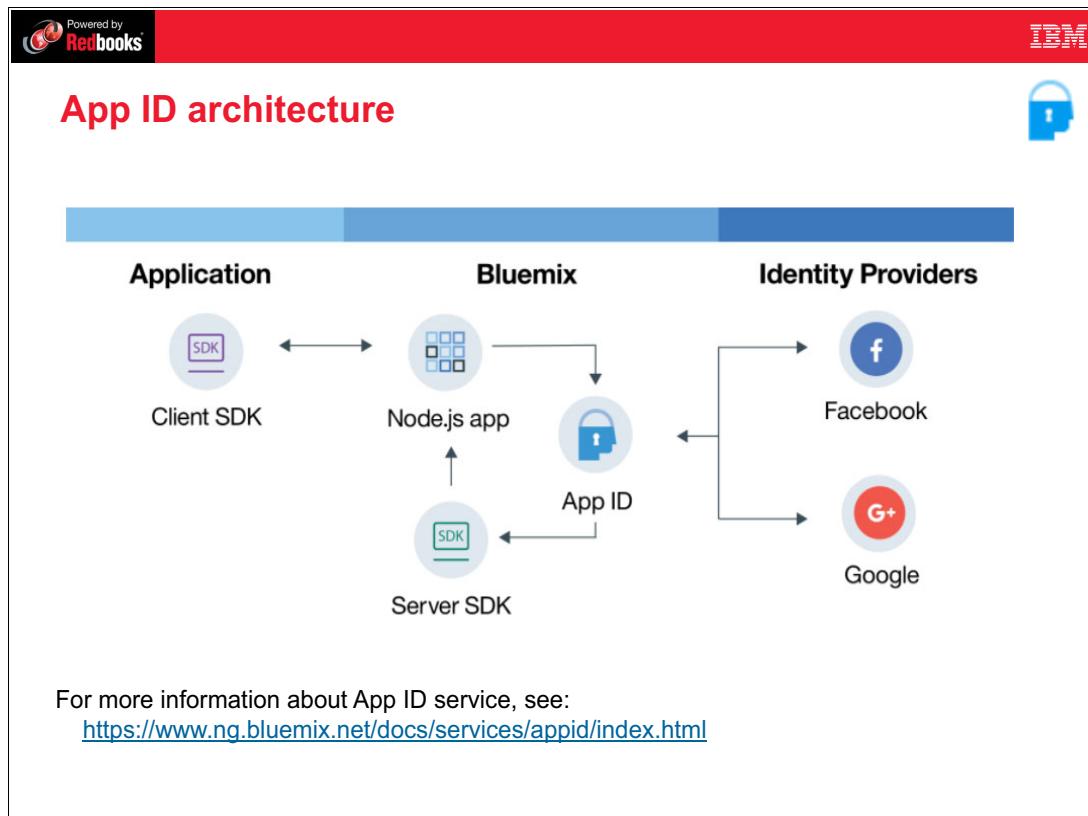


Figure 7-11 App ID architecture

Notes:

How the different components of App ID interact with each other is shown in Figure 7-11. Consider the following points:

- ▶ Use the App ID client SDK to make a request to your back-end resources that are protected with the App ID server SDK.
- ▶ The App ID server SDK detects an unauthorized request and returns an HTTP 401 and authorization scope.
- ▶ The client SDK automatically detects the HTTP 401 and starts the authentication process.
- ▶ When the client SDK contacts the service, the server SDK returns the login widget if more than one identity provider is configured. App ID calls the identity provider and presents the login form for that provider, or returns a grant code that allows them to authenticate if no identity providers are configured.
- ▶ App ID requests the client application to authenticate by supplying an authentication challenge.
- ▶ If Facebook or Google are configured and the user logs in, the authentication is handled by the respective identity provider OAuth Flow, which is an authorization framework that enables a third-party application to obtain limited access to an HTTP service.

- ▶ If the authentication ends with the same grant code, the code is sent to the token endpoint. The endpoint returns two tokens: An access token (for proving user's authorization to the resource) and an identity token (for proving authentication of the user).
From this point on, all requests that are made with the client SDK have a newly obtained authorization header.
- ▶ The client SDK automatically resends the original request that triggered the authorization flow.
- ▶ The server SDK extracts the authorization header from the request, validates the header with the service, and grants access to a back-end resource.

