Course Guide

Essentials of Cloud Application Development

IBM MEA Skills Academy
Course code SACAD ERC 3.0

Ahmed Azraq

Surbhi Agarwal

Hala Aziz

Sukumar Beri

Deepak Chandrashekar

Mihai Criveti

Karim El-Bawab

Haytham Elkhoja

Vikash Gupta

Norhan Khaled

Marcelo Mota Manhaes

Cong Nguyen

JungHyeon Yoo

Acknowledgments to members of

IBM Academy of Technology

IBM Training

July 2019 edition

Notices

This information was developed for products and services offered in the US.

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
United States of America

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.

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

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

Trademarks

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

Contents

Trademarks . . . . **xiii**

Course description . . . . **xiv**

Agenda . . . . **xvi**

Unit 1. Introduction to cloud computing **1-1**
Unit objectives 1-2
1.1. Introduction to cloud computing 1-3
Introduction to cloud computing 1-4
Topics 1-5
Before cloud computing 1-6
Challenges faced before cloud computing 1-7
What is cloud computing 1-9
Characteristics of the cloud 1-10
Benefits of the cloud 1-12
Factors contributing to the growth of the cloud 1-14
1.2. Cloud service models 1-15
Cloud service models 1-16
Topics 1-17
The pizza analogy 1-18
Cloud service models 1-20
Infrastructure as a Service 1-21
Platform as a Service 1-22
Software as a Service 1-24
Example of cloud services 1-25
Cloud provider and client responsibilities 1-26
Choices when building cloud applications 1-27
1.3. Cloud deployment models 1-28
Cloud deployment models 1-29
Topics 1-30
Cloud deployment models 1-31
Cloud-native applications 1-33
Cloud-native development methods 1-35
Cloud-native development methods (cont.) 1-36
Cloud-native development methods (cont.) 1-37
Unit summary 1-39
Review questions 1-40
Review questions (cont.) 1-41
Review answers 1-42
References 1-43

Unit 2. Getting started with IBM Cloud **2-1**
Unit objectives 2-2
2.1. Introduction to IBM Cloud 2-3
Introduction to IBM Cloud 2-4
Topics 2-5
Welcome to IBM Cloud 2-6
What is IBM Cloud 2-7
What is IBM Cloud (cont.) 2-9

What can you build with IBM Cloud . . . 2-10
2.2. IBM Cloud compute choices . . . 2-12
IBM Cloud compute choices . . . 2-13
Topics . . . 2-14
IBM Cloud service models . . . 2-15
Infrastructure as a service from IBM Cloud . . . 2-16
Platform as a service from IBM Cloud . . . 2-17
Platform as a service from IBM Cloud (cont.) . . . 2-18
IBM Cloud: Choices . . . 2-19
IBM Cloud: Choice of compute . . . 2-20
IBM Cloud: Choice of runtime . . . 2-22
IBM Cloud: Services . . . 2-23
2.3. IBM Cloud console . . . 2-25
IBM Cloud console . . . 2-26
Topics . . . 2-27
IBM Cloud: Signup . . . 2-28
IBM Cloud: Login . . . 2-30
IBM Cloud: Dashboard . . . 2-31
IBM Cloud: Docs . . . 2-32
IBM Cloud: Support . . . 2-33
2.4. IBM Cloud catalog . . . 2-34
IBM Cloud catalog . . . 2-35
Topics . . . 2-36
IBM Cloud: Catalog . . . 2-37
Working with resources . . . 2-38
IBM Cloud catalog: Categories . . . 2-39
IBM Cloud catalog: Starter kits . . . 2-40
IBM Cloud catalog: Starter kits (cont.) . . . 2-41
IBM Cloud catalog: Creating a starter kit . . . 2-43
IBM Cloud catalog: Cloud Foundry Apps . . . 2-44
IBM Cloud catalog: AI . . . 2-46
IBM Cloud catalog: Web and mobile . . . 2-47
IBM Cloud catalog: Developer tools . . . 2-48
IBM Cloud catalog: Functions . . . 2-50
2.5. Creating and managing a Cloud Foundry application. . . . 2-52
Creating and managing a Cloud Foundry application . . . 2-53
Topics . . . 2-54
Creating an IBM Cloud Foundry application (1 of 3) . . . 2-55
Creating an IBM Cloud Foundry application (2 of 3) . . . 2-56
Creating an IBM Cloud Foundry application (3 of 3) . . . 2-57
Host name must be unique . . . 2-58
Creating an IBM Cloud Foundry application (3 of 3) . . . 2-59
IBM Cloud: Application Details page . . . 2-60
Testing applications through the application route . . . 2-61
Adding an IBM Cloud service . . . 2-62
Binding a service to an application (1 of 2) . . . 2-63
Binding a service to an application (2 of 2) . . . 2-64
IBM Cloud: Environment variables . . . 2-65
2.6. IBM Cloud regions and availability zones (optional) . . . 2-66
IBM Cloud regions and availability zones (optional) . . . 2-67
Topics . . . 2-68
IBM Cloud: Choice of regions and availability zones . . . 2-69
IBM Cloud: Multi-Availability Zones . . . 2-71
2.7. Managing your IBM Cloud users and resources (optional). . . . 2-72
Managing your IBM Cloud users and resources (optional) . . . 2-73
Topics . . . 2-74

IBM Cloud: Identity and Access Management 2-75
IBM Cloud: Resources, users, and access control 2-76
IBM Cloud: Resource groups 2-77
IBM Cloud: Resource controller 2-79
IBM Cloud: Access group 2-81
IBM Cloud: IAM policies 2-82
IBM Cloud: Assigning access to group 2-83
IBM Cloud: Cloud IAM roles 2-84
IBM Cloud: Organizing resources 2-85
Cloud Foundry: Organizing resources 2-86
Cloud Foundry: Access 2-87
Cloud Foundry: Organizations 2-88
Cloud Foundry: Domains and quota 2-89
Cloud Foundry: Spaces 2-90
Cloud Foundry: User roles 2-91
Managing Cloud Foundry organizations 2-92
Inviting users to IBM Cloud 2-93
Unit summary 2-94
Related resources 2-95
Review questions 2-96
Review questions (cont.) 2-97
Review answers 2-98

Unit 3. Deploying applications to Cloud Foundry on IBM Cloud 3-1
Unit objectives 3-2
3.1. Introduction to Cloud Foundry 3-3
Introduction to Cloud Foundry 3-4
Topics 3-5
Cloud Foundry benefits 3-6
3.2. Deploying Cloud Foundry applications with IBM Cloud CLI 3-7
Deploying Cloud Foundry applications with IBM Cloud CLI 3-8
Topics 3-9
IBM Cloud CLI overview 3-10
Deploying your first Node.js application – Prerequisites: Prepare IBM Cloud 3-11
Deploying your first Node.js application - Prerequisites: Prepare IBM Cloud (cont.) 3-12
Deploying your first Node.js application - Prerequisites: Prepare IBM Cloud (cont.) 3-13
Deploying your first Node.js application - Step 1: Understanding the sample application 3-14
Deploying your first Node.js application - Step 1: Understanding the sample application (cont.) . . . 3-15
Deploying your first Node.js application - Step 1: Understanding the sample application (cont.) . . . 3-16
Deploying your first Node.js application - Step 2: Deploying the sample app 3-17
Deploying your first Node.js application - Step 3: Checking whether your app is running 3-18
Exploring your deployed application 3-19
3.3. Organizations and spaces 3-20
Organizations and spaces 3-21
Topics 3-22
Organizations and spaces 3-23
Orgs 3-24
Spaces 3-25
3.4. Buildpacks 3-26
Buildpacks 3-27
Topics 3-28
Introduction to buildpacks 3-29
IBM Cloud provided buildpacks 3-31
IBM Cloud catalog view of buildpacks 3-32
Using custom buildpacks 3-33
3.5. Resiliency 3-34

Resiliency 3-35
Topics 3-36
Making your app resilient 3-37
Making your app resilient by using IBM Cloud CLI 3-38
Making your app resilient by using IBM Cloud CLI (cont.) 3-39
3.6. Logging and debugging 3-40
Logging and debugging 3-41
Topics 3-42
Debugging your application’s deployment 3-43
Viewing logs from the IBM Cloud dashboard 3-44
3.7. Domains and routes 3-45
Domains and routes 3-46
Topics 3-47
Domains and routes 3-48
Example of a route 3-50
Setting up routes 3-51
3.8. Binding external services 3-53
Binding external services 3-54
Topics 3-55
External services 3-56
Example of bound services 3-57
Bound services in environment variables 3-58
3.9. Next steps 3-59
Next steps 3-60
Topics 3-61
Further reading 3-62
Unit summary 3-63
Review questions 3-64
Review questions (cont.) 3-65
Review questions (cont.) 3-66
Review answers 3-67
Exercise 1: Getting started with Cloud Foundry apps on IBM Cloud 3-68
Exercise objectives 3-69

Unit 4. Adopting a DevOps approach by using IBM Continuous Delivery 4-1
Unit objectives 4-2
4.1. Introduction to DevOps 4-3
Introduction to DevOps 4-4
Topics 4-5
What is DevOps 4-6
Software lifecycle for applying DevOps practices 4-8
DevOps practices 4-9
IBM Cloud Garage Method 4-11
Benefits of DevOps 4-13
Deployment strategies 4-14
Deployment strategies – Blue/green deployment 4-15
Deployment strategies – Canary deployment 4-16
Deployment strategies – A/B testing 4-17
4.2. DevOps services on IBM Cloud 4-18
DevOps services on IBM Cloud 4-19
Topics 4-20
DevOps services on IBM Cloud 4-21
4.3. IBM Cloud Continuous Delivery overview 4-23
IBM Cloud Continuous Delivery overview 4-24
Topics 4-25
What is IBM Cloud Continuous Delivery 4-26

Toolchain templates 4-28
What tools does IBM Cloud Continuous Delivery provide 4-29
Adding Continuous Delivery 4-30
Creating Continuous Delivery Toolchains 4-31
Adding Tools to the toolchain 4-32
Adding Tool Integration to a toolchain 4-33
4.4. Web IDE (Edit Code) 4-34
Web IDE (Edit Code) 4-35
Topics 4-36
Web IDE: Edit code features 4-37
Web-based integrated development environment 4-38
Editing source code 4-39
Editor features: Code completion 4-40
Editor features: Run bar 4-41
IBM Cloud Live Sync features 4-42
4.5. Source Code Management (Git repository) and Issue Tracker 4-43
Source Code Management (Git repository) and Issue Tracker 4-44
Topics 4-45
Git repository and Issue Tracking 4-46
Source control with a Git repository 4-47
Git repository overview from Web IDE 4-48
Connecting a Git client to your repository on IBM Cloud 4-49
Issue Tracking tool 4-50
Creating an issue 4-51
4.6. Automated build and deployment (Delivery Pipeline) 4-52
Automated build and deployment (Delivery Pipeline) 4-53
Topics 4-54
Delivery pipeline 4-55
Customizing Delivery Pipeline 4-56
Example: Default delivery pipeline 4-57
Configuring the build stage 4-58
Configuring build jobs 4-59
Configuring deploy jobs 4-61
Configuring test jobs 4-62
Example: A successful build and deploy result 4-63
Unit summary 4-65
Review questions 4-66
Review questions (cont.) 4-67
Review answers 4-68
Exercise 2: Developing IBM Cloud applications with IBM Cloud Continuous Delivery 4-69
Exercise objectives 4-70
References 4-71

Unit 5. REST architecture and Watson APIs 5-1
Unit objectives 5-2
5.1. REST concepts, architecture, and characteristics 5-3
REST concepts, architecture, and characteristics 5-4
Topics 5-5
What is REST 5-6
REST characteristics 5-7
REST architecture constraints 5-8
Resource Identifier and URIs 5-11
Best practices to model URIs in REST 5-13
5.2. Applying REST to server-side applications. 5-16
Applying REST to server-side applications 5-17
Topics 5-18

Applying REST to server-side applications . . . 5-19
Example: Application Model Architecture for REST services . . . 5-20
What is a RESTful web service . . . 5-21
Best practices for RESTful web services . . . 5-22
Example: Sending an HTTP GET request to a REST service . . . 5-24
Example: Receiving an HTTP GET response from a REST service . . . 5-25
Example: Sending an HTTP POST request to a REST service . . . 5-27
Example: Receiving an HTTP POST response from a REST service . . . 5-28
5.3. Resources representation and JSON . . . 5-30
Resources representation and JSON . . . 5-31
Topics . . . 5-32
Resource representation in REST . . . 5-33
Introduction to JSON . . . 5-34
JSON data types . . . 5-35
JSON data type: Objects . . . 5-36
JSON data type: Arrays . . . 5-37
5.4. Example: Using REST APIs with IBM Watson . . . 5-38
Example: Using REST APIs with IBM Watson . . . 5-39
Topics . . . 5-40
IBM Watson . . . 5-41
How Watson works . . . 5-43
Watson services on IBM Cloud - Summary . . . 5-44
AI services on IBM Cloud catalog . . . 5-46
Building AI solutions with IBM Watson services on IBM Cloud . . . 5-47
Getting started with Watson services on IBM Cloud . . . 5-48
Options to call Watson APIs . . . 5-49
Watson service example: Watson Natural Language Understanding . . . 5-51
Watson Natural Language Understanding demonstration: Input text . . . 5-54
Watson Natural Language Understanding demonstration: Analyze results . . . 5-56
Additional resources for developers . . . 5-57
Documentation and other information sources . . . 5-58
Unit summary . . . 5-59
Review questions . . . 5-60
Review answers . . . 5-61

