



Essentials of Application Development on IBM Cloud

Presentations Guide

Ahmed Azraq

Hala A. Aziz

Uzma Siddiqui



In partnership with
IBM Skills Academy



Redbooks



International Technical Support Organization

Essentials of Application Development on IBM Cloud

December 2017

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

Third Edition (December 2017)

This edition applies to IBM Cloud.

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

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

Contents

Notices	ix
Trademarks	x
Preface	xi
Authors	xii
Now you can become a published author, too!	xiii
Comments welcome	xiii
Stay connected to IBM Redbooks	xiv
Unit 1. Introduction to cloud computing and IBM Cloud.....	1
1.1 What you should be able to do	2
1.2 References	2
1.3 What is cloud computing?	3
1.4 Traditional on-premises computing model	4
1.5 Factors contributing to growth of cloud	5
1.6 Cloud and mobile computing are changing traditional IT	6
1.7 Cloud service models	7
1.8 Infrastructure as a service architecture	8
1.9 Platform as a service (PaaS) architecture.....	9
1.10 Software as a service (SaaS) architecture	10
1.11 Provider and client responsibilities	11
1.12 Cloud computing: Benefits for developers	12
1.13 IBM Cloud	13
1.14 What is IBM Cloud?	14
1.15 IaaS from IBM Cloud.....	15
1.16 PaaS from IBM Cloud, Part 1	16
1.17 PaaS from IBM Cloud, Part 2	17
1.18 IBM Cloud: Choice of runtimes	18
1.19 IBM Cloud: Services	19
1.20 IBM Cloud: Regions and locations	21
1.21 How Cloud Foundry works with IBM Cloud.....	23
1.22 Unit summary	25
1.23 Checkpoint questions	26
1.24 Checkpoint answers	27
Unit 2. Getting started with IBM Cloud	29
2.1 What you should be able to do	30
2.2 References	30
2.3 What can you build in IBM Cloud?	31
2.4 Getting started: Creating an IBM Cloud account.....	32
2.5 Getting Started: IBM Cloud accounts	33
2.6 IBM Cloud catalog.....	34
2.7 IBM Cloud: Infrastructure	35
2.8 IBM Cloud: Infrastructure types.....	36
2.9 Understanding containers and virtual servers.....	37
2.10 IBM Cloud catalog: Containers and virtual servers.....	39
2.11 IBM Cloud: Platform	40
2.12 IBM Cloud: Applications and services	41
2.13 IBM Cloud: Boilerplates	42

2.14 IBM Cloud catalog: Cloud Foundry Apps	43
2.15 IBM Cloud catalog: Functions	44
2.16 IBM Cloud catalog: Mobile	45
2.17 IBM Cloud catalog: Other services	46
2.18 Creating an IBM Cloud application (1 of 3)	47
2.19 Creating an IBM Cloud application (2 of 3)	48
2.20 IBM Cloud app name must be unique	49
2.21 Creating an IBM Cloud application (3 of 3)	50
2.22 IBM Cloud: Dashboard	51
2.23 IBM Cloud: Application Details page	52
2.24 Testing applications through the application route	53
2.25 Adding an IBM Cloud service	54
2.26 Binding a service to an application (1 of 2)	55
2.27 Binding a service to an application (2 of 2)	56
2.28 IBM Cloud: Environment variables	57
2.29 IBM Cloud: Web Apps	58
2.30 IBM Cloud: Identity and Access Management	59
2.31 IBM Cloud: Resources, Users, and Access control	60
2.32 IBM Cloud: Resource groups	61
2.33 IBM Cloud: Resource group access	62
2.34 IBM Cloud: Organizing resources	63
2.35 Cloud Foundry: Organizing resources	64
2.36 Cloud Foundry: Organizations	65
2.37 Cloud Foundry: Domains and quota	66
2.38 Cloud Foundry: Spaces	67
2.39 Cloud Foundry: User roles	68
2.40 Managing Cloud Foundry organizations	69
2.41 Inviting users to a Cloud Foundry organization	70
2.42 Unit summary	71
2.43 Checkpoint questions	72
2.44 Checkpoint answers	73
Unit 3. Developing IBM Cloud applications from a local workstation	75
3.1 What you should be able to do	76
3.2 References	76
3.3 Node.js	77
3.4 Options to develop and deploy your IBM Cloud app	78
3.5 Overview of bx and cf command-line interfaces	79
3.6 Why use command-line tools?	80
3.7 Reviewing the documentation	81
3.8 Using the IBM Cloud CLI	82
3.8.1 Step 1: Install the IBM Cloud CLI	82
3.8.2 Step 2: Connect to your IBM Cloud account	83
3.8.3 Step 3: Connect to your Cloud Foundry organization	84
3.8.4 Step 4: Deploy your application to IBM Cloud	85
3.8.5 Step 5: Test your IBM Cloud application	86
3.9 Using the cf CLI	87
3.9.1 Step 1: Install the CLI	87
3.9.2 Step 2: Connect to your IBM Cloud account	88
3.9.3 Step 3: Deploy your application to IBM Cloud	89
3.10 IBM Cloud and Eclipse	90
3.11 Using IBM Eclipse tools for IBM Cloud	91
3.11.1 Step 1: Review the documentation	91

3.11.2 Step 2: Install the latest version of JDK and Eclipse.....	92
3.11.3 Step 3: Install Eclipse tools for IBM Cloud	93
3.11.4 Step 4: Connect to your IBM Cloud account.....	94
3.11.5 Step 5: Create a Node.js application project.....	95
3.12 Unit summary	96
3.13 Checkpoint questions	97
3.14 Checkpoint answers	98
Unit 4. Adopting a DevOps approach by using IBM Cloud Continuous Delivery	99
4.1 What you should be able to do	100
4.2 References	100
4.3 What is DevOps?	101
4.4 Benefits of DevOps	102
4.5 What is IBM Cloud Continuous Delivery?.....	103
4.6 What tools does IBM Cloud Continuous Delivery provide?.....	105
4.7 Additional DevOps services	106
4.8 Adding Continuous Delivery	107
4.9 Creating Continuous Delivery Toolchains.....	108
4.10 Adding Tools to the Toolchain.....	109
4.11 Adding Tool Integration to a Toolchain	110
4.12 Web IDE: Edit Code features	111
4.13 Web integrated development environment.....	112
4.14 Editing source code.....	113
4.15 Editor features: Code completion	114
4.16 Editor features: Run bar	115
4.17 IBM Cloud Live Sync features.....	116
4.18 Git Repos and Issue Tracking.....	117
4.19 Source control with a Git repository	118
4.20 Git repository overview	119
4.21 Connect a Git client to your repository on IBM Cloud.....	120
4.22 Issue Tracking tool	122
4.23 Creating an issue	123
4.24 Delivery Pipeline	124
4.25 Customizing Delivery Pipeline.....	125
4.26 Example: Default delivery pipeline	126
4.27 Configuring the Build Stage	127
4.28 Configuring Build jobs	128
4.29 Configuring Deploy jobs	129
4.30 Configuring Test job	130
4.31 Example: A successful build and deploy result.....	131
4.32 Unit summary	132
4.33 Checkpoint question	133
4.34 Checkpoint answer	134
Unit 5. REST architecture and Watson APIs.....	135
5.1 What you should be able to do	136
5.2 References	136
5.3 What is REST?	137
5.4 Applying REST to server-side applications.....	139
5.5 Example: Application model architecture for REST services	140
5.6 What is a RESTful web service?.....	141
5.7 Example: Sending an HTTP request to a REST service.....	142
5.8 Example: Receiving an HTTP response from a REST service.....	143

5.9 REST characteristics	144
5.10 Introduction to JSON	145
5.11 JSON data types	146
5.12 JSON data type: Objects	147
5.13 JSON data type: Arrays	148
5.14 What is IBM Watson?	149
5.15 Watson Services in IBM Cloud	150
5.16 Watson API Explorer	151
5.17 Example: Watson API Explorer - Natural Language Understanding (Authors)	152
5.18 IBM Watson in our daily life	154
5.19 Unit summary	155
5.20 Checkpoint questions	156
5.21 Checkpoint answers	157
Unit 6. Introduction to data services in IBM Cloud	159
6.1 What you should be able to do	160
6.2 References	160
6.3 Database choices on IBM Cloud	161
6.4 Data services in IBM Cloud catalog	162
6.5 Cloudant capabilities	163
6.6 Documents in Cloudant	165
6.7 Getting started with Cloudant on IBM Cloud	166
6.8 IBM Cloud: VCAP_SERVICES	167
6.9 Cloudant Dashboard	168
6.10 Cloudant REST API	169
6.11 Sample database at Cloudant	170
6.12 Reading a document in Cloudant	171
6.13 View all documents	172
6.14 More Cloudant REST APIs	173
6.15 Cloudant Query	174
6.16 HTTP status codes	175
6.17 Unit summary	176
6.18 Checkpoint questions	177
6.19 Checkpoint answers	178
Unit 7. IBM Cloud mobile backend as a service	179
7.1 What you should be able to do	180
7.2 References	180
7.3 What is mobile backend as a service (MBaaS)?	181
7.4 IBM Mobile backend services (1 of 2)	182
7.5 IBM Mobile backend services (2 of 2)	183
7.6 MBaaS architecture	184
7.7 Push Notifications service	186
7.8 Push Notifications process	187
7.9 Configuring push notifications in IBM Cloud (1 of 2)	189
7.10 Configuring push notifications in IBM Cloud (2 of 2)	190
7.11 Sending manual notifications from IBM Cloud	191
7.12 App ID service	192
7.13 App ID Authentication options	194
7.14 App ID architecture	195
7.15 What is Kinetise?	197
7.16 How to use Kinetise	198
7.17 MobileFirst Services Starter Boilerplate	199

7.18 Unit summary	200
7.19 Checkpoint questions	201
7.20 Checkpoint answers	202

Notices

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

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

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

IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

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

AppScan®
Bluemix®
Cloudant®
Db2®
developerWorks®

Global Business Services®
IBM®
IBM MobileFirst™
IBM Watson®
Redbooks®

Redbooks (logo) ®
Watson™
WebSphere®

The following terms are trademarks of other companies:

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

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

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

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

Preface

This IBM® Redbooks® publication is designed to teach university students and developers the foundational skills that are required to develop, test, and deploy cloud-based applications on IBM Cloud. It shows the latest features of IBM Cloud for developing cloud applications, enhancing applications by using managed services, and the use of DevOps services to manage applications.

This book is used as presentation guide for the IBM Skills Academy track Cloud Application Developer and as preparation material for the IBM professional certification exam IBM Certified Application Developer - Cloud Platform.

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 or IBM Cloud can also benefit from this course.

This IBM Redbooks publication covers the following topics:

- ▶ Cloud computing overview
- ▶ IBM Cloud overview
- ▶ Understanding what you can build in IBM Cloud
- ▶ Creating and managing an IBM Cloud application
- ▶ Managing your IBM Cloud users and resources
- ▶ Introduction to Node.js
- ▶ Build an application with the Cloud Foundry and IBM Cloud command-line interface
- ▶ Build an application with the IBM Eclipse Tools for IBM Cloud
- ▶ IBM Cloud DevOps overview
- ▶ Web IDE (Eclipse Orion)
- ▶ Source Code Management (SCM)
- ▶ Automated Delivery Pipeline (Build & Deploy) and Agile Planning Tools
- ▶ Introduction to the Representational State Transfer (REST) API
- ▶ Introduction to JavaScript Object Notation (JSON)
- ▶ Using REST APIs with IBM Watson® (example)
- ▶ Database services in IBM Cloud
- ▶ IBM Cloudant®
- ▶ Introduction to Mobile Backend as a Service
- ▶ Push Notifications service
- ▶ App ID service
- ▶ Kinetise service and MobileFirst Starter Boilerplate

The workshop materials were created in November 2017. Therefore, all IBM Cloud features that are described in this book are current as of November 2017.

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

Authors

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

Ahmed Azraq is a Cloud Solution Lead in IBM. He has recently joined Global IBM Cloud Services and Solutioning - East Hub. His primary responsibility is to help clients across Middle East & Africa (MEA) and Asia Pacific to realize their journey towards adopting IBM Cloud and IBM Watson. Since joining IBM in 2012, Ahmed worked as a technical team leader, and architect in the IBM MEA Client Innovation Center, which is part of IBM Global Business Services® (GBS). Ahmed has acquired several professional certifications, including Open Group IT Specialist, IBM Certified Solution Advisor - Cloud Reference Architecture, IBM Certified Application Developer - Cloud Platform, Java EE, IBM Business Process Manager, Agile development process, and IBM Design Thinking. Ahmed also has delivered training on IBM Cloud, DevOps, hybrid cloud Integration, Node.js, and Watson APIs to IBM clients, IBM Business Partners, university students and professors around the world. He is the recipient of several awards, including Eminence and Excellence Award in the IBM Watson worldwide competition Cognitive Build, the IBM Service Excellence Award for showing excellent client value behaviors, and knowledge-sharing award. Ahmed also has authored several other IBM Redbooks publications.

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

Uzma Siddiqui is an IBM Certified Cloud Application Developer. Her current focus is IBM Cloud services. She works as a DevOps engineer for the API Connect service on IBM Cloud, focusing on microservices and Docker containers. She is passionate about building solutions that use cognitive, analytics, and the Internet of Things services on IBM Cloud. Uzma is also the author and reviewer of multiple IBM developerWorks® articles.

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

Thanks to the following authors of the previous edition of this IBM Redbooks publication:

- ▶ Mohamed El-Khouly
- ▶ Sally Fikry
- ▶ Ahmed S. Hassan
- ▶ Mahmoud Rasmy
- ▶ Ben Smith

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

Andrea Emlian
Ossama Hakim
Juan Pablo Napoli
Denny Punnoose
IBM Skills Academy

Karim El-Bawab
Global Business Services, IBM Egypt

Valerie Lampkin
IBM Cloud Services, IBM US

Mike Melick
IBM Watson and Cloud Platform, IBM Canada

Neal Timpe

Bo Yang

IBM Watson and Cloud Platform, IBM US

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 Cloud

This unit covers the following topics:

- ▶ Cloud computing overview
- ▶ IBM Cloud overview

1.1 What you should be able to do

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

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

1.2 References

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

- ▶ What is Cloud Computing?
<https://www.ibm.com/cloud-computing/what-is-cloud-computing>
- ▶ Cloud Service Models:
 - <https://www.ibm.com/cloud-computing/iaas-paas-saas>
 - <http://www.ibm.com/developerworks/cloud/library/cl-cloudservicemodels/>
- ▶ IBM Cloud:
<https://console.bluemix.net/docs/overview/ibm-cloud.html>

1.3 What is cloud computing?

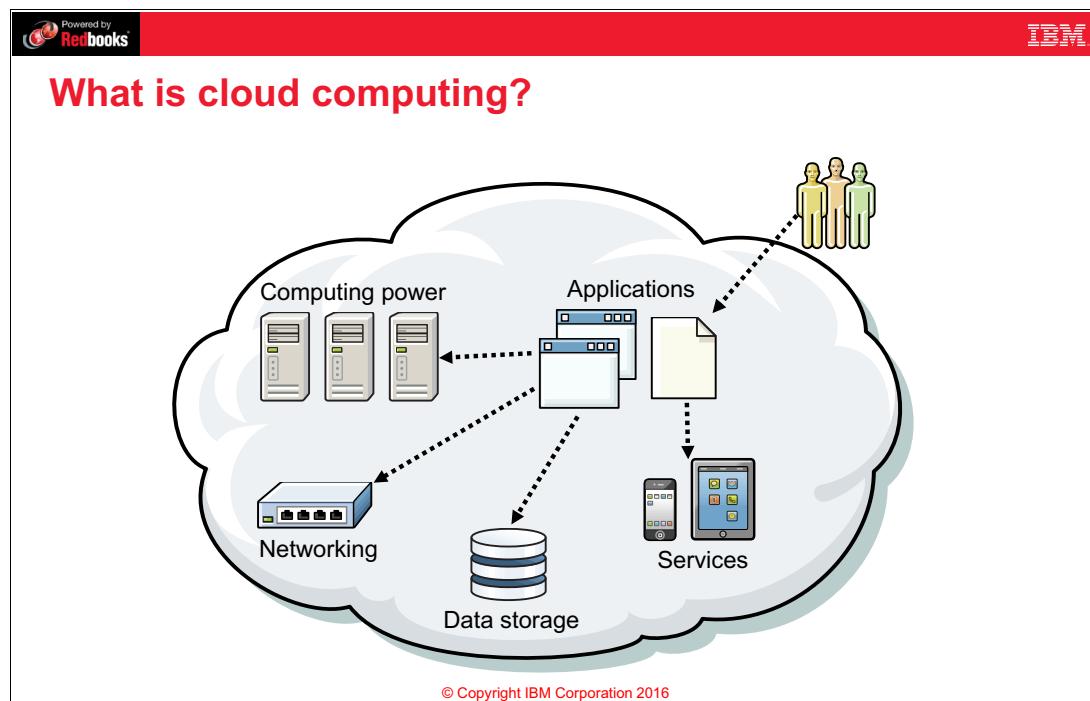


Figure 1-1 What is cloud computing?

Notes:

The term *cloud* is used as a metaphor for the internet and a virtualized set of hardware resources. The term also 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 Traditional on-premises computing model

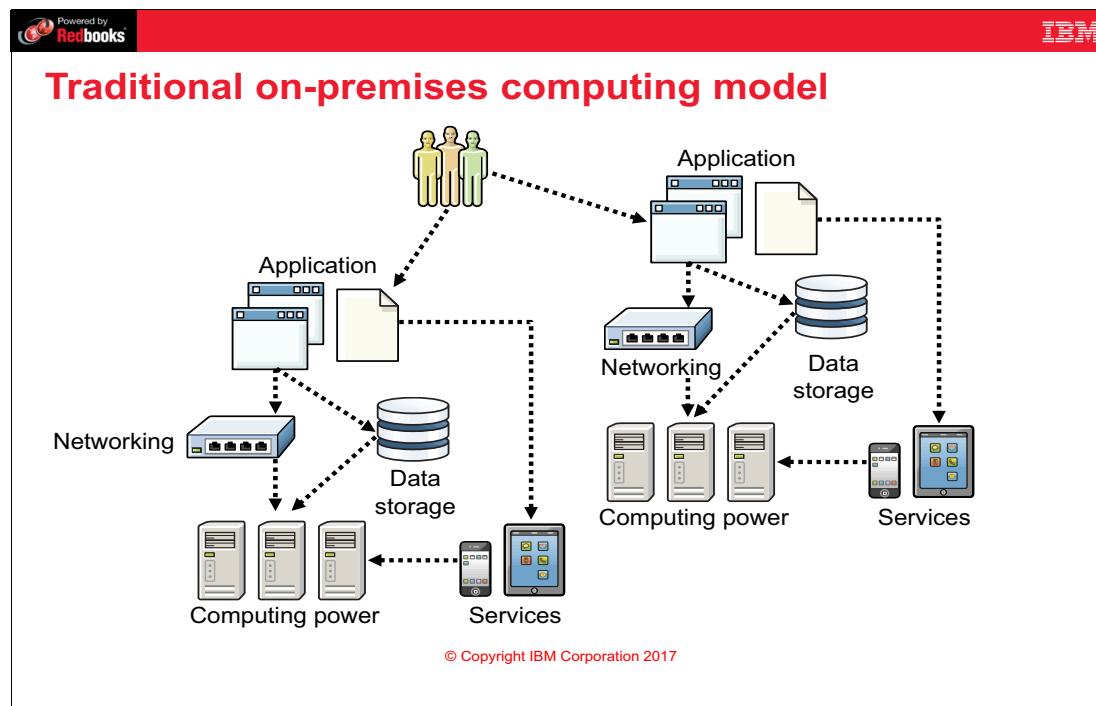


Figure 1-2 Traditional on-premises computing model

Notes:

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

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

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

1.5 Factors contributing to growth of cloud

The slide is titled "Factors contributing to growth of cloud" in red text. It features a bulleted list of four items. At the bottom right, there is a copyright notice: "© Copyright IBM Corporation 2017".

Factors contributing to growth of cloud

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

© Copyright IBM Corporation 2017

Figure 1-3 Factors contributing to growth of cloud

Notes:

One factor contributing to the growth of cloud computing is that today's applications must be delivered quickly. 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 and 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 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 resources to a specific application (*scaling up*, or *vertical scaling*), or add duplicate instances of an application (*scaling out*, or *horizontal scaling*) to handle increased customer load. Cloud platforms provide standardized methods to scale applications.

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

1.6 Cloud and mobile computing are changing traditional IT



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 methodologies:
 - Development processes
 - Application design
 - Development tools
- Virtualization and high-speed internet connectivity are the foundation for cloud computing:
 - Ability to represent a physical machine with software-defined machines

© Copyright IBM Corporation 2017

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 different from the traditional IT computing models. Cloud computing enables ubiquitous computing, where computing is made to appear anytime and everywhere, using any device, in any location, and in any format. The surge of mobile devices is greatly contributing to this model.

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

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

1.7 Cloud service models

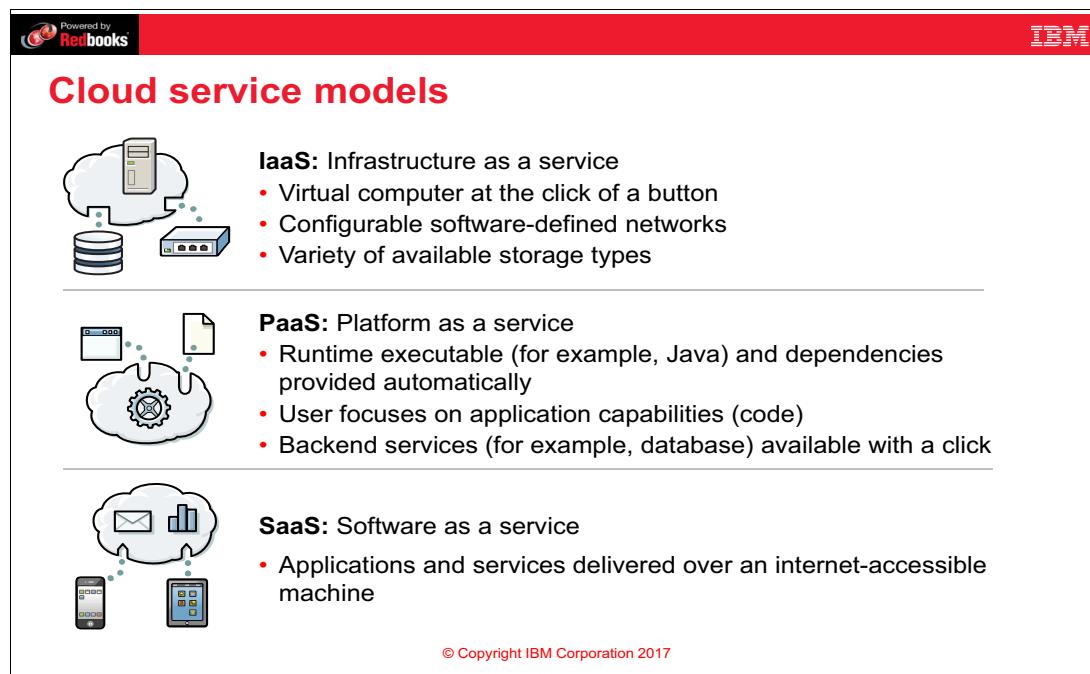


Figure 1-5 Cloud service models

Notes:

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

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.

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

1.8 Infrastructure as a service architecture

Powered by
Redbooks

IBM

Infrastructure as a service (IaaS) architecture

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



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

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

Notes:

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

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

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

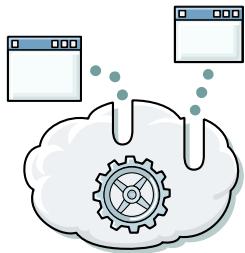
1.9 Platform as a service (PaaS) architecture

Powered by Redbooks

IBM

Platform as a service (PaaS) architecture

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



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

Figure 1-7 Platform as a service (PaaS) architecture

Notes:

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

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

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

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

1.10 Software as a service (SaaS) architecture

Powered by Redbooks

IBM

Software as a service (SaaS) architecture

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



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

Figure 1-8 Software as a service (SaaS) architecture

Notes:

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

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

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

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

1.11 Provider and client responsibilities

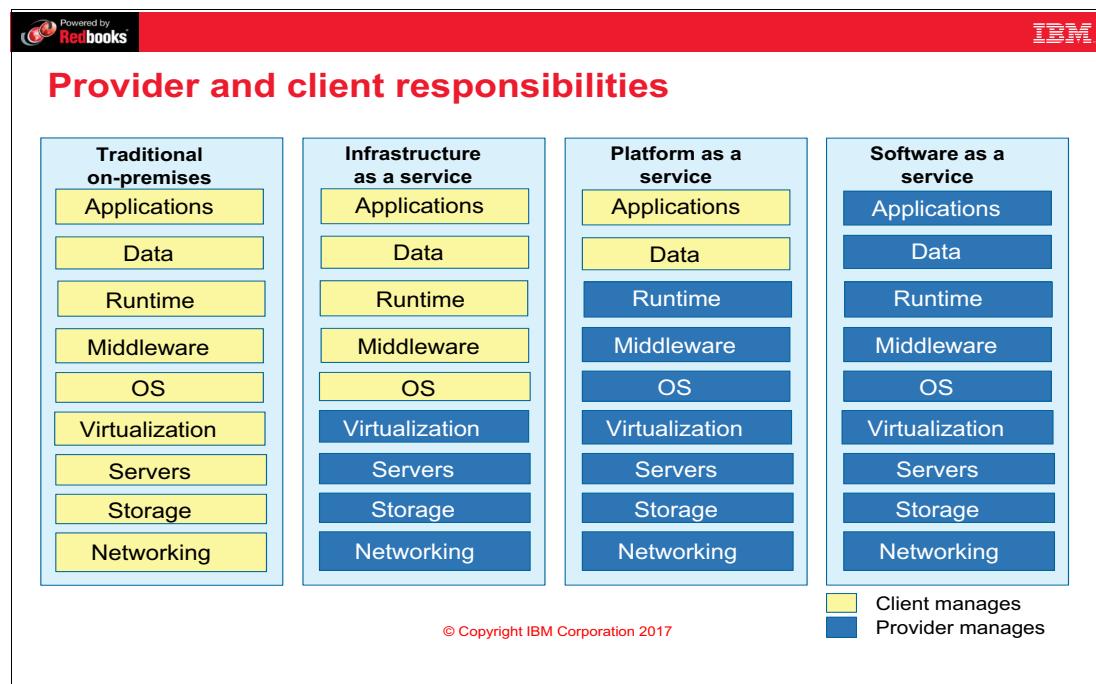


Figure 1-9 Provider and client responsibilities

Notes:

Figure 1-9 shows the split between the provider and client responsibilities when dealing with on-premises or “as a service” scenarios.

Typically, the cost decreases as you move to the right in the scenarios that are shown in Figure 1-9; however, the flexibility also is reduced.

Organizations or departments within an organization make their own cost-based decision about which delivery model to use for individual applications or projects.

Most enterprises end up using some combination of all of the models that are shown in Figure 1-9.

1.12 Cloud computing: Benefits for developers



Cloud computing: Benefits for developers

- Readily available sandbox and production environments:
 - Free trials offered with most products
 - Pre-built templates help developers get started fast
 - Easier to understand the application lifecycle
 - Environment set up 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 2017

Figure 1-10 Cloud computing: Benefits for developers

Notes:

Cloud computing brings the following benefits to developers:

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

1.13 IBM Cloud

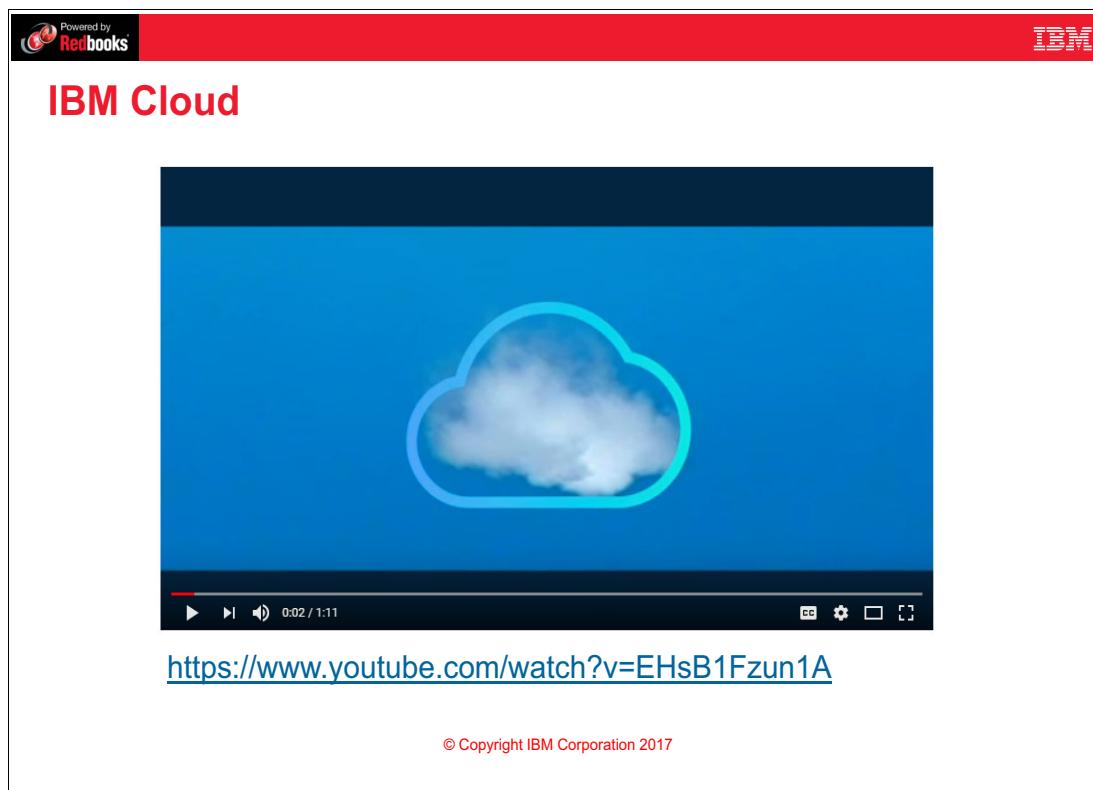


Figure 1-11 IBM Cloud

1.14 What is IBM Cloud?

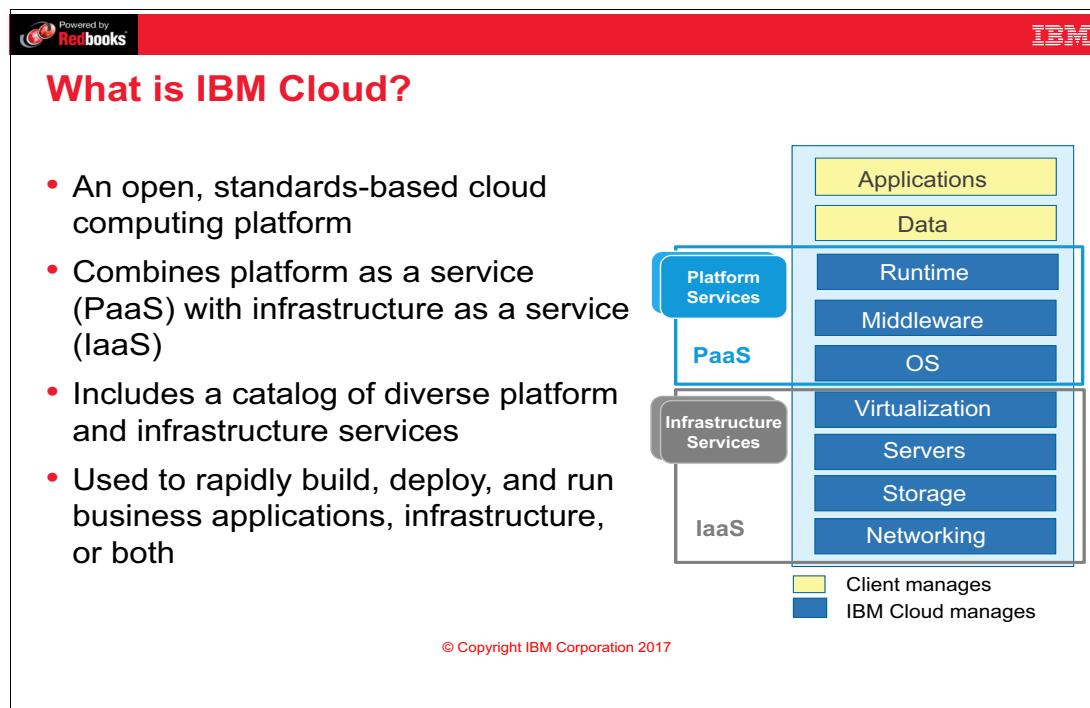


Figure 1-12 What is IBM Cloud?

Notes:

IBM Cloud is an open cloud computing platform that combines platform as a service (PaaS) with infrastructure as a service (IaaS), and includes a catalog of diverse cloud services, which can be used to rapidly build and deploy business applications or infrastructure.

As a PaaS, it provides developers access to IBM software for integration, security, transactions, and other key functions, and software from IBM Business Partners.

The application types can range from web, mobile, big data, and smart devices to the Internet of Things.

As an IaaS, it allows developers fine-grained control over the infrastructure on which their apps are deployed. Developers can deploy high-performance, bare-metal servers, virtual servers, containers, and cloud storage, in IBM Cloud data center locations around the world.

1.15 IaaS from IBM Cloud

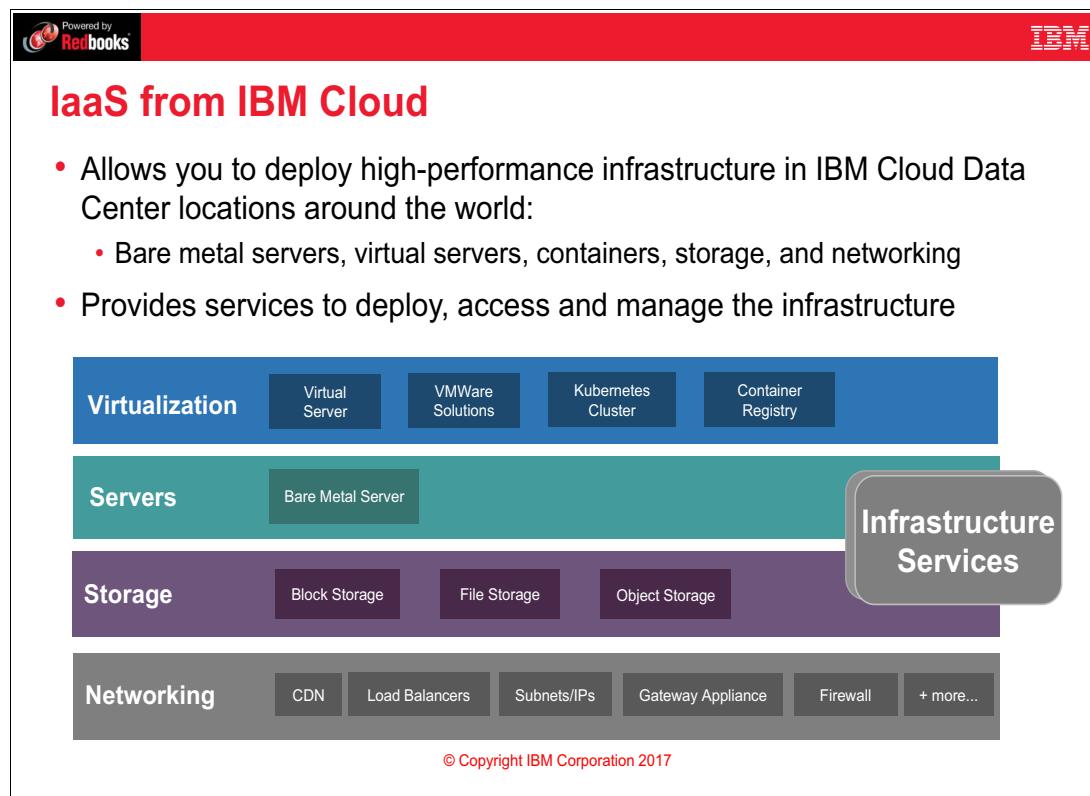


Figure 1-13 IaaS from IBM Cloud

Notes:

IBM Cloud enables you to deploy high-performance compute and storage infrastructure in nearly 60 IBM Cloud data centers around the world that are automated and standardized to provide a seamless global platform for cloud resources.

In addition to virtual servers, IBM Cloud offers bare metal servers, which provide the raw horsepower that many organizations require for processor-intensive and disk I/O-intensive workloads. Many organizations favor IBM Cloud because of the easy access it provides to bare metal servers.

IBM Cloud also allows you to deploy containers, storage, and networking resources across the world-wide data centers.

A catalog of services enables you to deploy, access, and manage the deployed infrastructure.

Note: The services that are available in IBM Cloud can change.

1.16 PaaS from IBM Cloud, Part 1

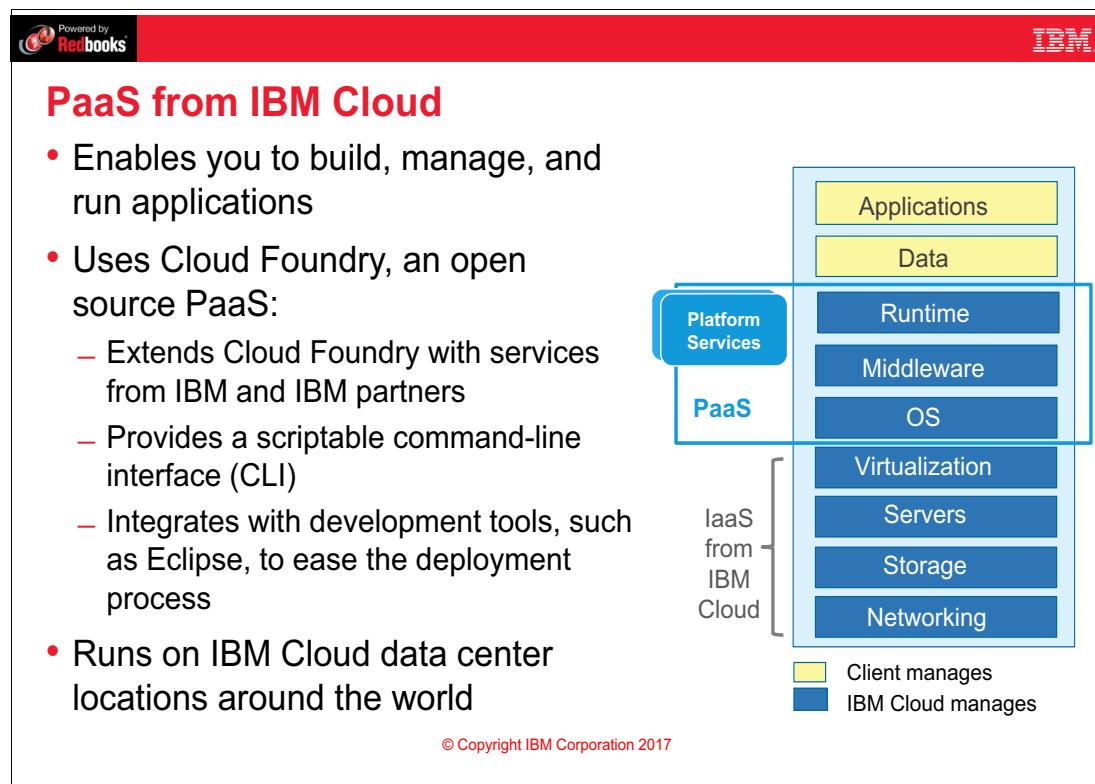


Figure 1-14 PaaS from Cloud

Notes:

As a PaaS provider, IBM Cloud allows you to build, manage, and run applications, such as web, mobile, big data, smart devices, and Internet of Things.

IBM Cloud uses Cloud Foundry, which is an open platform as a service offering that provides a choice of clouds, frameworks, and application services.

Cloud Foundry provides the monitoring, deployment, and logging tools for hosting apps.

IBM Cloud also adds the following enhancements to Cloud Foundry:

- Extends Cloud Foundry with services from IBM and IBM partners.
- Provides a scriptable command-line interface (CLI)
- Provides integration with development tools to ease the deployment process. DevOps services provide an online code editor, a build pipeline, and a version control system.

IBM Cloud runs on IBM Cloud data centers locations around the world.

