

Essentials of Cloud Application Development on IBM Bluemix

Presentations Guide

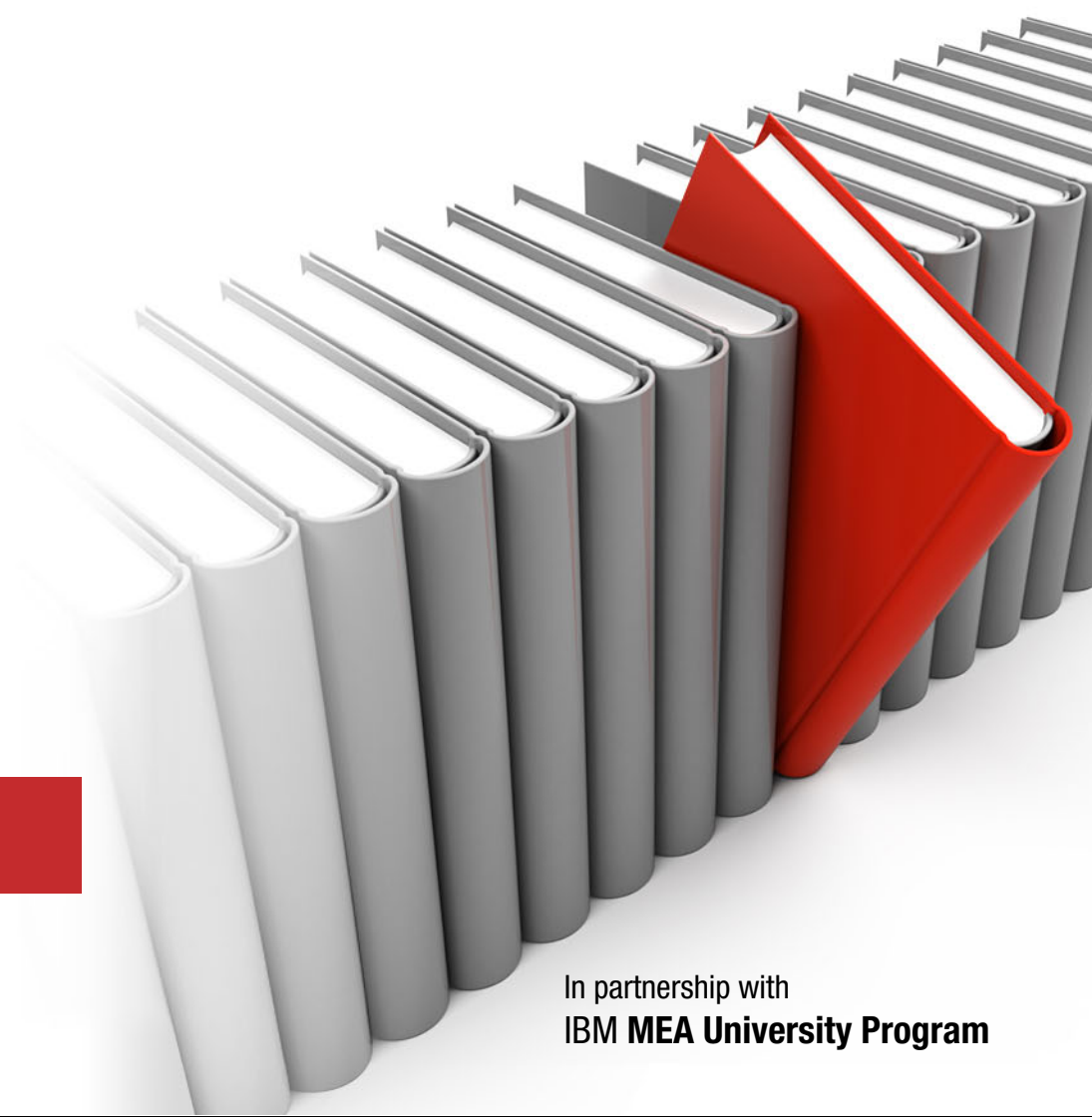
Hala A Aziz

Ahmed Azraq

Mohamed El-Khouly

Ben Smith

Sally Fikry



 **Cloud**

In partnership with
IBM MEA University Program



International Technical Support Organization

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

October 2016

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

First Edition (October 2016)

This edition applies to IBM Bluemix.

© Copyright International Business Machines Corporation 2016. 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!	xi
Comments welcome	xi
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 (IaaS) 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: Create 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 vs. Virtual Servers	32
2.8 Bluemix catalog: Boilerplates	34
2.9 Bluemix catalog: Runtimes	35
2.10 Bluemix catalog: Services	36
2.11 Create a Bluemix application (1 of 3)	37
2.12 Create a Bluemix application (2 of 3)	38
2.13 Bluemix application name must be unique across domain	39
2.14 Create a Bluemix application (3 of 3)	40
2.15 Bluemix application overview	41

2.16	Bluemix dashboard: Environment health	42
2.17	Testing applications through the application route	43
2.18	Add a Bluemix service.	44
2.19	Bind a service to an application	45
2.20	Bluemix environment variables	46
2.21	Bluemix organizations and users (team members)	47
2.22	Bluemix domains and quota	48
2.23	Bluemix organizations: Spaces	49
2.24	Bluemix organizations: User roles	50
2.25	Managing Bluemix organizations	51
2.26	Checkpoint questions	52
2.27	Checkpoint answers	53
Unit 3. Developing Bluemix applications from a local workstation		55
3.1	What you should be able to do	56
3.2	References	56
3.3	Create an application using the Bluemix dashboard	57
3.4	Node.js	58
3.5	Cloud Foundry command-line interface: Overview	59
3.6	Why use command-line tools?	60
3.7	Step 1: Review the documentation	61
3.8	Step 2: Install the Cloud Foundry command-line interface	62
3.9	Step 3: Connect to your IBM Bluemix account	63
3.10	Step 4: Deploy your application to IBM Bluemix	64
3.11	Step 5: Test your Bluemix application	65
3.12	IBM Bluemix and Eclipse	66
3.13	Step 1: Review the documentation	67
3.14	Step 2: Install Eclipse Luna and Eclipse tools for Bluemix	68
3.15	Step 3: Connect to your IBM Bluemix account	69
3.16	Step 4: Create a Node.js application project	70
3.17	Unit summary	71
3.18	Exercise 1 objectives	72
3.19	Exercise 2 objectives	73
3.20	Checkpoint questions	74
3.21	Checkpoint answers	75
Unit 4. Using IBM Bluemix DevOps services		77
4.1	What you should be able to do	78
4.2	References	78
4.3	What is DevOps?	79
4.4	Benefits of DevOps	80
4.5	What is IBM Bluemix DevOps Services?	81
4.6	What services does Bluemix DevOps provide?	82
4.7	Setting up a Bluemix DevOps Services project	83
4.8	Add Bluemix DevOps capabilities	84
4.9	Web IDE: Edit Code features	85
4.10	Web integrated development environment: Overview	86
4.11	Editing source code in Bluemix DevOps Services	87
4.12	Editor features: Code completion	88
4.13	Editor features: Run bar	89
4.14	Bluemix Live Sync features	90
4.15	Source control with a Git repository	92
4.16	Git repository overview	93

4.17	Connect a Git client to your Bluemix DevOps project	94
4.18	Bluemix DevOps Services: Build & Deploy	95
4.19	Customizing the delivery pipeline	96
4.20	Example: Default delivery pipeline	97
4.21	Example: Customizing the build stage	98
4.22	Example: Configuring build jobs	99
4.23	Example: Configuring deploy job	100
4.24	Example: A successful build and deploy result	101
4.25	Track & Plan Tools	102
4.26	Track & Plan Tools capabilities	103
4.27	Enable Track & Plan	105
4.28	Track & Plan: Example	106
4.29	Unit summary	107
4.30	Exercise 3 objectives	108
4.31	Checkpoint questions	109
4.32	Checkpoint answers	110
Unit 5. REST architecture		111
5.1	What you should be able to do	112
5.2	References	112
5.3	What is REST?	113
5.4	Applying REST to server-side applications	115
5.5	Example: Application model architecture for REST services	116
5.6	What is a RESTful web service?	117
5.7	Example: Sending an HTTP request to a REST service	118
5.8	Example: Receiving an HTTP response from a REST service	119
5.9	REST characteristics	120
5.10	Introduction to JSON	121
5.11	JSON data types	122
5.12	JSON data type: Objects	123
5.13	JSON data type: Arrays	124
5.14	What is Watson?	125
5.15	Watson Services in Bluemix	126
5.16	Watson API Explorer	127
5.17	Example: Watson API Explorer - AlchemyLanguage (Authors)	128
5.18	Checkpoint questions	130
5.19	Checkpoint answers	131
Unit 6. Introduction to data services in IBM Bluemix		133
6.1	What you should be able to do	134
6.2	References	134
6.3	Database choices on Bluemix	135
6.4	Data services in Bluemix catalog	136
6.5	Cloudant capabilities	137
6.6	Cloudant in IBM Bluemix vs. Cloudant.com	139
6.7	Documents in Cloudant	140
6.8	Getting started with Cloudant on IBM Bluemix	141
6.9	IBM Bluemix Cloudant: VCAP_SERVICES	142
6.10	Cloudant Dashboard	143
6.11	Cloudant REST API	144
6.12	Sample database at Cloudant	145
6.13	Reading a document in Cloudant	146
6.14	View all Documents	147

6.15 More Cloudant REST APIs	148
6.16 HTTP status codes	149
6.17 Exercise 4 objectives	150
6.18 Checkpoint questions	151
6.19 Checkpoint answers	152
Unit 7. IBM Bluemix Mobile Backend as a Service.	153
7.1 What you should be able to do	154
7.2 References	154
7.3 What is Mobile Backend as a Service (MBaaS)?	155
7.4 IBM Mobile backend services (1 of 2)	156
7.5 IBM Mobile backend services (2 of 2)	157
7.6 MBaaS architecture	158
7.7 Push Notifications service	160
7.8 Push notification process	161
7.9 Configure push notifications in Bluemix	162
7.10 Send manual notifications from Bluemix	163
7.11 Push notification message	164
7.12 Mobile Client Access (MCA) service	165
7.13 Mobile Client Access Authentication options	166
7.14 Mobile Client Access (MCA) architecture	167
7.15 Mobile Quality Assurance features	168
7.16 Mobile Quality Assurance window	170
7.17 Getting started with Mobile Quality Assurance	171
7.18 MobileFirst Services Starter Boilerplate	172
7.19 Exercise 5 objectives	173
7.20 Exercise 6 objectives	174
7.21 Checkpoint questions	175
7.22 Checkpoint answers	176

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](http://www.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.

AlchemyAPI®	developerWorks®	Rational®
Bluemix®	Global Business Services®	Rational Team Concert™
Cloudant®	IBM®	Redbooks®
Concert™	IBM MobileFirst™	Redbooks (logo)  ®
dashDB™	IBM Watson™	WebSphere®
DB2®	Jazz™	

The following terms are trademarks of other companies:

AlchemyAPI, and AlchemyAI logo are trademarks or registered trademarks of AlchemyAI, Inc., an IBM Company.

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

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

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

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

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

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 Middle East and Africa (MEA) University 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 benefit from this course.

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

- ▶ Describe the factors that lead to the adoption of cloud computing.
- ▶ Describe infrastructure as a service, platform as a service, and software as a service.
- ▶ Define cloud computing.
- ▶ Describe IBM Bluemix.
- ▶ Describe the architecture of IBM Bluemix.
- ▶ Identify the runtimes and services that Bluemix offers.
- ▶ Explain how to get started with Bluemix.
- ▶ Describe Bluemix organizations, domains, spaces, and users.
- ▶ Create Bluemix applications.
- ▶ Use services in a Bluemix application.
- ▶ Set environmental variables that are used with Bluemix services.
- ▶ Deploy and run Bluemix applications.
- ▶ Describe how to create an IBM SDK for Node.js application that runs on Bluemix.
- ▶ Explain how to manage a Bluemix account with the Cloud Foundry CLI.
- ▶ Describe how to integrate workstation development platforms with Bluemix.
- ▶ Manage application code and assets with IBM Bluemix DevOps services.
- ▶ Work with the Git repository that is used by DevOps services.
- ▶ Describe the characteristics of REST APIs.
- ▶ Describe the use of JSON as the preferred data format for REST APIs.
- ▶ Identify the data services that are available on Bluemix.
- ▶ Describe the features in Bluemix for developing mobile applications.
- ▶ Create a MobileFirst Services Starter application on Bluemix.
- ▶ Send push notifications from Bluemix and receive them on the mobile device emulator.

The workshop materials were created in August 2016. Thus, all IBM Bluemix features discussed in this Presentations Guide and Bluemix user interfaces used in the examples are current as of August 2016.

Note: This IBM Redbooks publication references exercises that are *not* included with this book. The exercises are only available to students attending 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 MEA University Programs.

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. She has also delivered IBM internal education and client enablement training workshops around the world.

Ahmed Azraq is a Certified IT Specialist in IBM Egypt. Since he joined IBM in 2012, he has worked as a senior cloud developer, technical team leader, and architect in the 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, Java, and IBM Watson™. Ahmed has acquired several professional certifications, including Open Group IT Specialist, IBM Bluemix, Java EE, IBM Business Process Manager, Agile development process, and IBM Design Thinking. Ahmed has delivered trainings on IBM Bluemix, DevOps, Node.JS, and IBM WebSphere Liberty Profile to IBM clients, business partners, and 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 known as Cognitive Build and the IBM Service Excellence Award for showing excellent client value behaviors and knowledge sharing.

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

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. He defended his dissertation, "Empirically Developing a Software Security Test Pattern Catalog Using a Grounded Theory Approach," and received his Doctorate in Computer Science in July of 2012, advised by Dr. Laurie Williams. 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. Further, Ben has 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.

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 played different roles including supporting pre-sales activities, consulting, solution design, and 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.

The project that produced this publication was managed by **Marcela Adan** and **Vasfi Gucer**, IBM Redbooks Project Leaders, ITSO.

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

Juan Pablo Napoli
MEA University Program Manager

Warren Fung
Client Technical Engagement, IBM Cloud

Keith Tilley
Client Technical Engagement, IBM Cloud

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 developers have 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 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
<http://www.ibm.com/developerworks/cloud/library/cl-bluemix-fundamentals-advantages-for-developers-trs/index.html>

1.3 What is cloud computing?

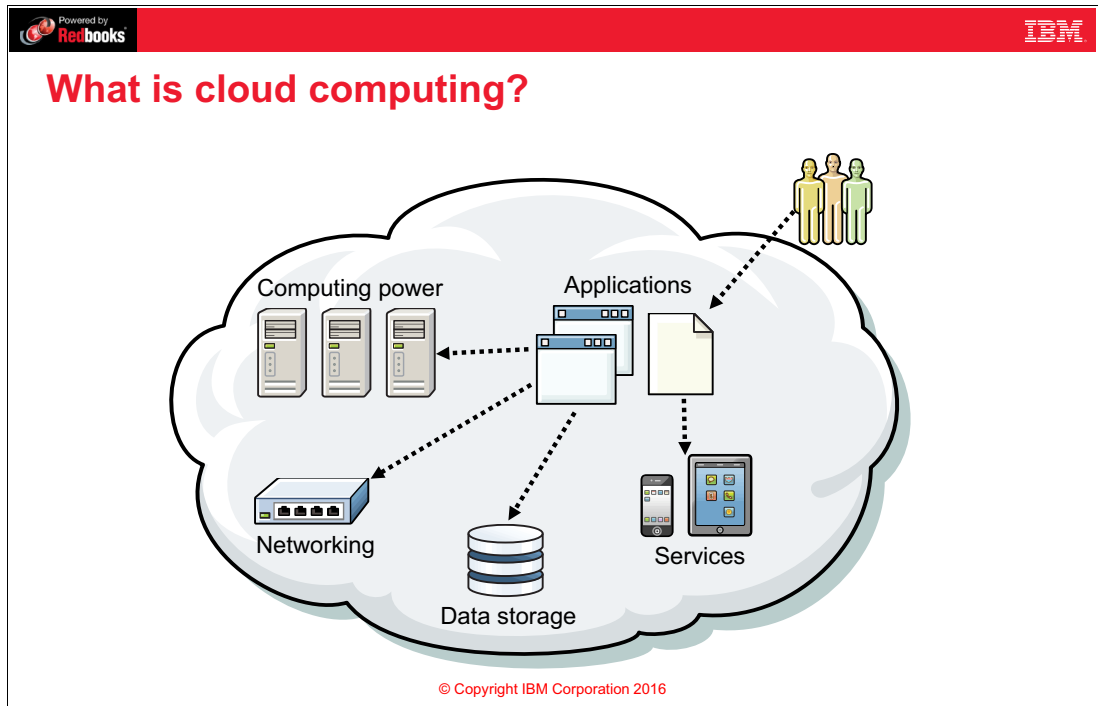


Figure 1-1 What is cloud computing?

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...

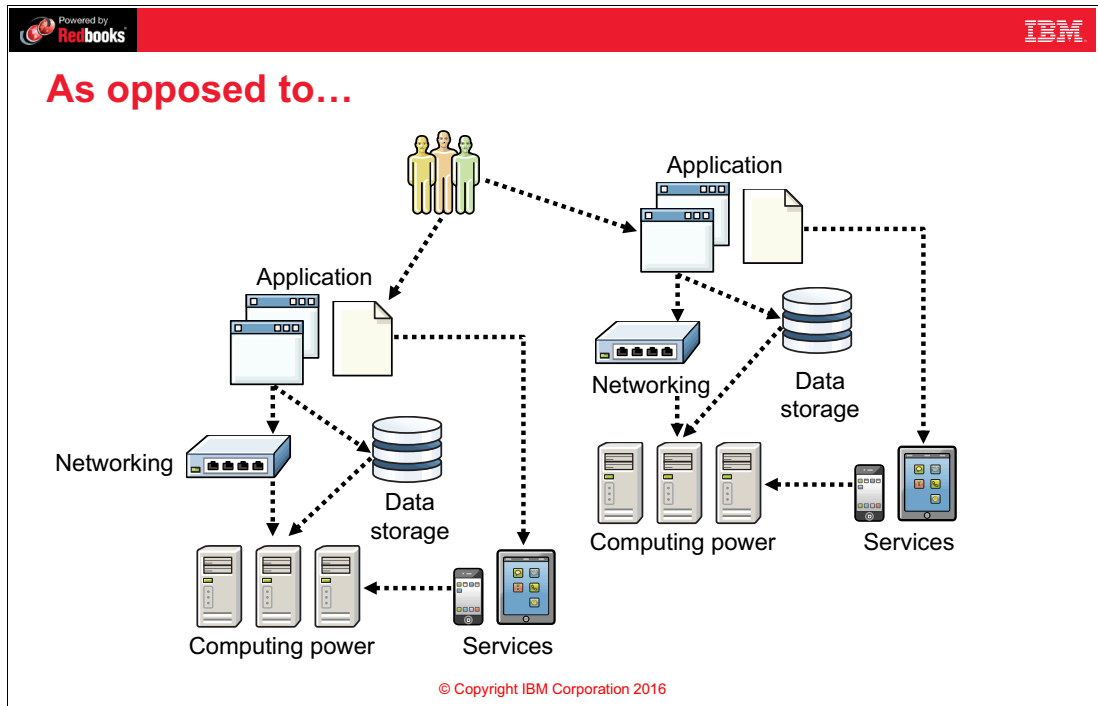


Figure 1-2 As opposed to...

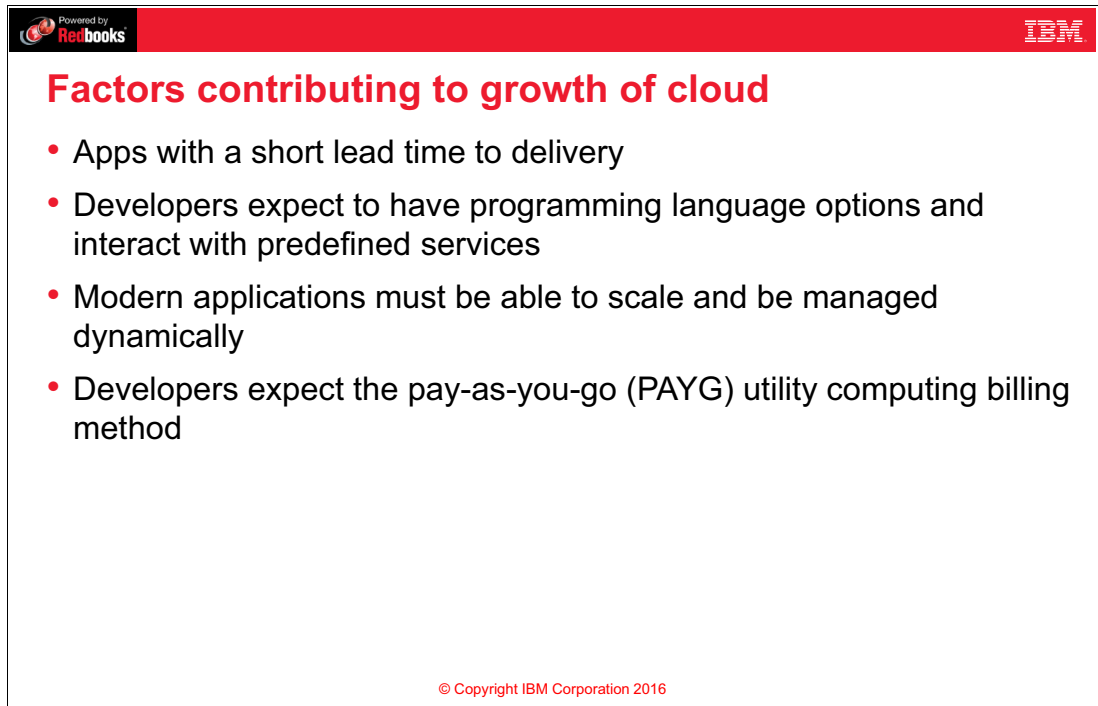
Notes:

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



Powered by Redbooks IBM

Factors contributing to growth of cloud

- Apps 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

© Copyright IBM Corporation 2016

Figure 1-3 Factors contributing to growth of cloud

Notes:

One factor contributing to the growth of cloud computing is that today's apps 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 given app (*scaling up*, or *vertical scaling*), or add more duplicate instances of an app (*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

Powered by Redbooks IBM

Cloud and mobile computing are changing traditional IT

- 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

© Copyright IBM Corporation 2016

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 lifecycle. 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

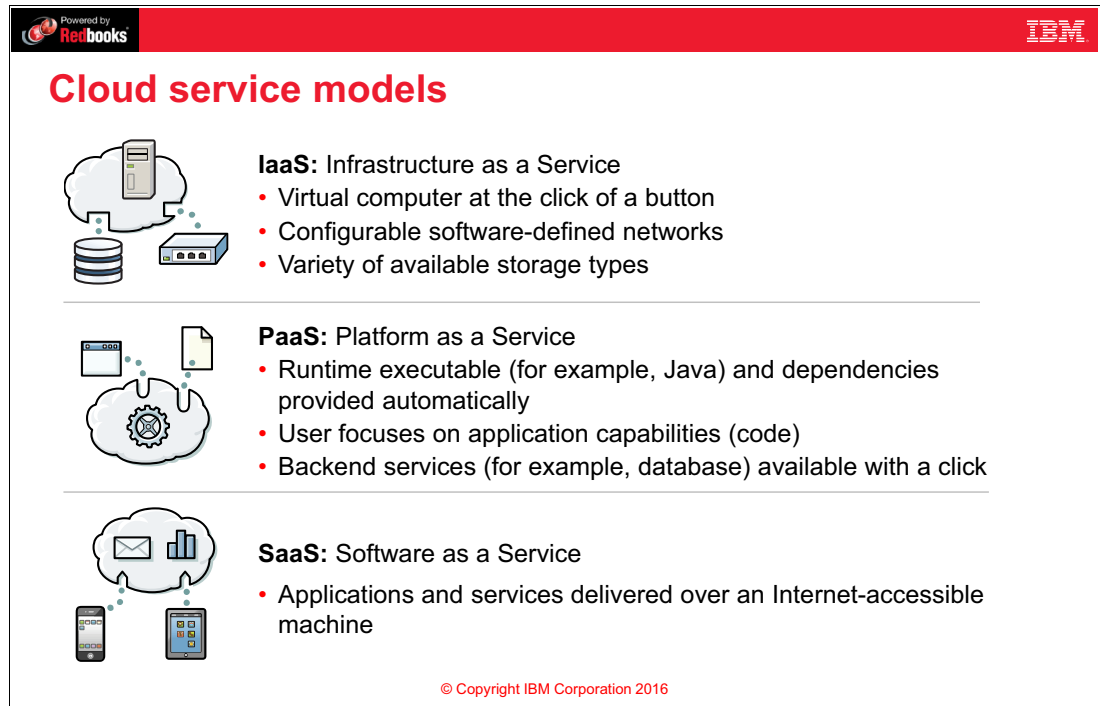


Figure 1-5 Cloud service models



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 may 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 (IaaS) architecture

Powered by 

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.

© Copyright IBM Corporation 2016

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 running on a given 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, setting up basic networking, and so on.

1.9 IBM SoftLayer: IaaS offering from IBM

The screenshot displays the IBM SoftLayer user interface. At the top, there is a navigation bar with links for HELP, CONTACT, NOTIFICATIONS, OPEN TICKETS, and user information (AHMED AZIAG, IBM - SUPPLY CHAIN VISIBILITY PLATFORM MVP (551272), LOG OUT). The main header includes the SoftLayer logo and navigation tabs for Devices, Storage, Network, Security, Services, Support, and Account. A search bar is also present.

The main content area is divided into several sections:

- Account Summary:** Shows a Current Balance of \$0.00 and an Estimated Next Balance of \$945.30. It also indicates the Current Payment Method as Manual.
- Scheduled Maintenance & Updates:** Displays two calendar views for September 2016 and October 2016, showing dates and times for scheduled events.
- Order:** Features icons for Devices, Storage, Network, Security, and Add-On Services.
- Bandwidth:** Lists bandwidth usage for three IP addresses: svp.poc-15.ibm.com (159.122.22.8), svp.poc-16.ibm.com (159.122.22.4), and svp.poc-12.ibm.com (159.122.22.17), all showing usage below 1%.

At the bottom of the screenshot, there is a copyright notice: © Copyright IBM Corporation 2016.

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 due to the easy access it provides to “bare metal” servers.

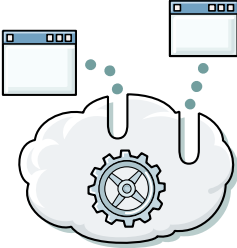
In addition to virtual servers, SoftLayer offers bare metal servers, which provide the raw horsepower that some organizations require for processor-intensive and disk I/O-intensive workloads.

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.



The diagram illustrates the PaaS architecture. It features a central cloud containing a gear, symbolizing the platform's internal processes. Two application windows are shown above the cloud, connected by dotted lines, representing applications running on the platform.

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

© Copyright IBM Corporation 2016

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

Notes:

Platform as a service (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 app code, or pointing to it and letting the PaaS complete the following tasks:

1. Obtain the runtime binaries and dependencies for the app.
2. Structure their app bits into the proper directory tree for containerization.
3. Provision a container (or set of containers) on which the app can run.
4. Automatically generate simple and basic networking configuration for access to the application.
5. Provide automatic and built-in monitoring of the app.
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. Applications are rapidly composed 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 app (*scaling up* or *vertical scaling*) or more instances of the app 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

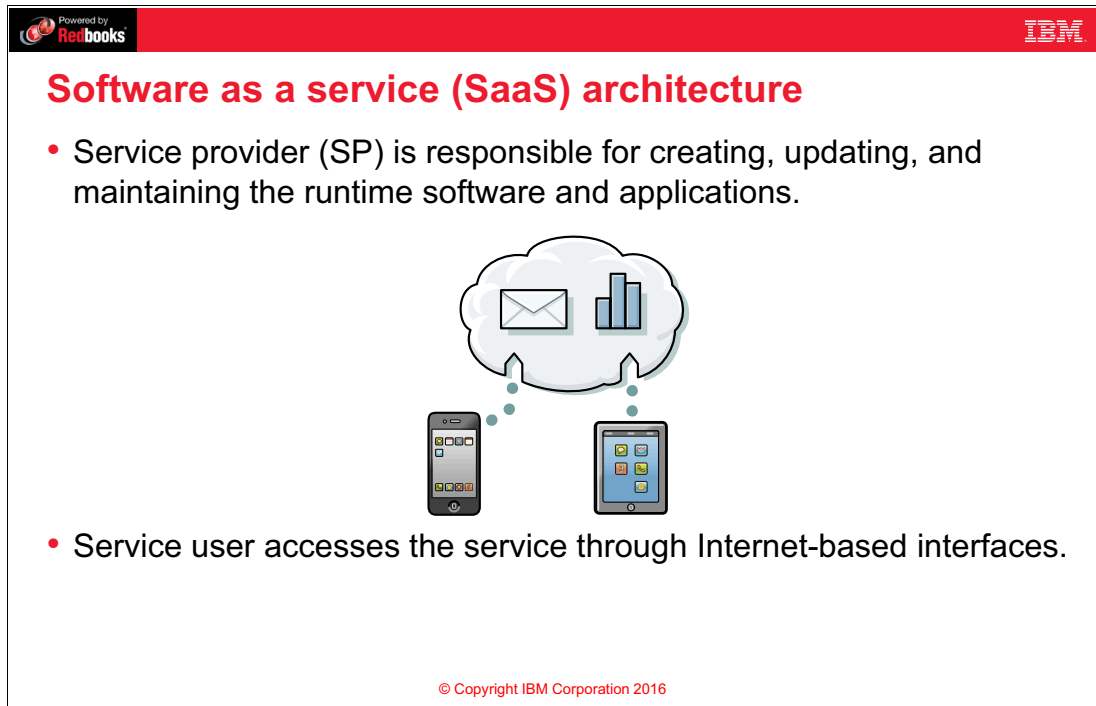


Figure 1-9 Software as a service (SaaS) architecture

Notes:

SaaS is a delivery model that provides access to capabilities through web-based services. Software as a service 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. Software as a service removes the need for organizations to handle installation, setup, and maintenance.

Sometimes SaaS apps are free and providers generate revenue, for example, from web ads. Alternatively, SaaS application providers generate revenue directly from the usage of the service. Do these scenarios sound familiar? That may be 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, then you are familiar with this cloud service model. These types of applications are just a couple of examples. There are literally thousands of SaaS applications, and the number grows daily, primarily because of Web 2.0 technologies.

Under the SaaS model, the software provider is responsible for the creation, updating, and maintenance of 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 itself, and not the platform or the infrastructure that the service is running on. 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

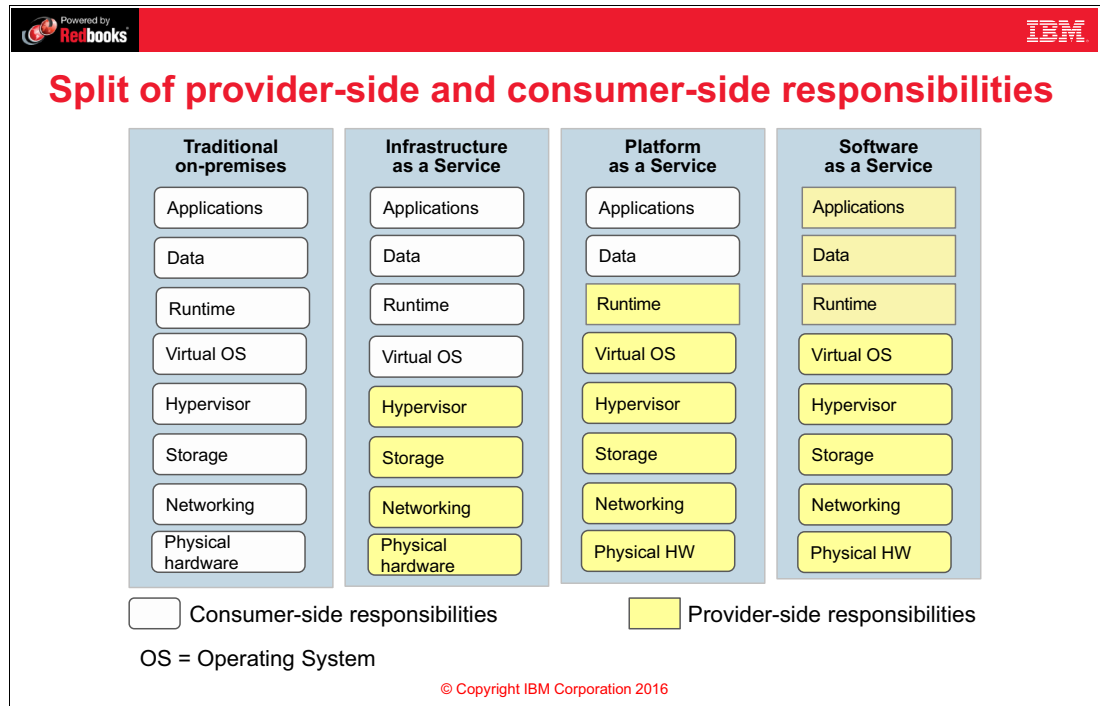


Figure 1-10 Split of provider-side and consumer-side responsibilities

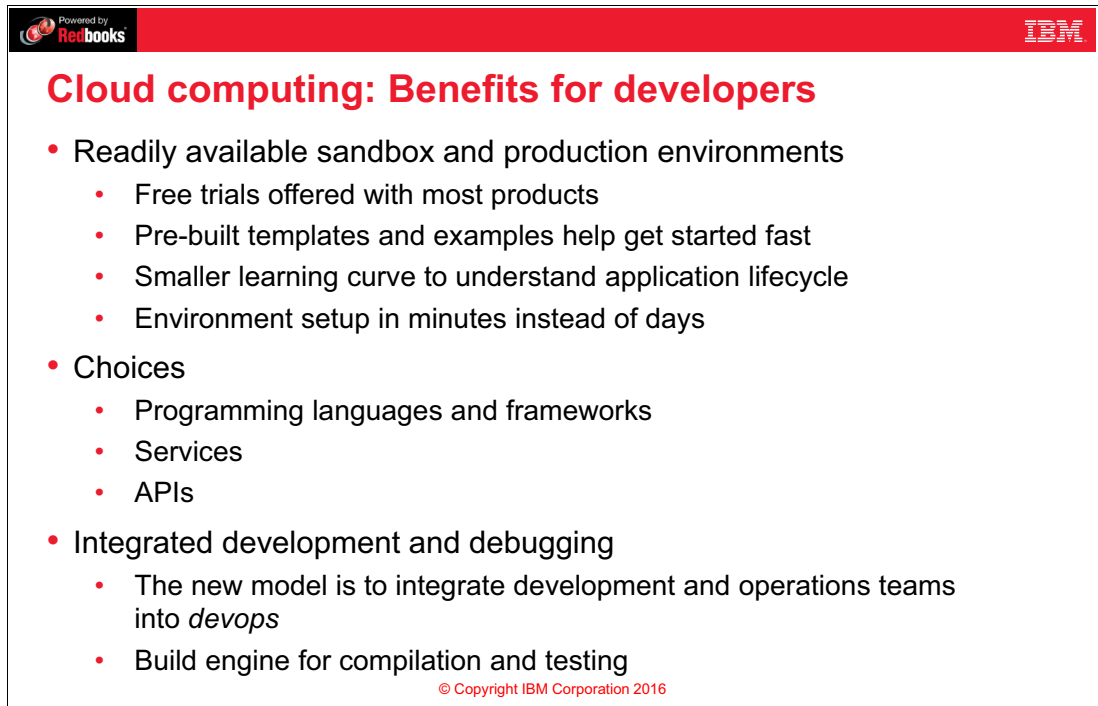
Notes:

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

Typically the cost goes down as you move to the right in the figure scenarios, but the flexibility goes down 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 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 that the hypervisor runs on.

1.13 Cloud computing: Benefits for developers



Powered by Redbooks

IBM

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

© Copyright IBM Corporation 2016

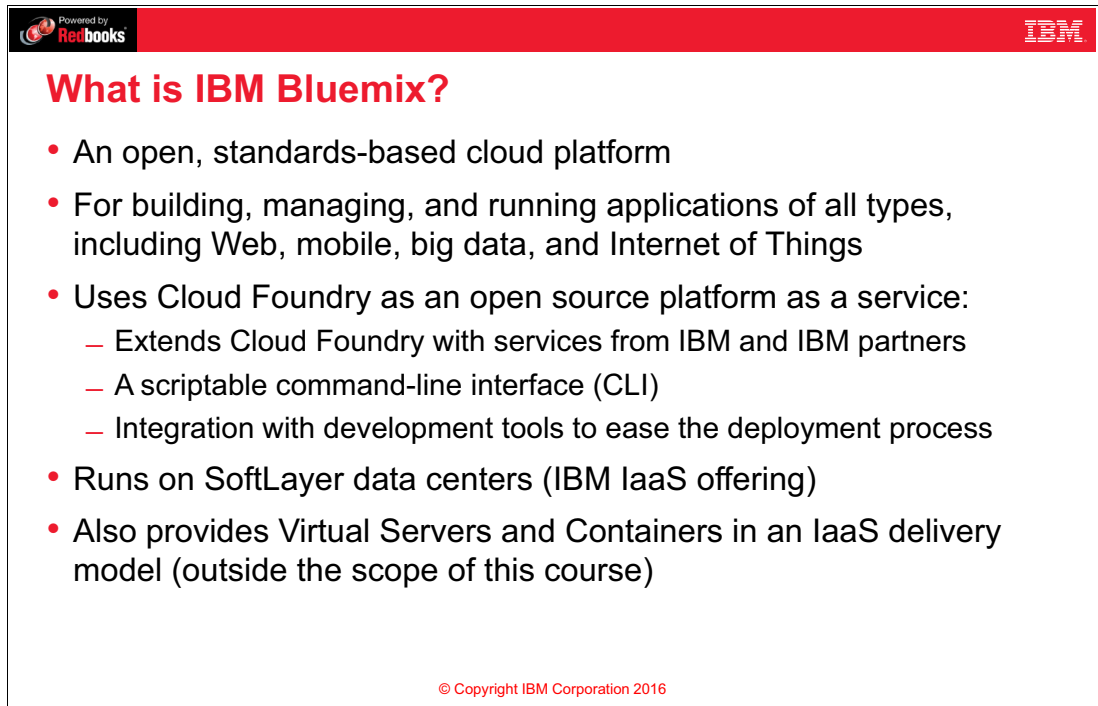
Figure 1-11 Cloud computing: Benefits for developers

Notes:

Cloud computing brings several benefits to developers:

- ▶ Cloud provides readily available sandbox and production environments. These environments offer several capabilities that are attractive for developers:
 - Free trials 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.
- ▶ Cloud brings a wide range of choices to developers in the following areas:
 - Programming languages and frameworks.
 - Services.
 - APIs.
- ▶ Cloud facilitates integrated development, test, and debugging:
 - The new model is to integrate development and operations teams into devops.
 - Build engine for compilation and testing.

1.14 What is IBM Bluemix?



What is IBM Bluemix?

- 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)

© Copyright IBM Corporation 2016

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, as well as 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.

Bluemix is deployed on IBM SoftLayer data centers.

1.15 IBM Bluemix ecosystem

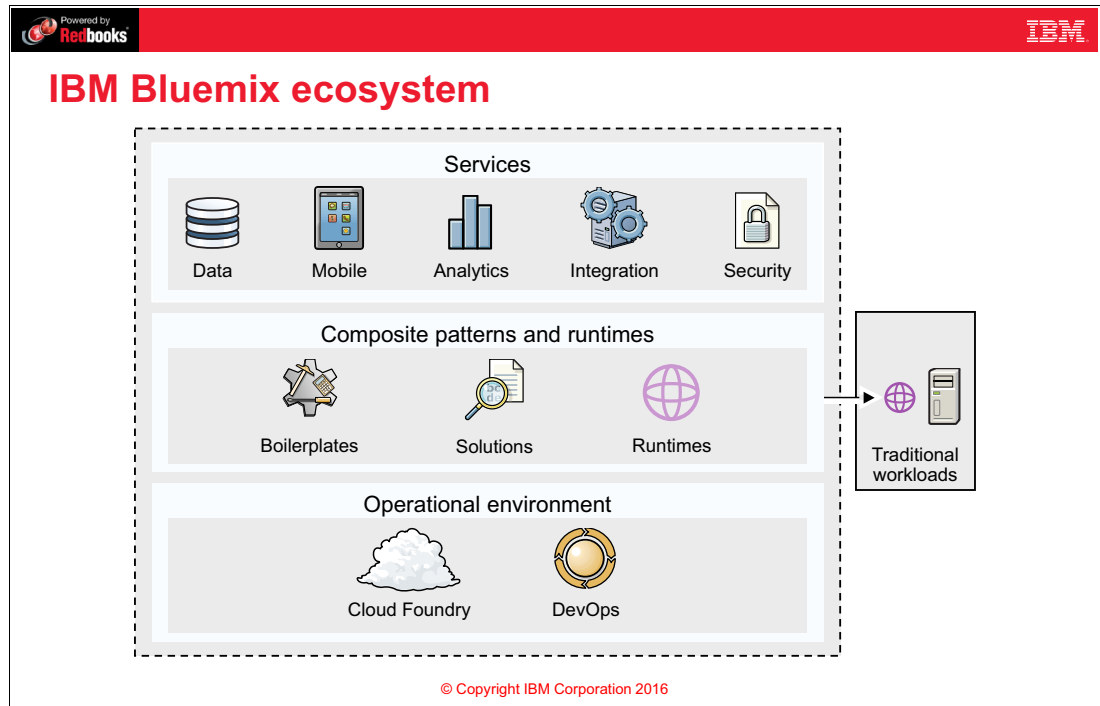


Figure 1-13 IBM Bluemix ecosystem

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 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 app.

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 app and improve the way that you do business.

1.16 Bluemix architectural overview

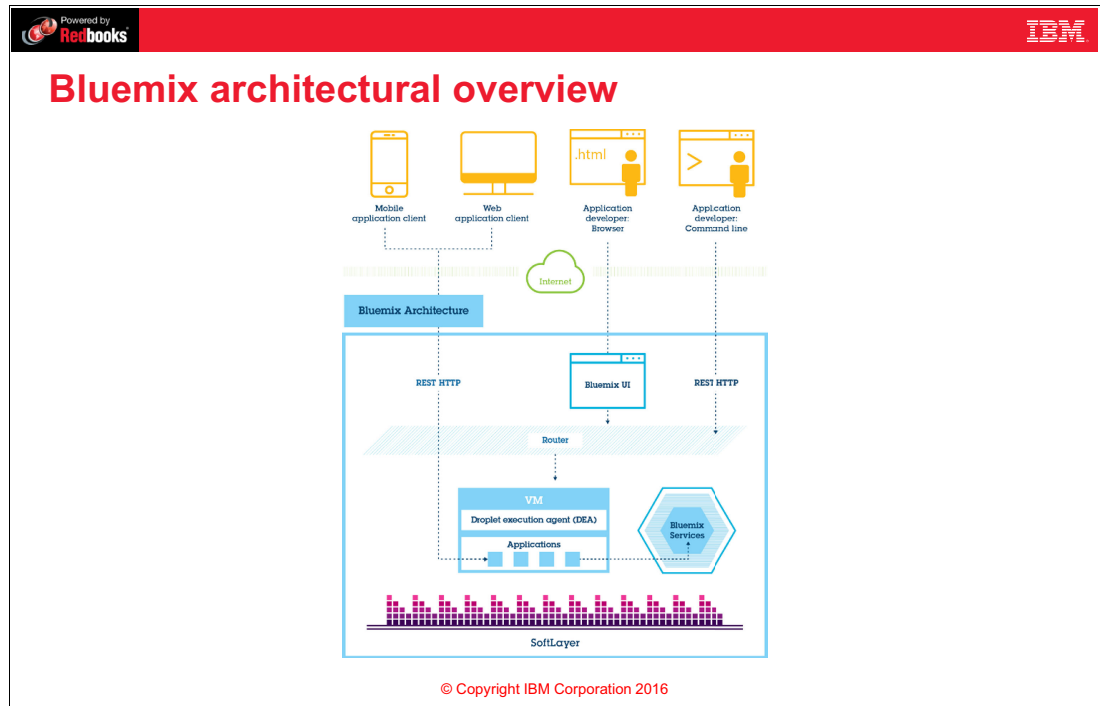


Figure 1-14 Bluemix architectural overview

Notes:

As a developer, you interact with the Bluemix infrastructure using your web browser or the command line tool. Your requests are sent to the Cloud Controller. The cloud controller is responsible for managing the lifecycle of applications.

When you push an application to Cloud Foundry, the Cloud Controller stores the raw application bits, creates a record to track the application metadata, and directs a droplet execution agent (DEA) node to stage and run the application. When you send your app code, Cloud Foundry packages your app with its dependencies and run time to build a *droplet*.

A droplet is an archive within Cloud Foundry that contains an application and its runtime and framework dependencies, prior to deployment to the cloud. It is a package created on the fly that contains everything to run your app in a particular operating system environment.

The Cloud Foundry Router sends incoming traffic to an instance of your application, which is running on a machine called a droplet execution agent (DEA). The DEA manages application instances, tracks started instances, and broadcasts state messages.

Application instances live inside containers, which helps ensure that application instances run in isolation, get their fair share of resources, and are protected from “noisy” neighbors. Clients, either mobile apps or web applications that run externally, can interact with Bluemix-hosted applications.

REST Clients can use APIs and make their requests through Bluemix Routers to one of the application instances. If the app is browser-based, then users use their browser to interact with the app through the routers. Bluemix routers send incoming traffic to the appropriate component in the environment.

For example, routers can send content to the cloud controller for management of applications in their lifecycle, or to a running application on a DEA node. When you create an application and deploy it to Bluemix, the cloud controller determines an appropriate DEA to send the droplet to, and then deploys and starts it there.

Bluemix is *multitenant* but *isolated*, because several applications can run on the same DEA machine at the same time, but not know that the others exist.