Unit 6. Introduction to data services on IBM Cloud . . . 6-1
Unit objectives . . . 6-2
6.1. Introduction to databases . . . 6-3
Introduction to databases . . . 6-4
Topics . . . 6-5
Importance of data . . . 6-6
How data is stored . . . 6-7
Database and data model . . . 6-8
Relational databases . . . 6-9
NoSQL databases . . . 6-10
Different types of NoSQL databases: Key-value . . . 6-12
Different types of NoSQL databases: Document . . . 6-13
Different types of NoSQL databases: Columnar . . . 6-14
Different types of NoSQL databases: Graph . . . 6-16
6.2. Data services in IBM Cloud . . . 6-17
Data services in IBM Cloud . . . 6-18
Topics . . . 6-19
Data services in the IBM Cloud catalog . . . 6-20
Databases services . . . 6-22
Storage services . . . 6-24
Analytics services . . . 6-25

Other related data services 6-26
6.3. IBM Cloudant 6-27
IBM Cloudant 6-28
Topics 6-29
Cloudant capabilities 6-30
Benefits of IBM Cloudant 6-32
Documents in Cloudant 6-33
Getting started with Cloudant in IBM Cloud 6-34
Cloudant with IBM Cloud: Creating credentials 6-36
Cloudant with IBM Cloud: Creating connections 6-37
Cloudant Dashboard 6-39
Cloudant HTTP API 6-40
Reading a document in Cloudant 6-41
View all documents 6-42
More Cloudant HTTP APIs 6-43
Cloudant Query 6-44
Cloudant indexes 6-45
HTTP status codes 6-47
Unit summary 6-48
Review questions 6-49
Review questions (cont.) 6-50
Checkpoint answers 6-51
Exercise 3: IBM Cloud with Cloudant 6-52
Exercise objectives 6-53
Documentation and other information sources 6-54
Importance of data (backup slide) 6-55
Atomicity – Consistency – Isolation – Durability (backup slide) 6-56
Basically Available – Soft State – Eventual consistency (backup slide) 6-57
Features of NoSQL databases (backup slide) 6-58
Cloudant Index and other types of queries (backup slide) 6-59
Cloudant best practices (backup slide) 6-61

Unit 7. Enriching your applications with IBM Cloud services 7-1
Unit objectives 7-2
7.1. Business problem and requirements 7-3
Business problem and requirements 7-4
Topics 7-5
Define your business problem and goals 7-6
Cognitive Tweets Analyzer: Business problem and objectives 7-8
Functional and Non-functional requirements 7-10
Cognitive Tweets Analyzer : Functional requirements 7-12
Cognitive Tweets Analyzer : Non-functional requirements 7-14
7.2. Solution architecture 7-15
Solution architecture 7-16
Topics 7-17
IBM Cloud Architecture Center 7-18
Cognitive Tweets Analyzer: Architecture 7-20
7.3. Components 7-23
Components 7-24
Topics 7-25
Node.js server app and Angular web UI 7-26
App ID service: Overview 7-29
App ID service: How it works 7-31
App ID service: Authentication options 7-33
App ID service: Scenarios and solutions (optional) 7-35
How is App ID used in Cognitive Tweets Analyzer 7-36

Retrieving and processing tweets 7-37
Watson Natural Language Understanding service 7-39
How is Natural Language Understanding used in Cognitive Tweets Analyzer 7-42
How is Natural Language Understanding used in Cognitive Tweets Analyzer 7-43
Watson Tone Analyzer service 7-44
How is Tone Analyzer used in Cognitive Tweets Analyzer 7-46
How is Tone Analyzer used in Cognitive Tweets Analyzer 7-47
Cloudant database 7-48
How is Cloudant used in Cognitive Tweets Analyzer 7-50
LogDNA service 7-51
Integrating LogDNA with Cognitive Tweets Analyzer 7-54
Integrating LogDNA with Cognitive Tweets Analyzer (cont.) 7-55
Integrating LogDNA with Cognitive Tweets Analyzer (cont.) 7-56
IBM Cloud Monitoring service 7-57
Integrating Monitoring with Cognitive Tweets Analyzer 7-59
Integrating Monitoring with Cognitive Tweets Analyzer 7-60
7.4. Cognitive Tweets Analyzer: Demonstration 7-61
Cognitive Tweets Analyzer: Demonstration 7-62
Topics 7-63
Cognitive Tweets Analyzer App: Demo 7-64
Cognitive Tweets Analyzer App: Demo (cont.) 7-65
Cognitive Tweets Analyzer App: Demo (cont.) 7-66
Cognitive Tweets Analyzer App: Demo (cont.) 7-67
Cognitive Tweets Analyzer App: Demo (cont.) 7-68
Cognitive Tweets Analyzer App: Demo (cont.) 7-69
Cognitive Tweets Analyzer App: Demo (cont.) 7-70
Cognitive Tweets Analyzer App: Demo (cont.) 7-71
Cognitive Tweets Analyzer App: Demo (cont.) 7-72
Unit summary 7-73
Review questions 7-74
Review questions 7-75
Review questions 7-76
Review answers 7-77
Review answers 7-78
Review answers 7-79
Exercise 4: Securing a web application with single sign-on (optional) 7-80
Exercise objectives 7-81

Unit 8. Developing containerized applications on Kubernetes 8-1
Unit objectives 8-2
8.1. Containers 8-3
Containers 8-4
Topics 8-5
Containers 8-6
Containers versus virtual machines and the benefits of containers 8-7
Dockerfile: Building a container? 8-9
Building a container image from a Dockerfile 8-10
8.2. Container orchestration 8-11
Container orchestration 8-12
Topics 8-13
Container orchestration 8-14
8.3. Introducing Kubernetes. 8-16
Introducing Kubernetes 8-17
Topics 8-18
What is Kubernetes 8-19
8.4. Kubernetes architecture 8-21

Kubernetes architecture . . . 8-22
Topics . . . 8-23
Kubernetes architecture . . . 8-24
8.5. Kubernetes objects . . . 8-25
Kubernetes objects . . . 8-26
Topics . . . 8-27
Interaction with a Kubernetes cluster . . . 8-28
Building a container and storing it in the container registry . . . 8-29
Understanding Kubernetes objects . . . 8-31
Pods . . . 8-32
Deployment . . . 8-33
Scaling the Deployment . . . 8-35
Services . . . 8-36
Services example: NodePort . . . 8-37
Storage: Persistence Volumes . . . 8-38
8.6. Next steps . . . 8-39
Next steps . . . 8-40
Topics . . . 8-41
Further reading . . . 8-42
Unit summary . . . 8-43
Review questions . . . 8-44
Review questions (cont.) . . . 8-45
Review answers . . . 8-46
Review answers (cont.) . . . 8-47

Unit 9. IBM Cloud Kubernetes Service overview . . . 9-1
Unit objectives . . . 9-2
9.1. IBM Cloud Kubernetes Service . . . 9-3
IBM Cloud Kubernetes Service . . . 9-4
Topics . . . 9-5
IBM Cloud Kubernetes Service . . . 9-6
IBM Cloud Kubernetes Service demonstration . . . 9-8
Creating a free cluster . . . 9-9
Instructor demonstration . . . 9-10
Accessing your cluster . . . 9-12
Scaling out . . . 9-13
Kubernetes Dashboard . . . 9-14
Exercise 5: Part 1 . . . 9-15
Planning your cluster for high availability . . . 9-16
Accessing IBM Cloud Kubernetes Service by using kubectl . . . 9-17
IBM Cloud Container Registry . . . 9-18
Vulnerability Advisor . . . 9-19
Using the Private Image Registry . . . 9-20
9.2. Deploying an application . . . 9-21
Deploying an application . . . 9-22
Topics . . . 9-23
Deploying an application on IBM Cloud Kubernetes Service . . . 9-24
Deploy an application on IBM Cloud Kubernetes Service . . . 9-25
9.3. Summary and further reading . . . 9-26
Summary and further reading . . . 9-27
Topics . . . 9-28
Further reading . . . 9-29
Unit summary . . . 9-30
Review questions . . . 9-31
Review questions (cont.) . . . 9-32
Review answers . . . 9-33

Review answers (cont.) 9-34
Exercise 5: Managing IBM Cloud Kubernetes Service clusters 9-35
Exercise objectives 9-36
Exercise 6: Deploying an application on Kubernetes 9-37
Exercise objectives 9-38

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

BLU Acceleration®	Cloudant®	DB™
Db2®	Express®	IBM BLU®
IBM Cloud Garage™	IBM Cloud™	IBM Watson®
OpenWhisk®	SlamTracker®	THINK®
UrbanCode®	Watson IoT™	Watson™
WebSphere®		

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

Microsoft and Windows 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.

Social® is a trademark or registered trademark of TWC Product and Technology, LLC, an IBM Company.

Other product and service names might be trademarks of IBM or other companies.

Course description

Essentials of Cloud Application Development

Duration: 3 days

Purpose

The Cloud Application Developer career path prepares students to develop, build, deploy, and test applications that use a cloud platform to build Software as a Service (SaaS) solutions.

Module II, *Cloud Application Developer,* consists of two courses:

- Course I - Essentials of Cloud Application Development.
- Course II - Developing Node.js Applications on IBM Cloud.

This book is the course guide for *Course I - Essentials of Cloud Application Development*. This course 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 course introduces containers and container orchestration and provides an overview of the Kubernetes platform.

Audience

Undergraduate senior students from IT related academic programs, for example, computer science, software engineering, information systems and others.

Prerequisites

Before attending *Course I - Essentials of Cloud Application Development*, students must meet the following prerequisites:

- Successful completion of Module I *Cloud Application Foundations* (self-study).
- Successful completion of Exercise 0, *Setting up your hands-on environment* (self-study).

Objectives

After completing this course, you should be able to:

- Define cloud computing.
- Describe the characteristics and benefits of cloud.
- Describe cloud services models (IaaS, PaaS and SaaS).
- Describe the cloud deployment options (Private, Public, Hybrid).
- Describe the choices that developers have when building cloud applications.

- Describe IBM Cloud.
- Identify the runtimes and services that IBM Cloud offers.
- Distinguish among the various compute options on IBM Cloud.
- Work with IBM Cloud resources.
- Explain Cloud Foundry basic concepts.
- Describe DevOps.
- Describe the capabilities of IBM Cloud Continuous Delivery.
- Explain how to build and deploy applications using DevOps tools on IBM Cloud.
- Explain the REST architecture style for designing networked applications.
- List best practices to follow when using REST in your application.
- Provide examples of REST APIs using IBM Watson.
- Describe different databases types and capabilities
- Describe the main types of data services in IBM Cloud.
- Design a simple architecture for cloud applications.
- Identify services listed in the IBM Cloud catalog that you can integrate in your applications.
- Explain containers and the difference between containers and virtual machines.
- Describe the Kubernetes building blocks.
- Explore IBM Cloud Kubernetes Service on Kubernetes platforms.

Agenda

Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Students in this course use an IBM Cloud Lite account to perform the exercises. This account will never expire, therefore students can continue working on IBM Cloud after the class.

Day 1

(00:30) Welcome
(01:00) Unit 1 - Introduction to cloud computing
(02:00) Unit 2 - Introduction to IBM Cloud
(01:00) Lunch break
(01:00) Unit 3 - Deploying applications to Cloud Foundry on IBM Cloud
(01:15) Exercise 1 - Getting started with Cloud Foundry apps on IBM Cloud
(01:30) Unit 4 - Adopting a DevOps approach by using IBM Continuous Delivery

Day 2

(01:30) Exercise 2 - Developing IBM Cloud applications with IBM Cloud Continuous Delivery
(01:30) Unit 5 - REST architecture and Watson APIs
(01:00) Lunch break
(01:30) Unit 6 - Introduction to data services on IBM Cloud
(00:45) Exercise 3 - IBM Cloud with Cloudant
(01:00) Unit 7 - Enriching your application with IBM Cloud services

Day 3

(02:00) Unit 8 - Developing containerized applications on Kubernetes
(01:00) Unit 9 - IBM Cloud Kubernetes Service overview
(01:00) Lunch break
(00:30) Exercise 5 - Managing IBM Kubernetes Service clusters
(01:30)Exercise 6 - Deploying an application on Kubernetes

Unit 1. Introduction to cloud computing

Estimated time

01:00

Overview

This unit provides an overview about cloud computing. It lists characteristics and benefits of cloud computing and describes cloud computing service and deployment models.