1.17 PaaS from IBM Cloud, Part 2

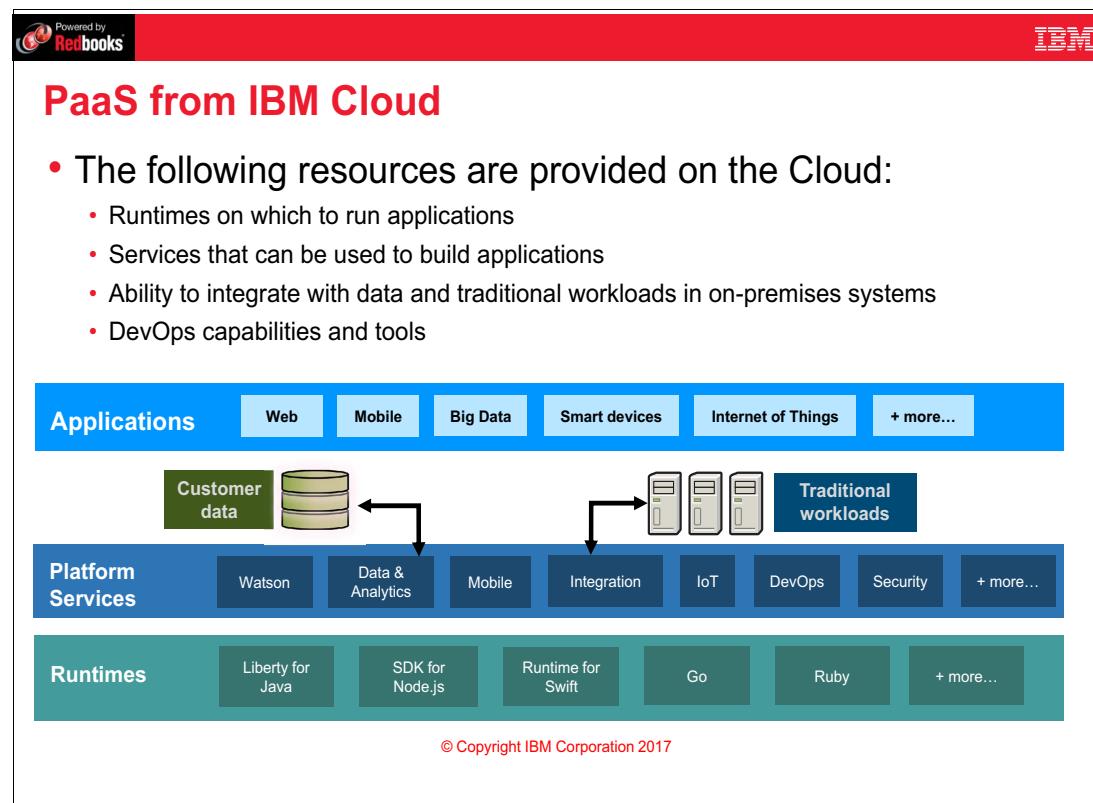


Figure 1-15 PaaS from IBM Cloud

Notes:

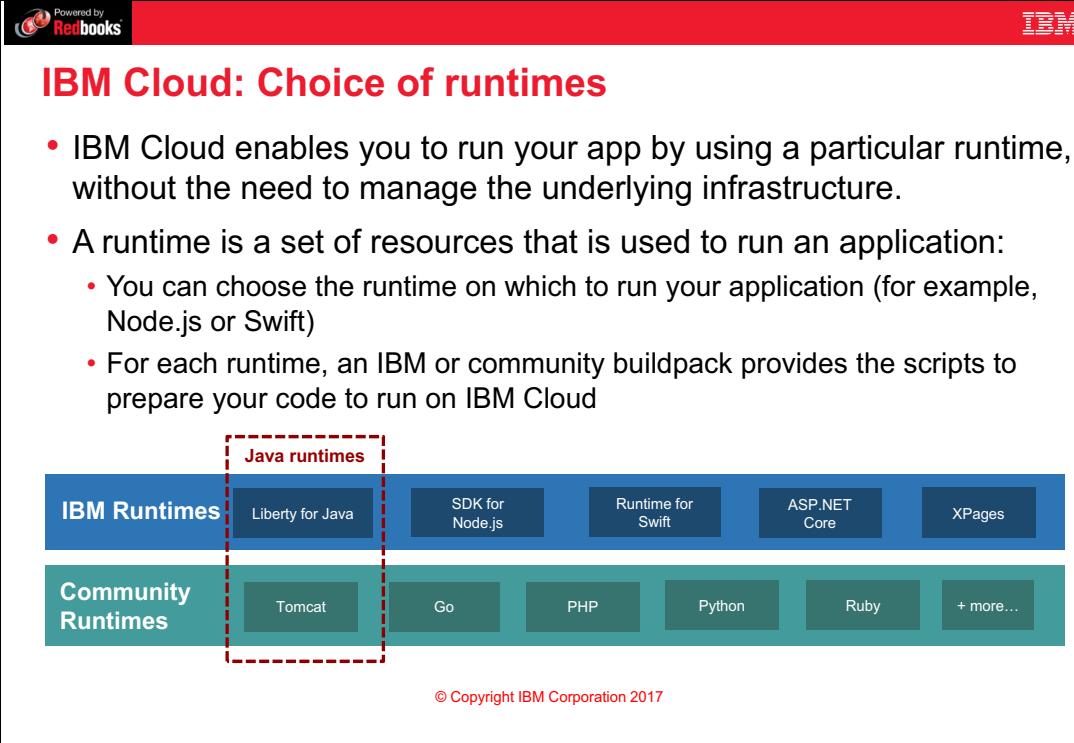
IBM Cloud enables application developers to focus on application capabilities by providing the following resources on the cloud:

- ▶ Runtimes on which to run applications
- ▶ A catalog of selectable services, such as databases, mobile support, analytics, artificial intelligence, and security, which are used to build applications
- ▶ Ability to integrate with data from the organization and traditional workloads that are running in on-premises systems
- ▶ DevOps capabilities and tools, including code-editors, version control, deployment pipelines, and hosting, monitoring, and scaling apps

Integration services allow applications to access traditional workloads that are running in the organization's on-premises environment.

Note: The available runtimes in IBM Cloud can change.

1.18 IBM Cloud: Choice of runtimes



The diagram illustrates the choice of runtimes available on IBM Cloud. It is organized into two main sections: 'IBM Runtimes' and 'Community Runtimes'. The 'IBM Runtimes' section includes Liberty for Java, SDK for Node.js, Runtime for Swift, ASP.NET Core, and XPages. The 'Community Runtimes' section includes Tomcat, Go, PHP, Python, Ruby, and '+ more...'. A red dashed box highlights the 'Java runtimes' category, which contains Liberty for Java and Tomcat.

© Copyright IBM Corporation 2017

Figure 1-16

Notes:

With IBM Cloud, developers are given a choice of runtimes on which to run their applications.

A *runtime* is a set of resources that is used to run an application. Each runtime features an associated buildpack, which is a collection of scripts that prepare your code to run on IBM Cloud.

Runtimes are provided by IBM or through Community Buildpacks. Consider the following points:

- ▶ The IBM runtimes include Liberty for Java, SDK for Node.js, and Runtime for Swift.
- ▶ IBM Cloud and Cloud Foundry support more runtimes through the Community Buildpacks. This open-source community features written buildpacks for other runtimes, such as Go, PHP, Python, Ruby, and Tomcat. For more information, see *Using community buildpacks*, which is available at the following website:

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

The following runtimes are available for Java:

- ▶ Tomcat: An open source Java web application server.
- ▶ Liberty for Java: IBM WebSphere Liberty is a Java EE application server which can deploy any Tomcat application. It also offers support for more Java web features; for example, Message Beans and JMX.

Note: The available runtimes in IBM Cloud can change.

1.19 IBM Cloud: Services

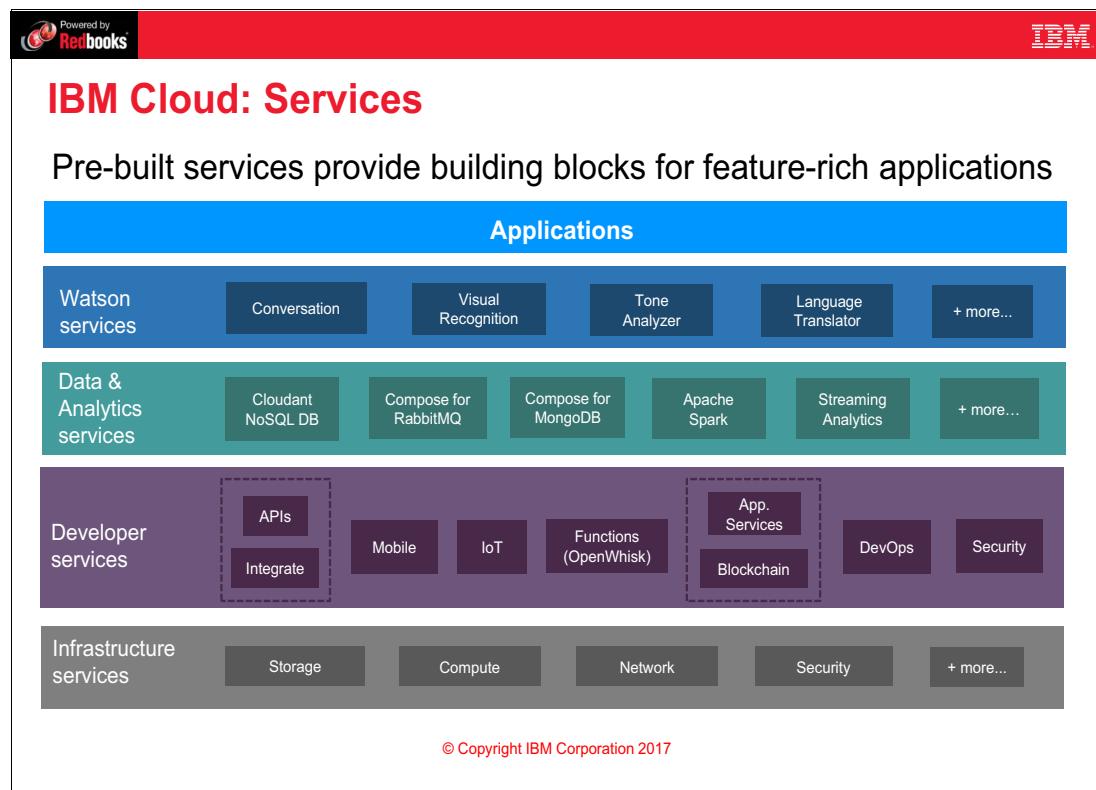


Figure 1-17 IBM Cloud: Services

Notes:

IBM Cloud provides a broad range of pre-built services (from IBM and third-party providers) that can be used when assembling your application.

Watson services enable you to add the power of artificial intelligence to your application with speech, vision, and natural language processing APIs.

Data & Analytics services help you to get data from integrated cloud databases, build data-driven applications, and analyze your data.

In addition, services in the following categories help you to develop key features within your application:

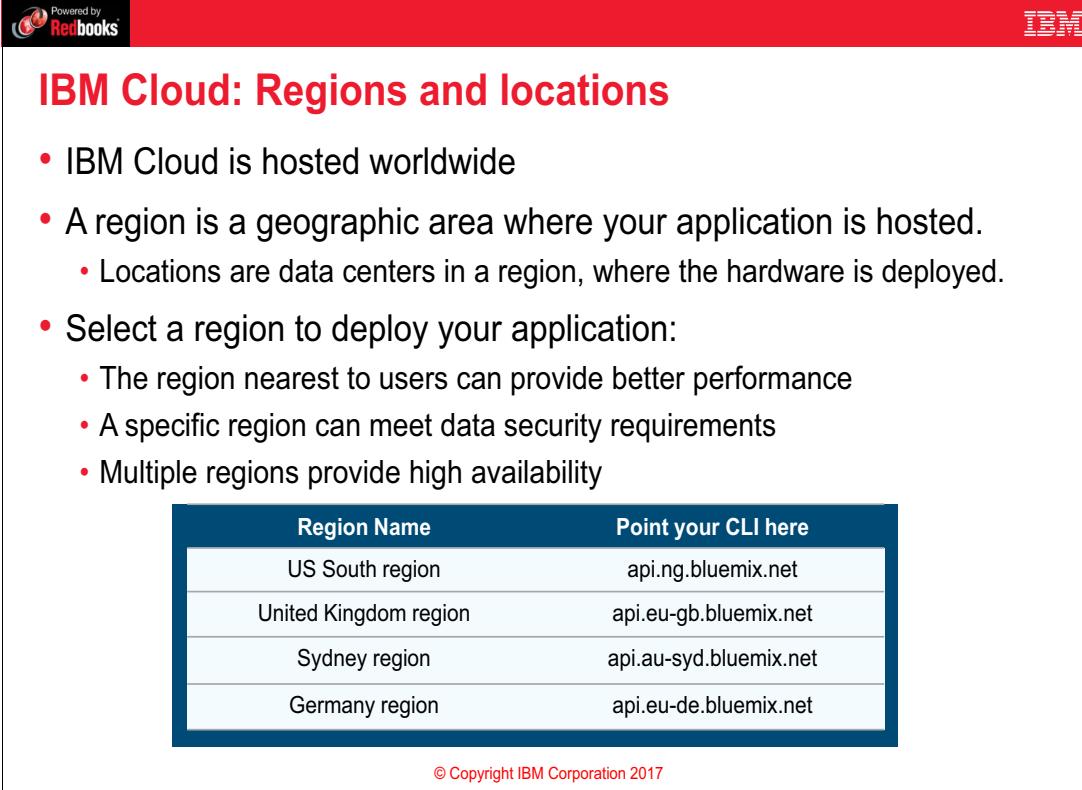
- ▶ Integration services:
 - APIs: Create, manage, enforce, and run APIs
 - Integrate: Access traditional workloads running in the organization's on-premises environment
- ▶ Mobile: Use mobile backend infrastructure to build, monitor, and test mobile apps.
- ▶ Internet of Things: Communicate with connected devices, sensors, and gateways.
- ▶ Functions: Execute in response to incoming events (based on Apache OpenWhisk).
- ▶ Application Services: Many application services, such as Blockchain, Message Hub, WebSphere Application Server, Business Rules, and other application services on the cloud.

- ▶ DevOps: Tools to help innovate new applications faster and cheaper.
- ▶ Security: Build security into your application design.

Infrastructure services help you to manage the underlying infrastructure your application runs on.

Note: The services available in IBM Cloud can change.

1.20 IBM Cloud: Regions and locations



The slide has a red header bar with the Redbooks logo on the left and the IBM logo on the right. The main title "IBM Cloud: Regions and locations" is in bold red text at the top. Below it is a bulleted list of points about regions and locations. At the bottom is a table showing region names and their corresponding CLI endpoints.

Region Name	Point your CLI here
US South region	api.ng.bluemix.net
United Kingdom region	api.eu-gb.bluemix.net
Sydney region	api.au-syd.bluemix.net
Germany region	api.eu-de.bluemix.net

© Copyright IBM Corporation 2017

Figure 1-18 IBM Cloud: Regions and locations

Notes:

A region is a defined geographical territory to which you can deploy your applications.

Locations are data centers within the region in which hardware is deployed. Locations are not specified during application deployment.

You can choose to deploy your application to a single region or multiple regions. Consider the following points:

- ▶ For low application latency, select the region that is nearest to your users.
- ▶ To meet certain countries' data security requirements, select the region where you are required to store the application data.
- ▶ For high availability, select multiple regions. If your application fails in one region, it is still available on another region.

Note: Use similar criteria when choosing a location to deploy infrastructure.

Within the IBM Cloud console, the region is automatically set to the closest healthy region. To switch to another region, click the user account link, expand the Region menu and select the region. If you use a Lite Account (For more information about IBM Cloud account types, see Unit 2), you can use only one region for your applications and services.

IBM Cloud offers a command line tool that is called IBM Cloud Command Line Interface (CLI). When this tool is used, use the second column that is shown in Figure 1-18 on page 21 “Point your CLI here” to specify the target region.

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

1.21 How Cloud Foundry works with IBM Cloud

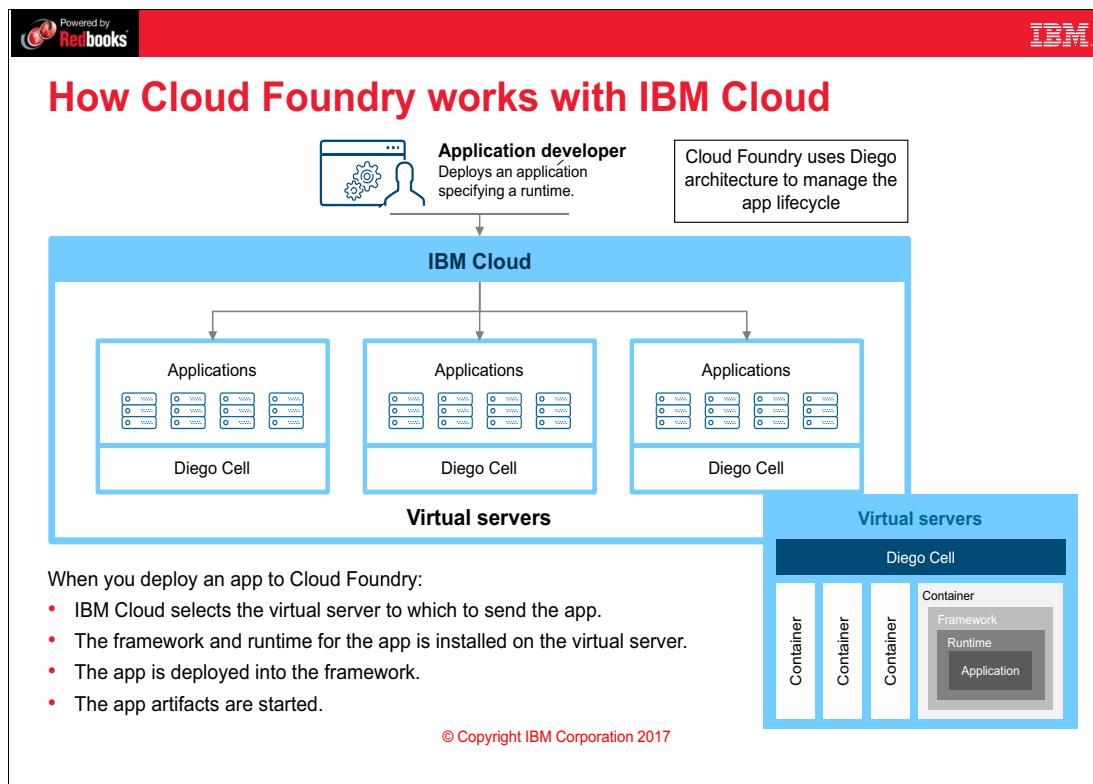


Figure 1-19 How Cloud Foundry works with IBM Cloud

Notes:

Cloud Foundry uses the Diego architecture to manage the application life-cycle for deploying and starting the application on Cloud Foundry.

As a developer, you interact with IBM Cloud by using your web browser or the CLI.

Application Deployment

When you create an app and deploy it to Cloud Foundry, the IBM Cloud environment determines an appropriate virtual server to which to send the app.

The virtual server is chosen based on several factors, including the hardware/file system that is required by the app, and the load that is on the machine.

After a virtual server is chosen, the following tasks are completed:

- ▶ An application manager on the virtual server installs the appropriate framework and runtime for the app.
- ▶ The app is deployed into that framework.
- ▶ When the deployment completes, the application artifacts are started.

When the app is deployed, if it has a web interface (as for a Java web app), or other REST-based services (such as mobile services exposed publicly to the mobile app), users of the app can communicate with it by using normal HTTP requests.

Virtual Servers

Each virtual server has multiple apps deployed on it.

In each virtual server, an application manager communicates with the rest of the IBM Cloud infrastructure and manages the apps that are deployed to this virtual server.

Each virtual server features containers that are used to separate and protect apps. In each container, IBM Cloud installs the appropriate framework and runtime that are required for each app.

1.22 Unit summary

The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top left of this area is the title "Unit summary" in red. Below it is a bulleted list of nine items, each starting with a red dot. The list covers various aspects of cloud computing, from definitions to specific services like IBM Cloud and Cloud Foundry.

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

© Copyright IBM Corporation 2017

Figure 1-20 Unit summary

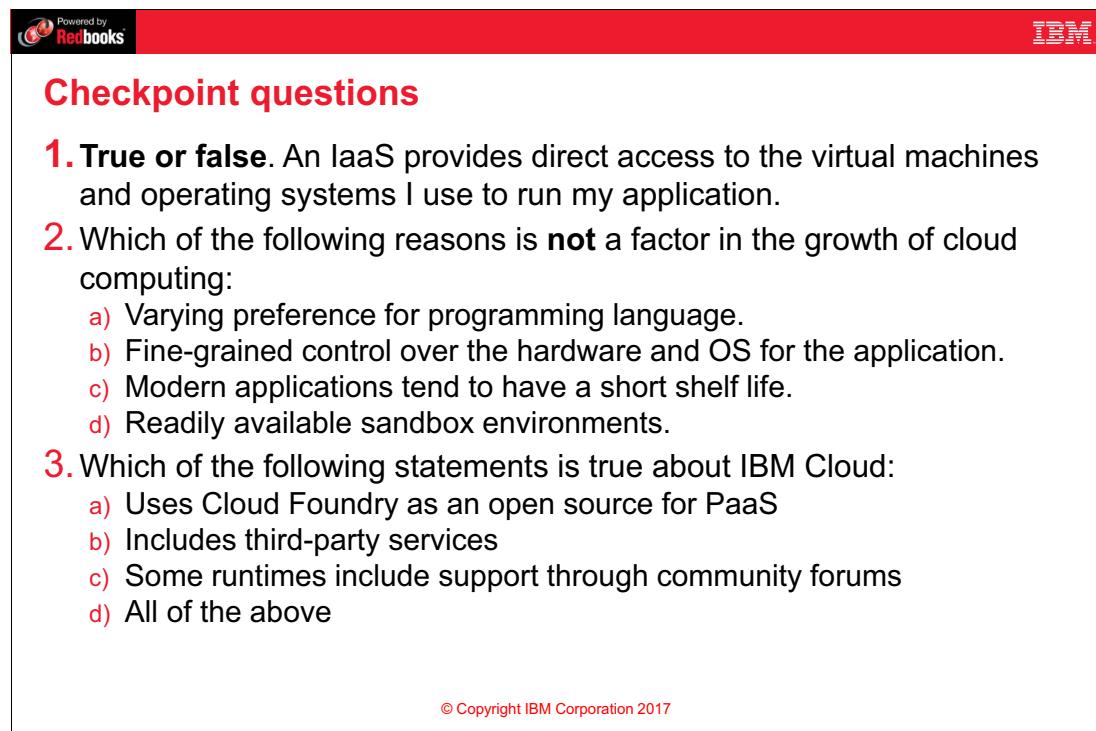
Notes:

In this unit, we defined cloud computing and described:

- ▶ The factors that lead to the adoption of cloud computing
- ▶ The choices that developers have when creating cloud applications
- ▶ IaaS, PaaS, and SaaS
- ▶ IBM Cloud
- ▶ How Cloud Foundry works with IBM Cloud

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

1.23 Checkpoint questions



The slide is titled "Checkpoint questions" in red text. It features a list of three numbered questions. Question 1 is a true/false statement. Questions 2 and 3 are multiple-choice questions with four options each. The slide is branded with the Redbooks logo at the top left and the IBM logo at the top right. A copyright notice at the bottom indicates it was created for IBM Corporation in 2017.

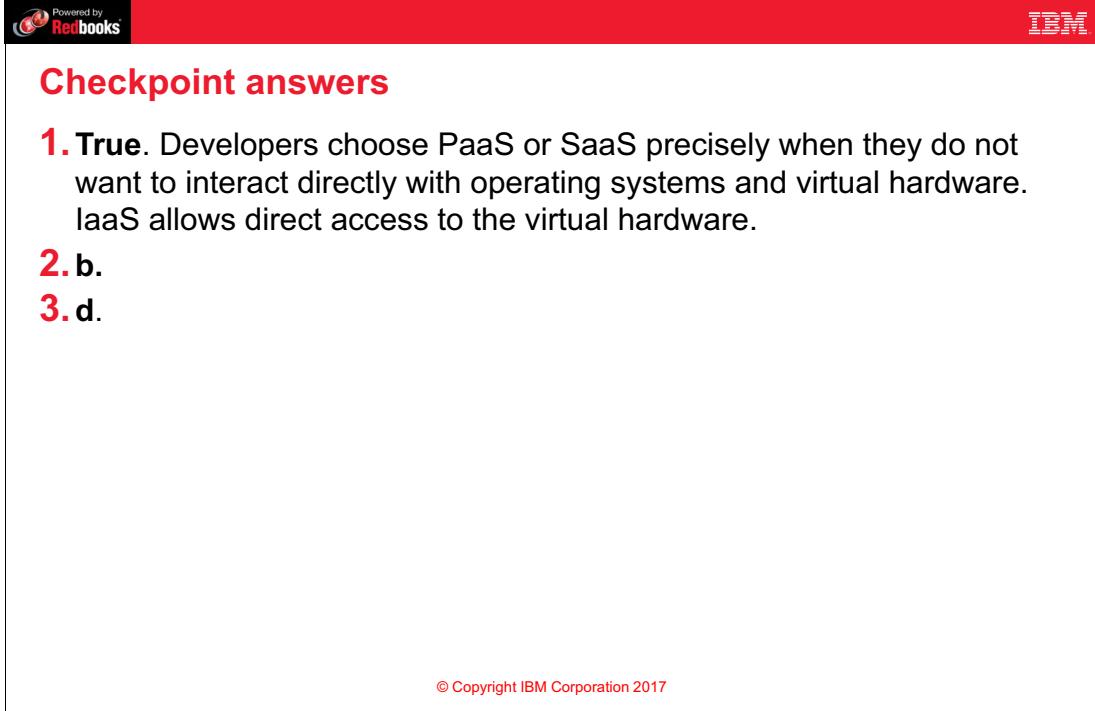
Checkpoint questions

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

© Copyright IBM Corporation 2017

Figure 1-21 Checkpoint questions

1.24 Checkpoint answers



The slide is titled "Checkpoint answers" in red text. It features the Redbooks logo and the IBM logo at the top. The content includes a numbered list of three items. A copyright notice at the bottom right indicates it was created in 2017.

Checkpoint answers

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

2. b.

3. d.

© Copyright IBM Corporation 2017

Figure 1-22 Checkpoint answers



Getting started with IBM Cloud

This unit covers the following topics:

- ▶ Understanding what you can build in IBM Cloud
- ▶ Creating and managing an IBM Cloud application
- ▶ Managing your IBM Cloud users and resources

2.1 What you should be able to do

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

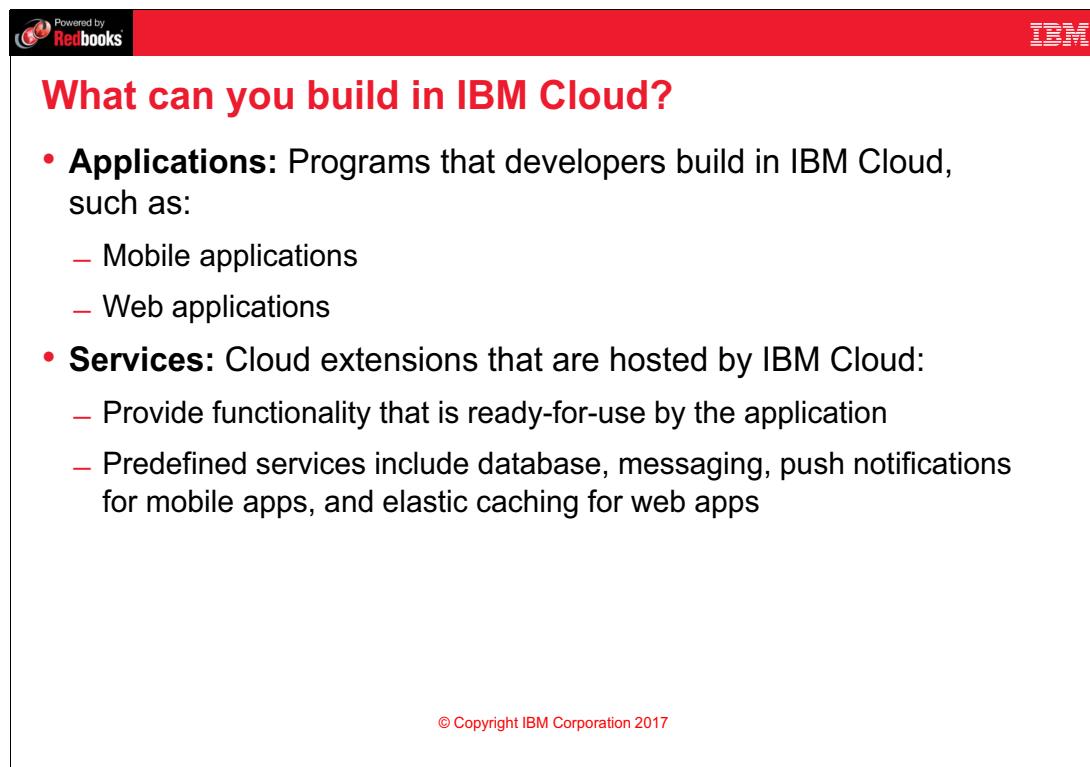
- ▶ IBM Cloud
- ▶ What you can build in IBM Cloud
- ▶ How to create an application in IBM Cloud
- ▶ The IBM Cloud dashboard, catalog, and documentation features
- ▶ How the application route is used to test an application from the browser
- ▶ How to create services in IBM Cloud
- ▶ How to bind services to an application in IBM Cloud
- ▶ The environmental variables that are used with IBM Cloud services
- ▶ How to manage your IBM Cloud users and resources

2.2 References

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

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

2.3 What can you build in IBM Cloud?



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "What can you build in IBM Cloud?" is in red. Below it is a bulleted list defining applications and services.

• Applications: Programs that developers build in IBM Cloud, such as:

- Mobile applications
- Web applications

• Services: Cloud extensions that are hosted by IBM Cloud:

- Provide functionality that is ready-for-use by the application
- Predefined services include database, messaging, push notifications for mobile apps, and elastic caching for web apps

© Copyright IBM Corporation 2017

Figure 2-1 What can you build in IBM Cloud?

Notes:

The following components can be built in IBM Cloud:

► Applications:

In IBM Cloud you can build applications, which are the programs that developers build in the Cloud Foundry environment.

You can build mobile apps that run outside the IBM Cloud environment and use services to which the mobile apps are exposed.

Web apps consist of the code that is required to be run or referenced at runtime.

IBM Cloud can also host application code that the developer prefers to run on a back-end server in a container-based environment.

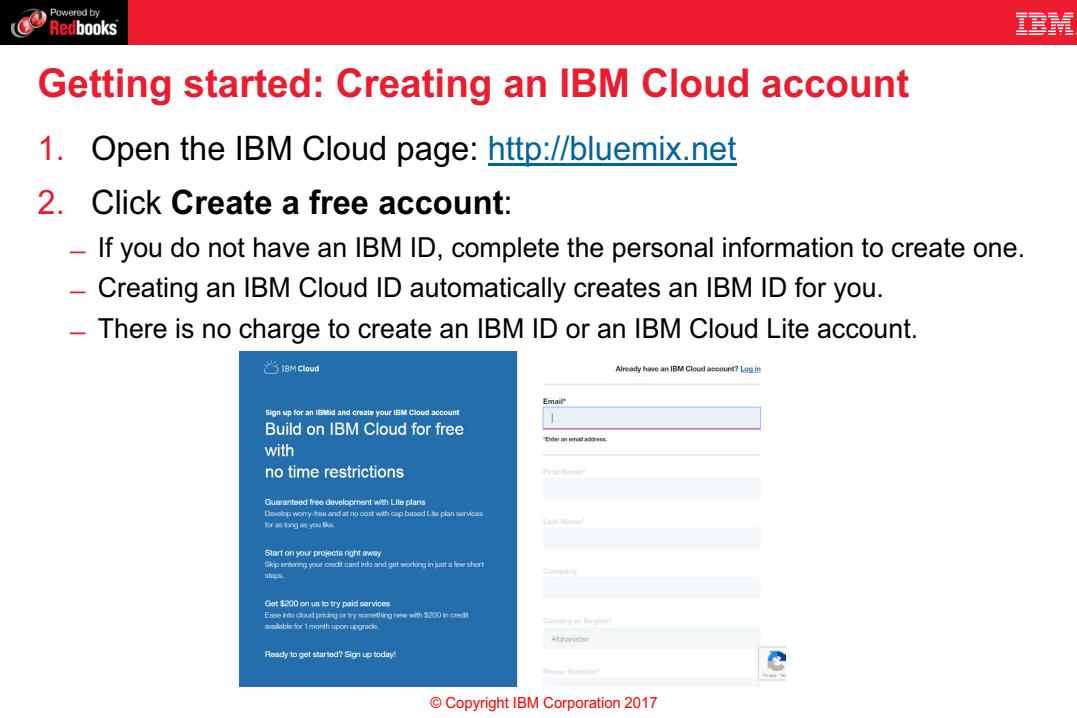
► Services:

A service is a cloud extension that is hosted by IBM Cloud. The service provides functionality that is ready-for-use by the running code of the application.

The predefined services that are provided by IBM Cloud include database, messaging, push notifications for mobile apps, and elastic caching for web applications.

You can create your own services in IBM Cloud. The services can be simple utilities, such as the functions you might see in a runtime library, or complex business logic that you might see in a business process modeling service or a database.

2.4 Getting started: Creating an IBM Cloud account



The screenshot shows the 'Getting started: Creating an IBM Cloud account' page. At the top left is the Redbooks logo, and at the top right is the IBM logo. The main heading is 'Getting started: Creating an IBM Cloud account'. Below it, a numbered list provides instructions:

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

The right side of the screenshot displays the IBM Cloud sign-up form. It includes fields for Email*, First Name*, Last Name*, Company, Country or Region*, and Phone Number*. A note at the bottom right indicates '© Copyright IBM Corporation 2017'.

Figure 2-2 Getting started: Creating an IBM Cloud account

Notes:

Before you work on any of the exercises in this course, you must sign up for an IBM Cloud account.

You can register for a free IBM Cloud Lite account.

For more information about IBM Cloud Lite account, see the following video:

<https://www.youtube.com/watch?v=0rMYXcbpHbI>

2.5 Getting Started: IBM Cloud accounts

The screenshot shows a comparison chart titled "IBM Cloud account comparison". The chart has three columns: Lite, Pay-As-You-Go, and Subscription. The rows list various features and their availability across the three account types. A red border surrounds the chart area.

	Lite	Pay-As-You-Go	Subscription
Access to free Cloud Foundry memory	256	512	512
Access to Lite service plans	✓	✓	✓
Access to all free plans		✓	✓
Access to the full catalog		✓	✓
No time restrictions	✓	✓	✓
Guaranteed zero cost	✓		
Negotiated pricing			✓
Best for learning or building POCs	✓	✓	
Fit for production use cases		✓	✓

© Copyright IBM Corporation 2017

Figure 2-3 Getting Started: IBM Cloud accounts

Notes:

When you sign up for a free IBM Cloud Account, you start with the Lite account. A Lite account includes the following main features:

- ▶ The account is free (no credit card required).
- ▶ The account never expires.
- ▶ You receive 256 MB of memory for your Cloud Foundry applications.
- ▶ You have access to a specific services that are tagged as Lite.

A Pay-As-You-Go account requires entering your credit card information. With a Pay-As-You-Go account, you are eligible for free runtime and service allowances. If you use more than the free allowance, you receive a monthly IBM Cloud invoice. This invoice provides 512 MB of free memory for your Cloud Foundry applications. In addition, all of the free services that are in the catalog are available to you. Charges are based on your use of IBM Cloud compute and services.

With a Subscription account, you commit to a minimum spending amount each month and receive a subscription discount that is applied to that minimum charge. You also pay for any usage that exceeds the minimum spending amount.

2.6 IBM Cloud catalog

The screenshot shows the IBM Cloud catalog interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "IBM Cloud catalog" is displayed in a large, bold, red font. A bulleted list follows, describing the capabilities of the catalog:

- IBM Cloud enables you to create resources listed in the catalog, such as Apps, services, VMs, and containers.
- The IBM Cloud catalog organizes the resources as:
 - Infrastructure:

Under the "Infrastructure" heading, there are several tabs: Infrastructure, Compute, Storage, Network, Security, Containers, and VMWare. The "Infrastructure" tab is highlighted with a thicker border. Below this section, another bulleted list continues:

- Platform:

Under the "Platform" heading, there are two rows of categories. The top row contains three items: Apps, Cloud Foundry Apps, and Boilerplates. The bottom row contains several items: Services, Watson, Data & Analytics, Mobile, Functions, IoT, and + more... .

© Copyright IBM Corporation 2017

Figure 2-4 IBM Cloud catalog

Notes:

You can create resources in IBM Cloud. *Resources* are entities in IBM Cloud catalog, such as Cloudant, a Cloud Foundry application, virtual machine, or container.

IBM Cloud catalog features the following categories of resources:

- Infrastructure: Resources that provide fine-grained control over the computing infrastructure
- Platform: Resources help you rapidly build and deploy your applications. Consider the following points:
 - Applications can be built by using Cloud Foundry Apps or Boilerplates, which provide the runtimes and optionally other services to create your applications.
 - Services enable you to easily add relevant functionality to your application.

2.7 IBM Cloud: Infrastructure

Powered by Redbooks

IBM Cloud catalog: Infrastructure

- Compute
- Storage
- Network
- Security
- Container
- VMWare

IBM Cloud

All Categories >

Infrastructure

Compute

Bare Metal Server

Virtual Server

Storage

Block Storage

File Storage

Object Storage

© Copyright IBM Corporation 2017

Figure 2-5 IBM Cloud: Infrastructure

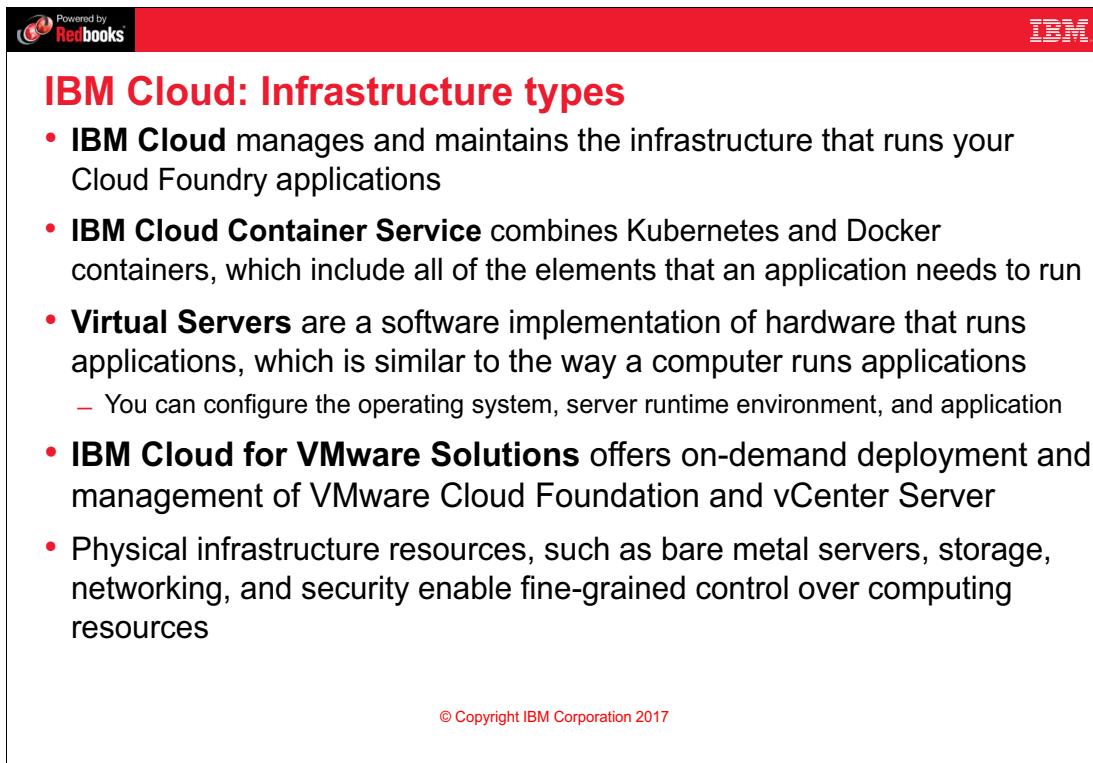
Notes:

You can select infrastructure resources in one of the categories shown in Figure 2-5. Consider the following points:

- ▶ You can order physical bare metal servers and virtual servers in the Compute category.
- ▶ The Container and VMWare categories provide alternative ways to run and manage your application. Although they allow more control over the underlying infrastructure and configuration, they require more work by the developer.
- ▶ For fine-grained control over computing resources, you can order physical resources from the remaining categories.

Note: The Lite tag on the Object Storage service means that this service is available for the users who registered with an IBM Cloud Lite account. The rest of the services are available to the users who have a Pay-As-You-Go account, or a subscription account.

2.8 IBM Cloud: Infrastructure types



The slide is titled "IBM Cloud: Infrastructure types" in red. It features a bulleted list of infrastructure types managed by IBM Cloud, including Cloud Foundry applications, Container Service, Virtual Servers, VMware Solutions, and physical infrastructure resources. The slide is powered by Redbooks and includes the IBM logo.

- **IBM Cloud** manages and maintains the infrastructure that runs your Cloud Foundry applications
- **IBM Cloud Container Service** combines Kubernetes and Docker containers, which include all of the elements that an application needs to run
- **Virtual Servers** are a software implementation of hardware that runs applications, which is similar to the way a computer runs applications
 - You can configure the operating system, server runtime environment, and application
- **IBM Cloud for VMware Solutions** offers on-demand deployment and management of VMware Cloud Foundation and vCenter Server
- Physical infrastructure resources, such as bare metal servers, storage, networking, and security enable fine-grained control over computing resources

© Copyright IBM Corporation 2017

Figure 2-6 IBM Cloud: Infrastructure types

Notes:

IBM Cloud manages and maintains the infrastructure that runs your Cloud Foundry applications.

IBM Cloud supports the following alternative infrastructure types:

- ▶ IBM Containers provide more control over the computing infrastructure to run the application and services through Docker-like APIs. Docker packages software into standardized units that are called *containers*. Containers are described later in this unit.
- ▶ Virtual servers are software implementation of hardware that runs applications, which is similar to the way a computer runs applications. You can configure the operating system, server runtime environment, and application. If you want to control the infrastructure down to the operating system level, consider the use of virtual servers.
- ▶ IBM Cloud for VMware Solutions offers on-demand deployment and management of VMware Cloud Foundation and vCenter Server.
- ▶ You also can deploy and use physical infrastructure resources, such as bare metal servers, storage, networking, and security, for fine-grained control over computing resources.

Most developers in the enterprise likely use some mix of these types of infrastructure. The exercises in this course introduce you to the Cloud Foundry infrastructure in IBM Cloud. As such, the use of containers, virtual servers, IBM Cloud for VMware Solutions, and physical infrastructure resources are out of scope of this course. This course focuses on Cloud Foundry applications.