7.14 What is Kinetise?

The screenshot shows a web page with a red header bar. On the left of the header is the Redbooks logo, and on the right is the IBM logo. The main content area has a white background. At the top left of this area, the text 'What is Kinetise?' is displayed in red. To the right of this text is a yellow square containing a black stylized letter 'k'. Below this title, there is a bulleted list of features:

- Kinetise is a codeless mobile application development suite.
- It allows users to create even the most advanced applications in under 24 hours.
- Web-based “drag and drop” editor enables rapid prototyping, connecting to APIs, and creating fully-functional native iOS and Android builds.

Figure 7-12 What is Kinetise?

Notes:

Kinetise is a codeless mobile applications development suite with which users can create even the most advanced applications in less than 24 hours.

A web-based “drag and drop” editor enables rapid prototyping, connecting to APIs, and creating fully-functional native iOS and Android builds.

To start, select **Kinetise service** from your dashboard and then, click **OPEN KINETISE DASHBOARD**.

7.15 How to use Kinetise

The screenshot shows a web page with a red header bar. On the left of the header is the Redbooks logo, and on the right is the IBM logo. The main content area has a white background. At the top left of this area, the title 'How to use Kinetise' is displayed in red. To the right of the title is a yellow square containing a black stylized letter 'k'. Below the title is a bulleted list of six items:

- Applications that are built with Kinetise can connect to any **JSON or XML**-based RESTful API.
- Data can be retrieved, displayed, and sent out based on the user input.
- Kinetise allows users to create applications with user authentication. The data processed by the application is personalized and secured.
- The integration point between Kinetise and other Bluemix services happens by means of API URLs and request parameters that can be fully configured in the Kinetise editor.
- For more information about Kinetise, see:
<https://helpcenter.kinetise.com/backend-developers>

Figure 7-13 How to use Kinetise

Notes:

Applications that are built by using Kinetise can connect to any JSON or XML-based RESTful API. IBM Bluemix-related examples include Cloudant NoSQL DB or APIs that are defined in API Connect service.

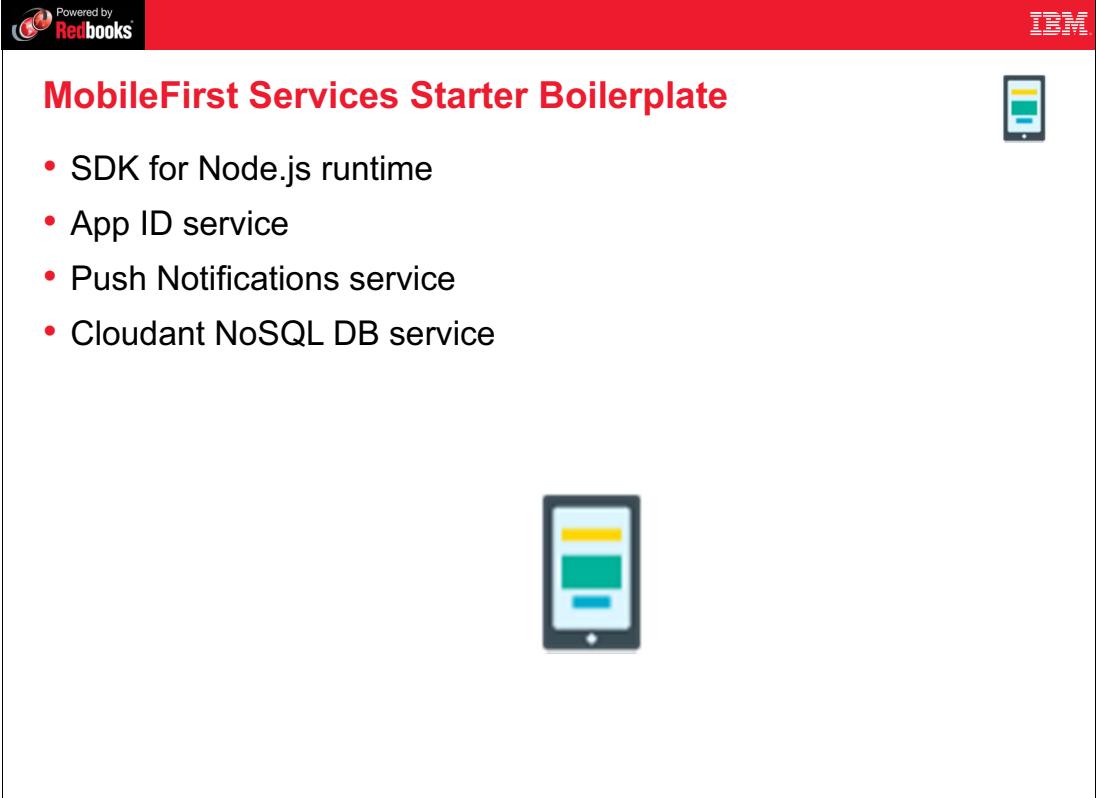
Data can be retrieved, displayed, and sent out based on the user input. Kinetise allows you to create applications with user authentication so that the data that is processed by the application is personalized and secured.