IBM Training

Unit objectives

- Define cloud computing.
- Describe the characteristics of Cloud.
- Describe the benefits of Cloud and the factors contributing to its growth.
- Describe cloud services models (IaaS, PaaS and SaaS).
- Describe the cloud deployment options (Private, Public, Hybrid).
- Describe cloud-native applications and development methods.
- Explain the Twelve-Factor App methodology.
- Describe the choices that developers have when building cloud applications.

Introduction to cloud computing

Figure 1-1. Unit objectives

1.1. Introduction to cloud computing

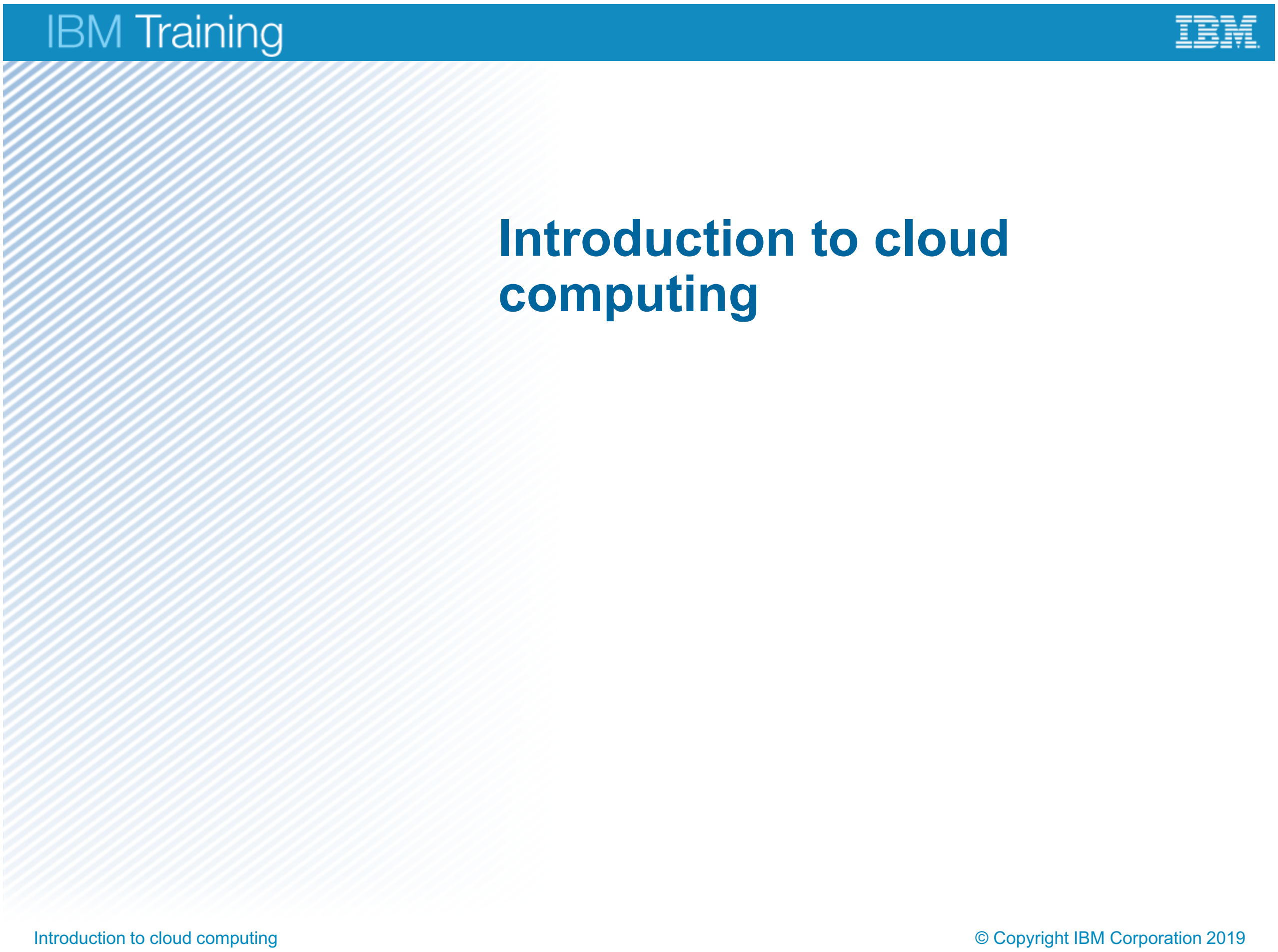


Figure 1-2. Introduction to cloud computing

IBM Training

Topics

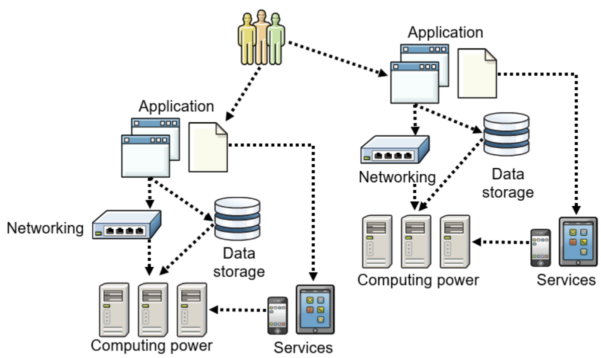
- Introduction to cloud computing
- Cloud service models
- Cloud deployment models

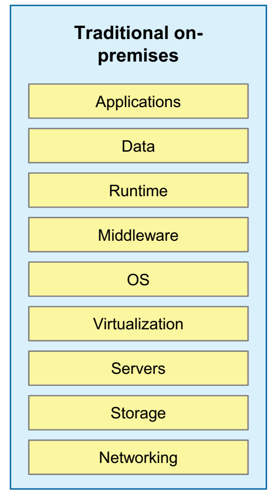
Introduction to cloud computing

Figure 1-3. Topics

IBM Training

Before cloud computing





Introduction to cloud computing

Figure 1-4. Before cloud computing

Before you learn about cloud computing, you should know more about the software industry *before* cloud computing.

Before cloud computing, when you created a basic website for your clients, you started by developing your application with a programming language, such as Java, Node.js, or PHP. Then, you deployed it on a physical machine (server). On this server, you had an operating system and set up the configurations and middleware that were needed to run your application. Also, you needed a run time to run your application, such as Apache Tomcat or IBM WebSphere Liberty application server (if you used Java). Your application had to be linked to a database.

Then, to expose this application to your client, you needed an IP and domain name, and handled the network configurations, the physical location for your servers, and the electricity that was required for your servers. Security had to be set up and maintained. You had to manage the upgrades for these resources. You needed a large team of experts to install, configure, test, run, secure, and update these resources to keep your website running.

IBM Training

Challenges faced before cloud computing

- Cost
- Scalability
- Reliability
- Security
- Mobility

Introduction to cloud computing

Figure 1-5. Challenges faced before cloud computing

Here some of the challenges that were faced before cloud computing:

Cost

It was costly to build reliable and maintainable software. You likely built you own infrastructure that might be needed to be turned into a whole data center, which included servers, network equipment, data storage, and so on. You needed to hire a team of experts to handle all these resources. All these factors made it difficult for small and midsized business to develop their own software and keep updating it.

Scalability

If there was high demand from your clients for your application, you needed to scale up the capacity of your application, which required more resources and some downtime to integrate and upgrade those resources. If demand decreased, you had some resources that were not used effectively.

Reliability

With any operation that is done to your server, such as maintenance and updates, you need downtime to perform those operations. Some issues that might have caused downtime for your application are power outages, hardware problems, general network issues, or even natural disasters.

Security

Security is necessary for all levels: application, network, infrastructure, and physical resources.

Mobility

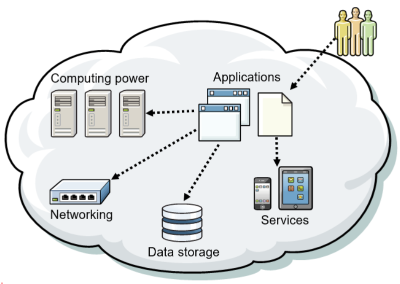
Part of the team worked onsite to, at least, set up the infrastructure and configured the network.

IBM Training

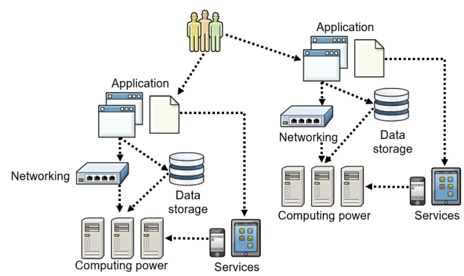
What is cloud computing

Cloud computing, often referred to as “the cloud,” is the delivery of on-demand computing resources (applications to data centers) on a pay-as-you-go basis.

- **Elastic resources**
- **Metered services**
- **Self-service**



Cloud computing model



Traditional on-premises computing model

Introduction to cloud computing

Figure 1-6. What is cloud computing

Cloud computing is a model for enabling convenient, on-demand access to provider-managed suite of both hardware and software resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing is a disruptive change in the IT industry that represents a new model for the IT infrastructure that is different from traditional IT computing models. Cloud computing enables ubiquitous computing, where computing is available 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.

Elastic resources: Scale up or down quickly and easily to meet changing demand.

Metered services: Pay only for what you use.

Self-service: Find all the IT resources that you need by using self-service access.

IBM Training

Characteristics of the cloud

Cloud makes hardware and platform resources readily available and quick to configure. Cloud provides the following characteristics to developers:

- On-demand resources
- Self-service
- Ubiquitous access
- Resource pooling
- Rapid elasticity
- Measured service

Introduction to cloud computing

Figure 1-7. Characteristics of the cloud

Modern 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.

The cloud makes hardware resources readily available and quick to configure, which shortens the time that is 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.

Characteristics of the cloud:

- On-demand resources: Have it when you need it with no need for tiresome preparation, downloads, and installations.
- Self-service: A customer can provision resources themselves by accessing a self-service portal and requesting the resource that they want.
- Ubiquitous access: Access the cloud from anywhere just by using an internet connection and a cloud account (user name and password).
- Resource pooling: Pooling hardware resources and abstracting them so that when resources are not being used by one customer, instead of sitting idle, those resources can be used by another customer.

- Rapid elasticity: Scaling up or down resource consumption is available on demand with any quantity and at any time.
- Measured service: Pay only for what you use, which helps you monitor any wastage of resources.

Examples of cloud resources include:

- Servers
- Storage
- Networks
- Security
- Applications
- Platforms
- Runtimes
- Databases
- Managed services

IBM Training

Benefits of the cloud

- Achieves economies of scale.
- Goes from CAPEX to OPEX.
- Runs anytime and anywhere.
- Facilitates the agile methodology.
- Ensures global availability.
- Built-in security
- Provides advanced capabilities.

Introduction to cloud computing

Figure 1-8. Benefits of the cloud

Enterprises eager to undergo digital transformations and modernize their applications are quick to see the value of adopting a cloud-computing platform. They are increasingly finding business agility or cost savings by renting software and infrastructure. Each cloud-computing service and deployment model type provides you with different levels of control, flexibility, and management.

- Achieves economies of scale (do more with less).
  Economies of scale decrease costs because of increased production. These economies became achievable in software because of the flexibility of the cloud.
- Reduces capital expenditure (CAPEX) by moving to the operational expenditure (OPEX) model (use only when needed).
  CAPEX is the money that is used to acquire or update the assets of a firm. OPEX is the money that is used on running operations. So, in the software industry the “pay as you go” model helps you go from CAPEX to OPEX.
- Runs anytime and anywhere (access to services, on any device, and anywhere in the world).

- Facilitates agile methodology (faster time to market).
  Agile methodology is a development methodology where you engage the client with the development team and constantly get changing requirements that are embraced for the client's competitive advantage. Applying the agile methodology became achievable because of the cloud.
- Ensures Global Availability (focus on developing applications, and the rest automatically follows).
  It helps to improve reliability and provide a good disaster recovery plan with high availability.
- Built-in security

  Cloud providers typically have standards to build their environments and standardized practices to run operations that meet the security needs of enterprise clients. As a user of cloud, your application could benefit from higher orders of security by virtue of it being built into the cloud offering for all.
- Provides advanced capabilities (advanced technology that is readily available and you can experiment with).
  Many advanced technologies, such as big data and AI services that need high-computing-power capabilities would not be available without cloud computing.

IBM Training IBM

Factors contributing to the growth of the 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.

Introduction to cloud computing

Figure 1-9. Factors contributing to the growth of the cloud

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.

The cloud makes hardware resources readily available and quick to configure, which shortens the time that is required for developers to show a working version of their products.

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 that drives 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.