2.9 Understanding containers and virtual servers

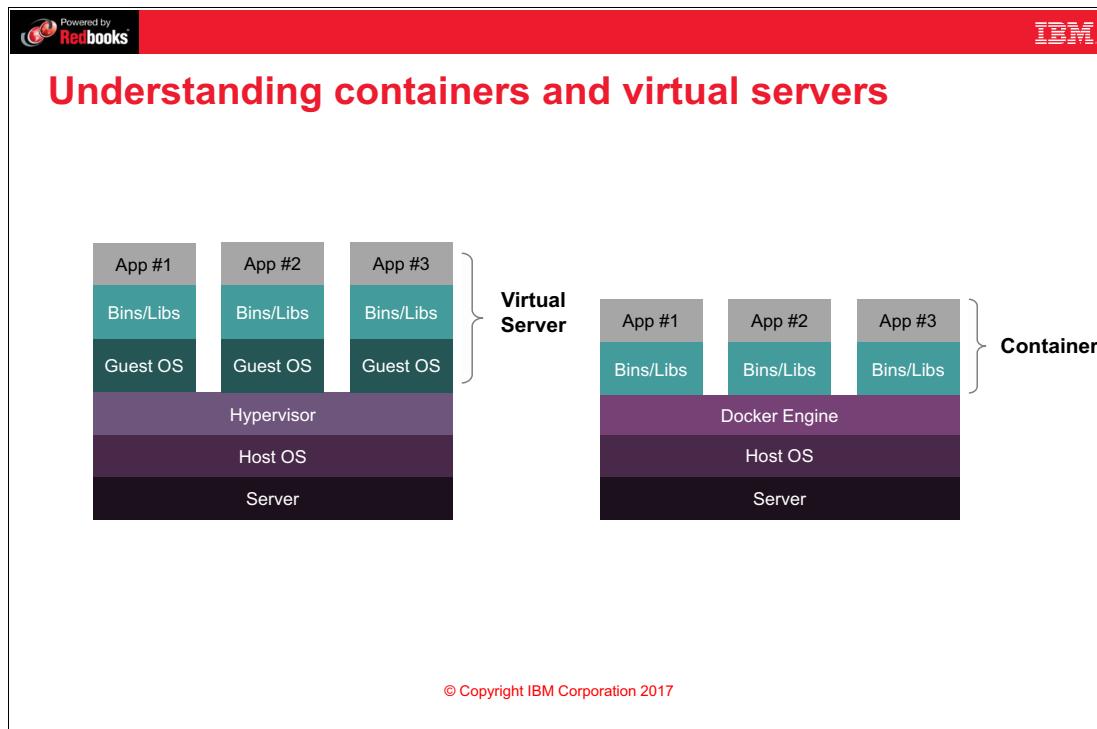


Figure 2-7 Understanding containers and virtual servers

Notes:

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

Virtual servers, or virtual machines (VM), are created by 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 on a VM is unaware that it is running on VM.

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

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

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

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

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

2.10 IBM Cloud catalog: Containers and virtual servers

The screenshot shows the 'IBM Cloud catalog: Containers and virtual servers' page. It features three main service cards:

- Kubernetes Cluster**: Deploy highly available apps to a cluster of compute hosts. Options: Lite, IBM.
- Container Registry**: Store and distribute Docker images by using this managed private registry. Make informed decisions. Options: Lite, IBM.
- Virtual Server**: Our virtual servers deliver a higher degree of transparency, predictability, and automation for all. Option: IBM.

At the bottom of the page is the copyright notice: © Copyright IBM Corporation 2017.

Figure 2-8 IBM Cloud catalog: Containers and virtual servers

Notes:

IBM Cloud supports the following alternative infrastructure types:

- ▶ Containers provide more control over the computing infrastructure to run the application and services through Docker-like APIs. Docker packages software into standardized units that are called *containers*. Containers include all of the elements that an application needs to run. IBM Cloud container service (Kubernetes cluster) combines Docker and Kubernetes to deliver powerful tools, an intuitive user experience, built-in security and isolation to automate the deployment, operation, scaling, and monitoring of containerized applications over a cluster. You can create a Kubernetes cluster that consists of one or more VMs that are called *worker nodes*. Every worker node represents a compute host in which you can deploy, run, and manage containerized applications.

You can use and extend public images, such as the IBM Integration Bus V10 Developer Edition, which you can use to start developing your own integration solution.

Containers give you more control, but still do not require you to manage an operating system or network. IBM containers resemble the popular open source container engine Docker, but have some IBM Cloud specific features, such as their ability to integrate with the Cloud Foundry routers.

- ▶ Virtual servers are software implementation of hardware that runs applications that is similar to a computer. You can configure the operating system, server runtime environment, and application. If you want to control the infrastructure down to the operating system level, consider the use of Virtual servers.

2.11 IBM Cloud: Platform

The screenshot shows the IBM Cloud catalog interface. On the left, a sidebar lists 'Platform resources consist of:' with two main categories: 'Applications, such as Cloud Foundry Apps and Boilerplates' and 'Services, such as Functions and Mobile'. The 'Platform' category is highlighted with a blue border. The main content area is titled 'IBM Cloud catalog: Platform' and shows a grid of application starters. The grid includes sections for 'Boilerplates' (ASP.NET Core Cloudant Starter, Forms Experience Builder), 'Internet of Things' (Internet of Things Platform Starter, IoT for Electronics Starter), 'Java' (Java Cloudant Web Starter, Java Workload Scheduler Web Starter), and 'Mobile' (LoopBack Starter, MobileFirst Services Starter). Each starter has a small icon, a title, a brief description, and an 'IBM' badge.

Figure 2-9 IBM Cloud: Platform

Notes:

Platform resources help you rapidly build and deploy your applications.

Applications can be built by using Cloud Foundry Apps or Boilerplates, which provide the runtimes and optionally other services to create your applications.

Services enable you to easily add relevant functionality to your application.

2.12 IBM Cloud: Applications and services

The screenshot shows the IBM Cloud Application Catalog interface. At the top, there's a navigation bar with the IBM logo and a search bar. Below the navigation bar, the title "IBM Cloud: Applications and services" is displayed in red. To the left of the main content area, there's a sidebar with a "Powered by Redbooks" logo and some navigation links.

Cloud Foundry Applications (3) 1GB/1GB Used

Name	Region	CF Org	CF Space	Status
test-app-node	United Ki...	aazraq@...	dev	Stopped (0/1)
vy30t-java-staging	United Ki...	aazraq@...	dev	Running (1/1)
vy30t-xxx-node	United Ki...	aazraq@...	dev	Running (1/1)

20 Items per page | 1-3 of 3 items 1 of 1 pages < | >

Cloud Foundry Services (1) 2/4 Used

Name	Region	CF Org	CF Space
Cloudant NoSQL DB-cm	United King...	aazraq@egi...	dev

20 Items per page | 1-1 of 1 items 1 of 1 pages < | >

Services (2) ⓘ

Name	Location	Resource...	Plan	Details
App ID-o2	United Ki...	Default	Graduate...	Provisioned

© Copyright IBM Corporation 2017

Figure 2-10 IBM Cloud: Applications and services

Notes:

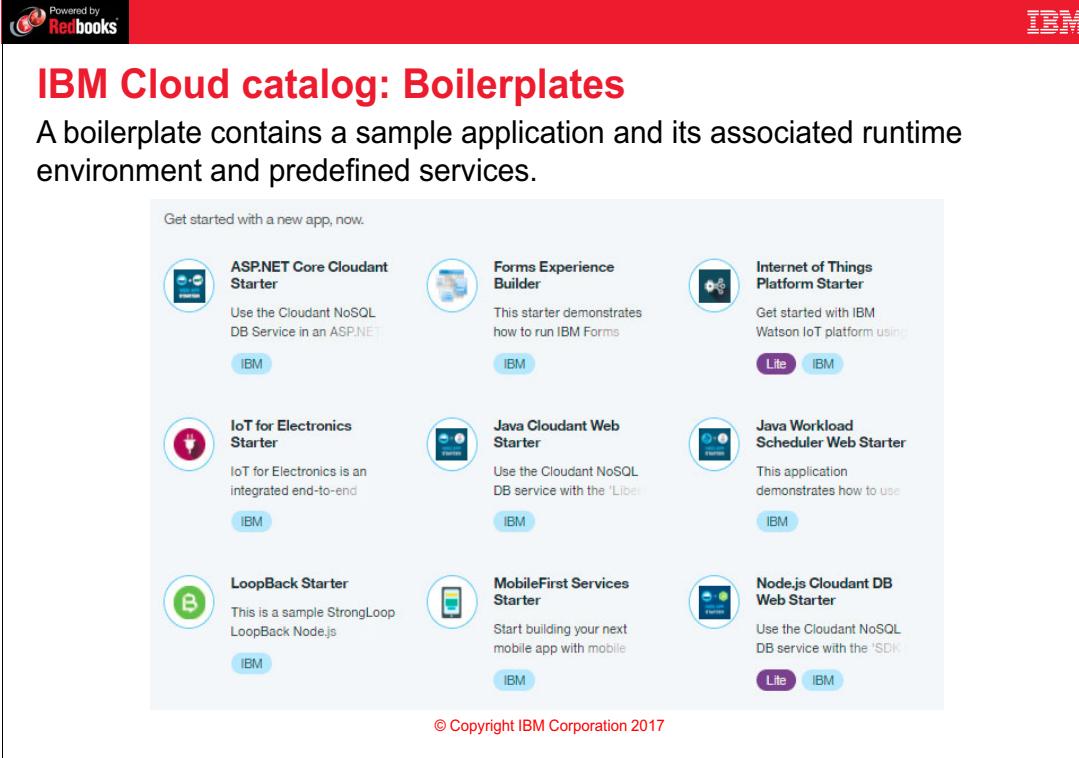
In IBM Cloud, you can build applications, which are the programs that developers build in the Cloud Foundry environment. You also can build mobile apps that run outside the IBM Cloud environment and use services to which the mobile apps are exposed. You can also build web apps that consist of the code that is required to be run or referenced at runtime.

A service is a cloud extension that is hosted by IBM Cloud. The service provides functionality that is ready-for-use by the application's running code.

Predefined services are provided by IBM Cloud, and include database, messaging, push notifications for mobile apps, and elastic caching for web apps. Predefined services are also provided by third parties.

You can create your own services in IBM Cloud. They can be simple utilities, such as the functions you might see in a runtime library, or complex business logic that you might see in a business process modeling service or a database.

2.13 IBM Cloud: Boilerplates



The screenshot shows the 'IBM Cloud catalog: Boilerplates' page. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title 'IBM Cloud catalog: Boilerplates' is displayed in large red font. A descriptive text follows: 'A boilerplate contains a sample application and its associated runtime environment and predefined services.' Below this, a section titled 'Get started with a new app, now.' lists nine different boilerplate starters, each with an icon, name, brief description, and an 'IBM' badge. The starters are:

- ASP.NET Core Cloudant Starter**: Use the Cloudant NoSQL DB Service in an ASP.NET application.
- Forms Experience Builder**: This starter demonstrates how to run IBM Forms.
- Internet of Things Platform Starter**: Get started with IBM Watson IoT platform using Cloudant.
- IoT for Electronics Starter**: IoT for Electronics is an integrated end-to-end solution.
- Java Cloudant Web Starter**: Use the Cloudant NoSQL DB service with the 'LibreOffice' Java API.
- Java Workload Scheduler Web Starter**: This application demonstrates how to use the Cloudant NoSQL DB service with the Java API.
- LoopBack Starter**: This is a sample StrongLoop LoopBack Node.js application.
- MobileFirst Services Starter**: Start building your next mobile app with mobile first services.
- Node.js Cloudant DB Web Starter**: Use the Cloudant NoSQL DB service with the 'SDK' Node.js API.

At the bottom of the page, a copyright notice reads: '© Copyright IBM Corporation 2017'

Figure 2-11 IBM Cloud: Boilerplates

Notes:

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

A boilerplate contains an application 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 MobileFirst Services Starter boilerplate to host mobile and web apps and accelerate development time of server-side scripts by using the mobile app template and SDK.

2.14 IBM Cloud catalog: Cloud Foundry Apps

The screenshot shows the IBM Cloud catalog interface for Cloud Foundry Apps. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below the header, the title "IBM Cloud catalog: Cloud Foundry Apps" is displayed in large red font. A sub-header "Deploy your app without managing underlying infrastructure." is present. The main content area lists nine runtime environments, each with a small icon, the runtime name, a brief description, and an "IBM" or "Community" badge:

- Liberty for Java™**: Develop, deploy, and scale Java web apps with ease. IBM WebSphere.
- SDK for Node.js™**: Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM.
- .NET Core**: Develop, deploy, and scale ASP.NET Core web apps with ease.
- Runtime for Swift**: A Kitura based server application that you can use as a starting point to g.
- XPages**: Develop, deploy and scale IBM XPages applications with ease. The IBM.
- Go**: Develop, deploy, and scale Go web apps with ease.
- PHP**: Develop, deploy, and scale PHP web apps with ease.
- Python**: Develop, deploy, and scale Python web apps with ease.
- Ruby**: Develop, deploy, and scale Ruby web apps with ease.
- Tomcat**: Develop, deploy, and scale Tomcat web apps with ease.

Figure 2-12 IBM Cloud catalog: Cloud Foundry Apps

Notes:

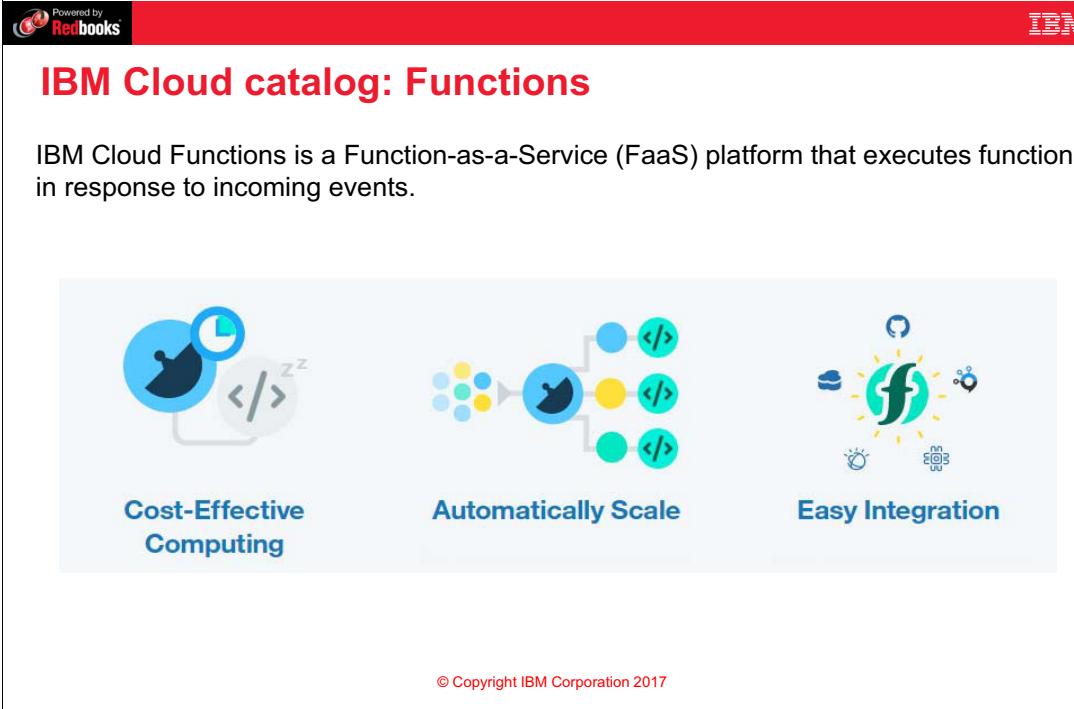
To create your own application, select from the following runtime environments as a starting point:

- ▶ IBM runtime environments, which are supported by IBM buildpacks, such as Liberty for Java.
- ▶ Community runtimes, which rely on open source and third-party buildpacks, such as Tomcat.

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

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

2.15 IBM Cloud catalog: Functions



The image shows the 'IBM Cloud catalog: Functions' page. At the top left is the 'Powered by Redbooks' logo, and at the top right is the 'IBM' logo. The main title 'IBM Cloud catalog: Functions' is centered above a brief description: 'IBM Cloud Functions is a Function-as-a-Service (FaaS) platform that executes functions in response to incoming events.' Below the description are three icons with corresponding text labels: 'Cost-Effective Computing' (a blue icon with a person and code), 'Automatically Scale' (a central node connected to multiple colored circles), and 'Easy Integration' (a central 'f' icon with various integration symbols). At the bottom of the page is a copyright notice: '© Copyright IBM Corporation 2017'.

Figure 2-13 IBM Cloud catalog: Functions

Notes:

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

Consider the following IBM Cloud Functions basic concepts:

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

IBM Cloud Functions has the following benefits:

- ▶ Cost-Effective Computing: Pay only for the exact time your actions run, down to one-tenth of a second; no memory, no cost.
- ▶ Automatically scale: Run your action thousands of times in a fraction of a second, or once a week. Action instances scale to meet demand, then disappear.
- ▶ Easy integration: Trigger your actions from events in your favorite services, or directly by using REST API.

2.16 IBM Cloud catalog: Mobile

Powered by
Redbooks

IBM Cloud catalog: Mobile

IBM Cloud catalog includes several capabilities for mobile development.

Quickly get started with your next app.

App ID Add authentication to your apps, and host user profile IBM	Mobile Analytics Mobile app developers and business stakeholders: Use IBM	Mobile Foundation Comprehensive Mobile Backend for your digital apps IBM
Push Notifications Scalable and reliable Push Notifications service for mobile Lite IBM	Bitbar Testing Mobile testing cloud service Third Party	Kinetise Rapid development of mobile apps. With Native Source Code. Third Party
Twilio Build apps that communicate. Integrate voice, messaging & Third Party	Zuznow Automatically develop mobile apps Third Party	

© Copyright IBM Corporation 2017

Figure 2-14 IBM Cloud catalog: Mobile

Notes:

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

2.17 IBM Cloud catalog: Other services

The screenshot shows the IBM Cloud catalog interface with a red header bar. On the left is the Redbooks logo, and on the right is the IBM logo. The main title "IBM Cloud catalog: Other services" is centered in red. Below the title, there are six service cards arranged in two rows of three. Each card includes an icon, the service name, a brief description, and two buttons: "Lite" and "IBM".

Service	Description	Options
Conversation	Add a natural language interface to your application.	Lite IBM
Discovery	Add a cognitive search and content analytics engine.	Lite IBM
Language Translator	Translate text from one language to another.	Lite IBM
Natural Language Classifier	Natural Language Classifier performs natural language	IBM
Natural Language Understanding	Analyze text to extract metadata from content such as	Lite IBM
Personality Insights	The Watson Personality Insights derives insights from	Lite IBM

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

© Copyright IBM Corporation 2017

Figure 2-15 IBM Cloud catalog: Other services

Notes:

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

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

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

Many IBM Cloud services are available and more are being added. A sample of the available Watson services is shown in Figure 2-15.

2.18 Creating an IBM Cloud application (1 of 3)

The screenshot shows the IBM Cloud application catalog interface. At the top, there's a red header bar with the IBM logo on the right and a "Create resource" button in the center. Below the header, the title "Creating an IBM Cloud application (1 of 3)" is displayed in red. The main content area contains two numbered steps:

1. To create an IBM Cloud application, click **Create resource** from the dashboard.
2. From the catalog, click **Cloud Foundry Apps**. Then, choose your application runtime.

Below step 2, there's a grid of runtime options:

Runtime	Description	Category
Liberty for Java™	Develop, deploy, and scale Java web apps with ease. IBM WebSphere™ Liberty.	IBM
SDK for Node.js™	Develop, deploy, and scale server-side JavaScript apps with ease. This is the IBM version of the Node.js runtime.	IBM
ASP.NET Core	Develop, deploy, and scale ASP.NET Core web apps with ease.	IBM
Runtime for Swift	A Kitura based server application that you can use as a starting point to build your own server.	IBM
XPages	Develop, deploy, and scale IBM XPages applications with ease. The IBM XPages runtime.	IBM
Go	Develop, deploy, and scale Go web apps with ease.	Community
PHP	Develop, deploy, and scale PHP web apps with ease.	Community
Python	Develop, deploy, and scale Python web apps with ease.	Community
Ruby	Develop, deploy, and scale Ruby web apps with ease.	Community
Tomcat	Develop, deploy, and scale Tomcat web apps with ease.	Community

Figure 2-16 Creating an IBM Cloud application (1 of 3)

Notes:

The following runtimes are now available as Cloud Foundry Apps:

- ▶ Liberty for Java
- ▶ SDK for Node.js
- ▶ ASP.NET Core
- ▶ Runtime for Swift
- ▶ Xpages
- ▶ Go
- ▶ PHP
- ▶ Python
- ▶ Ruby
- ▶ Tomcat

Note: These runtime choices might change in future.

2.19 Creating an IBM Cloud application (2 of 3)

The screenshot shows the 'Creating an IBM Cloud application (2 of 3)' page. At the top, there's a red header bar with the IBM logo on the right and a 'Powered by Redbooks' logo on the left. Below the header, the title 'Creating an IBM Cloud application (2 of 3)' is displayed in red. A sub-section title '3. Enter the App name and Host name and choose the pricing plan.' is shown in black. The main form area has a light gray background. It includes the following fields:

- App name:** A text input field with placeholder text 'Enter a unique name'.
- Host name:** A text input field with placeholder text 'Enter a unique name'.
- Domain:** A dropdown menu set to 'mybluemix.net'.
- Select region to deploy in:** A dropdown menu set to 'US South'.
- Choose an organization:** A dropdown menu set to 'aazraq@eg.ibm.com'.
- Choose a space:** A dropdown menu set to 'dev'.

On the left side of the form, there's a sidebar with the following information:

- SDK for Node.js™**: Described as 'Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.'
- Lite** and **IBM** buttons (the **IBM** button is highlighted).
- View Docs** link.
- VERSION**: 3.x
- TYPE**: Application

On the right side, there's a note: 'Monthly prices shown are for country or region: Egypt'. At the bottom, there's a copyright notice: '© Copyright IBM Corporation 2017'.

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

Notes:

Enter your application name and host name. Then, choose the region, organization, and space, and select the pricing plan for this application.

If you are using the IBM Cloud Lite account, you are not charged for any usage.

2.20 IBM Cloud app name must be unique

The screenshot shows a 'Create App' dialog box from the IBM Cloud interface. The title bar says 'IBM Cloud host name must be unique'. The main content area has a red background with white text that reads: 'By default, IBM Cloud hosts your application on the [mybluemix.net](#) domain.' Below this, there are two bullet points: 'You do not share your memory and application instances with other IBM Cloud accounts.' and 'You must choose a host name that is unique across all applications from all IBM Cloud users.' Underneath these points, another bullet point states: 'You cannot create an IBM Cloud application with a host name that is used by another application.' A small error message box is overlaid on the dialog, containing the text: 'Someone already used the host name 'helloworld''. Try entering a different name.' At the bottom of the dialog, there is a copyright notice: '© Copyright IBM Corporation 2017'.

Figure 2-18 IBM Cloud application name must be unique across domain

Notes:

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

Your app's name is different than its host name and must be unique only within your organization. You can have the same app name as another user if the host name is unique.

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

2.21 Creating an IBM Cloud application (3 of 3)

The screenshot shows a web-based interface for creating an IBM Cloud application. At the top left is the Redbooks logo with the text "Powered by Redbooks". At the top right is the IBM logo. The main title "Creating an IBM Cloud application (3 of 3)" is displayed in red. Below it, step 4 is labeled "Select your type of development style." Three options are presented in separate boxes:

- Eclipse tools for IBM Cloud**: Develop, integrate, and push apps to IBM Cloud using Eclipse. It features an icon of a blue and white striped circle.
- Command line interface**: Run your code locally. Manually push to IBM Cloud. It features an icon of a terminal window with a greater than sign and a minus sign.
- Continuous delivery**: Build, test, and deliver by using DevOps practices. It features an icon of a pink and blue circular arrow.

Figure 2-19 Creating an IBM Cloud application (3 of 3)

Notes:

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

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

2.22 IBM Cloud: Dashboard

The screenshot shows the IBM Cloud Dashboard interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "IBM Cloud: Dashboard" is displayed in large red font. The main content area is divided into three sections:

- Cloud Foundry Applications (3) 1GB/1GB Used**: A table showing three applications: "test-app-node" (Region: United Kingdom, CF Org: aazraaq@...), "vy30t-java-staging" (Region: United Kingdom, CF Org: aazraaq@...), and "vy30t-xxx-node" (Region: United Kingdom, CF Org: aazraaq@...). All three are marked as "Running (1/1)".
- Cloud Foundry Services (1) 2/4 Used**: A table showing one service: "Cloudant NoSQL DB-cm" (Region: United Kingdom, CF Org: aazraaq@...). It is listed as "dev".
- Services (2)**: A table showing two services: "App ID-a2" (Location: United Kingdom, Resource: Default, Plan: Graduate..., Details: Provisioned).

At the bottom of the dashboard, a note reads: "© Copyright IBM Corporation 2017".

• The Dashboard lists the following information:

- Available memory for all applications
- Number of services that are running in the environment

• The available memory and number of services depend on your IBM Cloud account type

Figure 2-20 IBM Cloud: Dashboard

Notes:

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

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

2.23 IBM Cloud: Application Details page

The screenshot shows the IBM Cloud Application Details page for an application named 'test-app-node'. The application is running in the United Kingdom space 'dev'. The runtime is set to Node.js. The page displays the following configuration:

- BUILDPACK:** SDK for Node.js™
- INSTANCES:** 1 instance, all are running, health is 100%.
- MB MEMORY PER INSTANCE:** 256 MB.
- TOTAL MB ALLOCATION:** 256 MB, with 768 MB still available.

At the bottom of the page, it says "© Copyright IBM Corporation 2017".

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

Figure 2-21 IBM Cloud: Application Details page

Notes:

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

2.24 Testing applications through the application route

The screenshot shows the IBM Cloud Application Details page for an application named 'test-app-node'. The application is listed as 'Running'. A red arrow points to the 'Visit App URL' button, which is highlighted in blue. Below the button, the text 'Click to open the app URL in the browser' is displayed.

Powered by Redbooks IBM

Testing applications through the application route

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

Figure 2-22 Testing applications through the application route

Notes:

To view the application route, click **Visit App URL** from the Application Details page.

2.25 Adding an IBM Cloud service

The screenshot shows a web interface for adding a service. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below the header, the title 'Adding an IBM Cloud service' is displayed in red. A sub-instruction reads: 'Add a service by completing the following steps:' followed by a numbered list. The list includes: 1. From the Dashboard, click **Create resource**. 2. Select a service from the Catalog. 3. Enter a name for the service. 4. Choose a Pricing Plan and then, click **Create**. The main form area has a light gray background. It displays the service details for 'Cloudant NoSQL DB'. On the left, there's a brief description of the service: 'Cloudant NoSQL DB provides access to a fully managed NoSQL JSON data layer that's always on. This service is compatible with CouchDB, and accessible through a simple to use HTTP interface for mobile and web application models'. There are two buttons below this description: 'Lite' and 'IBM'. In the center, there are input fields: 'Service name' (containing 'Cloudant NoSQL DB-y0'), 'Credential name' (containing 'Credentials-1'), 'Select region to deploy in' (set to 'United Kingdom'), 'Choose an organization' (containing 'aazraq@eg.ibm.com'), and 'Choose a space' (containing 'dev'). At the bottom of the form, there are links for 'View Docs', 'Need Help?', 'Contact IBM Cloud Sales', 'Estimate Monthly Cost', and 'Cost Calculator'. A large blue 'Create' button is positioned at the bottom right.

Figure 2-23 Add an IBM Cloud service

Notes:

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

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

2.26 Binding a service to an application (1 of 2)

The screenshot shows the IBM Cloud Application Details page for an application named 'test-app-node'. The left sidebar has tabs for 'Getting started', 'Overview', 'Runtime', 'Connections' (which is selected), 'Logs', 'Monitoring', and 'API Management'. The main content area displays the application's status as 'Running' in the United Kingdom space. A prominent blue button labeled 'Create connection' with a plus sign is visible. Below it, a large circular icon with a question mark indicates 'No connected services'. A small note at the bottom says 'Click Connect existing or Connect new to add service connections to your app.'

Figure 2-24 Bind a service to an application (1 of 2)

Notes:

To bind the service to your application, browse to your Application Details page and select the **Connections** tab on the left navigation bar. Then, click **Create connection**.

2.27 Binding a service to an application (2 of 2)

The screenshot shows a web-based interface for IBM Cloud. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "Binding a service to an application (2 of 2)" is displayed in red. A numbered list follows:

2. Select the service to bind to your application:
 - IBM Cloud lists the services that are compatible with your application and runtime environment only.
 - You can bind a service to more than one application.

Below the list is a modal window titled "Connect Existing Compatible Service". It contains a table with three rows of service data:

SERVICES	RESOURCE GROUP
App ID-o2	Default
Cloud Object Storage-ij	dev
Cloudant NoSQL DB-y0	--

A blue "Connect" button is located at the bottom right of the table. At the very bottom of the modal, there's a small copyright notice: "© Copyright IBM Corporation 2017".

Figure 2-25 Binding a service to an application (2 of 2)

Notes:

To bind a service to your application, hover over the compatible service and click **Connect**. A dialog box prompts you to restage the application. Click **Restage**.

Restaging the application makes the service available for use by your application.

IBM Cloud allows you to choose one of the services that was created earlier to bind it to your application. IBM Cloud lists the services that you created that are compatible with your service and runtime environment. Some services are specific to certain runtimes, such as a Java Runtime monitor, and might not be applicable or indeed usable from your application.

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

2.28 IBM Cloud: Environment variables

The screenshot shows the IBM Cloud Application Details page for a deployed application named 'test-app-node'. The left navigation bar includes 'Getting started', 'Overview', **Runtime**, 'Connections', 'Logs', 'Monitoring', and 'API Management'. The main area displays the application's status as 'Running' with a green dot, and provides links to 'Visit App URL', 'Routes', 'Logs', 'Memory and instances', 'Environment variables' (which is the active tab), and 'SSH'. Below this, the 'VCAP_SERVICES' environment variable is expanded, showing its JSON content:

```
{ "cloudantNoSQLDB": [ { "credentials": { "username": "16843797-63b7-4cb9-bbbe-fa9b7a132420-bluemix", "password": "1fb8cff7e83e738a148dcc49a9febf8afcf559fb870c89fe54dcc2668ec5", "port": 443, "url": "https://16843797-63b7-4cb9-bbbe-fa9b7a132420-bluemix.1fb8cff7e83e738a148dcc49a9febf8afcf559fb870c89fe54dcc2668ec5@16843797-63b7-4cb9-bbbe-fa9b7a132420-bluemix.cloudant.com" }, "syslog_drain_url": null } ] }
```

At the bottom of the page, a copyright notice reads: © Copyright IBM Corporation 2017

Figure 2-26 IBM Cloud: Environment variables

Notes:

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

Use the user-defined environment variables for configuration settings, instead of hard-coding the values in your application. For example, you can save the web service endpoint, user name, and password for a cloud-based database as a user-defined environment variable.

You can access the environment variables of the application from the Application Details page by clicking **Runtime** on the left navigation bar then, clicking the **Environment variables** tab.

2.29 IBM Cloud: Web Apps

The screenshot shows the IBM Cloud: Web Apps interface. At the top left is the Redbooks logo, and at the top right is the IBM logo. The main title "IBM Cloud: Web Apps" is centered above a grid of six starter kit cards. A large "Create Project" button with a plus sign icon is on the left. The cards are arranged in two rows of three:

- Express.js React** (Web App): A starter that provides a rich React frontend delivered from a Node.js application, including key... [Select Starter Kit](#)
- Java Spring Basic** (Web App): A starter that provides a basic web serving application in Java, using the Spring framework. [Select Starter Kit](#)
- Python Basic** (Web App): A starter that provides a basic web serving application using the Flask framework. [Select Starter Kit](#)
- Swift Kitura Basic** (Web App): A starter that provides a basic web serving application in Swift, using the Kitura framework. [Select Starter Kit](#)
- MongoDb + Express + React + Node** (Web App): A starter project for setting up a mongodb, express, react and node application. [Select Starter Kit](#)

At the bottom center of the page is the copyright notice: © Copyright IBM Corporation 2017.

Figure 2-27 IBM Cloud: Web Apps

Notes:

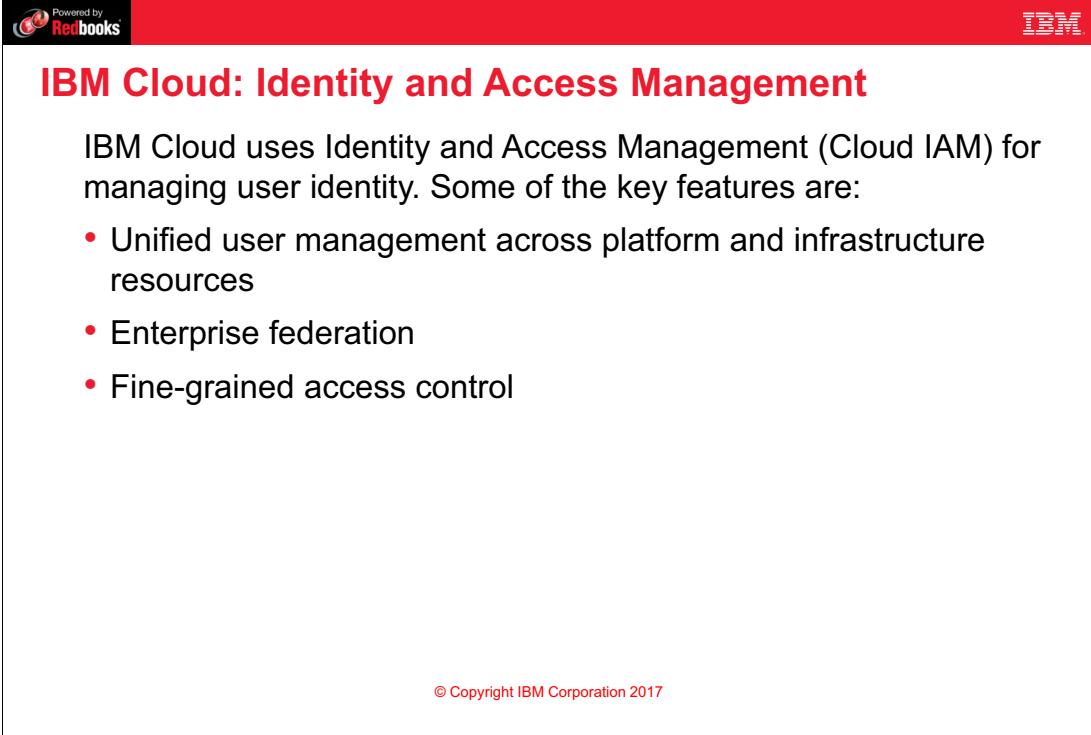
To create a Web App, click the three horizontal bars that are next to the IBM Cloud logo, choose **Web Apps**, and then, click **Get Starter Kit**.

Web Apps contains starter-kits that are pre-configured and integrated with various technologies enable you to quickly develop your web application. For example, Express.js React is a starter-kit which provides a rich React frontend delivered from a Node.js application.

Note: Express is a Node.js framework; React is a JavaScript library that is used for building user interfaces.

For more information about Node.js, Express, and React, see *Developing Node.js Applications on IBM Cloud*, SG24-8406.

2.30 IBM Cloud: Identity and Access Management



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "IBM Cloud: Identity and Access Management" is in bold red text. Below it is a text block and a bulleted list. At the bottom right is a small copyright notice.

IBM Cloud uses Identity and Access Management (Cloud IAM) for managing user identity. Some of the key features are:

- Unified user management across platform and infrastructure resources
- Enterprise federation
- Fine-grained access control

© Copyright IBM Corporation 2017

Figure 2-28 IBM Cloud: Identity and Access Management

Notes:

Some of the key IBM Cloud Identity and Access Management (Cloud IAM) features are:

- ▶ Unified user management across IBM Cloud PaaS and IaaS
A unified user management console is used to manage your users across IBM Cloud platform and infrastructure services.
- ▶ Enterprise federation
A federated ID can be used to sign up for IBM Cloud only if your company worked with IBM to register. Registering a company's domain with IBM enables users to log in to IBM products and services by using their company user credentials. Authentication is then handled by your company's identity provider. When you log in to IBM Cloud with a federated ID, you are prompted to log in through your company's login page.
- ▶ Fine-grained access control
With fine-grained access control, users can be assigned access to only the resources they need.

Note: For more information about Cloud IAM, see this website:

<https://console.bluemix.net/docs/iam/index.html>

2.31 IBM Cloud: Resources, Users, and Access control

The screenshot shows the IBM Cloud dashboard with a red header bar. The title 'IBM Cloud: Resources, Users, and Access control' is displayed in the center of the header. Below the header, there is a list of bullet points describing resources, users, and access control. At the bottom of the dashboard, there is a 'Dashboard' section with four dropdown menus: 'RESOURCE GROUP dev', 'CLOUD FOUNDRY ORG org-redbook', 'CLOUD FOUNDRY SPACE space-redbook', and 'LOCATION US South'. A copyright notice '© Copyright IBM Corporation 2017' is visible at the bottom of the dashboard area.

Figure 2-29 Resources, Users, and Access control

Notes:

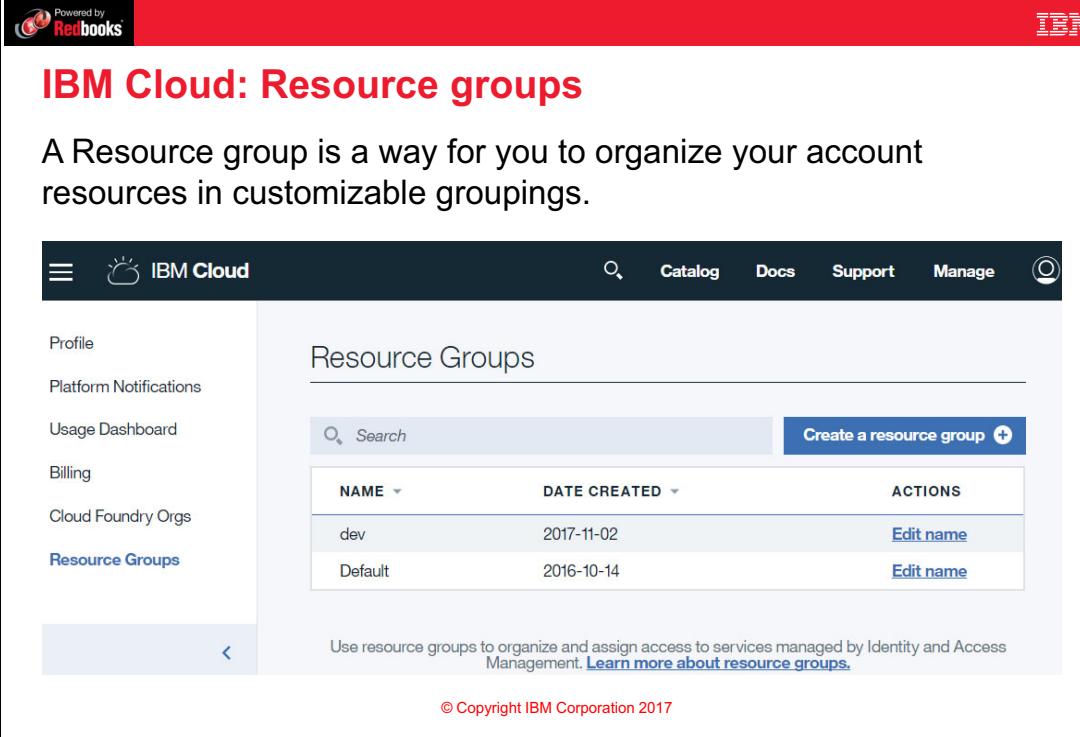
A resource is an entity in your account that you create from the IBM Cloud catalog, such as a provisioned instance of an IBM Cloud service; for example, Cloudant, a Cloud Foundry application, a virtual machine, or a container. Each account can have multiple resources.

Multiple users (identified by their IBM IDs) can be invited to an account.

Users can be granted access to resources in an account in the following ways:

- ▶ Organize resources that are enabled to use Cloud IAM into resource groups that you define in your account and assign users access to the resource groups.
- ▶ Rely on a user's role in a Cloud Foundry region, organization, and space to determine whether a user has permission to access to Cloud Foundry Apps, and services that have not yet enabled the use of Cloud IAM. These resources cannot be added to a resource group.

2.32 IBM Cloud: Resource groups



The screenshot shows the IBM Cloud Resource Groups interface. At the top, there's a red banner with the Redbooks logo and the IBM logo. Below it, the main title "IBM Cloud: Resource groups" is displayed in red. A sub-header explains that a Resource group is a way to organize account resources. The navigation bar includes links for Catalog, Docs, Support, and Manage, along with a search icon and a user profile icon. On the left, a sidebar menu lists Profile, Platform Notifications, Usage Dashboard, Billing, Cloud Foundry Orgs, and Resource Groups, with Resource Groups being the active item. The main content area is titled "Resource Groups" and contains a search bar and a "Create a resource group" button. A table lists existing resource groups: "dev" (created 2017-11-02) and "Default" (created 2016-10-14). Below the table, a note encourages users to use resource groups for organization and access management, with a link to learn more. The footer includes a copyright notice for IBM Corporation 2017.

Figure 2-30 IBM Cloud: Resource groups

Notes:

A resource group is a way for you to organize your account resources in customizable groupings so that you can quickly assign users access to more than one resource at a time.

You can use resource groups within your account to group resources that were created from services that support Cloud IAM for access control. Consider the following points:

- ▶ Users are granted access to resources in a Resource Group.
- ▶ Any account resource that is managed by using Cloud IAM access control belongs to a resource group within your account.
- ▶ Access to resources is not restricted to Cloud Foundry regions, organizations, and spaces.

Complete the following steps to create a Resource Group:

1. Click **User Profile**.
2. Click **Resource Groups**.
3. Click **Create a resource group**.
4. Specify the name of the resource group.
5. Click **Add**.

Resource groups are not restricted by Cloud Foundry regions, organizations, and spaces.