1.17 Bluemix: Choice of runtimes

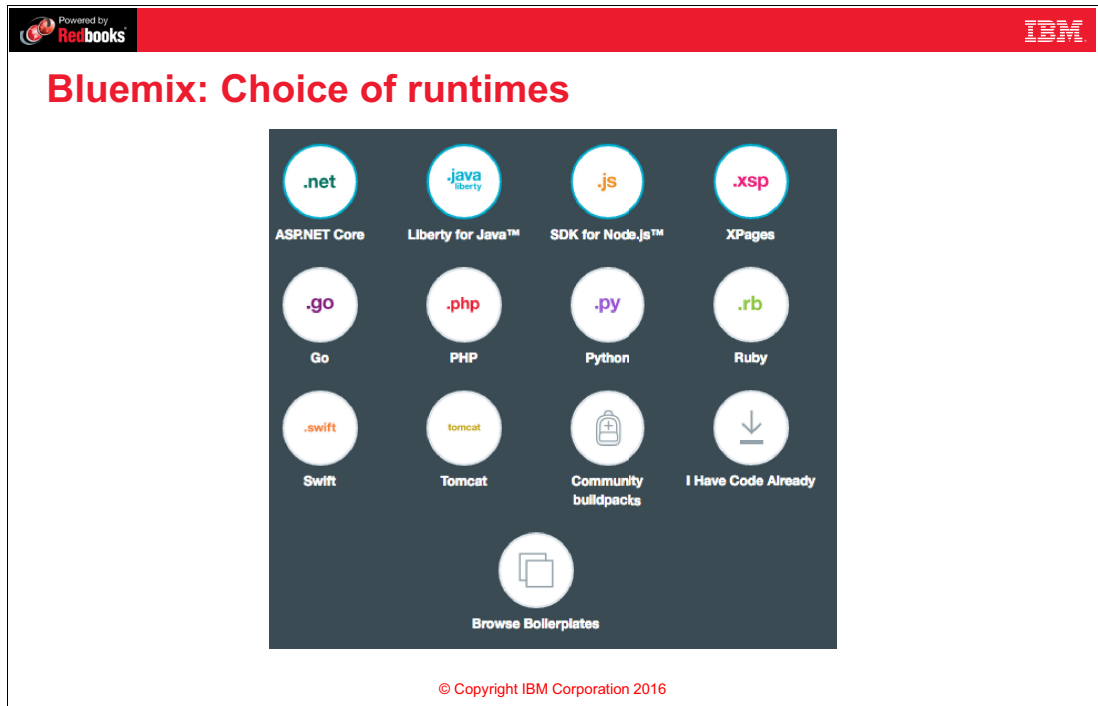


Figure 1-15 Bluemix: Choice of runtimes

Notes:

With 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. There are two runtimes available for Java:

- ▶ Tomcat, which is an open source Java web application server.
- ▶ IBM WebSphere Liberty. With WebSphere Liberty, you can get support from IBM support organizations or from forums such as IBM developerWorks®, which focuses on WebSphere products. Liberty can deploy any Tomcat application, but additionally offers support for many more Java web features, for example MBeans and JMX.

Additional runtimes are available, and are 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 app and any of its dependencies into the droplets mentioned previously.

Note: Bluemix and Cloud Foundry support more runtimes through the *Community Buildpacks*. This open source community has written buildpacks for just about every language you can think of. For more information, see *Using community buildpacks* on the following website:

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

1.18 Bluemix: Services

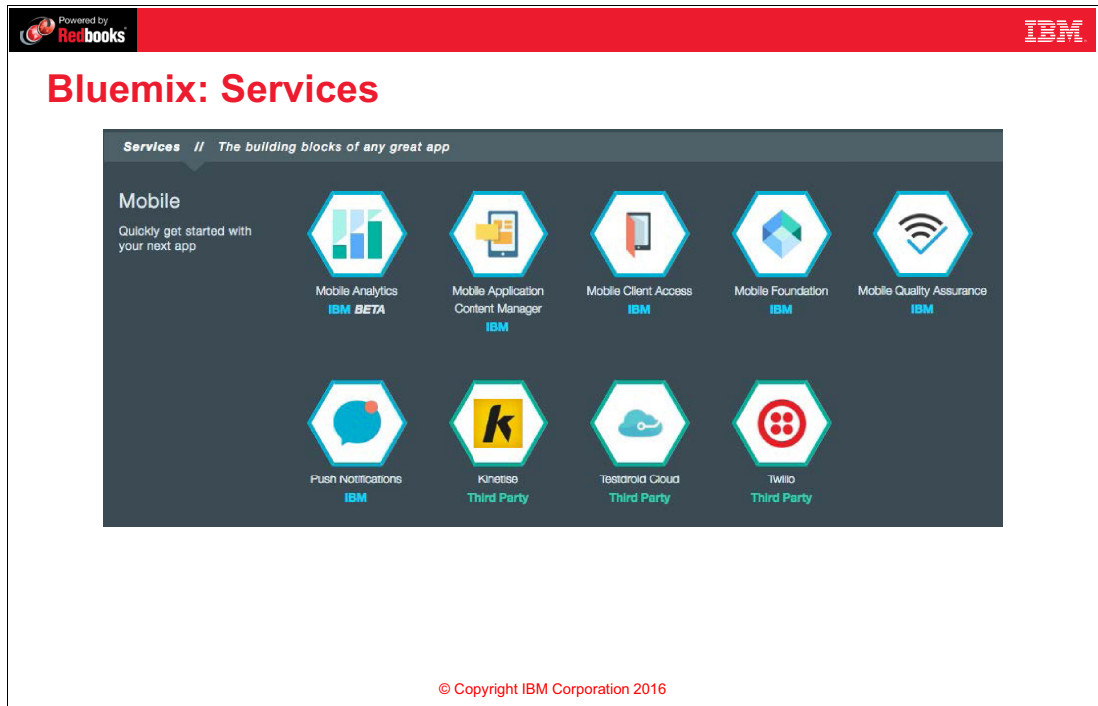




Figure 1-16 Bluemix: Services

Notes:

Bluemix services provide the building blocks for delivering great apps. There are far too many services available in Bluemix to show on one slide. Figure 1-16 is a screen capture of just those services available for Mobile developers. Bluemix provides a broad range of pre-built services that can be used when assembling your app. The following service categories are available:

- ▶ Watson
- ▶ Mobile
- ▶ DevOps
- ▶ Web and Application
- ▶ Network
- ▶ Integration
- ▶ Data and Analytics
- ▶ Security
- ▶ Storage
- ▶ Business Analytics
- ▶ Internet of Things
- ▶ APIs

1.19 Bluemix: Regions

Powered by 

Bluemix: Regions

- A Bluemix region is a geographically defined territory where a cluster of Bluemix hardware, cloud controller, and router are deployed.
- Select the region nearest to your customers to deploy your application
- Applications can provide better performance by being physically closer to the users

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

© Copyright IBM Corporation 2016

Figure 1-17 Bluemix: Regions

Notes:

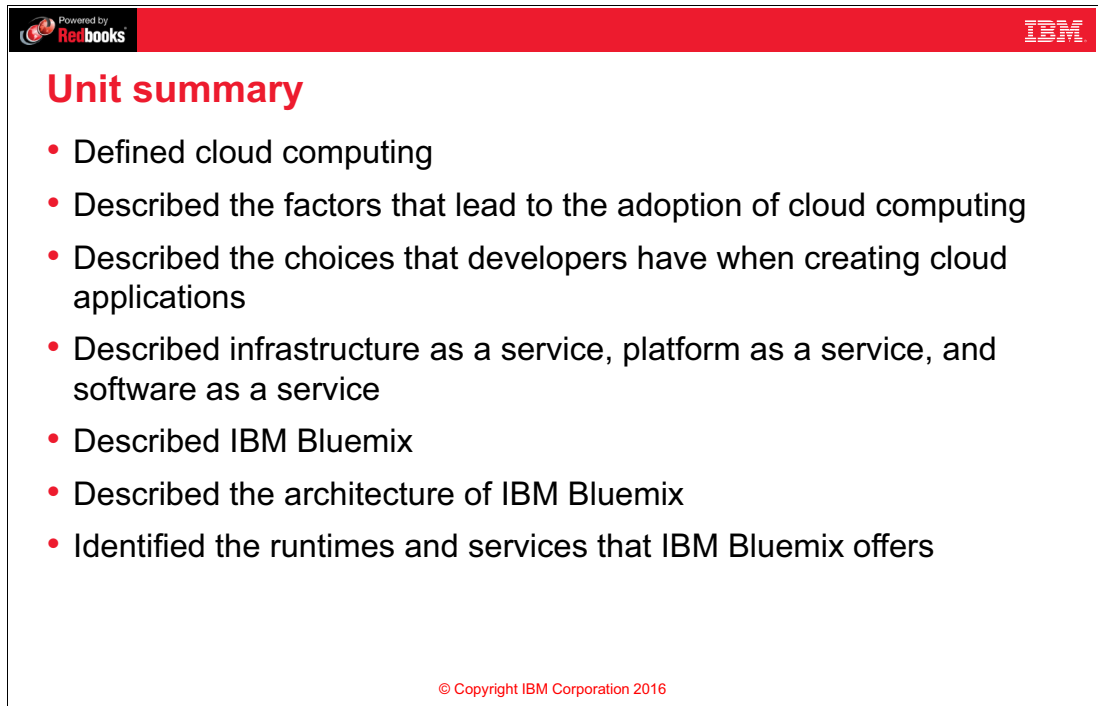
A Bluemix region is a defined geographical territory to which you can deploy your applications. Select the region that is nearest to your users, and deploy your apps to this region to achieve low application latency.

Different countries with different data security requirements might require you to run your Bluemix app in a different region. Running your app in multiple regions also helps make your app highly available. If your app goes down in one region, it will still be available from another region.

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

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

1.20 Unit summary



Powered by
Redbooks

Unit summary

- 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 infrastructure as a service, platform as a service, and software as a service
- Described IBM Bluemix
- Described the architecture of IBM Bluemix
- Identified the runtimes and services that IBM Bluemix offers

© Copyright IBM Corporation 2016

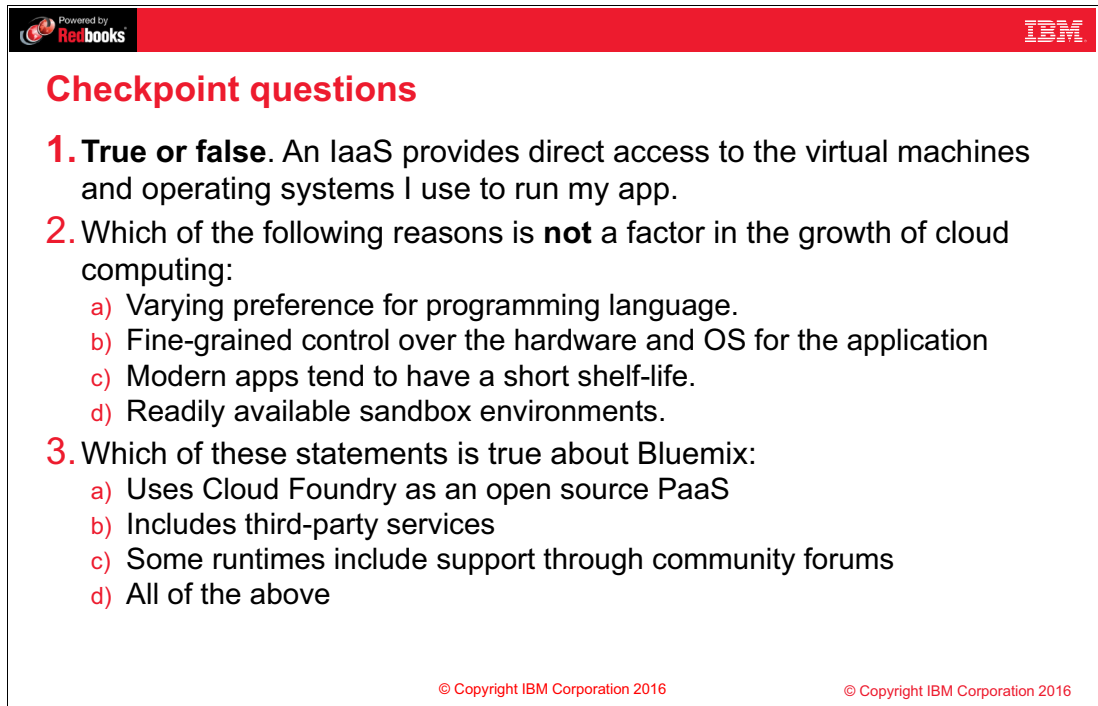
IBM

Figure 1-18 Unit summary

Notes:

In this unit, we have 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 infrastructure as a service, platform as a service, and software as a service; described IBM Bluemix; described the architecture of IBM Bluemix; and identified the runtimes and services that Bluemix offers.

1.21 Checkpoint questions



Powered by **Redbooks** **IBM**

Checkpoint questions

- 1. True or false.** An IaaS provides direct access to the virtual machines and operating systems I use to run my app.
- 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 apps 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

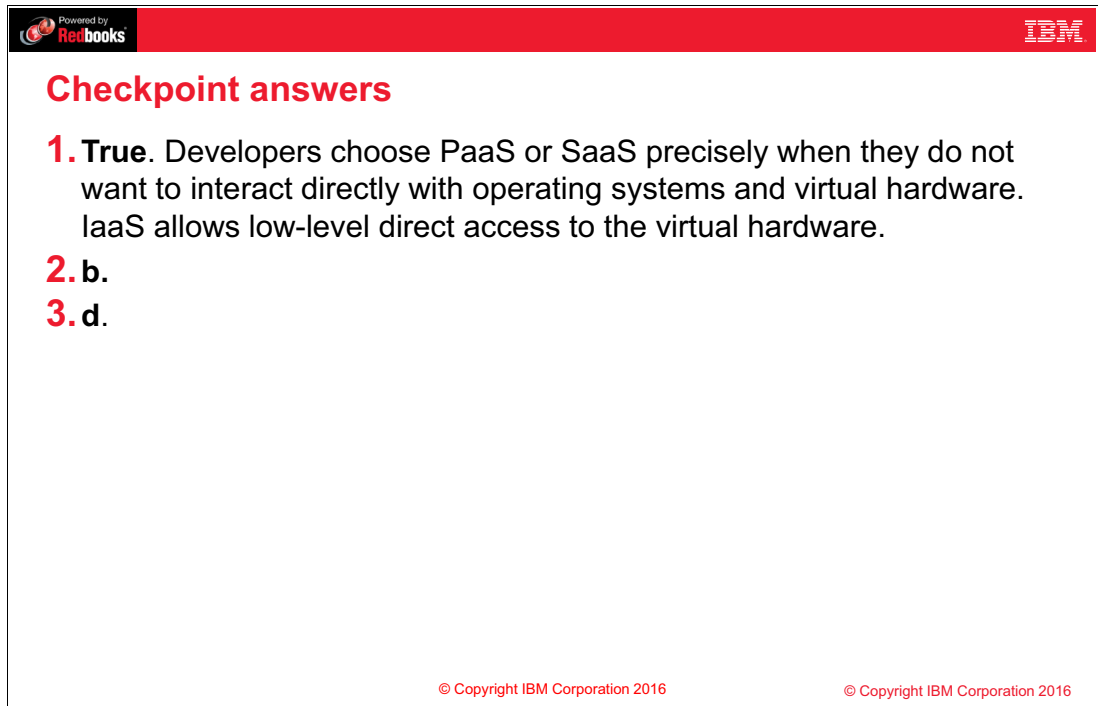
© Copyright IBM Corporation 2016 © Copyright IBM Corporation 2016

Figure 1-19 Checkpoint questions

Notes:

(none)

1.22 Checkpoint answers



The image is a screenshot of a document page. At the top left, there is a logo for 'Powered by Redbooks'. At the top right, there is the 'IBM' logo. The main content area has a red header with the text 'Checkpoint answers' in white. Below this, there are three numbered items:

- 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.**

At the bottom of the page, there are two copyright notices: '© Copyright IBM Corporation 2016' on the left and '© Copyright IBM Corporation 2016' on the right.

Figure 1-20 Checkpoint answers

Notes:

(none)



Getting started with IBM Bluemix

This chapter describes the following considerations for getting started with IBM Bluemix:

- ▶ Understanding what you can build in IBM Bluemix
- ▶ Creating and managing a Bluemix application
- ▶ 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
- ▶ Describe what you can build in Bluemix
- ▶ Describe how to create an application in Bluemix
- ▶ Describe the Bluemix dashboard, catalog, and documentation features
- ▶ Describe how the application route is used to test an application from the browser
- ▶ Describe how to create services in Bluemix
- ▶ Describe how to bind services to an application in Bluemix
- ▶ Describe the environmental variables that are used with Bluemix services
- ▶ Describe Bluemix organizations, domains, spaces, and users

2.2 References

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

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

2.3 Getting started: Create an IBM Bluemix account

Powered by **Redbooks** **IBM**

Getting started: Create an IBM Bluemix account

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 will automatically create an IBM ID for you.
 - There is no charge to create an IBM ID, and there is no charge to create a Bluemix trial account.

Sign up for IBM Bluemix

Your 90-day trial is free, with no credit card required. You get access to 2 GB of runtime and container memory to run apps, unlimited IBM services and APIs, and complimentary support.

Already signed up for Bluemix? [Log In](#)

Email Address*

First Name*

Last Name*

Company

Select your country or region.
UNITED STATES



© Copyright IBM Corporation 2016

Figure 2-1 Getting started: Create 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 do not need to specify a method of payment to create an account. You can register for a free IBM Bluemix trial account at bluemix.net.

2.4 IBM Bluemix infrastructure types

Powered by  **Redbooks** 

IBM Bluemix infrastructure types

- **Cloud Foundry** provides the platform as a service (PaaS) environment to run your applications:
 - IBM Bluemix manages and maintains the infrastructure that runs the applications.
- **IBM Containers** provide more fine-grained control over the computing infrastructure, to run the application and services through Docker-like APIs:
 - You can use and extend public images from the IBM Bluemix catalog, or images from the public Docker hub.
- **Virtual Servers** are a software implementation of hardware that runs applications like a computer:
 - You can configure the operating system, server runtime environment, and application.



© Copyright IBM Corporation 2016

Figure 2-2 IBM Bluemix infrastructure types

Notes:

IBM Bluemix supports three 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*, Bluemix allows basic management capability of your app and allows it to connect with prepackaged services. Cloud Foundry boilerplates and starter apps are the simplest way to get started with Bluemix.
- ▶ *IBM Containers* provide more fine-grained control over the computing infrastructure to run the application and services through Docker-like APIs. You can use and extend public images from the IBM Bluemix catalog or images from the public Docker hub.

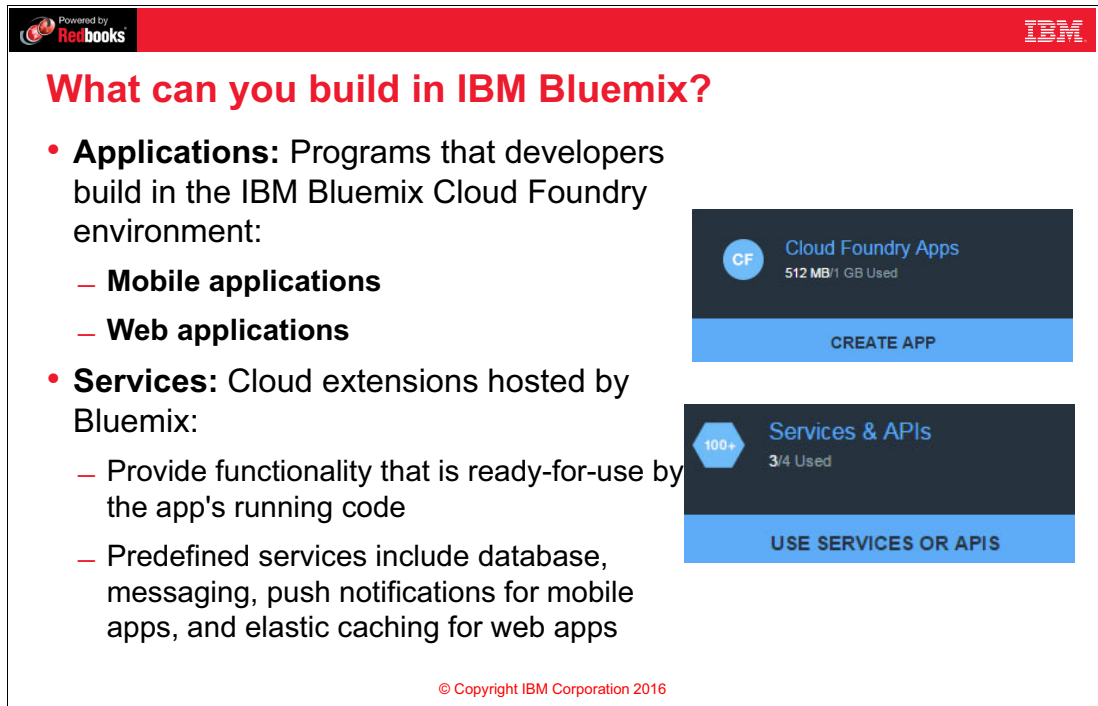
If you want to port an image from another cloud infrastructure provider, or if you want to use an existing public image, consider the *IBM Containers* technology. Containers give you a bit more control, but still do not require you to manage an operating system or network. IBM Containers resemble the popular open source container engine Docker, but have some Bluemix-specific features such as their ability to integrate with the Cloud Foundry routers.
- ▶ *Virtual Servers* are software implementation of hardware that runs applications like 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 the three 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 will be 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 interface. At the top, there is a red header with the 'Powered by Redbooks' logo on the left and the 'IBM' logo on the right. Below the header, the main title 'What can you build in IBM Bluemix?' is displayed in red. The content is organized into two main sections:

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

On the right side of the screenshot, there are two interactive buttons:

- The top button is labeled 'Cloud Foundry Apps' with a 'CF' icon and '512 MB/1 GB Used' below it. Below this button is a blue bar with the text 'CREATE APP'.
- The bottom button is labeled 'Services & APIs' with a '100+' icon and '3/4 Used' below it. Below this button is a blue bar with the text 'USE SERVICES OR APIS'.

At the bottom right of the screenshot, there is a small red copyright notice: '© Copyright IBM Corporation 2016'.

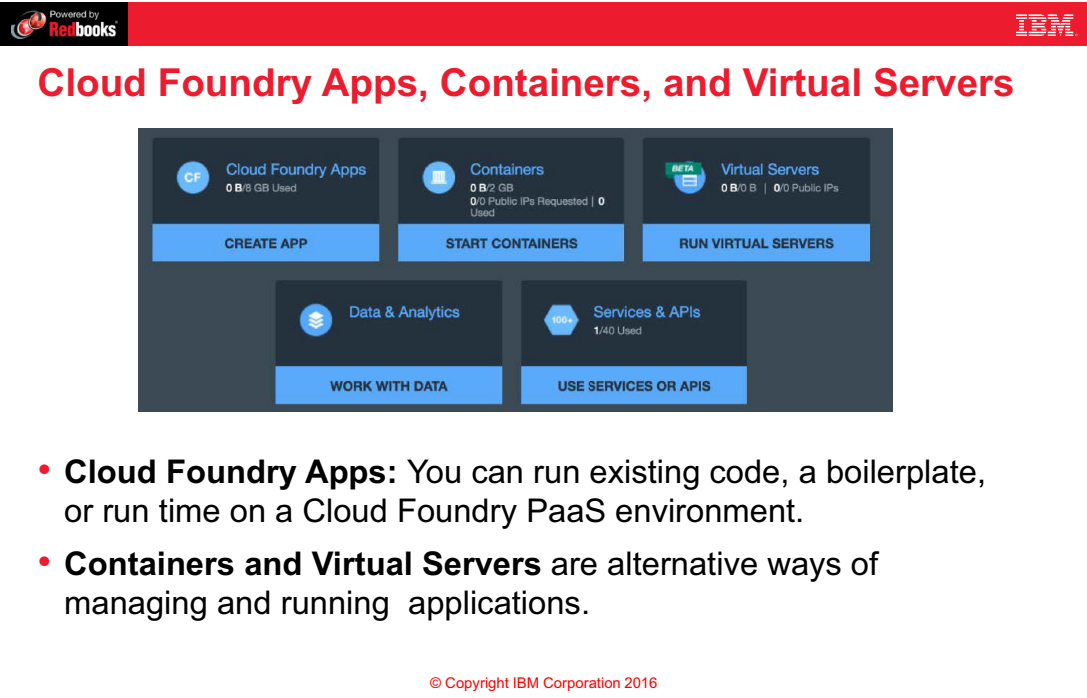
Figure 2-3 What can you build in IBM Bluemix?

Notes:

In 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 Bluemix environment and use services that the mobile apps are exposed to. 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 run time.

A *service* is a cloud extension that is hosted by Bluemix. The service provides functionality that is ready-for-use by the app's running code. The predefined services that are provided by Bluemix include database, messaging, push notifications for mobile apps, and elastic caching for web apps. You can create your own services in Bluemix. They can be simple utilities, such as the functions you might see in a runtime library, or they can be 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 screenshot shows the IBM Bluemix console dashboard. At the top, there is a red header with the 'Powered by Redbooks' logo on the left and the 'IBM' logo on the right. Below the header, the main title 'Cloud Foundry Apps, Containers, and Virtual Servers' is displayed in red. The dashboard features five main tiles, each with a blue button:

- Cloud Foundry Apps:** Shows '0 B/6 GB Used' and a 'CREATE APP' button.
- Containers:** Shows '0 B/2 GB' and '0/0 Public IPs Requested | 0 Used', with a 'START CONTAINERS' button.
- Virtual Servers:** Shows '0 B/0 B | 0/0 Public IPs' and a 'RUN VIRTUAL SERVERS' button.
- Data & Analytics:** Shows a 'WORK WITH DATA' button.
- Services & APIs:** Shows '1/40 Used' and a 'USE SERVICES OR APIS' button.

At the bottom of the dashboard, there is a copyright notice: '© Copyright IBM Corporation 2016'.

- **Cloud Foundry Apps:** You can run existing code, a boilerplate, or run time on a Cloud Foundry PaaS environment.
- **Containers and Virtual Servers** are alternative ways of managing and running applications.

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

Notes:

Cloud Foundry Apps use the Bluemix architecture. That is, these apps are composed into droplets, run on a droplet execution agent (DEA), and are accessed through a Cloud Foundry router.

Containers and *Virtual Servers* are alternative ways to run and manage an app. 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 vs. Virtual Servers

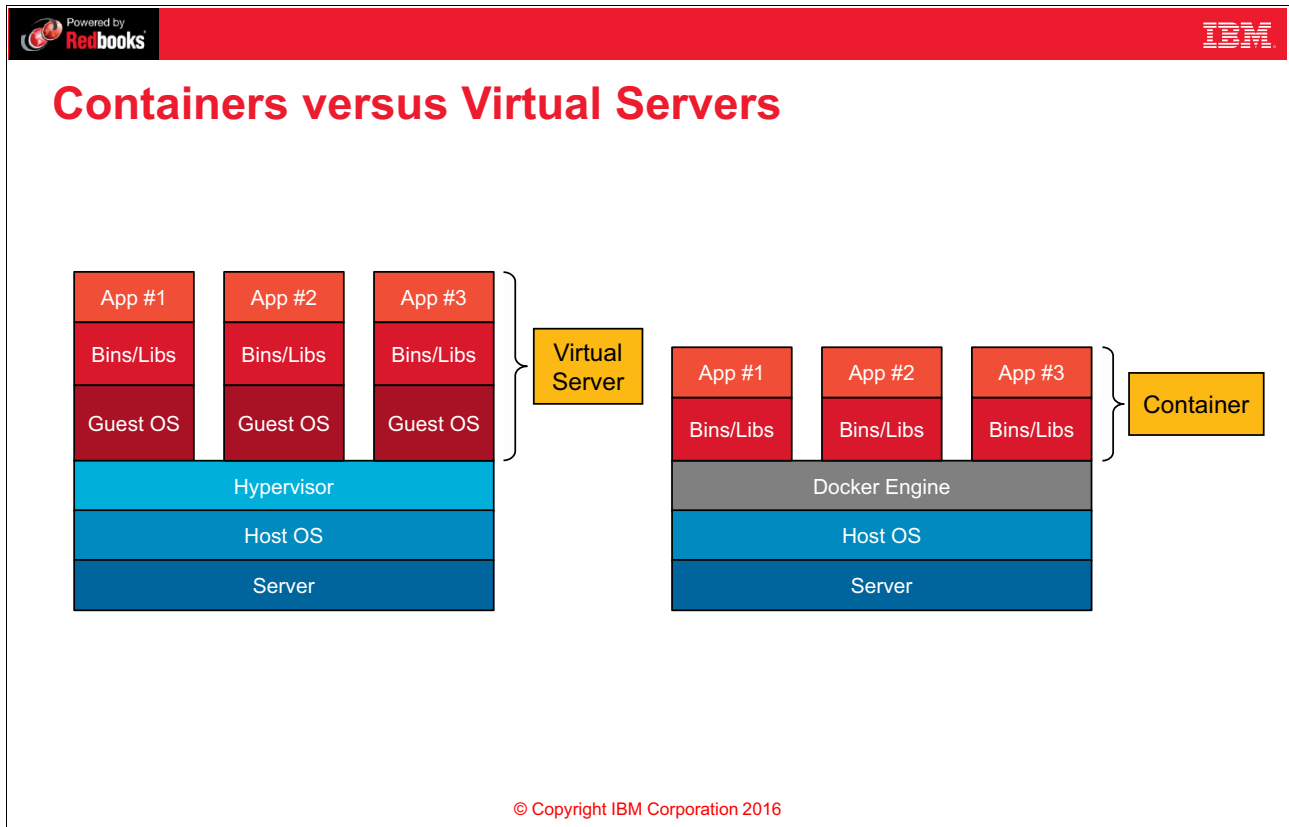


Figure 2-5 Containers versus Virtual Servers

Notes:

Containers and 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, allowing for more efficient use of computing resources.

As covered in Unit 1, “Introduction to cloud computing and IBM Bluemix” on page 1, 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 virtual machine does not have any awareness that it is running in a virtual machine. 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 which provides hardware virtualization, a container provides operating-system-level virtualization.

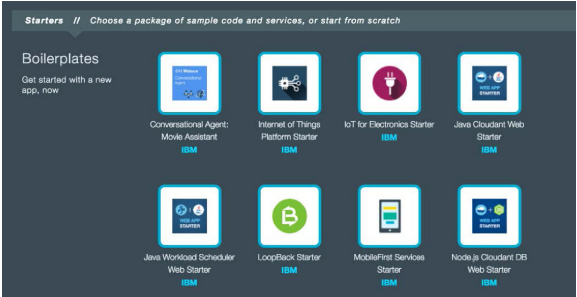
A container library, such as Docker, separates different user spaces for each container. On the surface, these spaces may look exactly like a virtual machine to a user. The main difference between containers and virtual machines is that containers share the operating system kernel with other containers running on the machine. In virtual machines, this is not the case. This is why you may see a virtual machine running Linux on a Windows server, or Windows on a Linux server. But you will 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 is managed differently in

containers as well. In virtual machines, virtual disks are created and are similar to physical disks in that once 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 Bluemix catalog: Boilerplates

Powered by **Redbooks** **IBM**

Bluemix catalog: Boilerplates



The screenshot shows a 'Starters' section with the heading 'Boilerplates' and the subtext 'Get started with a new app, now'. Below this, there are eight boilerplate options arranged in two rows of four. Each option consists of a small icon, a title, and the IBM logo. The options are: 1. Conversational Agent: Movie Assistant; 2. Internet of Things Platform Starter; 3. IoT for Electronics Starter; 4. Java Cloudant Web Starter; 5. Java Workload Scheduler Web Starter; 6. LoopBack Starter; 7. MobileFirst Services Starter; 8. Node.js Cloudant DB Web Starter.

- Boilerplates are packages of *templates* and *sample applications*.
 - When you create an app 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.

© Copyright IBM Corporation 2016

Figure 2-6 Bluemix catalog: Boilerplates

Notes:

In IBM Bluemix, Boilerplates are packages of templates and sample applications. When you create an app with boilerplate, Bluemix configures the services for the application. 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.

2.9 Bluemix catalog: Runtimes

Powered by **Redbooks** **IBM**

IBM Bluemix catalog: Run times

- To create your own application, select the server runtime environment as a starting point:
 - IBM packages support **IBM** runtime environments, such as Liberty.
 - **Community** run times rely on open source and third-party packages.

© Copyright IBM Corporation 2016

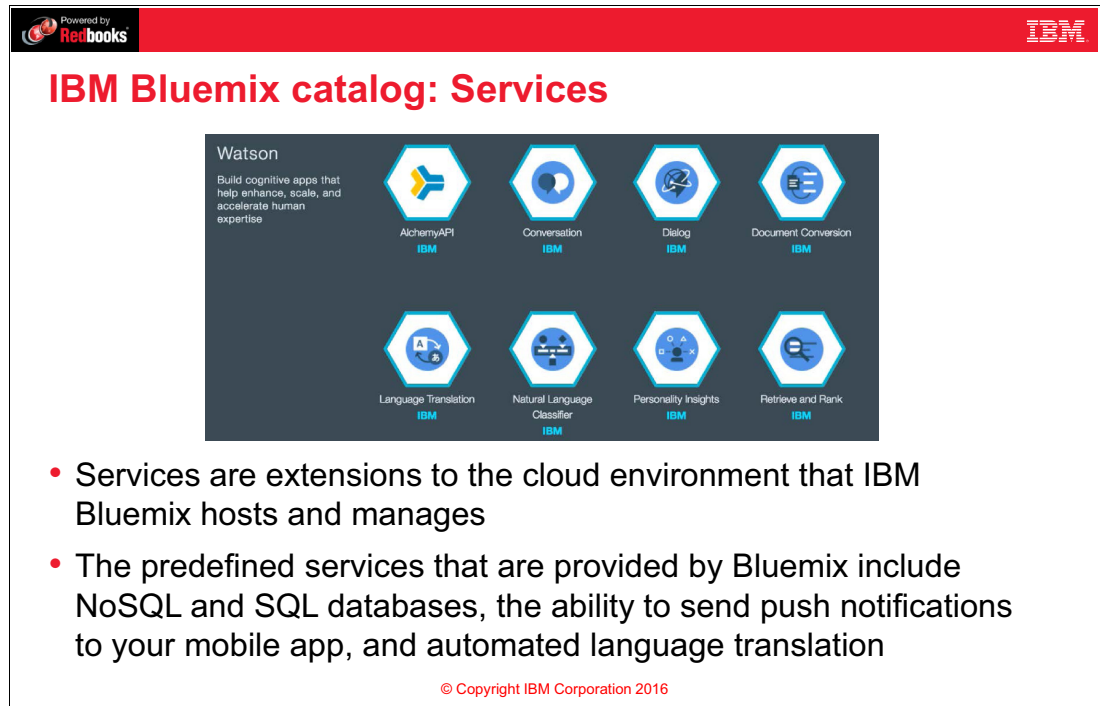
Figure 2-7 Bluemix catalog: Runtimes

Notes:

A *runtime* is the set of resources that is used to run an app. Bluemix provides runtime environments as containers for different types of apps. The runtime environments are integrated as buildpacks into 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 come with any services by default, but you can add and bind your own services later.

2.10 Bluemix catalog: Services



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

Figure 2-8 Bluemix catalog: Services

Notes:

Services are extensions to the cloud environment that IBM Bluemix hosts and manages. The predefined services that are provided by 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 Bluemix application from the 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 completely managed by Bluemix, and your app needs to only focus on the provided endpoint. You can bind more than one app to a service to share services between your apps.

There are many Bluemix services and more are being added. Figure 2-8 shows a sample of the services available for IBM Watson.

2.11 Create a Bluemix application (1 of 3)

Creating an IBM Bluemix application (1 of 3)

1. To create an IBM Bluemix application, click **Create an app** from the dashboard.
2. Click the application type: **Web** application, or **Mobile** application.
3. Choose a **run time** for your application.

© Copyright IBM Corporation 2016

Figure 2-9 Create a Bluemix application (1 of 3)

Notes:

To create a Bluemix application perform the following steps:

1. Click **Create an app**
2. Select the type of app, either web or mobile application
3. Select a runtime from a wide array of available runtimes in Bluemix.

You can also indicate that you have code already, or use community buildpacks that provide even more runtimes than you see in the Bluemix catalog.

2.12 Create a Bluemix application (2 of 3)

The screenshot shows a web interface for creating an IBM Bluemix application. At the top, there is a red header with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "Create an IBM Bluemix application (2 of 3)" is displayed in red. The main content area contains step 4: "4. For more options, click **View details in catalog**. Otherwise, click **Continue**." Below this text is a card for the "SDK for Node.js™". The card contains the following text: "Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability." To the right of this text is a green "CONTINUE" button. Below the text, it says "375 GB-hours free each month" and provides two links: "View Docs" and "View Details in Catalog". Step 5 follows: "5. Enter the **Name** for your app." Below this is a dark gray form titled "What do you want to name your new app?". The form has a text input field labeled "APP NAME" and a "FINISH" button. At the bottom right of the form area, there is a small copyright notice: "© Copyright IBM Corporation 2016".

Figure 2-10 Create a Bluemix application (2 of 3)

Notes:

From here, you can click **View Details in Catalog** to set your own domain name or further customize your app.

You can just click **Continue**, and you will be prompted for your application name. If you do not specify your own domain, your app will get the `mybluemix.net` domain, which is provided to all public user applications in Bluemix.

2.13 Bluemix application name must be unique across domain

The IBM Bluemix app name must be unique across domain

- By default, 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 Bluemix users:
 - You cannot create a Bluemix application with the same **host** name already in used by another application.

What do you want to name your new app?

APP NAME

vy301-node-example

Someone already used that name. Try entering a different name.

FINISH

© Copyright IBM Corporation 2016

Figure 2-11 Bluemix application name must be unique across domain

Notes:

By default, 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 Bluemix users. You cannot create a Bluemix application with the same host name already in used by another application. The exercises in this course handle this issue by having each team use a randomly-generated key.

Your app's name is different than its host name and only needs to be unique within your organization. You can have the same name as another user, as long as you do not have the same host name as another user. By default, Bluemix sets your hostname and your app name to be the same. The app name is for your reference only to use in scripts, commands to the CLI, and to find your app within the Bluemix UI. The application's host name will become part of the application's route, which is how users access your app over the internet.

2.14 Create a Bluemix application (3 of 3)

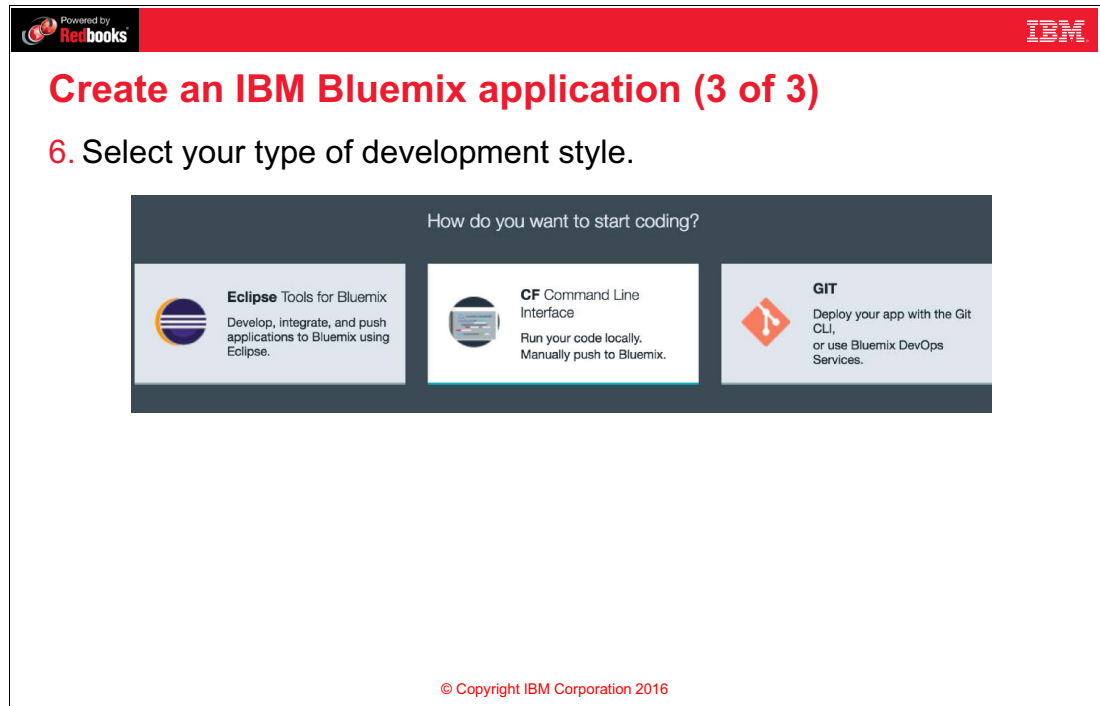


Figure 2-12 Create a Bluemix application (3 of 3)

Notes:

Next, select your development style. You can:

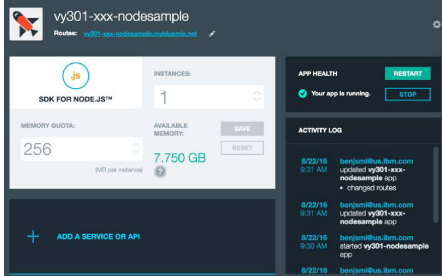
- ▶ Download the Eclipse Tools for Bluemix, which allow you to use the Bluemix APIs from the Eclipse integrated development environment.
- ▶ Edit your app in your favorite text editor and use the Cloud Foundry command line to deploy.
- ▶ Use the git command line client and Bluemix DevOps Services to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

2.15 Bluemix application overview

Powered by **Redbooks** **IBM**

Bluemix application overview

- Check and control the status of applications in the **Overview** page:
 - View and adjust the memory resources that are used by the application
 - Start and stop your application
 - View your application in the browser
 - Manage your application with Bluemix DevOps, and add the source code to a Git repository



© Copyright IBM Corporation 2016

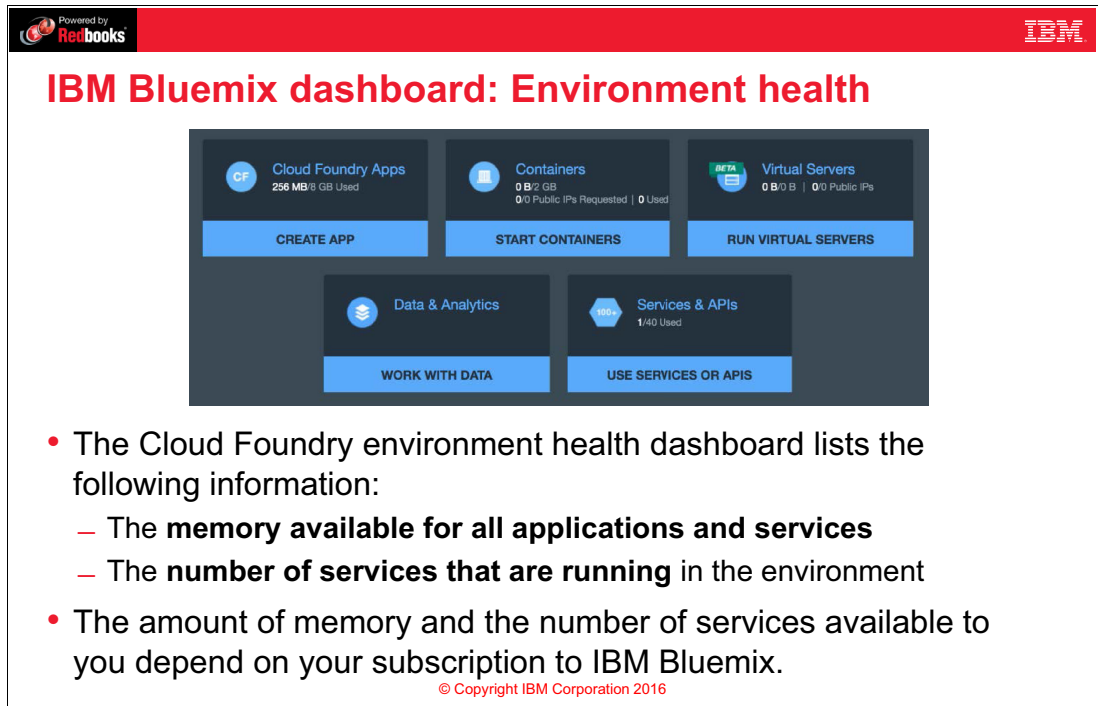
Figure 2-13 Bluemix application overview

Notes:

From the Bluemix application overview page you can:

- ▶ View and adjust the memory resources that are used by the application
- ▶ Start and stop your application
- ▶ Navigate to your application as a user
- ▶ Manage your application with Bluemix DevOps Services and add the source code of your app to a git repository

2.16 Bluemix dashboard: Environment health



The screenshot displays the IBM Bluemix dashboard's 'Environment health' section. At the top, there is a red header with 'Powered by Redbooks' on the left and the 'IBM' logo on the right. Below the header, the title 'IBM Bluemix dashboard: Environment health' is centered in red. The main content area features five dark grey cards with blue accents. The first row contains three cards: 'Cloud Foundry Apps' (256 MB / 8 GB Used, 'CREATE APP' button), 'Containers' (0 B / 2 GB, 0 / 0 Public IPs Requested | 0 Used, 'START CONTAINERS' button), and 'Virtual Servers' (0 B / 0 B, 0 / 0 Public IPs, 'RUN VIRTUAL SERVERS' button). The second row contains two cards: 'Data & Analytics' ('WORK WITH DATA' button) and 'Services & APIs' (1 / 40 Used, 'USE SERVICES OR APIS' button). A small copyright notice '© Copyright IBM Corporation 2016' is located at the bottom right of the dashboard area.

- The Cloud Foundry environment health dashboard lists the following information:
 - The **memory available for all applications and services**
 - The **number of services that are running** in the environment
- The amount of memory and the number of services available to you depend on your subscription to IBM Bluemix.

Figure 2-14 Bluemix dashboard: Environment health

Notes:

The Cloud Foundry environment health view in the IBM Bluemix dashboard shows you the amount of memory you are using in your space and the number of services you are using in your space.

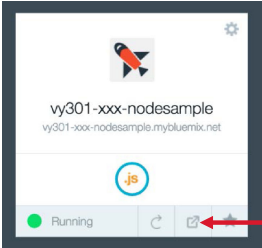
If you find that your organization is running low on memory, you can always stop an application without deleting it and start it again later when needed. If you are running out of available services, however, you must delete some to free up space.

The health view provides similar overviews for virtual servers and containers, but these are outside the scope of this course.