1.2. Cloud service models

IBM Training

Cloud service models

Introduction to cloud computing

Figure 1-10. Cloud service models

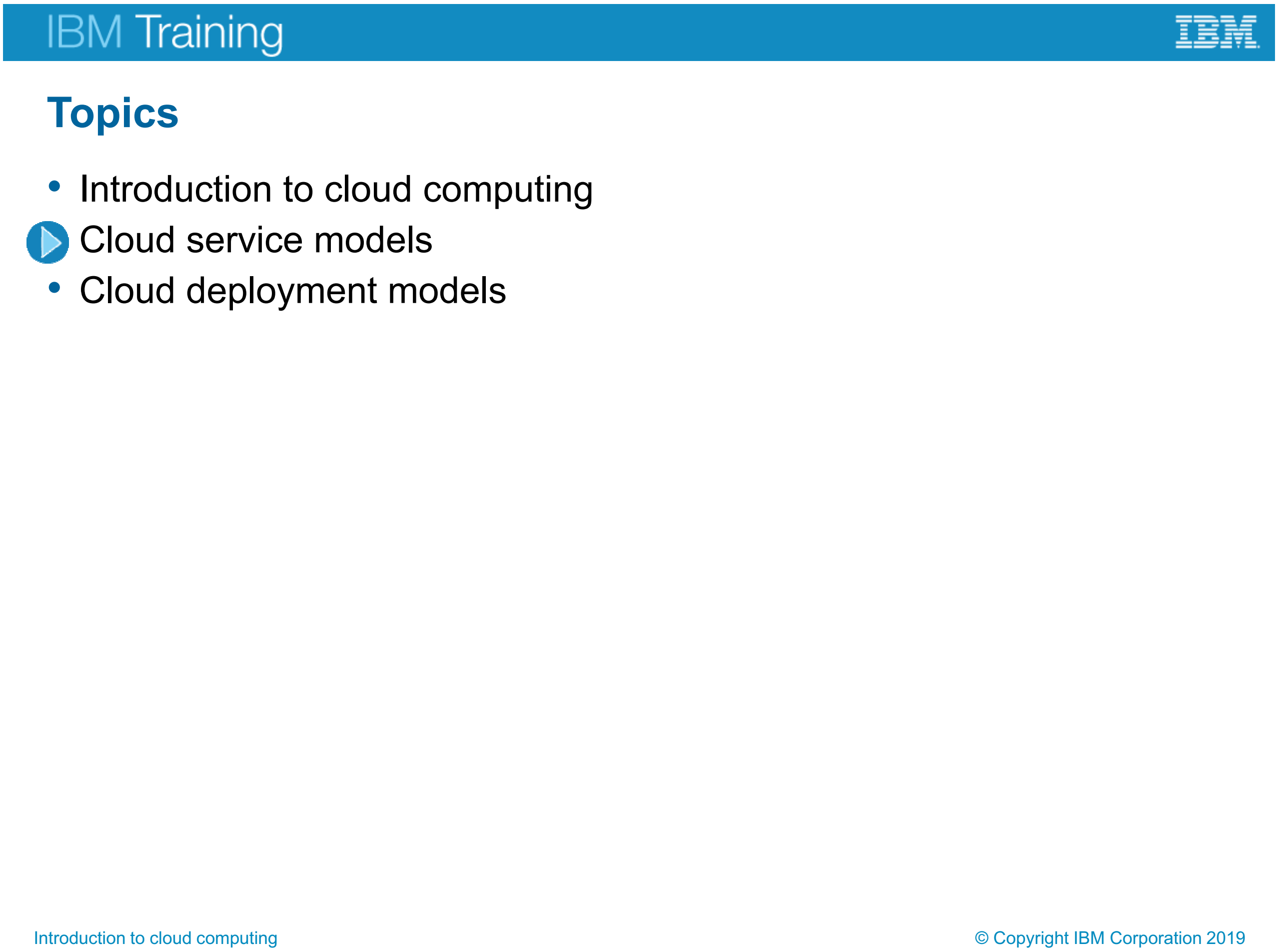


Figure 1-11. Topics

IBM Training

The pizza analogy

The cloud has different service models. With platform, infrastructure, and software offered as services, the pizza analogy is an easy way to understand this approach.

Traditional	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
○ Table & Chairs	○ Table & Chairs	○ Table & Chairs	● Table & Chairs
○ Drinks	○ Drinks	○ Drinks	● Drinks
○ Oven	○ Oven	● Oven	● Oven
○ Toppings	● Toppings	● Toppings	● Toppings
○ Dough Base	● Dough Base	● Dough Base	● Dough Base
Make from scratch at home	**Buy pizza and bake home**	**Get pizza delivered**	**Dine at Pizza Restaurant**

○ = you furnish; ● = vendor furnishes

Introduction to cloud computing

Figure 1-12. The pizza analogy

Cloud computing can be explained by using the pizza analogy:

- You build a pizza by preparing the dough, purchasing certain toppings, heating the oven, baking it, and then serving and eating the pizza along with drinks at home.
- *Infrastructure as a Service* (IaaS) is like buying a pre-made pizza from the supermarket. You bake it in your oven, serve it with drinks, and eat the pizza at home.
- *Platform as a Service* (PaaS) is like ordering a pizza from a pizza delivery restaurant. The pizza is prepared by the restaurant and delivered to your front door. You provide the drinks and eat it at home.
- *Software as a Service* (SaaS) is like going to a restaurant and eating the pizza there while enjoying the company of others and sharing the atmosphere of the restaurant.

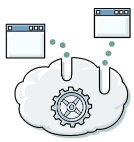
When translating this analogy to the cloud, we can say that:

- To build an application, you must provide the infrastructure, platforms, operating systems, networking components, and so on, which is like making a pizza at home.
- With IaaS, you order hardware and an infrastructure. Often, this infrastructure is managed for you. You deploy only the middleware, runtime, and your application. The infrastructure is like the pizza that is pre-made, and you bake it to your liking.
- PaaS is like getting a pizza delivered. The pizza is ready to be eaten, and need only to provide drinks to go with it. In the cloud, this means that the cloud provider offers access to the platform and runtime and you only need to push the application.
- SaaS is using an application that is hosted at the cloud provider, which is similar to going to a restaurant and enjoying your pizza there.

IBM Training

Cloud service models

IaaS: Infrastructure as a Service



PaaS: Platform as a Service

SaaS: Software as a Service

Introduction to cloud computing

Figure 1-13. Cloud service models

IaaS

A cloud provider offers clients pay-as-you-go access to storage, networking, servers, and other computing resources in the cloud.

PaaS

A cloud provider offers access to a cloud-based development environment in which users can build and deliver applications. The provider supplies and manages the underlying infrastructure.

SaaS

A cloud provider delivers software and applications through the internet that are ready to be consumed. Users subscribe to the software and access it through the web or vendor application programming interfaces (APIs).

IBM Training

Infrastructure as a Service

Key features:

- Instead of purchasing hardware, users pay for IaaS on demand.
- Infrastructure is scalable depending on your processing and storage needs.
- You avoid the cost of buying and maintaining your own hardware.
- Enables the virtualization of administrative tasks, which frees time for other work.

Introduction to cloud computing

Figure 1-14. Infrastructure as a Service

IaaS is a cloud computing offering in which a vendor provides users access to computing resources, such as servers, storage, and networking.

IaaS offerings are built on top of a standardized, secure, and scalable infrastructure. The virtualization of the hardware is performed by a program that is known as a hypervisor. A hypervisor manages *virtual machines (VMs)* or *virtual servers*, which hosts 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.

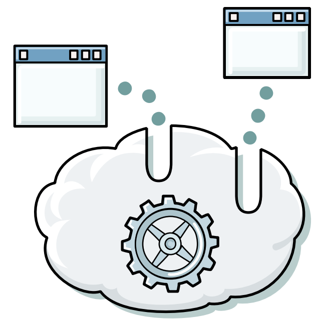
Organizations use their own platforms and applications within a service provider's infrastructure.

IBM Training

Platform as a Service

Key features:

- PaaS provides a platform with tools to test, develop, and host applications in the same environment.
- Enables organizations to focus on software development without having to worry about the underlying infrastructure.
- Providers manage security, operating systems, server software, and backups.
- Facilitates collaborative work even if teams work remotely.



Introduction to cloud computing

Figure 1-15. Platform as a Service

PaaS is a cloud-computing offering that provides users with a cloud environment in which they can develop, manage, and deliver applications.

In addition to instances and other computing resources, users can use a suite of prebuilt tools and run times to develop, customize, and test their own applications.

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 binary files and dependencies for the application.
2. Structure their application code into the correct directory tree for containerization.
3. Provision a container (or set of containers) on which the application can run.
4. Automatically generate a simple and basic networking configuration for access to the application.
5. Provide automatic and built-in monitoring of the application.
6. You can 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 uses 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 an automatic method for scaling. For example, consider a situation where the developer wants more hardware resources that are 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.

IBM Training

Software as a Service

Key features:

- SaaS vendors provide users with software and applications through a subscription model.
- Users do not have to manage, install, or upgrade software; SaaS providers manage all of those items.
- Data is secure in the cloud; equipment failure does not result in loss of data.
- Applications are accessible from almost any internet-connected device from anywhere in the world.

Introduction to cloud computing

Figure 1-16. Software as a Service

SaaS is a cloud-computing offering that provides users with access to a vendor's cloud-based software. Users do not install applications on their local devices.

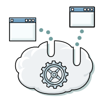
Instead, the applications are on a remote cloud network that is accessed through the web or an API. Through the application, users can store and analyze data and collaborate on projects.

IBM Training

Example of cloud services

IaaS

- Virtual servers
- Bare metal machines
- Block storage
- File share storage
- Object storage
- Backup
- IP management
- Virtual private networks
- Firewalls
- Load balancers
- Automation



PaaS

- Run times and development platforms
- Databases
- Analytics
- Integration
- Starter kits
- Mobile platforms
- Push notifications
- Messaging
- Developer tools
- Continuous integration / continuous delivery

SaaS

- Email and Collaboration
- Customer relationship manager (CRM)
- Enterprise resource planning (ERP)
- Video streaming
- Marketing
- Talent management
- Advertising

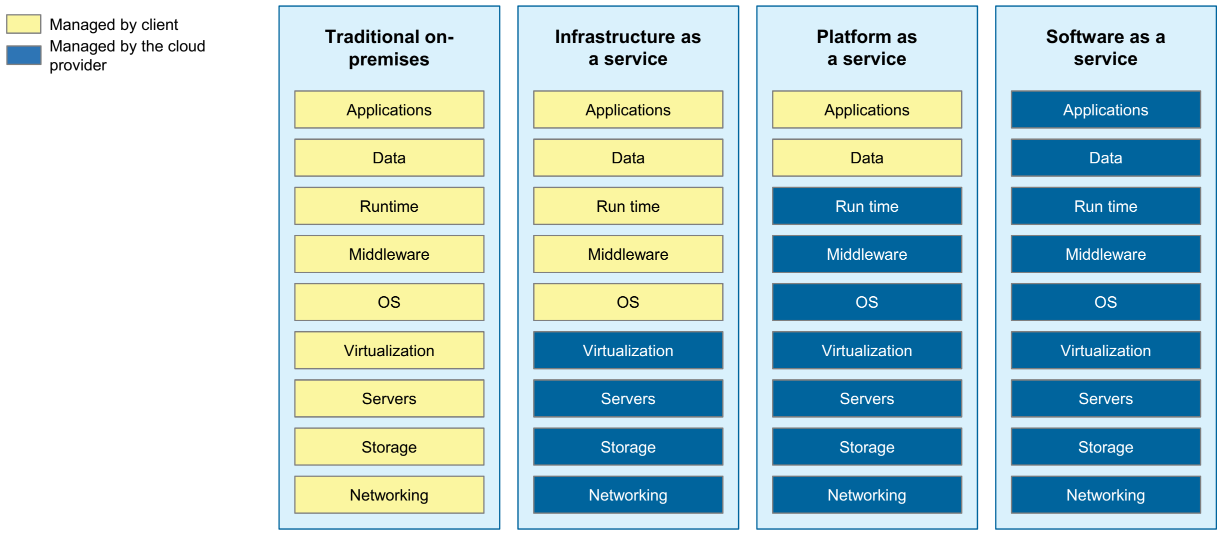
Introduction to cloud computing

Figure 1-17. Example of cloud services

This slide shows examples of the services that are available for each model.

IBM Training

Cloud provider and client responsibilities



Introduction to cloud computing

Figure 1-18. Cloud provider and client responsibilities

This slide 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 the slide; however, the flexibility also is reduced.

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

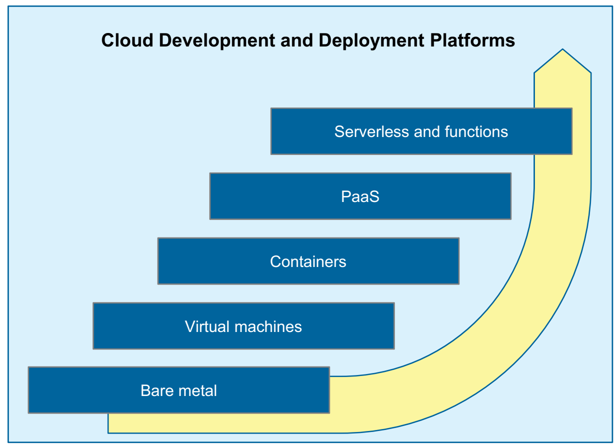
Most enterprises end up using some combination of all of the models that are shown in the slide.