For more information about Cloud Foundry organizations, see 2.36, “Cloud Foundry: Organizations” on page 65.

2.33 IBM Cloud: Resource group access

The screenshot shows a user interface for assigning access to a resource group. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below the header, the title 'IBM Cloud: Resource group access' is displayed in red. A bullet list below the title says: '• Assign users access to a resource group' and '– Quickly assign users access to more than one resource at a time.' The main content area has a light gray background. It starts with a section titled 'Assign access to a resource group to Ahmed Azraq' with an email input field 'aazraq@eg.ibm.com'. Below this is a 'Resource group' dropdown menu set to 'dev'. Underneath, there's a section titled 'Assign platform access roles' with a note: 'These roles enable users to complete tasks on platform resources, such as creating instances, connecting instances to apps, and assigning user permissions.' Three checkboxes are listed: 'Administrator' (selected), 'Editor' (unchecked), and 'Viewer' (unchecked). Each checkbox has a detailed description below it. At the bottom of the interface, a copyright notice reads '© Copyright IBM Corporation 2017'.

Figure 2-31 IBM Cloud: Resource group access

Notes:

Complete the following steps to assign access to users to resource groups:

1. Click **Manage** in the top toolbar.
2. Click **Accounts** → **Users**.
3. Select the user.
4. Click **Assign access**.
5. Click **Assign to a resource group**.
6. Select the resource group and the access role, as shown in Figure 2-31.

Resource group access

Assigning a role to a user for accessing a resource group gives the user the ability to manage, edit, or view the resource group with policy access roles Administrator, Editor, and Viewer.

This type of access does not give the user access to the resources within a resource group. That access must be assigned separately on the Assign by resource page.

Resource groups and IBM Cloud accounts

If you have a Pay-As-You-Go or Subscription account, you can create multiple resource groups to make managing quota and viewing billing usage for a set of resources easier. You can also group resources to make it easier for you to assign users access to more than one instance at a time.

If you have a Lite account, you cannot create multiple resource groups, but you can rename your default resource group.

2.34 IBM Cloud: Organizing resources

The screenshot shows the IBM Cloud interface for creating a new service. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below it, the title "IBM Cloud: Organizing resources" is displayed in red. A bullet point lists: "Specify the resource group when creating a resource". The main form is for "Cloud Object Storage". It includes a description of the service, fields for "Service name" (set to "Cloud Object Storage-oi"), "Select a resource group" (set to "dev"), "Select location to deploy in" (set to "Global"), and buttons for "Need Help?", "Estimate Monthly Cost", and "Cost Calculator". A large blue "Create" button is at the bottom right. A copyright notice at the bottom left reads "© Copyright IBM Corporation 2017".

Figure 2-32 IBM Cloud: Organizing resources

Notes:

When creating a resource, you can assign it to a particular resource group by selecting the resource group, as shown in Figure 2-32.

Note: Most of the services in the IBM Cloud catalog do not yet use Cloud IAM for access control. They rely on Cloud Foundry regions, organizations, and spaces.

Cloud Foundry based services are expected to migrate to the use of Cloud IAM and resource groups for access management.

Any services that are introduced in the IBM Cloud catalog use Cloud IAM and resource groups for access control.

For more information about organizing resources by using resource groups, see this website:

<https://ibm.biz/BdjCPF>

2.35 Cloud Foundry: Organizing resources

The screenshot shows the IBM Cloud Catalog interface. At the top, there's a red bar with the Redbooks logo and the IBM logo. Below it, the title 'Cloud Foundry: Organizing resources' is displayed in red. A bullet point states: 'A Cloud Foundry resource is organized by region, organization, and space'. The main area is titled 'Create a Cloud Foundry App'. On the left, there's a section for 'Python' with a brief description: 'Develop, deploy, and scale Python web apps with ease.' It shows two tabs: 'Lite' (selected) and 'Community'. Below these are links to 'View Docs', 'VERSION 1.5.15', and 'TYPE Application'. On the right, there are fields for 'App name:' (containing 'vy301-python'), 'Host name:' (containing 'vy301-python'), 'Domain:' (containing 'mybluemix.net'), 'Select region to deploy in:' (set to 'US South'), 'Choose an organization:' (containing 'aazraq@eg.ibm.com'), and 'Choose a space:' (containing 'dev'). At the bottom, a copyright notice reads '© Copyright IBM Corporation 2017'.

Figure 2-33 Cloud Foundry: Organizing resources

Notes:

Most of the services in the IBM Cloud catalog are based on Cloud Foundry.

When you create a Cloud Foundry resource (application or service), you must specify the region, organization, and space to which the resource is assigned, as shown in the slide.

You can create a resource only in a space for which you have *developer* access.

Note: Services that use Cloud IAM have the following advantages over services that are based on Cloud Foundry:

- ▶ They can connect to apps and services in any Cloud Foundry space, which allows you to connect apps and services from different regions.
- ▶ Each resource that is managed by Cloud IAM belongs to a resource group, and resource groups are not scoped by region. Therefore, you can provision apps and services from different regions into the same resource group.
- ▶ You can use fine-grained access control down to an individual resource.

2.36 Cloud Foundry: Organizations



The slide is titled "Cloud Foundry: Organizations" in red text. It features a bulleted list of points about organizations and users. At the bottom right, there is a copyright notice: "© Copyright IBM Corporation 2017".

Cloud Foundry: Organizations

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

© Copyright IBM Corporation 2017

Figure 2-34 Cloud Foundry: Organizations

Notes:

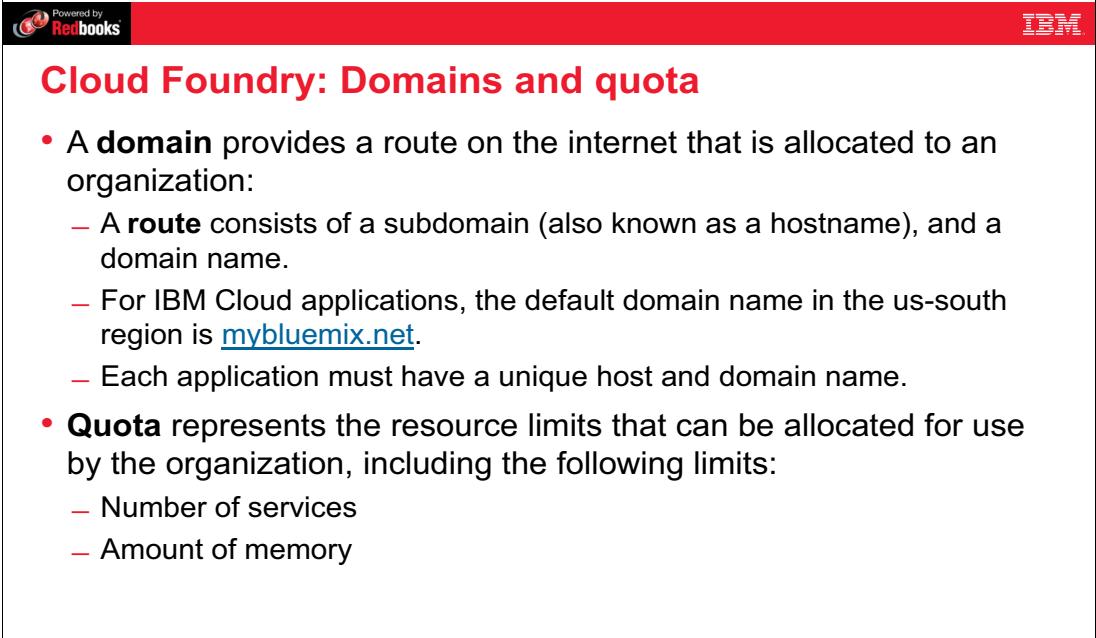
Organizations are defined by the following items:

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

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. A user can belong to only one account and 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.

2.37 Cloud Foundry: Domains and quota



The slide has a red header bar. On the left is the 'Powered by Redbooks' logo, and on the right is the 'IBM' logo. The main title 'Cloud Foundry: Domains and quota' is centered in bold red font. Below it is a bulleted list of points.

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

Figure 2-35 Cloud Foundry: Domains and quota

Notes:

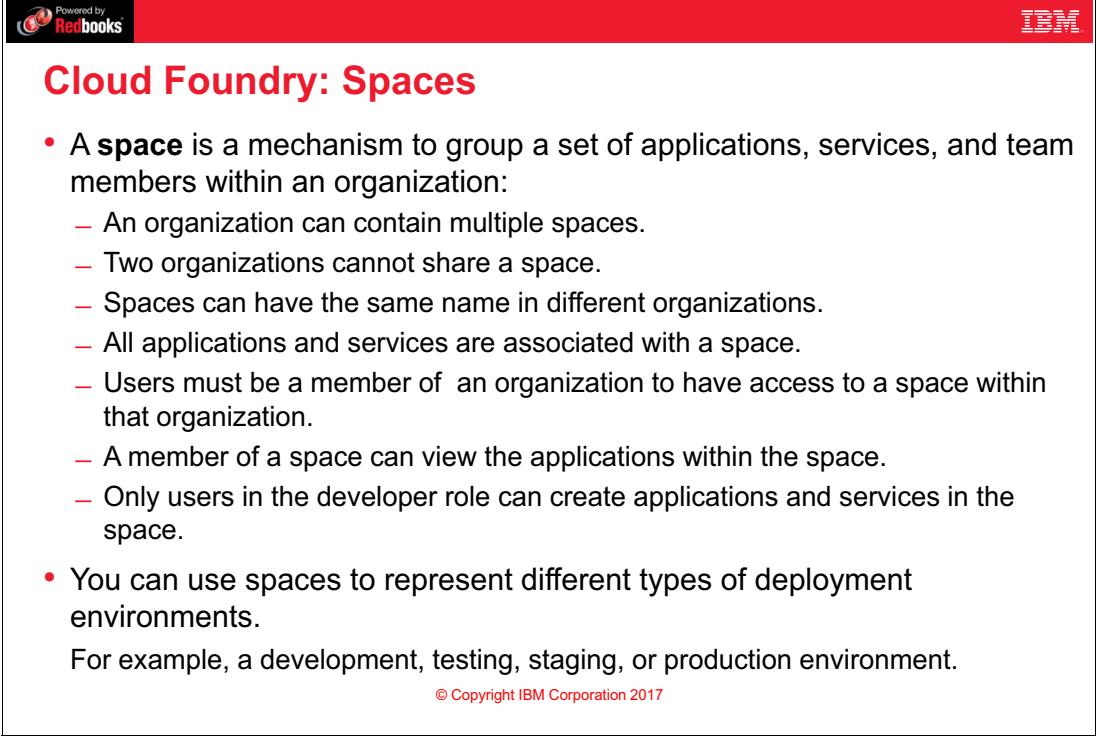
Domains provide the route on the internet that is allocated to the organization. Consider the following points:

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

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

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

2.38 Cloud Foundry: Spaces



The slide is titled "Cloud Foundry: Spaces" in red text. It features a bulleted list of points about spaces. At the bottom, it says "For example, a development, testing, staging, or production environment." and includes a copyright notice.

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

© Copyright IBM Corporation 2017

Figure 2-36 Cloud Foundry: Spaces

Notes:

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

- ▶ A space is a mechanism to group a set of applications, services, and team members within an organization.
- ▶ Although an organization can have multiple spaces, 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.
- ▶ IBM Cloud users typically use spaces to delineate different types of deployment environments, such as development, testing, staging, and production.

2.39 Cloud Foundry: User roles



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

© Copyright IBM Corporation 2017

Figure 2-37 Cloud Foundry: User roles

Notes:

Users have the following 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 applications and services within the space.
- ▶ Space auditors have read only access to settings, logs, applications, and services.

2.40 Managing Cloud Foundry organizations

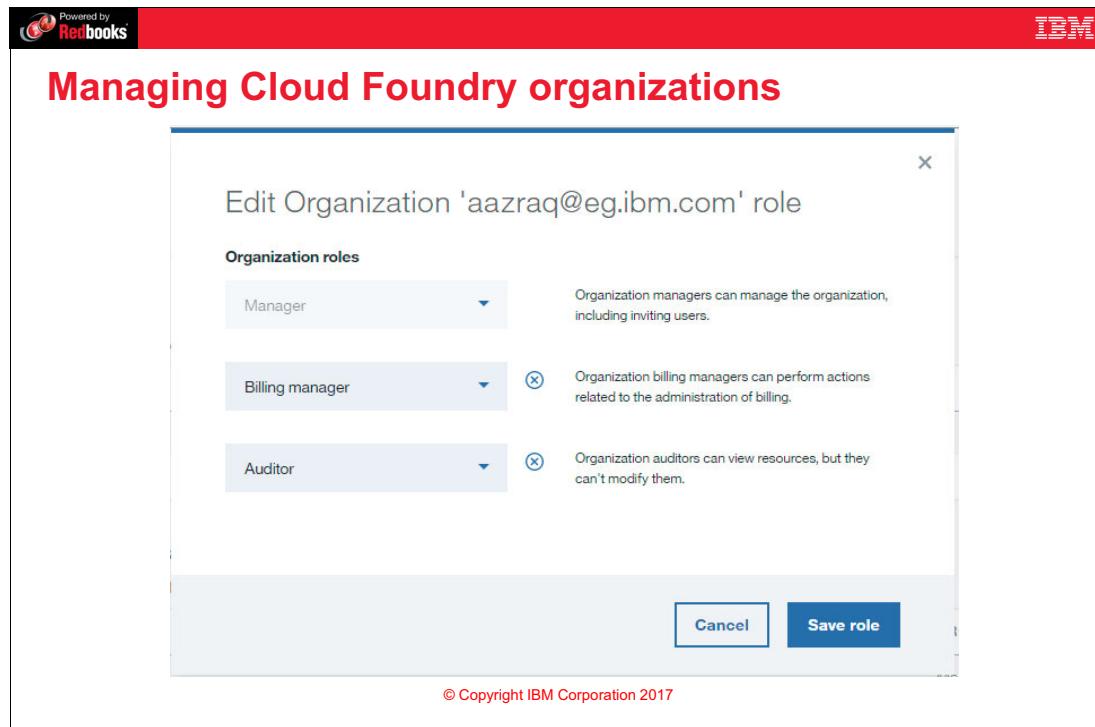


Figure 2-38 Managing Cloud Foundry organizations

Notes:

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

To edit organization access for a specific user, click **Manage** on the top toolbar then, click **Account** then, **Users**. Click the user, Manage User window opens. Click Actions (3 dots icon) next to the organization then, click **Edit organization role**.

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

2.41 Inviting users to a Cloud Foundry organization

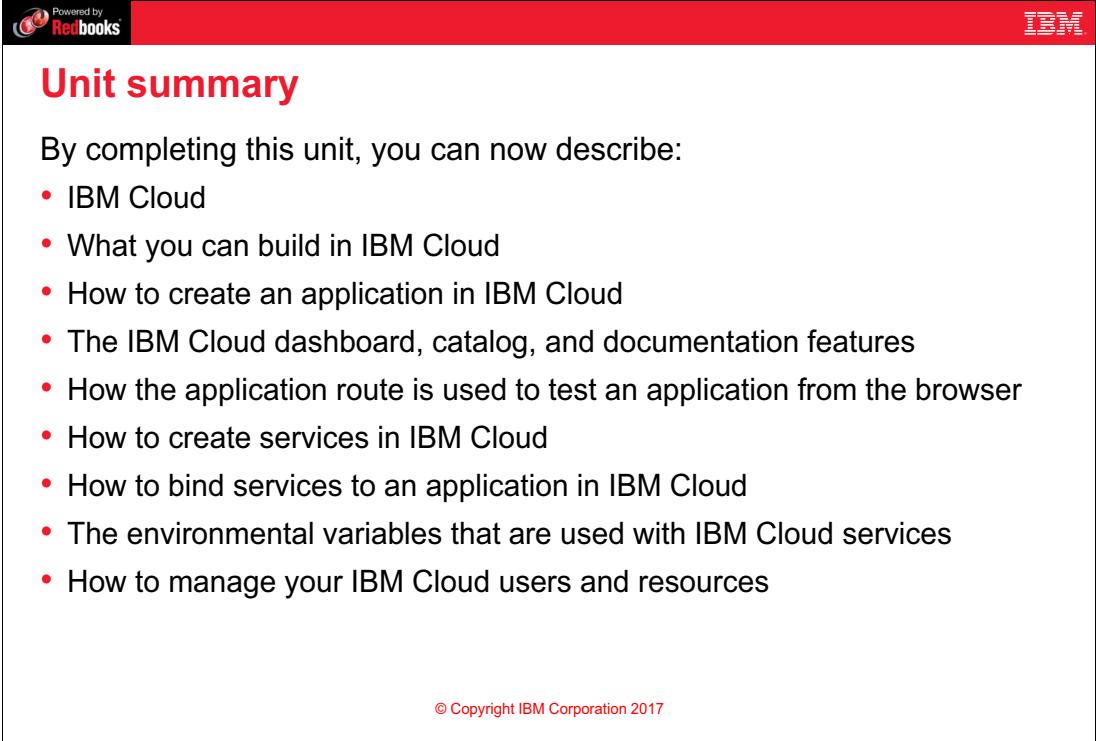
The screenshot shows the IBM Cloud Identity & Access Management interface. At the top, there's a red banner with the Redbooks logo and the word "IBM". Below the banner, the title "Inviting users to a Cloud Foundry organization" is displayed in large red text. A sub-header "The account manager can invite users to the account." follows. The main content area has a dark header bar with "IBM Cloud", a search icon, and links for "Catalog", "Docs", "Support", "Manage", and a user profile icon. On the left, a sidebar under "Identity & Access" has "Users" selected, with other options like "Service IDs", "Authorizations", and "Platform API Keys". The main panel shows a breadcrumb "Users /" and a heading "Invite users to 'Ahmed Azraq's Account'". Below this is a "Users" section with a note: "You can invite one or more users. All users invited together are granted the same access." A text input field labeled "Email address or existing IBMid" contains the placeholder text "Enter the user's email address or IBMid separated by commas, spaces, or line breaks.". At the bottom, there's an "Access" section with a note: "Assign access for all invited users. The default is no access." The footer of the page includes the copyright notice "© Copyright IBM Corporation 2017".

Figure 2-39 Inviting users to a Cloud Foundry organization

Notes:

To invite users, click **Manage** in the top toolbar. Then, click **Account** → **Users** → **Invite users**.

2.42 Unit summary



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top left of this area is the title "Unit summary" in bold red font. Below the title is a paragraph of text. To the right of the text is a bulleted list of 11 items. At the bottom right of the content area is a small copyright notice.

By completing this unit, you can now describe:

- IBM Cloud
- What you can build in IBM Cloud
- How to create an application in IBM Cloud
- The IBM Cloud dashboard, catalog, and documentation features
- How the application route is used to test an application from the browser
- How to create services in IBM Cloud
- How to bind services to an application in IBM Cloud
- The environmental variables that are used with IBM Cloud services
- How to manage your IBM Cloud users and resources

© Copyright IBM Corporation 2017

Figure 2-40 Unit summary

2.43 Checkpoint questions

Powered by
 Redbooks

IBM

Checkpoint questions

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

© Copyright IBM Corporation 2017

Figure 2-41 Checkpoint questions

2.44 Checkpoint answers



Checkpoint answers

1. Answers:

- a. No. You must authorize each person to access your organization and therefore see your applications.
- b. No. Backwards, organization access is how you control services.
- c. No. Users can have any number of spaces in many organizations.
- d. Correct. Cloud Foundry based resources belong to exactly one space.

2. False. NoSQL databases are IBM Cloud **services**.

3. False. Only one app can use the hostname for a specific domain name.

4. The *application route* in IBM Cloud is _____.
c. the Internet URL where users can access the application.

© Copyright IBM Corporation 2017

Figure 2-42 Checkpoint answers



Developing IBM Cloud applications from a local workstation

This unit covers the following topics:

- ▶ Node.js introduction
- ▶ Building an application by using the Cloud Foundry and IBM Cloud command-line interface
- ▶ Building an application by using the IBM Eclipse Tools for IBM Cloud

3.1 What you should be able to do

After completing this unit, you should be able to:

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

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://console.bluemix.net/docs/starters/install_cli.html
- ▶ *Developing with Eclipse tools* documentation:
https://console.bluemix.net/docs/starters/deploy_eclipsetools.html

3.3 Node.js

Node.js

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

Figure 3-1 Node.js

Notes:

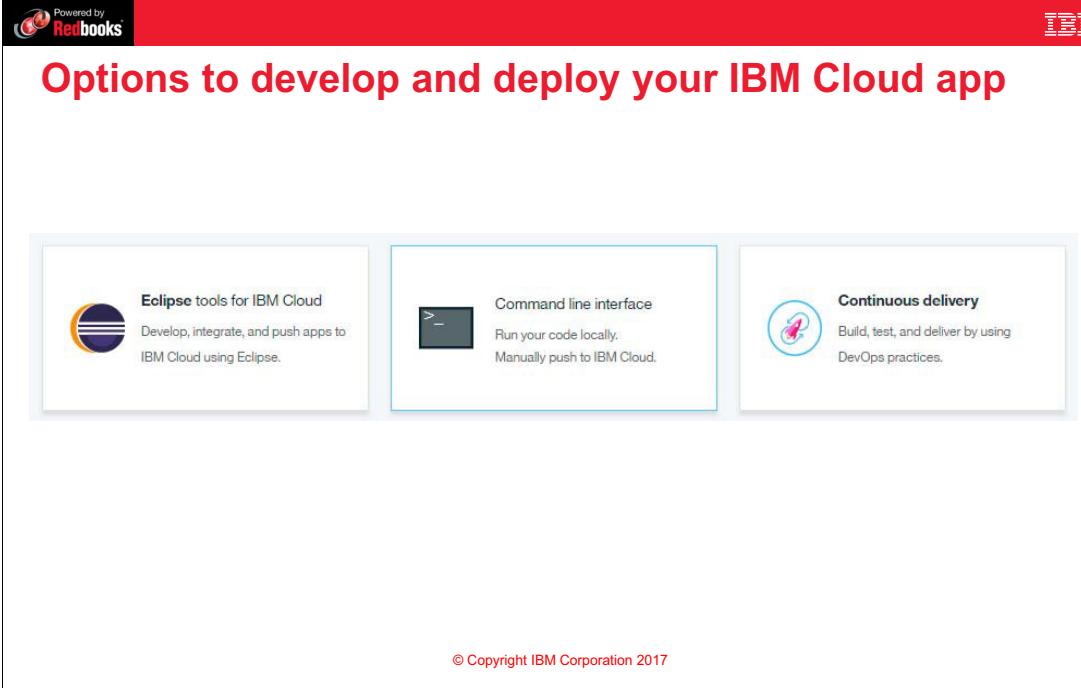
Node.js is an open-source language that runs on V8. V8 is an open source engine that was developed by Google for the Google Chrome browser.

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

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

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

3.4 Options to develop and deploy your IBM Cloud app



The image shows a slide titled "Options to develop and deploy your IBM Cloud app". It features three main options: "Eclipse tools for IBM Cloud", "Command line interface", and "Continuous delivery". Each option has a small icon and a brief description. The slide is branded with "Powered by Redbooks" and the IBM logo.

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

© Copyright IBM Corporation 2017

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

Notes:

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

- ▶ Eclipse
- ▶ Command-line interface (CLI):
 - IBM Cloud CLI: The IBM Cloud CLI (bx CLI) provides the CLI to manage applications, containers, infrastructure, services, and other resources on IBM Cloud. It also bundles cf CLI in its installation to manage Cloud Foundry resources on IBM Cloud.
 - Cloud Foundry CLI: The Cloud Foundry command-line interface (cf CLI) is a client tool from the Cloud Foundry community. You can use this tool to manage Cloud Foundry applications and services on IBM Cloud.
- ▶ Continuous delivery using DevOps practices

For more information about the use of DevOps services, see Chapter 4, “Adopting a DevOps approach by using IBM Cloud Continuous Delivery” on page 99.

3.5 Overview of bx and cf command-line interfaces



Overview of bx and cf command-line interfaces

- Cloud Foundry is an open-source platform as a service (PaaS) that provides a range of cloud environments, frameworks, and application services.
- IBM Cloud PaaS implements IBM Open Cloud Architecture, which is based on Cloud Foundry.
- The IBM Cloud command-line interface (bx cli) provides a simple way to manage all your resources on IBM Cloud from your workstation. The Cloud Foundry command-line interface (cf CLI) can be used to manage your Cloud Foundry resources on IBM Cloud.
- Use the bx cli to log in to IBM Cloud, deploy applications, and manage resources. Alternatively, you can use cf cli if you are working with Cloud Foundry resources only.

© Copyright IBM Corporation 2017

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

Notes:

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

You can use bx CLI or to log in to IBM Cloud and deploy an application to IBM Cloud, and to manage resources. Alternatively, you can use cf cli to deploy Cloud Foundry applications and manage Cloud Foundry resources on IBM Cloud.

3.6 Why use command-line tools?



Powered by
Redbooks

IBM

Why use command-line tools?

- Deploy, update, and manage your IBM Cloud environment with a fast, minimal tool.
- Automate the management and deployment process with shell scripts.
- Use specific tools for each project to provide some level of separation.

© Copyright IBM Corporation 2017

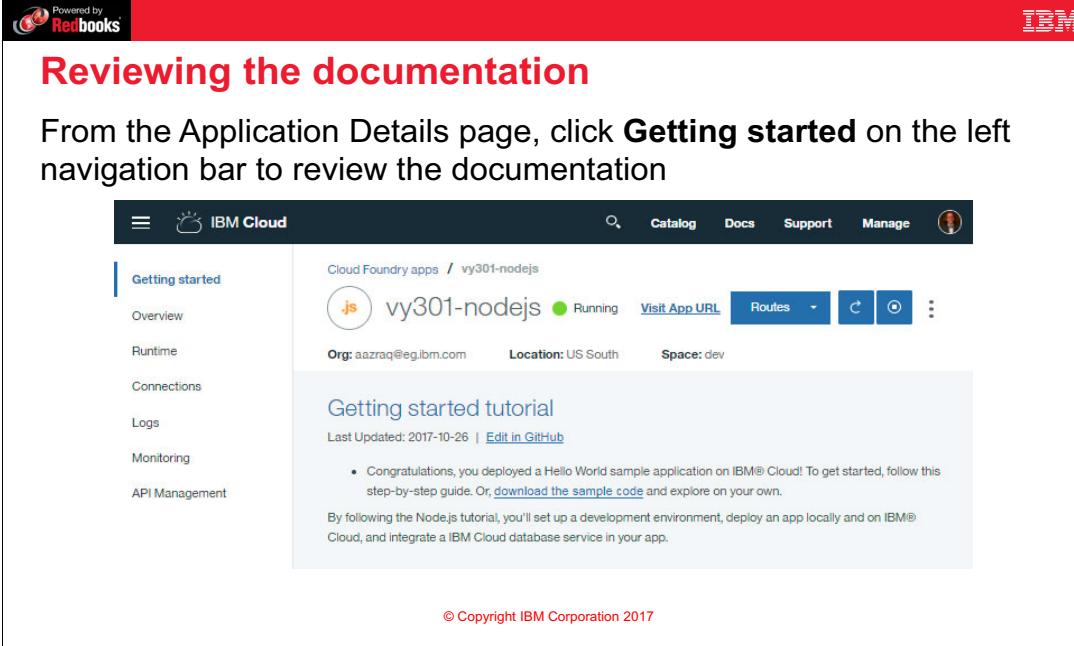
Figure 3-4 Why use command-line tools?

Notes:

IBM Cloud integrates into the Eclipse IDE. However, you might still want to use command-line tools in the following scenarios:

- ▶ You need a tool that is quick to install and has a minimal footprint on your development workstation.
- ▶ You want to automate the management and deployment process with shell scripts, or you want to use different tools for different projects to provide some project-level separation.

3.7 Reviewing the documentation



The screenshot shows the IBM Cloud Application Details page for an application named 'vy301-nodejs'. The left sidebar has 'Getting started' selected. The main content area displays a 'Getting started tutorial' for Node.js, last updated on 2017-10-26. It includes a step-by-step guide and a link to download sample code. The top navigation bar includes links for Catalog, Docs, Support, Manage, and a user profile.

Figure 3-5 Reviewing the documentation

Notes:

You can review the documentation for the command-line tool from Getting Started in Application Details.

To work with IBM Cloud by using the CLI, you can use bx CLI or cf CLI.

Note: bx CLI enables you to manage applications, containers, infrastructures, services and other resources in IBM Cloud. It also bundles cf CLI, to manage Cloud Foundry resources on IBM Cloud. cf CLI allows you to manage only Cloud Foundry resources on IBM Cloud.

How to download and use the command-line interface to manage IBM Cloud applications is described next.

3.8 Using the IBM Cloud CLI

The following sections show you how to use the IBM Cloud CLI.

3.8.1 Step 1: Install the IBM Cloud CLI

The screenshot shows a web page titled 'Using the IBM Cloud CLI' under the heading 'Step 1: Install the IBM Cloud CLI'. A bullet point lists the download link: https://console.bluemix.net/docs/cli/reference/bluemix_cli/get_started.html. Below the link, there's a section titled 'Getting started with Bluemix CLI' with a note about last update and edit history. It also includes a note about the Bluemix CLI providing a command line interface to manage resources. At the bottom, there's a copyright notice: '© Copyright IBM Corporation 2017'.

Figure 3-6 Step 1: Install the IBM Cloud CLI

Notes:

The IBM Cloud CLI provides the command line interface to manage all resources on IBM Cloud.

Download the most recent IBM Cloud Command-line Interface installer from https://console.bluemix.net/docs/cli/reference/bluemix_cli/get_started.html, which includes a list of downloadable installers for Mac OS X, Linux, and Windows. After you download and install the installer to your local workstation, open a command prompt on your system and run the command `bx --version` to check if the tool works successfully in your environment.

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

3.8.2 Step 2: Connect to your IBM Cloud account

The screenshot shows a terminal window with a red header bar. On the left of the header is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top, in large red font, it says "Using the IBM Cloud CLI" and "Step 2: Connect to your IBM Cloud account". Below that, in black font, is the instruction "Log in to IBM Cloud and choose your account:". A command line session follows:

```
$ bx login -a https://api.ng.bluemix.net -u ibmv301@gmail.com  
API endpoint: https://api.ng.bluemix.net  
Password>  
Authenticating...  
OK  
Select an account (or press enter to skip):  
1. IBM (b01c84218f070f72af3664551bf4c1a9)  
Enter a number> 1  
Targeted account IBM (b01c84218f070f72af3664551bf4c1a9)
```

In the bottom right corner of the terminal window, there is a small red copyright notice: "© Copyright IBM Corporation 2017".

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

Notes:

Log in to IBM Cloud by using the `bx` CLI. When you log in by using `bx`, you also must specify the account. After you specify the account, you can specify the resource group or Cloud Foundry organization.

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

3.8.3 Step 3: Connect to your Cloud Foundry organization

The screenshot shows a terminal window with a red header bar. On the left of the header is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top, in large red font, it says "Using the IBM Cloud CLI" and "Step 3: Connect to your Cloud Foundry organization". Below that, in black font, it says "Write the following command to target a specific organization and space". Then there is a code block in black font:

```
$ bx target -o vy301-q3w -s dev
Targeted org vy301-q3w
Targeted space dev

API endpoint: https://api.ng.bluemix.net (API version: 2.75.0)
Region: us-south
User: ibmvy301@gmail.com
Account: IBM (b01c84218f070f72af3664551bf4c1a9)
Resource group: default
Org: vy301-q3w
Space: dev
```

© Copyright IBM Corporation 2017

Figure 3-8 Step 3: Connect to your Cloud Foundry organization

Notes:

The use of the **bx target** command allows you to select the cloud foundry organization and space on which you work.

3.8.4 Step 4: Deploy your application to IBM Cloud

The screenshot shows a terminal window with the following command and its output:

```
$ bx app push ibm301-xxx-nodesample  
Using manifest file /Users/ibm301/bluemix/manifest.yml  
  
Updating app ibm301-xxx-nodesample in org vy301-q3w / space dev as ibm301@gmail.com  
OK  
  
Using route ibm301-xxx-nodesample.mybluemix.net  
Uploading ibm301-xxx-nodesample...  
Uploading app files from: /users/ibm301/bluemix  
Done uploading  
OK
```

Annotations with arrows point to specific parts of the output:

- An arrow points from the text "Using manifest file /Users/ibm301/bluemix/manifest.yml" to a callout box containing the text: "The manifest includes stored deployment settings."
- An arrow points from the text "Uploading ibm301-xxx-nodesample..." to a callout box containing the text: "Write down the route to your application. Test your application with this entry point."

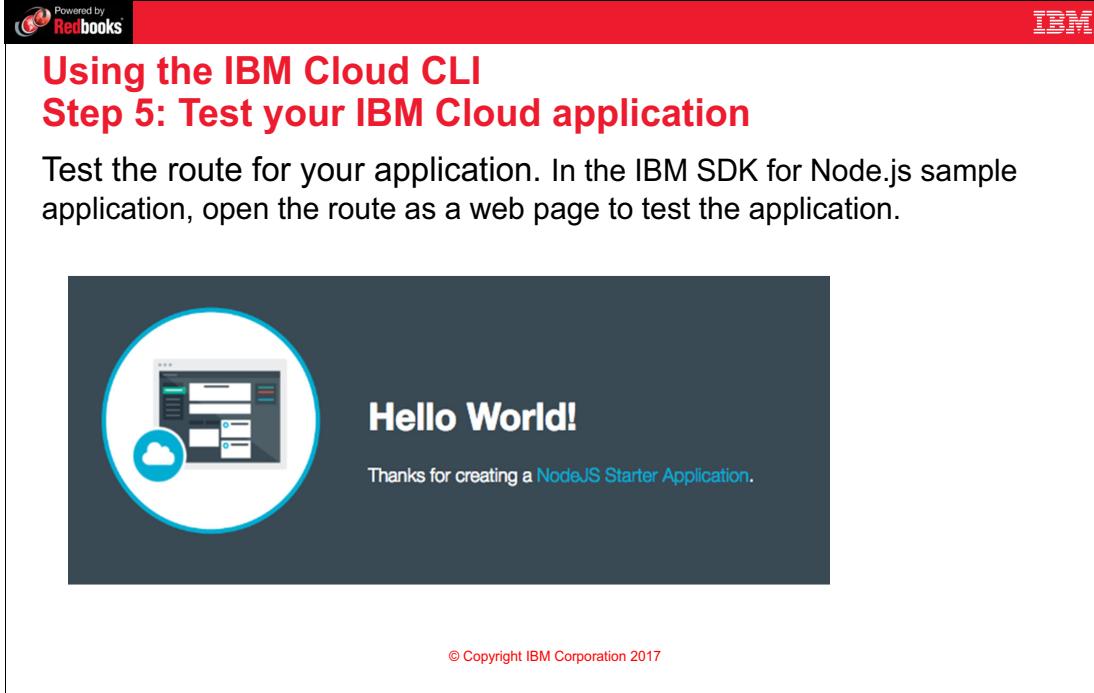
© Copyright IBM Corporation 2017

Figure 3-9 Step 4: Deploy your application to IBM Cloud

Notes:

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

3.8.5 Step 5: Test your IBM Cloud application



The screenshot shows a web-based interface for testing an IBM Cloud application. At the top left is the Redbooks logo with the text "Powered by Redbooks". At the top right is the IBM logo. The main title "Using the IBM Cloud CLI" is in red, followed by "Step 5: Test your IBM Cloud application" in red. Below this, a sub-instruction reads: "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." A large central image features a white circle containing a blue icon of a computer monitor displaying a dashboard, with a smaller blue cloud icon to its left. To the right of the icon, the text "Hello World!" is displayed in large, bold, white letters. Below it, in smaller white text, is "Thanks for creating a [NodeJS Starter Application](#)." At the bottom center of the interface, a small copyright notice reads "© Copyright IBM Corporation 2017".

Figure 3-10 Step 5: Test your IBM Cloud 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.

3.9 Using the cf CLI

The following sections show you how to use the Cloud Foundry Command Line Interface (CLI).

3.9.1 Step 1: Install the CLI

Using the cf CLI
Step 1: Install the cf CLI

- Download the most recent cf installer from the Git project:
<https://github.com/cloudfoundry/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 [64 bit / 32 bit \(deb\)](#)
- Redhat [64 bit / 32 bit \(rpm\)](#)
- Mac OS X [64 bit \(pkg\)](#)
- Windows [64 bit / 32 bit \(zip\)](#)

Binaries

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

© Copyright IBM Corporation 2017

Figure 3-11 Step 1: Install the cf CLI

Notes:

The cf CLI can be used to manage Cloud Foundry applications and services on IBM Cloud. Download the most recent cf installer from the Git project, which includes a list of installers and binaries downloadable for most systems. Generally, download an installer because they are faster and easier to use than the raw binaries.

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

Note: IBM Cloud CLI bundles cf CLI. All `cf` commands can be used with IBM Cloud CLI by using the following syntax:

```
bx cf <cf_command>
```

3.9.2 Step 2: Connect to your IBM Cloud account

Powered by
Redbooks

IBM

Using the cf CLI

Step 2: Connect to your IBM Cloud account

Log in to the organization and space in your IBM Cloud account:

```
$ cf login -a https://api.ng.bluemix.net -u ibmv301@gmail.com
-o vy301-q3w -s dev
API endpoint: https://api.ng.bluemix.net
Password>
Authenticating...
OK
Targeted org vy301-q3w.
Targeted space dev

API endpoint: https://api.ng.bluemix.net (API version: 2.75.0)
User: ibmv301@gmail.com
Org: vy301-q3w
Space: dev
```

© Copyright IBM Corporation 2017

Figure 3-12 Step 2: Connect to your IBM Cloud account

Notes:

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

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

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 for UK region.

You can use the flags (-a, -u, and so on) to specify the parameters at the command line. If you do not specify one of the parameters, the command-line utility prompts you to enter it later. In the example that is shown in Figure 3-12, the password was not entered; therefore, the command-line utility prompts the user to enter the password.

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

3.9.3 Step 3: Deploy your application to IBM Cloud

The screenshot shows a terminal window with the following command and its output:

```
$ cf push ibm301-xxx-nodesample  
Using manifest file /Users/ibm301/bluemix/manifest.yml  
  
Updating app ibm301-xxx-nodesample in org vy301-q3w / space dev as ibm301@gmail.com  
OK  
  
Using route ibm301-xxx-nodesample.mybluemix.net  
Uploading ibm301-xxx-nodesample...  
Uploading app files from: /users/ibm301/bluemix  
Done uploading  
OK
```

Annotations highlight specific parts of the output:

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

© Copyright IBM Corporation 2017

Figure 3-13 Step 3: Deploy your application to IBM Cloud

Notes:

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

When you are ready to deploy your application to IBM Cloud complete the following steps:

1. Browse to your application directory (the directory where `manifest.yml` is stored).
2. Run the **cf push** command.

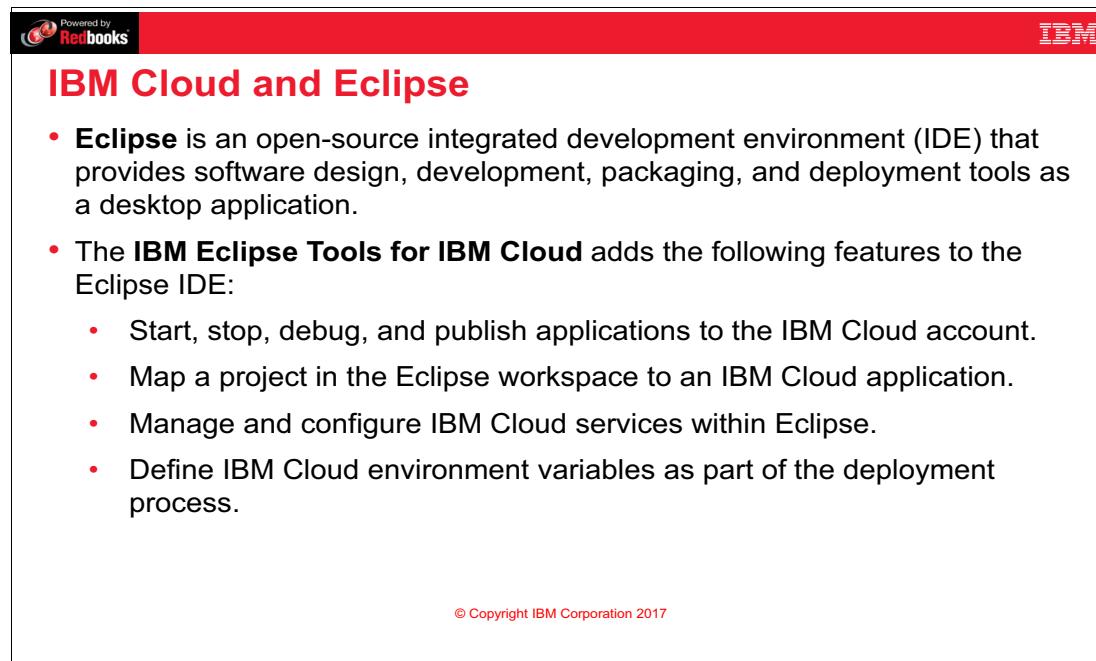
"Pushing your app" is the terminology that is used to indicate that you are going to deploy the latest changes of your app to IBM Cloud.

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

Write down the application route. You can test your application with this entry point.

Figure 3-13 shows the command-line utility accessing a manifest file. You can control various aspects of the deployment, such as the amount of memory an application receives and the route of the application, by specifying them as flags in the command. However, you can also store these settings in a `manifest.yml` file to make the command you need shorter to enter in later iterations of your application.

3.10 IBM Cloud and Eclipse



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "IBM Cloud and Eclipse" is in red. Below it is a bulleted list of features. At the bottom right is a small copyright notice.

IBM Cloud and Eclipse

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

© Copyright IBM Corporation 2017

Figure 3-14 IBM Cloud and Eclipse

Notes:

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

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

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

3.11 Using IBM Eclipse tools for IBM Cloud

The next sections show you how to use IBM Eclipse tools for IBM Cloud.