2.17 Testing applications through the application route

Powered by **Redbooks** **IBM**

Testing applications through the application route



- The *application route* is the entry point for end-users into your application in 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.

© Copyright IBM Corporation 2016

Figure 2-15 Testing applications through the application route

Notes:

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

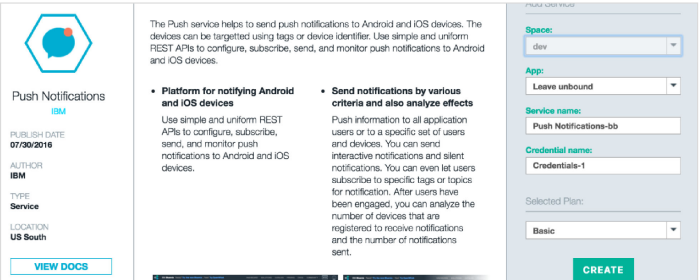
To view the application route, click the Open URL icon. You can access the application route for a running application only. The application route must be unique across all applications in the domain

2.18 Add a Bluemix service

Powered by **Redbooks** **IBM**

Add a Bluemix service

1. On the application overview page, click **ADD A SERVICE OR API**.
2. Select a service from the **Bluemix catalog**.
3. Provide a **Service name**, **Space**, and a **Selected Plan** for the service:
 - You can choose to bind the service to an application, or leave it unbound.
4. You can also use multiple credential sets for different services.



© Copyright IBM Corporation 2016

Figure 2-16 Add a Bluemix service

Notes:

You can add a Bluemix service to your app by clicking **Use Services or APIs** and then selecting a service from the **Bluemix Catalog**. You must then provide a name for the service and the space you want it to run in, and select a 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 whether you share hardware or use dedicated hardware.

You can bind the service to your app from this page, or just leave the service unbound. If you leave the service unbound, Bluemix will create and provide credentials for accessing the service, which you can access from within the service dashboard. If you bind the service to an app, the credentials will show up in the application’s environment variables.

The **Credential Name** field is used for creating multiple credentials to the same service in case you need further control over who can use the service. You can leave this field at its default value.

2.19 Bind a service to an application


Powered by **Redbooks** **IBM**

Bind a service to an application

1. From the application overview page, select **BIND A SERVICE OR API**.
2. Select the service to bind to your application:
 - IBM Bluemix lists the services that are compatible with your application and runtime environment only.
 - You can bind a service to more than one application.

Add Service to 'vy301-xxx-nodesample' Application

Select the previously used service instance that you want to add to your application.

	NAME	SERVICE	VERSION
<input type="radio"/>	 Cloudant NoSQL DB...	cloudantNoSQLDB	

© Copyright IBM Corporation 2016

Figure 2-17 Bind a service to an application

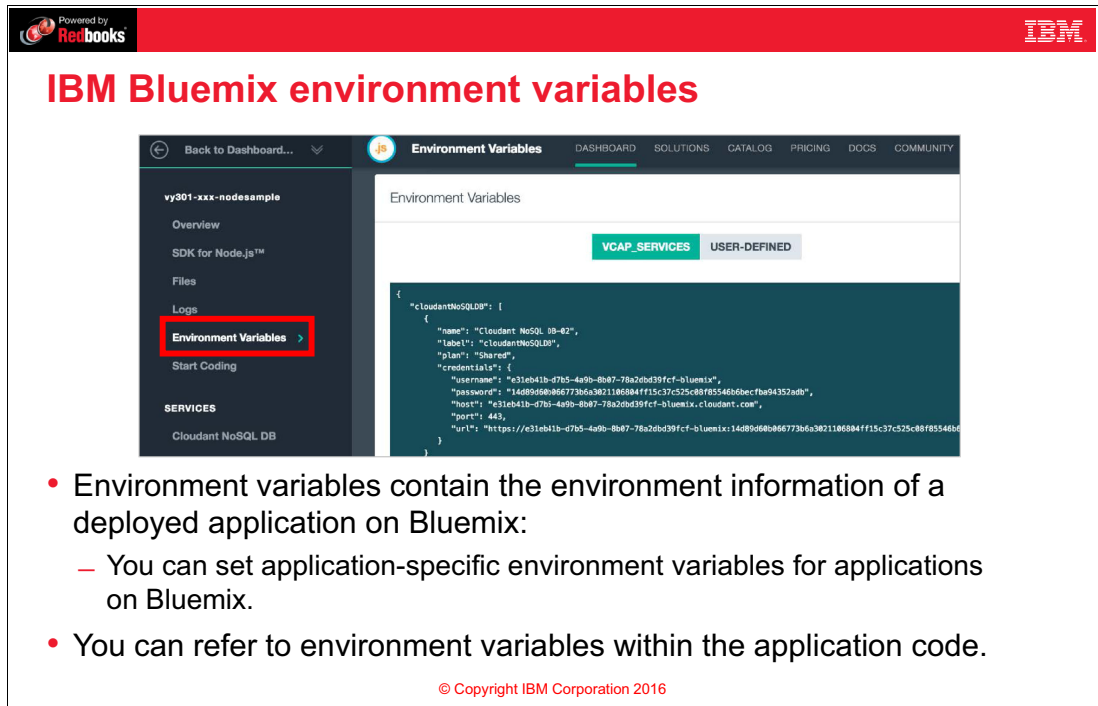
Notes:

To bind the service to your application, go to your application overview page and select **BIND A SERVICE OR API**. Bluemix will list the services that you have 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 add a service, Bluemix creates a new or additional instance of that service in your organization.

When you bind a service to an app, Bluemix creates a set of credentials to access that service and places the connection information for this service in the app's environment variables.

2.20 Bluemix environment variables



Environment Variables

```
{
  "cloudantNoSQLDB": [
    {
      "name": "Cloudant NoSQL DB",
      "label": "cloudantNoSQLDB",
      "plan": "Shared",
      "credentials": {
        "username": "e31eb41b-d7b5-4a9b-8b67-78a2bd39fcf-bluemix",
        "password": "14d89d98b66773b6a3821186884ff15c37c525c8f85546b6ecfba94352adb",
        "host": "e31eb41b-d7b5-4a9b-8b67-78a2bd39fcf-bluemix.cloudant.com",
        "port": 443,
        "url": "https://e31eb41b-d7b5-4a9b-8b67-78a2bd39fcf-bluemix:14d89d98b66773b6a3821186884ff15c37c525c8f85546b6ecfba94352adb@e31eb41b-d7b5-4a9b-8b67-78a2bd39fcf-bluemix.cloudant.com"
      }
    }
  ]
}
```

- Environment variables contain the environment information of a deployed application on Bluemix:
 - You can set application-specific environment variables for applications on Bluemix.
- You can refer to environment variables within the application code.

© Copyright IBM Corporation 2016

Figure 2-18 Bluemix environment variables

Notes:

Environment variables contain the environment information of a deployed application on Bluemix. Bluemix automatically populates the environment variable **VCAP_SERVICES** with the services that you bound to your 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.

2.21 Bluemix organizations and users (team members)



Powered by Redbooks

IBM

Bluemix organizations and users (team members)

- **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 (this is how you share access to control and monitor your apps and services!)

© Copyright IBM Corporation 2016

Figure 2-19 Bluemix organizations and users (team members)

Notes:

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

Organizations are defined by the following items:

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

In 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.

2.22 Bluemix domains and quota

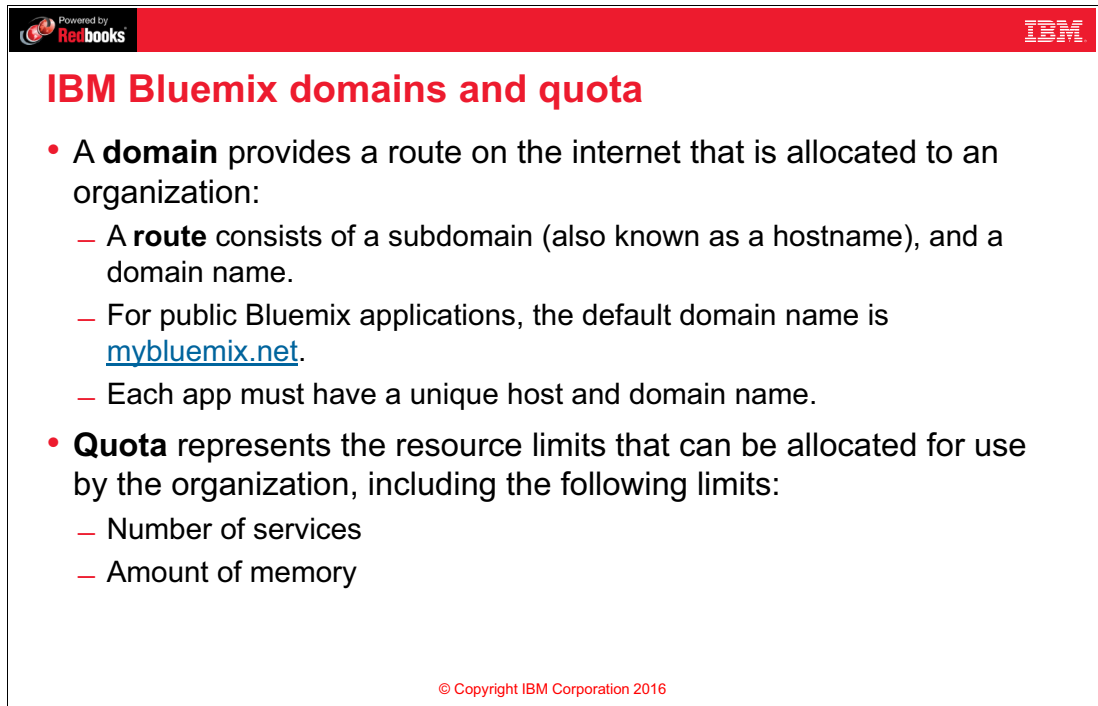
The image is a screenshot of a Redbooks document. At the top left, there is a logo for 'Powered by Redbooks'. At the top right, there is the IBM logo. The main title of the document is 'IBM Bluemix domains and quota' in red text. Below the title, there are two main bullet points. The first bullet point is 'A domain provides a route on the internet that is allocated to an organization:', followed by three sub-bullets: 'A route consists of a subdomain (also known as a hostname), and a domain name.', 'For public Bluemix applications, the default domain name is mybluemix.net.', and 'Each app must have a unique host and domain name.'. The second main bullet point is 'Quota represents the resource limits that can be allocated for use by the organization, including the following limits:', followed by two sub-bullets: 'Number of services' and 'Amount of memory'. At the bottom right of the document, there is a copyright notice: '© Copyright IBM Corporation 2016'.

Figure 2-20 Bluemix domains and quota

Notes:

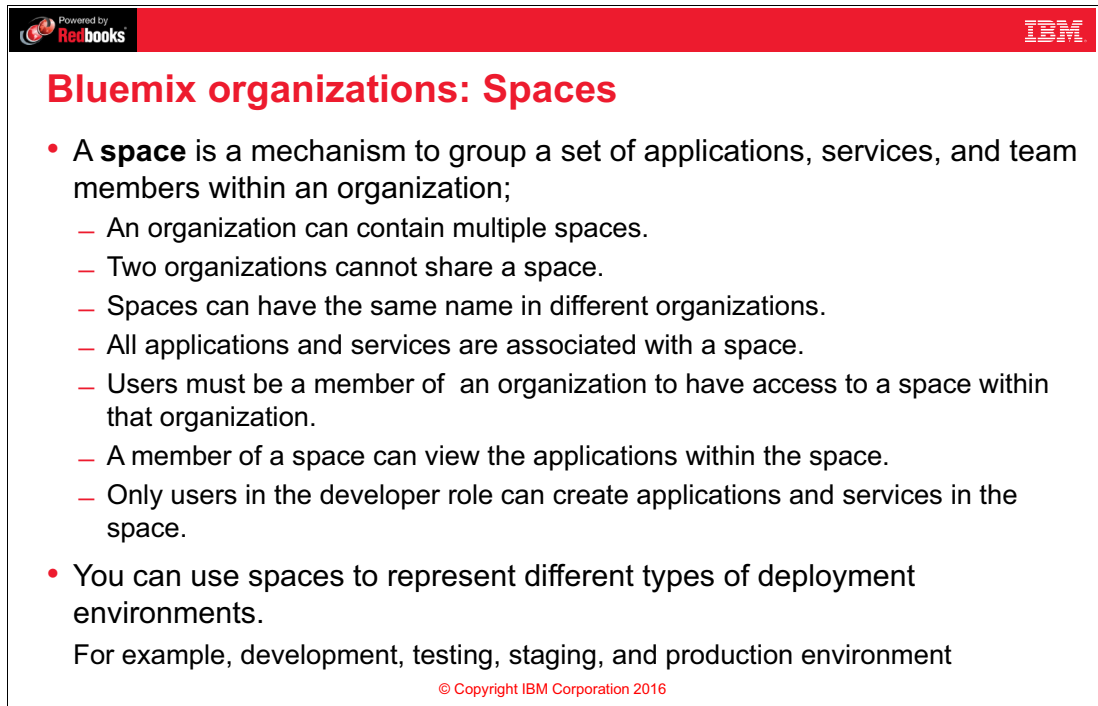
Domains provide the route on the internet that is allocated to the organization:

- ▶ 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 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:

- ▶ Quotas are assigned when organizations are created.
- ▶ Any application or service in a space of the organization contributes to the usage 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.

2.23 Bluemix organizations: Spaces



Bluemix organizations: Spaces

- 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, development, testing, staging, and production environment

© Copyright IBM Corporation 2016

Figure 2-21 Bluemix organizations: Spaces

Notes:

Spaces in Bluemix are the next level down from organizations:

- ▶ 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 Bluemix user.
- ▶ Bluemix users typically use spaces to delineate different types of deployment environments, such as development, testing, staging and production.

2.24 Bluemix organizations: User roles

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

© Copyright IBM Corporation 2016

Figure 2-22 Bluemix organizations: User roles

Notes:

Users have roles in both 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.

2.25 Managing Bluemix organizations

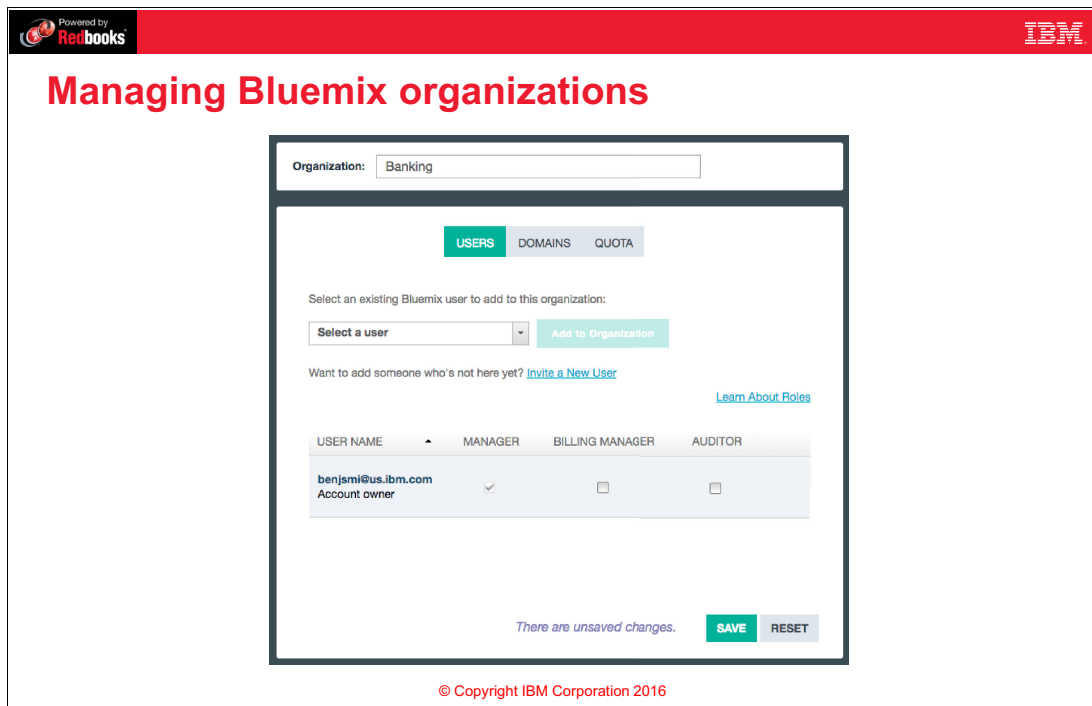
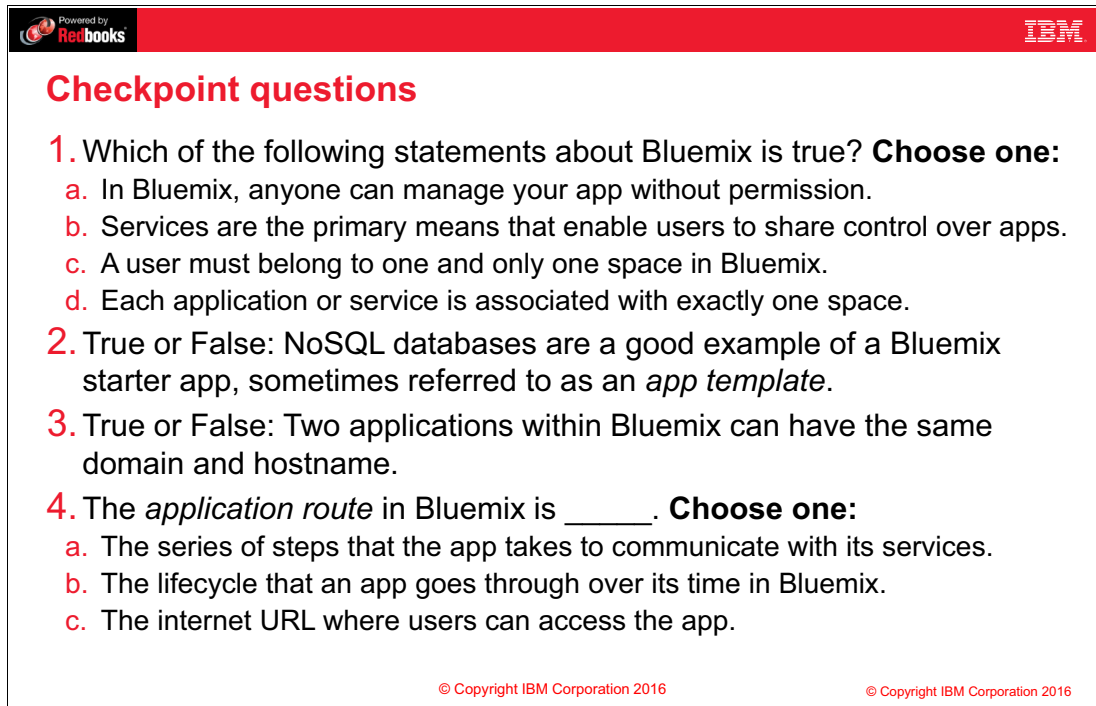


Figure 2-23 Managing Bluemix organizations

Notes:

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

2.26 Checkpoint questions



Powered by **Redbooks** **IBM**

Checkpoint questions

1. Which of the following statements about Bluemix is true? **Choose one:**
 - a. In Bluemix, anyone can manage your app 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 Bluemix.
 - d. Each application or service is associated with exactly one space.
2. True or False: NoSQL databases are a good example of a Bluemix starter app, sometimes referred to as an *app template*.
3. True or False: Two applications within Bluemix can have the same domain and hostname.
4. The *application route* in Bluemix is _____. **Choose one:**
 - a. The series of steps that the app takes to communicate with its services.
 - b. The lifecycle that an app goes through over its time in Bluemix.
 - c. The internet URL where users can access the app.

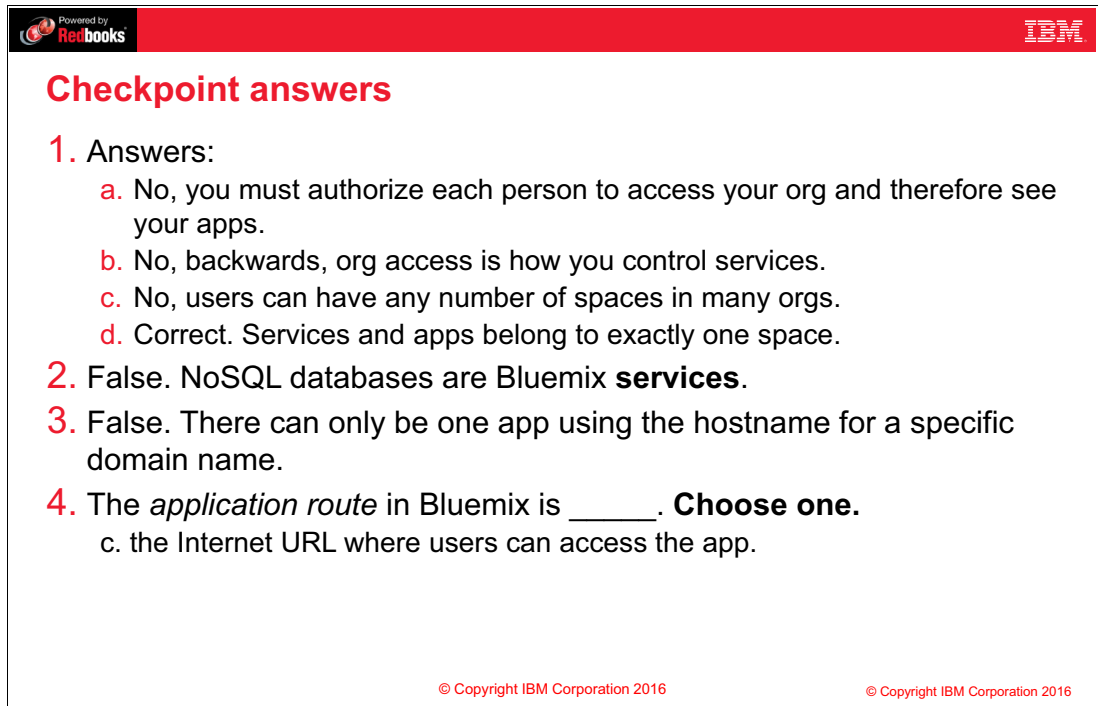
© Copyright IBM Corporation 2016 © Copyright IBM Corporation 2016

Figure 2-24 Checkpoint questions

Notes:

(none)

2.27 Checkpoint answers



The image is a screenshot of a document page. At the top left, there is a logo for 'Powered by Redbooks'. At the top right, there is the 'IBM' logo. The main title of the document is 'Checkpoint answers' in a large, bold, red font. Below the title, there are four numbered items. Item 1 is 'Answers:' followed by four sub-points (a, b, c, d). Item 2 is 'False. NoSQL databases are Bluemix **services**.' Item 3 is 'False. There can only be one app using the hostname for a specific domain name.' Item 4 is 'The *application route* in Bluemix is _____. **Choose one.**' followed by a sub-point 'c. the Internet URL where users can access the app.' At the bottom of the page, there are two small copyright notices: '© Copyright IBM Corporation 2016' on the left and '© Copyright IBM Corporation 2016' on the right.

Powered by
Redbooks

IBM

Checkpoint answers

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

© Copyright IBM Corporation 2016

© Copyright IBM Corporation 2016

Figure 2-25 Checkpoint answers

Notes:

(none)



Developing Bluemix applications from a local workstation

This unit covers the following topics:

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

3.1 What you should be able to do

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

- ▶ Describe how to create an IBM SDK for Node.js application that runs on Bluemix
- ▶ Describe the features in Bluemix that help you to set up a cooperative workstation environment
- ▶ Explain how to manage your Bluemix account with the Cloud Foundry CLI
- ▶ Describe how to setup and use the 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 Create an application using the Bluemix dashboard

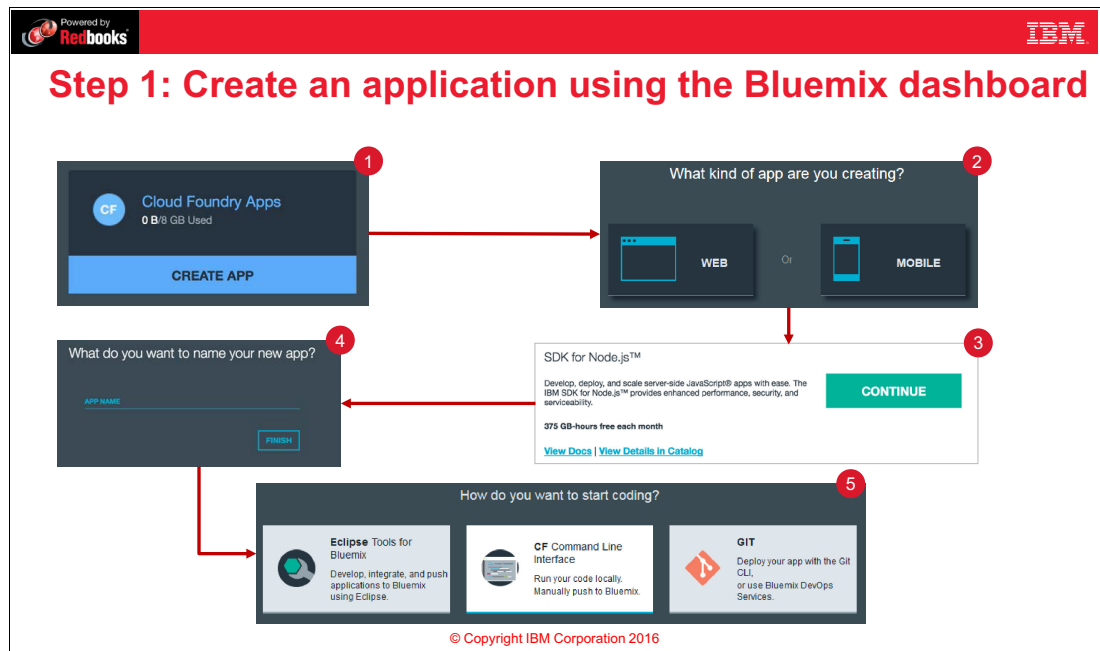


Figure 3-1 Create an application using the Bluemix dashboard

Notes:

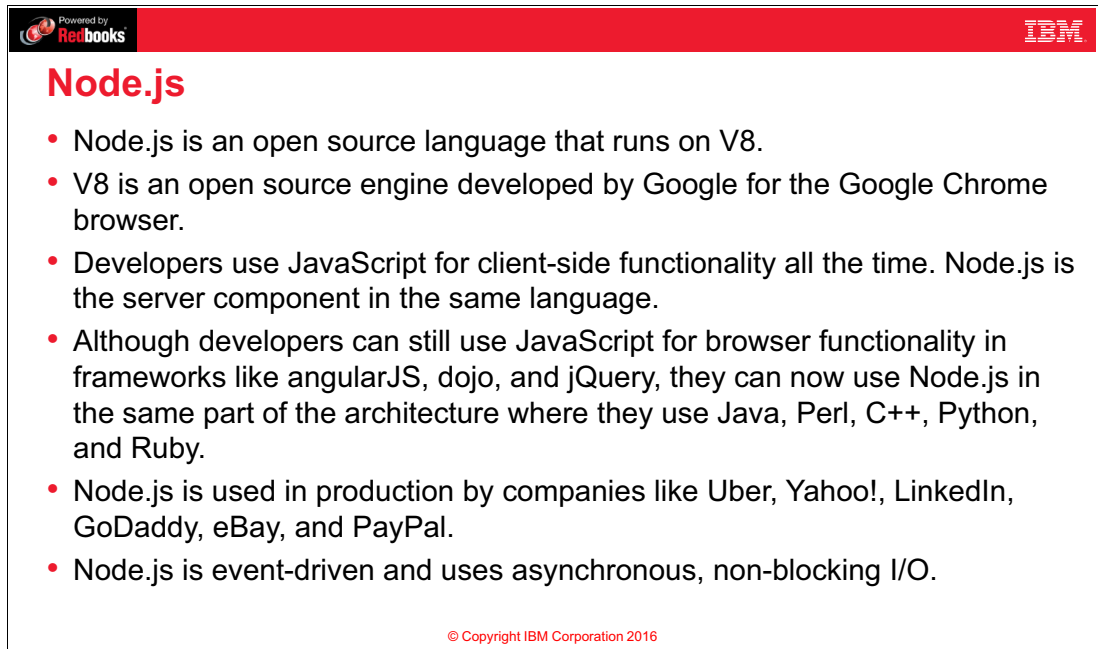
As a review from the last unit, the steps to create an app are:

1. At the Bluemix dashboard, click **CREATE APP** under Cloud Foundry Apps.
2. Select **WEB** to build a web application.
3. Select **SDK for Node.js** as the runtime language and framework on which to base your application and click **CONTINUE**.
4. Enter a name for your app and click **FINISH**.

After your app is created and deployed, you have three ways to modify your Bluemix application:

- ▶ You can install the Eclipse tools for Bluemix in the Eclipse integrated development environment (IDE) on your workstation.
- ▶ You can use the Cloud Foundry command line interface to push changes from your local directory to your IBM Bluemix account.
- ▶ If you want to develop your application in the cloud, you can manage your source code with your own Git repository, or from a Git repository that is provided by the Bluemix DevOps services.

3.4 Node.js



Powered by **Redbooks** **IBM**

Node.js

- Node.js is an 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 all the time. Node.js is the server component in the same language.
- Although developers can still use JavaScript for browser functionality in frameworks like 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 like Uber, Yahoo!, LinkedIn, GoDaddy, eBay, and PayPal.
- Node.js is event-driven and uses asynchronous, non-blocking I/O.

© Copyright IBM Corporation 2016

Figure 3-2 Node.js

Notes:

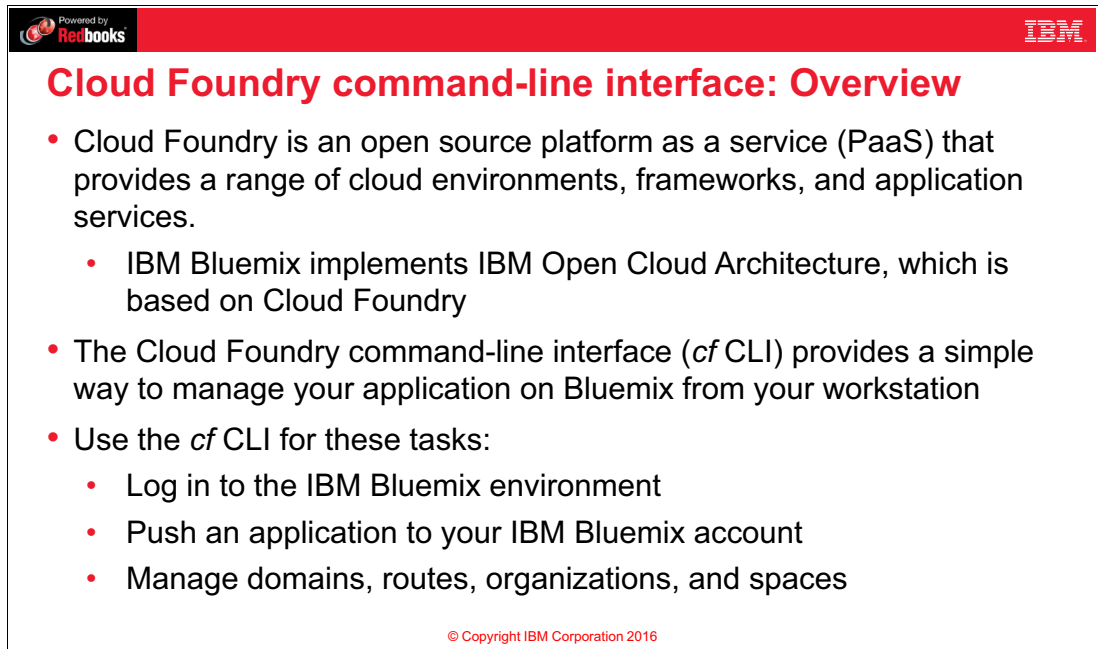
Node.js is open source language that runs on V8. V8 is an open source engine developed by Google for the Google Chrome browser.

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

Although developers can still use JavaScript for browser functionality in frameworks like 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.

3.5 Cloud Foundry command-line interface: Overview



The image shows a screenshot of a Redbooks document. At the top left, there is a logo for Redbooks with the text 'Powered by Redbooks'. At the top right, there is the IBM logo. The main title of the document is 'Cloud Foundry command-line interface: Overview' in red text. Below the title, there is a bulleted list of information about Cloud Foundry and the cf CLI. At the bottom right of the document content, there is a small copyright notice: '© Copyright IBM Corporation 2016'.

Cloud Foundry command-line interface: Overview

- 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
- The Cloud Foundry command-line interface (*cf* CLI) provides a simple way to manage your application on Bluemix from your workstation
- Use the *cf* CLI for these tasks:
 - Log in to the IBM Bluemix environment
 - Push an application to your IBM Bluemix account
 - Manage domains, routes, organizations, and spaces

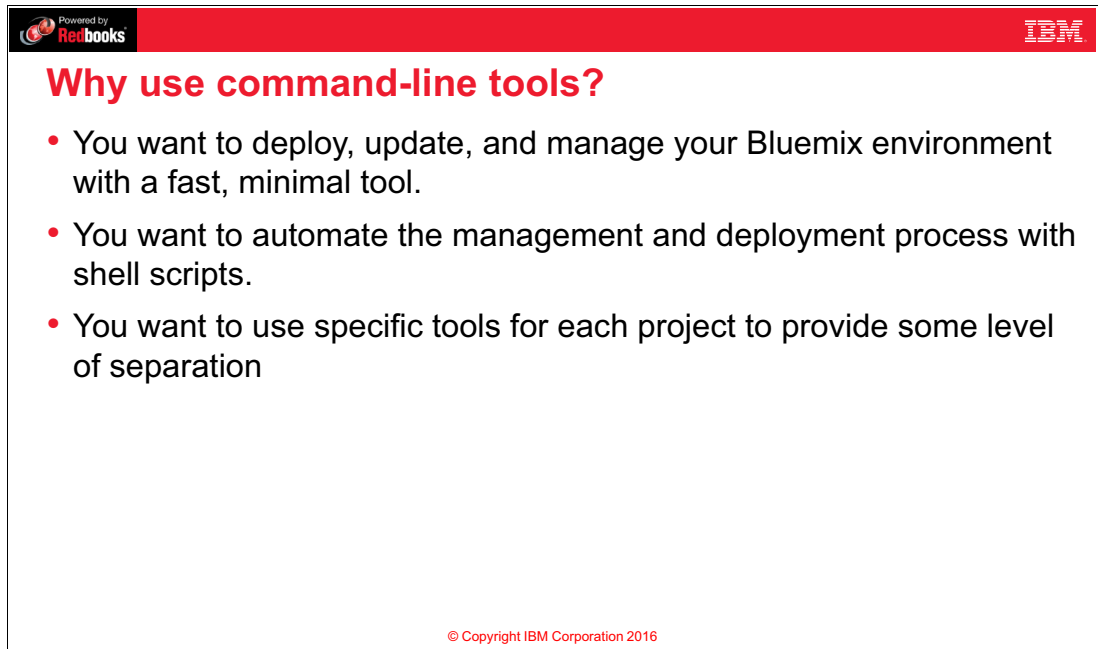
© Copyright IBM Corporation 2016

Figure 3-3 Cloud Foundry command-line interface: Overview

Notes:

Cloud Foundry is an open source PaaS. Cloud Foundry provides an open source command-line interface, available for multiple platforms. You can use the command-line interface to log in to Bluemix, deploy an application to 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 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

© Copyright IBM Corporation 2016

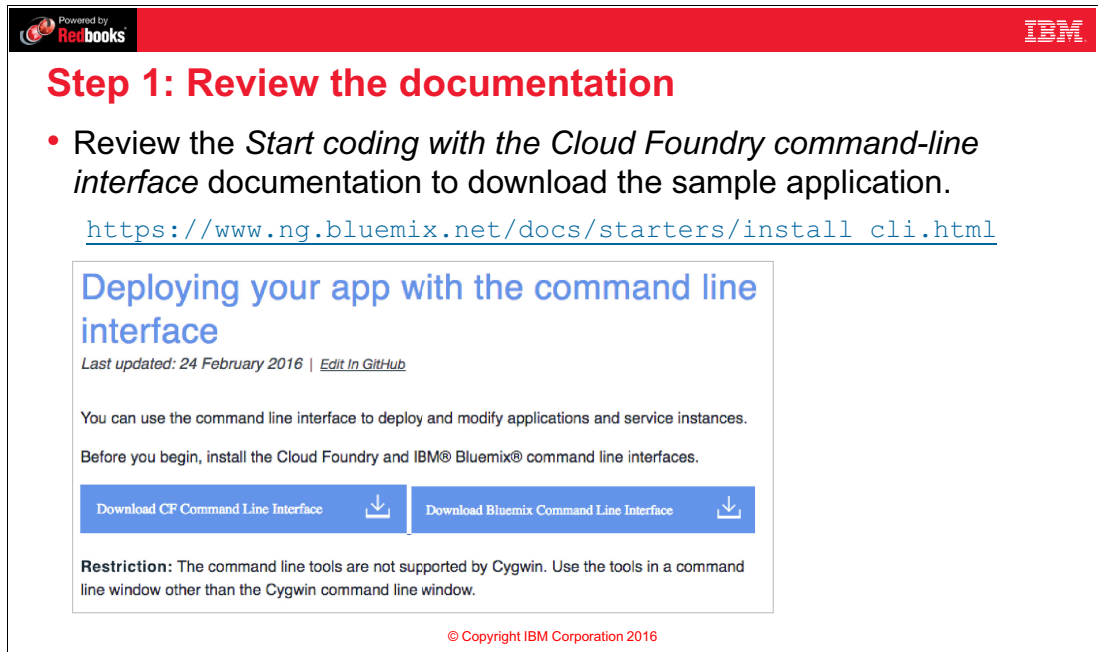
Figure 3-4 Why use command-line tools?

Notes:

As mentioned previously, Bluemix integrates into the Eclipse IDE. With this graphical user interface available, why would you want to use command-line tools? Some possible reasons are that you want to use a tool that is quick to install and has a minimal footprint on your development workstation.

You may 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 Step 1: Review the documentation



The screenshot shows a Redbooks document page. At the top left, it says 'Powered by Redbooks'. At the top right, the IBM logo is visible. The main heading is 'Step 1: Review the documentation' in red. Below this is a bullet point: 'Review the *Start coding with the Cloud Foundry command-line interface* documentation to download the sample application.' A URL is provided: https://www.ng.bluemix.net/docs/starters/install_cli.html. Below the URL is a preview of a document titled 'Deploying your app with the command line interface'. The preview includes the text: 'Last updated: 24 February 2016 | [Edit In GitHub](#)', 'You can use the command line interface to deploy and modify applications and service instances.', and 'Before you begin, install the Cloud Foundry and IBM® Bluemix® command line interfaces.' There are two blue buttons: 'Download CF Command Line Interface' and 'Download Bluemix Command Line Interface', each with a download icon. Below the buttons is a 'Restriction' note: 'The command line tools are not supported by Cygwin. Use the tools in a command line window other than the Cygwin command line window.' At the bottom right of the preview, it says '© Copyright IBM Corporation 2016'.

Figure 3-5 Step 1: Review the documentation

Notes:

You can review the documentation for the command-line tool at the webpage shown on this slide. From this page, you can also download the Cloud Foundry command-line interface by clicking **Download CF Command Line Interface**.

3.8 Step 2: Install the Cloud Foundry command-line interface

The slide features a red header with the Redbooks logo on the left and the IBM logo on the right. The main content is on a white background. It includes a title, three bullet points with instructions, a URL, and two lists of download links. A copyright notice is at the bottom.

Step 2: Install the Cloud Foundry command-line interface

- Download the most recent cf installer from the Git project:
<https://github.com/cloudfoundry/cli/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 32 bit
- Debian 64 bit
- Redhat 32 bit
- Redhat 64 bit
- Mac OS X 64 bit
- Windows 32 bit
- Windows 64 bit

Binaries

- Linux 32 bit binary
- Linux 64 bit binary
- Mac OS X 64 bit binary
- Windows 32 bit binary
- Windows 64 bit binary

© Copyright IBM Corporation 2016

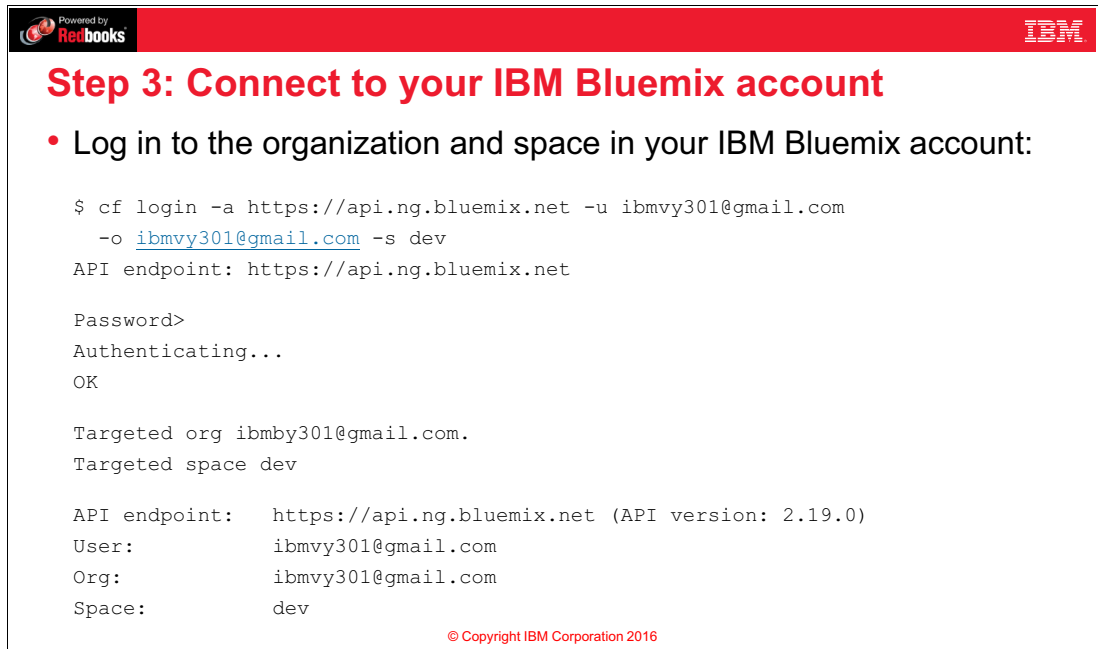
Figure 3-6 Step2: Install the Cloud Foundry command-line interface

Notes:

The **Download CF Command Line Interface** link will take you to the link provided on this slide, which has 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 details on this process are included in the exercises accompanying this unit.

3.9 Step 3: Connect to your IBM Bluemix account



The slide features a red header with the Redbooks logo on the left and the IBM logo on the right. The main content is a terminal window showing the following text:

```
$ 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 2016

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

Notes:

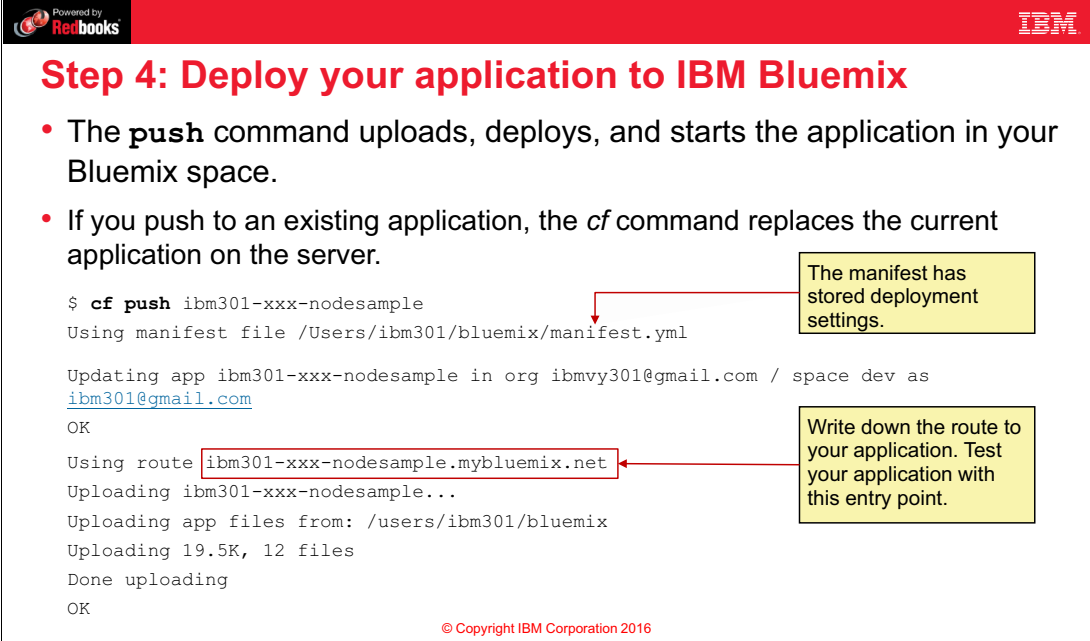
Next, log in to Bluemix using the command-line interface. The login command requires five parameters:

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

The exercises provide step-by-step instructions for how to log in this way. 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 will prompt you to enter it later. In the example shown in the slide, the password was not entered, so the command-line utility prompts the user to enter the password.

3.10 Step 4: Deploy your application to IBM Bluemix



```
Powered by Redbooks IBM  
Step 4: Deploy your application to IBM Bluemix  


- The push command uploads, deploys, and starts the application in your Bluemix space.
- If you push to an existing 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
Uploading 19.5K, 12 files
Done uploading
OK
```



The manifest has stored deployment settings.



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



© Copyright IBM Corporation 2016


```

Figure 3-8 Step 4: Deploy your application to IBM Bluemix

Notes:

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

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

Keep in mind that when you push your application, it overwrites the current contents of your application in your Bluemix account. Write down the application route to your application. You can test your application with this entry point.


Note that this slide shows the command-line utility accessing a manifest file. You can control various aspects of the deployment, such as the amount of memory an app receives, the route of the application, and so on, by specifying them as flags in the command. However, you can also store these settings in a manifest YAML file to make the command you need shorter to type in later iterations of your app.

3.11 Step 5: Test your Bluemix application

Powered by **Redbooks** **IBM**

Step 5: Test your Bluemix application

- Test the route for your application. In the IBM SDK for Node.js sample application, open the route as a web page to test the application.