IBM Training

Choices when building cloud applications

When developing applications for the cloud, developers have many options to choose from in terms of platforms, frameworks, tools, and services:

1. Traditional development
   - Example: Bare metal or VMs
2. Containerization
   - Example: Docker or Kubernetes
3. PaaS
   - Example: Cloud Foundry
4. Serverless and functions
   - Example: OpenWhisk



Introduction to cloud computing

Figure 1-19. Choices when building cloud applications

Developers have many options when deploying their applications to the cloud:

1. Traditional development involves deploying applications to bare metal servers or VMs. Bare metal servers enable isolation and single tenancy, and VMs enable shared resources and multi-tenancy. When using traditional deployment, the developers are responsible for deploying and managing all the needed runtimes and dependencies for the application.

2. With containers, developers can package applications with runtimes by using a lightweight packaging medium that is called *containers*. Unlike VMs, which provide hardware virtualization, containers provide operating system-level virtualization.

3. PaaS solutions such as Cloud Foundry enable developers to focus on coding and pushing code to a platform that takes care of deployment and scaling. Developers do not need to worry about packaging or maintaining the host that runs the applications.

4. Server-less is a way for developers to focus on the development of event-based applications. The application code is broken into separate functions, and each function can be independently triggered by an event or an action, such as an API call.

1.3. Cloud deployment models

Figure 1-20. Cloud deployment models

IBM Training

Topics

- Introduction to cloud computing
- Cloud service models
- Cloud deployment models

Introduction to cloud computing

Figure 1-21. Topics

IBM Training

Cloud deployment models

The various types of cloud-computing deployment models include *public cloud*, *private cloud*, and *hybrid cloud*.

Public

Public clouds are owned and operated by cloud providers that offer rapid access over a public network to affordable computing resources.

Private

A private cloud is infrastructure that is operated solely for a single organization.

Hybrid

A hybrid cloud uses a private cloud foundation that is combined with the strategic integration and use of public cloud services.

Introduction to cloud computing

Figure 1-22. Cloud deployment models

Public cloud

Public clouds are owned and operated by cloud providers that offer rapid access over a public network to affordable computing resources.

Here are key aspects of a public cloud:

- Enables flexible and scalable IaaS for storage and compute services at a moment’s notice.
- Enables powerful PaaS for cloud-based application development and deployment environments.
- Gives access to innovative SaaS business apps for applications ranging from customer resource management (CRM) to transaction management and data analytics.

Private cloud

A private cloud is an infrastructure that is operated solely for a single organization. Private clouds can take advantage of cloud's efficiencies while providing more control of resources and allowing clients to steer clear of multitenancy.

Here are key aspects of a private cloud:

- Provides self-service interface controls services, which enable IT staff to provision, allocate, and deliver quickly on-demand IT resources.
- Facilitates highly automated management of resource pools for everything from compute capability to storage, analytics, and middleware.
- Provides sophisticated security and governance for a company's specific requirements.

Hybrid cloud

A hybrid cloud uses a private cloud foundation that is combined with the strategic integration and use of public cloud services. Most companies with private clouds evolve to manage workloads across data centers, private clouds, and public clouds, which creates hybrid clouds.

Here are key aspects of a hybrid cloud:

- Enables companies to keep critical applications and sensitive data within a traditional data center environment or private cloud.
- Enables taking advantage of public cloud resources like SaaS for the latest applications and IaaS for elastic virtual resources.
- Facilitates portability of data, apps, and services and more choices for deployment models.

IBM Training

Cloud-native applications

Cloud-native applications capitalize on the scalability and flexibility of the cloud:

- Applications are broken into separate services called *microservices*.
- Microservices can be developed in different programming languages (polyglot development).
- Microservices communicate with each other by using an agreed upon protocol (such as REST or gRPC).
- Microservices work together as a whole to make up an application, yet each can be independently scaled, continuously improved, and quickly iterated through automation and orchestration processes.

Introduction to cloud computing

Figure 1-23. Cloud-native applications

Here are some advantages of cloud-native apps:

- Compared to traditional monolithic apps, cloud-native applications can be easier to manage as iterative improvements occur through agile and DevOps processes.
- Composed of individual microservices, cloud-native applications can be improved incrementally and automatically to add continuously new and improved application features.
- Improvements can be made non-intrusively, causing no downtime or disruption of the user experience.
- Scaling up or down proves easier with the elastic infrastructure that underpins cloud-native apps.
- The cloud-native development process more closely matches the speed and innovation that is demanded by today's business environment.

Even with the advantages that are provided by cloud-native applications, there are also some disadvantages to consider:

- Although microservices enable an iterative approach to application improvement, they also create the necessity of managing more elements. Rather than one large application, it becomes necessary to manage far more small and discrete services.
- Cloud-native apps demand more toolsets to manage the DevOps pipeline, replace traditional monitoring structures, and control microservices architecture.
- Cloud-native applications allow for rapid development and deployment, but they also demand a business culture that can cope with the pace of that innovation.

Cloud-native versus cloud-enabled

A cloud-enabled application is an application that was developed for deployment in a traditional data center, but was later changed so that it also could run in a cloud environment. However, cloud-native applications are built to operate only in the cloud. Developers design cloud-native applications to be scalable, platform-neutral, and composed of microservices.

IBM Training

Cloud-native development methods

When developing cloud-native applications, developers must understand and adopt new methods and patterns to maximize the capability that is provided by the cloud provider:

- Readily available sandbox and production environments
- Programming languages and frameworks
- APIs
- Developer toolchains

Introduction to cloud computing

Figure 1-24. Cloud-native development methods

When developing cloud-native applications, developers must understand and adopt new methods and patterns to maximize the capability that is provided by the cloud provider:

- 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.
- A developer toolchain facilitates integrated development, test, and debugging:
    - The new model is to integrate development and operations teams into DevOps.
    - Build engine for compilation and testing.

IBM Training

Cloud-native development methods (cont.)

Cloud-native development introduces the 12-factor app methods and patterns to development:

I. Codebase: One codebase that is tracked in revision control, but there are many deployments.
II. Dependencies: Explicitly declare and isolate dependencies.
III. Configuration: Store the configuration in the environment.
IV. Backing services: Treat backing services as attached resources.
V. Build, release, and run: Strictly separate build and run stages.
VI. Processes: Run the app as one or more stateless processes.

Introduction to cloud computing

Figure 1-25. Cloud-native development methods (cont.)

Examples for the first six factors:

- Factor 1. One codebase that is tracked in revision control with multiple deployments: Use one codebase for tracking and revision control, and from that codebase, you can deploy many times.
- Factor 2. Explicitly declare and isolate dependencies. For example, using a package.json file for a Node.js application lists all the external dependencies.
- Factor 3. Store configuration in the environment: Provide any extra configuration information as environment variable that can be bootstrapped when the application starts.
- Factor 4. Treat backing services as attached resources: Bind services to applications by using attach services (such as DBaaS or load balancers) to an application.
- Factor 5. Strictly separate build and run stages: For example, it is impossible to make changes to the code at run time, since there is no way to propagate those changes back to the build stage.
- Factor 6. Run the app as one or more stateless processes: When you design applications, use multiple processes or services as needed. Avoid dependencies on sticky sessions and keep session data in a persistent store to ensure traffic can be routed to other processes without service disruption.

IBM Training

Cloud-native development methods (cont.)

Cloud-native development introduces the 12-actor app methods and patterns to development:

VII. Port binding: Export services by using port binding.
VIII. Concurrency: Scale out by using the process model.
IX. Disposability: Maximize robustness with fast startup and graceful shutdown.
X. Dev/prod parity: Keep development, staging, and production similar.
XI. Logs: Treat logs as event streams.
XII. Admin processes: Run admin and management tasks as one-off processes.

Introduction to cloud computing

Figure 1-26. Cloud-native development methods (cont.)

Examples of the last six factors:

- Factor 7. Export services by using port binding: Cloud providers offer processes or services with a port for binding and then handle routing traffic to the process over this port automatically. Application code reads the port from the environment and binds to this port.
- Factor 8. Scale out by using the process model: Horizontal scaling of application instances in cloud providers.
- Factor 9. Maximize robustness with fast startup and graceful shutdown: Use a disposable approach to the design of a process in the application. There should be minimal startup actions required. When a process is terminated, it should exit with minimal housekeeping, which improves robustness and responsiveness to horizontal scaling events.
- Factor 10. Keep development, staging, and production similar: This approach enables agile software delivery and continuous integration.
- Factor 11. Treat logs as event streams: Cloud providers have ways to accumulate log data across various components of the application to enable analyzing or exporting to a third-party logging service.

- Factor 12. Run administrator and management tasks as one-off processes: Design tasks that must run once or occasionally as separate components that can be run when needed instead of adding the code directly into another component. For example, if an application must migrate data into a database, place this task into a separate component instead of adding it to the main application code at startup.

IBM Training

Unit summary

- Define cloud computing.
- Describe the characteristics of Cloud.
- Describe the benefits of Cloud and the factors contributing to its growth.
- Describe cloud services models (IaaS, PaaS and SaaS).
- Describe the cloud deployment options (Private, Public, Hybrid).
- Describe cloud-native applications and development methods.
- Explain the Twelve-Factor App methodology.
- Describe the choices that developers have when building cloud applications.

Introduction to cloud computing

Figure 1-27. Unit summary

IBM Training

Review questions

1. **True or false:** An IaaS provides access to the VMs and operating systems.
2. **True or false:** Examples of PaaS are Email and Collaboration systems, and CRM systems.
3. **True or false:** In a SaaS model, customers are responsible for deploying and managing the data and application.
4. Which of the following describe a hybrid cloud:
   a. Developers can develop in multiple development languages and use different run times.
   b. Developers can deploy applications for multiple processor types, such as ARM, Intel (x86), and POWER processors.
   c. Hybrid cloud describes the ability for organizations to use and integrate private clouds and public clouds.
   d. Hybrid cloud describes the ability for developers to deploy applications in containers and in VMs simultaneously.

Introduction to cloud computing

Figure 1-28. Review questions

IBM Training

Review questions (cont.)

5. **True or False:** On-demand resources means that developers must estimate, plan, and provision the environment they require ahead of time.
6. Which of the following reasons is *not* a factor in the growth of cloud computing:
   a. Varying preferences for programming languages.
   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.

Introduction to cloud computing

Figure 1-29. Review questions (cont.)

IBM Training

Review answers

1. **True**.
2. **False**. Email and Collaboration systems and CRMs are examples of SaaS.
3. **False**. Customers are responsible for deploying and managing the data and application in the PaaS model.
4. **C**.
5. **False**. On-demand resources mean that a developer has the resources when they need it.
6. **B**.

Introduction to cloud computing

Figure 1-30. Review answers

IBM Training

References

- What is Cloud Computing?
  https://www.ibm.com/cloud-computing/learn-more/what-is-cloud-computing/
- Cloud Service Models
  https://www.ibm.com/cloud/learn/iaas-paas-saas
- IBM Cloud Docs
  https://cloud.ibm.com/docs/
- Cloud Native
  https://www.ibm.com/cloud/cloud-native
- IBM Cloud Services
  https://www.ibm.com/cloud/products
  https://cloud.ibm.com/docs/home/alldocs
- Cloud Explained with a Pizza Analogy
  https://www.ibm.com/blogs/insights-on-business/government/cloud-explained-with-a-pizza-analogy/
- 12 Factor Application
  https://12factor.net

Introduction to cloud computing

Figure 1-31. References

Unit 2. Getting started with IBM Cloud

Estimated time

02:00

Overview

This unit gets you started with IBM Cloud. It provides an overview of IBM Cloud services and the type of applications you can build on IBM Cloud. It explains how to create and manage cloud apps on IBM Cloud and how to manage users and resources.

IBM Training

Unit objectives

- Describe IBM Cloud.
- Distinguish among the various compute options in IBM Cloud.
- Identify the runtimes and services that IBM Cloud offers.
- Describe IBM Cloud regions, zones, and multi-availability zones.
- Describe the IBM Cloud dashboard, catalog, and documentation features.
- Work with IBM Cloud resources.
- Explain starter kits and Cloud Foundry boilerplates.
- Describe how to manage your IBM Cloud users and resources (optional).
- Explain Identity and Access Management (IAM) and Resource Groups (RGs) (optional).
- Describe how the application route is used to test an application in the browser.
- Bind services to an application in IBM Cloud.
- Describe the environmental variables that are used with IBM Cloud services.
- Explain function as a service.