3.11.1 Step 1: Review the documentation

The screenshot shows the IBM Cloud Application Details page for a Node.js application named "vy301-nodejs". The application is currently running. On the left, there's a sidebar with various options like Overview, Runtime, Connections, Logs, Monitoring, and API Management. The "Getting started" option is highlighted. The main content area displays basic application details: Org: aazraq@eg.ibm.com, Location: US South, and Space: dev. Below this, there's a section titled "Developing with Eclipse Tools" which contains instructions for installing the plugin. The instructions mention Eclipse Neon for Java EE Developers (4.6.1) and provide a link to edit the plugin in GitHub. There's also an "Install" button.

Figure 3-15 Step 1: Review the documentation

Notes:

For more information about the documentation that is related to developing IBM Cloud Applications by using Eclipse tools, see the following website:

<https://ibm.biz/Bdju8x>

To start developing with the IBM Eclipse Tools for IBM Cloud plugin, the Java Development Kit, and Eclipse for Java EE Developers (any version after Neon) must be installed.

3.11.2 Step 2: Install the latest version of JDK and Eclipse

The screenshot shows a Redbooks page with a red header bar. On the left is the Redbooks logo, and on the right is the IBM logo. The main title is "Using IBM Eclipse tools for IBM Cloud" and the subtitle is "Step 2: Install the latest version of JDK and Eclipse". Below this, there are two bullet points:

- Download and install the latest version of Java Development Kit (JDK) from Oracle:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Download and install the latest version of Eclipse IDE for Java EE Developers:
<https://www.eclipse.org/downloads/packages/>

Below the list, there is a download link for the Eclipse IDE for Java EE Developers:

 **Eclipse IDE for Java EE Developers**
331 MB - Downloaded 203,548 Times

On the right side, there are download links for different operating systems:

Windows 32-bit 64-bit
Mac Cocoa 64-bit
Linux 32-bit 64-bit

At the bottom center, it says "© Copyright IBM Corporation 2017".

Figure 3-16 Step 2: Install the latest version of JDK and Eclipse

Notes:

As of time of this writing, Eclipse version Oxygen and JDK 8 were used. Any newer version will be also compatible.

3.11.3 Step 3: Install Eclipse tools for IBM Cloud

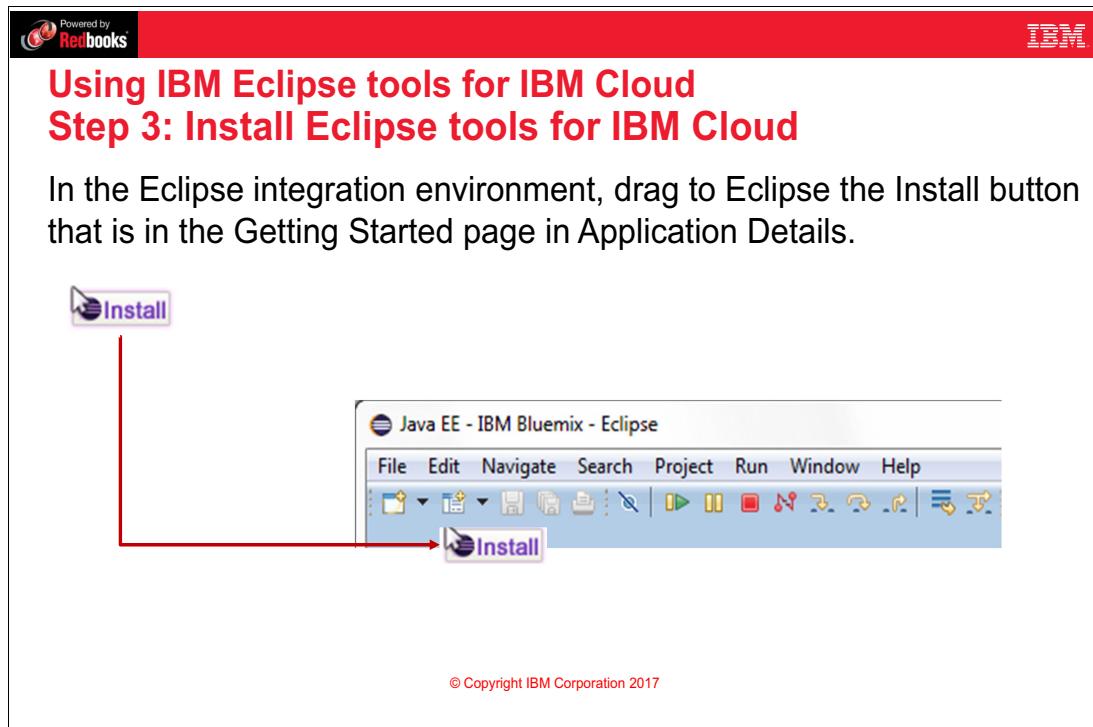


Figure 3-17 Step 3: Install Eclipse tools for IBM Cloud

Notes:

The simplest way to install the IBM Eclipse Tools for IBM Cloud is to drag the Install button from the documentation on top of your Eclipse toolbar, as shown in Figure 3-17.

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

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

3.11.4 Step 4: Connect to your IBM Cloud account

The screenshot shows the Eclipse IDE interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title 'Using IBM Eclipse tools for IBM Cloud' and the section 'Step 4: Connect to your IBM Cloud account' are displayed in red. A bulleted list provides instructions: 'In the Eclipse integrated development environment, create a server connection for your IBM Cloud account:' and 'Define an IBM Cloud server connection and enter the IBM ID and password for your account:'. Below the list is a screenshot of the Eclipse 'Servers' view, which shows a message: 'No servers are available. Click this link to create a new server...'. Below this is a screenshot of the 'New Server' dialog for 'IBM Bluemix Account'. The dialog has tabs for 'New Server' and 'Recent Servers'. It contains fields for 'Email' (warren@ca.ibm.com), 'Password' (redacted), and 'URL' (IBM Bluemix - https://api.ng.bluemix.net). Buttons at the bottom include 'Validate Account', 'Register Account...', and 'Sign Up'. A copyright notice at the bottom right of the dialog reads '© Copyright IBM Corporation 2017'.

Figure 3-18 Step 4: Connect to your IBM Cloud account

Notes:

After you successfully install the IBM Cloud tools for Eclipse plugin, you can add your IBM Cloud account as a remote server in Eclipse. Complete the following steps:

1. Open Java EE perspective in Eclipse.
2. Click the **Servers** tab.
3. Click the link that is shown in Figure 3-18 to create a server.
4. Click **IBM Bluemix**.
5. Click **Next** and enter your user IBM ID and password to log in to your IBM Cloud account. Click **Next**.
6. Choose the organization and space. Click **Finish**.

3.11.5 Step 5: Create a Node.js application project

The screenshot shows the IBM Eclipse interface with a red header bar. The title in the header is "Using IBM Eclipse tools for IBM Cloud" and the sub-section is "Step 5: Create a Node.js application project". Below the header, there is a bulleted list of instructions:

- To deploy your IBM SDK for Node.js application to IBM Cloud, 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:

A screenshot of the Eclipse facets view is shown, with "Node.js Application" selected. Below the list, another bullet point says:

- Add the project to the IBM Cloud server:

A screenshot of the Eclipse context menu for a server entry in the Servers view is shown, with "Add and Remove..." selected. At the bottom right of the interface, it says "© Copyright IBM Corporation 2017".

Figure 3-19 Step 5: Create a Node.js application project

Notes:

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

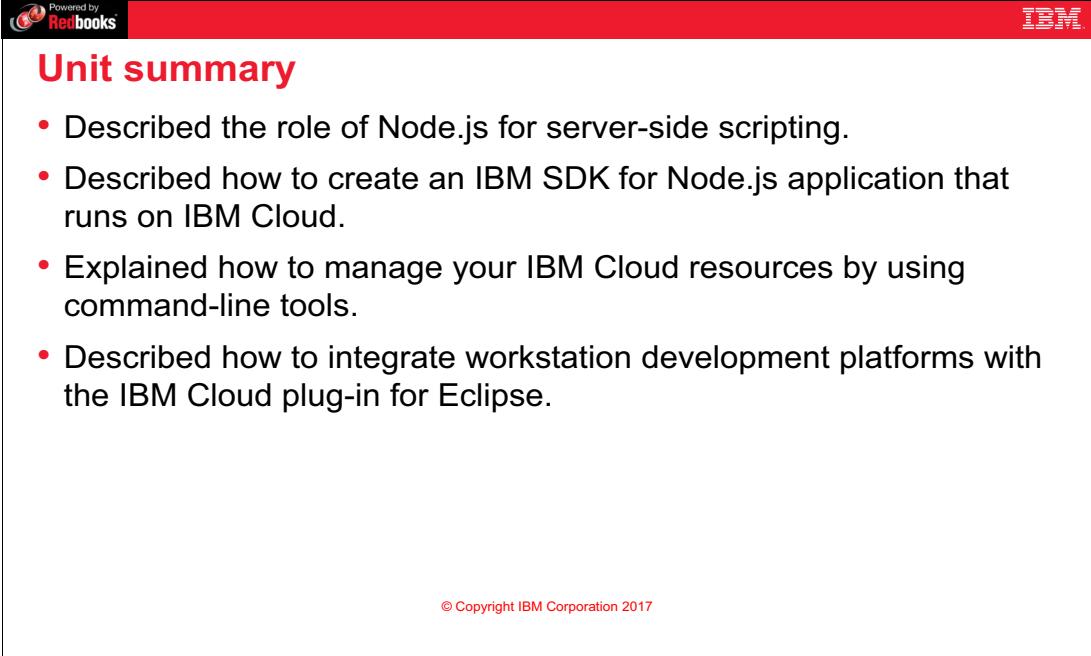
To view or modify the facets for your project, complete the following steps:

- Right-click the project and select **Properties**.
- Select **Project Facets**.
- Click **Convert to faceted form**.
- Select **Node.js Application**.

After the Node.js facet is added to your application, it shows in Eclipse as an item that you can deploy to IBM Cloud.

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

3.12 Unit summary



Powered by
Redbooks

IBM

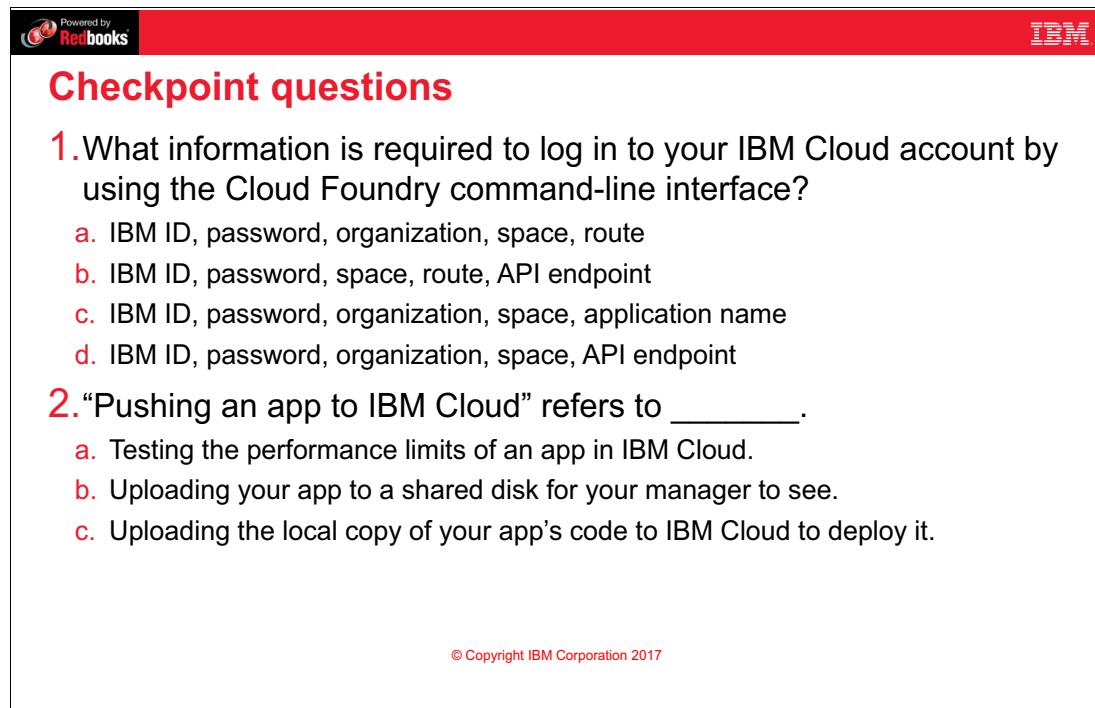
Unit summary

- Described the role of Node.js for server-side scripting.
- Described how to create an IBM SDK for Node.js application that runs on IBM Cloud.
- Explained how to manage your IBM Cloud resources by using command-line tools.
- Described how to integrate workstation development platforms with the IBM Cloud plug-in for Eclipse.

© Copyright IBM Corporation 2017

Figure 3-20 Unit summary

3.13 Checkpoint questions



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top, the title "Checkpoint questions" is displayed in bold red font. Below the title are two numbered questions in black font, each with four multiple-choice options labeled a, b, c, and d.

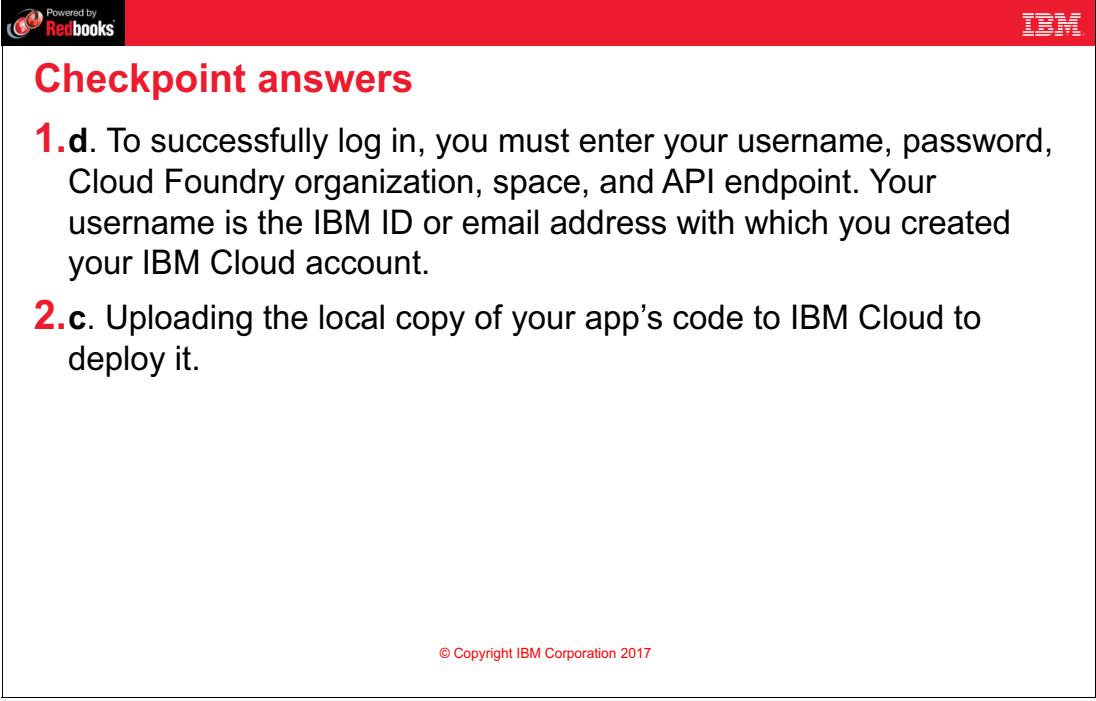
Checkpoint questions

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

© Copyright IBM Corporation 2017

Figure 3-21 Checkpoint questions

3.14 Checkpoint answers



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top, the title "Checkpoint answers" is displayed in red. Below the title are two numbered items, each starting with a red dot and followed by black text.

1.d. To successfully log in, you must enter your username, password, Cloud Foundry organization, space, and API endpoint. Your username is the IBM ID or email address with which you created your IBM Cloud account.

2.c. Uploading the local copy of your app's code to IBM Cloud to deploy it.

© Copyright IBM Corporation 2017

Figure 3-22 Checkpoint answers



Adopting a DevOps approach by using IBM Cloud Continuous Delivery

This unit covers the following topics:

- ▶ IBM Cloud Continuous Delivery overview
- ▶ Web IDE (Edit Code)
- ▶ Source Code Management (Git Repo) and Issue Tracker
- ▶ Automated Delivery Pipeline (Build & Deploy)

4.1 What you should be able to do

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

- ▶ Describe DevOps
- ▶ Describe the capabilities of IBM Cloud Continuous Delivery
- ▶ Identify the Web IDE features in IBM Cloud Continuous Delivery
- ▶ Describe how to use Git Repos and Issue Tracking
- ▶ Explain the pipeline build and deploy processes

4.2 References

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

- ▶ Open source Git project:
<https://www.git-scm.com/>
- ▶ Continuous Delivery Docs:
<https://ibm.biz/BdjTKJ>
- ▶ IBM Cloud Garage Method:
<https://www.ibm.com/cloud/garage/>
- ▶ IBM DevOps tools and methodologies from IBM:
<https://www.ibm.com/cloud/devops>
- ▶ Develop a Cloud Foundry app toolchain:
<https://ibm.biz/BdjTS3>

4.3 What is DevOps?

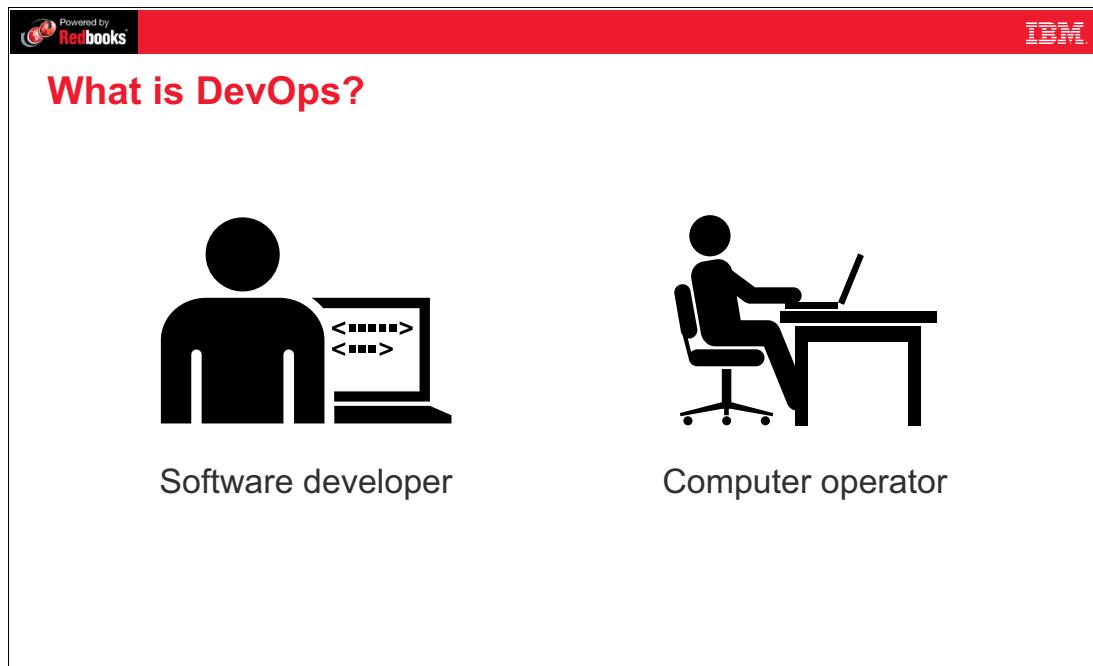


Figure 4-1 What is DevOps?

Notes:

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

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

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

A developer can use DevOps processes, automation, and other tools to automate the build process, code testing, deployment to the different environments, and monitoring. It is an active and continuous task to keep the automation, pipelines, and other entities in this realm up to date and even improve them, as it is also an active and continuous task developing code.

4.4 Benefits of DevOps

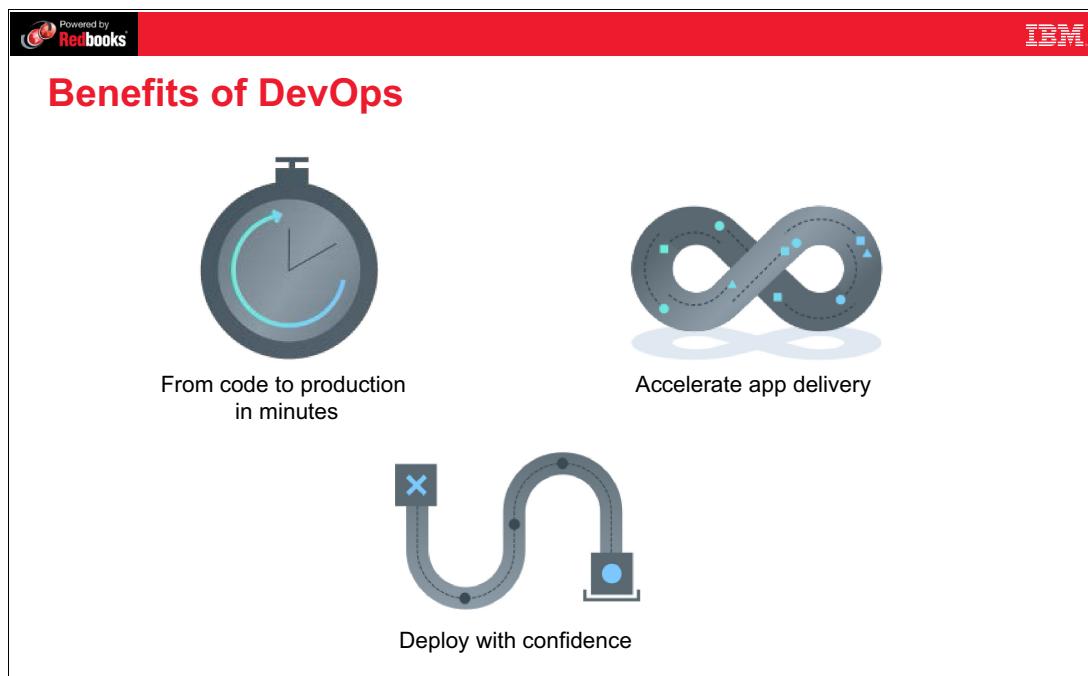


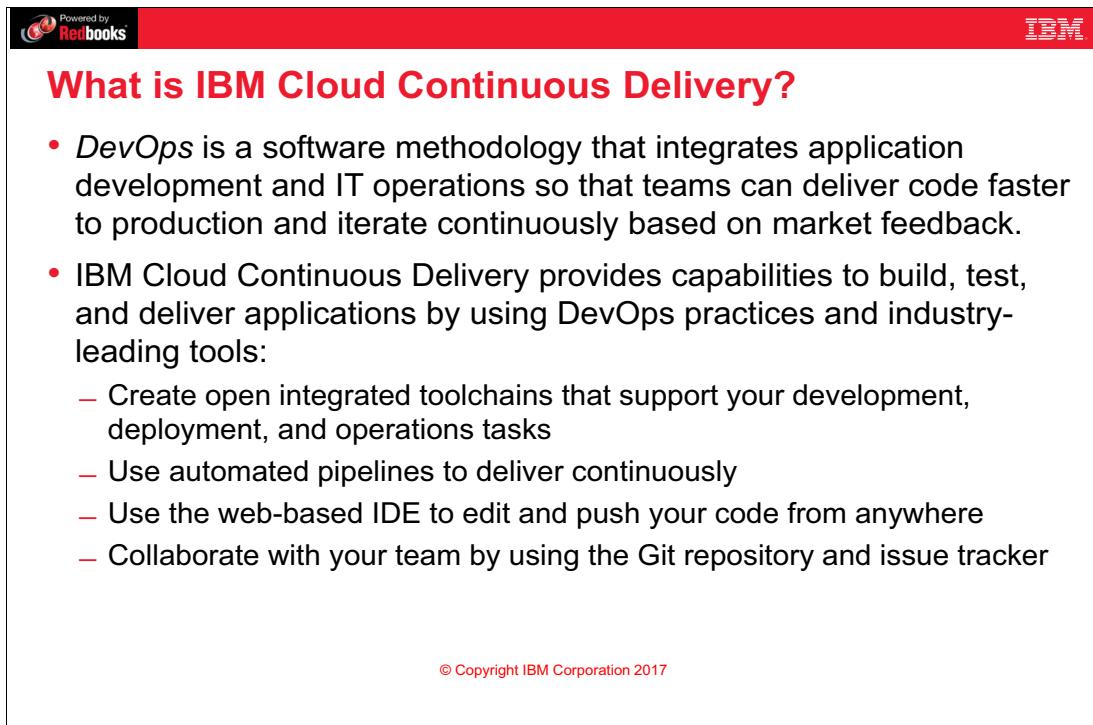
Figure 4-2 Benefits of DevOps

Notes:

DevOps provides many benefits:

- ▶ 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:
 - Leverage the repeatability of the process by using automation, which ensures that the same steps are executed every time you deploy.
 - Leverage automated testing as a quality gate before deploying code to a specific environment, or in some cases, to enable the monitoring or testing of the code in a live environment.

4.5 What is IBM Cloud Continuous Delivery?



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "What is IBM Cloud Continuous Delivery?" is in bold red text. Below it is a bulleted list of points.

What is IBM Cloud Continuous Delivery?

- DevOps is a software methodology that integrates application development and IT operations so that teams can deliver code faster to production and iterate continuously based on market feedback.
- IBM Cloud Continuous Delivery provides capabilities to build, test, and deliver applications by using DevOps practices and industry-leading tools:
 - Create open integrated toolchains that support your development, deployment, and operations tasks
 - Use automated pipelines to deliver continuously
 - Use the web-based IDE to edit and push your code from anywhere
 - Collaborate with your team by using the Git repository and issue tracker

© Copyright IBM Corporation 2017

Figure 4-3 What is IBM Cloud Continuous Delivery?

Notes:

DevOps is a software methodology that integrates application development and IT operations so that teams can deliver code faster to production and iterate continuously based on market feedback.

Although the IBM Cloud catalog provides multiple tools for DevOps, this unit is focused on IBM Cloud Continuous Delivery.

By using Continuous Delivery, you can build, test, and deliver applications by using DevOps practices and industry-leading tools.

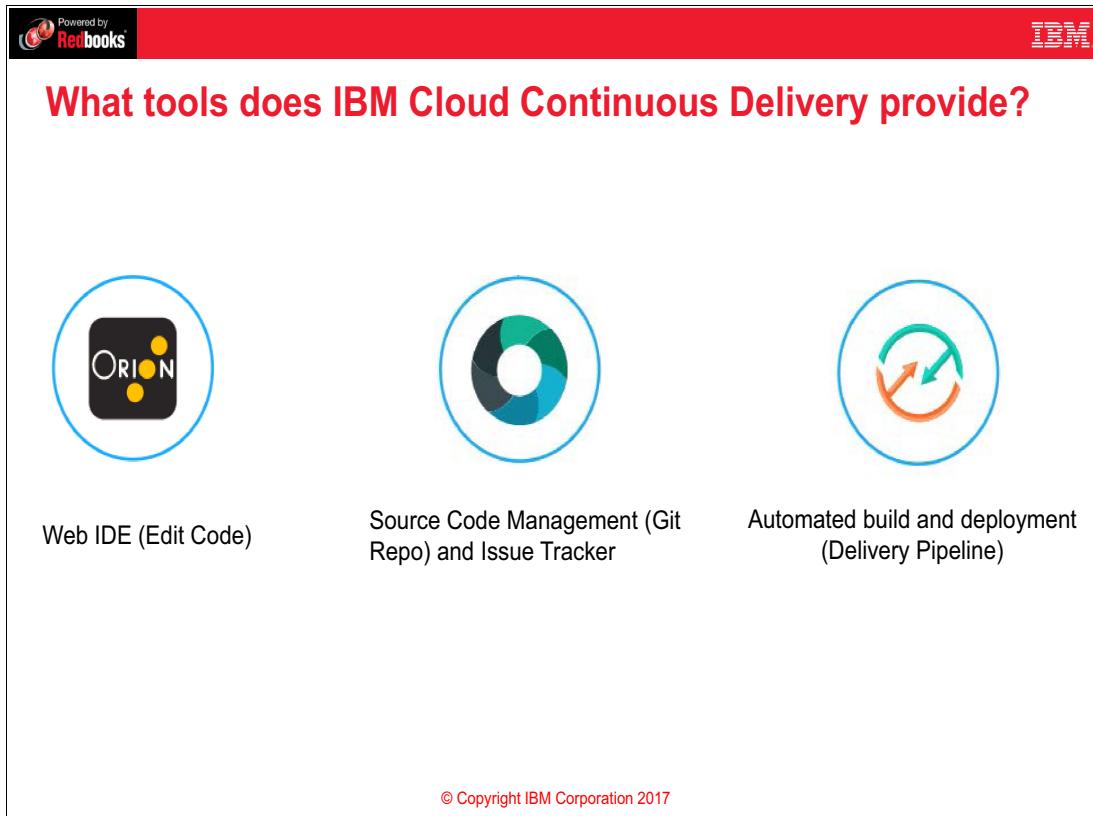
The Continuous Delivery service supports your DevOps workflows with the following capabilities:

- ▶ Create integrated DevOps open toolchains to enable tool integrations that support your development, deployment, and operations tasks. A toolchain is an integrated set of tools that you can use to collaboratively develop, build, deploy, test, and manage applications and make operations repeatable and easier to manage. Toolchains can include open source tools, IBM Cloud services, such as IBM Cloud DevOps Insights, and third-party tools, such as GitHub, PagerDuty, and Slack.
- ▶ Deliver continuously by using automated pipelines. Automate builds, unit tests, deployments, and more. Build, test, and deploy in a repeatable way with minimal human intervention. Be ready to release into production at any time.
- ▶ Edit and push your code from anywhere by using the web-based IDE. Create, edit, run, and debug, and complete source-control tasks in GitHub. Seamlessly move from editing your code to deploying it to production.

- ▶ Collaborate with your team and manage your source code with a Git repository (repos) and issue tracker that is hosted by IBM and built on GitLab Community Edition. Manage Git repos through fine-grained access controls that keep code secure. Review code and enhance collaboration through merge requests. Track issues and share ideas through the issue tracker. Document projects on the wiki system.

For more information about how to create an open toolchain, which includes the minimum tools that you need to develop and deploy a “Hello World” app to Cloud Foundry (including Git Repos and Issue Tracking), see [Develop a Cloud Foundry app toolchain](#).

4.6 What tools does IBM Cloud Continuous Delivery provide?



The diagram is titled "What tools does IBM Cloud Continuous Delivery provide?" in red text at the top. It features three circular icons with blue outlines: the first icon contains a black square with the word "ORION" and three yellow dots; the second icon is a teal and white circular logo; the third icon shows a teal circle with an orange arrow forming a loop. Below each icon is a brief description: "Web IDE (Edit Code)" for the first, "Source Code Management (Git Repo) and Issue Tracker" for the second, and "Automated build and deployment (Delivery Pipeline)" for the third. The bottom right corner of the slide contains the copyright notice "© Copyright IBM Corporation 2017".

Figure 4-4 What tools does IBM Cloud Continuous Delivery provide?

Notes:

IBM Cloud Continuous Delivery provides the following tools:

- ▶ Web IDE

You can develop your code (for example, Node.js, Java, or other languages) in this web-based integrated development environment (Web IDE). No software other than your web browser is necessary. The Web IDE saves your current work in a cloud-based file directory, which is known as the *local repository*. IBM Cloud provides Eclipse Orion as the Web IDE. You can also use a desktop IDE, such as Eclipse with DevOps services.

- ▶ Source control management (Git Repo) and Issue Tracker

IBM Cloud Continuous Delivery creates a Git repository as a change management system. The local repository is a copy of your latest edits before you submit your work to a SCM system. The Git repository, which is known as the *remote repository*, is hosted on the IBM Cloud DevOps infrastructure. Issue Tracker is part of the functionality of the Git Repo; it tracks your work, including defects, enhancements, and tasks.

- ▶ Delivery Pipeline

This tool automates the process of building and deploying your code as an IBM Cloud application. You can also configure the build, deploy, or test scripts within the Web IDE.

4.7 Additional DevOps services

The screenshot displays the IBM Cloud Catalog interface. At the top left is the Redbooks logo, and at the top right is the IBM logo. The main title "Additional DevOps services" is centered above a grid of service cards. Each card contains a service icon, name, brief description, and deployment status indicators (Lite, IBM, Beta, Third Party). The services listed are:

- Activity Tracker**: Capture, store, and visualize your Bluemix metrics.
- Alert Notification**: Never miss critical alerts. Notify the right people.
- Auto-Scaling**: Automatically increase or decrease the number of instances.
- Log Analysis**: Collect, store, and analyze your logs.
- Monitoring**: Collect, store, and analyze metrics from your applications.
- Availability Monitoring**: Around the world, around the clock.
- Continuous Delivery**: Build, test and deliver using DevOps best practices.
- Continuous Release**: Manage software deployments with this service.
- BlazeMeter**: Performance Testing Platform.
- Hipchat**: The most simple and powerful test messaging platform.
- DevOps Insights**: Improve agility, reliability, and security.
- Event Management**: Consolidated operational event and log management.
- Globalization Pipeline**: Manage the translation of your cloud and mobile applications.
- Load Impact**: Performance and load testing for DevOps.
- New Relic**: Manage and monitor your apps.
- Monitoring and Analytics**: Gain the visibility and control you need over your application.
- jKool**: jKool provides real-time and historical monitoring of your application.
- PagerDuty**: Incident Management and Resolution.

At the bottom center of the catalog is the copyright notice: © Copyright IBM Corporation 2017.

Figure 4-5 Additional DevOps services

Notes:

In addition to the Continuous Delivery service, the IBM Cloud Catalog provides multiple services to enable end-to-end DevOps, such as Auto-scaling and Monitoring. You can access these services from the Catalog by clicking **DevOps** in the platform category.

We focus on the Continuous Delivery service.

4.8 Adding Continuous Delivery

The screenshot shows a section titled "Adding Continuous Delivery". It includes a list of steps, a callout box for enabling continuous delivery, and a copyright notice.

Powered by Redbooks **IBM**

Adding Continuous Delivery

To add Continuous Delivery to your application, complete the following steps:

1. In the IBM Cloud dashboard, select your application.
2. Click **Overview** in the left navigation bar.
3. In the Continuous delivery panel, click **Enable**.

Continuous delivery

Continuous delivery is not enabled for this app.
Enable continuous delivery to automate builds, tests, and deployments through the Delivery Pipeline, GitHub, and more.

Enable

© Copyright IBM Corporation 2017

Figure 4-6 Adding Continuous Delivery

Notes:

After you enable Continuous Delivery for your app, a new tab opens to configure Continuous Delivery Toolchains.

4.9 Creating Continuous Delivery Toolchains

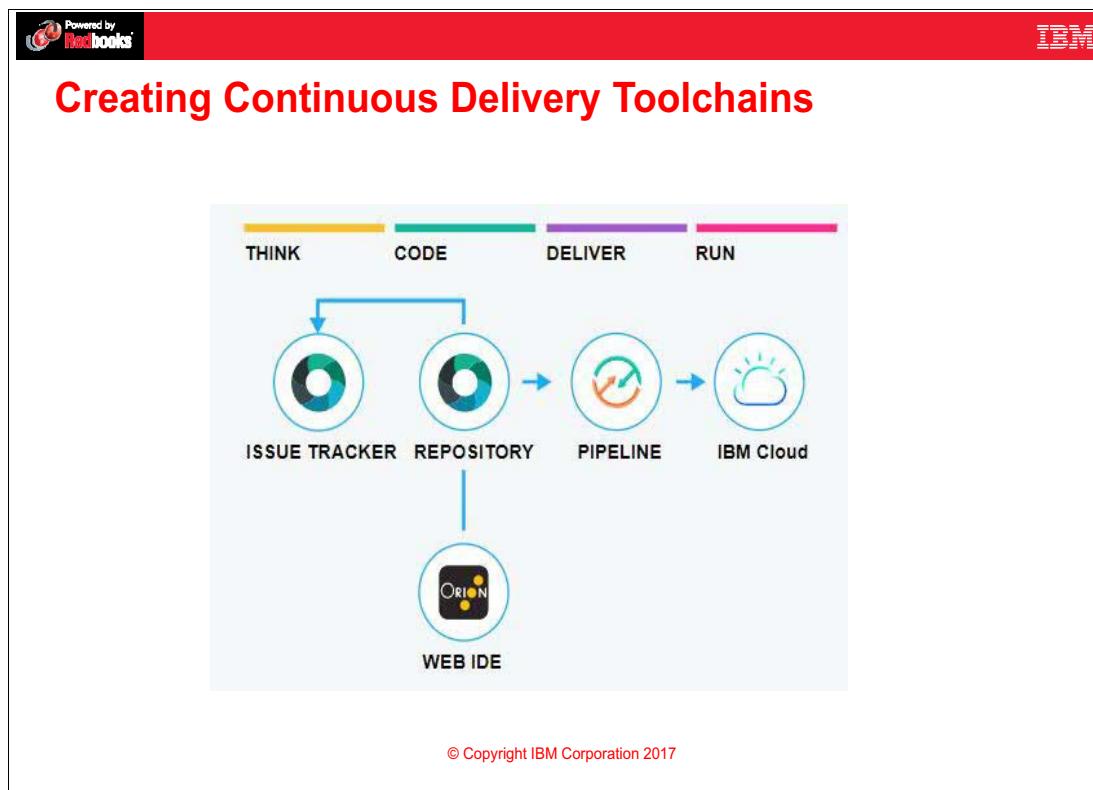


Figure 4-7 Creating Continuous Delivery Toolchains

Notes:

A *toolchain* is a set of tool integrations that support development, deployment, and operations tasks.

The UI to create a toolchain groups the tools into the following phases:

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

For more information about the phases of developing a Cloud Foundry app toolchain, refer to the IBM Cloud Garage method:

<https://ibm.biz/BdjTS3>

4.10 Adding Tools to the Toolchain

The screenshot shows the IBM Cloud Continuous Delivery interface. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below the header, the title "Adding Tools to the Toolchain" is displayed in red. A sub-header below it says "You can add tools into your continuous delivery toolchain." The main content area is titled "Toolchains / vy301-x23-nodejs". It features a grid of tool components categorized under "THINK", "CODE", and "DELIVER". Each component has a status indicator: "Issues vy301-x23-nodejs" is "Configured", "Git vy301-x23-nodejs" is "Configured", "Delivery Pipeline vy301-x23-nodejs" is "Configured", and "Eclipse Orion Web IDE" is also "Configured". A blue button labeled "Add a Tool + :)" is located at the top right of the grid. The footer of the page includes the copyright notice "© Copyright IBM Corporation 2017".

Figure 4-8 Adding Tools to the Toolchain

Notes:

To add tools, click **Add a Tool** from within a toolchain.

4.11 Adding Tool Integration to a Toolchain

The screenshot shows a grid of tool integration options. The columns are:

- Alert Notification:** Alert Notification (IBM Experimental)
- Artifactory:** Artifactory (Third-Party)
- Availability Monitoring:** Availability Monitoring (IBM)
- Other Tool:** Other Tool (Create a custom integration with any tool)
- Bitbucket:** Bitbucket (Third-Party)
- Cloud Event Management:** Cloud Event Management (IBM Experimental)
- Delivery Pipeline:** Delivery Pipeline (IBM)
- Slack:** Slack (Third-Party)
- DevOps Insights:** DevOps Insights (IBM Beta)
- Eclipse Orion Web IDE:** Eclipse Orion Web IDE (IBM)
- Git Repos and Issue Tracking:** Git Repos and Issue Tracking (IBM)
- SonarQube:** SonarQube (Third-Party)
- GitHub:** GitHub (Third-Party)
- GitHub Enterprise Whitewater:** GitHub Enterprise Whitewater (IBM)
- GitLab:** GitLab (Third-Party)
- UrbanCode Deploy:** UrbanCode Deploy (IBM Beta)
- Jenkins:** Jenkins (Third-Party)
- JIRA:** JIRA (Third-Party)
- Nexus:** Nexus (Third-Party)
- Sauce Labs:** Sauce Labs (Third-Party)
- PagerDuty:** PagerDuty (Third-Party)
- Rational Team Concert:** Rational Team Concert (IBM)

© Copyright IBM Corporation 2017

Figure 4-9 Adding Tool Integration to a Toolchain

Notes:

The collective power of a toolchain is greater than the sum of its individual tool integrations.

In the Tool Integrations section, select each tool integration that you want to configure for your toolchain. Consider the following examples:

- If you configure Sauce Labs, the toolchain is set up to enable adding Sauce Labs, which you can use to add test jobs to the pipelines.
- If you configure PagerDuty, the toolchain is set up to send alert notifications to the specified PagerDuty service when a major issue occurs.
- If you configure Slack, the toolchain is set up to send notifications to the specified Slack channel.
- If you configure a source code tool integration, such as GitHub, the sample repo can be cloned or forked (for example) into your GitHub account.

4.12 Web IDE: Edit Code features

Powered by
Redbooks

IBM

Web IDE: Edit Code features

- Web IDE:
 - Eclipse Orion
 - No installations, just start coding now
- IBM Cloud Live Sync for Node.js apps:
 - Live Edit
 - Debug

WEB IDE

© Copyright IBM Corporation 2017

Figure 4-10 Web IDE: Edit Code features

Notes:

The Web IDE features in IBM Cloud Continuous Delivery provide the following capabilities:

- ▶ Uses Eclipse Orion Web IDE.
- ▶ No installations required; use a browser to code.
- ▶ The Edit Code feature provides a workspace to develop source code and configuration files.
- ▶ Full-featured environment for writing your application code by using your web browser.
- ▶ Rich code completion capabilities for CSS, HTML, and JavaScript.
- ▶ Deploy, stop, and run applications from the Run bar.
- ▶ View the logs from the Run bar.