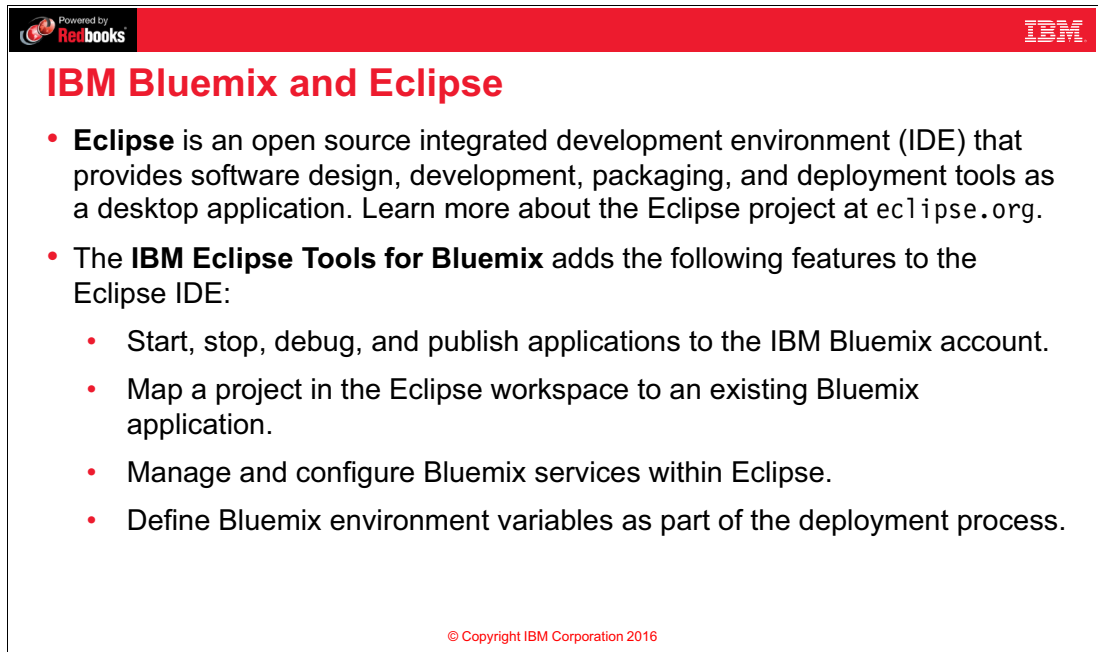
© Copyright IBM Corporation 2016

Figure 3-9 Step 5: Test your 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.

3.12 IBM Bluemix and Eclipse



The image shows a screenshot of a Redbooks document. At the top left, there is a logo for 'Powered by Redbooks'. At the top right, there is the IBM logo. The main title of the document is 'IBM Bluemix and Eclipse'. Below the title, there is a bulleted list of features and information about Eclipse and the IBM Eclipse Tools for Bluemix. At the bottom right of the document content, there is a copyright notice: '© Copyright IBM Corporation 2016'.

IBM Bluemix and Eclipse

- **Eclipse** is an open source integrated development environment (IDE) that provides software design, development, packaging, and deployment tools as a desktop application. Learn more about the Eclipse project at eclipse.org.
- The **IBM Eclipse Tools for 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 existing Bluemix application.
 - Manage and configure Bluemix services within Eclipse.
 - Define Bluemix environment variables as part of the deployment process.

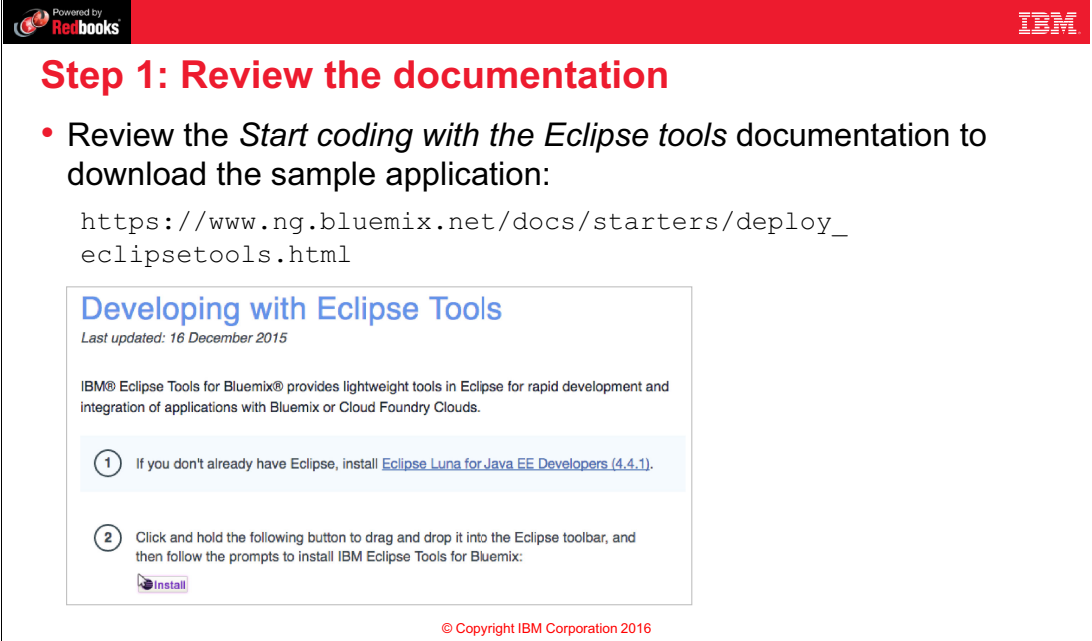
© Copyright IBM Corporation 2016

Figure 3-10 IBM Bluemix and Eclipse

Notes:

Eclipse is a popular open source integrated development environment or IDE that provides development tools in a desktop application. You can learn more about Eclipse at eclipse.org. IBM Eclipse tools for Bluemix is an open source plugin for interacting with Bluemix to manage and deploy your applications and services from within the Eclipse interface.

3.13 Step 1: Review the documentation



Powered by **Redbooks** **IBM**


Step 1: Review the documentation

- Review the *Start coding with the Eclipse tools* documentation to download the sample application:
`https://www.ng.bluemix.net/docs/starters/deploy_eclipsetools.html`

Developing with Eclipse Tools

Last updated: 16 December 2015

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

- 1 If you don't already have Eclipse, install [Eclipse Luna for Java EE Developers \(4.4.1\)](#).
- 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:


© Copyright IBM Corporation 2016

Figure 3-11 Step 1: Review the documentation

Notes:

To start developing with the IBM Eclipse Tools for Bluemix plugin, you must first have Eclipse Luna for Java EE Developers installed, which requires having the Java Development Kit installed. You can then browse to the URL on this slide to review the documentation for Eclipse tools.

3.14 Step 2: Install Eclipse Luna and Eclipse tools for Bluemix

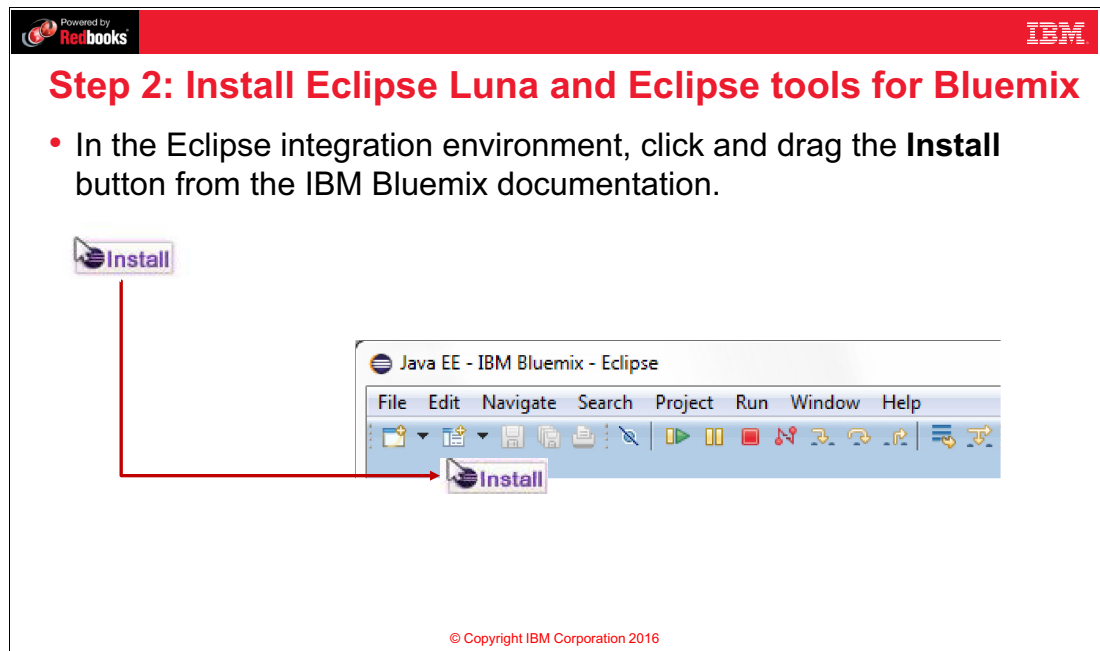


Figure 3-12 Step 2: Install Eclipse Luna and Eclipse tools for Bluemix

Notes:

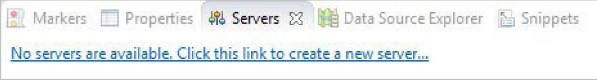
The simplest way to install the IBM Eclipse Tools for Bluemix is to drag the **Install** button from the documentation on top of your Eclipse toolbar.

3.15 Step 3: Connect to your IBM Bluemix account

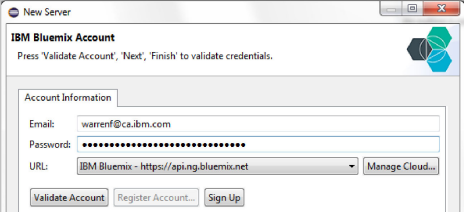
Powered by **Redbooks** **IBM**

Step 3: Connect to your IBM Bluemix account

- 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.





© Copyright IBM Corporation 2016

Figure 3-13 Step 3: Connect to your IBM Bluemix account

Notes:


After you have successfully installed the IBM Bluemix tools for Eclipse plugin, you can set up a server object within Eclipse that links to your Bluemix account. Create a server object, and then enter the IBM ID and password that you use to log in to your Bluemix account.

3.16 Step 4: Create a Node.js application project

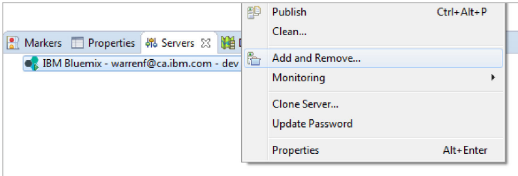
Powered by 

Step 4: Create a Node.js application project

- 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.



- Add the project to the IBM Bluemix server configuration.



© Copyright IBM Corporation 2016

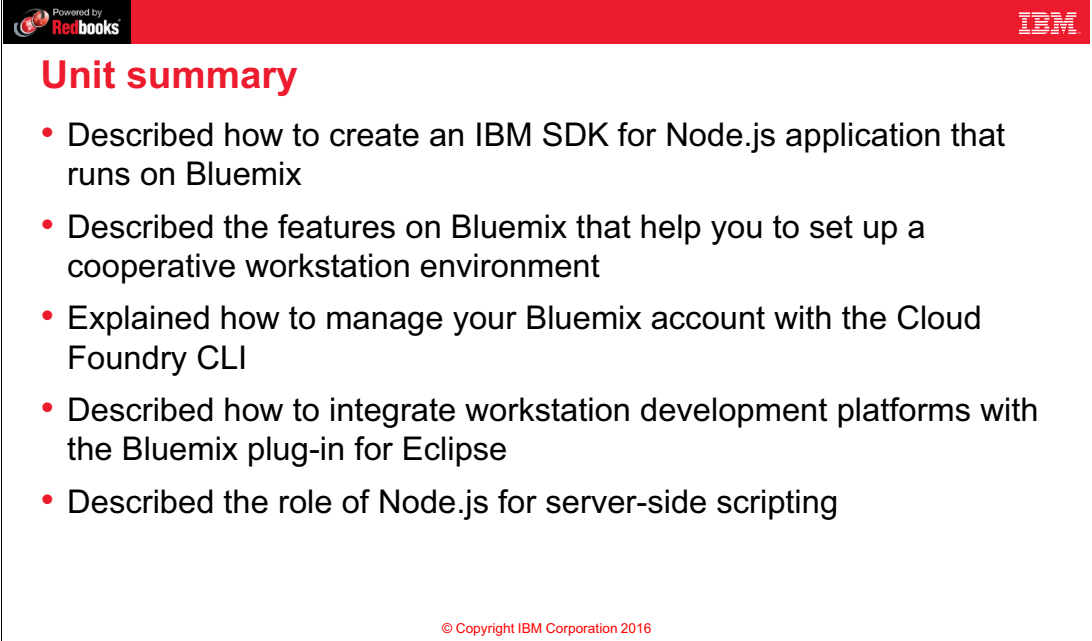
Figure 3-14 Step 4: Create a Node.js application project

Notes:

You can only add projects for server runtime environments that 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 using Eclipse project facets.

To view or modify the facets for your project, select the **Properties** for your project and select **Project Facets**. Once you have added the Node.js facet to your application, it will show up in Eclipse as an item that you can deploy to Bluemix.

3.17 Unit summary



Powered by **Redbooks** **IBM**

Unit summary

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

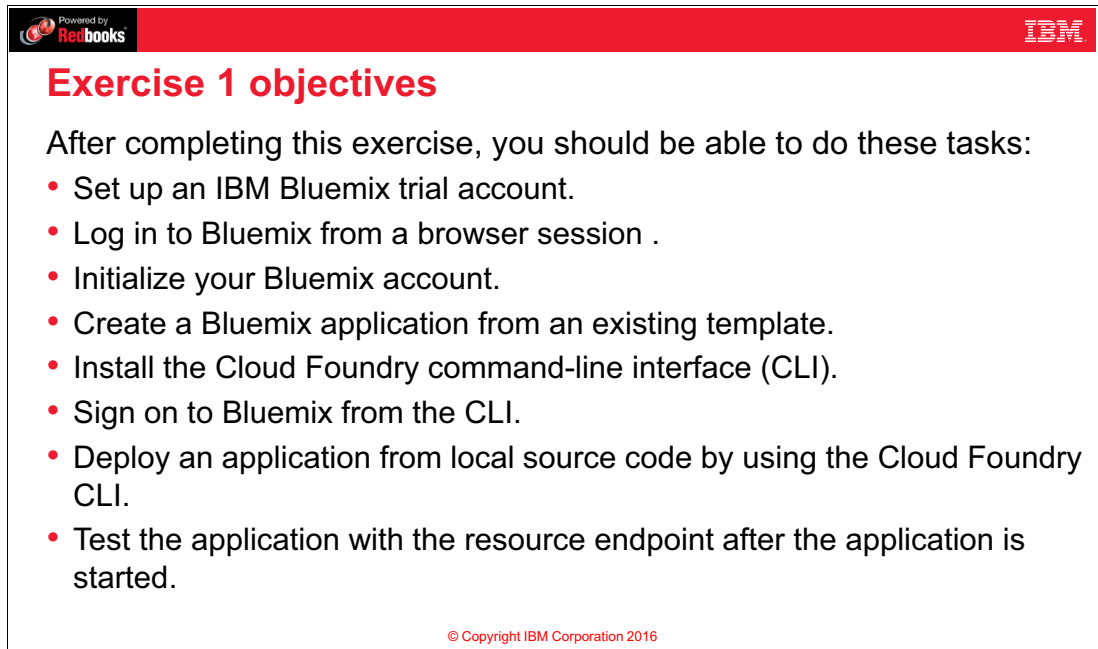
© Copyright IBM Corporation 2016

Figure 3-15 Unit summary

Notes:

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

3.18 Exercise 1 objectives



Powered by **Redbooks** **IBM**

Exercise 1 objectives

After completing this exercise, you should be able to do these tasks:

- Set up an IBM Bluemix trial account.
- Log in to Bluemix from a browser session .
- Initialize your Bluemix account.
- Create a Bluemix application from an existing template.
- Install the Cloud Foundry command-line interface (CLI).
- Sign on to Bluemix from the CLI.
- Deploy an application from local source code by using the Cloud Foundry CLI.
- Test the application with the resource endpoint after the application is started.

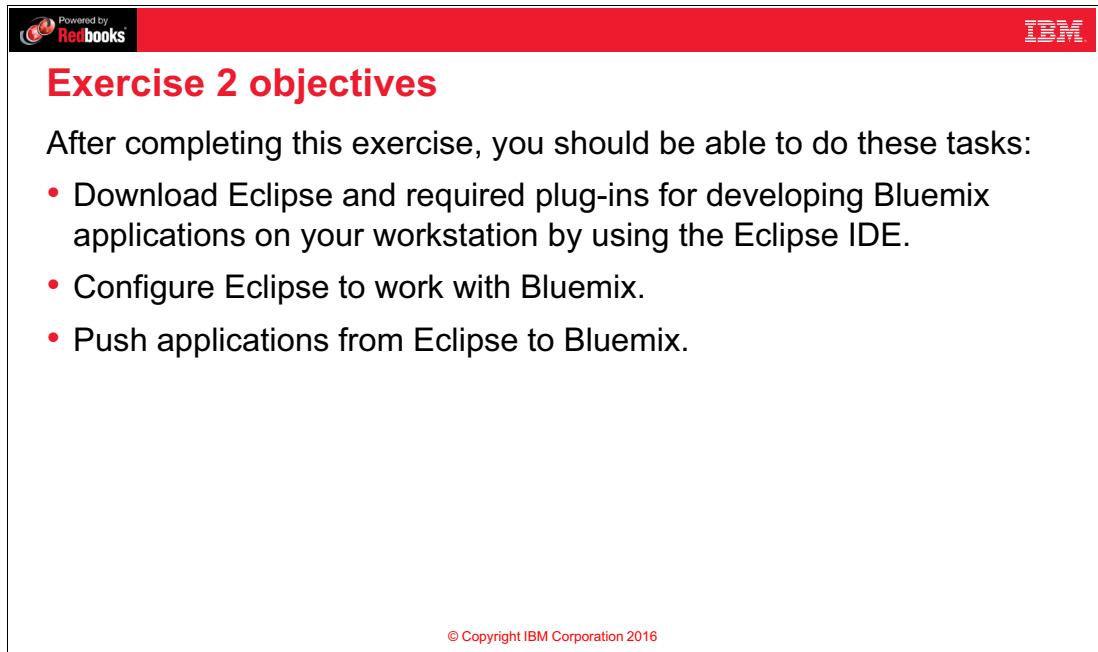
© Copyright IBM Corporation 2016

Figure 3-16 Exercise 1 objectives

Notes:

(none)

3.19 Exercise 2 objectives



Powered by **Redbooks** **IBM**

Exercise 2 objectives

After completing this exercise, you should be able to do these tasks:

- Download Eclipse and required plug-ins for developing Bluemix applications on your workstation by using the Eclipse IDE.
- Configure Eclipse to work with Bluemix.
- Push applications from Eclipse to Bluemix.

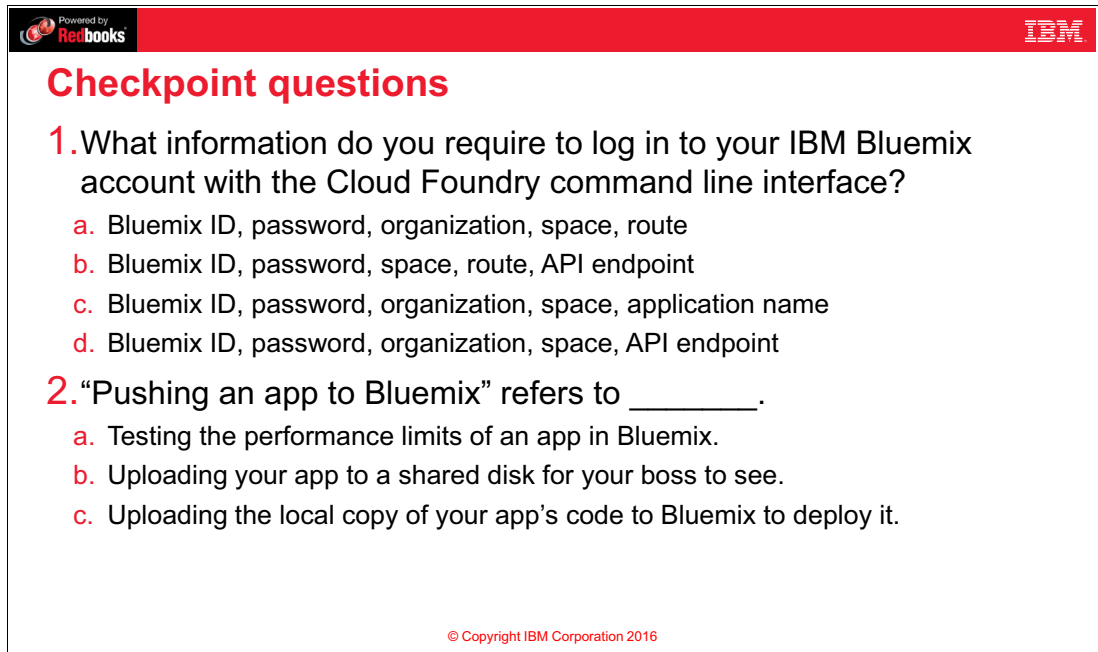
© Copyright IBM Corporation 2016

Figure 3-17 Exercise 2 objectives

Notes:

(none)

3.20 Checkpoint questions



Powered by **Redbooks** **IBM**

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 Bluemix” refers to _____.
 - a. Testing the performance limits of an app in Bluemix.
 - b. Uploading your app to a shared disk for your boss to see.
 - c. Uploading the local copy of your app’s code to Bluemix to deploy it.

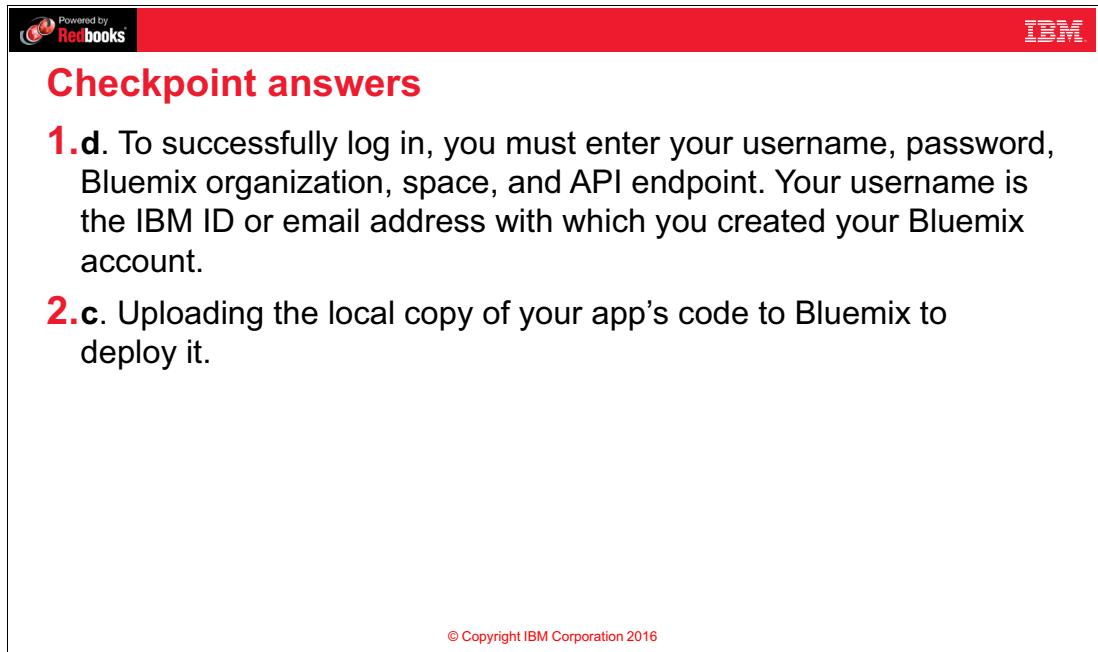
© Copyright IBM Corporation 2016

Figure 3-18 Checkpoint questions

Notes:

(none)

3.21 Checkpoint answers



The screenshot shows a document header with the Redbooks logo on the left and the IBM logo on the right. The main title is 'Checkpoint answers' in red. Below the title are two numbered items:

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

At the bottom right of the document, there is a copyright notice: © Copyright IBM Corporation 2016.

Figure 3-19 Checkpoint answers

Notes:

(none)



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 perform these tasks:

- ▶ 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/>
- ▶ An 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>
https://console.ng.bluemix.net/docs/starters/deploy_devops.html
<https://hub.jazz.net/tutorials/trackplan/>
https://hub.jazz.net/docs/deploy_ext/#activedeploy

4.3 What is DevOps?

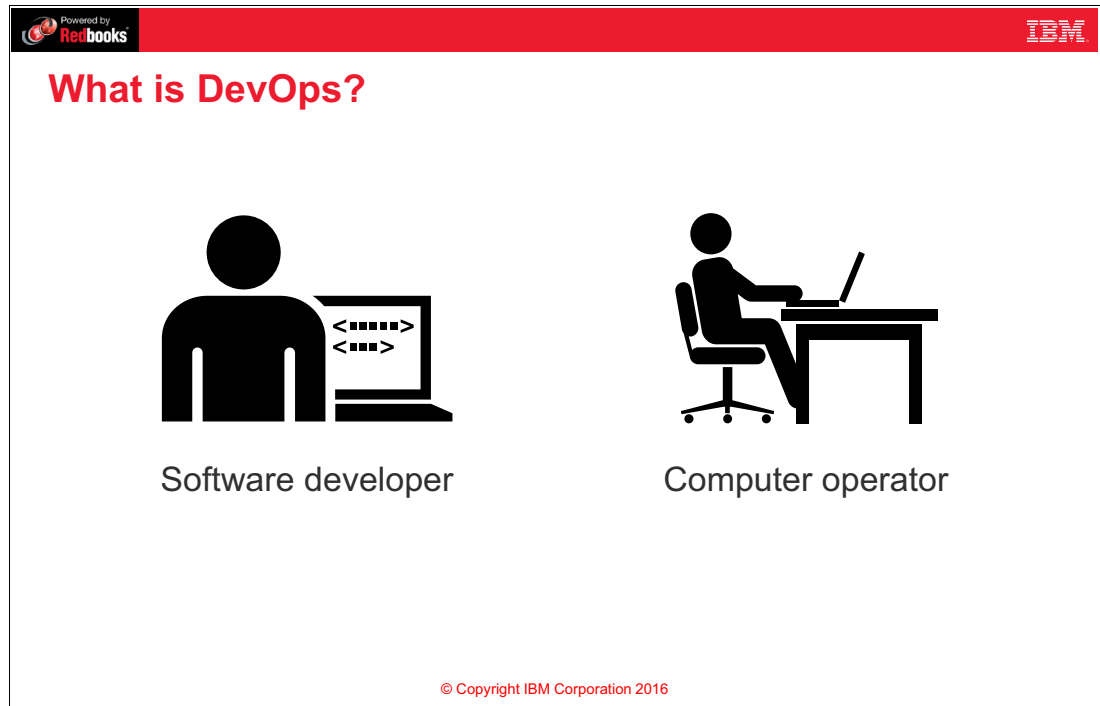


Figure 4-1 What is DevOps?

Notes:

Before DevOps, there were two teams responsible for code delivery:

- ▶ Development team: Designed the code, delivered new features, fixed bugs, and tested the code.
- ▶ Operations 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 the operations and development teams work together to support the software lifecycle, from code design all the way to deployment to production. This includes automating all 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

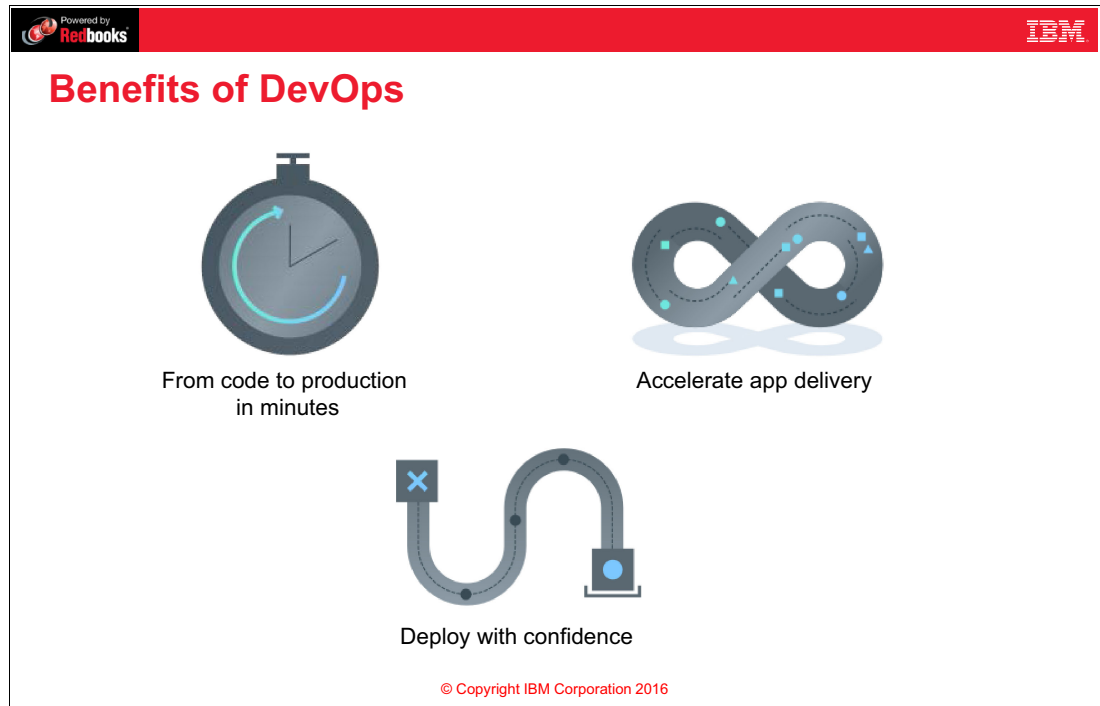


Figure 4-2 Benefits of DevOps

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?

Powered by Redbooks IBM

What is IBM Bluemix DevOps Services?

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

© Copyright IBM Corporation 2016

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

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

4.6 What services does Bluemix DevOps provide?

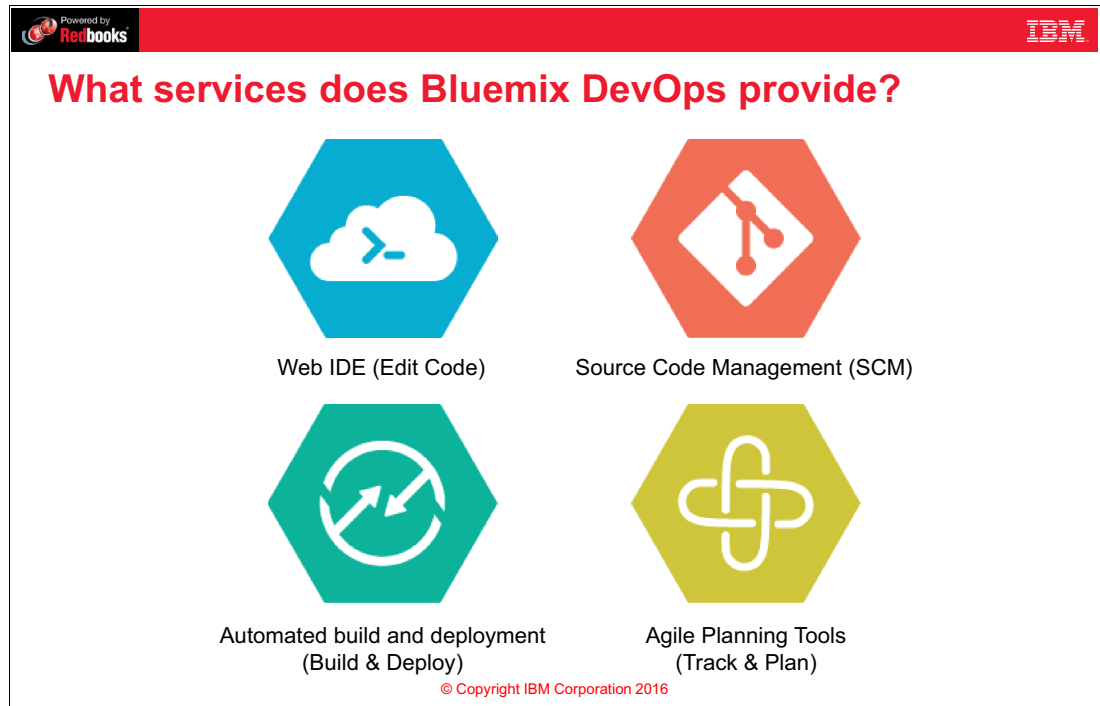


Figure 4-4 What services does Bluemix DevOps provide?

Notes:

IBM Bluemix DevOps provides a comprehensive set of features for the operations and software development teams.

You develop your code (for example, Node.js, Java, or any other code) in a web-based integrated development environment (**Web IDE**). No additional 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*. Although using the Web IDE is great, you can still use a desktop IDE such as Eclipse with DevOps Services.

The local repository is a copy of your latest edits before you submit your work to a **source control system (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 Bluemix DevOps Services infrastructure.

Bluemix **Build & Deploy** automates the process of building and deploying your code as a Bluemix application. You can also configure the build, deploy, or test scripts within the Web IDE.

To track the progress of your development and operations, Bluemix DevOps Services provide agile planning tools. You can use these tools to track and plan your agile project. You can manage any type of project with an agile approach by using the **Track & Plan** feature in IBM Bluemix DevOps Services. The Track & Plan tools simplify project planning and speed your workflow.

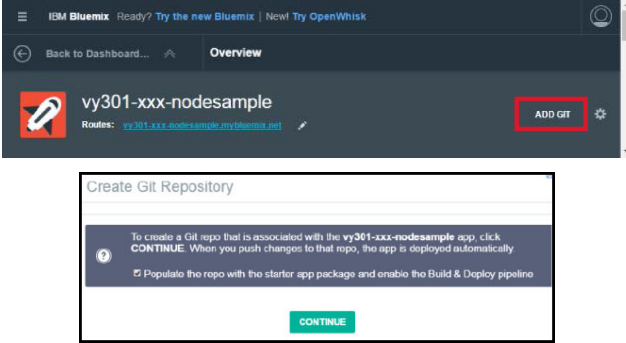
4.7 Setting up a Bluemix DevOps Services project

Powered by **Redbooks** **IBM**

Setting up a Bluemix DevOps Services project

To create a Bluemix DevOps Services project, complete these steps:

1. In the Bluemix dashboard, select your app.
2. Click **ADD GIT**.
3. At the Create Git Repository window, click **CONTINUE**.



© Copyright IBM Corporation 2016

Figure 4-5 Setting up a Bluemix DevOps Services project

Notes:

You must create a project to manage, build, and deploy your application from Bluemix DevOps Services.

From the Bluemix dashboard, select your app and then click **ADD GIT**. Create Git Repository creates a Git repository that contains the code of your app. It also creates a DevOps Services project that you can use to develop the app online.

4.8 Add Bluemix DevOps capabilities

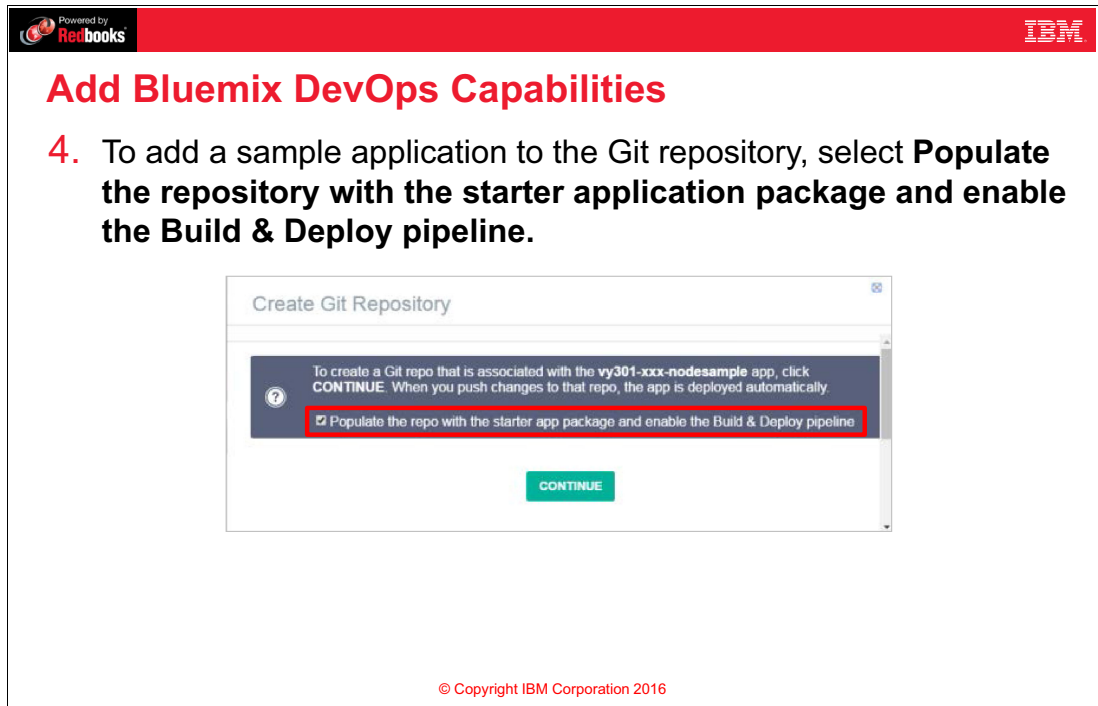
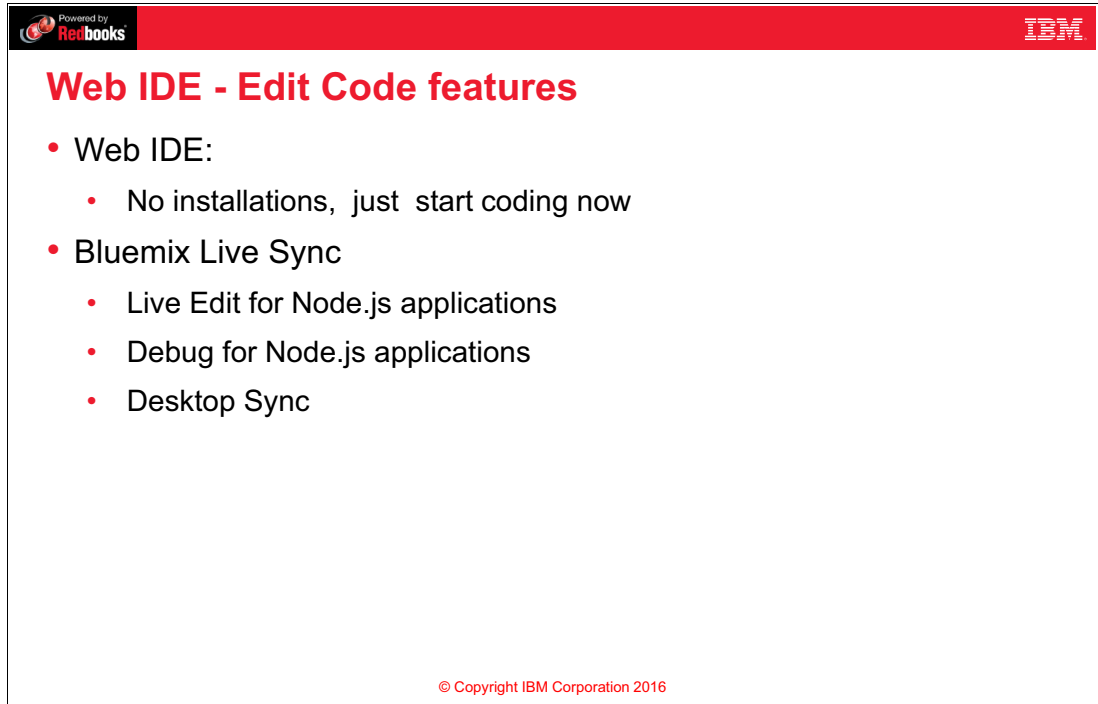


Figure 4-6 Add Bluemix DevOps capabilities

Notes:

You can create an empty project or selecting **Populate the repository with the starter application package and enable the Build & Deploy pipeline**. This option populates the local repository with a starter application, and also creates a simple delivery pipeline that packages the application and pushes it to your Bluemix account.

4.9 Web IDE: Edit Code features



The image shows a slide with a red header bar. On the left of the header is the 'Powered by Redbooks' logo, and on the right is the 'IBM' logo. The main title of the slide is 'Web IDE - Edit Code features' in red. Below the title is a bulleted list of features. At the bottom of the slide, there is a small copyright notice: '© Copyright IBM Corporation 2016'.

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

© Copyright IBM Corporation 2016

Figure 4-7 Web IDE: Edit Code features

Notes:

The Edit Code features in IBM Bluemix DevOps provide two capabilities:

- ▶ Web IDE capabilities:
 - 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, right within 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 only allowed for Node.js applications): Allows you to make changes to your application from the Web IDE without the need of redeploying it.
 - Debug (Currently only allowed for Node.js applications): While 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.

4.10 Web integrated development environment: Overview

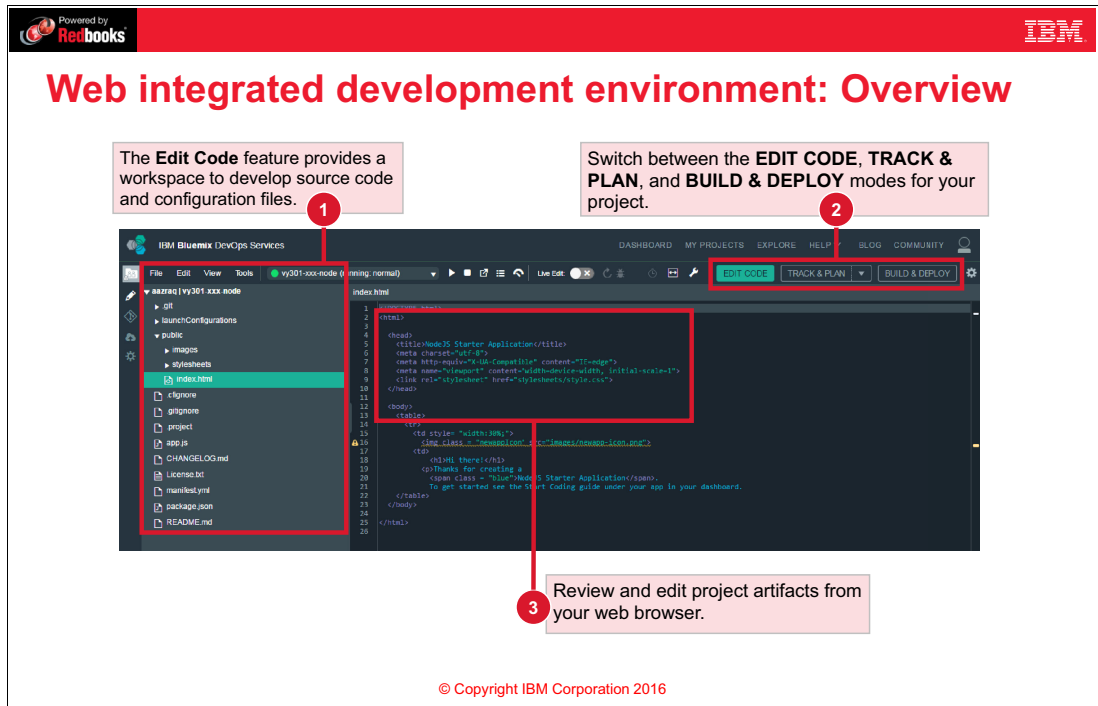


Figure 4-8 Web integrated development environment: Overview

Notes:

The figure shows the Edit Code perspective for the Web IDE in Bluemix DevOps. Project Navigator displays the contents of the Bluemix DevOps user directory for the project.

You can quickly switch between the **EDIT CODE**, **TRACK & PLAN**, and **BUILD & DEPLOY** modes with the buttons on the top of the page.

When you select a file, the editor displays its 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 will be described in the next sections

4.11 Editing source code in Bluemix DevOps Services

The screenshot displays the Bluemix DevOps Services editor interface. At the top, there are logos for 'Powered by Redbooks' and 'IBM'. The main title is 'Editing source code in Bluemix DevOps Services'. Below the title is a bullet point: 'The Bluemix DevOps editor provides a fully featured environment for writing your application in your web browser.' The central part of the image shows a code editor with a dark background and light text. The code is a JavaScript file named 'app.js' that uses Express.js and Jade. Three red callout boxes with numbers 1, 2, and 3 point to specific features: 1. 'Real-time validation and syntax checking' points to the top of the code editor. 2. 'Editor parses source code for Node.js, Java, Python, and Markdown documents' points to the code editor. 3. 'Document preview, flagging warnings and errors' points to a preview window on the right side of the editor. The code in the editor includes comments and a 'var express = require('express');' line. A copyright notice '© Copyright IBM Corporation 2016' is visible at the bottom of the screenshot.

Powered by Redbooks IBM

Editing source code in Bluemix DevOps Services

- The Bluemix DevOps editor provides a fully featured environment for writing your application in your web browser.

1 Real-time validation and syntax checking

2 Editor parses source code for Node.js, Java, Python, and Markdown documents

3 Document preview, flagging warnings and errors

```
1 /*jshint node:true*/
2
3 // app.js
4 // This file contains the server side JavaScript code for your application.
5 // This sample application uses express as web application framework (http://
6 // and jade as template engine (http://jade-lang.com/).
7
8 var express = require('express');
9
10 // setup middleware
11 var app = express();
12 app.use(app.router);
13 app.use(express.errorHandler());
14 app.use(express.static(__dirname + '/public')); //setup static public direc
15 app.set('view engine', 'jade');
16 app.set('views', __dirname + '/views'); //optional since express defaults t
```



© Copyright IBM Corporation 2016

Figure 4-9 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 like JSHint, which is a JavaScript 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.