Getting started with IBM Cloud

Figure 2-1. Unit objectives

2.1. Introduction to IBM Cloud

IBM Training

Introduction to IBM Cloud

Getting started with IBM Cloud

Figure 2-2. Introduction to IBM Cloud

IBM Training

Topics

- Introduction to IBM Cloud
- IBM Cloud compute choices
- IBM Cloud console
- IBM Cloud catalog
- Creating and managing a Cloud Foundry application
- IBM Cloud regions and availability zones (optional)
- Managing your IBM Cloud users and resources (optional)

Getting started with IBM Cloud

Figure 2-3. Topics

IBM Training

Welcome to IBM Cloud

[Welcome to IBM Cloud](https://www.youtube.com/watch?v=VXqbRNwXC2A) on youtube.com

Getting started with IBM Cloud

Figure 2-4. Welcome to IBM Cloud

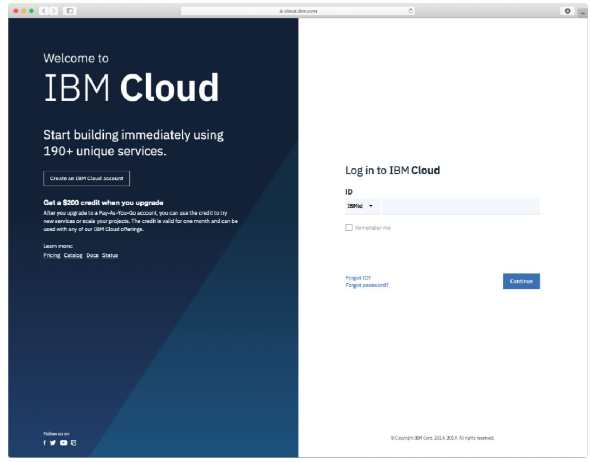
For an introduction to IBM Cloud, see the following YouTube video:

https://www.youtube.com/watch?v=VXqbRNwXC2A

IBM Training

What is IBM Cloud

- An open, standards-based cloud computing platform that helps developers and enterprises rapidly build, deploy, run, and scale applications and services.
- Combines platform as a service (PaaS) with infrastructure as a service (IaaS)
- Includes a catalog of 190+ unique and diverse services.



Getting started with IBM Cloud

Figure 2-5. What is IBM Cloud

The IBM Cloud platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud spins up fast and performs reliably in a tested and supported environment that you can trust. The IBM Cloud platform is an enterprise-grade and full-stack platform that is purpose-built for data-intensive artificial intelligence (AI) workloads and cloud-native application suites that are delivered on a software-defined infrastructure (SDI). It is an open cloud computing platform that combines PaaS with IaaS, and includes a catalog of diverse cloud services, which can be used to build and deploy rapidly business applications or infrastructure.

- PaaS provides developers access to IBM software for integration, security, transaction, 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 (IoT).
- IaaS gives 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.

The IBM Cloud platform is composed of multiple components that work together to provide a consistent and dependable cloud experience:

- A catalog that consists of hundreds of IBM Cloud offerings
- A robust console that serves as the front-end for creating, viewing, and managing your cloud resources.
- An IAM component that securely authenticates users for both platform services and controls access to resources consistently across IBM Cloud.
- A search and tagging mechanism for filtering and identifying your resources.
- An account and billing management system that provides exact usage for pricing plans and secure credit card fraud protection.

IBM Training

What is IBM Cloud (cont.)

Getting started with IBM Cloud

Figure 2-6. What is IBM Cloud (cont.)

IBM Cloud allows developers and enterprises to deploy workloads in over 60 data centers, and now into six regions and 18 availability zones globally with local access, low latency, and certified security:

- *Regions* are geographically separated from one another, and have a physically distinct infrastructure from all other regions and unrelated zones.
- *Zones* are logically isolated data centers within a single campus, each having isolated electrical, mechanical, and network infrastructures. Zones are separated physically from one another. Within a zone, resources are connected by ultra-low latency, high-bandwidth networks.

IBM Cloud offers many choices for you to decide where and how your data and workloads should run. IBM availability zone design provides an easier and more effective way to design and operate applications and databases, making them highly available, fault-tolerant, and scalable.

IBM Training

What can you build with IBM Cloud

- Infrastructure: Hardware that is provided in single or multi-tenancy configurations, such as:
  - Bare Metal Servers, virtual servers, and containers.
  - Object storage.
  - Networking.
- Applications: Programs that developers build in IBM Cloud, such as:
  - Mobile applications.
  - Web applications.
  - APIs.
- Services: Cloud extensions that are hosted by IBM Cloud:
  - Provide functions that are ready-for-use by the application.
  - Predefined services include database, messaging, push notifications for mobile apps, and elastic caching for web apps.

Getting started with IBM Cloud

Figure 2-7. What can you build with IBM Cloud

The following components can be built in IBM Cloud:

- Infrastructure:
  - In IBM Cloud, you can provision infrastructure that can host all kinds of data-intensive workloads.
  - Hardware can be single-tenant (single customer) or multi-tenant (multiple customers sharing the hardware).
  - Hardware is virtualized or software-defined.
- 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 run time.
  - 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 functions that are ready-for-use by the running the code of the application.
  - The predefined services that are provided by IBM Cloud include database, AI, 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 that you might see in a runtime library, or complex business logic that you might see in a business process modeling service or a database.

2.2. IBM Cloud compute choices

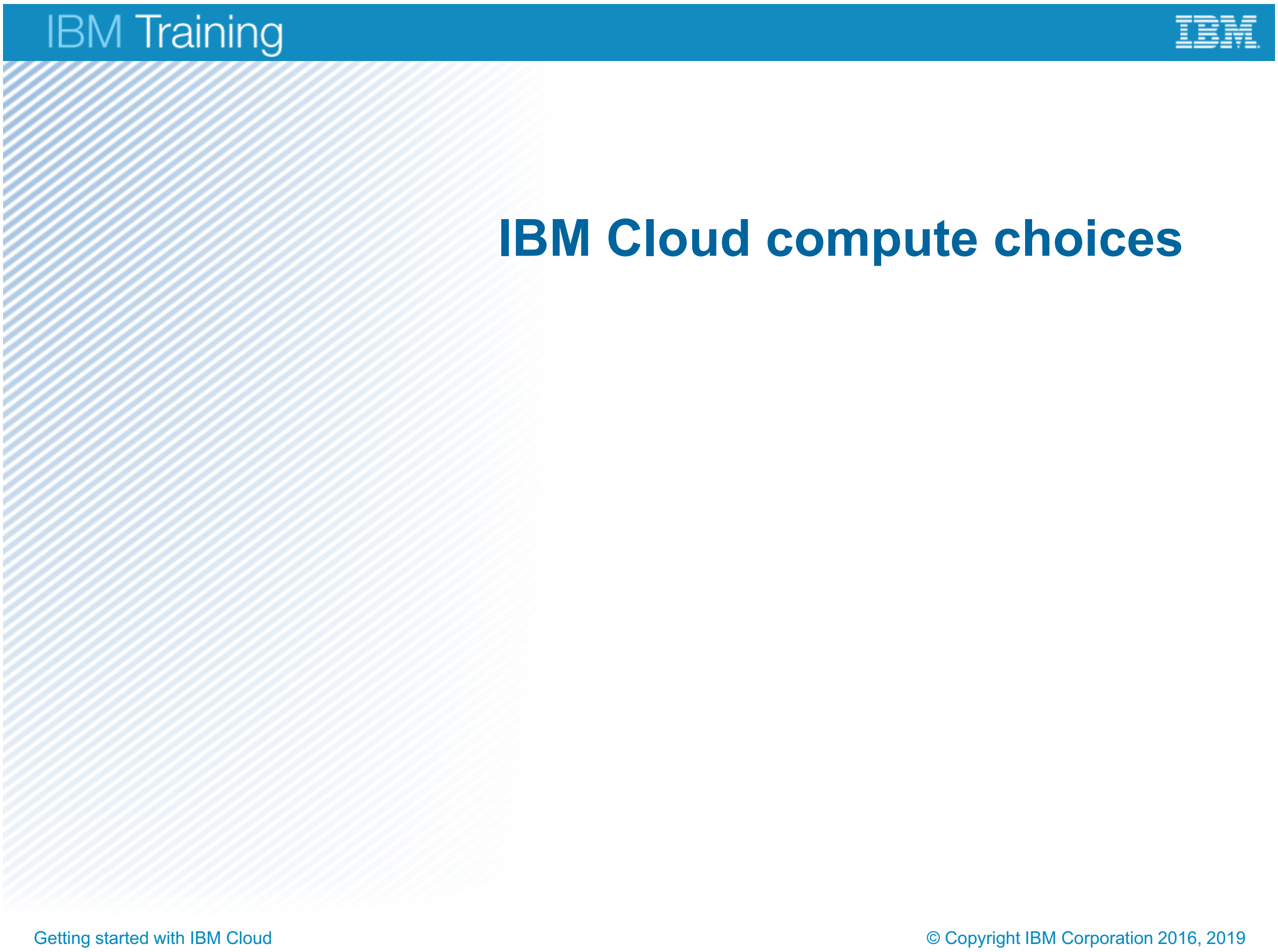


Figure 2-8. IBM Cloud compute choices

IBM Training

Topics

- Introduction to IBM Cloud
- IBM Cloud compute choices
- IBM Cloud console
- IBM Cloud catalog
- Creating and managing a Cloud Foundry application
- IBM Cloud regions and availability zones (optional)
- Managing your IBM Cloud users and resources (optional)

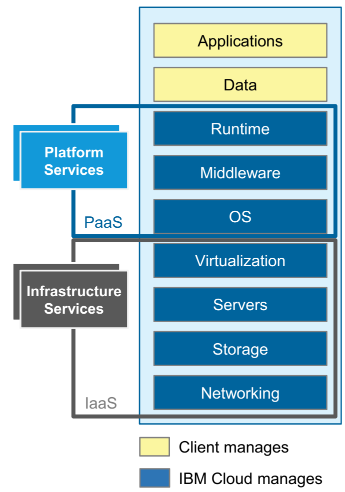
Getting started with IBM Cloud

Figure 2-9. Topics

IBM Training

IBM Cloud service models

- Infrastructure as a Service (IaaS) allows customers to have full control of workloads and their dependencies.
- Platform as a Service (PaaS) allows customers to accelerate building and scaling applications and managing workloads while consuming services.



Getting started with IBM Cloud

Figure 2-10. IBM Cloud service models

IaaS is a cloud computing model in which a vendor provides users access to computing resources, such as servers, storage, and networking. Organizations use their own platforms and applications within a service provider's infrastructure.

PaaS is a cloud computing model that provides users with a cloud environment in which they can develop, manage, and deliver applications. In addition to storage and other computing resources, users can use a suite of prebuilt tools to develop, customize, and test their own applications.

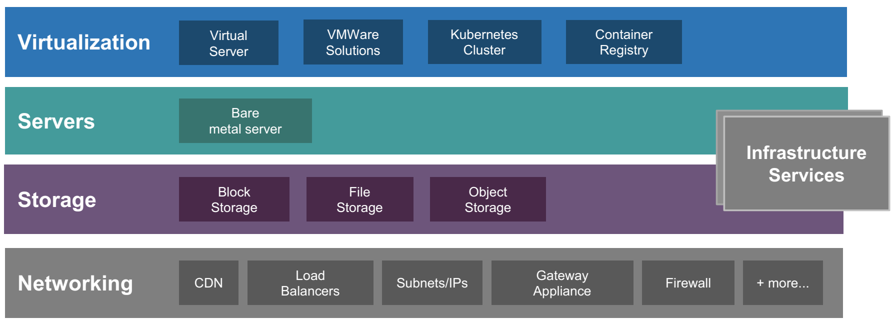
Reference:

https://www.ibm.com/cloud/learn/iaas-paas-saas

IBM Training

Infrastructure as a service from IBM Cloud

- Allows you to deploy a high-performance infrastructure in IBM Cloud Data Center locations around the world.
- Bare metal servers, virtual servers, containers, storage, networking, CDNs, and others.
- Provides services to deploy, access, and manage the infrastructure.



Getting started with IBM Cloud

Figure 2-11. Infrastructure as a service from IBM Cloud

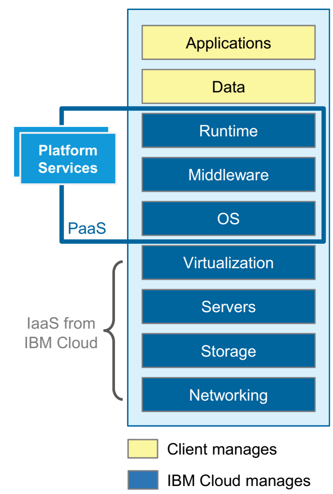
IaaS from IBM Cloud

- IBM Cloud enables you to deploy a high-performance compute and storage infrastructure in over 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 worldwide data centers.
- A catalog of services enables you to deploy, access, and manage the deployed infrastructure.

IBM Training

Platform as a service from IBM Cloud

- Enables you to build, manage, run, and scale applications.
- Provides multiple choices of runtimes:
  - Kubernetes
  - Cloud Foundry
  - OpenWhisk
- Extends services from IBM and Business Partners.
- Provides a scriptable command-line interface (CLI).



Getting started with IBM Cloud

Figure 2-12. Platform as a service from IBM Cloud

As a PaaS provider, IBM Cloud allows you to build, manage, and run applications, such as web, mobile, big data, smart devices, and IoT. IBM Cloud PaaS uses Cloud Foundry, which is an open PaaS 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 Business 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.