IBM Cloud Live Sync features (as of this writing, available for Node.js applications only):

- ▶ Live Edit: Allows you to make changes to your application from the Web IDE without the need to redeploy it.
- ▶ Debug: When a Node.js application is in Live Edit mode, you can shell into it and debug it. You can edit code dynamically, insert breakpoints, step through code, restart the runtime, and more by using the Node Inspector debugger.

4.13 Web integrated development environment

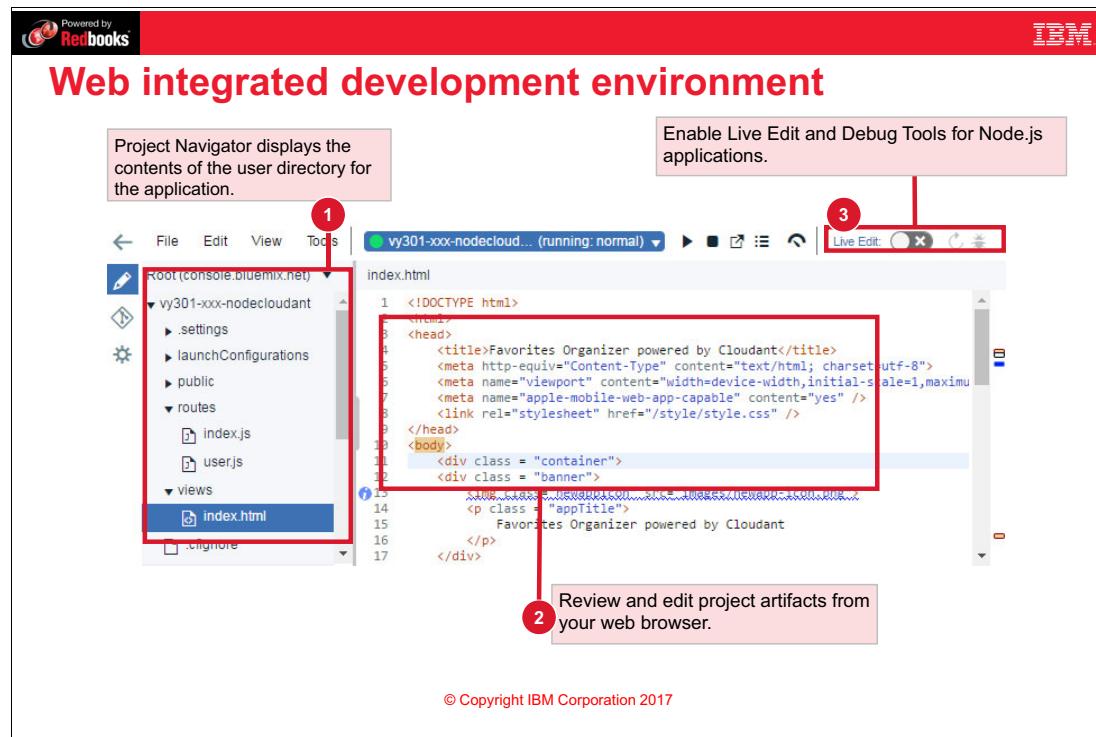


Figure 4-11 Web integrated development environment

Notes:

The Edit Code perspective for Eclipse Orion Web IDE is shown in Figure 4-11.

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

This view is for your local workspace on IBM Cloud. To commit the code changes, switch to the GIT view (described next).

4.14 Editing source code

The screenshot shows a web-based code editor interface. At the top left is the Redbooks logo with the text "Powered by Redbooks". At the top right is the IBM logo. The main title "Editing source code" is displayed in red at the top of the editor window. Below the title is a bulleted list of features:

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

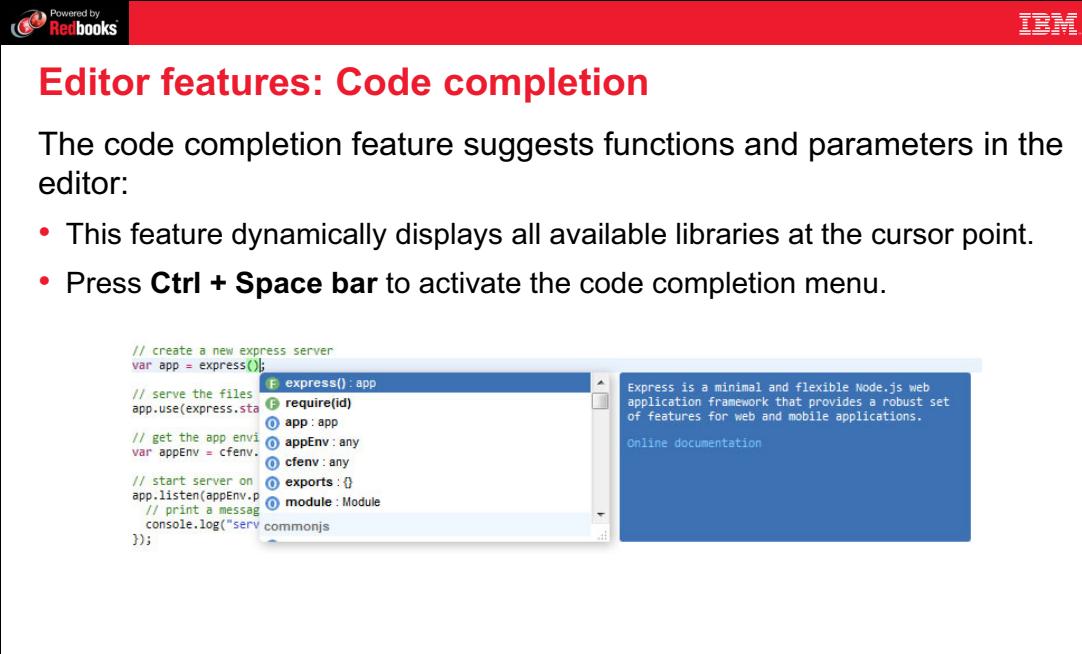
At the bottom center of the editor window, there is a small copyright notice: "© Copyright IBM Corporation 2017".

Figure 4-12 Editing source code

Notes:

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

4.15 Editor features: Code completion



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

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

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

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

Online documentation

Figure 4-13 Editor features: Code completion

Notes:

To use the code completion shortcut, place your cursor within the editor and press **Ctrl + Space bar**. This feature displays the libraries that are available at the cursor point, including third-party modules that you imported in the script. For example, as shown in Figure 4-13, the code completion feature displays functions and templates for the Express web application framework for Node.js because this point is the cursor point.

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

4.16 Editor features: Run bar

Editor features: Run bar

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

- 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 **Delivery Pipeline**.

© Copyright IBM Corporation 2017

Figure 4-14 Editor features: Run bar

Notes:

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

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

4.17 IBM Cloud Live Sync features

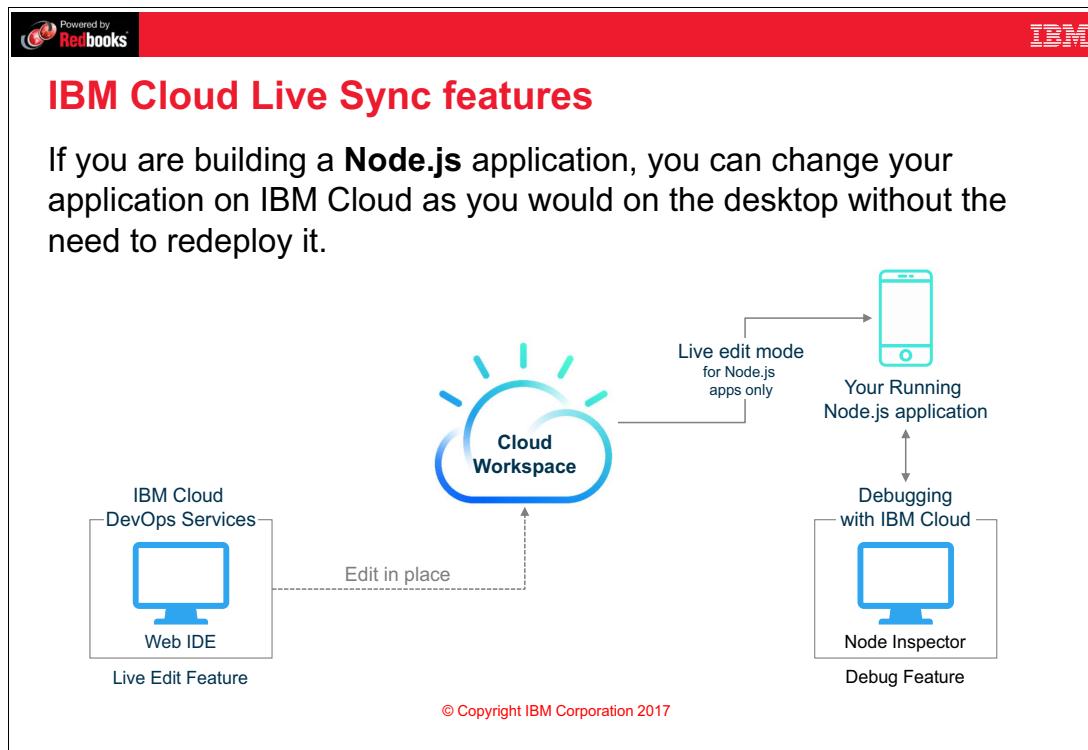


Figure 4-15 Cloud Live Sync features

Notes:

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

When you make a change, you can see that change in your running IBM Cloud application immediately (without the need to recompile and redeploy).

IBM Cloud Live Sync works from the Web IDE and consists of the following features for Node.js applications:

- ▶ **Live Edit**

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

- ▶ **Debug**

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

Note: You must use a Chrome browser to use the Debug feature.

4.18 Git Repos and Issue Tracking

The screenshot shows a presentation 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 'Git Repos and Issue Tracking' is centered in a large, bold, red font. Below the title is a bulleted list of features:

- Collaborate with your team and manage your source code with a Git repository (repo) and issue tracker that is hosted by IBM and built on GitLab Community Edition.
- The Git Repos and Issue Tracking tool integration supports teams to manage code and collaborate in many ways:
 - Manage Git repositories through fine-grained access controls that keep code secure
 - Review code and enhance collaboration through merge requests
 - Track issues and share ideas through the issue tracker
 - Document projects on the wiki system

© Copyright IBM Corporation 2017

Figure 4-16 Git Repos and Issue Tracking

4.19 Source control with a Git repository

Source control with a Git repository

- When you enable continuous delivery for an application, a Git repository is created to manage your source code.
- When you develop your application online, you can enter common Git commands through the Web IDE.
- If you develop your application on your own workstation, use a Git client to synchronize your own workspace and push your changes to the Git repository.
- For more information, see the open source Git project:

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

Figure 4-17 Source control with a Git repository

Notes:

By default, enabling Continuous Delivery for an application creates a Continuous Delivery Toolchain for your application. This Toolchain includes a Git repository that is based on GitLab. Git is an open-source change management system.

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

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

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

4.20 Git repository overview

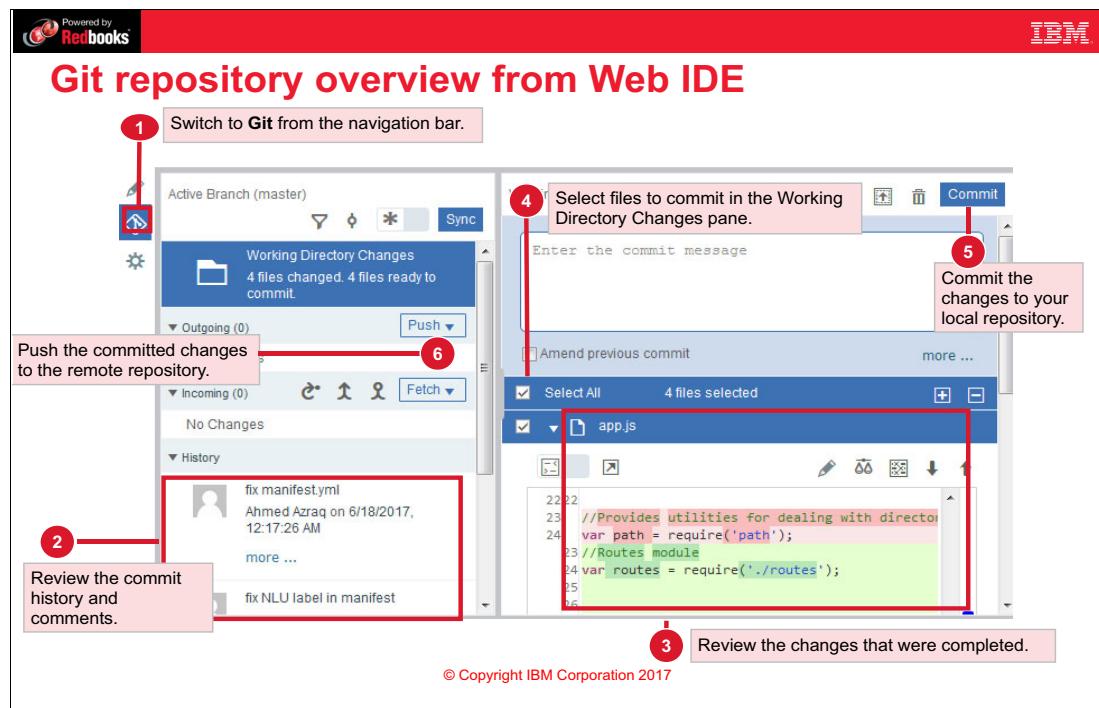


Figure 4-18 Git repository overview

Notes:

The numbers that are shown in Figure 4-18 correspond to the following steps:

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

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

4.21 Connect a Git client to your repository on IBM Cloud

The screenshot shows a Redbooks page with a red header bar. On the left is the Redbooks logo, and on the right is the IBM logo. The main title 'Connect a Git client to your repository on IBM Cloud' is in red. Below it is a bulleted list of instructions:

- If you want to develop your application on your own workstation, use a Git client to save your changes to the git repository that is provided by IBM Cloud continuous delivery.
- 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://git.ng.bluemix.net/$username/$appName
```
- 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 application description.'  
git push
```

© Copyright IBM Corporation 2017

Figure 4-19 Connect a Git client to your repository on IBM Cloud

Notes:

You can still develop your application on your own local workstation.

If you do not have a Git client installed, download and install the latest version of the Git client from this website:

<https://git-scm.com/downloads>

To verify that the installation is successful, issue the `git --version` command from the command line.

To find the URL of the git repository that is associated with your application, click **View toolchain** from the Application Details; then, click **Git**.

Start by retrieving a copy of the IBM Cloud application source code by using the `git clone` command. Issue `git clone $git-URL` on your Command Prompt. 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.

To save the updated source code, run the `git add` command to stage the files to be committed.

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

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

4.22 Issue Tracking tool

The screenshot shows a web-based application interface for issue tracking. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "Issue Tracking tool" is displayed in large red text. A main content area contains the text: "IBM Cloud Continuous Delivery toolchain includes a tool to track issues". To the right of this text is a small screenshot of a dashboard titled "Issues" with the identifier "vy301-xxx-nodecloudant" and a status indicator "✓ Configured". Below this, there's a section with the heading "The Issue Tracker is the place to add things that need to be improved or solved in a project". It also states: "Issues can be bugs, tasks or ideas to be discussed. Also, issues are searchable and filterable." A green button labeled "New issue" is visible. To the right of the text is a light gray illustration featuring a magnifying glass over a circular target, with dashed lines connecting to icons of a rocket ship and a gear.

Figure 4-20 Issue Tracking tool

Notes:

The Issue Tracking tool (Issue Tracker in Figure 4-20) is part of Git Repos and Issue Tracking. It is used to track items (issues) that require resolution or improvements in a project.

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

4.23 Creating an issue

Creating an issue

To create a bug or task, click **New issue**

New Issue

Title

Add description templates to help your contributors communicate effectively!

Description

Write [Preview](#) B I

Write a comment or drag your files here...

Markdown and slash commands are supported [Attach a file](#)

This issue is confidential and should only be visible to team members with at least Reporter access.

Assignee Assignee [Assign to me](#) Due date

Milestone Milestone

Labels Labels

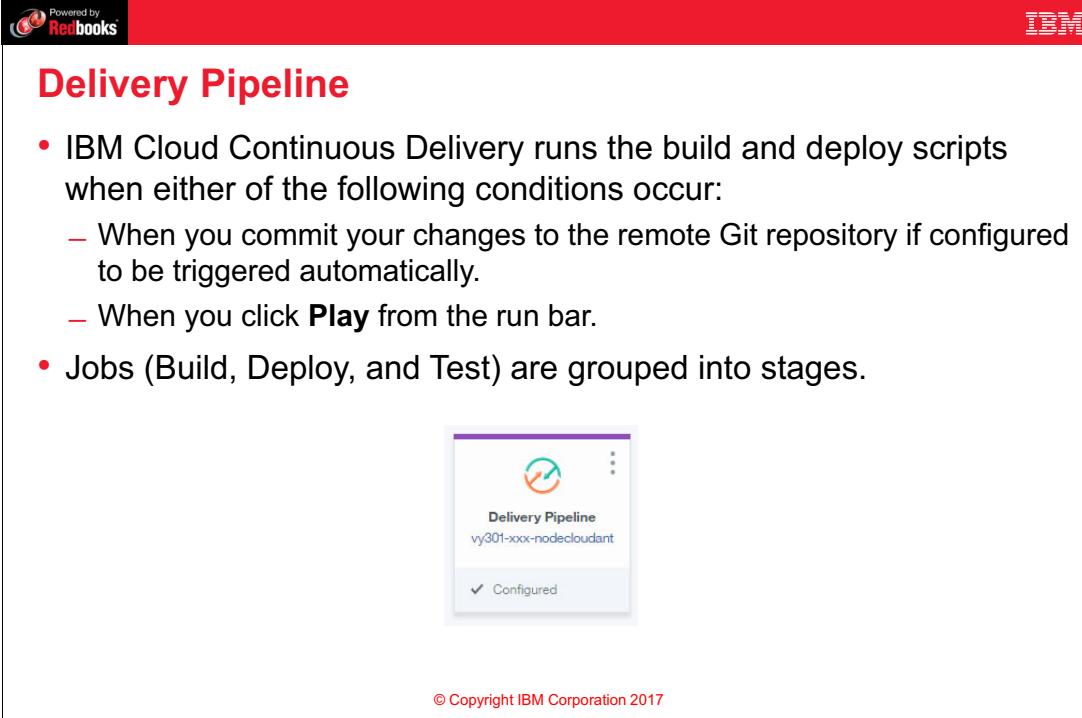
[Submit issue](#) [Cancel](#)

Figure 4-21 Creating an issue

Notes:

Complete the form that is shown in Figure 4-21. Then, click **Submit issue** to create the bug or task.

4.24 Delivery Pipeline



The screenshot shows a delivery pipeline card titled "Delivery Pipeline" with the identifier "vy301-xxx-nodecloudant". The card indicates it is "Configured". The IBM logo is visible in the top right corner.

Delivery Pipeline

- IBM Cloud Continuous Delivery runs the build and deploy scripts when either of the following conditions occur:
 - When you commit your changes to the remote Git repository if configured to be triggered automatically.
 - When you click **Play** from the run bar.
- Jobs (Build, Deploy, and Test) are grouped into stages.

© Copyright IBM Corporation 2017

Figure 4-22 Delivery Pipeline

Notes:

By default, IBM Cloud Continuous Delivery automatically runs the build and deploy tasks when you commit changes to the Git repository. The pipeline also features a “run” icon that runs a stage of the pipeline but only runs on committed code or builds, depending on the stage.

You can access the Continuous Delivery toolchain by clicking **View toolchain** from the Application Details page. To browse to the delivery pipeline (or any other tool), click its tool card in the toolchain overview page.

Jobs (Build, Deploy, and Test) are grouped into stages. You can change the order of the stages by moving them by dragging and dropping.

4.25 Customizing Delivery Pipeline

Powered by
Redbooks

IBM

Customizing Delivery Pipeline

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

Deploy Stage

INPUT JOBS ENVIRONMENT PROPERTIES

Deploy > Test

ADD JOB

© Copyright IBM Corporation 2017

Figure 4-23 Customizing Delivery Pipeline

Notes:

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

4.26 Example: Default delivery pipeline

The screenshot shows the IBM Cloud Delivery Pipeline interface. It consists of two main sections: 'Build Stage' on the left and 'Deploy Stage' on the right, separated by a horizontal arrow pointing from left to right.

Build Stage:

- STAGE PASSED:** A green bar at the top.
- LAST INPUT:** Shows a commit icon and the text "Last commit by Ahmed Azraq 23h ago fix manifest.yml".
- JOB:** A card showing a green checkmark, the word "Build", and the status "Passed 23h ago".
- LAST EXECUTION RESULT:** Shows a build artifact icon and the text "Build 3".

Deploy Stage:

- STAGE PASSED:** A green bar at the top.
- LAST INPUT:** Shows the text "Stage: Build Stage / Job: Build".
- JOB:** A card showing a green checkmark, the word "Deploy", and the status "Passed 23h ago".
- LAST EXECUTION RESULT:** Shows a deployment artifact icon and the text "vy102-XXX-express vy102-XXX-express.mybluemix.net".

A red callout box on the right side contains the following numbered points:

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

At the bottom of the interface, there is a copyright notice: © Copyright IBM Corporation 2017.

Figure 4-24 Example: Default delivery pipeline

Notes:

The default build script is simple: It takes the files that are pushed to the remote Git repository and triggers IBM Cloud to build your code in the server runtime.

After the build stage completes successfully, the deploy stage runs. Delivery Pipeline deploys the built files to your IBM Cloud space.

4.27 Configuring the Build Stage

Configuring the Build Stage

Build Stage

INPUT JOBS ENVIRONMENT PROPERTIES

Input Settings

Input Type: Git Repository

Git Repository: vy102-XXX-express-1

Git URL: https://git.ng.bluemix.net/aazraq/vy102-XXX-express-1.git

Branch: master

Stage Trigger

Run jobs whenever a change is pushed to Git

Run jobs only when this stage is run manually

SAVE CANCEL

① By default, IBM Cloud DevOps runs the Build stage when a client pushes any change to the master branch in the remote Git repository.

② It is not necessary for you to run the build stage manually. However, you can disable automatic builds by using the State Trigger setting.

© Copyright IBM Corporation 2017

Figure 4-25 Configuring the Build Stage

Notes:

By default, Delivery Pipeline runs the Build stage when a client pushes any change to the master branch in the remote Git repository.

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

4.28 Configuring Build jobs

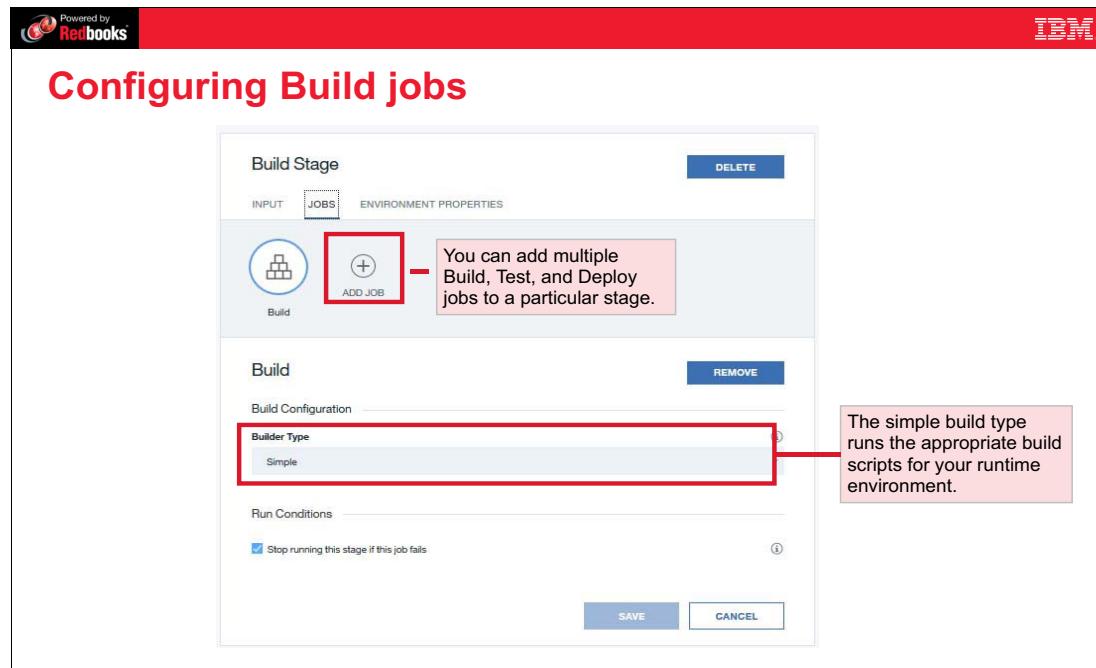


Figure 4-26 Configuring Build jobs

Notes:

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

At the time of this writing, Builder Type supports the following types:

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

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

4.29 Configuring Deploy jobs

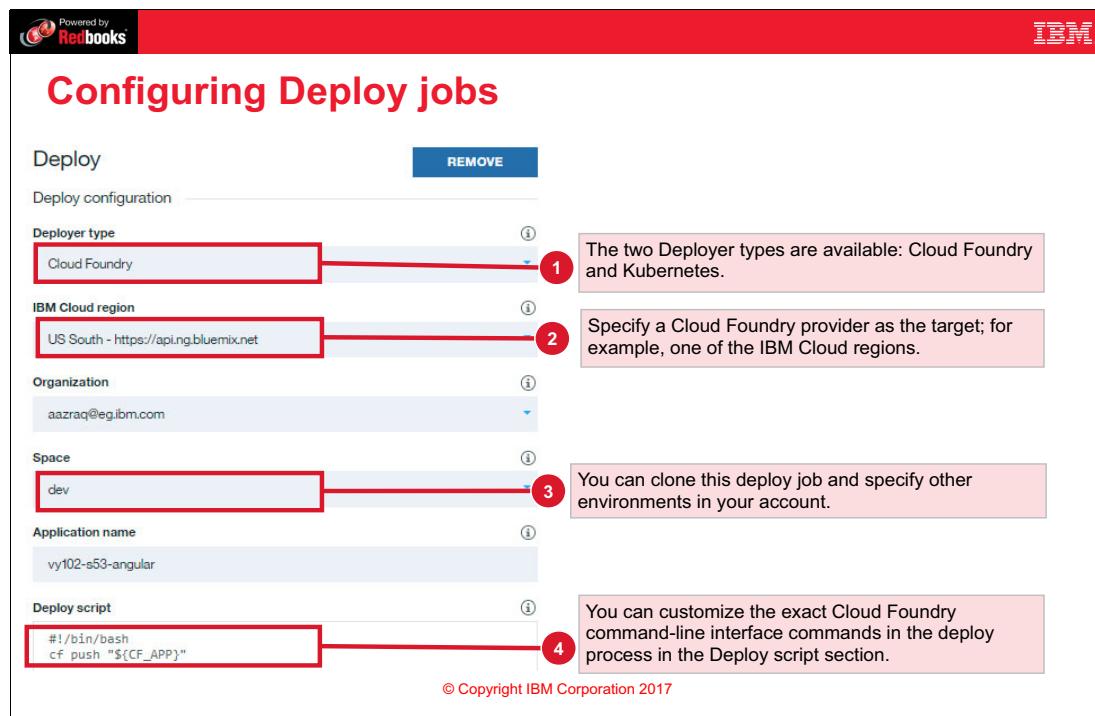


Figure 4-27 Configuring Deploy jobs

Notes:

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

The following Deployer Types are available:

- ▶ Cloud Foundry: Deploys applications to Cloud Foundry servers. This type is the default deployer type for the deploy job.
- ▶ Kubernetes: Provides a set of default deployment patterns for containers on Kubernetes Cluster. Containers are typically built by using the Docker Builder.

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

You can customize the Cloud Foundry command-line interface commands (cf CLI) in the deploy process in the Deploy Script section. The default deploy action is equivalent to running the cf push command from the CLI. You also can add custom shell script commands. (\${CF_APP} in Figure 4-27 refers to the application name).

4.30 Configuring Test job

The screenshot shows the 'Configuring Test job' interface. At the top, there's a navigation bar with the Redbooks logo and the IBM logo. Below the navigation bar, the title 'Configuring Test job' is displayed. The main area is divided into two sections: 'Deploy Stage' and 'Test'. In the 'Deploy Stage', there are two jobs: 'Deploy' and 'Test'. The 'Test' job is currently selected, indicated by a blue circle around it. Below the 'Test' section, there are fields for 'Test Configuration', specifically 'Tester Type' set to 'Simple', and 'Test Command' containing the script '#!/bin/bash # invoke tests here'. A 'REMOVE' button is located at the top right of the 'Test' section.

Figure 4-28 Configuring Test job

Notes:

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

The following Tester types are available:

- ▶ Simple
- ▶ Advanced Tester
- ▶ DevOps Insights Gate
- ▶ IBM AppScan® Dynamic Analyzer
- ▶ IBM Security Static Analyzer
- ▶ Sauce Labs
- ▶ Vulnerability Adviser

4.31 Example: A successful build and deploy result

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

- Build Stage:**
 - STAGE PASSED**
 - LAST INPUT:** Last commit by Ahmed Azraq, 23h ago, from [git://github.com/ahmedazraq/test.git](#).
 - JOBS:** Build 3 (Passed 23h ago). A red box highlights this job.
 - LAST EXECUTION RESULT:** Build 3. A red box highlights this entry.
 - A callout box with a red border and a red circle containing the number 1 points to the "Build 3" entry in the Jobs section.
 - A callout box with a red border and a red circle containing the number 3 points to the "Build 3" entry in the Last Execution Result section.
- Deploy Stage:**
 - STAGE PASSED**
 - LAST INPUT:** Stage: Build Stage / Job: Build.
 - JOBS:** Deploy 3 (Passed 23h ago). A red box highlights this job.
 - LAST EXECUTION RESULT:** vy102-XXX-express, vvt02-XXX-express.mybluemix.net. A green dot indicates it is running. A red box highlights this entry.
 - A callout box with a red border and a red circle containing the number 2 points to the "Deploy 3" entry in the Jobs section.

Figure 4-29 Example: A successful build and deploy result

Notes:

The default settings for the Delivery Pipeline are the following stages:

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

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

The LAST EXECUTION RESULT pane shows that the application was successfully deployed to your IBM Cloud account, and that the application is running on IBM Cloud.

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 Cloud account by clicking the settings wheel icon on top of Deploy Stage and selecting Clone Stage.

4.32 Unit summary

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

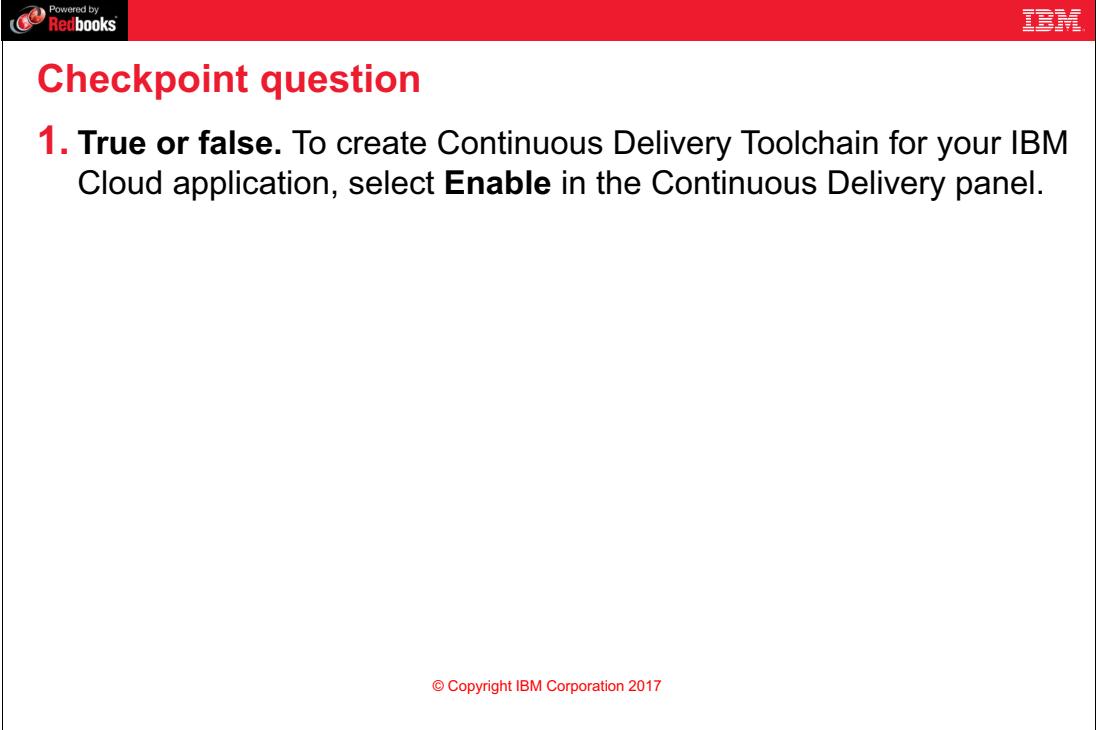
Unit summary

- Described DevOps
- Described the capabilities of IBM Cloud Continuous Delivery
- Identified the Web IDE features in IBM Cloud Continuous Delivery
- Described how to use Git Repos and Issue Tracking
- Explained the pipeline build and deploy processes

© Copyright IBM Corporation 2017

Figure 4-30 Unit summary

4.33 Checkpoint question



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "Checkpoint question" is in red. Below it is a numbered list.

1. True or false. To create Continuous Delivery Toolchain for your IBM Cloud application, select **Enable** in the Continuous Delivery panel.

© Copyright IBM Corporation 2017

Figure 4-31 Checkpoint question

4.34 Checkpoint answer

Powered by
Redbooks

IBM

Checkpoint answer

- 1. True.** To create Continuous Delivery Toolchain for your IBM Cloud application, select **Enable** in the Continuous Delivery panel.

© Copyright IBM Corporation 2017

Figure 4-32 Checkpoint answer



REST architecture and Watson APIs

This unit covers the following topics:

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

5.1 What you should be able to do

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

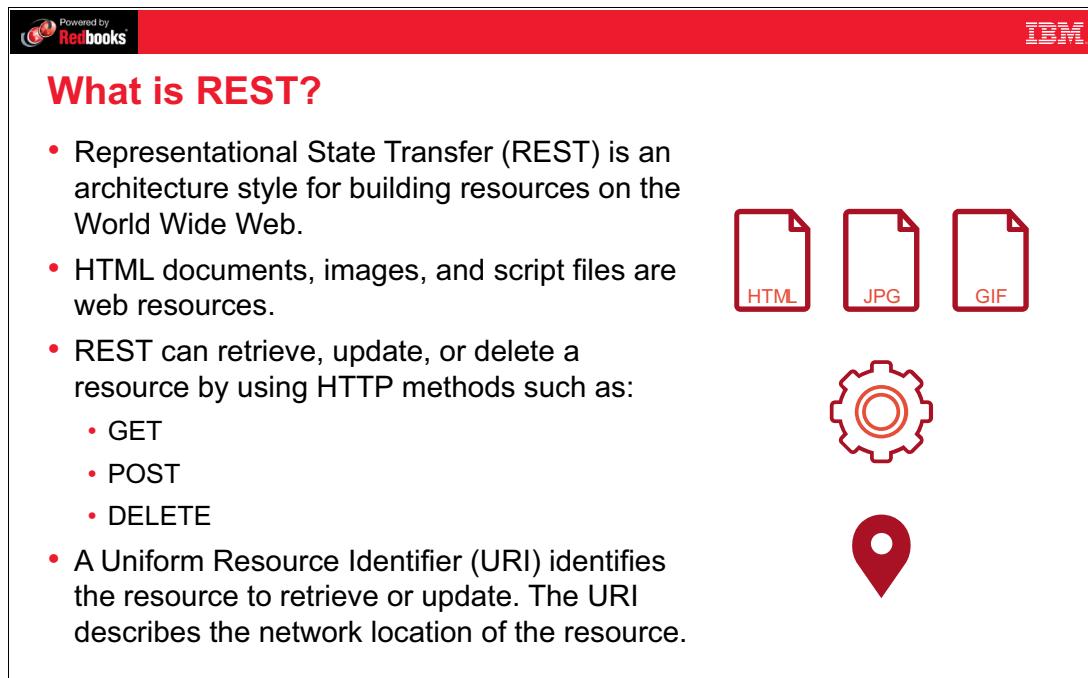
- ▶ Describe the characteristics of REST APIs.
- ▶ Explain the advantages of the JSON data format.
- ▶ Provide examples of REST APIs using IBM Watson.

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/>
- ▶ IBM CEO Ginni Rometty describes a new era in technology and business:
<https://www.youtube.com/watch?v=bMLYKhiZCVI>

5.3 What is REST?



Powered by Redbooks

IBM

What is REST?

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



Figure 5-1 What is REST?

Notes:

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

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

REST provides the following HTTP methods:

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

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

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

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

or

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

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

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

5.4 Applying REST to server-side applications



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 make available your server applications as a web resource:
 - A REST service is an entry point to an 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 2017

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

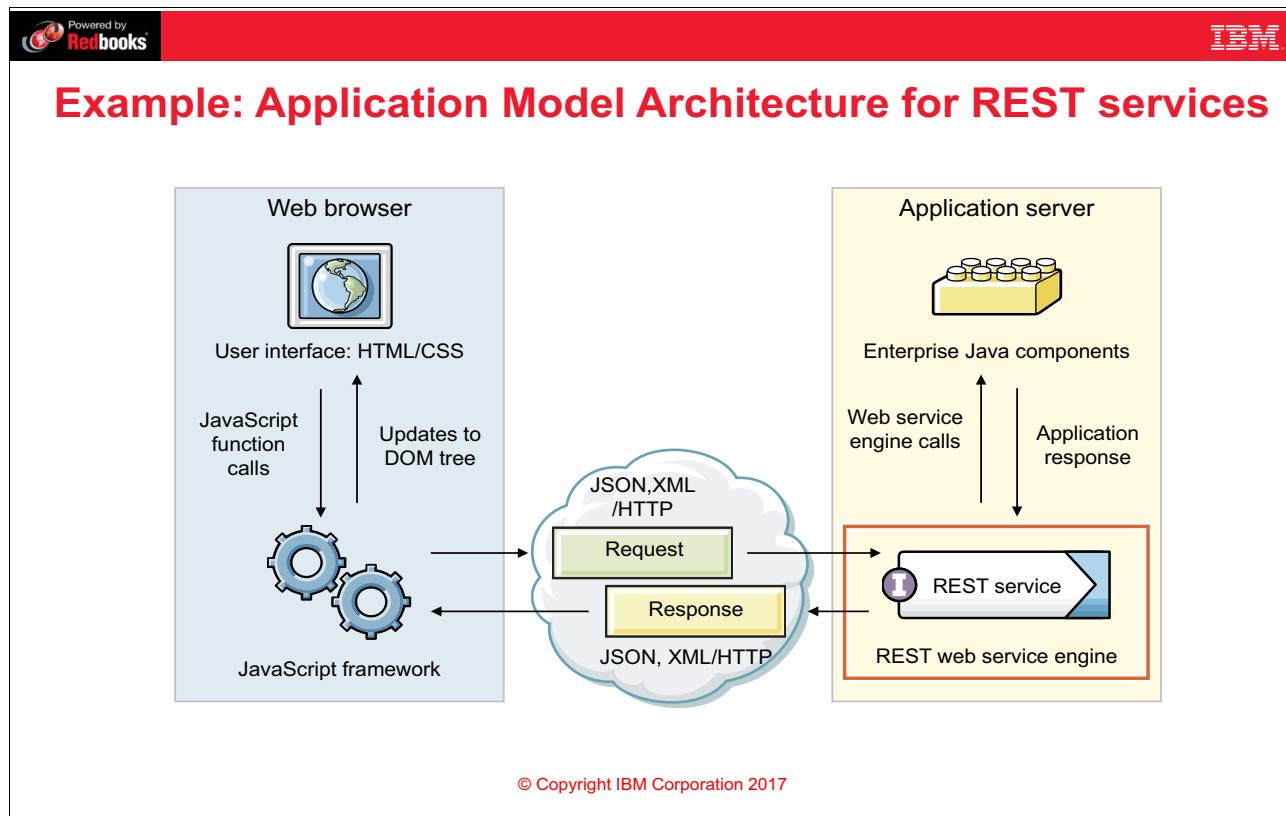


Figure 5-3 Example: Application model architecture for REST services

Notes:

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

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

5.6 What is a RESTful web service?



IBM

What is a RESTful web service?

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

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

Figure 5-4 What is a RESTful web service?