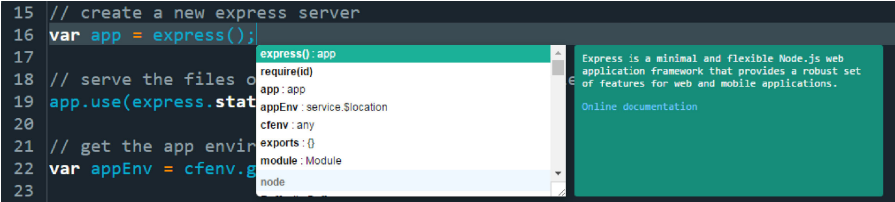
4.12 Editor features: Code completion

Powered by 

Editor features: Code completion

The **code completion** feature suggests functions and parameters within the editor:

- This feature dynamically displays all libraries available at the cursor point.
- Use **templates** to add blocks of code for common tasks, such as error handling or object creation.
- Press **Ctrl + Space bar** to activate the code completion menu.



© Copyright IBM Corporation 2016

Figure 4-10 Editor features: Code completion

Notes:

To execute 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, in this figure the code completion feature displays functions and templates for the Express web application framework for Node.js because that is what the cursor is near.

4.13 Editor features: Run bar

Powered by **Redbooks** **IBM**

Editor features: Run bar

- You can deploy applications from the **run** bar directly from the Web IDE:

1. The status area shows which launch configuration is selected, and the deployment status.
2. Press **Play** to deploy the app.
3. Press **Stop** to stop the application.
4. Open the deployed app.
5. Open the logs.
6. Open the Bluemix application overview page

- Click **Play** to deploy the code in the current state of your workspace.
- To deploy only the changes that you checked in to the repository, use **BUILD & DEPLOY**.

© Copyright IBM Corporation 2016

Figure 4-11 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 status area displays which launch configuration the run bar uses during the build and deploy task.
- When you press the Play button from the Run bar, you build and deploy the code in your user directory to your Bluemix account.
- You can Stop the application
- You can open the application route
- You can open application logs
- You can access the dashboard from the same bar as shown in points.

4.14 Bluemix Live Sync features

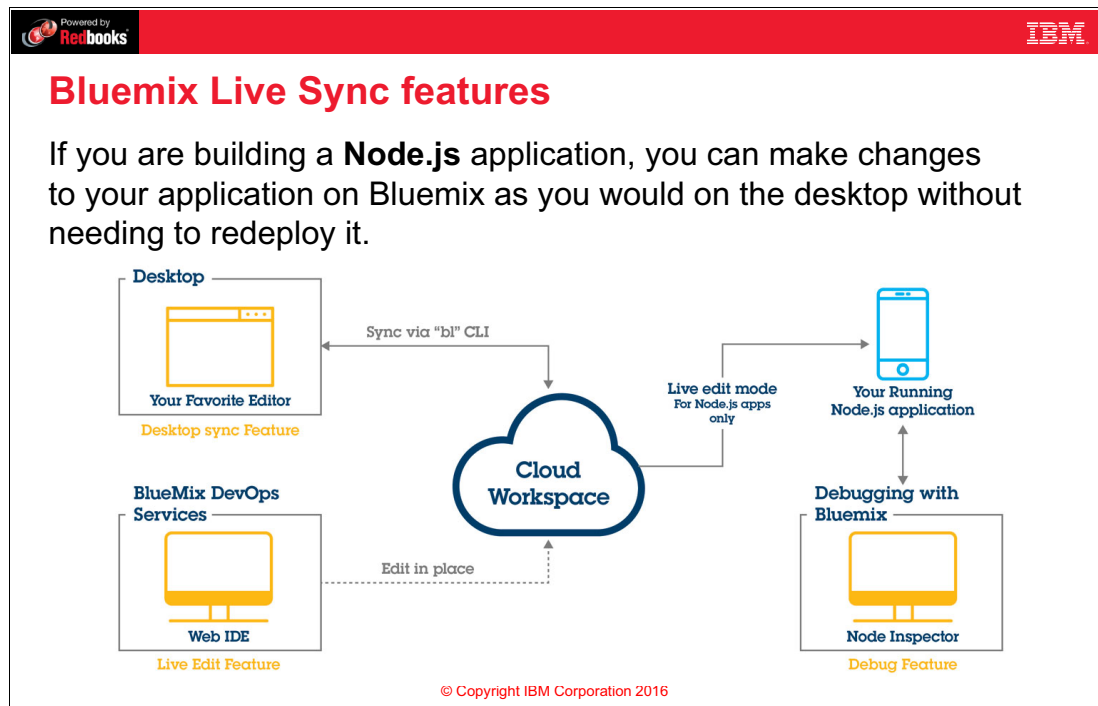


Figure 4-12 Bluemix Live Sync features

Notes:

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

When you make a change, you can see that change in your running 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.

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

Bluemix Live Sync consists of three 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 Bluemix Live Sync command line. Desktop sync works with all types of applications.

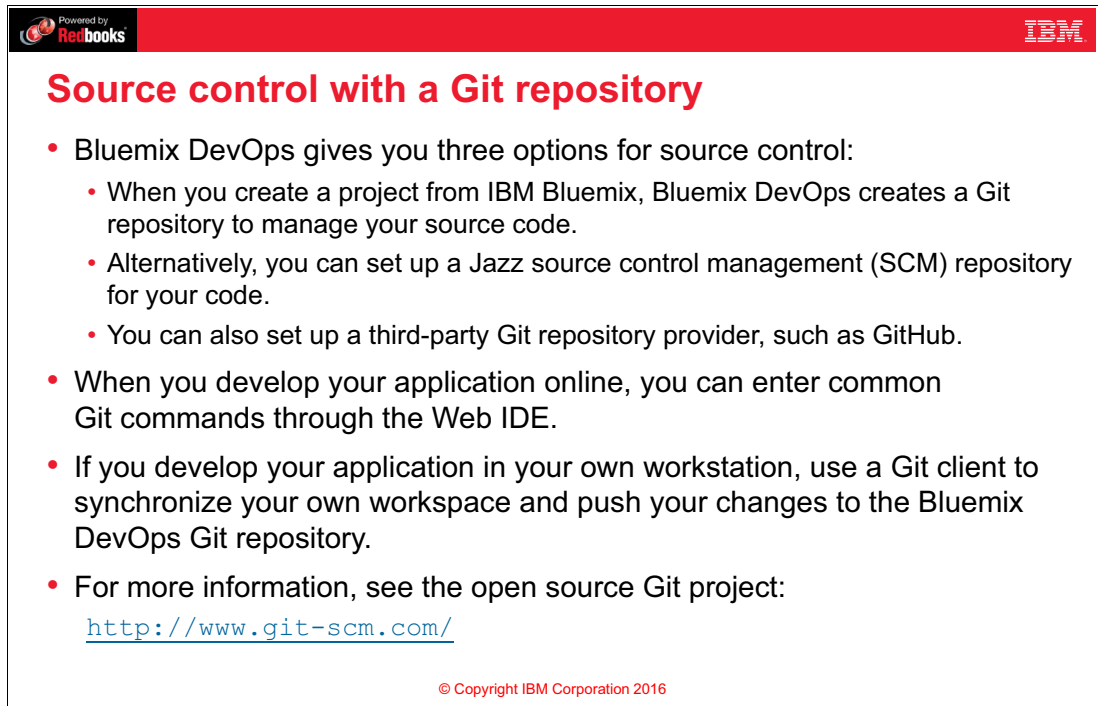
- ▶ Live Edit (only for Node.js applications)

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

- ▶ Debug (only for Node.js applications)

While 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 be able to use the Debug feature.

4.15 Source control with a Git repository



The screenshot shows a Redbooks article header with the IBM logo. The title is 'Source control with a Git repository'. The main content is a bulleted list of three options for source control in Bluemix DevOps, followed by instructions for using Git commands in the Web IDE and a Git client on a workstation. A URL is provided for more information. A copyright notice for IBM Corporation 2016 is at the bottom.

Powered by Redbooks IBM

Source control with a Git repository

- Bluemix DevOps gives you three options for source control:
 - When you create a project from IBM Bluemix, Bluemix DevOps creates a Git repository to manage your source code.
 - Alternatively, you can set up a Jazz source control management (SCM) repository for your code.
 - You can also set up a third-party Git repository provider, such as GitHub.
- 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 Bluemix DevOps Git repository.
- For more information, see the open source Git project:
<http://www.git-scm.com/>

© Copyright IBM Corporation 2016

Figure 4-13 Source control with a Git repository

Notes:

By default, the **Add Git** feature in the Bluemix dashboard creates a Bluemix DevOps Services project and Git repository. Git is an open source change management system. If your organization uses IBM Rational® Team Concert™, you can set up a Jazz™ SCM environment to manage your code. You can also point your Bluemix DevOps Services project to an external, third-party Git repository provider, such as GitHub.

The **Git repository** perspective in the 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 www.git-scm.com.

4.16 Git repository overview

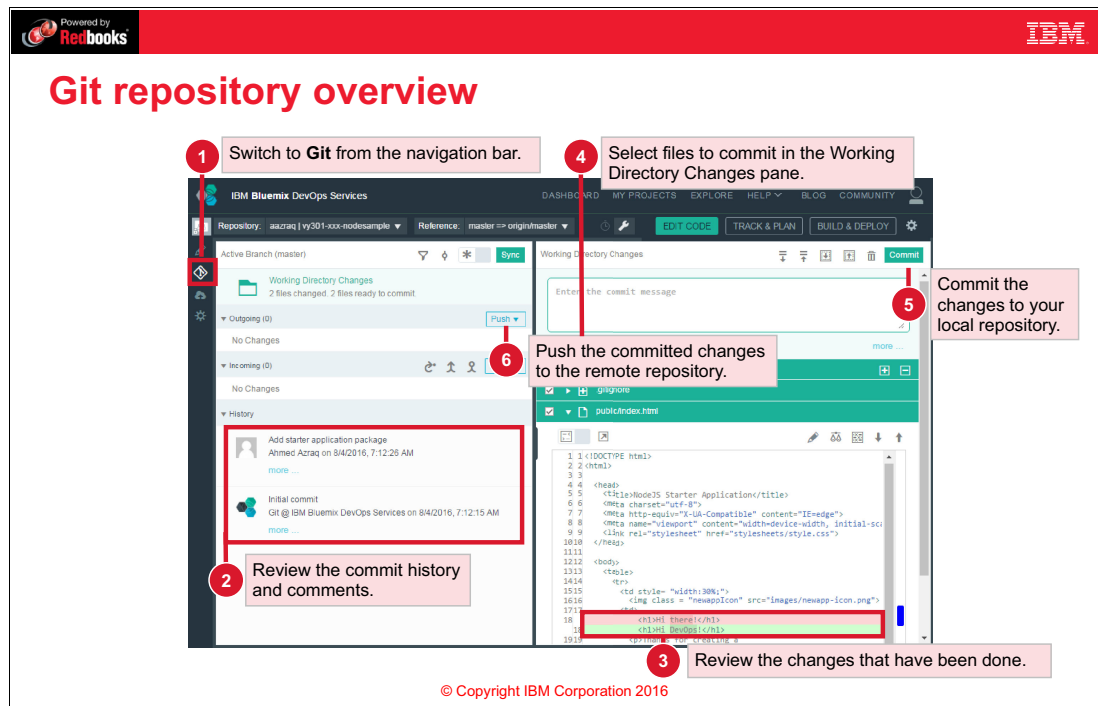


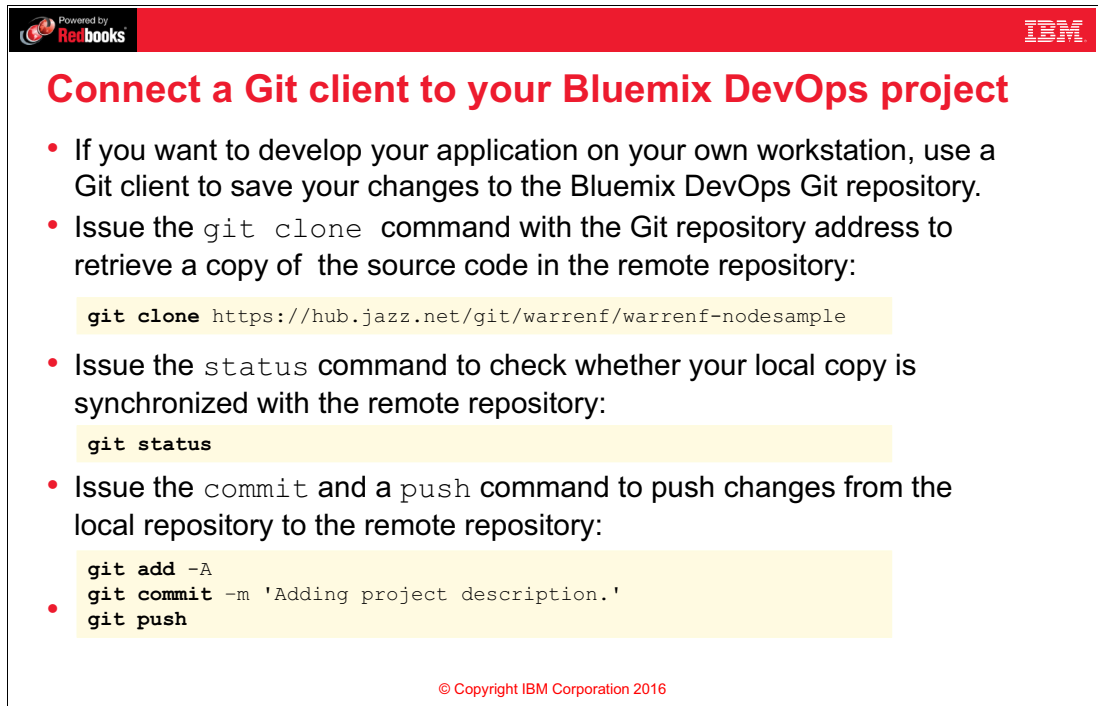
Figure 4-14 Git repository overview

Notes:

The numbers in the figure correspond to these steps:

1. Switch to the Git perspective from the navigation bar on the left side of the web application.
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 introduced by any commit.
3. The Working Directory Changes pane detects any updated files in the Bluemix DevOps user directory. Click any file to review the changes that have been made in that file.
4. Select the files that you want to commit, add a descriptive comment about that change, then click **Commit** to commit the changes to your local repository.
5. 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.
6. If another user updated the files in the remote repository, the Incoming pane lists the updated files.

4.17 Connect a Git client to your Bluemix DevOps project



The screenshot shows a Redbooks article header with the IBM logo. The title is "Connect a Git client to your Bluemix DevOps project". The article contains three bullet points with associated Git commands highlighted in yellow boxes. The first bullet point discusses cloning a repository with the command `git clone https://hub.jazz.net/git/warrenf/warrenf-nodesample`. The second bullet point discusses checking the local copy with `git status`. The third bullet point discusses pushing changes with `git add -A`, `git commit -m 'Adding project description.'`, and `git push`. A copyright notice for IBM Corporation 2016 is at the bottom.

Powered by Redbooks IBM

Connect a Git client to your 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
```

© Copyright IBM Corporation 2016

Figure 4-15 Connect a Git client to your Bluemix DevOps project

Notes:

You can still develop your application on your own local workstation. Start by retrieving a copy of the Bluemix application source code with 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.

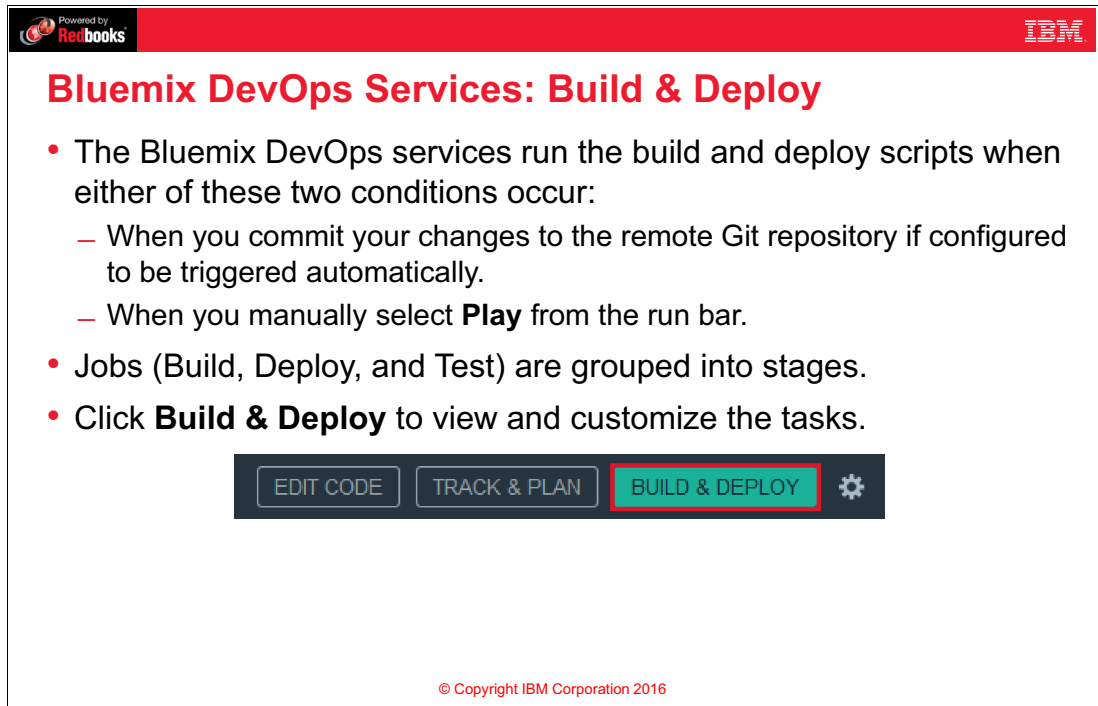
To check whether another developer on your team updated the source code, run the `git status` command.

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

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

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

4.18 Bluemix DevOps Services: Build & Deploy



The screenshot shows a Bluemix DevOps Services interface. At the top left, there is a 'Powered by Redbooks' logo. At the top right, there is an 'IBM' logo. The main heading is 'Bluemix DevOps Services: Build & Deploy'. Below the heading, there is a list of bullet points:

- The Bluemix DevOps services run the build and deploy scripts when either of these two conditions occur:
 - When you commit your changes to the remote Git repository if configured to be triggered automatically.
 - When you manually select **Play** from the run bar.
- Jobs (Build, Deploy, and Test) are grouped into stages.
- Click **Build & Deploy** to view and customize the tasks.

Below the list, there is a dark grey bar with four buttons: 'EDIT CODE', 'TRACK & PLAN', 'BUILD & DEPLOY', and a gear icon. The 'BUILD & DEPLOY' button is highlighted with a red border.

© Copyright IBM Corporation 2016

Figure 4-16 Bluemix DevOps Services: Build & Deploy

Notes:

By default, 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.

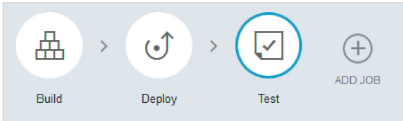
Each job can be either to **Build**, **Deploy**, or **Test** the application. These jobs are grouped into stages.

4.19 Customizing the delivery pipeline

Powered by **Redbooks** **IBM**

Customizing the delivery pipeline

- When you commit changes to the Git repository, Bluemix DevOps pushes out the changes to your 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:
 - Creating multiple stages that deploy code to the testing, staging, and production environments.
 - Add jobs within a stage to run automated tests of your code.
 - Customize the build stage to run a script before deployment.



© Copyright IBM Corporation 2016

Figure 4-17 Customizing the delivery pipeline

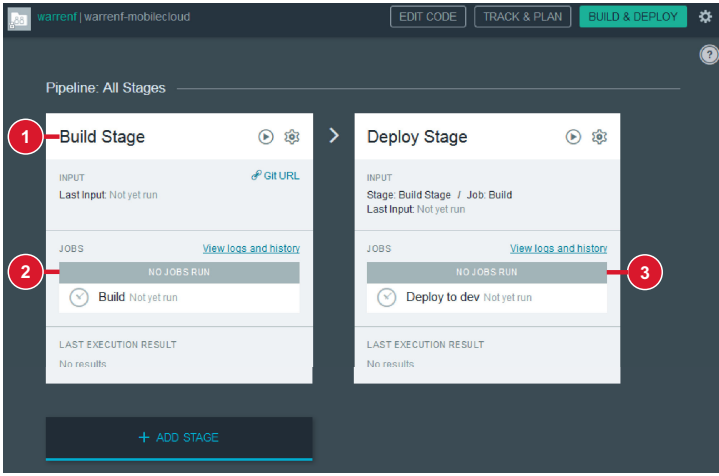
Notes:

When you select **Add Git** from the Bluemix dashboard, the wizard provides you with the **Populate the repository with the starter application package and enable the Build & Deploy pipeline** option. If you do not select this option, Bluemix DevOps does not create any Build & Deploy stages for your project. You will have no jobs defined in your delivery pipeline. In this case, you must manually configure the Build & Deploy jobs.

4.20 Example: Default delivery pipeline

Powered by **Redbooks** **IBM**

Example: Default delivery pipeline



© Copyright IBM Corporation 2016

1. The default delivery pipeline takes the files that are pushed to the remote Git repository.
2. The `manifest.yml` file in your workspace defines the sizing and Cloud Foundry options for your app.
3. The `deploy` operation deploys the built application to your Bluemix space.

Figure 4-18 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 Bluemix to build your code in the server runtime. For example, 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. Bluemix DevOps deploys the built files to your IBM Bluemix space.

4.21 Example: Customizing the build stage

The screenshot displays the 'Stage Configuration' window for a 'Build Stage'. It features three tabs: 'INPUT', 'JOBS', and 'ENVIRONMENT PROPERTIES'. The 'INPUT' tab is active, showing 'Input Settings'. Under 'Input Type', 'SCM Repository' is selected. The 'Git URL' field contains 'https://hub.jazz.net/git/warrenf/warrenf-mobilecloud', and the 'Branch' dropdown is set to 'master'. The 'Stage Trigger' section has two radio buttons: 'Automatically execute jobs when a change is pushed to Git' (selected) and 'Only execute jobs when a user manually runs this stage'. A 'DELETE' button is in the top right, and 'SAVE' and 'CANCEL' buttons are at the bottom. Two callout boxes are present: one labeled '1' points to the Git URL field with the text 'By default, Bluemix DevOps runs the Build stage when a client pushes any change to the master branch in the remote Git repository.'; another labeled '2' points to the 'Only execute jobs when a user manually runs this stage' radio button with the text 'It is not necessary for you to run the build stage manually. However, you can disable automatic builds with the **state trigger** setting.'

© Copyright IBM Corporation 2016

Figure 4-19 Example: Customizing the build stage

Notes:

By default, 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 Bluemix account, change the **State trigger** setting to **Only execute jobs when a user manually runs this stage** in the build stage.

4.22 Example: Configuring build jobs

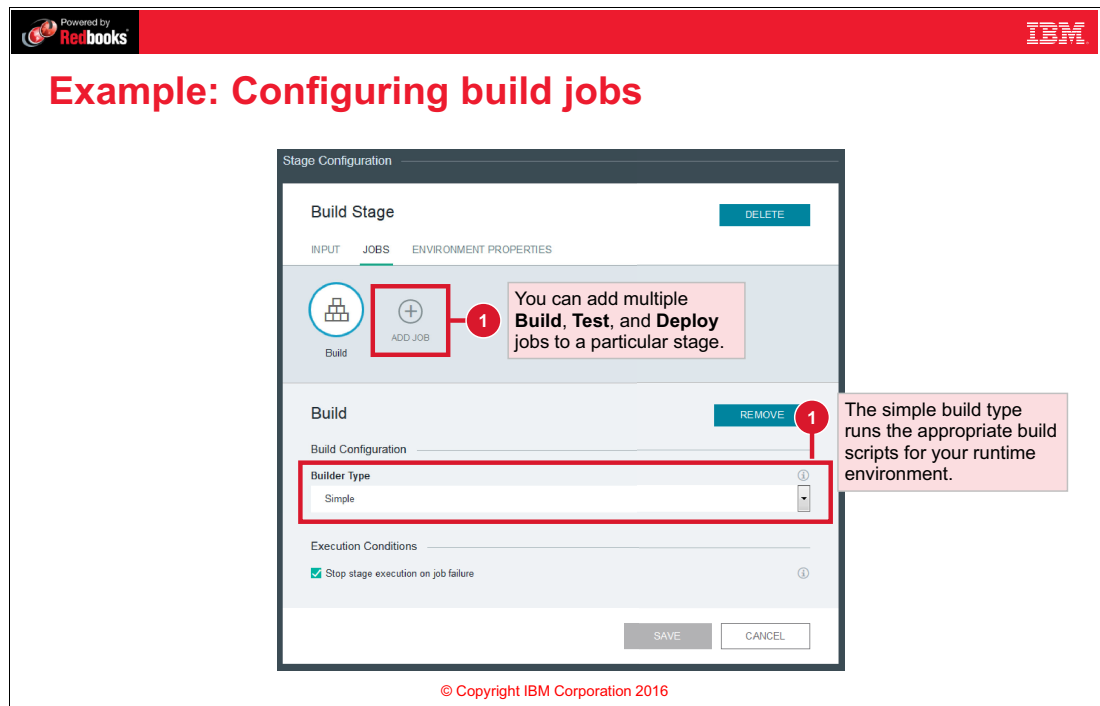


Figure 4-20 Example: Configuring build jobs

Notes:

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

Builder Type currently supports these 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 run time resolves modules that your application requires.

4.23 Example: Configuring deploy job

The screenshot shows the 'Deploy to dev' configuration page in the IBM Bluemix console. The page is titled 'Deploy to dev' and has a 'REMOVE' button in the top right corner. The configuration is divided into several sections:

- Deployer Type:** Set to 'Cloud Foundry'. Callout 1: 'The three deployer types are **Cloud Foundry**, **Active Deploy**, and **IBM Container Service on Bluemix**.'
- Target:** Set to 'IBM Bluemix US South - https://api.ng.bluemix.net'. Callout 2: 'Specify a Cloud Foundry provider as the target like IBM Bluemix.'
- Organization:** Set to 'vy301-xxx'. Callout 3: 'You can clone this deploy job and specify other environments in your account.'
- Space:** Set to 'dev'. Callout 3: 'You can clone this deploy job and specify other environments in your account.'
- Application Name:** Set to 'vy301-xxx-nodesample'.
- Deploy Script:** Contains a shell script:

```
#!/bin/bash
cf push "${CF_APP}"
# View logs
#cf logs "${CF_APP}" --recent
```

 Callout 4: 'You can customize the exact Cloud Foundry command line interface commands in the deploy process in the **Deploy Script** section.'

© Copyright IBM Corporation 2016

Figure 4-21 Example: Configuring deploy job

Notes:

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

There are three Deployer Types:

- ▶ **Cloud Foundry:** Deploys applications to Cloud Foundry servers, such as IBM Bluemix. This is the default deployer type for the deploy job.
- ▶ **Active Deploy:** You can update running apps with zero downtime when you use the IBM Active Deploy service in your pipeline. The Active Deploy - Begin job contains a script that starts the deployment process to increase instances of your new app until both versions of your app are live in production. Active Deploy - Complete ends that deployment process and decreases the original version of the app 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 using the Docker Builder.

In this example, your application is pushed to the dev 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 commands in the deploy process in the Deploy Script section. The default deploy action is equivalent to running `cf push` from the command-line interface. You can add custom shell script commands as well. `${CF_APP}` in the figure refers to the application name.

4.24 Example: A successful build and deploy result

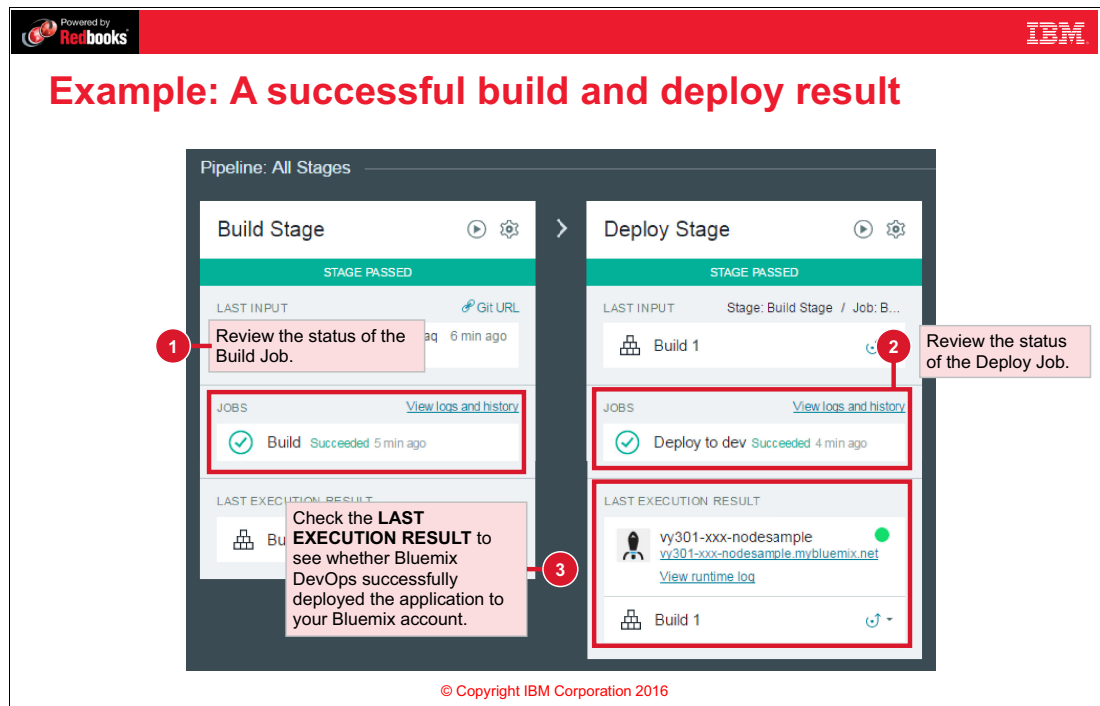


Figure 4-22 Example: A successful build and deploy result

Notes:

The default settings for the delivery pipeline are two 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 the build artifacts produced from the Build Stage as input.
 - Jobs: Deploys the application to IBM Bluemix.

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

The LAST EXECUTION RESULT pane shows that Bluemix DevOps successfully deployed the application to your Bluemix account, and shows that the application is currently running on 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 on top of Deploy Stage, then selecting **Clone Stage**.

4.25 Track & Plan Tools

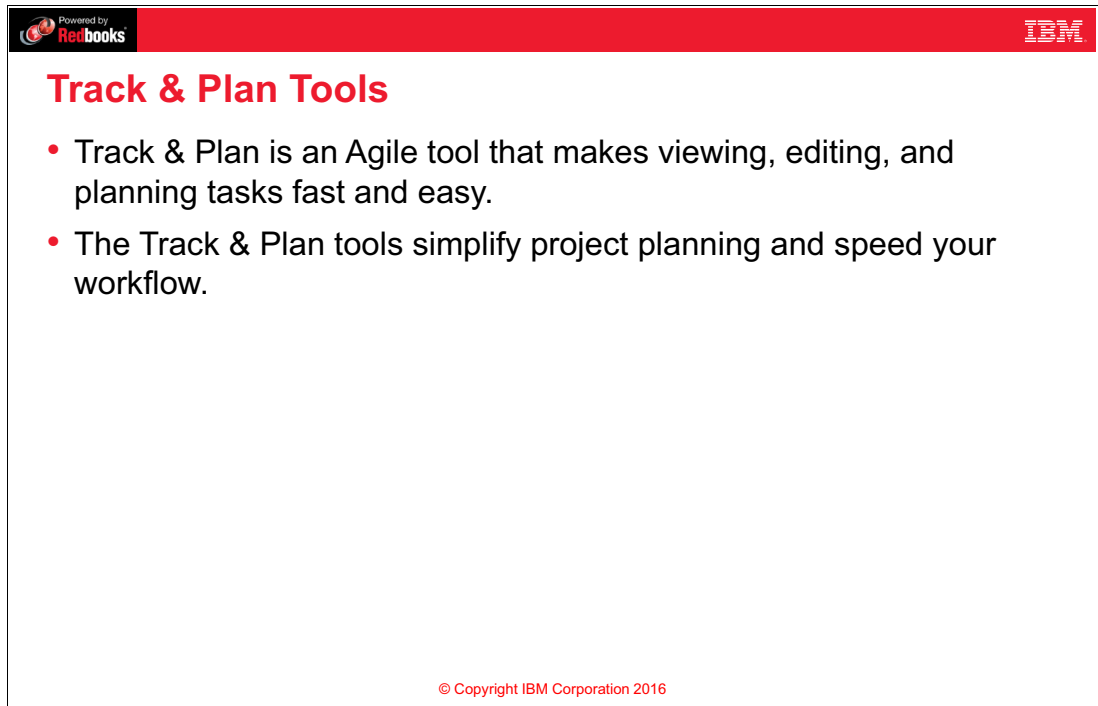


Figure 4-23 Track & Plan Tools

Notes:

Agile software development values:¹

- ▶ Individuals and interactions over processes and tools.
- ▶ Working software over comprehensive documentation.
- ▶ Customer collaboration over contract negotiation.
- ▶ Responding to change over following a plan.

IBM Bluemix DevOps supports the agile development ideology. Track & Plan is one of the features of Bluemix DevOps Services. It is an agile tool that makes viewing, editing, and planning tasks fast and easy. It provides these features:

- ▶ Track work for you and your team
- ▶ Quickly create work items of different types including Defect, Task, and Story.
- ▶ Check what is incoming to your project
- ▶ Groom your backlog
- ▶ Plan work for future releases and sprints

You can manage any type of project with an agile approach by using the Track & Plan feature in Bluemix DevOps Services. The Track & Plan tools simplify project planning and speed your workflow.

¹ <http://agilemanifesto.org/>

4.26 Track & Plan Tools capabilities



Figure 4-24 Track & Plan Tools capabilities

Notes:

The Track & Plan Tools allow the project members to perform these tasks:

► Create Work Items

- In Agile programming, an Epic captures a big chunk of functionality that is decomposed later into Stories. Each Story has a group of assigned tasks, and each task is assigned to a user. You can model that using the Track & Plan Tools in Bluemix DevOps. You can create a work item of type **Epic** and compose it into sub work items of type **Story**. For each created story, you can create list of relevant work items of type **Task** or **Defect**.
- You can assign work items to any member of the team by using the **Owned By** attribute.
- You can set the priority of the work item according to urgency to **Low, Medium, or High**.
- You can subscribe to any work item or manage the **Subscribers**.

► View Incoming Work

- By default the new work items are added into INCOMING WORK view. You can triage the needed work items to the backlog.

► View and Manage the Backlog

- This view contains all the work items that have been triaged from the incoming work but have not been assigned to any sprint yet.
- Work Items in the backlog are ranked automatically based on **Priority**. You have the option to change the ranking manually.

► View and Manage Sprint Planning

- It allows the members to add a Sprint, add new work items to a Sprint, and change the assign any work item from the backlog to the Sprint.
- ▶ **View Team's Work**
 - View the status of the work items assigned to the current sprint (**New, In Progress, Implemented, Done (Completed Testing), Deferred, or Rejected (Invalid)**).
- ▶ **View All Work**
 - View the status of all work items.

4.27 Enable Track & Plan

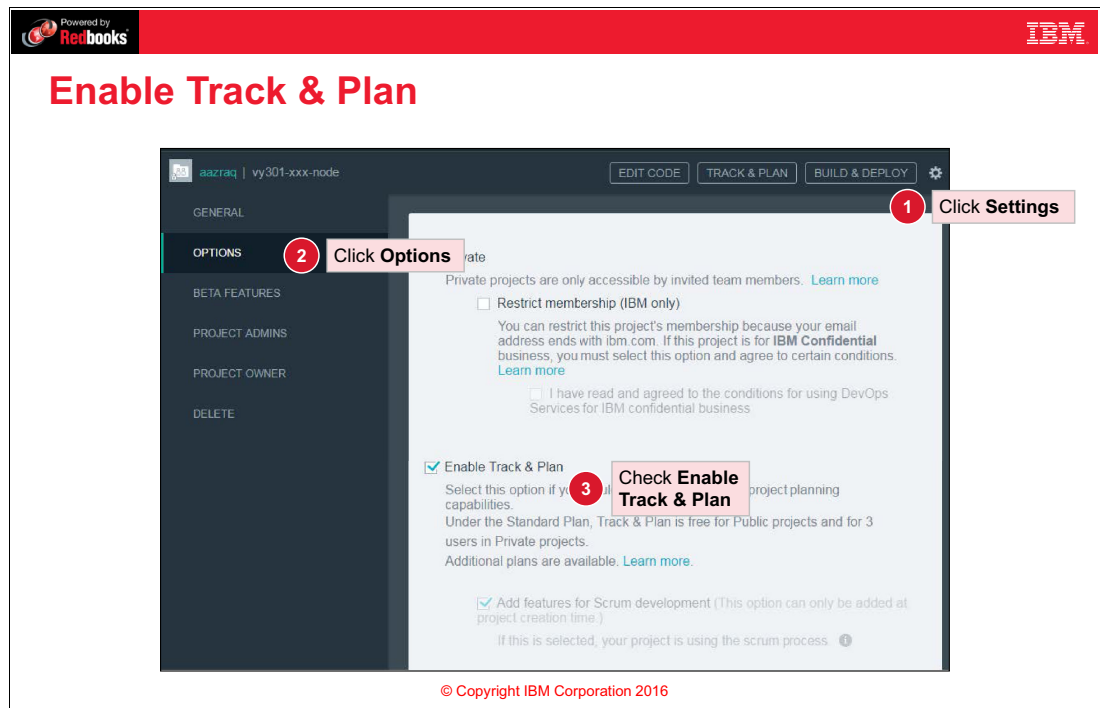


Figure 4-25 Enable Track & Plan

Notes:

Track & Plan is not enabled by default for the projects on Bluemix DevOps Services. To enable Track & Plan, complete these steps:

1. Open **Settings** by clicking the wheel icon on the upper right.
2. Click **OPTIONS** on the left navigation bar.
3. Select **Enable Track & Plan**.
4. Click **SAVE**.

4.28 Track & Plan: Example

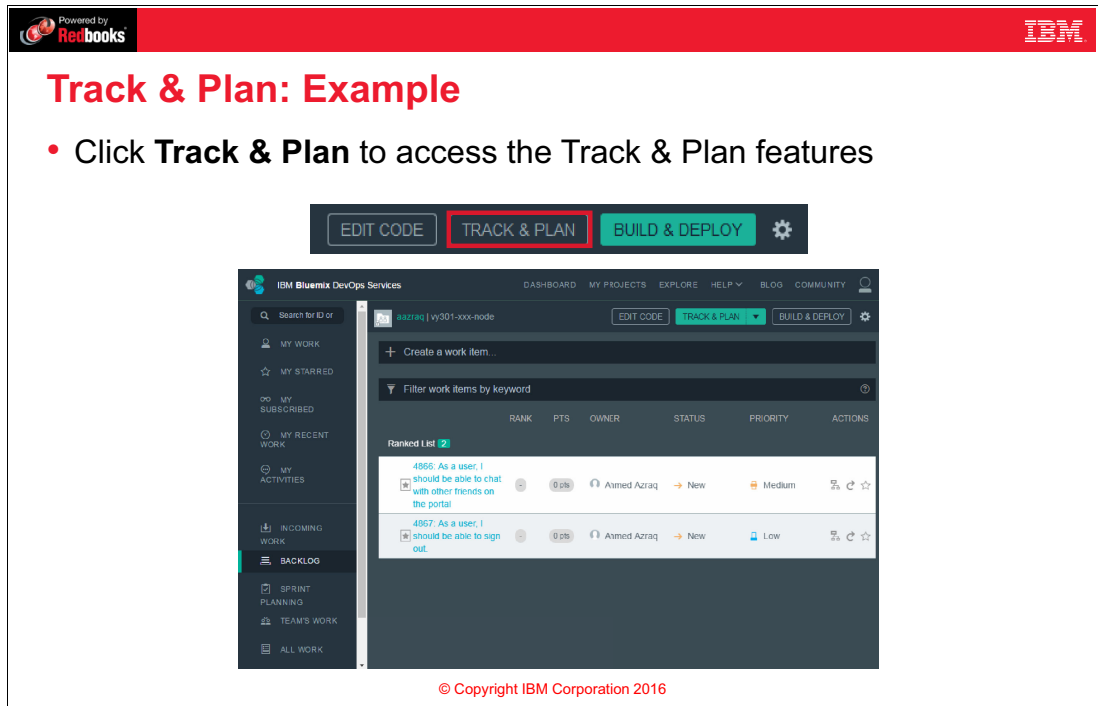


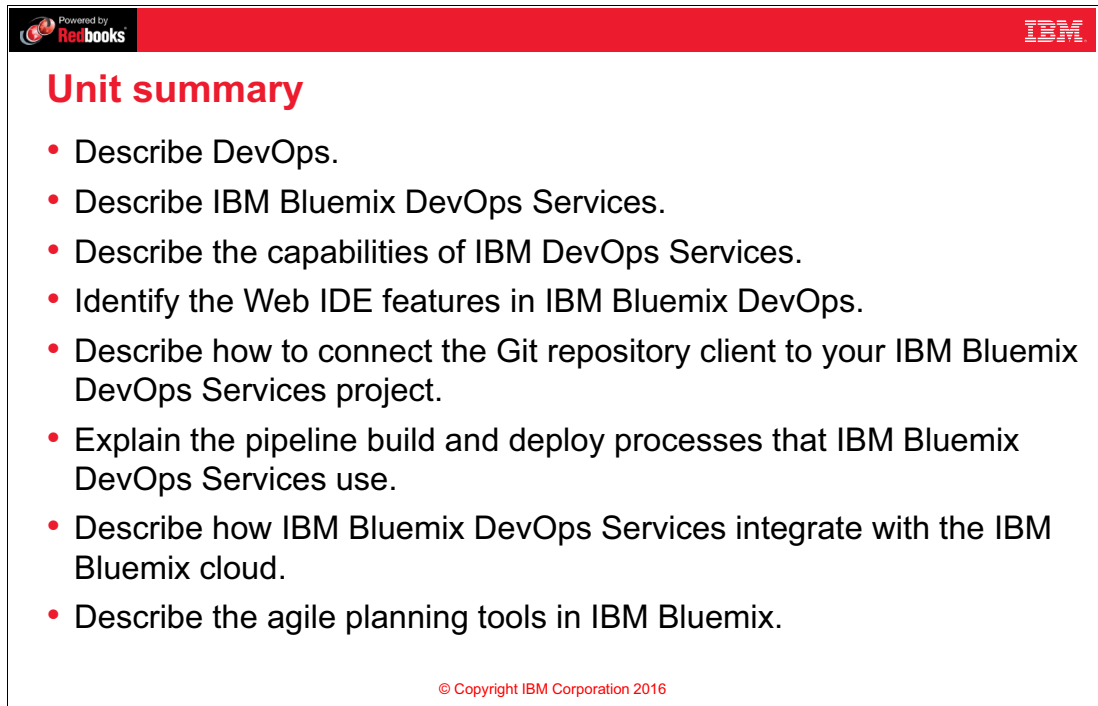
Figure 4-26 Track & Plan: Example

Notes:

To access the Track & Plan features, click **Track & Plan** on the navigation bar shown here in Bluemix DevOps.

This figure shows an example of a backlog of two stories. The first one has a **Medium** priority and the second one has a **Low** priority. Notice that the story with a higher priority is automatically sorted to the top of the list.

4.29 Unit summary



Powered by **Redbooks** **IBM**


Unit summary

- 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.

© Copyright IBM Corporation 2016

Figure 4-27 Unit summary

4.30 Exercise 3 objectives



Exercise 3 objectives

After completing this exercise, you should be able to perform these tasks:

- Sign in to DevOps.
- Explore public projects in DevOps.
- Sign in to Bluemix and DevOps.
- Create a Git repository in DevOps services to manage your source code.
- View and edit code in DevOps.
- Build and deploy code from DevOps to Bluemix.
- Test the application in Bluemix.

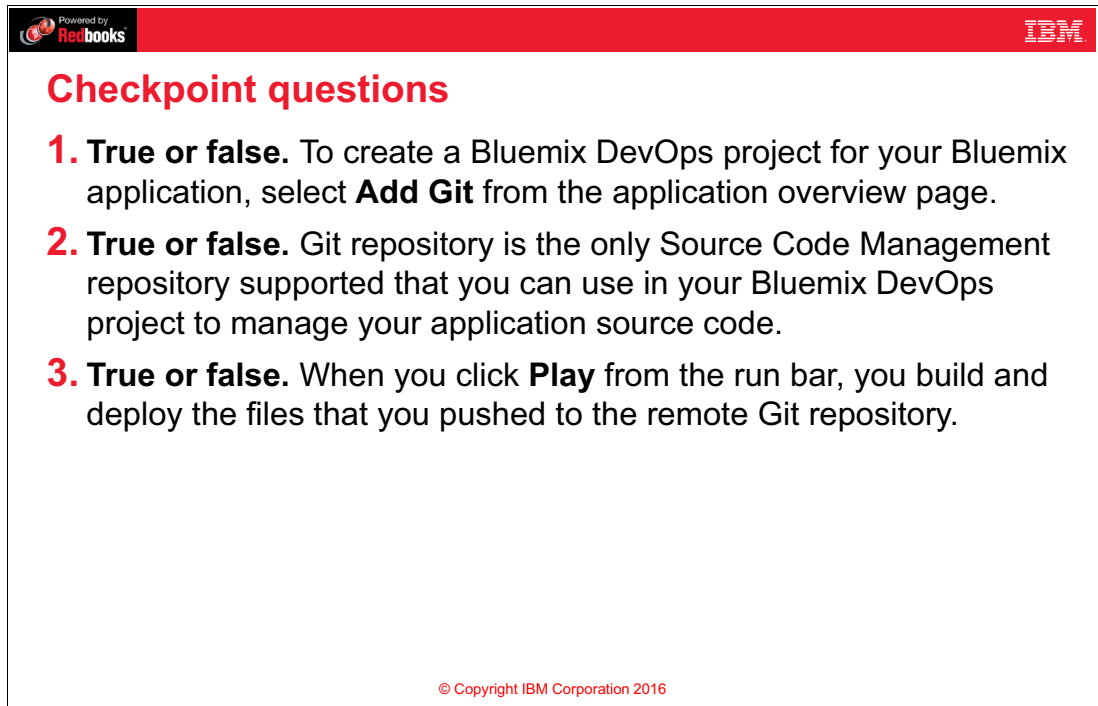
© Copyright IBM Corporation 2016

Figure 4-28 Exercise 3 objectives

Notes:

(none)

4.31 Checkpoint questions



Powered by **Redbooks** **IBM**

Checkpoint questions

- 1. True or false.** To create a Bluemix DevOps project for your Bluemix application, select **Add Git** from the application overview page.
- 2. True or false.** Git repository is the only Source Code Management repository supported that you can use in your 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.

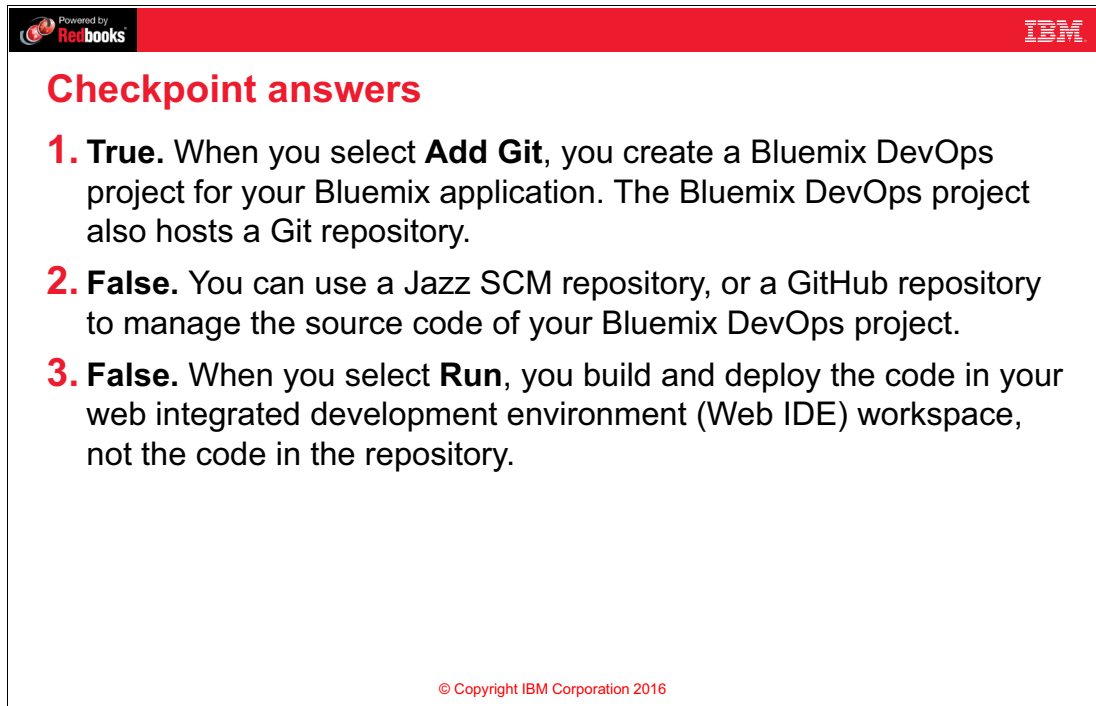
© Copyright IBM Corporation 2016

Figure 4-29 Checkpoint questions

Notes:

(none)

4.32 Checkpoint answers



The screenshot shows a document header with 'Powered by Redbooks' on the left and the 'IBM' logo on the right. The main content is titled 'Checkpoint answers' in red. It contains three numbered items:

- 1. True.** When you select **Add Git**, you create a Bluemix DevOps project for your Bluemix application. The Bluemix DevOps project also hosts a Git repository.
- 2. False.** You can use a Jazz SCM repository, 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.

At the bottom right of the document, there is a copyright notice: © Copyright IBM Corporation 2016.

Figure 4-30 Checkpoint questions

Notes:

(none)



REST architecture

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.
- ▶ Example of REST APIs using Watson.

5.2 References

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

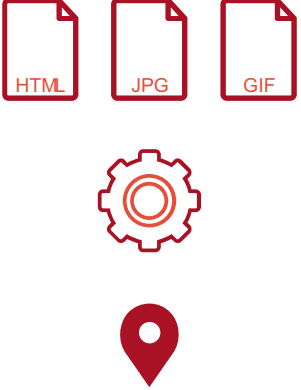
- ▶ REST APIs
 - ECMA-262 ECMAScript language standard, third edition
<http://www.ecma-international.org/publications/standards/Ecma-262.htm>
 - RESTful Web services: The basics at IBM developerWorks
<https://www.ibm.com/developerworks/library/ws-restful/>
- ▶ Watch the following videos to learn more about Watson:
 - IBM Watson is the platform for cognitive business
<https://www.youtube.com/watch?v=6SNs9kvRWSA>
 - 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=gJEzuYynaiw>
 - 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.



© Copyright IBM Corporation 2016

Figure 5-1 What is REST?

Notes:

REST stands for Representational State Transfer. It is an architecture style for building resources on the World Wide Web.

For a website, HTML documents, images, and script files are all examples of web resources.

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, or URI, to describe the network location of the resource.

REST provides these 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 containing the parameters needed for the request are sent in the URL by placing “?” at the end of the URI, then writing the parameters.

Each parameter is represented as a name-value pair. The parameters are separated by an ampersand (&). For example, the URI for a GET request can be like `http://example.com/personDetail?firstName=Ahmed&age=28` or can be like `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

Powered by Redbooks IBM

Applying REST to server-side applications

- 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.

© Copyright IBM Corporation 2016

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

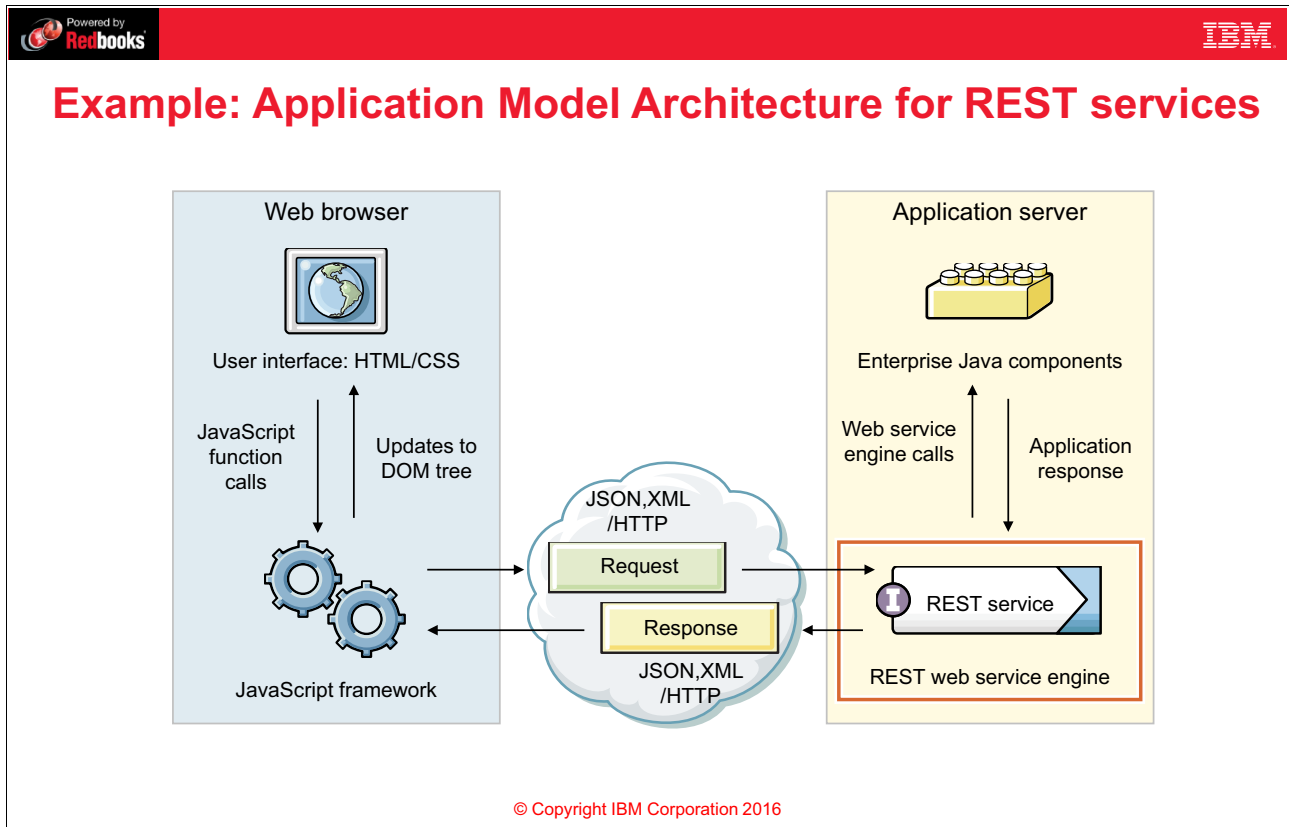


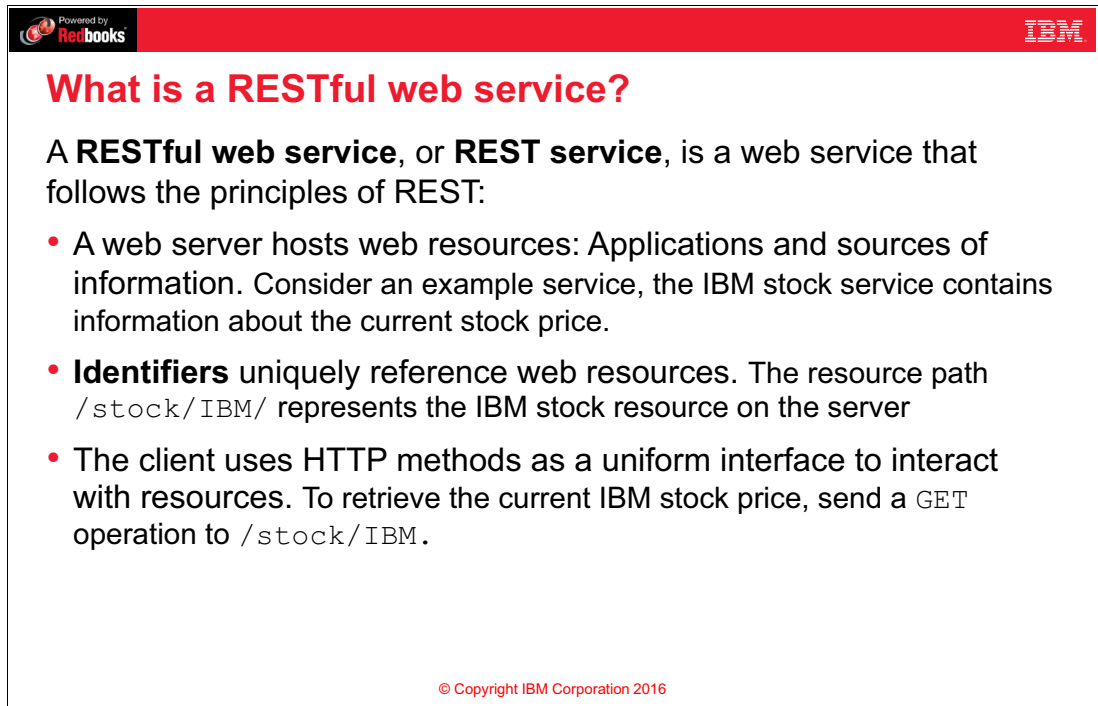
Figure 5-3 Example: Application model architecture for REST services

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 using one of the HTTP methods. The request and the response can be either JSON or XML over HTTP Protocol.

You do not have to code the server side in Java and the client in JavaScript. There are plenty of people who 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`.

© Copyright IBM Corporation 2016

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: applications and sources of information, such as 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 the example, to retrieve the current IBM stock price, send a `GET` operation on `/stock/IBM`.

5.7 Example: Sending an HTTP request to a REST service

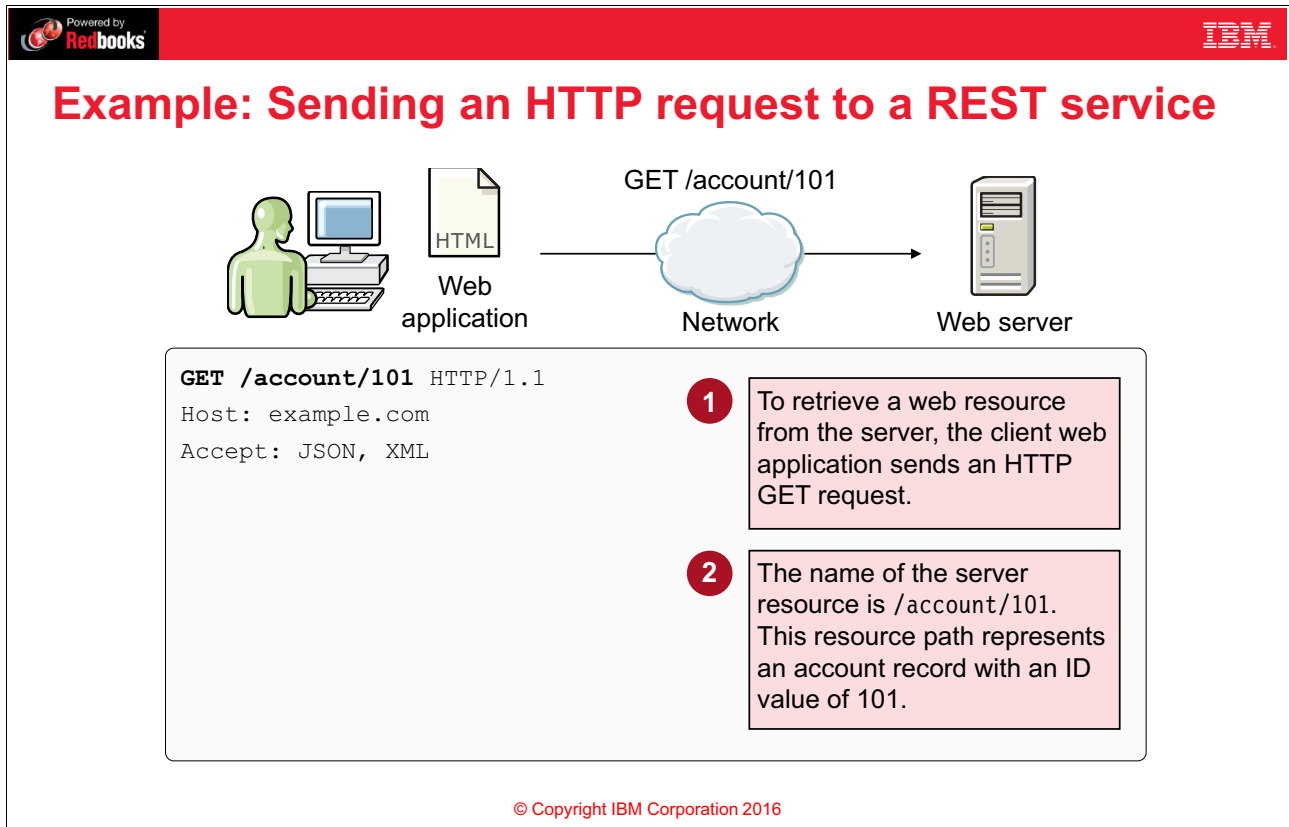


Figure 5-5 Example: Sending an HTTP request to a REST service

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 exactly the same as making a request for a web page by using an HTTP GET request. When you navigate to a URL on 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

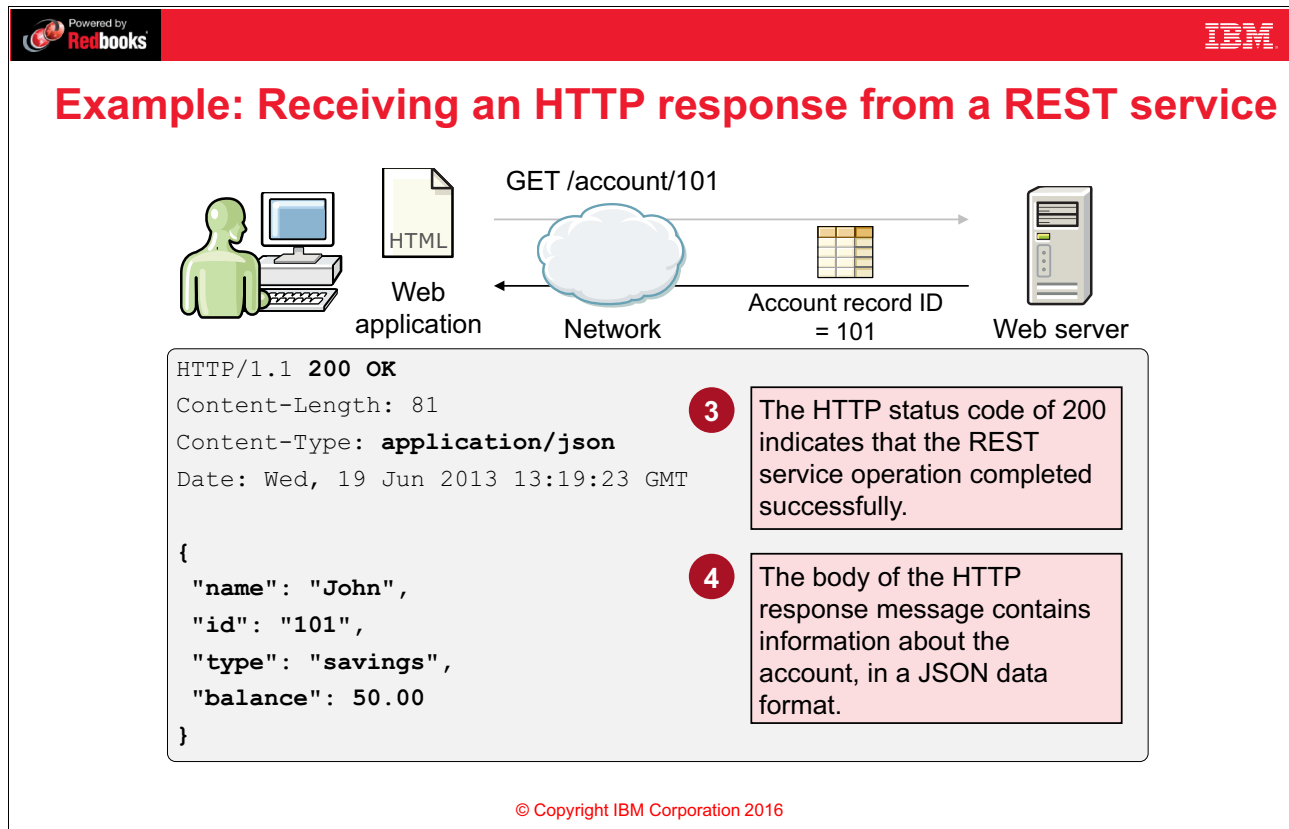


Figure 5-6 Example: Receiving an HTTP response from a REST service

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, 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 which in this example is 81 characters.

Content-Type describes the data type of the response which in this example is JSON.


Response Body is a JSON object that contains four name-value pairs, containing 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.



© Copyright IBM Corporation 2016

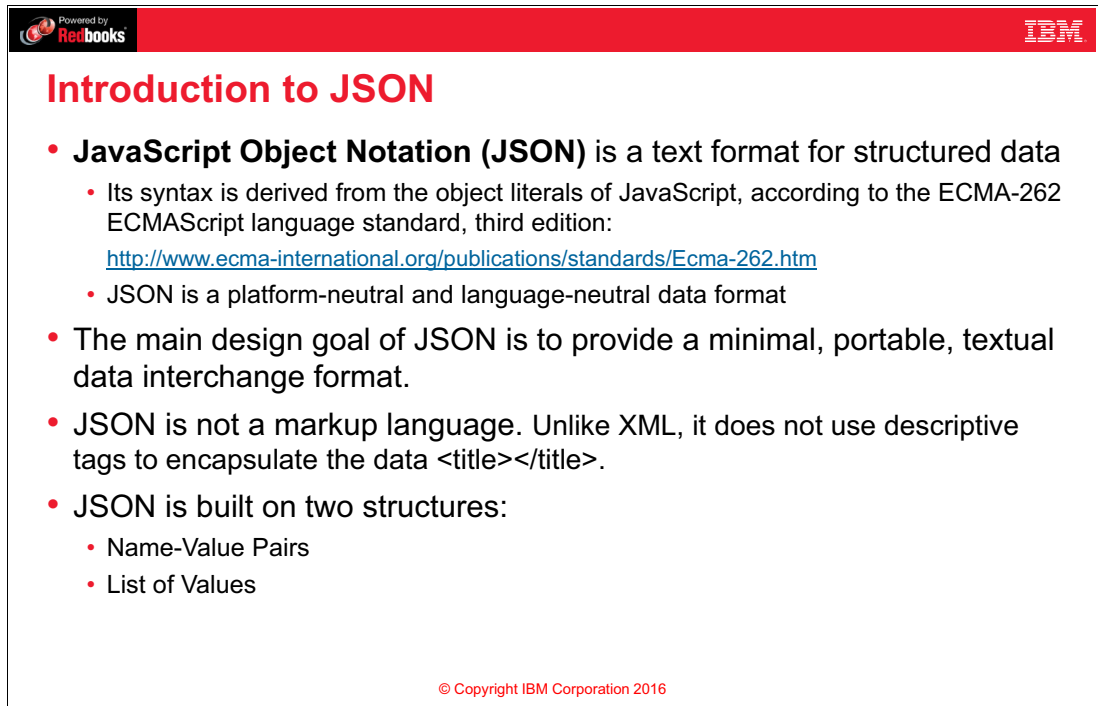
Figure 5-7 REST characteristics

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 is not entirely true. Although the most common scenarios for using REST is over the HTTP protocol, REST can be used over other transfer protocols like 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



The image is a screenshot of a presentation slide. At the top left, there is a logo for 'Powered by Redbooks'. At the top right, there is the 'IBM' logo. The main title of the slide is 'Introduction to JSON' in a large, bold, red font. Below the title, there is a bulleted list of points. The first point states that JavaScript Object Notation (JSON) is a text format for structured data, with a sub-bullet explaining its syntax is derived from JavaScript object literals according to the ECMA-262 standard, and providing a URL. The second point states JSON is a platform-neutral and language-neutral data format. The third point states the main design goal is to provide a minimal, portable, textual data interchange format. The fourth point states JSON is not a markup language, unlike XML, and does not use descriptive tags. The fifth point states JSON is built on two structures: Name-Value Pairs and List of Values. At the bottom right of the slide, there is a small copyright notice: '© Copyright IBM Corporation 2016'.

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 like `<title></title>` to declare the title of the page. Whereas 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

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

© Copyright IBM Corporation 2016

Figure 5-9 JSON data types

Notes:

JSON has these 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 have 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. That 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 will receive 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 either 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, null. JSON does not support the JavaScript keyword `undefined`. Either use null or another set value to represent an undefined value.

5.12 JSON data type: Objects

Powered by **Redbooks** **IBM**

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"
}
```

© Copyright IBM Corporation 2016

Figure 5-10 JSON data type: Objects

Notes:

JSON Object is an unordered collection of key/value pairs with these 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 the 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

Powered by **Redbooks** **IBM**

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]  
]
```

© Copyright IBM Corporation 2016

Figure 5-11 JSON data type: Arrays

Notes:

JSON Array is an ordered sequence of values with these characteristics:

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

The figure shows two examples. 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 either an object or an array at the top level.

5.14 What is Watson?

Powered by **Redbooks** **IBM**

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 80% of all data today is unstructured.*



*Source: <https://www.ibm.com/watson/what-is-watson.html>
© Copyright IBM Corporation 2016

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 analyses unstructured data. 80% of all data today is unstructured. Unstructured means not structured in machine-readable format. Unstructured data includes news articles, research reports, social media posts, and enterprise system data.

5.15 Watson Services in Bluemix

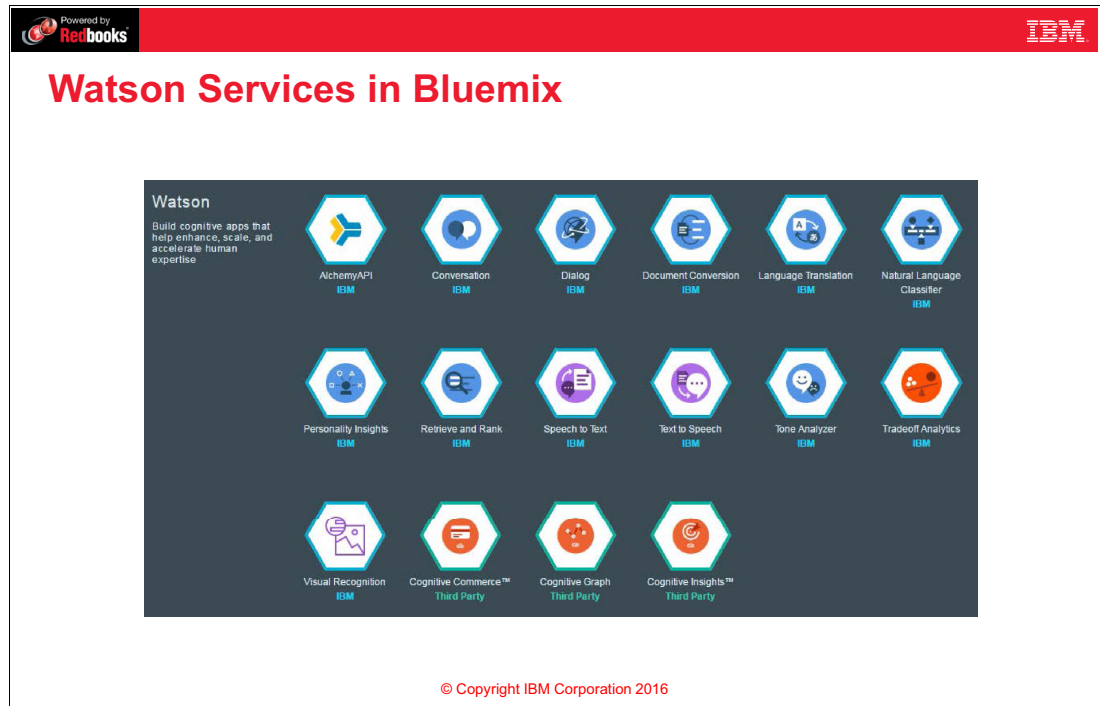


Figure 5-13 Watson Services in Bluemix

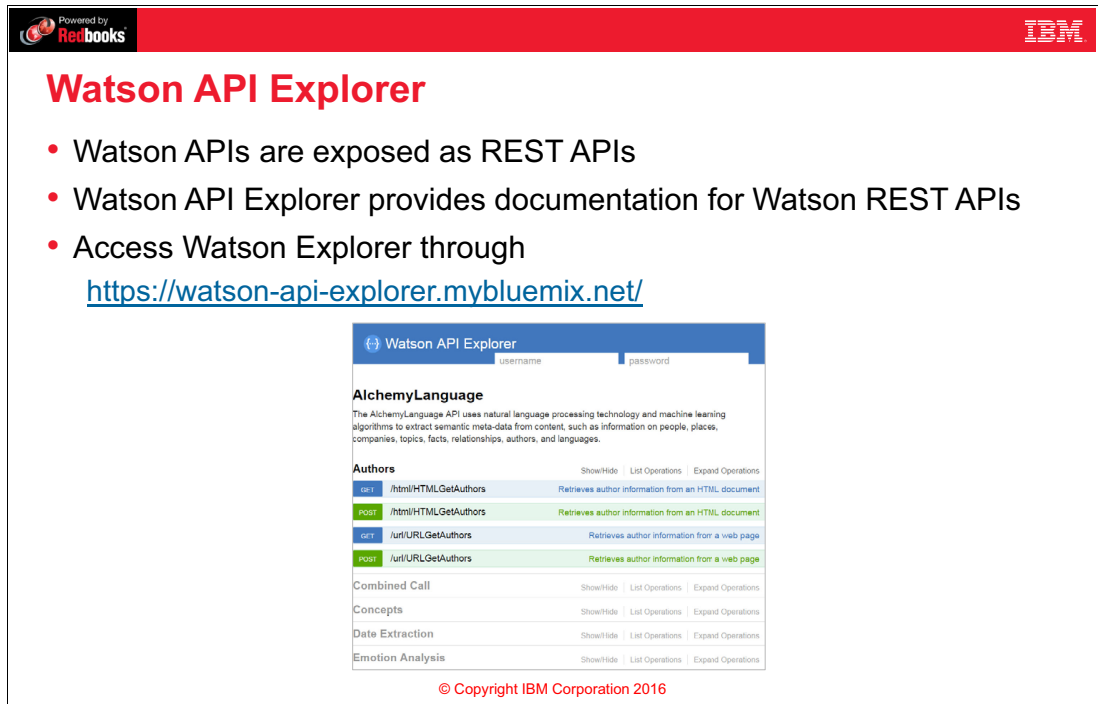
Notes:

The Bluemix Catalog contains several Watson Services, and new services are introduced periodically.

Watson Services include these applications:

- ▶ **AlchemyAPI®:** This app analyzes unstructured text and image content. You can use it to extract semantic metadata from content, such as information on people, places, companies, topics, facts, relationships, authors, and languages.
- ▶ **Language Translation:** Translates text from one language to another.
- ▶ **Personality Insights:** Personality Insights extracts and analyses a spectrum of personality attributes to help discover actionable insights about people and entities, and in return guides end users to highly personalized interactions. It takes a text as an input, and outputs the analysis of the personality according to the input text. You can try a demo of this service at the following link:
<https://personality-insights-livedemo.mybluemix.net/>
- ▶ **Speech to text:** This app converts the human voice into written words. It currently supports English, Japanese, Arabic, Mandarin, Portuguese, Spanish, and French.

5.16 Watson API Explorer



The screenshot shows the Watson API Explorer interface. At the top, there is a red header with the "Powered by Redbooks" logo on the left and the "IBM" logo on the right. Below the header, the title "Watson API Explorer" is displayed in red. A list of bullet points follows: "Watson APIs are exposed as REST APIs", "Watson API Explorer provides documentation for Watson REST APIs", and "Access Watson Explorer through <https://watson-api-explorer.mybluemix.net/>". Below the text is a screenshot of the API Explorer interface. The interface has a blue header with the title "Watson API Explorer" and input fields for "username" and "password". The main content area is titled "AlchemyLanguage" and includes a description: "The AlchemyLanguage API uses natural language processing technology and machine learning algorithms to extract semantic meta-data from content, such as information on people, places, companies, topics, facts, relationships, authors, and languages." Below this, there is a section for "Authors" with a "Show/Hide" button, "List Operations", and "Expand Operations" links. The "Authors" section lists four endpoints: a GET endpoint for "/html/HTMLGetAuthors" (Retrieves author information from an HTML document), a POST endpoint for "/html/HTMLGetAuthors" (Retrieves author information from an HTML document), a GET endpoint for "/url/URLGetAuthors" (Retrieves author information from a web page), and a POST endpoint for "/url/URLGetAuthors" (Retrieves author information from a web page). Below the "Authors" section, there are sections for "Combined Call", "Concepts", "Date Extraction", and "Emotion Analysis", each with "Show/Hide", "List Operations", and "Expand Operations" links. At the bottom of the screenshot, there is a copyright notice: "© Copyright IBM Corporation 2016".

Figure 5-14 Watson API Explorer

Notes:

Watson APIs are exposed as REST APIs. You can interact with each API 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, and 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. You can access Watson API Explorer through this link:

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

5.17 Example: Watson API Explorer - AlchemyLanguage (Authors)

The screenshot shows the Watson API Explorer interface for the AlchemyLanguage (Authors) service. The interface is divided into several sections:

- GET /url/URLGetAuthors**: Retrieves author information from a web page.
- Implementation Notes**: The URLGetAuthors call is utilized to extract author information contained within the specified web page. AlchemyLanguage will download the requested URL, process the retrieved HTML document, looking for author information.
- Response Class (Status 200)**: Successful request.
- Model**: Model Schema.
- Authors Response { }**:
 - `status` (string, optional): success / failure status indicating whether the request was processed.
 - `url` (string, optional): http url information was requested for.
 - `authors` (Array[string], optional)
- Parameters**:

Parameter	Value	Description	Parameter Type	Data Type
apikey	<input type="text"/>	Your API key	query	string
url	http://www.reuters.com	Public facing URL	query	string
outputMode	json	Desired response format (default XML)	query	string
jsonp	<input type="text"/>	JSONP callback (requires outputMode to be set to json)	query	string
- Response Body**:

```
{
  "status": "OK",
  "usage": "By accessing AlchemyAPI or using information generated by AlchemyAPI, y",
  "url": "http://www.reuters.com/article/us-africa-japan-idUSKCN112077",
  "authors": {
    "names": [
      "George Obulutsa"
    ]
  }
}
```

© Copyright IBM Corporation 2016

Figure 5-15 Example: Watson API Explorer - AlchemyLanguage (Authors)

Notes:

In this example, the Get Author REST API is called from AlchemyLanguage Service to get the author of an article in Reuters.

The Resource Path, which is the path of the exposed REST resource on the server, is `/url/URLGetAuthors`. The HTTP Verb used is GET, which is used to retrieve the author information of a particular URL. The example uses the following parameters:

- ▶ `apikey`: An API Key is required to call most of the Alchemy API services. You get this API key once you add AlchemyAPI service from Bluemix Catalog. However, it's optional for Get Author.
- ▶ `url`: The URL of the article that you would like to get the author name for.
- ▶ `outputMode`: Desired response format (default XML). In this example, `json`.
- ▶ `jsonp`: This is a JSON extension that is allowed to cross domains more easily. JSONP response can be passed as an argument to a callback function.

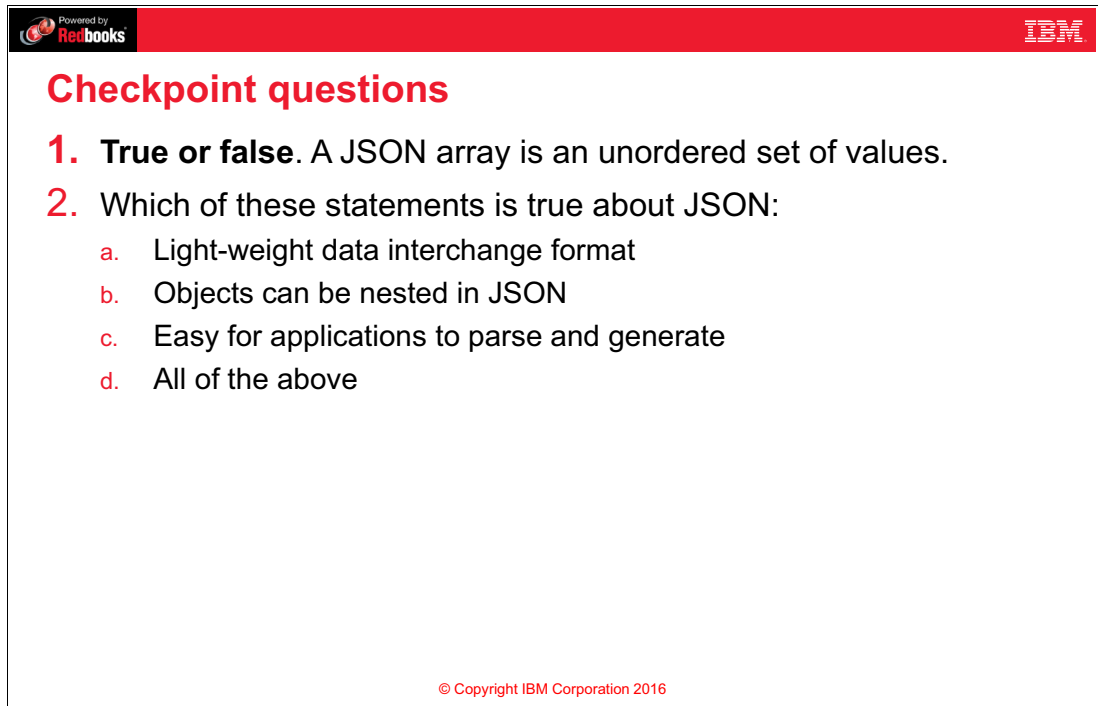
After filling the parameters, click **Try it out**. Watson API Explorer will build the Request URI, as shown below:

```
https://watson-api-explorer.mybluemix.net/alchemy-api/calls/url/URLGetAuthors?url=http%3A%2F%2Fwww.reuters.com%2Farticle%2Fus-africa-japan-idUSKCN112077&outputMode=json
```

The service URL is <https://watson-api-explorer.mybluemix.net/alchemy-api/calls>, the resource path is `/url/URLGetAuthors`, and the parameters passed are `url` and `outputMode`. Because it is a GET method, you can try it from your browser directly as well.

The JSON response is a JSON object with a collection of name-value pairs with keys `status`, `usage`, `url`, and `authors`. `Authors` is another nested JSON object that contains the key names which contains a JSON array. This JSON array contains the names of the authors of the article.

5.18 Checkpoint questions



Powered by **Redbooks** **IBM**

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

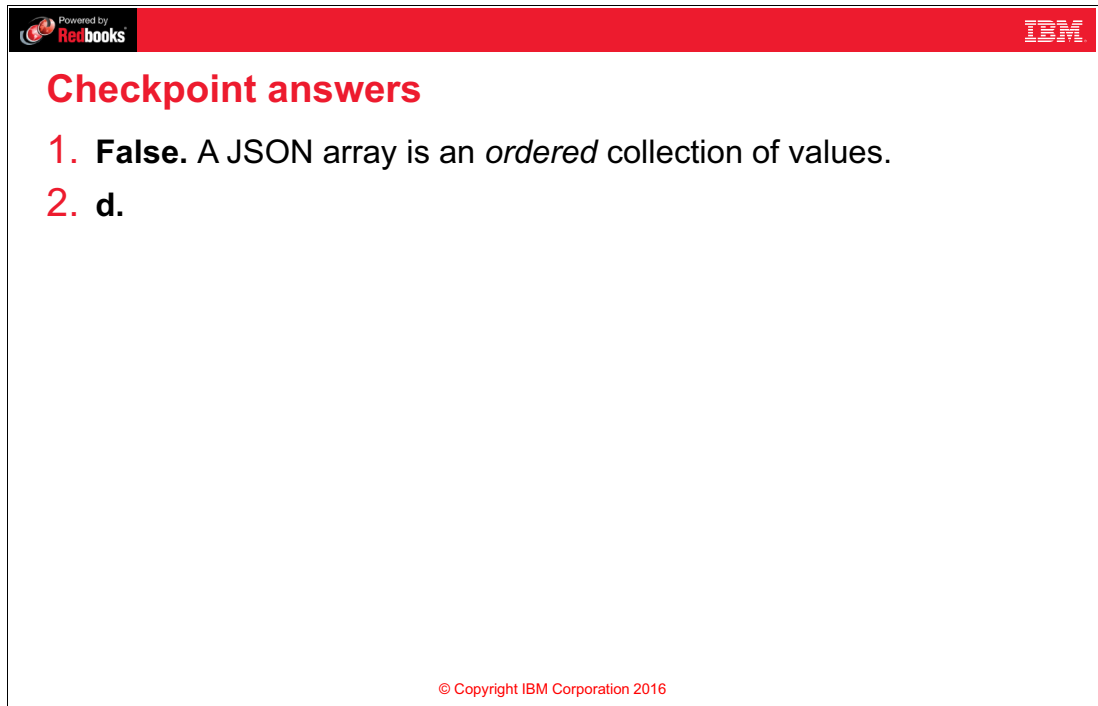
© Copyright IBM Corporation 2016

Figure 5-16 Checkpoint questions

Notes:

(none)

5.19 Checkpoint answers



Powered by Redbooks

IBM

Checkpoint answers

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

© Copyright IBM Corporation 2016

Figure 5-17 Checkpoint answers

Notes:

1. **False.** A JSON array is an ordered collection of values.
2. **d.** JSON is a light-weight data interchange format that is easy for applications to parse and generate, and JSON objects can be nested.



Introduction to data services in IBM Bluemix

This unit covers the following topics:

- ▶ Database services in Bluemix
- ▶ IBM Cloudant®

6.1 What you should be able to do

After completion of 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

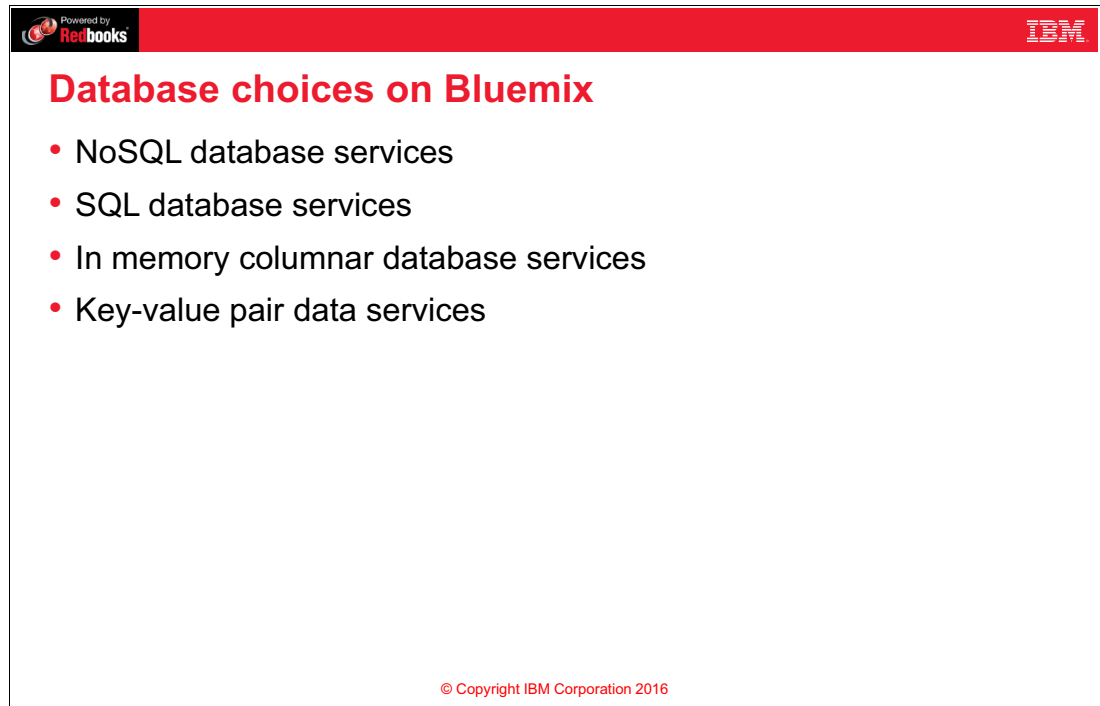


Figure 6-1 Database choices on Bluemix

Notes:

Bluemix provides the main types of databases and offers a wide variety of 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 like JSON, or XML. To do a more complex query on a NoSQL database, you typically need to 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 have been used extensively to store data in enterprise computing. SQL allows you to write queries combining 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 if 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: 73Each column will be stored in a database record as follows:
 - 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 Bluemix catalog

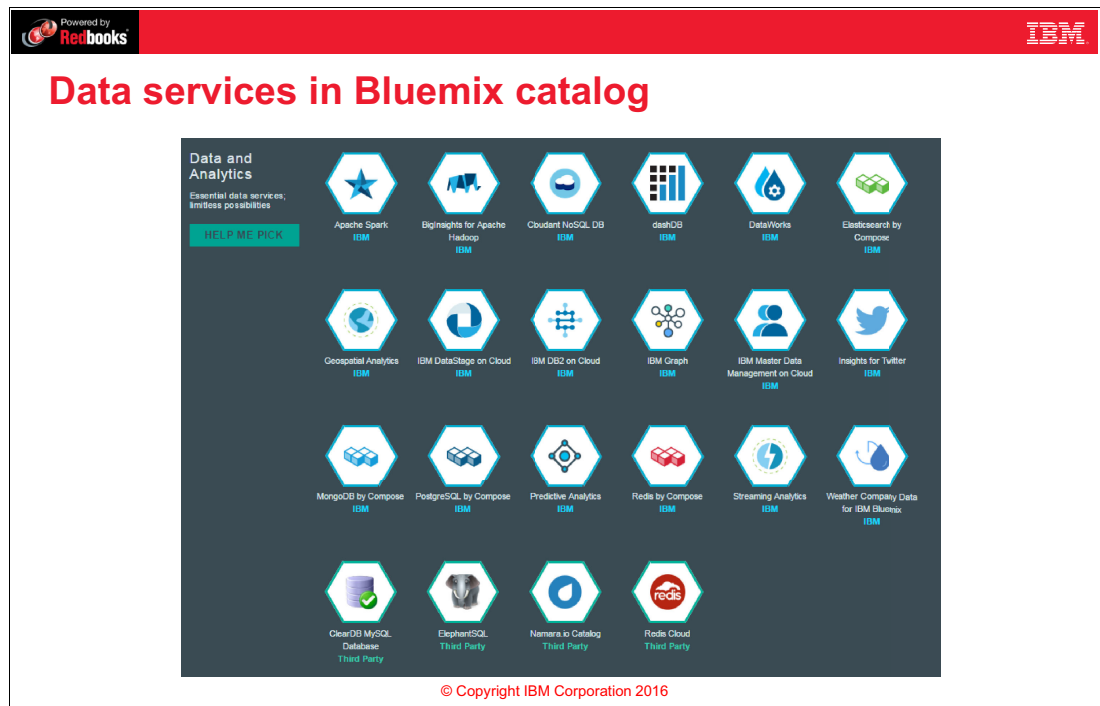


Figure 6-2 Data services in Bluemix catalog

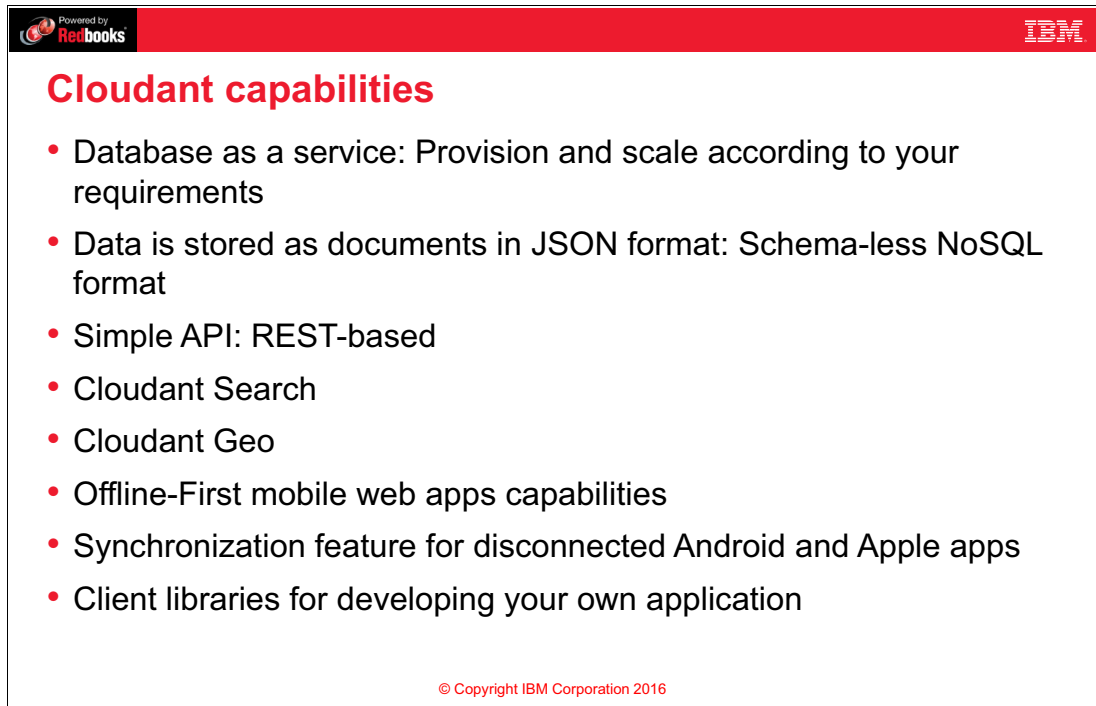
Notes:

IBM Bluemix offers a wide range of database management systems that are supported by IBM and other vendors to use with cloud computing:

- ▶ Examples of NoSQL database services in Bluemix include Cloudbant NoSQL DB and MongoDB.
- ▶ Examples of SQL database services in Bluemix include IBM DB2® on Cloud, PostgreSQL by Compose, ElephantSQL and ClearDB MySQL Database.
- ▶ Examples of In-memory columnar database services in Bluemix include dashDB™.
- ▶ Examples of key-value pair data services in Bluemix include these apps:
 - Redis: An open-source, fast key/value low maintenance store.
 - Data cache: Improves the performance and user experience of web applications by retrieving information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.
 - Session Cache: Improves application resiliency by storing session state information across many HTTP requests.

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

6.5 Cloudant capabilities



Powered by **Redbooks** **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

© Copyright IBM Corporation 2016

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) 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 a wide variety of 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, enabling 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 have 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, both offline and online, by storing and accessing data locally and then synchronizing this data with the cloud database when an Internet connection is available.

There are language-specific libraries for Cloudant that are wrappers that help you work with a simple API.

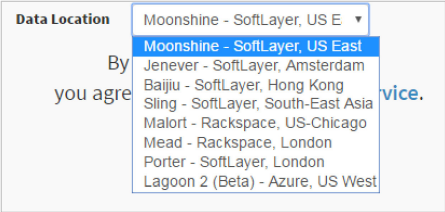
Source: <https://cloudant.com/>

6.6 Cloudbant in IBM Bluemix vs. Cloudbant.com

Powered by **Redbooks** **IBM**

Cloudbant in IBM Bluemix vs. Cloudbant.com

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



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

© Copyright IBM Corporation 2016

Figure 6-4 Cloudbant in IBM Bluemix vs. Cloudbant.com

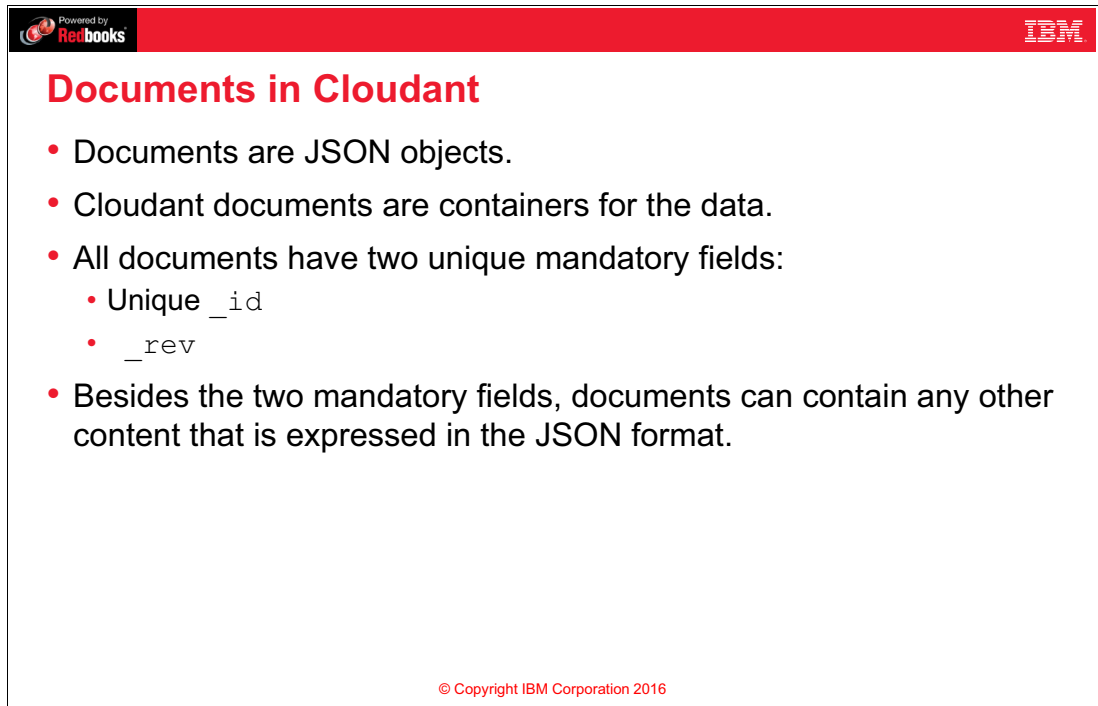
Notes:

You can sign-up to use the Cloudbant database at <https://cloudbant.com/> or use the Cloudbant data management services that are provided by IBM Bluemix.

When you sign up on <https://cloudbant.com/>, users are offered a choice of the underlying infrastructure as a service, including SoftLayer, Rackspace, and Azure.

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

6.7 Documents in Cloudbant



Documents in Cloudbant

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

© Copyright IBM Corporation 2016

Figure 6-5 Documents in Cloudbant

Notes:

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

- ▶ A unique identifier `_id` field serves as the document key. It can be created by the application or generated automatically by Cloudbant.
- ▶ A revision number `_rev` field is automatically generated and used internally by the Cloudbant 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 else your request will fail. It helps avoid conflicting data states.

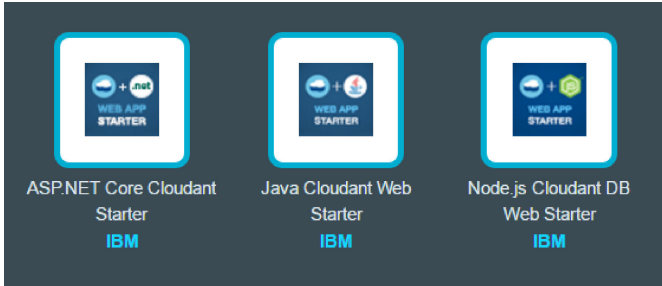
6.8 Getting started with Cloudant on IBM Bluemix

Powered by **Redbooks** **IBM**

Getting started with Cloudant on IBM Bluemix

These starter boilerplates on Bluemix use Cloudant:

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



The image shows three starter boilerplate icons arranged horizontally. Each icon is a square with a blue border and a white background. The first icon on the left features the ASP.NET logo and the text 'WEB APP STARTER'. The middle icon features the Java logo and the text 'WEB APP STARTER'. The third icon on the right features the Node.js logo and the text 'WEB APP STARTER'. Below each icon, the name of the starter is written in a sans-serif font, followed by the IBM logo.

ASP.NET Core Cloudant Starter
IBM

Java Cloudant Web Starter
IBM

Node.js Cloudant DB Web Starter
IBM

© Copyright IBM Corporation 2016

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.

Cloudant has a very rich documentation.

6.10 Cloudbant Dashboard

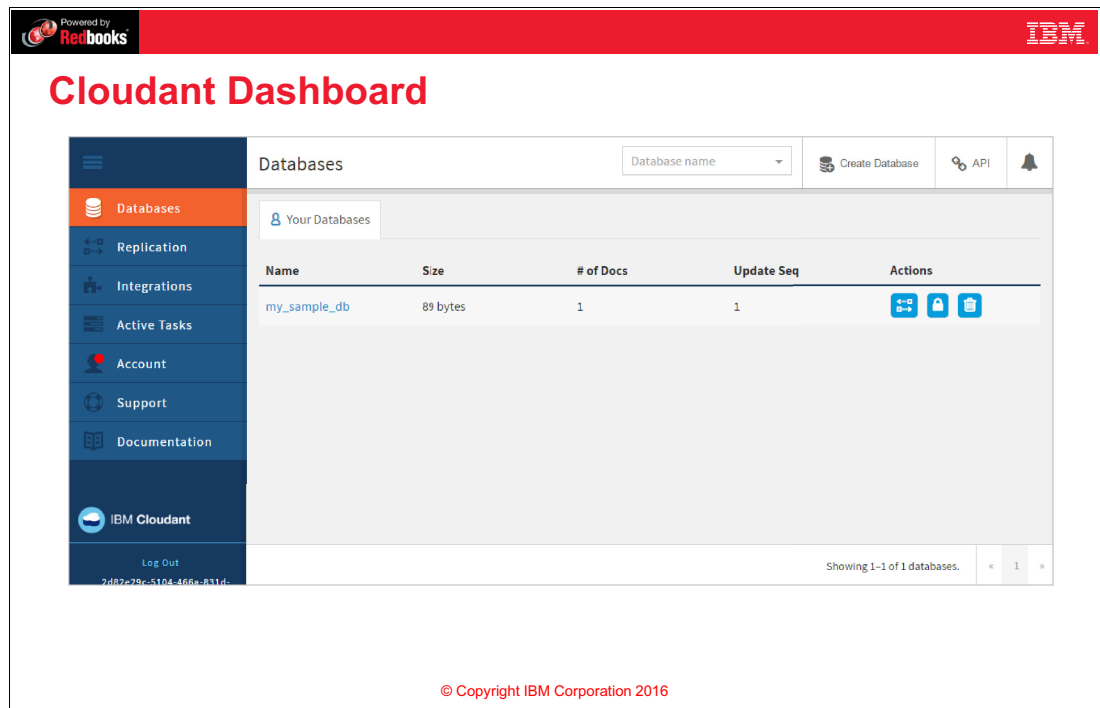


Figure 6-8 Cloudbant Dashboard

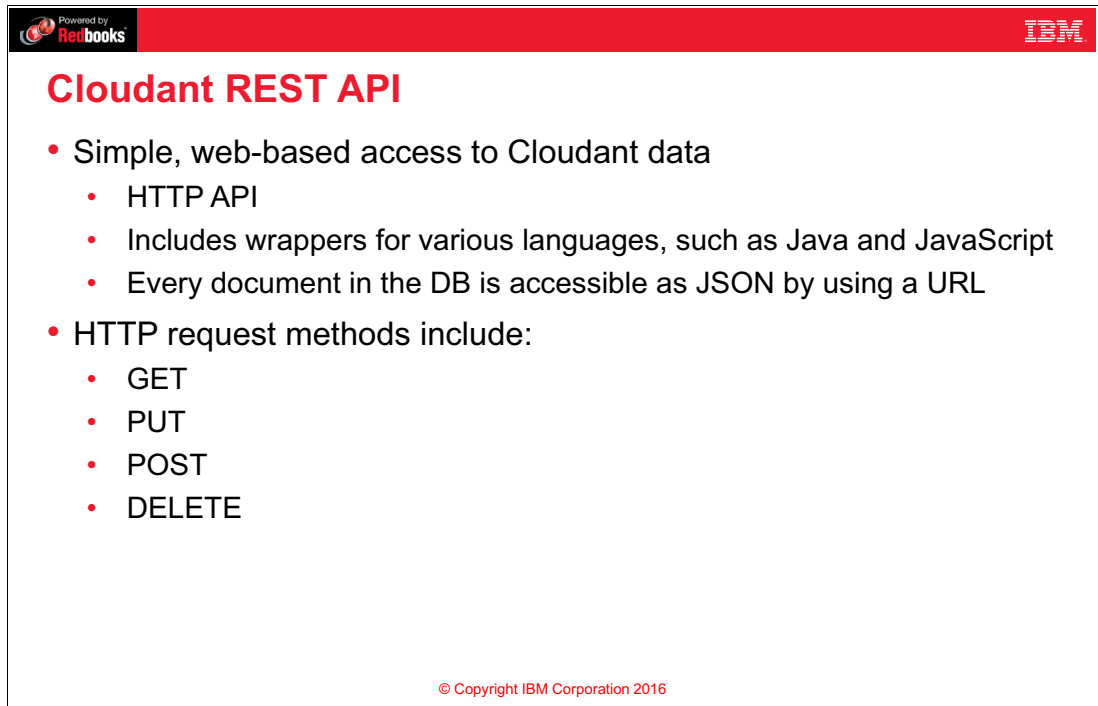
Notes:

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

You can display the contents of a Cloudbant document in 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 Cloudbant Dashboard user interface is similar whether you use <https://cloudbant.com/> or IBM Bluemix.

6.11 Cloudant REST API



The image is a screenshot of a presentation slide. At the top left, there is a logo for 'Powered by Redbooks'. At the top right, there is the 'IBM' logo. The main title of the slide is 'Cloudant REST API' in a large, bold, red font. Below the title, there is a bulleted list of features and HTTP methods. At the bottom right of the slide, there is a small copyright notice: '© Copyright IBM Corporation 2016'.

- 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

© Copyright IBM Corporation 2016

Figure 6-9 Cloudant REST API

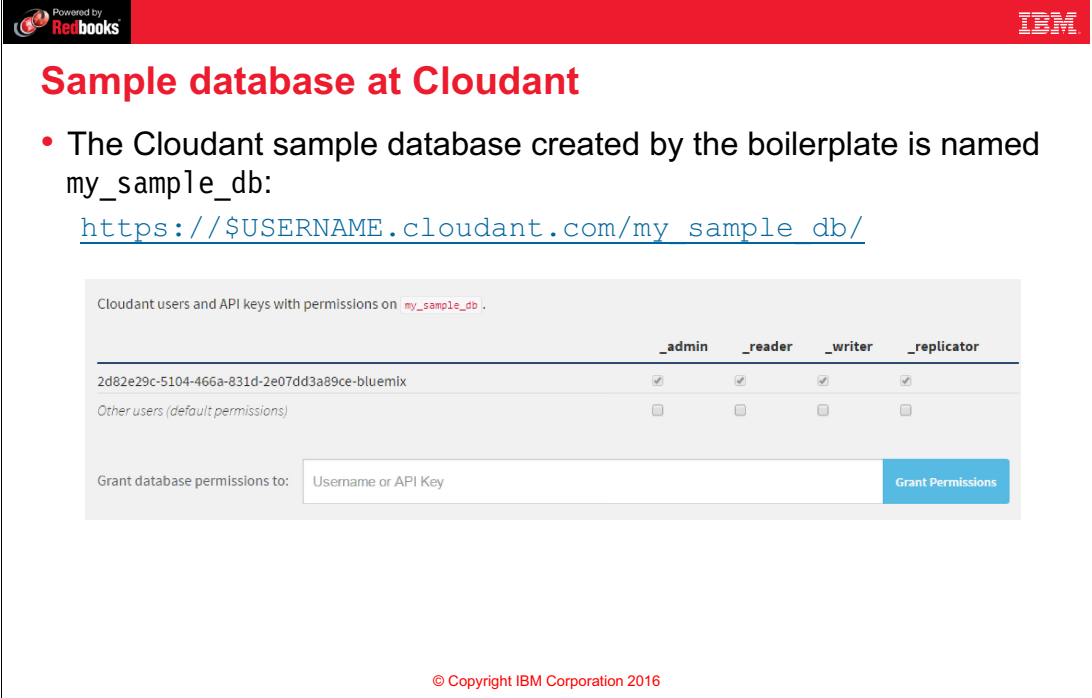
Notes:

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 a number of REST access methods for data read, add, update, and delete functions.

You can use these HTTP Request methods:

- ▶ GET: Used to request a specific JSON document.
- ▶ PUT: Used to create databases and documents.
- ▶ POST: Used to set values, and create documents.
- ▶ DELETE: Used to delete a specific document.

6.12 Sample database at Cloudant



The screenshot shows the Cloudant dashboard interface. At the top, there is a red header with the 'Powered by Redbooks' logo on the left and the 'IBM' logo on the right. Below the header, the title 'Sample database at Cloudant' is displayed in red. A bullet point indicates that the database created by the boilerplate is named 'my_sample_db' and provides the URL: [https://\\$USERNAME.cloudant.com/my_sample_db/](https://$USERNAME.cloudant.com/my_sample_db/). Below this, a table lists 'Cloudant users and API keys with permissions on my_sample_db'. The table has four columns: '_admin', '_reader', '_writer', and '_replicator'. The first row, representing the user '2d82e29c-5104-466a-831d-2e07dd3a89ce-blue mix', has all four permission checkboxes checked. The second row, 'Other users (default permissions)', has all four checkboxes unchecked. At the bottom of the table, there is a form labeled 'Grant database permissions to:' with a text input field containing 'Username or API Key' and a blue 'Grant Permissions' button. A copyright notice '© Copyright IBM Corporation 2016' is visible at the bottom of the screenshot.

	_admin	_reader	_writer	_replicator
2d82e29c-5104-466a-831d-2e07dd3a89ce-blue mix	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Other users (default permissions)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 6-10 Sample database at Cloudant

Notes:

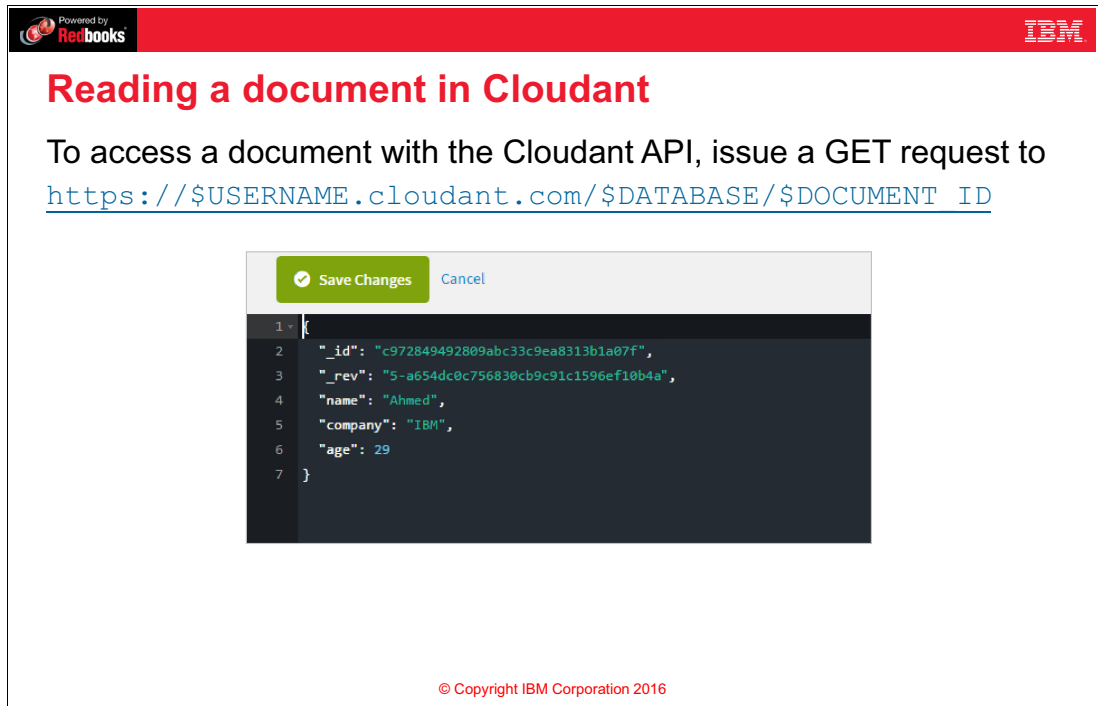
The sample database created by Node.js Cloudant DB Web Starter Boilerplate is named `my_sample_db`.

You can get the details of the database of 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.

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

6.13 Reading a document in Cloudant



The screenshot shows the Cloudant Dashboard interface. At the top, there is a red header with the 'Powered by Redbooks' logo on the left and the 'IBM' logo on the right. Below the header, the title 'Reading a document in Cloudant' is displayed in red. The main content area contains the text: 'To access a document with the Cloudant API, issue a GET request to [https://\\$USERNAME.cloudant.com/\\$DATABASE/\\$DOCUMENT_ID](https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID)'. Below this text is a code editor window with a dark background. The code editor shows a JSON document with the following fields: `"_id": "c972849492809abc33c9ea8313b1a07f",`, `"_rev": "5-a654dc0c756830cb9c91c1596ef10b4a",`, `"name": "Ahmed",`, `"company": "IBM",`, and `"age": 29`. The code editor has a line number column on the left with numbers 1 through 7. Above the code editor, there are two buttons: a green 'Save Changes' button with a checkmark icon and a blue 'Cancel' button. At the bottom right of the screenshot, there is a small red copyright notice: '© Copyright IBM Corporation 2016'.

Figure 6-11 Reading a document in Cloudant

Notes:

This figure represents accessing a document with Cloudant Dashboard. 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`, which is accessed using a **GET HTTP REST** request provided that 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 figure shows a sample document in the Cloudant Dashboard that contains three fields in addition to the mandatory fields `_id` and `_rev`: `name`, `company`, and `age`.

6.14 View all Documents

Powered by **Redbooks** **IBM**

View all documents

To view all documents at a database, issue a GET request to:
[https://\\$USERNAME.cloudant.com/\\$DATABASE/ all docs?include docs=true](https://$USERNAME.cloudant.com/$DATABASE/ all docs?include docs=true)

JSON Table Include Docs Document ID Options API

id	"c972849492809abc33c9ea8313b1a07f"
...	...
id	"c972849492809abc33c9ea8313b1a07f"

API URL
https://2d02e29c-6104-466a-831d-2d07dd3a89ce-bluemix.cloudant.com/my_sample_db/ Copy URL
View JSON

© Copyright IBM Corporation 2016

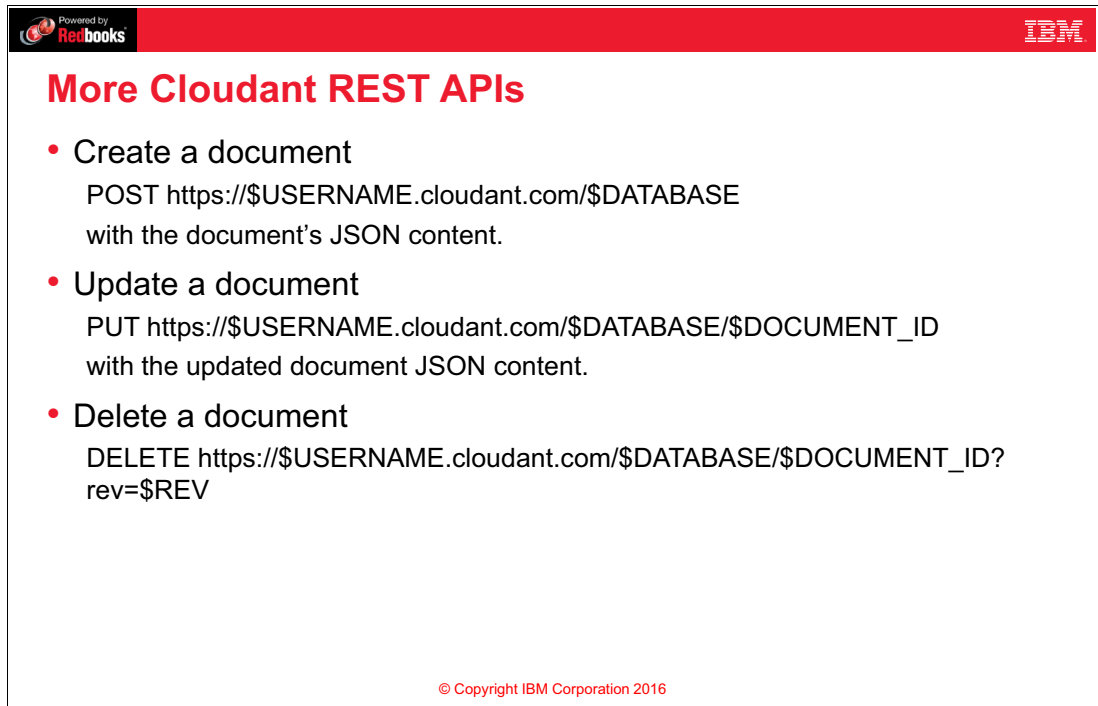
Figure 6-12 View all Documents

Notes:

Cloudant includes an index called `_all_docs` that allows you to 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`.

The figure shows that at any stage in Cloudant Dashboard, you can click the API to view the generated REST API for the page that the user is viewing in the Cloudant Dashboard.

6.15 More Cloudant REST APIs



Powered by **Redbooks** **IBM**

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`

© Copyright IBM Corporation 2016

Figure 6-13 More Cloudant REST APIs

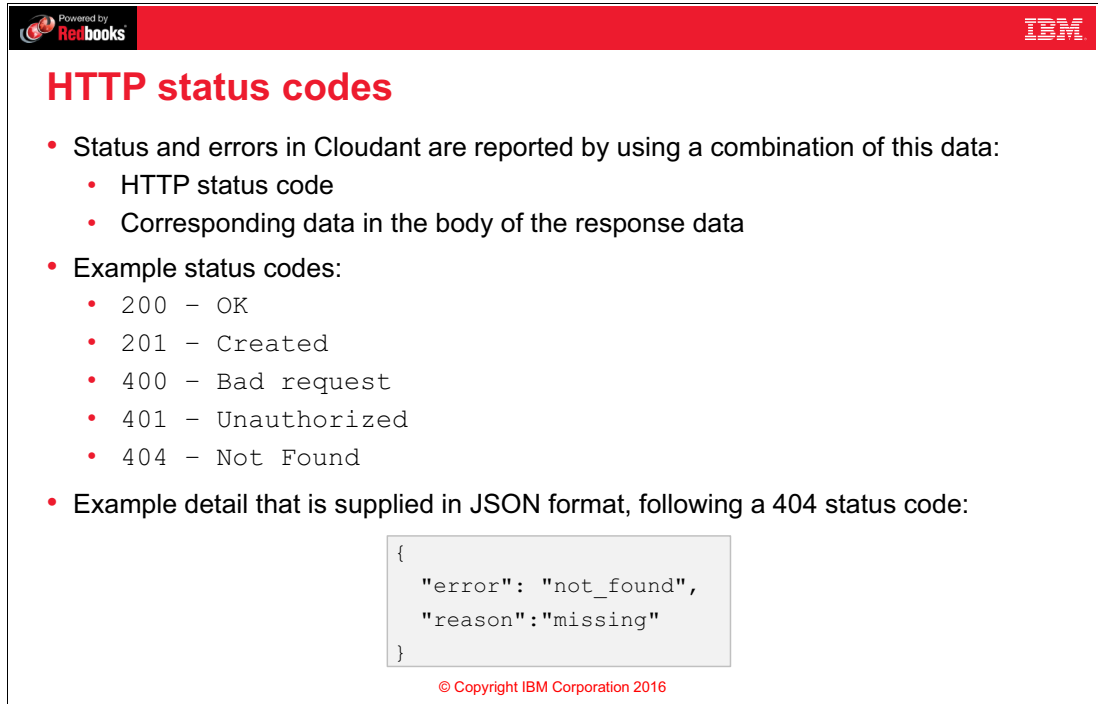
Notes:

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 new 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 HTTP status codes



Powered by **Redbooks** **IBM**

HTTP status codes

- Status and errors in Cloudant are reported by using a combination of this 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"}  
```

© Copyright IBM Corporation 2016

Figure 6-14 HTTP status codes

Notes:

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

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

Example status codes follows the widely-accepted status codes for HTTP including the following:

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 will respond with status code 404 in the header, and additional details about the error will be returned in the response as JSON as shown in the figure.

The language-specific libraries often have error handling built-in for these various cases.

6.17 Exercise 4 objectives

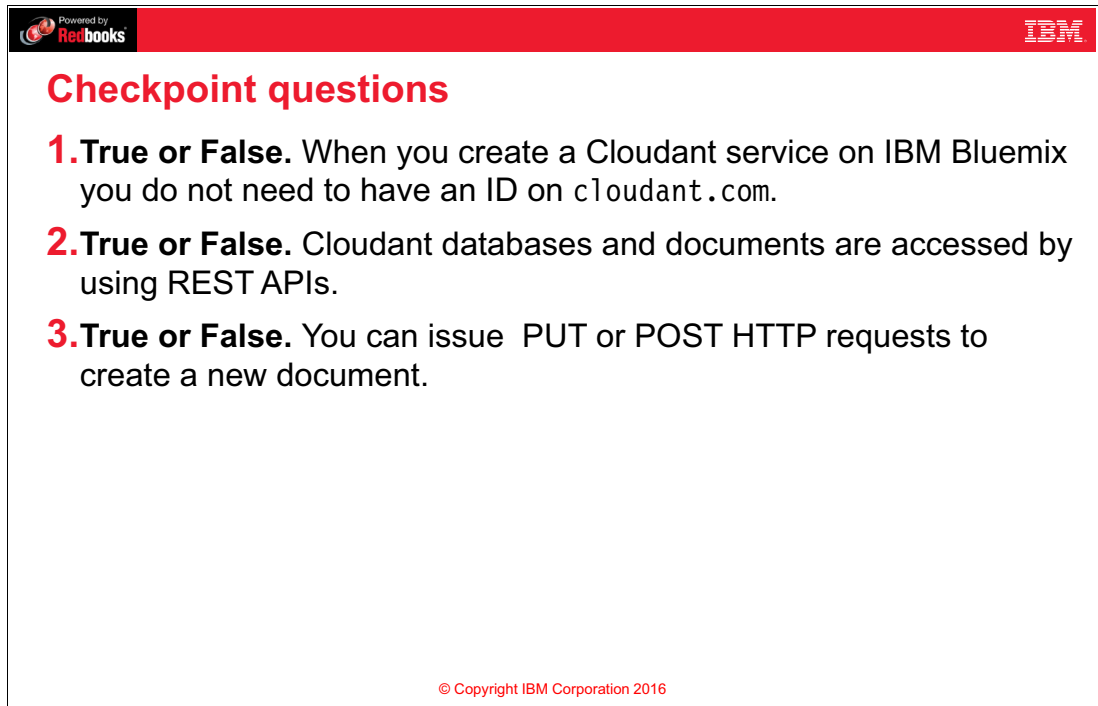


Exercise 4 objectives

- Create a starter Node.js application with a Cloudant database service from a Bluemix boilerplate.
- Review the service environment variables.
- Download the application package to your workstation.
- Review the application source code in Eclipse.
- Push the application to Bluemix.
- Access the Cloudant documentation.
- Explore the features of the Cloudant dashboard.
- Run the Bluemix Cloudant application.
- Run the application and add data to the Cloudant database.
- Verify the data that is stored in the database from the Cloudant dashboard.

© Copyright IBM Corporation 2016

6.18 Checkpoint questions



Powered by **Redbooks** **IBM**

Checkpoint questions

- 1. True or False.** When you create a Cloudant service on IBM Bluemix you do not need to have an ID on cloudant.com.
- 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 new document.

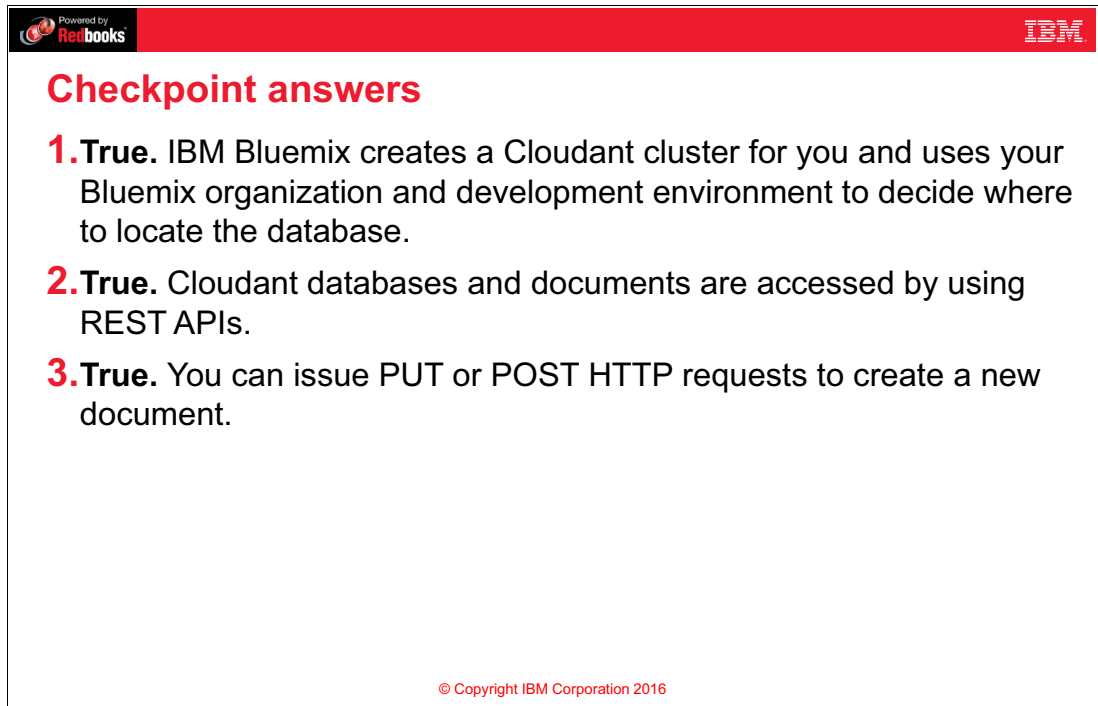
© Copyright IBM Corporation 2016

Figure 6-15 Checkpoint questions

Notes:

(none)

6.19 Checkpoint answers



The screenshot shows a document header with 'Powered by Redbooks' on the left and the 'IBM' logo on the right. The main title is 'Checkpoint answers' in red. Below it are three numbered items, each starting with '1.True.', '2.True.', and '3.True.' respectively. At the bottom right of the document area, there is a small copyright notice: '© Copyright IBM Corporation 2016'.

Checkpoint answers

- 1.True.** 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 new document.

© Copyright IBM Corporation 2016

Figure 6-16 Checkpoint answers

Notes:

1. True. 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 make a PUT or POST HTTP request to create a document. To deal with documents in Cloudant, PUT is used to create a document if the provided ID does not exist, and update it if the provided document ID exists.



IBM Bluemix Mobile Backend as a Service

This unit covers the following topics:

- ▶ Introduction to Mobile Backend as a Service
- ▶ How does MBaaS work?
- ▶ Push Notifications service
- ▶ Mobile Client Access service
- ▶ Mobile Quality Assurance service
- ▶ MobileFirst Services Starter Boilerplate

7.1 What you should be able to do

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

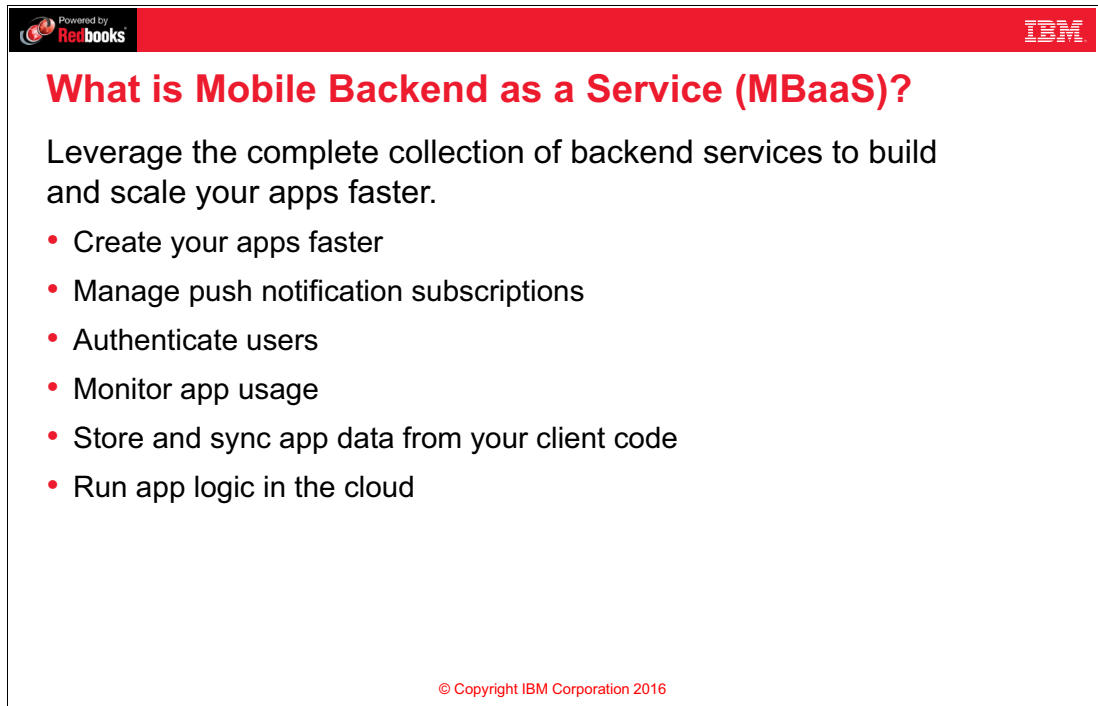
- ▶ Describe Bluemix Mobile Backend as a Service (MBaaS)
- ▶ Describe the MBaaS architecture
- ▶ Describe the IBM Push Notifications service
- ▶ Describe the Mobile Client Access service
- ▶ Describe the Mobile Quality Assurance service
- ▶ Describe how to create Bluemix Mobile applications by using 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>
- ▶ Mobile Client Access
<https://console.ng.bluemix.net/docs/services/mobileaccess/index.html>
- ▶ Push Notifications
<https://console.ng.bluemix.net/docs/services/mobilepush/index.html>
- ▶ Mobile Quality Assurance
<https://console.ng.bluemix.net/docs/services/MobileQualityAssurance/index.html>
- ▶ How to enhance mobile application security with Bluemix Mobile Services
Part 1:
<https://www.ibm.com/developerworks/library/mo-mobile-application-security-bluemix-mobile-services-1-trs/index.html>
Part 2:
<https://www.ibm.com/developerworks/library/mo-mobile-application-security-bluemix-mobile-services-2-trs/index.html>
- ▶ How to protect your application using Mobile Client Access
<https://developer.ibm.com/bluemix/2016/07/26/protect-web-apps-with-mobile-client-access/>
- ▶ How to extend an Android app 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)?



Powered by **Redbooks** **IBM**

What is Mobile Backend as a Service (MBaaS)?

Leverage the complete collection of backend services to build and scale your apps faster.

- 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

© Copyright IBM Corporation 2016

Figure 7-1 What is Mobile Backend as a Service (MBaaS)?

Notes:



With IBM Bluemix Mobile Backend as a Service, you can incorporate pre-built, managed, and scalable cloud services into your mobile applications without relying on IT involvement. Using the complete set of mobile backend services, you can accomplish these objectives:

- ▶ 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)

Powered by **Redbooks** **IBM**

IBM Mobile backend services (1 of 2)

- **Mobile Client Access**
 - Adds single sign-on (SSO) functionality to mobile apps.
 - Can be configured for users to log in using existing Google and Facebook accounts.
- **Mobile Foundation**
 - Expedites the setup of an IBM MobileFirst™ Platform Foundation, to develop, test, and operate enterprise apps.
 - For more information: <https://www.ibm.com/mobilefirst/>.
- **Mobile Quality Assurance**
 - Used to set up mobile quality services for mobile apps.
 - Collects information for crashes, bugs, user feedback and user sentiment.

© Copyright IBM Corporation 2016



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

Notes:




The following IBM mobile services are provided on Bluemix:

- ▶ **Mobile Client Access service:** Use the IBM Mobile Client Access for Bluemix service to add security functionality to your mobile app. You can configure client authentication and identity providers so that users can log in to the app with their existing Google or Facebook accounts.
- ▶ **Mobile Foundation:** Use the IBM Mobile Foundation for Bluemix service to expedite setting up an IBM MobileFirst Platform Foundation environment. IBM MobileFirst enterprise mobility solutions deliver a complete environment to develop apps and enable engagements that are designed specifically for mobile users.

7.5 IBM Mobile backend services (2 of 2)

Powered by  **Redbooks** 

IBM Mobile backend services (2 of 2)

- **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.
- **Mobile Application Content Manager**
 - Used to provide a consistent and easy content authoring experience with a workflow process for content approval.
 - For more information, see:
<https://www.ng.bluemix.net/docs/services/macm/index.html>.


© Copyright IBM Corporation 2016

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

Notes:

► Mobile Quality Assurance

Use the IBM Mobile Quality Assurance for Bluemix service to discover and set up mobile quality services for your apps. You can view high-level quality metrics for your mobile apps to get a quick understanding of the issues for apps that you are working on. These metrics include information for crashes, bugs, user feedback, and user sentiment. By viewing this information for your apps, you can determine whether to investigate specific issues further.

► 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. With 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.

► Mobile Application Content Manager

IBM Mobile Application Content Manager for Bluemix provides a consistent and easy content authoring experience with a workflow process for content approval. It merges the goals of the content authors, reviewers, content managers, and mobile application developers to create a simplified, cloud-based authoring experience.

7.6 MBaaS architecture

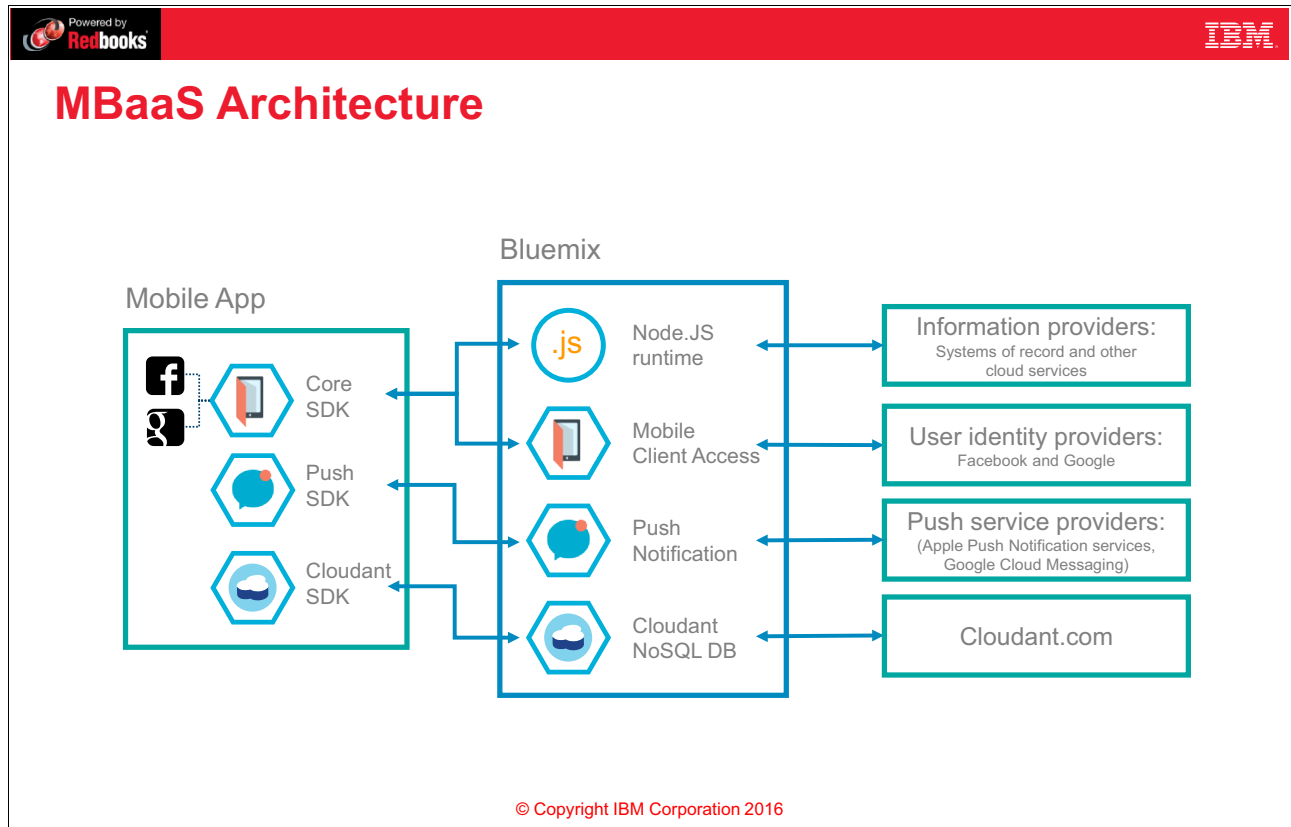


Figure 7-4 MBaaS architecture

Notes:

The diagram shows the structure of the APIs using Node.js runtime and some of the backend services provided by Bluemix. It shows how the different components interact with each other in the mobile environment.