The integration point between Kinetise and other IBM Bluemix services occurs by means of API URLs and request parameters that can be fully configured in the Kinetise editor.

For more information about tutorials and samples of integrating mobile applications that are built in Kinetise with IBM Bluemix services and applications, see the following website:

<https://helpcenter.kinetise.com/backend-developers>

7.16 MobileFirst Services Starter Boilerplate



The image shows a screenshot of the 'MobileFirst Services Starter Boilerplate' landing page. At the top left is the Redbooks logo with the text 'Powered by Redbooks'. At the top right is the IBM logo. The main title 'MobileFirst Services Starter Boilerplate' is centered at the top in red text. Below the title is a bulleted list of features:

- SDK for Node.js runtime
- App ID service
- Push Notifications service
- Cloudant NoSQL DB service

In the center of the page is a small icon of a smartphone displaying a mobile interface with three colored bars (yellow, green, blue) on its screen.

Figure 7-14 MobileFirst Services Starter Boilerplate

Notes:

You can use all of the IBM Bluemix Mobile services together by using Bluemix IBM MobileFirst Services Starter Boilerplate, or you can use individual services from the IBM Bluemix catalog.

You can also use individual services if you want a different type of runtime for the back-end application rather than Node.js, or if you do not want to use all the services that are included with the Boilerplate.

The MobileFirst Services Starter Boilerplate is a template that features a runtime and a set of services to develop, deploy, and scale server-side JavaScript applications with ease. It includes the following features:

- ▶ App ID service
- ▶ Push notifications service
- ▶ Cloudant NoSQL DB service
- ▶ SDK for Node.js runtime

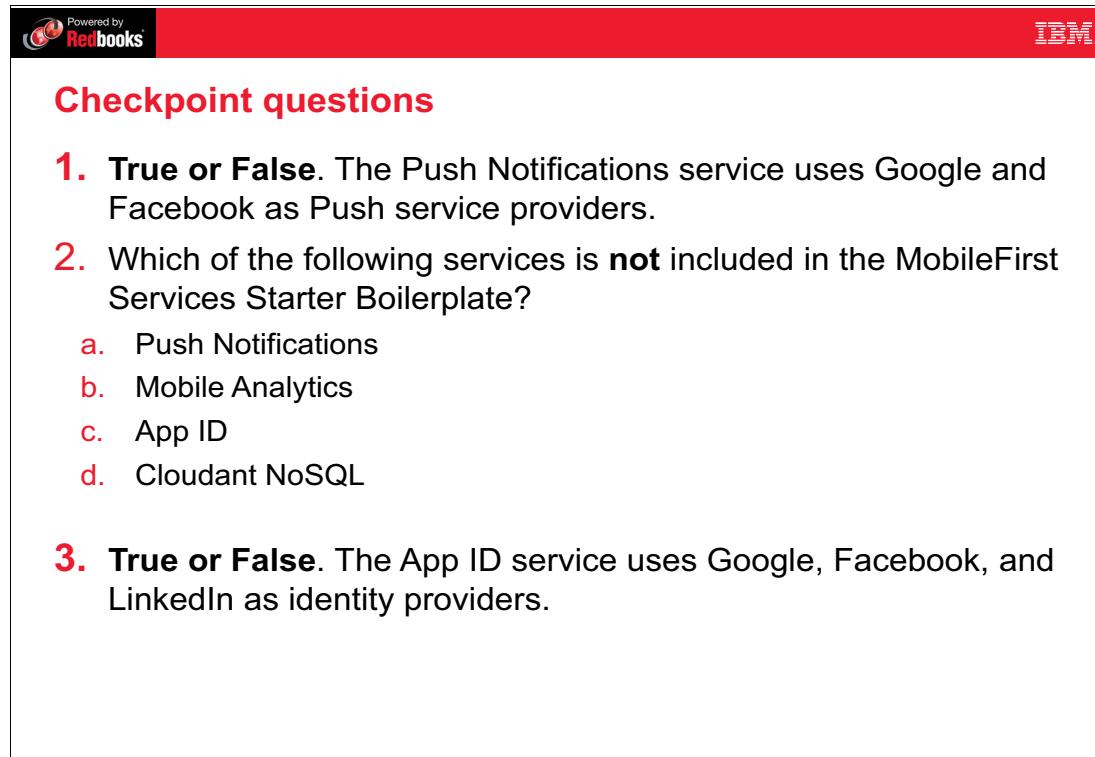
7.17 Unit summary

The slide has a red header bar. On the left of the bar is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background and a title "Unit summary" in red. Below the title is a bulleted list of six items.