Notes:

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

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

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

5.7 Example: Sending an HTTP request to a REST service

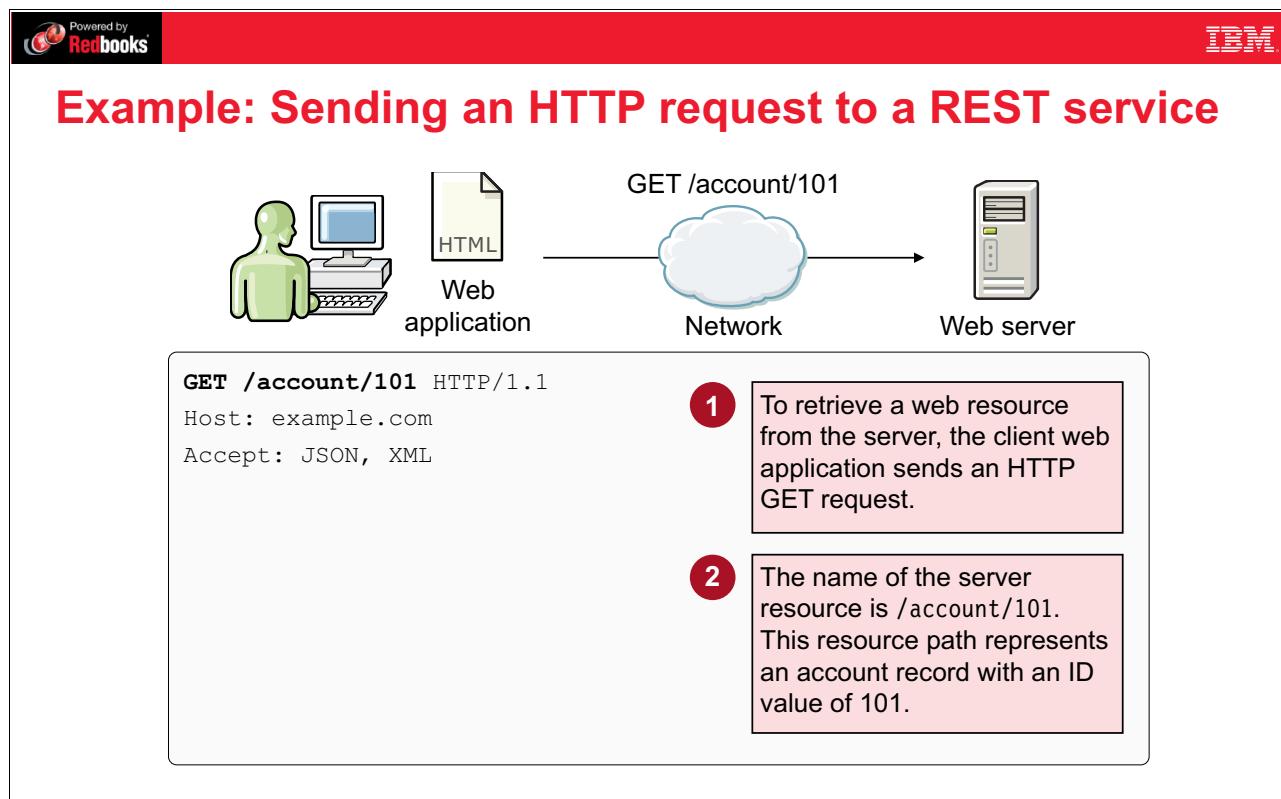


Figure 5-5 Example: Sending an HTTP request to a REST service

Notes:

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

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

5.8 Example: Receiving an HTTP response from a REST service

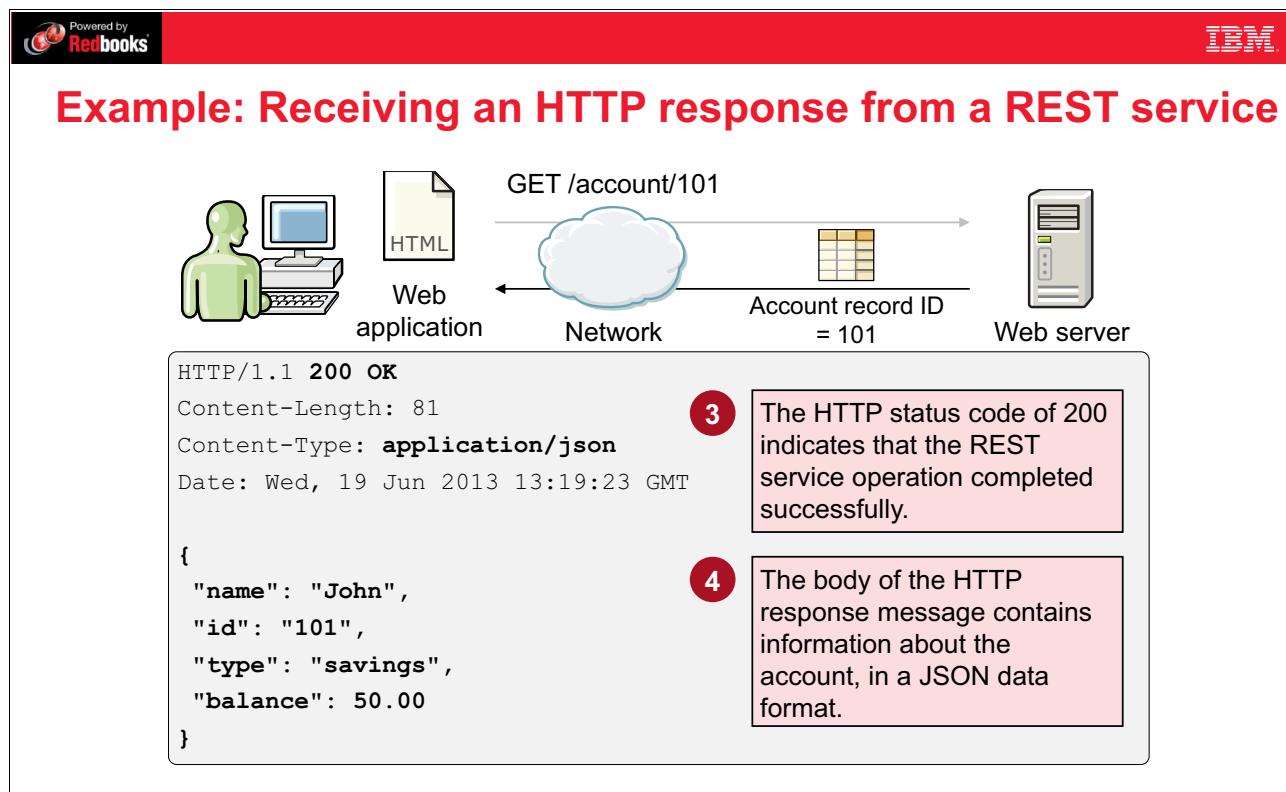


Figure 5-6 Example: Receiving an HTTP response from a REST service

Notes:

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

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

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

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

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

5.9 REST characteristics

 Powered by
Redbooks



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 2017

Figure 5-7 REST characteristics

Notes:

REST has the following characteristics:

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

5.10 Introduction to JSON



Powered by
Redbooks

IBM

Introduction to JSON

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

Figure 5-8 Introduction to JSON

Notes:

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

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

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

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

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

5.11 JSON data types

The screenshot shows a slide titled "JSON data types" from a Redbooks presentation. The slide is sponsored by IBM. It lists six JSON data types with examples and definitions:

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

Figure 5-9 JSON data types

Notes:

JSON features the following data types:

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

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

5.12 JSON data type: Objects



JSON data type: Objects

An unordered collection of key/value pairs:

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

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

© Copyright IBM Corporation 2017

Figure 5-10 JSON data type: Objects

Notes:

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

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

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

5.13 JSON data type: Arrays



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]
]
```

Figure 5-11 JSON data type: Arrays

Notes:

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

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

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

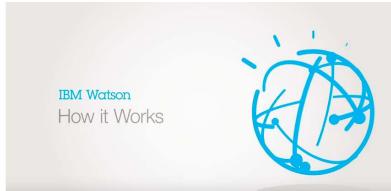
JSON must start with an object or an array at the top level.

5.14 What is IBM Watson?

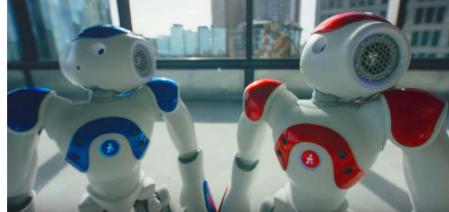
Powered by Redbooks

What is IBM Watson?

IBM Watson is the artificial intelligence (AI) offering from IBM. Watson uses natural language processing (NLP), computer vision, and machine learning technologies to reveal insights from large amounts of unstructured data.



IBM Watson
How it Works



IBM Watson: How it Works
Learn how IBM Watson works and has similar thought processes to a human:
<https://www.youtube.com/watch?v=Xcmh1LQB9I>

The future of IBM Watson
Learn about insights into the future of Watson and cognitive computing:
<https://www.youtube.com/watch?v=L5QJs6byoal>

© Copyright IBM Corporation 2017

Figure 5-12 What is Watson?

Notes:

IBM Watson is the artificial intelligence (AI) offering from IBM. It uses natural language processing (NLP), computer vision, and machine learning technologies to reveal insights from large amounts of unstructured data.

Watson analyzes unstructured data. Unstructured data is not contained in a regular database and is growing exponentially, making up most of all the data in the world.

Unstructured data includes news articles, research reports, and social media posts.

5.15 Watson Services in IBM Cloud



The screenshot shows the 'Watson services in IBM Cloud' catalog page. At the top left is the 'Powered by Redbooks' logo, and at the top right is the 'IBM' logo. The main title 'Watson services in IBM Cloud' is centered above a grid of nine service cards. Each card includes a small icon, the service name, a brief description, and two buttons: 'Lite' and 'IBM'. The services listed are:

- Conversation**: Add a natural language interface to your application. Buttons: Lite, IBM.
- Discovery**: Add a cognitive search and content analytics engine to... Buttons: Lite, IBM.
- Language Translator**: Translate text from one language to another for spec... Buttons: Lite, IBM.
- Natural Language Classifier**: Natural Language Classifier performs natural language... Button: IBM.
- Natural Language Understanding**: Analyze text to extract meta-data from content such as... Buttons: Lite, IBM.
- Personality Insights**: The Watson Personality Insights derives insights from... Buttons: Lite, IBM.
- Speech to Text**: Low-latency, streaming transcription. Buttons: Lite, IBM.
- Text to Speech**: Synthesizes natural-sounding speech from text. Buttons: Lite, IBM.
- Tone Analyzer**: Tone Analyzer uses linguistic analysis to detect three types... Buttons: Lite, IBM.
- Visual Recognition**: Find meaning in visual content! Analyze images for scenes, ... Button: IBM.

At the bottom center of the page is the copyright notice: '© Copyright IBM Corporation 2017'.

Figure 5-13 Watson services in IBM Cloud

Notes:

Watson enables you to build cognitive apps that help enhance, scale, and accelerate human expertise.

The following Watson services are available in the IBM Cloud catalog:

- ▶ **Natural Language Understanding:** Analyzes unstructured text content. Use it to extract semantic meta-data from content, such as information about people, places, companies, topics, facts, relationships, authors, and languages.
- ▶ **Language Translator:** Translates text from one language to another.
- ▶ **Personality Insights:** Extracts and analyses a spectrum of personality attributes to help discover actionable insights about people and entities, and in return guides users to highly personalized interactions. It takes text as an input and outputs the analysis of the personality according to the input text. For more information about an interactive demonstration of this service, see the following website:
<https://personality-insights-livedemo.mybluemix.net/>
- ▶ **Speech to text:** This Watson service converts the human voice into written words. At the time of this writing, English, Japanese, Arabic, Mandarin, Portuguese, Spanish, and French are supported.

5.16 Watson API Explorer

The screenshot shows the Watson API Explorer interface. At the top left is the Redbooks logo with the text "Powered by Redbooks". At the top right is the IBM logo. The main title "Watson API Explorer" is in red at the top center. Below it is a bulleted list:

- Watson APIs are exposed as REST APIs
- Watson API Explorer provides documentation for Watson REST APIs
- For more information about Watson API Explorer, see:
<https://watson-api-explorer.mybluemix.net/>

To the right of the list is a sidebar with the "Watson Developer Cloud" menu. The "Watson API Explorer" section is highlighted in purple, showing the text "A collection of Swagger documentation for the Watson APIs". Below this is a "Language" section with icons and links to various Watson APIs:

- AlchemyLanguage
- Conversation
- Dialog (Deprecated)
- Document Conversion
- Language Translator
- Natural Language Classifier

At the bottom of the sidebar is the copyright notice: "© Copyright IBM Corporation 2017".

Figure 5-14 Watson API Explorer

Notes:

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

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

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

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

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

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

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

On the right, under 'Response', is a JSON object representing the API's response:

```
{ "retrieved_url": "http://www.reuters.com/article/us-africa-japan-idUSKCN1QD0JU", "metadata": { "title": "Japan pledges $30 billion for Africa over next three years", "publication_date": "2016-08-27T00:00:00", "image": "", "feeds": [ { "link": "http://www.reuters.com/tools/rss" } ], "authors": [ { "name": "George Obulutsa" } ] }, "language": "en" }
```

Below the Request and Response sections, a copyright notice reads: © Copyright IBM Corporation 2017

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

Notes:

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

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

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

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

The following parameters are used in the example:

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

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

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

The service URL is

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

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

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

5.18 IBM Watson in our daily life

Powered by Redbooks

IBM Watson in our daily life

Watson and the Jeopardy! Challenge
See how Watson won Jeopardy! and what it meant for the future of cognitive systems:
<https://www.youtube.com/watch?v=P18EdAKuC1U>

Morgan | IBM Creates First Movie Trailer by AI [HD] | 20th Century FOX
Scientists at IBM Research collaborated with 20th Century Fox to create the first-ever cognitive movie trailer for the movie Morgan:
<https://www.youtube.com/watch?v=qJEzuYynaiw>

World's First Cognitive Dance Party - Daybreaker with Watson
IBM Watson and Daybreaker hosted the World's First Cognitive Dance Party in San Francisco:
<https://www.youtube.com/watch?v=rMqrs3nBU10>

© Copyright IBM Corporation 2017

Figure 5-16 IBM Watson in our daily life

5.19 Unit summary

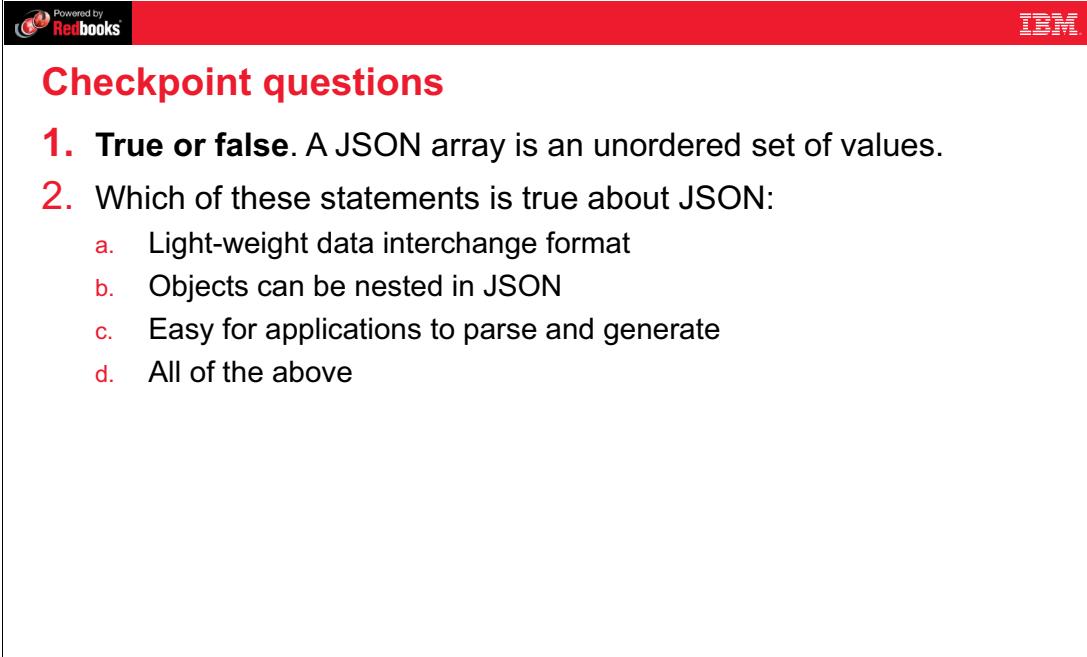
The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top left of this area is the title "Unit summary" in red. Below it is a bulleted list of three items:

- Described the characteristics of REST APIs.
- Explained the advantages of the JSON data format.
- Described an example of REST APIs that use Watson.

At the bottom center of the slide is the copyright notice "© Copyright IBM Corporation 2017".

Figure 5-17 Unit summary

5.20 Checkpoint questions



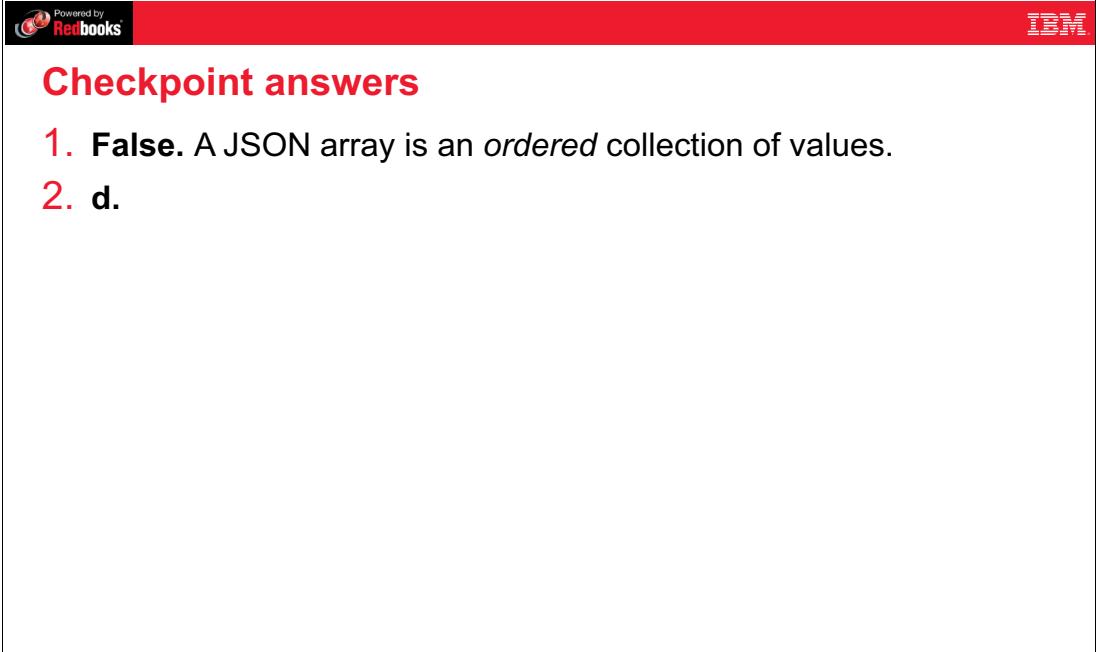
The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo.

Checkpoint questions

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

Figure 5-18 Checkpoint questions

5.21 Checkpoint answers



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

Checkpoint answers

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

Figure 5-19 Checkpoint answers



Introduction to data services in IBM Cloud

This unit covers the following topics:

- ▶ Database services in IBM Cloud
- ▶ IBM Cloudant

6.1 What you should be able to do

After completing this unit, you should be able to:

- ▶ Describe the benefits of IBM Cloudant.
- ▶ Explain how Cloudant databases and documents are accessed from IBM Cloud.
- ▶ 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 documentation:
<https://console.bluemix.net/docs/services/Cloudant/getting-started.html>
- ▶ IBM Cloud Data Services documentation:
<https://developer.ibm.com/clouddataservices/>
- ▶ IBM Cloudant videos on YouTube:
<https://www.youtube.com/channel/UCSMx6Fgq1RJLq58em2mJeKQ>
- ▶ IBM developerWorks articles and resources:
<https://www.ibm.com/developerworks>
- ▶ Online Learning Labs:
<https://www.ibm.com/cloud/garage/category/courses>
- ▶ Get IBM Cloud Essentials Open Badge:
<https://developer.ibm.com/courses/all/bluemix-essentials/>

6.3 Database choices on IBM Cloud



Database choices on IBM Cloud

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

© Copyright IBM Corporation 2017

Figure 6-1 Database choices on IBM Cloud

Notes:

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

- ▶ NoSQL databases have the advantage of storing objects in the same format that is expected by applications. Data is transferred and stored in standard formats, such as JSON, or XML. To perform a more complex query on a NoSQL database, you typically must create a view. A view is a self-maintained index that the database management system provides for quickly traversing the data set.
- ▶ SQL database service provides databases that are based on relational tables. They were 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).
 - ID:1, Name: Ahmed, Age: 29, Weight: 65
 - ID:2, Name: Ben, Age:34, Weight: 70
 - ID:3, Name: John, Age: 32, Weight: 73
- ▶ Columnar database is the database that stores data in form of columns instead of rows; for example, to model the following information:
 - ID:1, Name: Ahmed, Age: 29, Weight: 65
 - ID:2, Name: Ben, Age:34, Weight: 70
 - ID:3, Name: John, Age: 32, Weight: 73
- ▶ Each column is stored in the following database records:
 - Ahmed:1, Ben: 2, John: 3
 - 29:1, 34:2, 32:3
 - 65:1, 70:2, 73:3
- ▶ Key-value pair data services allow efficient storage of key-value pair data.

6.4 Data services in IBM Cloud catalog

The screenshot shows the 'Data services in IBM Cloud catalog' page. At the top left is the 'Powered by Redbooks' logo, and at the top right is the 'IBM' logo. Below the header, there are ten service cards arranged in two columns of five. Each card includes a small icon, the service name, a brief description, and two buttons: 'Lite' and 'IBM'. The services listed are:

- Apache Spark**: IBM Analytics for Apache Spark for Bluemix. (Lite, IBM)
- Cloudant NoSQL DB**: Cloudant NoSQL DB is a fully managed data layer designed. (Lite, IBM)
- Compose Enterprise**: IBM Compose Enterprise is a service which provides a private. (IBM)
- Compose for Elasticsearch**: Elasticsearch combines the power of a full text search engine with the. (IBM)
- Compose for etcd**: etcd is a key/value store developers can use to hold the always-correct. (IBM, Beta)
- Compose for JanusGraph**: JanusGraph is a scalable graph database optimized for storing. (IBM, Beta)
- Compose for MongoDB**: MongoDB with its powerful indexing and querying, aggregation and. (IBM)
- Compose for MySQL**: MySQL is probably the most popular open source relational. (IBM, Beta)
- Compose for PostgreSQL**: Postgres is a powerful, open source object-relational database that is. (IBM)
- Compose for RabbitMQ**: RabbitMQ asynchronously handles the messages between your. (IBM)
- Compose for Redis**: Redis is an open-source, blazingly fast, key/value low maintenance. (IBM)
- Compose for RethinkDB**: RethinkDB is a JSON document based, distributed database with. (IBM)

At the bottom center of the page is the copyright notice: © Copyright IBM Corporation 2017.

Figure 6-2 Data services in IBM Cloud catalog

Notes:

IBM Cloud offers the following database management systems that are supported by IBM and third parties to use with cloud computing:

- ▶ NoSQL database services, including, Cloudant NoSQL DB and Compose for MongoDB.
- ▶ SQL database services. including IBM Db2® hosted, Compose for PostgreSQL, ElephantSQL, and ClearDB Managed MySQL Database.
- ▶ Columnar database services, including Compose for ScyllaDB.
- ▶ Key-value pair in-memory data services, including Compose for Redis, which is an open-source, fast key/value, low-maintenance store.

6.5 Cloudant capabilities



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) that is optimized for handling heavy workloads of concurrent reads and writes in the cloud. These workloads are typical for large, fast-growing web and mobile apps. It is built to scale globally, run continuously, and handle various data types, such as JSON, full-text, and geospatial.

Cloudant is designed to ensure that the flow of data between an application and its database remains uninterrupted and performs to the users' satisfaction. The data replication technology also allows developers to put data closer to where their applications need it most.

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

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

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

- ▶ Cloudant's search is powered by Apache Lucene, which is 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 applications.

- ▶ 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, which allows them to work in disconnected mode. Offline First applications that are built with Cloudant Sync provide a better and faster offline and online user experience by storing and accessing data locally and then synchronizing this data with the cloud database when an internet connection is available.

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

For more information, see the following website:

<https://www.ibm.com/cloud/cloudant>

6.6 Documents in Cloudant

The screenshot shows a presentation slide with a red header bar. On the left of the header is the Redbooks logo with the text 'Powered by Redbooks'. On the right is the IBM logo. The main content area has a white background with a title and a bulleted list.

Documents in Cloudant

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

Figure 6-4 Documents in Cloudant

Notes:

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

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

6.7 Getting started with Cloudant on IBM Cloud

The screenshot shows a web page titled "Getting started with Cloudant on IBM Cloud". At the top left is the "Powered by Redbooks" logo, and at the top right is the "IBM" logo. The main heading is "Getting started with Cloudant on IBM Cloud". Below it, a text block says: "The following starter boilerplate applications on IBM Cloud use Cloudant:" followed by a bulleted list of three items. Each item has a small icon, a name, and a brief description. At the bottom of the page is a copyright notice: "© Copyright IBM Corporation 2017".

Getting started with Cloudant on IBM Cloud

The following starter boilerplate applications on IBM Cloud use Cloudant:

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

ASP.NET Core Cloudant Starter
Use the Cloudant NoSQL DB Service in an IBM application.

Java Cloudant Web Starter
Use the Cloudant NoSQL DB service with the Java application.

Node.js Cloudant DB Web Starter
Use the Cloudant NoSQL DB service with the Node.js application.

© Copyright IBM Corporation 2017

Figure 6-5 Getting started with Cloudant on IBM Cloud

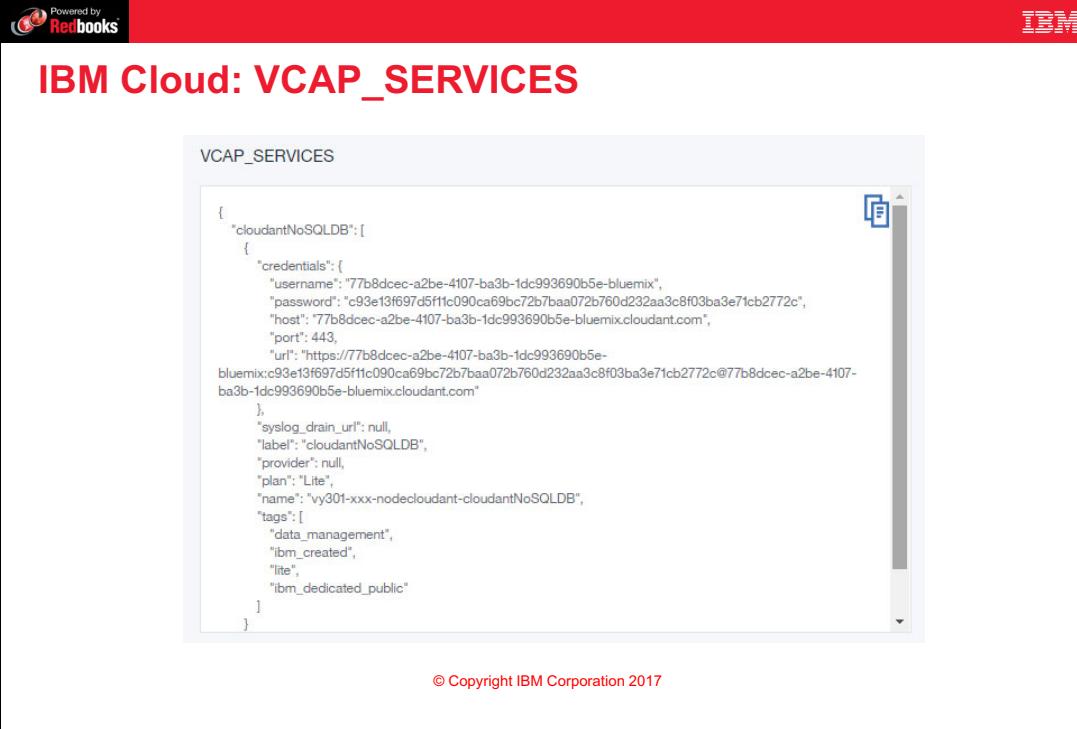
Notes:

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

For more information about Cloudant documentation, see the following website:

<https://console.bluemix.net/docs/services/Cloudant/getting-started.html>

6.8 IBM Cloud: VCAP_SERVICES



The screenshot shows the IBM Cloud VCAP_SERVICES interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "IBM Cloud: VCAP_SERVICES" is displayed in red. Underneath the title, the heading "VCAP_SERVICES" is shown in a light blue box. A large JSON object is displayed in a code editor-like window. The JSON object represents a Cloudant database service with the following structure:

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

At the bottom of the JSON window, there's a small copyright notice: "© Copyright IBM Corporation 2017".

Figure 6-6 IBM Cloud: VCAP_SERVICES

Notes:

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

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

6.9 Cloudant Dashboard

The screenshot shows the Cloudant Dashboard interface. At the top, there's a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header is a title bar with the text "Cloudant Dashboard". On the left side, there's a vertical navigation menu with the following items: Monitoring (selected), Databases (highlighted in orange), Replication, Analytics, Active Tasks, Account, Support, Documentation, and IBM Cloudant. The main content area is titled "Databases" and contains a table with one row. The table has columns for Name, Size, # of Docs, and Actions. The single database listed is "my_sample_db", which is 1.2 MB in size and contains 4 documents. The "Actions" column for this database includes icons for replication, locking, and deleting. At the bottom of the main content area, it says "Showing 1–1 of 1 databases." and has a page number indicator "1".

Figure 6-7 Cloudant Dashboard

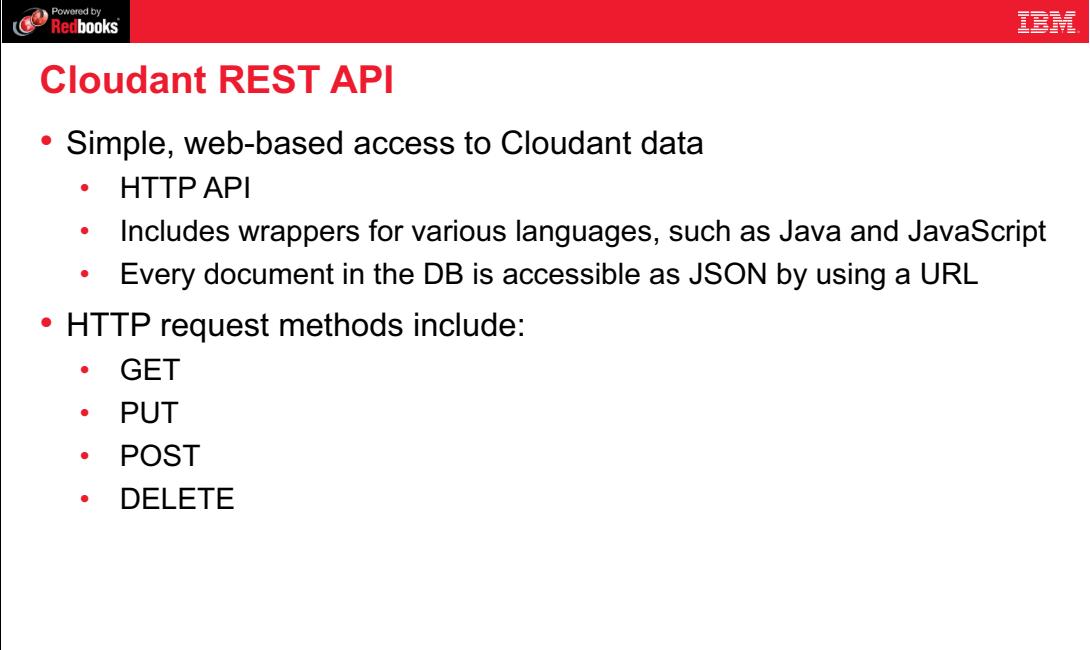
Notes:

Cloudant Dashboard is a cloud-based web interface that makes it easy to develop, administer, and monitor your databases. You can perform many tasks, such as:

- ▶ View and manage Cloudant databases
- ▶ View and create documents
- ▶ Create and run queries
- ▶ Manage the permissions to the database

You can also display the contents of a Cloudant document in IBM Cloud 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.

6.10 Cloudant REST API



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "Cloudant REST API" is in bold red text. Below it is a bulleted list of features and methods.

Cloudant REST API

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

Figure 6-8 Cloudant REST API

Notes:

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

The following HTTP Request methods can be used:

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

6.11 Sample database at Cloudant

The screenshot shows a Cloudant dashboard titled "Sample database at Cloudant". It displays the permissions assigned to the database "my_sample_db". The permissions table has four columns: "_admin", "_reader", "_writer", and "_replicator". Under the "_admin" column, the row for "77b8dcec-a2be-4107-ba3b-1dc993690b5e-bluemix" has all checkboxes checked. Under the "_reader" column, the same row has all checkboxes checked. Under the "_writer" column, the same row has all checkboxes checked. Under the "_replicator" column, the same row has all checkboxes checked. The row for "Unauthenticated connections" has all checkboxes unchecked. At the bottom, there is a "Grant database permissions to:" input field containing "Username or API Key" and a "Grant Permissions" button.

Figure 6-9 Sample database at Cloudant

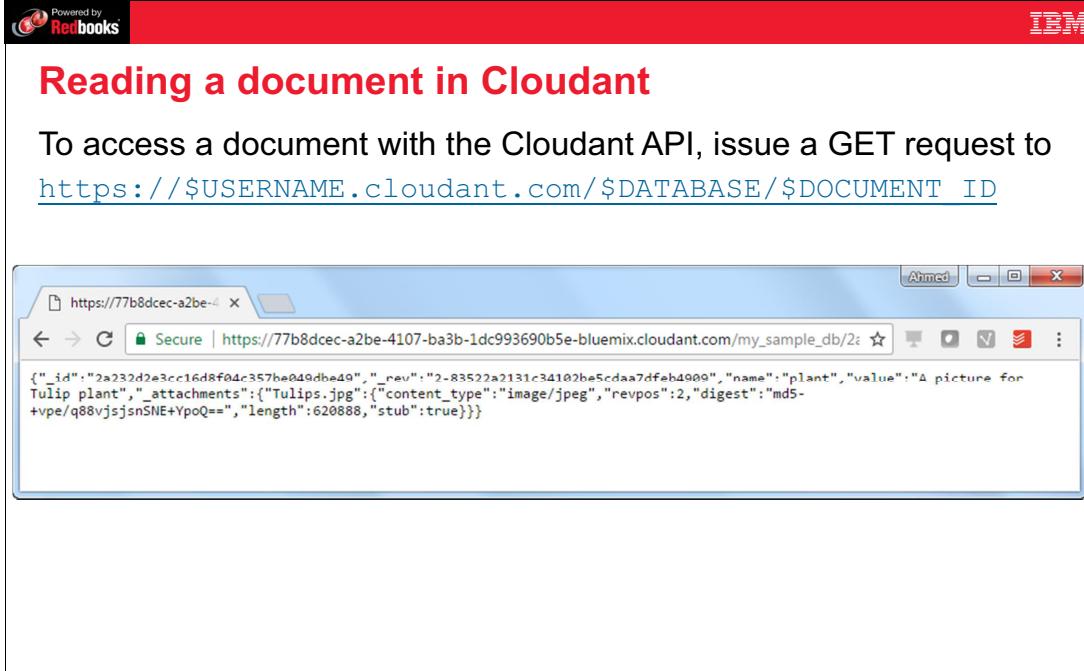
Notes:

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

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

The permissions that are assigned to the creator of the database as seen in the Cloudant Dashboard UI are shown in Figure 6-9.

6.12 Reading a document in Cloudant



The screenshot shows a web browser window with the title "Reading a document in Cloudant". The URL in the address bar is `https://77b8dcec-a2be-4107-ba3b-1dc993690b5e-bluemix.cloudant.com/my_sample_db/_id`. The page content displays a JSON document:

```
{"_id": "2a232d2e3c16d8f04c357ba049d8e49", "_rev": "2-83522a2131c34102be5cdcaa7dfcb4909", "name": "plant", "value": "A picture for Tulip plant", "_attachments": {"Tulips.jpg": {"content_type": "image/jpeg", "revpos": 2, "digest": "md5-vpe/q88vjsjsnSNE+YpoQ==", "length": 620888, "stub": true}}}
```

Figure 6-10 Reading a document in Cloudant

Notes:

This figure in this slide shows accessing a document from the browser. 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 that is used to access this document in the Cloudant sample database is `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID`, which is accessed with a GET HTTP REST Request if you have sufficient permissions to access the DB. The `_id` is a unique key that is used when reading a document in the Cloudant database.

The sample document in the Cloudant Dashboard that is shown in Figure 6-10 includes three fields (name, value, and `_attachments`) in addition to the mandatory fields `_id` and `_rev`.

6.13 View all documents

The screenshot shows the IBM Cloudant Dashboard interface. At the top, there's a red header bar with the Redbooks logo and the IBM logo. Below the header, the title "View all documents" is displayed in red. A sub-instruction follows: "To view all documents at a database, issue a GET request to: [https://\\$USERNAME.cloudant.com/\\$DATABASE/_all_docs?include_docs=true](https://$USERNAME.cloudant.com/$DATABASE/_all_docs?include_docs=true)". The main content area is titled "Your Databases". It contains a table with one row, showing a database named "my_sample_db" with a size of "0.9 KB" and "1" document. The table has columns for "Name", "Size", "# of Docs", and "Actions". The "Actions" column includes icons for edit, lock, and delete. At the bottom of the table, it says "Showing 1–1 of 1 databases." and has navigation arrows. On the left side, there's a sidebar with various icons and a "Database..." button. The bottom right corner of the dashboard has a copyright notice: "© Copyright IBM Corporation 2017".

Figure 6-11 View all Documents

Notes:

Cloudant includes an index that is named `_all_docs` with which you can build a URL to list all the documents in the database. You can pass to it an optional parameter that is named `include_docs` to return the contents of the documents, not just the `_id` and `_rev`. To change this parameter from the Cloudant Dashboard, click **Options** on the top toolbar then, click **Include Docs**.

As shown in Figure 6-11, you can click the `{ JSON` link at any stage in Cloudant Dashboard to view the generated REST API for the page that the user is viewing in the Cloudant Dashboard.

6.14 More Cloudant REST APIs



More Cloudant REST APIs

- Create a document:
POST [https://\\$USERNAME.cloudant.com/\\$DATABASE](https://$USERNAME.cloudant.com/$DATABASE)
with the document's JSON content in the body.
- Update a document:
PUT [https://\\$USERNAME.cloudant.com/\\$DATABASE/\\$DOCUMENT_ID](https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID)
with the updated document JSON content, including latest_rev in the body.
- Delete a document:
DELETE [https://\\$USERNAME.cloudant.com/\\$DATABASE/\\$DOCUMENT_ID?rev=\\$REV](https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID?rev=$REV)

© Copyright IBM Corporation 2017

Figure 6-12 More Cloudant REST APIs

Notes:

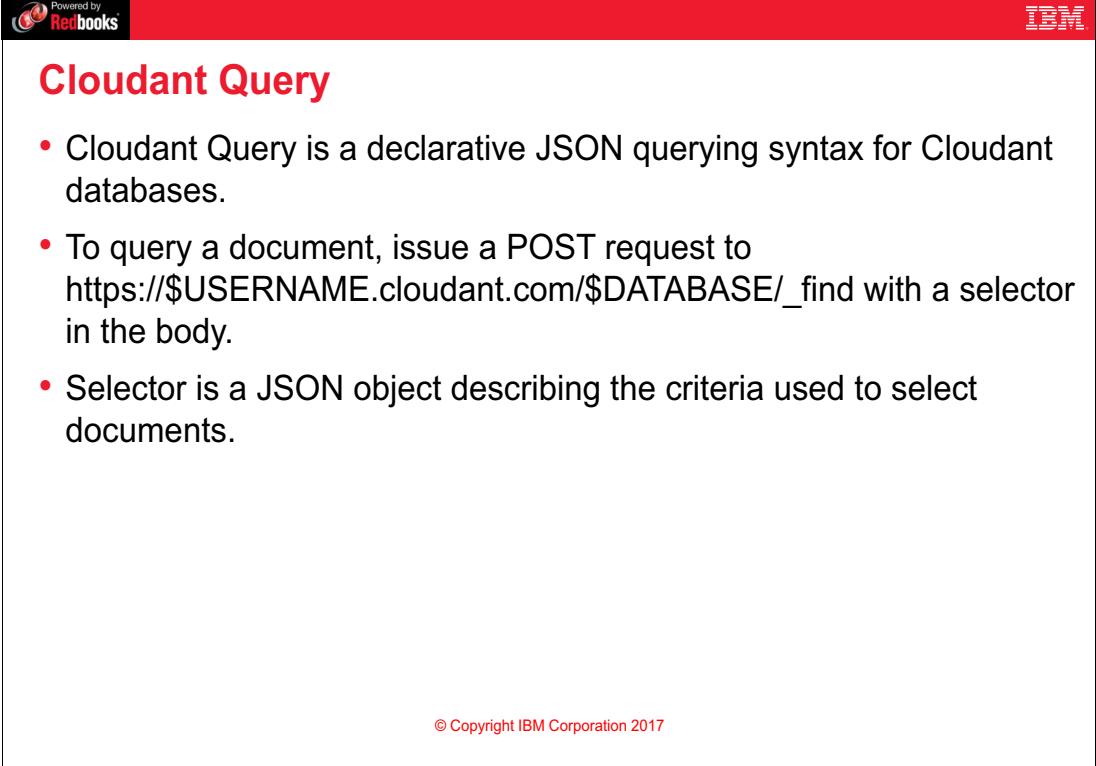
To create a document, send a POST request to [https://\\$USERNAME.cloudant.com/\\$DATABASE](https://$USERNAME.cloudant.com/$DATABASE) with the document's JSON content in the request body.

To update (or create) a document, send a PUT request to [https://\\$USERNAME.cloudant.com/\\$DATABASE/\\$DOCUMENT_ID](https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID) with the updated JSON content, including the latest_rev value in the request body.

To delete a document, send a DELETE request to [https://\\$USERNAME.cloudant.com/\\$DATABASE/\\$DOCUMENT_ID?rev=\\$REV](https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID?rev=$REV) where \$REV is the document's latest_rev.

Note: All the previous operations on the database (Create, Update and Delete) can be done by using POST and PUT methods, with different parameters sent.

6.15 Cloudant Query



Powered by Redbooks

Cloudant Query

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

© Copyright IBM Corporation 2017

Figure 6-13 Cloudant query

Notes:

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

6.16 HTTP status codes



Powered by
Redbooks

IBM

HTTP status codes

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

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

© Copyright IBM Corporation 2017

Figure 6-14 HTTP status codes

Notes:

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

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

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

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

For example, if you try to use `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID` to retrieve a document that does not exist in the database, Cloudant responds with status code 404 in the header, and more information about the error is returned in the response as JSON, as shown in Figure 6-14.

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

6.17 Unit summary

Powered by
Redbooks

IBM

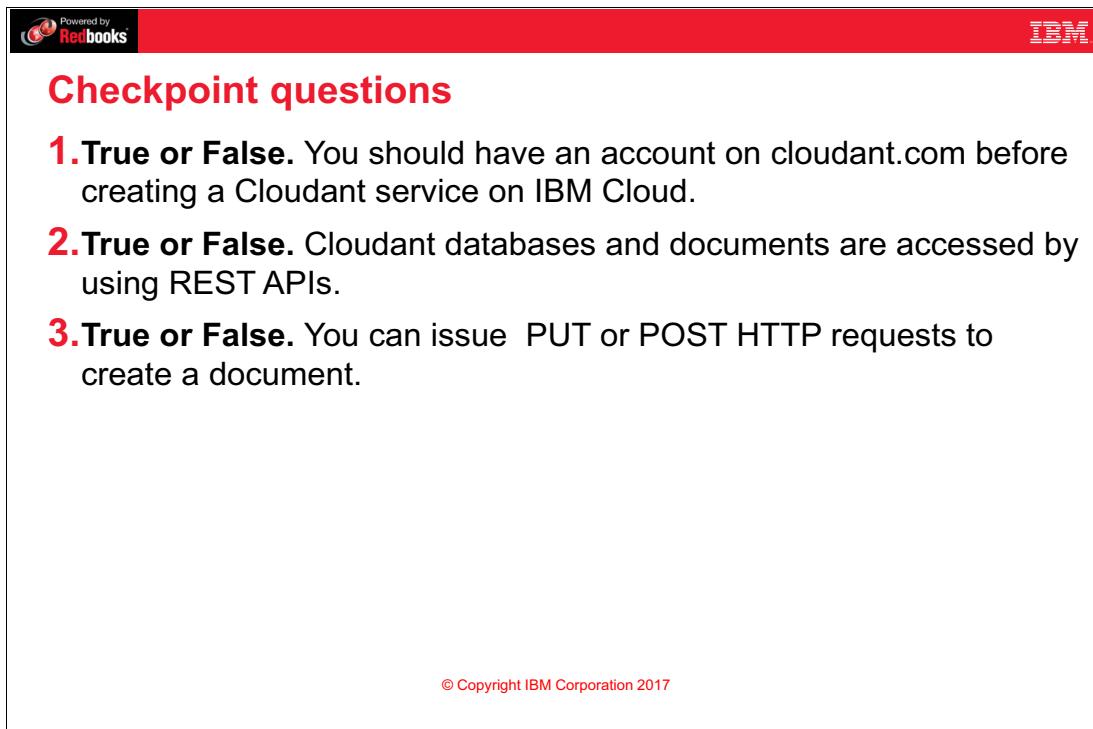
Unit summary

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

© Copyright IBM Corporation 2017

Figure 6-15 Unit summary

6.18 Checkpoint questions



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top, the title "Checkpoint questions" is displayed in red. Below the title is a numbered list of three true/false questions. At the bottom right of the slide, there is a small copyright notice: "© Copyright IBM Corporation 2017".

Checkpoint questions

- 1. True or False.** You should have an account on cloudant.com before creating a Cloudant service on IBM Cloud.
- 2. True or False.** Cloudant databases and documents are accessed by using REST APIs.
- 3. True or False.** You can issue PUT or POST HTTP requests to create a document.

© Copyright IBM Corporation 2017

Figure 6-16 Checkpoint questions

6.19 Checkpoint answers

The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top left of this area is the title "Checkpoint answers" in bold red font. Below it is a numbered list of three items, each starting with a red dot and bolded text. At the bottom center of the slide is a small line of text: "© Copyright IBM Corporation 2017".

Checkpoint answers

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

© Copyright IBM Corporation 2017

Figure 6-17 Checkpoint answers



IBM Cloud 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
- ▶ App ID service
- ▶ Kinetise service
- ▶ MobileFirst Services Starter Boilerplate

7.1 What you should be able to do

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

- ▶ Mobile Backend as a Service (MBaaS)
- ▶ MBaaS architecture
- ▶ Push Notifications service
- ▶ App ID service
- ▶ Kinetise service
- ▶ How to create 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 IBM Cloud:
<https://ibm.biz/BdjUUE>
- ▶ App ID:
<https://ibm.biz/BdjxpN>
- ▶ Push Notifications:
<https://console.bluemix.net/docs/services/mobilepush/index.html>
- ▶ Kinetise:
 - <https://helpcenter.kinetise.com/>
 - https://bluemix.kinetise.com/bluemix_tutorial.html

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



What is mobile backend as a service (MBaaS)?

Use the complete collection of backend services to build and scale your applications faster:

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

© Copyright IBM Corporation 2017

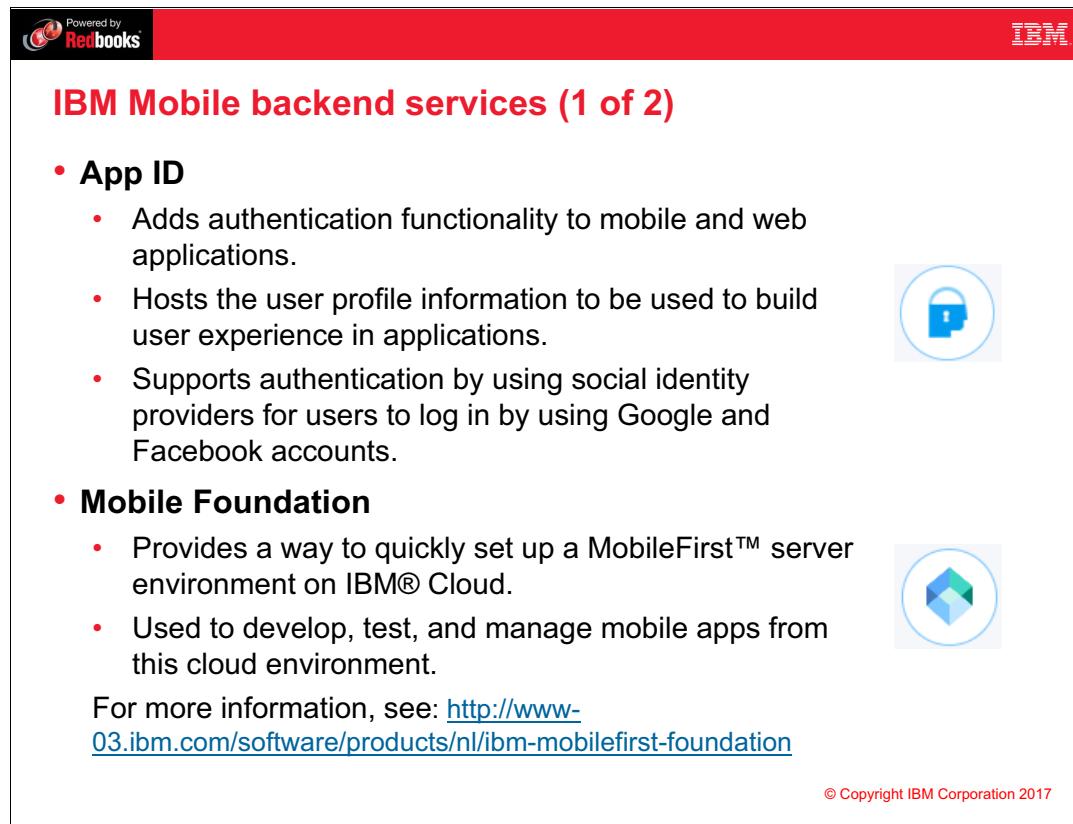
Figure 7-1 What is mobile backend as a service (MBaaS)?

Notes:

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

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

7.4 IBM Mobile backend services (1 of 2)



The slide is titled "IBM Mobile backend services (1 of 2)" in red. It features two circular icons: one with a blue padlock and another with a blue diamond shape. A copyright notice at the bottom right reads "© Copyright IBM Corporation 2017".

- **App ID**
 - Adds authentication functionality to mobile and web applications.
 - Hosts the user profile information to be used to build user experience in applications.
 - Supports authentication by using social identity providers for users to log in by using Google and Facebook accounts.
- **Mobile Foundation**
 - Provides a way to quickly set up a MobileFirst™ server environment on IBM® Cloud.
 - Used to develop, test, and manage mobile apps from this cloud environment.

For more information, see: <http://www-03.ibm.com/software/products/nl/ibm-mobilefirst-foundation>

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

Notes:

The following IBM mobile services are provided on IBM Cloud:

- ▶ App ID
 - Use the IBM Cloud App ID service to add authentication to your mobile and web application. App ID supports authentication by using social identity providers so that users can log in with their Facebook and Google accounts. You can also store user profile information that you can use to build engaging experiences.
- ▶ Mobile Foundation
 - The Mobile Foundation service provides a way to quickly set up a MobileFirst server environment on IBM Cloud. You can then develop, test, and manage mobile apps from this cloud environment. Mobile Foundation is available under two different service plans: Developer and Professional 1 Application.

7.5 IBM Mobile backend services (2 of 2)

IBM Mobile backend services (2 of 2)

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

© Copyright IBM Corporation 2017

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

Notes:

► Mobile Analytics

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

► Push Notifications

Use the IBM Push Notifications for IBM Cloud 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.

► Kinetise

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

7.6 MBaaS architecture

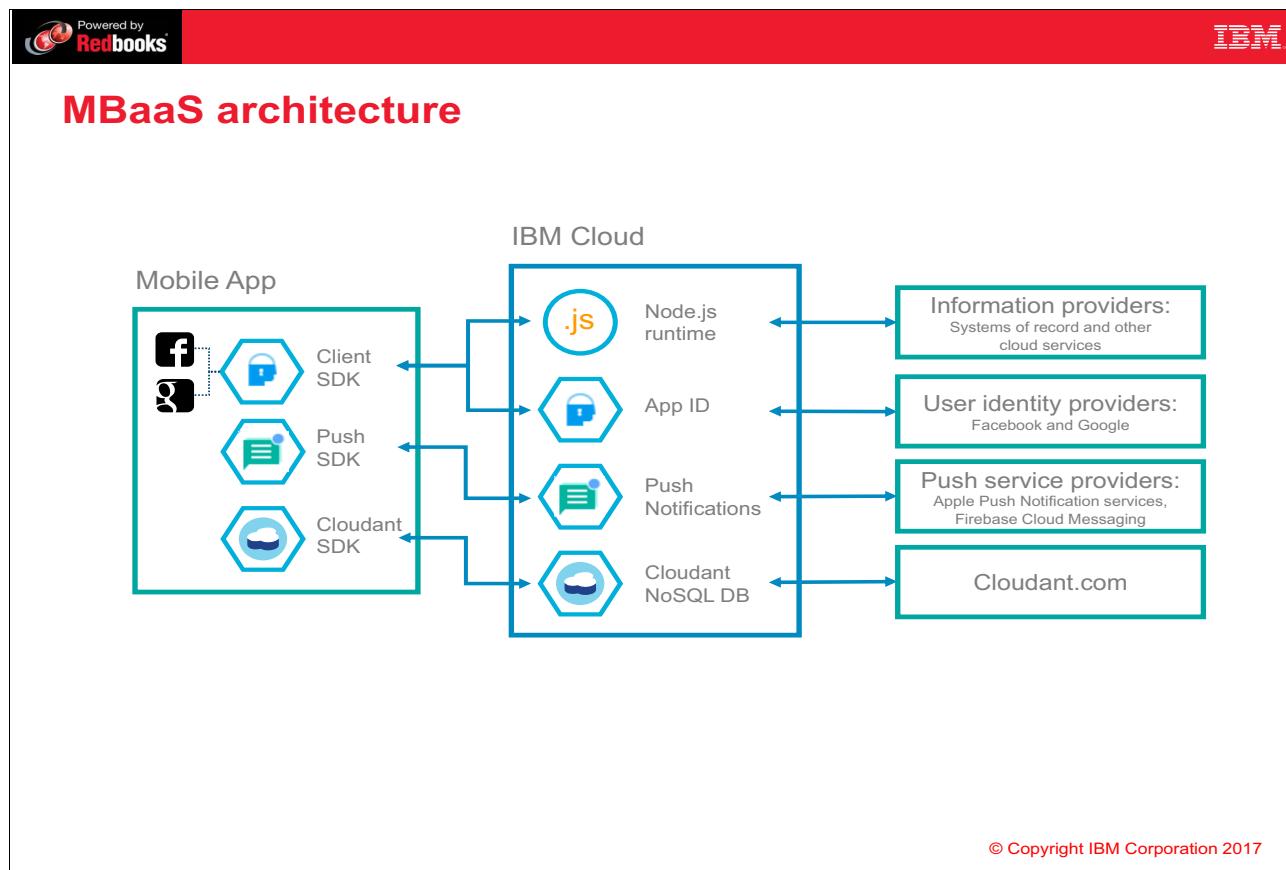


Figure 7-4 MBaaS architecture

Notes:

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

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

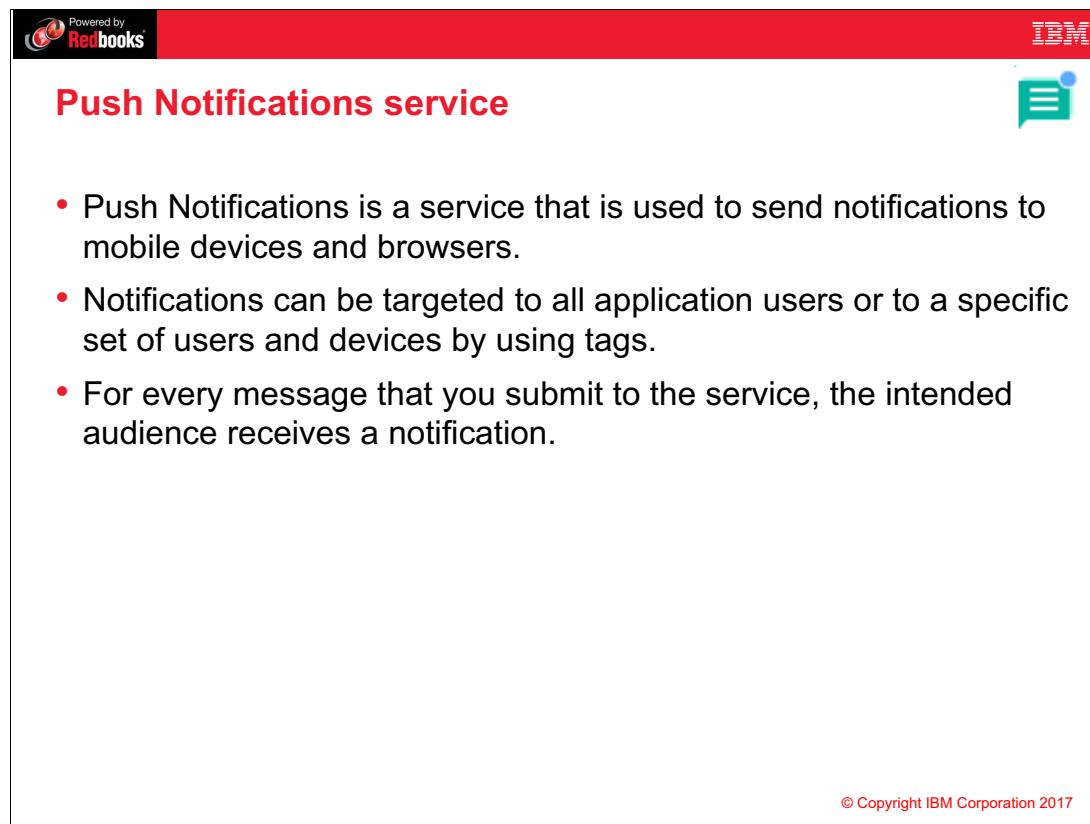
- ▶ Client SDK is used to interact with the server-side components (Node.js runtime and App ID) to provide clients with Google and Facebook authentication.
- ▶ Push SDK to interact with the device-specific push service providers.
- ▶ Cloudant SDK to implement data persistence with the Cloudant NoSQL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

IBM Cloud component features the following mobile back-end services that are attached to Node.js runtime:

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

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

7.7 Push Notifications service



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main title "Push Notifications service" is in bold red text at the top left. To the right of the title is a green speech bubble icon with three white dots. Below the title is a bulleted list of four points. At the bottom right of the slide is the copyright notice "© Copyright IBM Corporation 2017".

- Push Notifications is a service that is used to send notifications to mobile devices and browsers.
- Notifications can be targeted to all application users or to a specific set of users and devices by using tags.
- For every message that you submit to the service, the intended audience receives a notification.

Figure 7-5 Push Notifications service

Notes:

IBM Push Notifications is a service that you can use to send notifications to mobile devices and browsers.

Notifications can be targeted to all application users or to a specific set of users and devices by using tags. For every message that you submit to the service, the intended audience receives a notification.

7.8 Push Notifications process

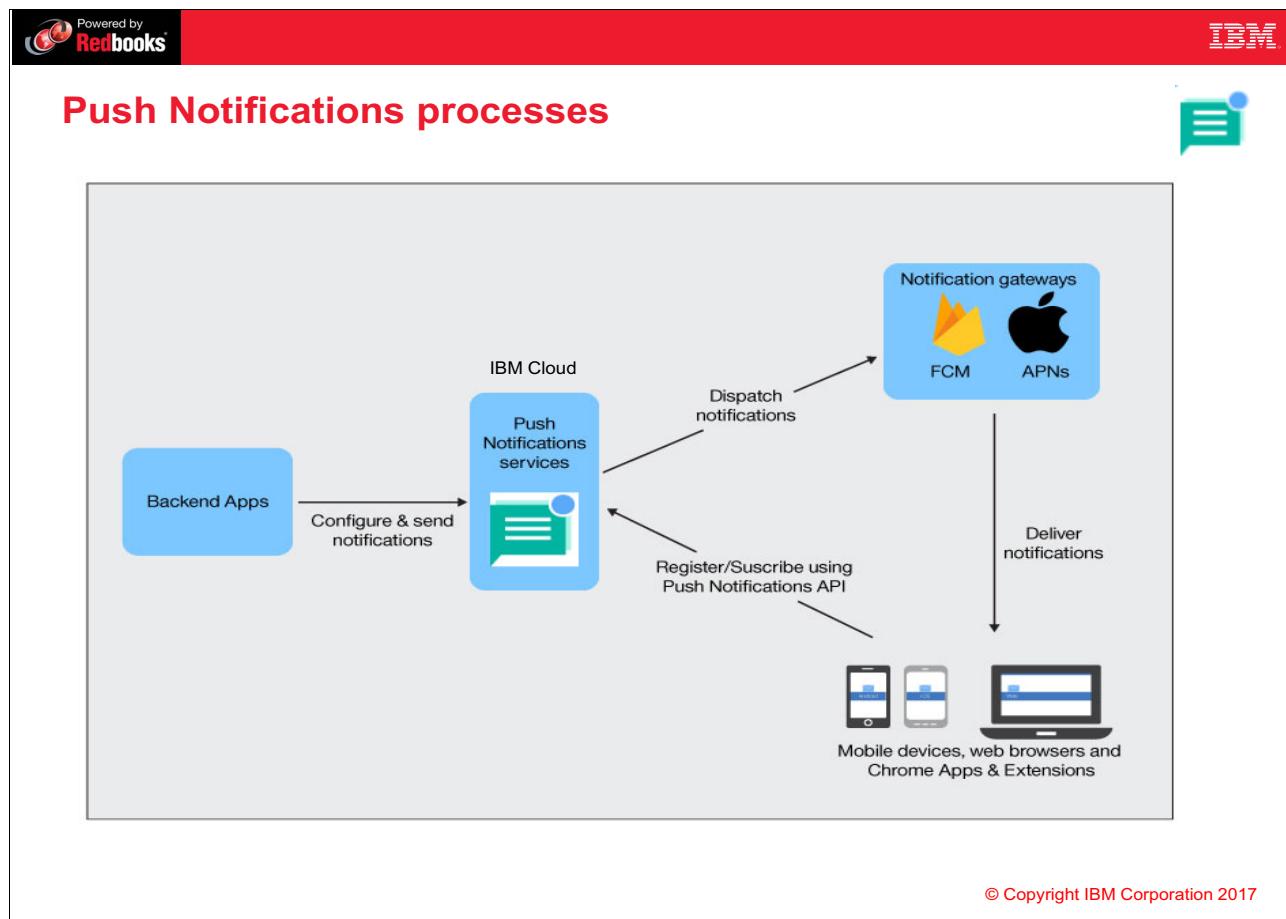


Figure 7-6 Push Notifications process

Notes:

The Push Notifications processes to register and receive notifications on mobile devices or web browsers is shown in Figure 7-6.

The following components in the processes also are shown in Figure 7-6:

- ▶ Mobile, browser applications and Chrome Apps & Extensions
 - On start-up, client applications register and subscribe themselves to the Push Notifications service to receive notifications.
- ▶ Backend Apps
 - Backend applications (on premises or on cloud) use the Push Notifications service to send notifications to mobile, browser application, Chrome apps, and extensions users.
- ▶ Push Notifications services
 - The Push Notifications service manages all information that is related to mobile devices and web browser clients that are registered for notifications. The service shields your applications from the technology details of sending notifications to heterogeneous mobile and web browser platforms.

- ▶ Notifications gateways

Platform-specific Push Notifications cloud services, such as Firebase Cloud Messaging (FCM) or Apple Push Notification Service (APNs), that are used by IBM Push Notifications service to dispatch notifications to the mobile and browser applications.

The end-to-end flow of push notifications features the following steps:

1. Push Notifications are configured in IBM Cloud. Here, the credentials from the notification gateway (FCM, or APN) are used.
2. The mobile application is created by using IBM Cloud or outside IBM Cloud (Android studio or Xcode).
3. The mobile application calls the IBM Cloud back-end application.
4. The IBM Cloud application uses the IBM Push API to send and dispatch notifications to the push notification gateway.
5. The notification gateway sends notifications to the mobile devices.
6. Notifications are sent to your application through both providers from a single interface by using the IBM Cloud Push Notification service.

7.9 Configuring push notifications in IBM Cloud (1 of 2)

The screenshot shows a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "Configuring push notifications in IBM Cloud (1 of 2)" is displayed in red. To the right of the title is a blue icon representing messaging or notifications. The main content area contains three bullet-point sections: "For iOS Apps", "For Android Apps and Chrome Apps & Extensions", and "For Chrome browsers". Each section has a corresponding list of items to enter.

- **For iOS Apps:**
 - Environment: Sandbox (development) or Production (distribution)
 - Upload .p12 certificate
 - Enter password associated with .p12 certificate
- **For Android Apps and Chrome Apps & Extensions:**
 - Sender ID/Project Number
 - API Key
- **For Chrome browsers:**
 - Website URL
 - Google Server API Key

© Copyright IBM Corporation 2017

Figure 7-7 Configure push notifications in IBM Cloud (1 of 2)

Notes:

After the application owner registers to use the Google/Firebase Cloud Messaging or Apple Push Notifications APIs, you can use the credentials that are provided by the push notification provider to configure push notifications for mobile devices or web browsers in IBM Cloud:

- ▶ iOS Apps
Select the Environment, Sandbox (development), or Production (distribution), upload the.p12 certificate that you created, and enter password that is associated with the .p12 certificate.
- ▶ Android and Chrome Apps & Extensions
Enter the Sender ID/Project Number and the API Key.
- ▶ Chrome browsers
Enter the FCM API key and the URL of your website that is registered to receive push notifications.

7.10 Configuring push notifications in IBM Cloud (2 of 2)

The screenshot shows a red header bar with the Redbooks logo on the left and the IBM logo on the right. Below the header, the title "Configuring push notifications in IBM Cloud (2 of 2)" is displayed in red. The main content area contains two bulleted lists under the heading "Configuring push notifications in IBM Cloud (2 of 2)".

- **For Safari browsers:**
 - Website Name
 - Website Push ID
 - Website URL
 - Allowed domains
 - URL format string
 - Notification icons
 - Upload .p12 certificate
 - Password associated with .p12 certificate.
- **For Firefox browsers:**
 - Website URL

© Copyright IBM Corporation 2017

Figure 7-8 Configure push notifications in IBM Cloud (2 of 2)

Notes:

- ▶ Safari browsers: Enter the following information:
 - Website name in the APNs configuration
 - Website URL that is registered to receive push notifications
 - Website push ID, which is the reverse-domain string for your Website Push
 - Allowed domains (optional)
 - URL format string to resolve when the user clicks the notification
 - Notification icons to be associated with the notification alert
 - The .p12 certificate and password that is associated with it
- ▶ Firefox browsers: Enter the website URL that will be registered to receive push notifications.

7.11 Sending manual notifications from IBM Cloud

The screenshot shows the 'Send Notifications' interface. At the top, there's a red header bar with the Redbooks logo and the IBM logo. The main title 'Sending manual notifications from IBM Cloud' is centered above a form. The form has two main sections: 'Compose' on the left and 'Optional Settings' on the right. In the 'Compose' section, there's a dropdown for 'Send To' set to 'All Devices'. Below it is a text area for 'Message' with placeholder text 'Type your message here'. In the 'Optional Settings' section, there are two tabs: 'Mobile' (which is selected) and 'Web'. Under 'Mobile', there are two options: 'iOS' and 'Android'. At the bottom of the form is a 'Send' button. On the right side of the interface, there's a small green icon of a speech bubble with a dot. At the very bottom right, there's a copyright notice: '© Copyright IBM Corporation 2017'.

Figure 7-9 Sending manual notifications from IBM Cloud

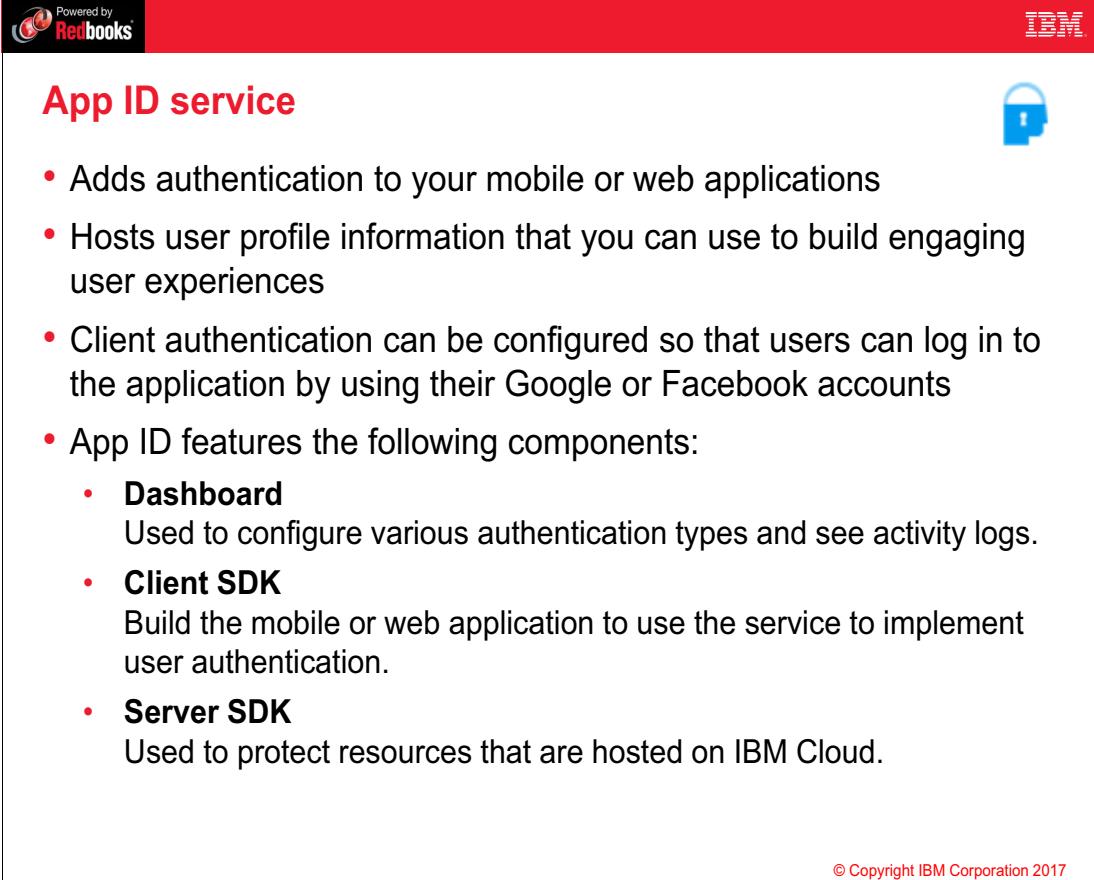
Notes:

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

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

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

7.12 App ID service



The image shows a screenshot of the IBM App ID service landing page. At the top left is the Redbooks logo with the text "Powered by Redbooks". At the top right is the IBM logo. The main title "App ID service" is centered above a bulleted list of features. To the right of the list is a blue padlock icon. At the bottom right of the page is a small copyright notice: "© Copyright IBM Corporation 2017".

App ID service

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

© Copyright IBM Corporation 2017

Figure 7-10 App ID service

Notes:

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

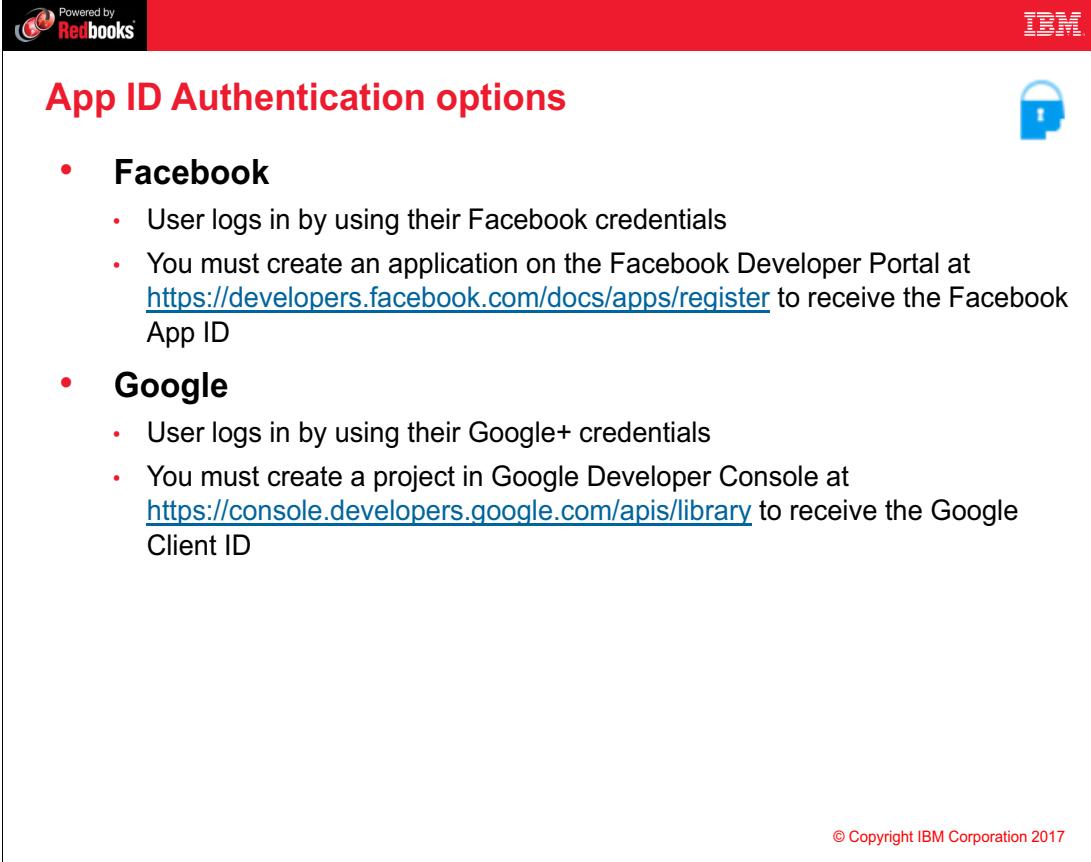
You can also host user profile information that you can use to build engaging user experiences.

App ID features the following components:

- ▶ **Dashboard**
Download on boarding samples, see activity logs, configure various authentication types and identity providers.
- ▶ **Client SDK**
Build mobile and web applications that use the service to implement user authentication. The following platforms are supported:
 - iOS 9+
 - MacOS 10.11.5
 - Xcode 8.2

- Android API 25 or higher
 - Java 8.x
 - Android SDK tools 25.2.5 or higher
 - Android SDK Platform Tools 25.0.3 or higher
 - Android Build Tools version 25.0.2
- Server SDK
- Protect resources that are hosted on IBM Cloud. At the time of this writing, Node.js and Swift runtimes also are supported.

7.13 App ID Authentication options



The image shows a web page titled "App ID Authentication options". The page is part of the "Powered by Redbooks" series, indicated by a logo in the top left corner. The top right corner features the IBM logo. The main content area contains two bullet points: one for Facebook and one for Google. Each point has a list of steps. A blue padlock icon is located in the top right corner of the content area. At the bottom right of the page, there is a copyright notice: "© Copyright IBM Corporation 2017".

App ID Authentication options

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

© Copyright IBM Corporation 2017

Figure 7-11 App ID Authentication options

Notes:

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

► Facebook

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

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

► Google

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

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

7.14 App ID architecture

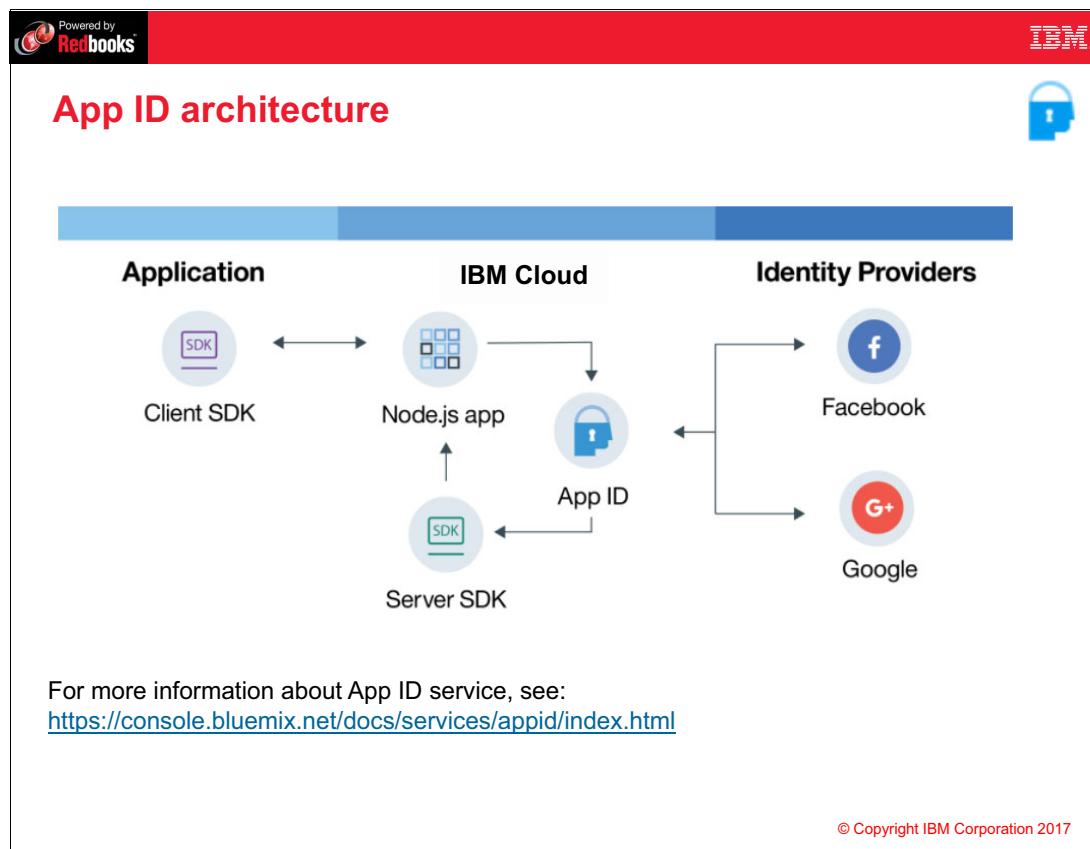


Figure 7-12 App ID architecture

Notes:

How the different components of App ID interact with each other is shown in Figure 7-12. Consider the following points:

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

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

7.15 What is Kinetise?

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

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

Figure 7-13 What is Kinetise?

Notes:

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

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

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

7.16 How to use Kinetise

The screenshot shows a presentation slide with a red header bar. On the left of the header is the Redbooks logo, and on the right is the IBM logo. The main title 'How to use Kinetise' is centered at the top in red text. To the right of the title is a yellow square containing a black stylized letter 'k'. Below the title is a bulleted list of six items describing Kinetise's capabilities. At the bottom right of the slide is a small copyright notice: '© Copyright IBM Corporation 2017'.

How to use Kinetise

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

© Copyright IBM Corporation 2017

Figure 7-14 How to use Kinetise

Notes:

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

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

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

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

<https://helpcenter.kinetise.com/>

7.17 MobileFirst Services Starter Boilerplate

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

- SDK for Node.js runtime
- Mobile Analytics
- Push Notifications service
- Cloudant NoSQL DB service

In the center of the page is a small icon of a smartphone displaying a mobile interface. At the bottom right of the page is a copyright notice: "© Copyright IBM Corporation 2017".

Figure 7-15 MobileFirst Services Starter Boilerplate

Notes:

You can use all IBM Cloud Mobile services together by using MobileFirst Services Starter boilerplate, or you can use individual services from the IBM Cloud catalog. You can use individual services if you want a different type of runtime for the back-end application rather than Node.js, or if you do not want to use all the services that are included with the Boilerplate.

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

The template includes the following features:

- ▶ SDK for Node.js runtime
- ▶ Mobile Analytics service
- ▶ Cloudant NoSQL DB service
- ▶ Push Notifications service

7.18 Unit summary

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

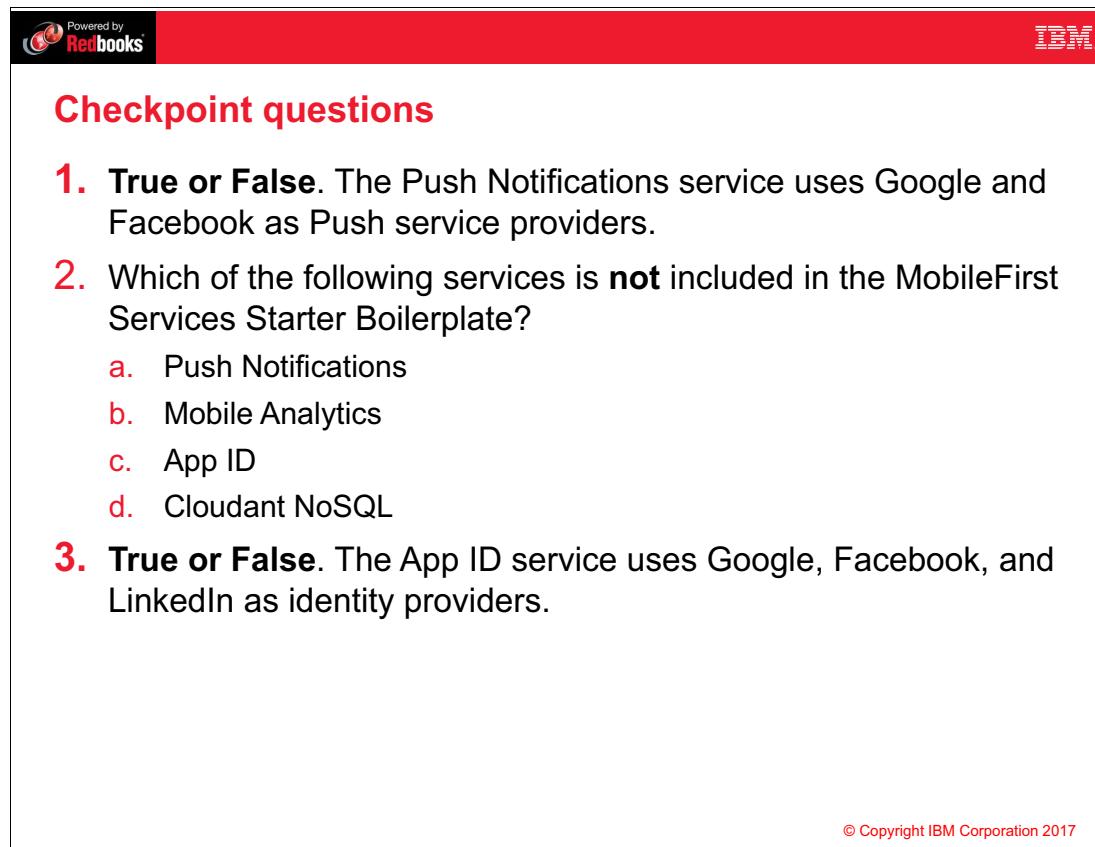
Unit summary

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

© Copyright IBM Corporation 2017

Figure 7-16 Unit summary

7.19 Checkpoint questions



The slide has a red header bar. On the left is the Redbooks logo with the text "Powered by Redbooks". On the right is the IBM logo. The main content area has a white background. At the top left of the content area is the title "Checkpoint questions" in red. Below it is a numbered list of three questions. Question 1 asks about Push Notifications providers. Question 2 asks which service is not included in the MobileFirst Services Starter Boilerplate, with options a, b, c, and d. Question 3 asks about the App ID service identity providers. At the bottom right of the content area is the copyright notice "© Copyright IBM Corporation 2017".

Checkpoint questions

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

© Copyright IBM Corporation 2017

Figure 7-17 Checkpoint questions

7.20 Checkpoint answers



Powered by
Redbooks

IBM

Checkpoint answers

- 1. False.** The Push Notifications service uses Apple and Google as Push Notifications Providers.
- 2. c.** App ID is not included in the MobileFirst Services Starter Boilerplate.
- 3. False.** The App ID uses only Google and Facebook as identity providers.

© Copyright IBM Corporation 2017

Figure 7-18 Checkpoint answers

Redbooks

Essentials of Application Development on IBM Cloud

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



SG24-8374-02

ISBN 0738442860

Printed in U.S.A.

Get connected