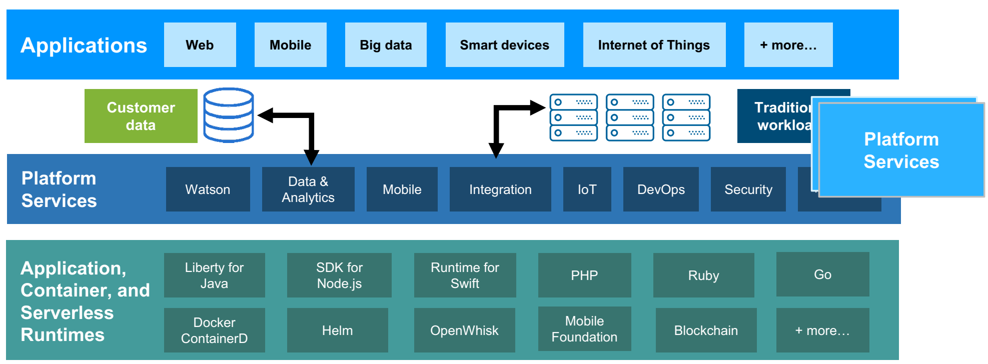
IBM Cloud runs on IBM Cloud data centers locations around the world.

IBM Training

Platform as a service from IBM Cloud (cont.)

The following resources are provided on IBM Cloud:

- Runtimes on which to run applications.
- Services that can be used to build applications.
- Ability to integrate with data and traditional workloads in on-premises systems.
- DevOps capabilities and tools.



Getting started with IBM Cloud

Figure 2-13. Platform as a service from IBM Cloud (cont.)

PaaS from IBM Cloud

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, AI, 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.

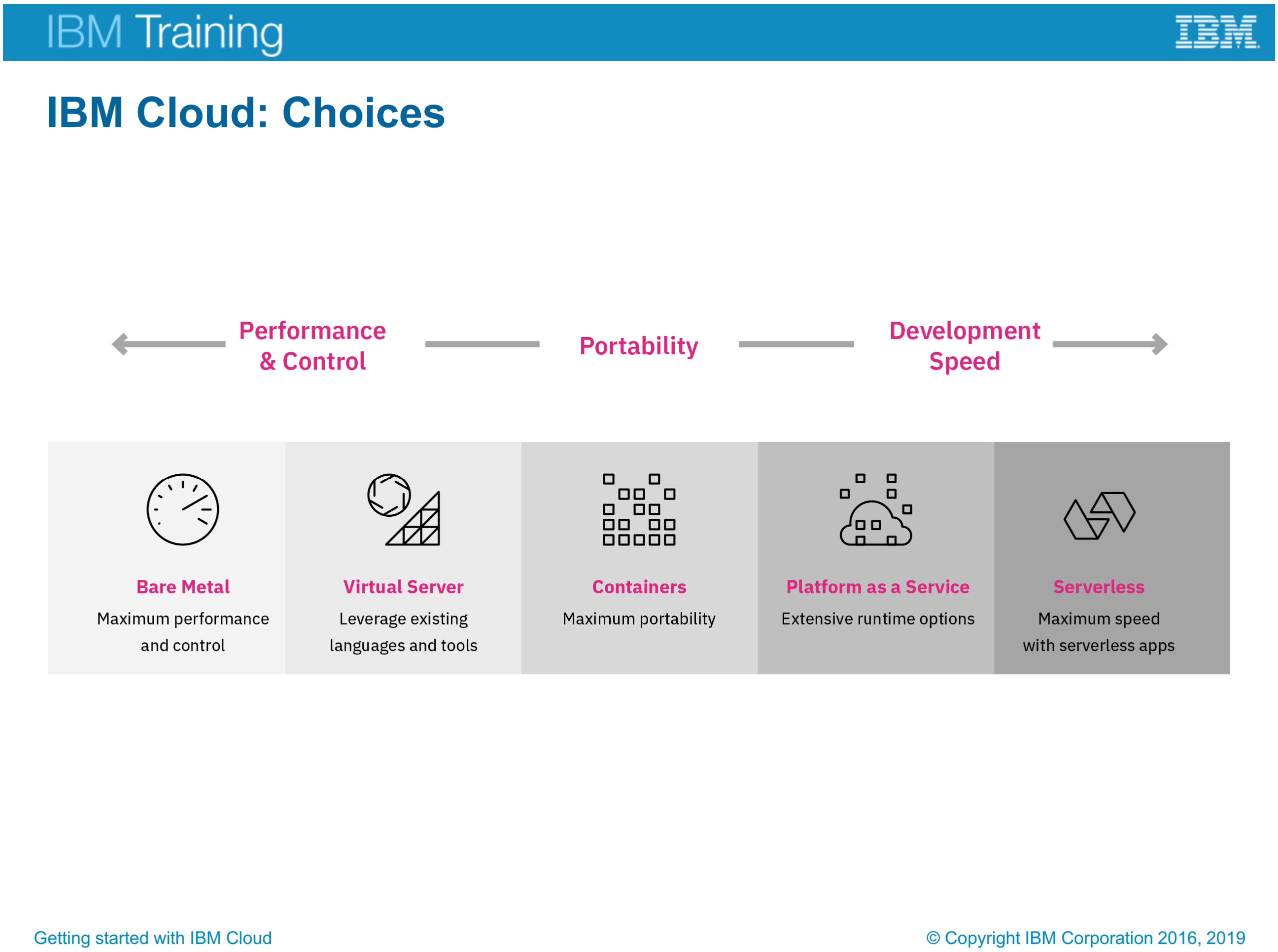


Figure 2-14. IBM Cloud: Choices

With IBM Cloud, developers are given a choice of runtimes on which to run their applications. A *runtime* is a set of computing resources that are used to run an application. IBM provides five main ways to deploy, run, and scale workloads:

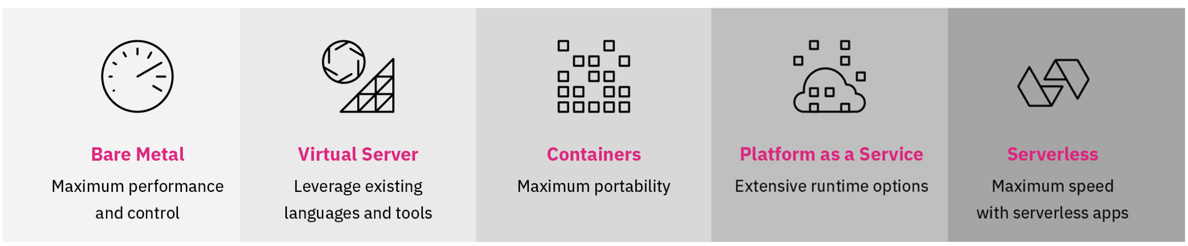
1. Bare metal servers: High-performance cloud servers that can be configured as hourly and monthly options.
2. Virtual servers.
3. Containers.
4. Platforms (Cloud Foundry).
5. Serverless.

IBM Training

IBM Cloud: Choice of compute

IBM Cloud provides developers with multiple choices to deploy and run workloads:

- Bare metal servers
- Virtual servers
- Containers (IBM Cloud Kubernetes Services)



Getting started with IBM Cloud

Figure 2-15. IBM Cloud: Choice of compute

- Bare metal servers: bare metal servers provide users with sole access to the entire server. Unlike a virtual server with multiple tenants, the bare metal server is single tenant and offered without a hypervisor, which eliminates the “noisy neighbor” effect and any performance “tax” from the hypervisor. Bare metal servers can be acquired in a preconfigured form or custom-configured to exact specifications.
- Virtual servers: Virtual servers are scalable and come with dedicated core and memory allocations that can be added in minutes, with access to features like image templates. The hypervisor is fully managed by IBM Cloud, and developers can perform configuration and management tasks by using both the IBM Cloud customer portal and the API. Virtual servers are deployed to the same VLANs as physical servers, allowing developers to spread workloads across virtual servers and bare metal servers while maintaining interoperability. Virtual servers are fully customizable with options to scale up as your compute needs grow.

- Containers: IBM Cloud Kubernetes Service is a managed container service for the rapid delivery of applications that can bind to advanced services such as IBM Watson and blockchain. As a certified Kubernetes provider, IBM Cloud Kubernetes Service provides intelligent scheduling, self-healing, horizontal scaling, service discovery and load balancing, automated rollouts and rollbacks, and secret and configuration management. Kubernetes Service also has advanced capabilities around simplified cluster management, container security and isolation policies, the ability to design your own cluster, and integrated operational tools for consistency in deployment.

Compute (bare metal, virtual servers, and containers) allow the developers to have total control of the compute and platform that deploys and runs the workloads with granular control over the scalability, customization, and management.

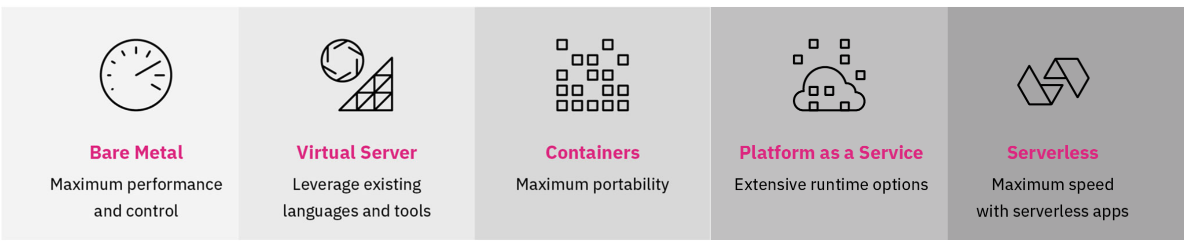
Note

The containers definition and the difference between containers and virtual machines (VMs) are explained in later units.

IBM Training

IBM Cloud: Choice of runtime

- IBM Cloud provides developers with two choices to bootstrap and run applications:
  - IBM Cloud Foundry
  - IBM Cloud Functions
- Runtimes focus on accelerating development and deployment of 12-factor apps.



Getting started with IBM Cloud

Figure 2-16. IBM Cloud: Choice of runtime

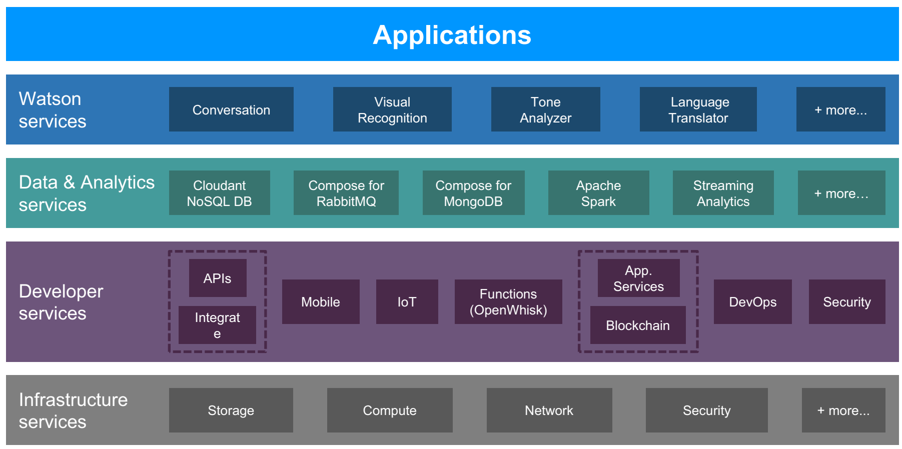
- Platforms (Cloud Foundry): Cloud Foundry is the premier industry standard PaaS that ensures the fastest, easiest, and most reliable deployment of cloud-native applications. Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services, which result in quick, consistent, and reliable iterating of applications. 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.
- Serverless: Based on Apache OpenWhisk, IBM Cloud Functions is a polyglot function as a service (FaaS) programming platform for developing lightweight code that scales and runs on demand. IBM Cloud Functions provides access to the Apache OpenWhisk infrastructure in which anyone can contribute their action code as building blocks to the expanding repository.

Runtimes (Cloud Foundry and Cloud Functions) allow the developers to focus on development and delegate management to the cloud provider.

IBM Training

IBM Cloud: Services

Pre-built services provide building blocks for feature-rich applications.



Getting started with IBM Cloud

Figure 2-17. IBM Cloud: Services

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 AI to your application with speech, vision, and natural language processing (NLP) 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 a mobile back-end infrastructure to build, monitor, and test mobile apps.
- IoT: Communicate with connected devices, sensors, and gateways.
- Functions: Run in response to incoming events (based on Apache OpenWhisk).

- Application services: Many application services, such as IBM 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 on which your application runs.