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

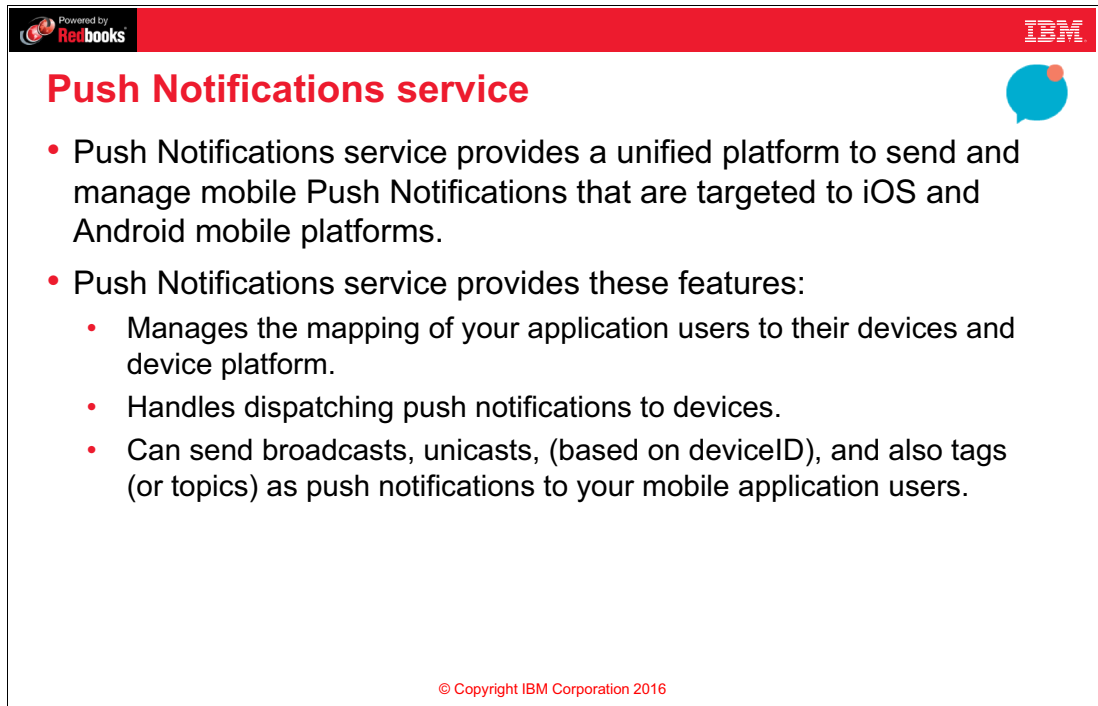
- ▶ Core SDK is used to interact with the server-side components (Node.js runtime and Mobile Client Access) to provide clients with Google+ and Facebook authentication as well as a custom Authentication provider.
- ▶ Push SDK to interact with the device-specific push service providers (iOS and Android only).
- ▶ Cloudant SDK to implement data persistence with the Cloudant NoSQL database service. It also provides offline data and backend synchronization.

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

- ▶ Node.js runtime interacts with external Information providers such as systems of record and other cloud services.
- ▶ Mobile Client Access service interacts with the user identity providers such as Google and Facebook accounts or other Identity providers.

- ▶ Push Notifications service must be 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



Powered by Redbooks

Push Notifications service

- 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 these features:
 - Manages the mapping of your application users to their devices and device platform.
 - Handles dispatching push notifications to devices.
 - Can send broadcasts, unicasts, (based on deviceId), and also tags (or topics) as push notifications to your mobile application users.

© Copyright IBM Corporation 2016

Figure 7-5 Push Notifications service

Notes:

You can use the 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 Bluemix Push Notifications service provides a unified platform to send and manage mobile Push Notifications that are targeted to iOS and Android mobile platforms. The Push Notifications service provides these 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), and also tags (or topics) as push notifications to your mobile application users

7.8 Push notification process

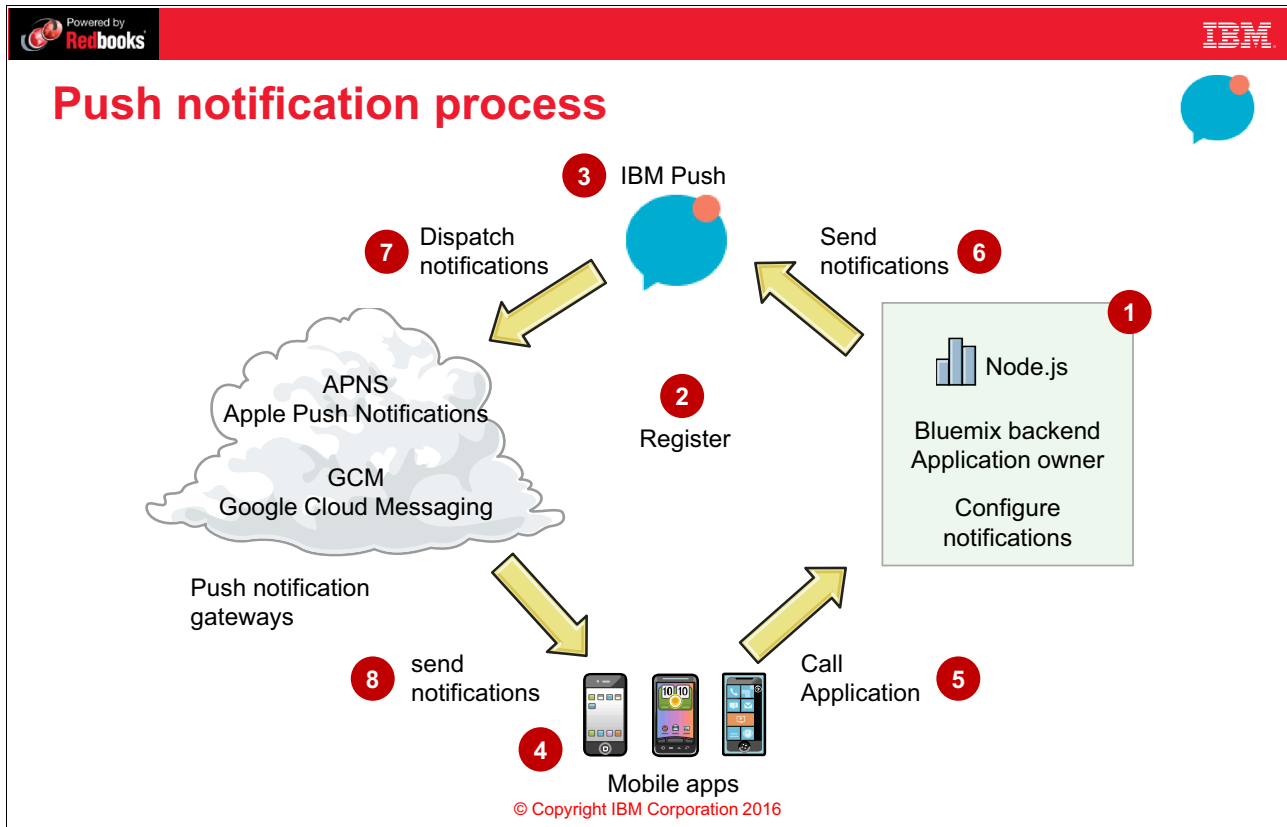


Figure 7-6 Push notification process

Notes:

A simplified push notification process is described here:

1. Create a 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 Bluemix. Here you use the credentials that you get from the notification provider.
4. Create the mobile client app that uses the IBM Push Notifications service and APIs.
5. The mobile app calls the Bluemix backend application.
6. The 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. Now you can send notifications to your app through both providers from a single interface using the Bluemix Push Notifications service.

7.9 Configure push notifications in Bluemix

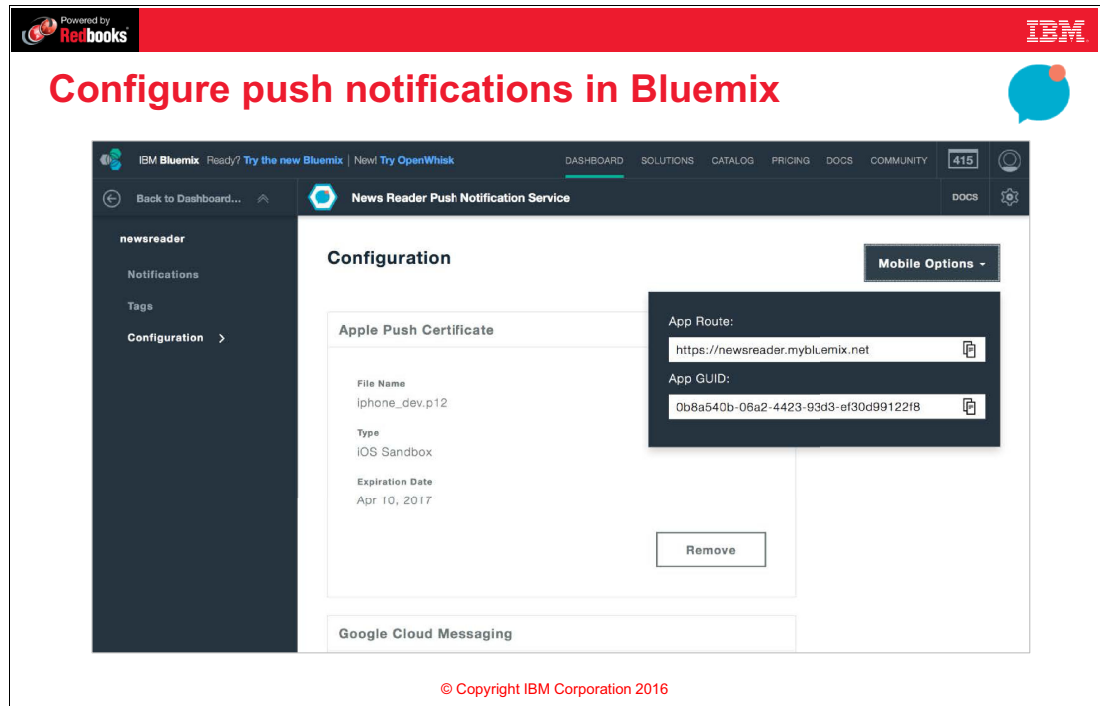


Figure 7-7 Configure push notifications in Bluemix

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 Bluemix.

The slide shows an example that uses the Application Route and Application GUID (both related to the Push Notifications service) to configure calling the service in the application. This configuration is shown in detail in Exercise 6 of this course.

7.10 Send manual notifications from Bluemix

Powered by Redbooks

IBM

Send manual notifications from Bluemix

Send Push

Enter the content and audience for the notification

Send To*
All Devices

Message Text
hello , this is a push from bluemi

iOS Custom Settings

Additional Payload

Specify the custom payload values.

SEND

© Copyright IBM Corporation 2016

Figure 7-8 Send manual notifications from Bluemix

Notes:

You can also send messages to a user or group by Platform, User ID, Device ID or Tag. A Tag is a topic that a group of users is following, related to the business domain of the mobile app.

You can also use the Bluemix Push Notifications service Rest API to send Push messages.

7.11 Push notification message

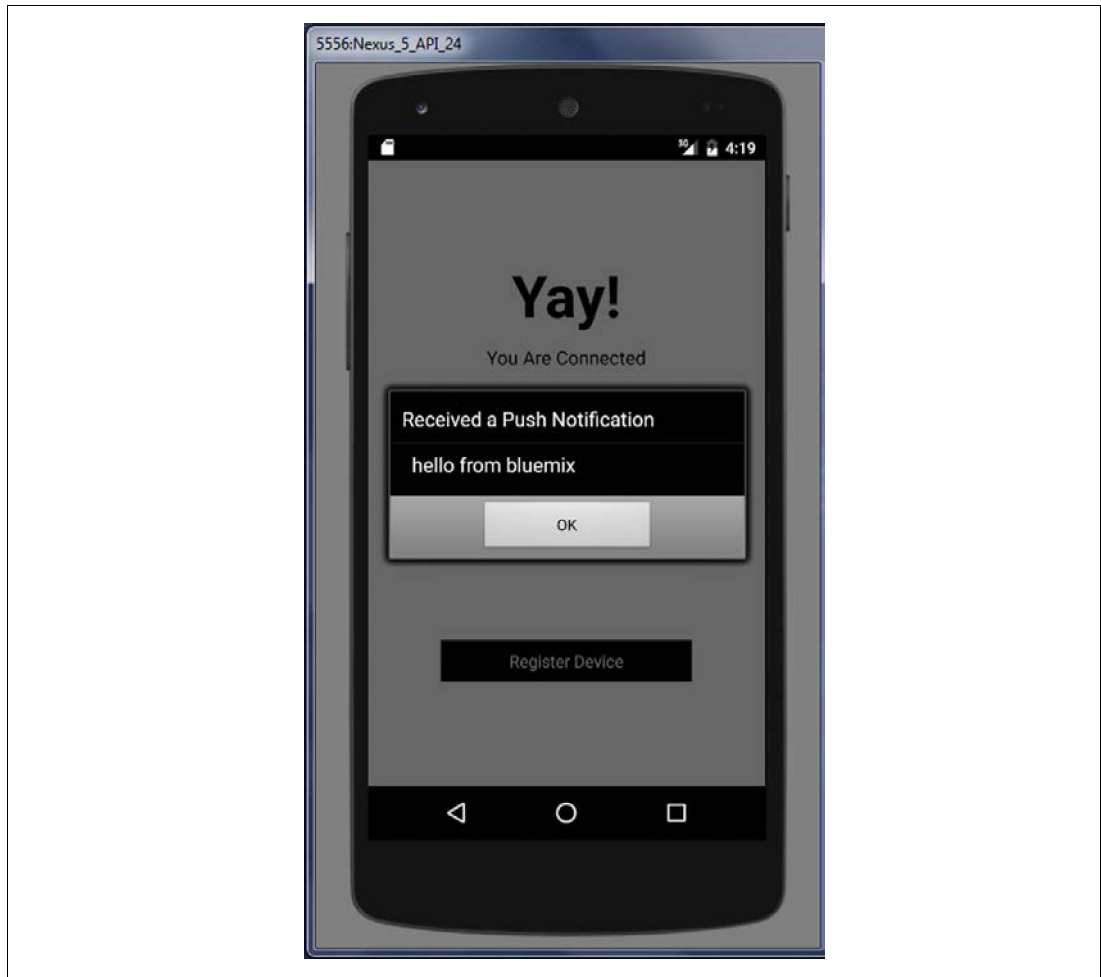




Figure 7-9 Push notification message

Notes:

The figure shows a push notification dialog box as seen in an Android device emulator running in the Android Studio environment. You can use an emulator when you develop a mobile app to enable application development and test on multiple platforms. Android Studio is the most popular IDE for Android platform application development.

The dialog box displays the text of the push notification that was sent from the Notification tab of the Bluemix IBM Push service.

7.12 Mobile Client Access (MCA) service

Powered by  **Redbooks** 

Mobile Client Access (MCA) service

- Adds security to your mobile app.
- You can configure client authentication and identity providers so that users can log in to the app with their existing Google or Facebook accounts.
- Mobile Client Access components:
 - **Dashboard**
Used to configure various authentication types and monitor the app.
 - **MCA Client SDK**
Used to instrument the app to use the MCA service.
 - **MCA Server SDK**
Used to protect resources hosted on Bluemix. Node.js and Liberty for Java™ are supported.

© Copyright IBM Corporation 2016

Figure 7-10 Mobile Client Access (MCA) service

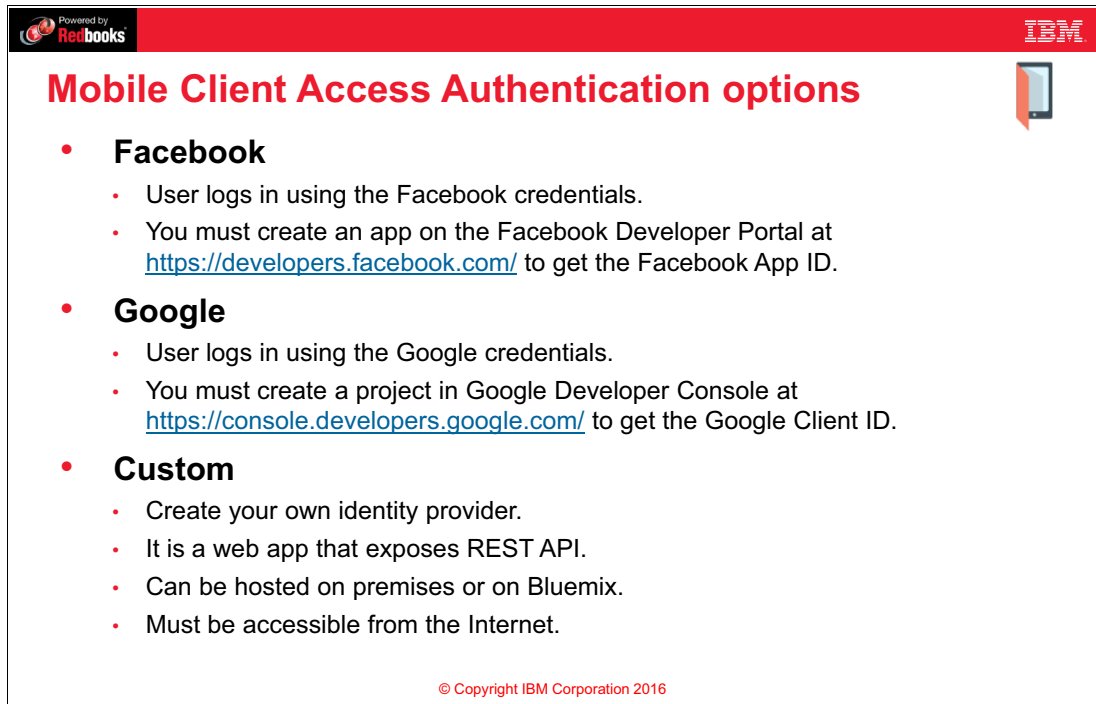
Notes:

You can add security to your mobile app with the IBM Mobile Client Access for Bluemix service. You can configure client authentication and identity providers so that users can log in to the app with their existing Google or Facebook accounts. You can also use the Mobile Client Access service to protect Node.js and Liberty for Java applications that are hosted on Bluemix with various authentication types.

Mobile Client Access includes these components:

- ▶ **Dashboard**
Configure various authentication types and monitor mobile application performance, analytics, usage statistics, and device logs.
- ▶ **MCA Client SDK**
Instrument mobile applications to use Mobile Client Access functionality. Supported platforms are: iOS 8 and later, Android 4 and later, and Cordova.
- ▶ **MCA Server SDK**
Protect resources that are hosted on Bluemix. Currently, supported runtimes are Node.js and Liberty for Java.

7.13 Mobile Client Access Authentication options



The screenshot shows a document header with 'Powered by Redbooks' on the left and the 'IBM' logo on the right. The main title is 'Mobile Client Access Authentication options' in red. Below the title is a list of three authentication options: Facebook, Google, and Custom. Each option has a bulleted list of requirements. At the bottom of the document, there is a copyright notice: '© Copyright IBM Corporation 2016'.

- **Facebook**
 - User logs in using the Facebook credentials.
 - You must create an app on the Facebook Developer Portal at <https://developers.facebook.com/> to get the Facebook App ID.
- **Google**
 - User logs in using the Google credentials.
 - You must create a project in Google Developer Console at <https://console.developers.google.com/> to get the Google Client ID.
- **Custom**
 - Create your own identity provider.
 - It is a web app that exposes REST API.
 - Can be hosted on premises or on Bluemix.
 - Must be accessible from the Internet.

© Copyright IBM Corporation 2016

Figure 7-11 Mobile Client Access Authentication options

Notes:

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

- ▶ Facebook
 - Your users log in to the mobile app with 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
 - Your users log in to the mobile app with 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 the Bluemix application.
- ▶ Custom
 - Use the IBM Mobile Client Access for Bluemix service and implement your own logic for collecting and validating credentials.
 - A custom identity provider is a web application that exposes a REST API. You can host the custom identity provider on premises or on IBM Bluemix.
 - The only requirement is that the custom identity provider must be accessible from the public internet so that it can communicate with the Mobile Client Access service.

7.14 Mobile Client Access (MCA) architecture

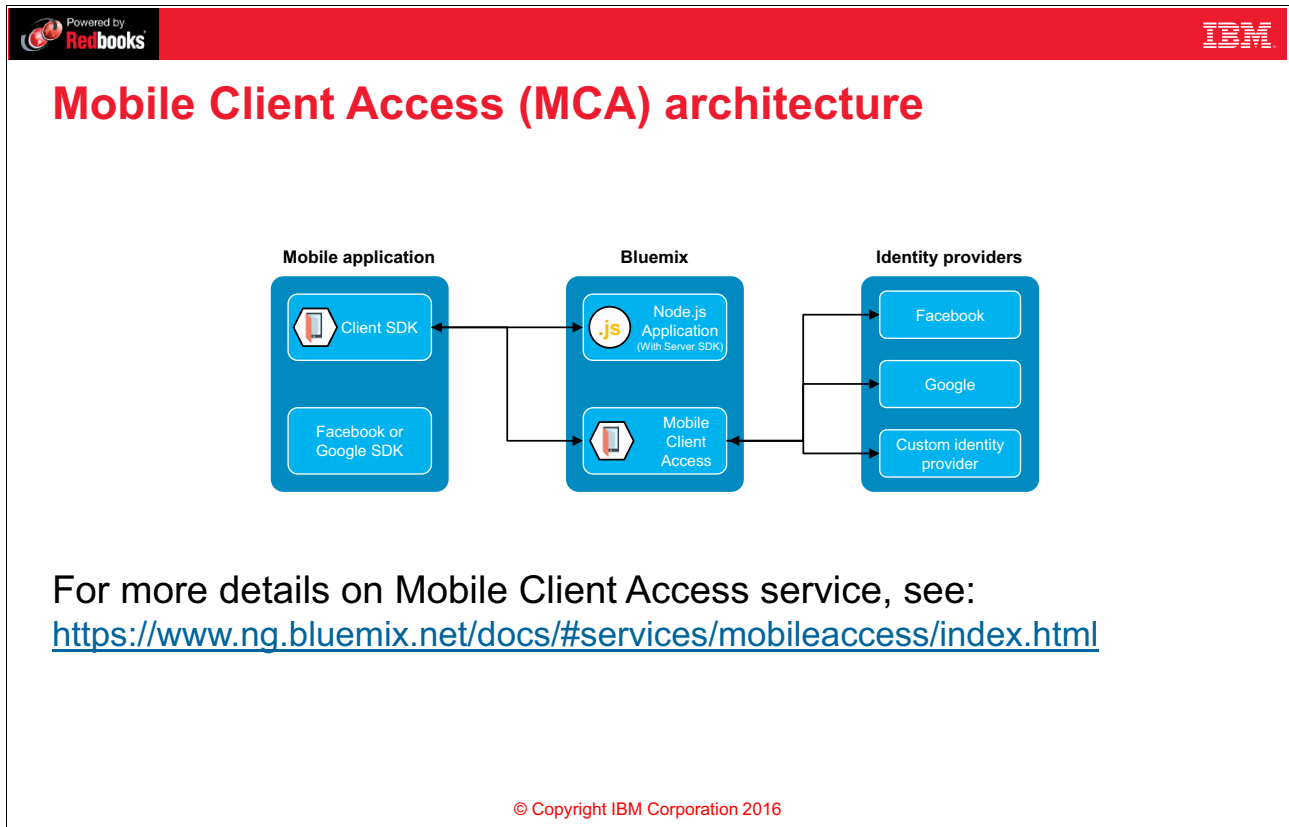


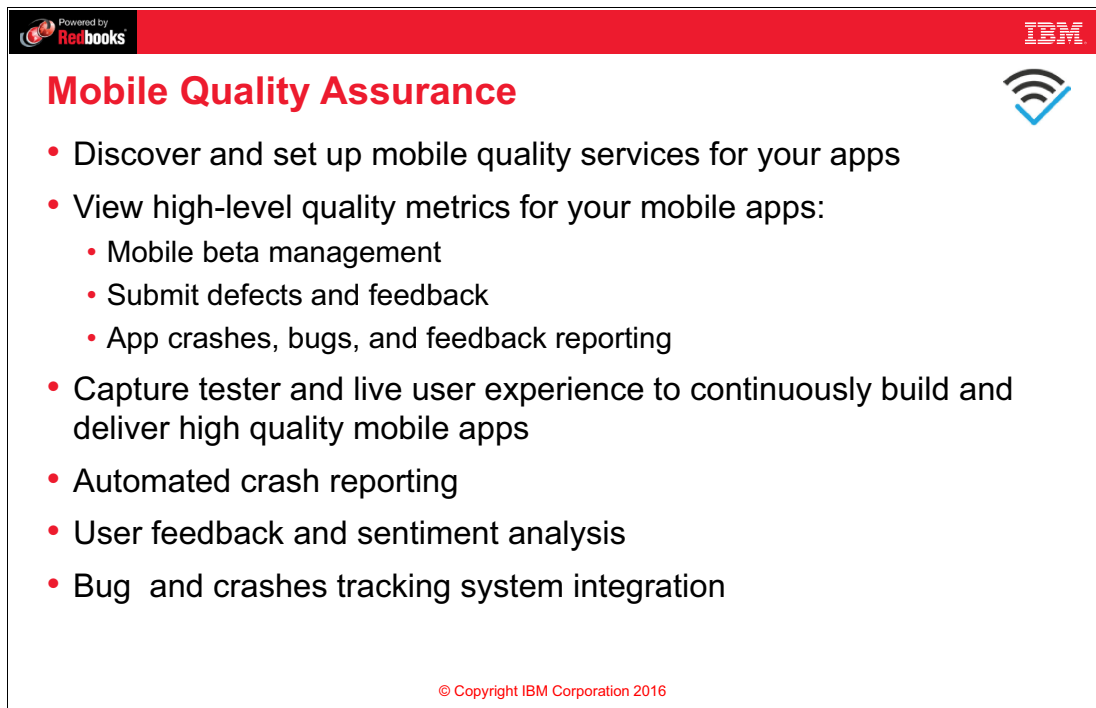
Figure 7-12 Mobile Client Access (MCA) architecture

Notes:

This slide shows how the different components of Mobile Client Access interact with each other:

- ▶ Protect your cloud resources (Node.js or Liberty for Java applications) with Mobile Client Access Server SDK.
- ▶ Use Request class provided by the Mobile Client Access Client SDK to communicate with your protected cloud resources.
- ▶ The Mobile Client Access Server SDK detects an unauthorized request and returns an HTTP 401 authorization challenge.
- ▶ Mobile Client Access SDK detects the HTTP 401 Authorization challenge and automatically starts the authentication process with the Mobile Client Access service.
- ▶ Facebook, Google, or Custom authentication is attempted.
- ▶ After successful authentication, Mobile Client Access returns an authorization token.
- ▶ The Mobile Client Access SDK automatically adds the authorization token to the original request, and re-sends the request to the cloud resource.
- ▶ The Mobile Client Access Server SDK extracts the access token from the request and validates it with the Mobile Client Access service.
- ▶ Access is granted. The response is returned to the mobile application.

7.15 Mobile Quality Assurance features



Mobile Quality Assurance

- Discover and set up mobile quality services for your apps
- View high-level quality metrics for your mobile apps:
 - Mobile beta management
 - Submit defects and feedback
 - App crashes, bugs, and feedback reporting
- Capture tester and live user experience to continuously build and deliver high quality mobile apps
- Automated crash reporting
- User feedback and sentiment analysis
- Bug and crashes tracking system integration

© Copyright IBM Corporation 2016

Figure 7-13 Mobile Quality Assurance features

Notes:

Use IBM Mobile Quality Assurance for Bluemix to discover and set up mobile quality services for your apps. You can view high-level quality metrics for your mobile apps to get a quick understanding of the issues for the apps that you are working on. These metrics include information for crashes, bugs, user feedback, and user sentiment. By viewing this information for your apps, you can determine whether to investigate specific issues further. For example, if the crash rate for an app spikes, you can click the crash rate to view more detailed information about crash reports for that app.

Mobile Quality Assurance helps you to accomplish these tasks:

- ▶ Create, manage, and run an efficient mobile beta program. Recruit prospective beta users, customize beta programs, and continuously engage with beta participants.
- ▶ Submit defects and provide feedback in seconds while using the application from your mobile device.
- ▶ Add screen shots and in app screen videos.
- ▶ Capture tester and live user experience to continuously build and deliver high quality mobile apps.
- ▶ Automated crash reporting.
- ▶ Get aggregated crash logs from pre-production and production environments automatically.
- ▶ User feedback and sentiment analysis.
- ▶ Get feedback about your app straight from your customers through APIs, and mine app ratings and reviews to extract actionable insights.

- ▶ Bug Tracking System Integrations.
- ▶ Plug Mobile Quality Assurance into your existing test flow by integrating with a wide variety of bug tracking systems. (for example IBM Rational Team Concert™, JIRA Software, Microsoft Team Foundation Server Bug Tracking, GitHub, Bugzilla, Rally)

7.16 Mobile Quality Assurance window



Figure 7-14 Mobile Quality Assurance window

Notes:

The slide shows the following information displayed in the Mobile Quality Assurance GUI:

- ▶ Number of Sessions for Preproduction and Production builds for a demo app on iOS.
- ▶ For preproduction sessions, 10 sessions with crashes, 10 with bugs, and 0 with feedback.
- ▶ From this GUI you can upload builds, show app key, distribute to test and configure integrations.

7.17 Getting started with Mobile Quality Assurance



The screenshot shows a document header with 'Powered by Redbooks' on the left and the 'IBM' logo on the right. The main title is 'Getting started with Mobile Quality Assurance' in red text, accompanied by a blue Wi-Fi icon. Below the title, the text reads: 'Complete these steps to get started with Mobile Quality Assurance:'. This is followed by a bulleted list: '• Add a mobile app to the Mobile Quality Assurance service.', '• Prepare the app for reporting.', and '• View reports from Service UI.'. At the bottom of the document, there is a copyright notice: '© Copyright IBM Corporation 2016'.

Figure 7-15 Getting started with Mobile Quality Assurance

Notes:

Complete these steps to get started with Mobile Quality Assurance:

1. Add a mobile app to the Mobile Quality Assurance service.

Important: Mobile apps in the Mobile Quality Assurance service represent actual mobile apps running on the users' devices. They are not the same as Bluemix backend applications. Therefore, you will not be billed on number of apps running or memory used because they are not running on Bluemix datacenters. However, you will be billed based on the number of Mobile Quality Assurance services that you are consuming.

2. Prepare the app for reporting:
 - a. Add platforms (Android or iOS).
 - b. Download and install libraries for the selected platforms. Each mobile app platform gets an app key.
 - c. Use the downloaded SDK and app key to instrument your app.
3. From the Bluemix service user interface, view reports related to Crashes, Bugs, Feedback, and Sentiment score.

7.18 MobileFirst Services Starter Boilerplate

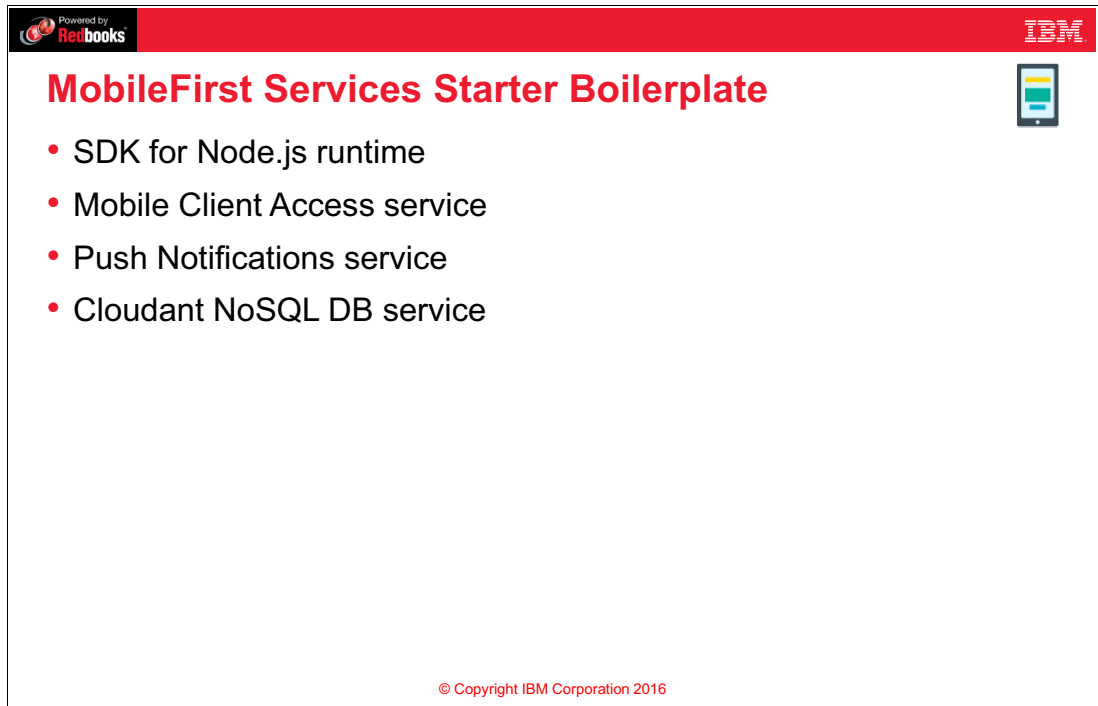


Figure 7-16 MobileFirst Services Starter Boilerplate

Notes:

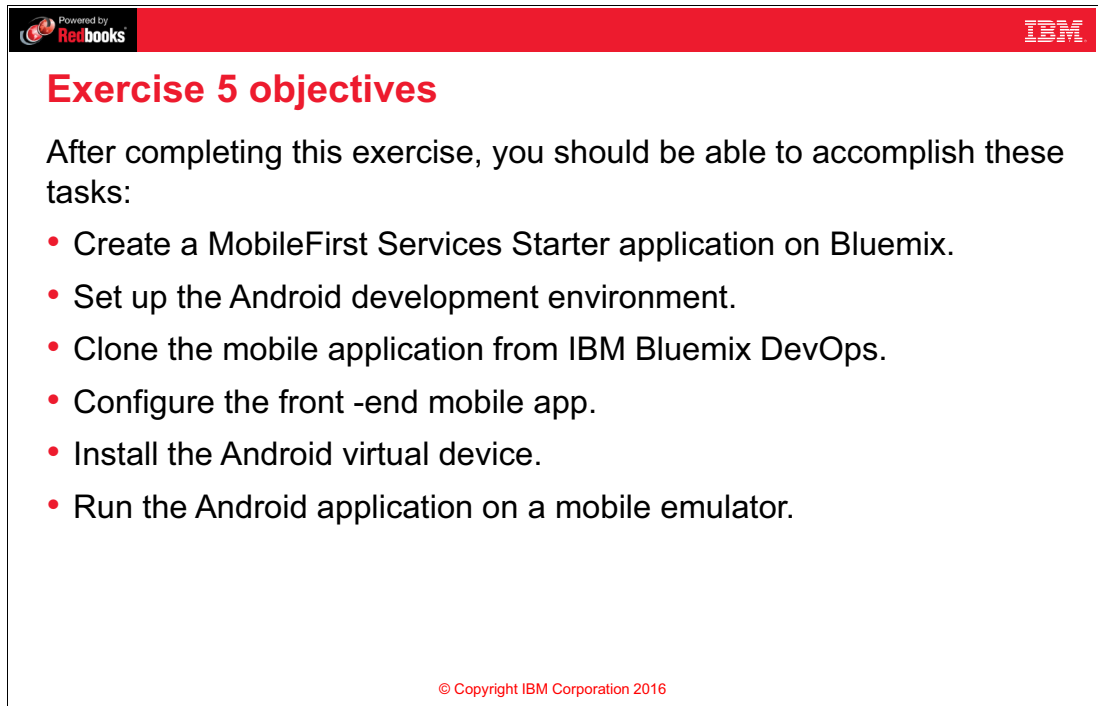
You can either use all of the Bluemix Mobile services together by using Bluemix IBM MobileFirst Services Starter boilerplate, or you can use individual services from the Bluemix catalog. You can use individual services if you want a different type of runtime for the backend app rather than Node.js, or if you do not want to use all the services which come with the Boilerplate.

Note: Exercise 5 highlights the use of the Node.js backend app with the MobileFirst Services Starter Boilerplate.

The MobileFirst Services Starter Boilerplate contains these features:

- ▶ A Mobile Client Access service.
- ▶ A Push Notifications service.
- ▶ A Cloudant NoSQL DB service.
- ▶ SDK for Node.js runtime.

7.19 Exercise 5 objectives



Powered by **Redbooks** **IBM**

Exercise 5 objectives

After completing this exercise, you should be able to accomplish these tasks:

- Create a MobileFirst Services Starter application on Bluemix.
- Set up the Android development environment.
- Clone the mobile application from IBM Bluemix DevOps.
- Configure the front -end mobile app.
- Install the Android virtual device.
- Run the Android application on a mobile emulator.

© Copyright IBM Corporation 2016

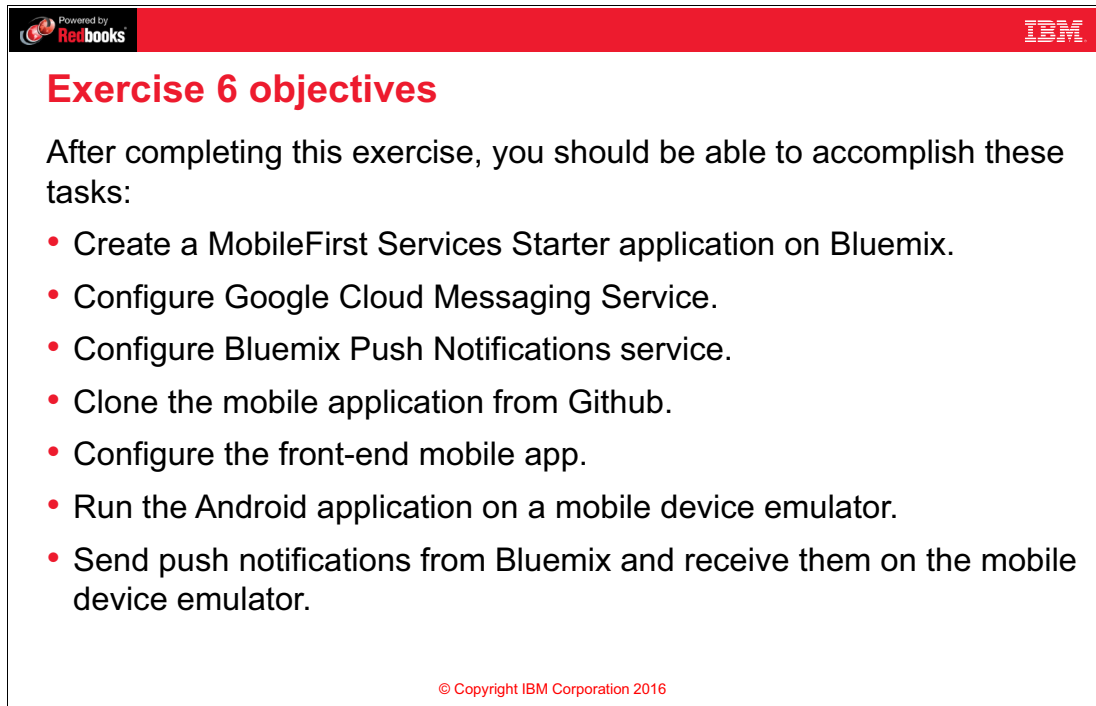
Figure 7-17 Exercise 5 objectives

Notes:

After completing this exercise, you should be able to:

- ▶ Create a MobileFirst Services Starter application on Bluemix.
- ▶ Set up the Android development environment.
- ▶ Clone the mobile application from IBM Bluemix DevOps.
- ▶ Configure the front-end mobile app.
- ▶ Install the Android virtual device.
- ▶ Run the Android application on a mobile emulator.

7.20 Exercise 6 objectives



The slide features a red header bar with the 'Powered by Redbooks' logo on the left and the 'IBM' logo on the right. The main content area is white with a red title 'Exercise 6 objectives'. Below the title, it states 'After completing this exercise, you should be able to accomplish these tasks:' followed by a bulleted list of seven tasks. At the bottom right of the slide, there is a small copyright notice: '© Copyright IBM Corporation 2016'.

Exercise 6 objectives

After completing this exercise, you should be able to accomplish these tasks:

- Create a MobileFirst Services Starter application on Bluemix.
- Configure Google Cloud Messaging Service.
- Configure Bluemix Push Notifications service.
- Clone the mobile application from Github.
- Configure the front-end mobile app.
- Run the Android application on a mobile device emulator.
- Send push notifications from Bluemix and receive them on the mobile device emulator.

© Copyright IBM Corporation 2016

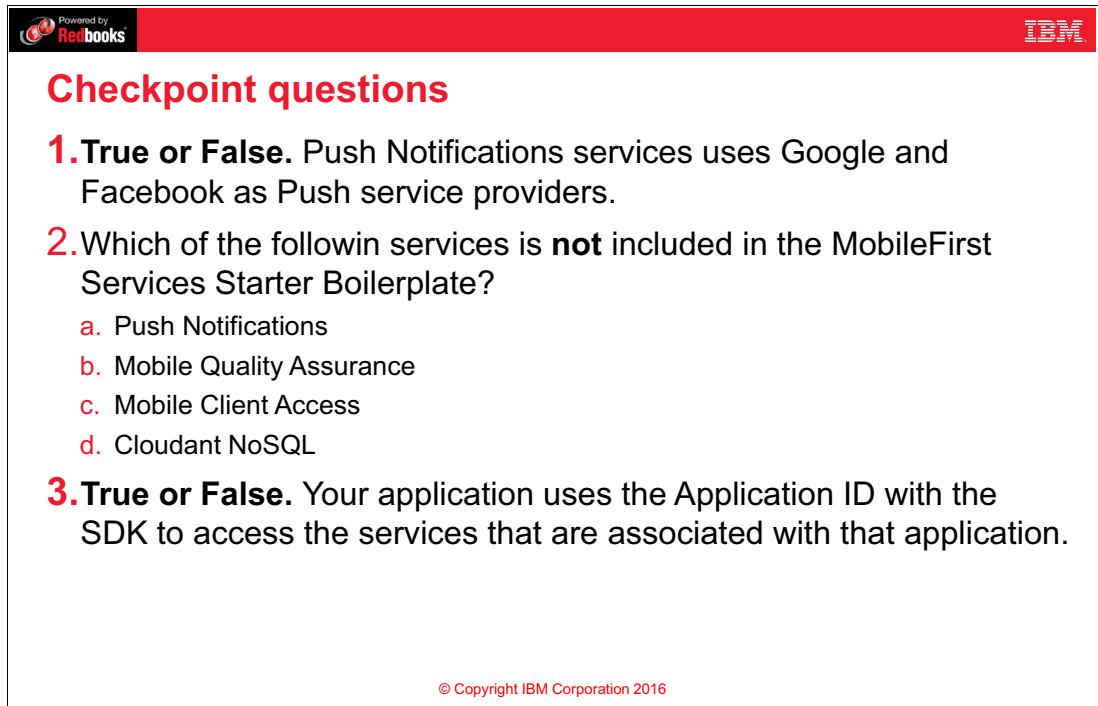
Figure 7-18 Exercise 6 objectives

Notes:

After completing this exercise, you should be able to:

- ▶ Create a MobileFirst Services Starter application on Bluemix.
- ▶ Configure Google Cloud Messaging Service.
- ▶ Configure Bluemix Push Notifications service.
- ▶ Clone the mobile application from Github.
- ▶ Configure the front-end mobile app.
- ▶ Run the Android application on a mobile device emulator.
- ▶ Send push notifications from Bluemix and receive them on the mobile device emulator.

7.21 Checkpoint questions



Powered by **Redbooks** **IBM**

Checkpoint questions

- 1. True or False.** Push Notifications services 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 Quality Assurance
 - c. Mobile Client Access
 - d. Cloudant NoSQL
- 3. True or False.** Your application uses the Application ID with the SDK to access the services that are associated with that application.

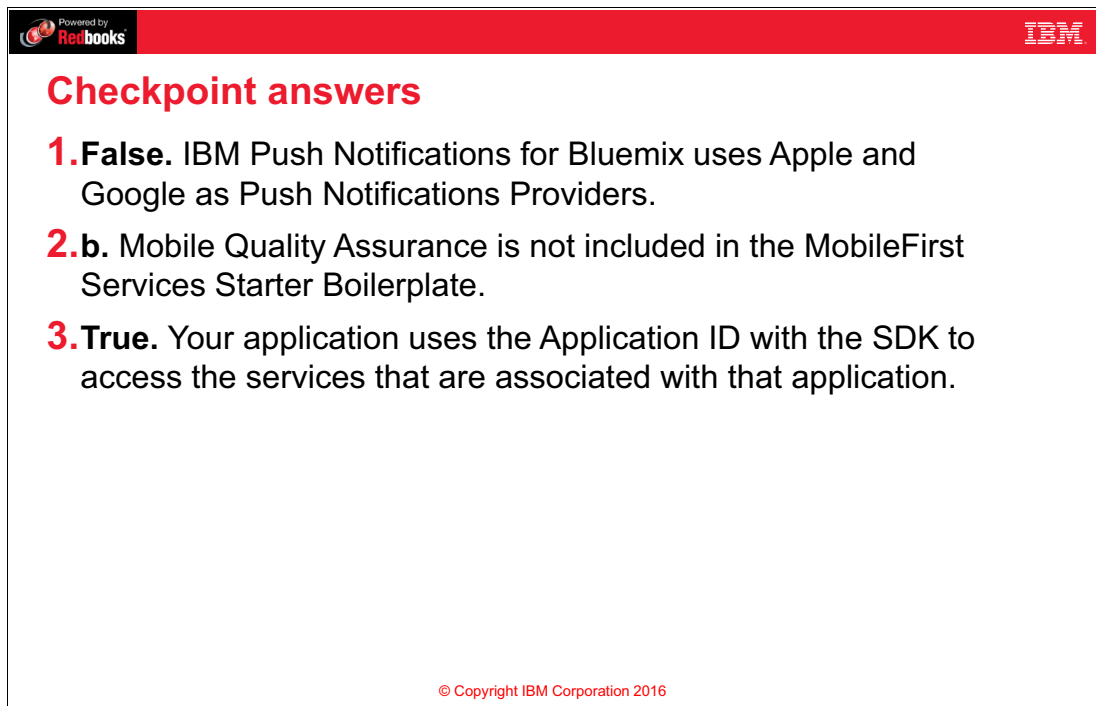
© Copyright IBM Corporation 2016

Figure 7-19 Checkpoint questions

Notes:

(none)

7.22 Checkpoint answers



The screenshot shows a document header with 'Powered by Redbooks' on the left and the 'IBM' logo on the right. The main content is titled 'Checkpoint answers' in red. It contains three numbered items:

- 1.False.** IBM Push Notifications for Bluemix uses Apple and Google as Push Notifications Providers.
- 2.b.** Mobile Quality Assurance is not included in the MobileFirst Services Starter Boilerplate.
- 3.True.** Your application uses the Application ID with the SDK to access the services that are associated with that application.

At the bottom right of the document, there is a copyright notice: © Copyright IBM Corporation 2016.

Figure 7-20 Checkpoint answers

Notes:

(none)

Redbooks

Essentials of Cloud Application Development on IBM Bluemix

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



SG24-8374-00

ISBN 0738442062

Printed in U.S.A.

Get connected