- Described Bluemix Mobile Backend as a Service (MBaaS)
- Described the MBaaS architecture
- Described the Push Notifications service
- Described the App ID service
- Described the Kinetise service
- Described how to create Bluemix Mobile applications by using the MobileFirst Services Starter Boilerplate

Figure 7-15 Unit summary

7.18 Checkpoint questions



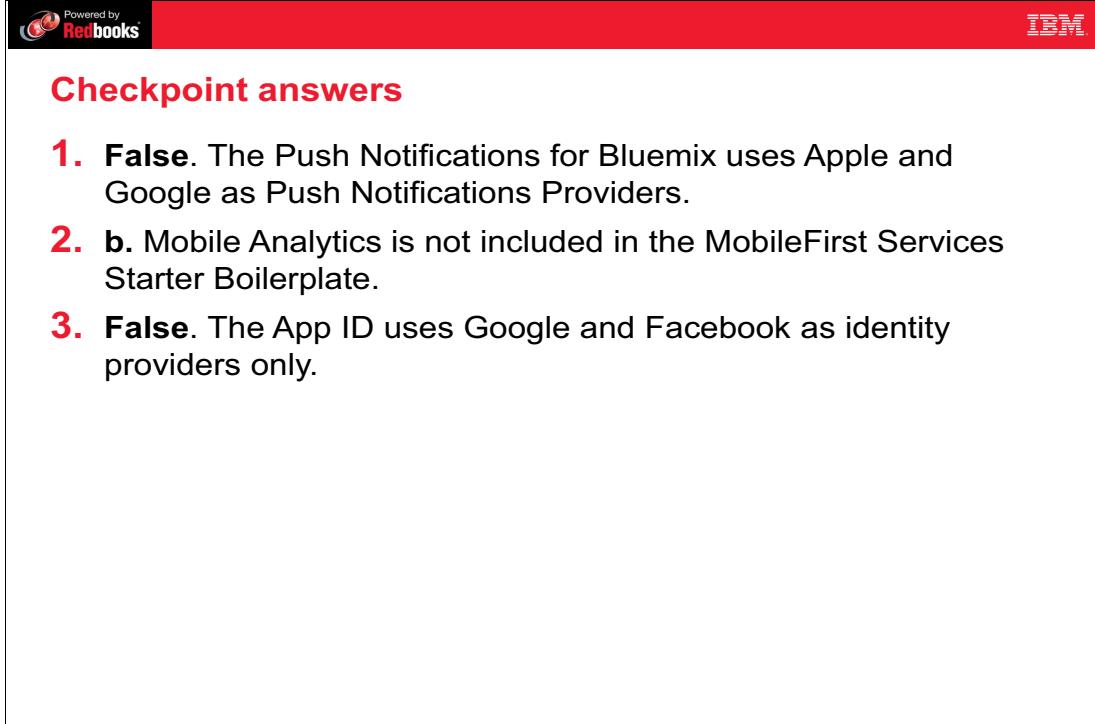
The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo.

Checkpoint questions

- 1. True or False.** The Push Notifications service uses Google and Facebook as Push service providers.
- 2.** Which of the following services is **not** included in the MobileFirst Services Starter Boilerplate?
 - a. Push Notifications
 - b. Mobile Analytics
 - c. App ID
 - d. Cloudant NoSQL
- 3. True or False.** The App ID service uses Google, Facebook, and LinkedIn as identity providers.

Figure 7-16 Checkpoint questions

7.19 Checkpoint answers



The slide is titled "Checkpoint answers" in red text. It features the Redbooks logo and the IBM logo at the top. The content area contains three numbered items:

1. **False.** The Push Notifications for Bluemix uses Apple and Google as Push Notifications Providers.
2. **b.** Mobile Analytics is not included in the MobileFirst Services Starter Boilerplate.
3. **False.** The App ID uses Google and Facebook as identity providers only.

Figure 7-17 Checkpoint answers

Redbooks

Essentials of Cloud Application Development on IBM Bluemix

(0.2"spine)
0.17" <-> 0.473"
90<->249 pages



SG24-8374-01

ISBN 0738442666

Printed in U.S.A.

Get connected