2.3. IBM Cloud console

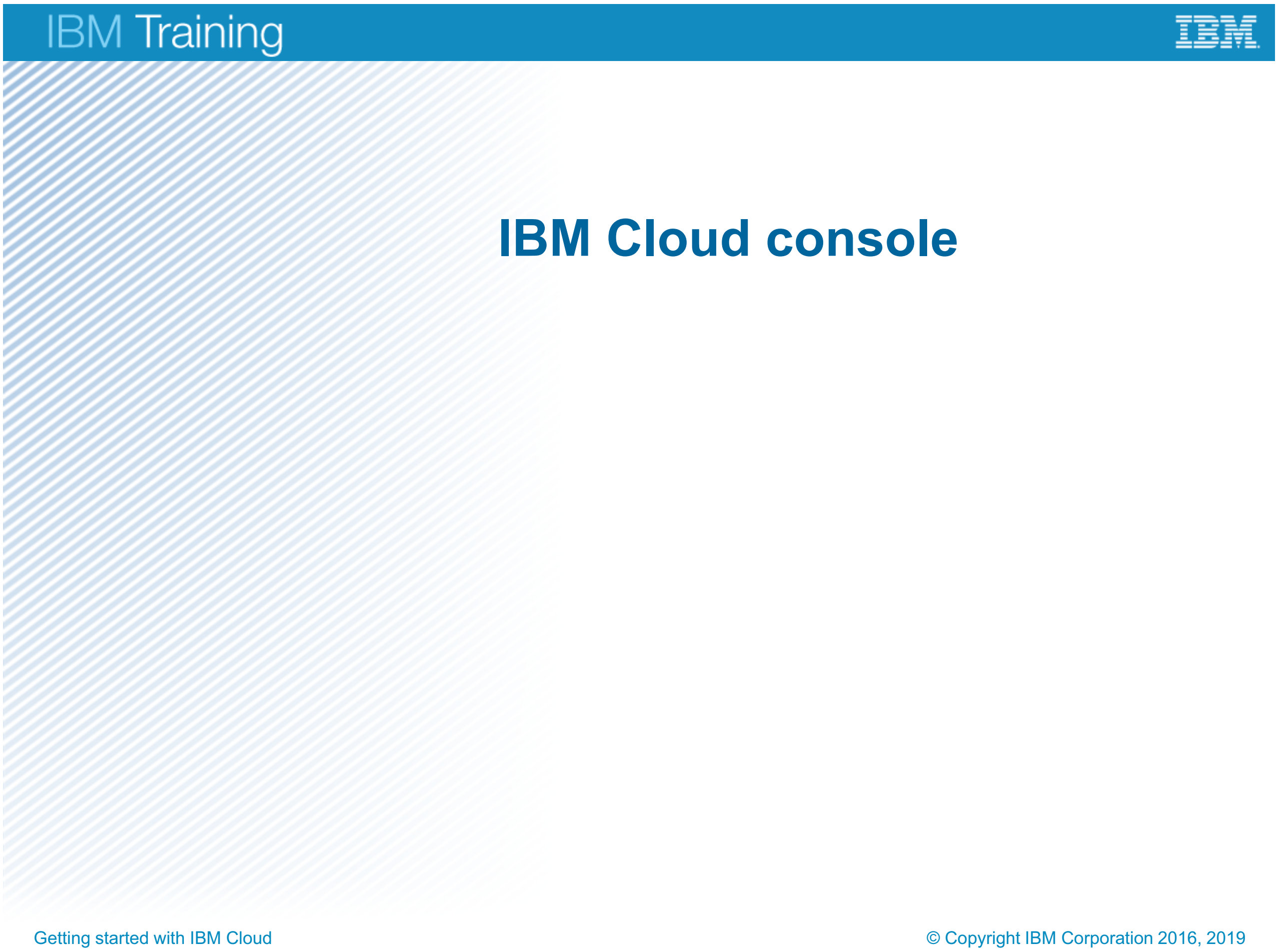


Figure 2-18. IBM Cloud console

IBM Training IBM

Topics

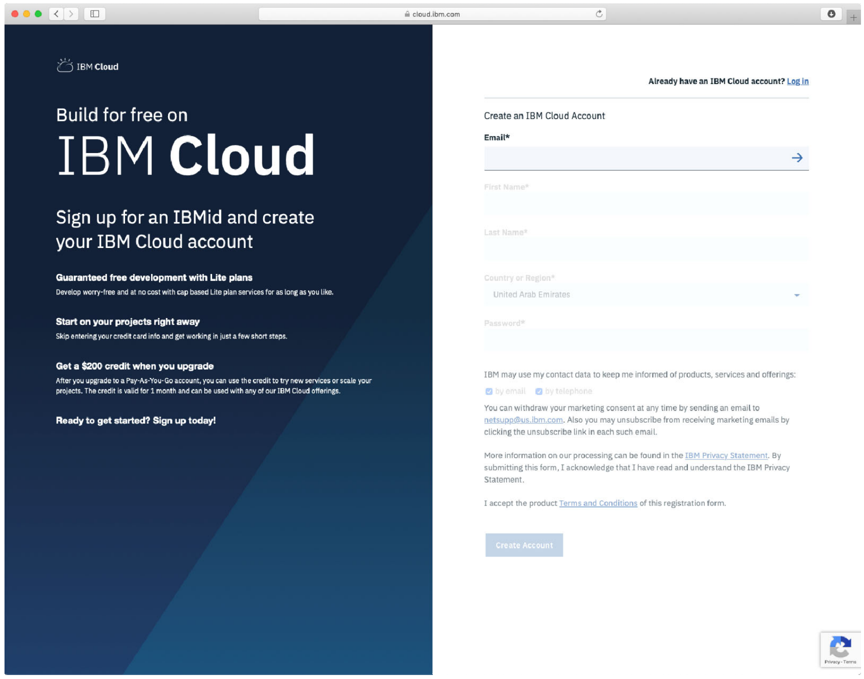
- Introduction to IBM Cloud
- IBM Cloud compute choices
- IBM Cloud console
- IBM Cloud catalog
- Creating and managing a Cloud Foundry application
- IBM Cloud regions and availability zones (optional)
- Managing your IBM Cloud users and resources (optional)

Getting started with IBM Cloud

Figure 2-19. Topics

IBM Training

IBM Cloud: Signup



Getting started with IBM Cloud

Figure 2-20. IBM Cloud: Signup

The IBM Cloud Dashboard console is a user interface that helps you manage all your IBM Cloud resources. When you access the dashboard, you can create a free account, log in, access documentation, access the catalog, view pricing information, get support, or check the status of IBM Cloud components. After you log in, the menu bar contains a Menu icon and more links.

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

IBM Cloud Lite account

You can register for a free IBM Cloud Lite account. For more information about an IBM Cloud Lite account, see the following video: https://www.youtube.com/watch?v=0rMYXcbpHbI

For more information about IBM Cloud Lite accounts, see the following resources:

https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-pricing

https://cloud.ibm.com/docs/account?topic=account-accounts

https://cloud.ibm.com/docs/overview?topic=overview-quickstart_lite

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

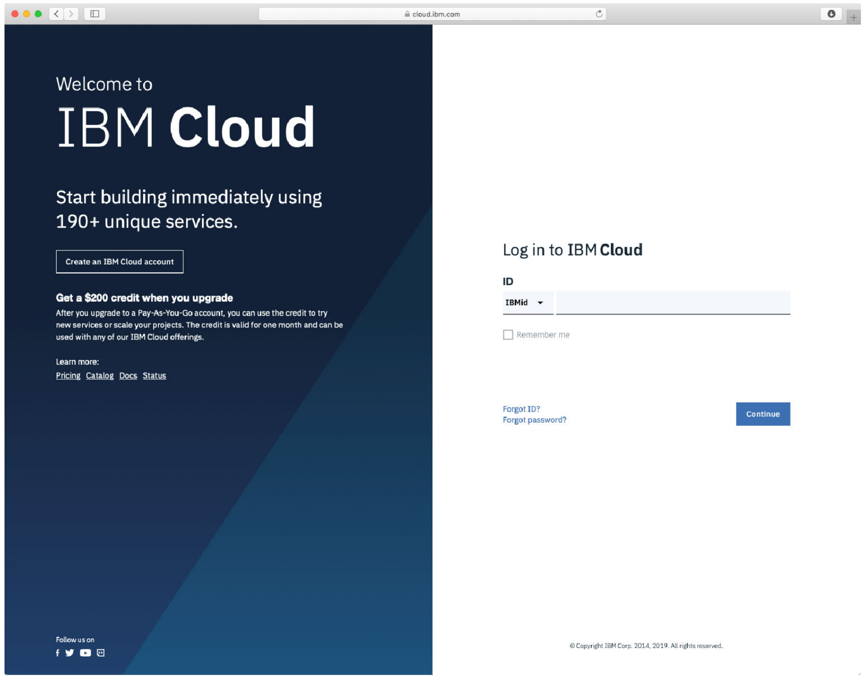
- The account is free (no credit card is required).
- The account never expires.
- You receive 256 MB of memory for your Cloud Foundry applications.
- You have access to specific services that are tagged as Lite.

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 resources 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.

IBM Training

IBM Cloud: Login



Getting started with IBM Cloud

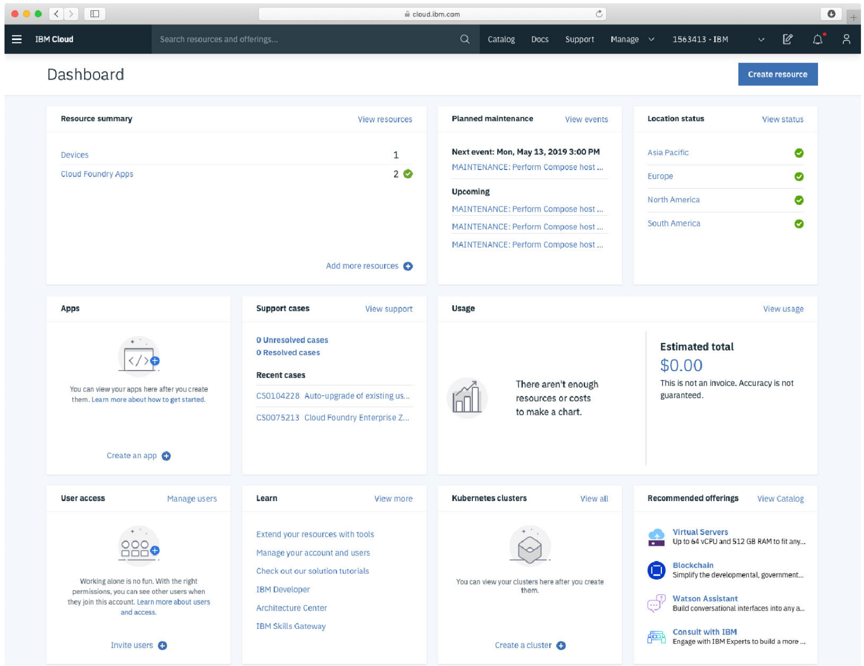
© Copyright IBM Corporation 2016, 2019

Figure 2-21. IBM Cloud: Login

IBM Cloud provides several ways to log in: Either you log in by using the user interface and providing your credentials to access the IBM Cloud dashboard or you use the CLI. You can also use the CLI to perform other actions, such as deploying applications and checking available services.

IBM Training

IBM Cloud: Dashboard



Getting started with IBM Cloud

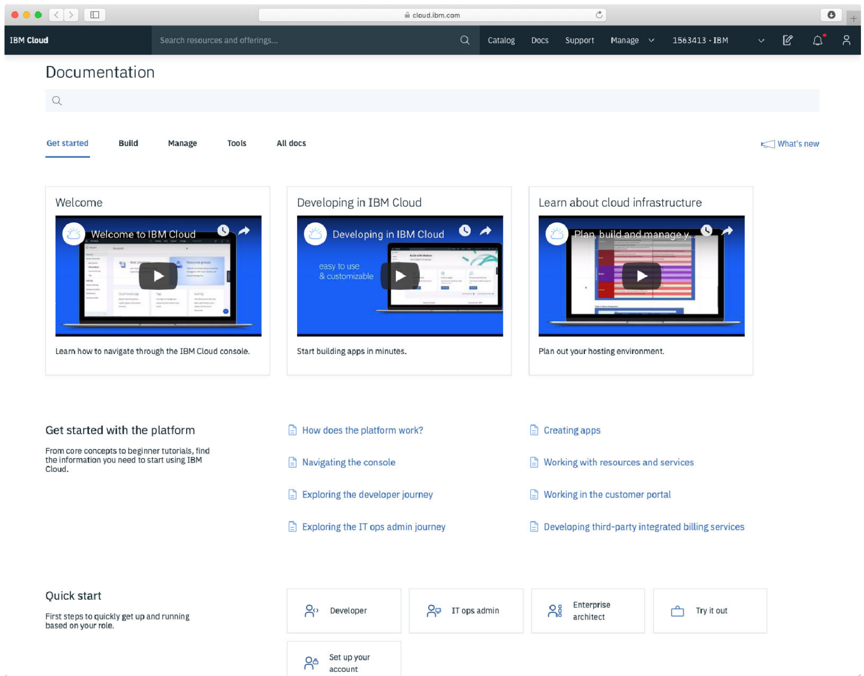
Figure 2-22. IBM Cloud: Dashboard

When you log in to IBM Cloud, the first page that you see is the dashboard, which shows widgets that summarize the status of your account. From here, you can manage your resources. Click the **Menu** icon and select **Resource List** to view all the existing resources in your account. You can then take the following actions:

- Use the **Catalog** link to create resources.
- Use the **Docs** link to access useful information about IBM Cloud.
- Use the **Support** link to access the Support Center.
- From the **Manage** menu, you can access your account, billing and usage, and IAM options.
- Click the **Cost estimator** icon to open the cost estimator.
- Click the **Notifications** icon to access announcements and planned and unplanned events.

IBM Training

IBM Cloud: Docs



Getting started with IBM Cloud

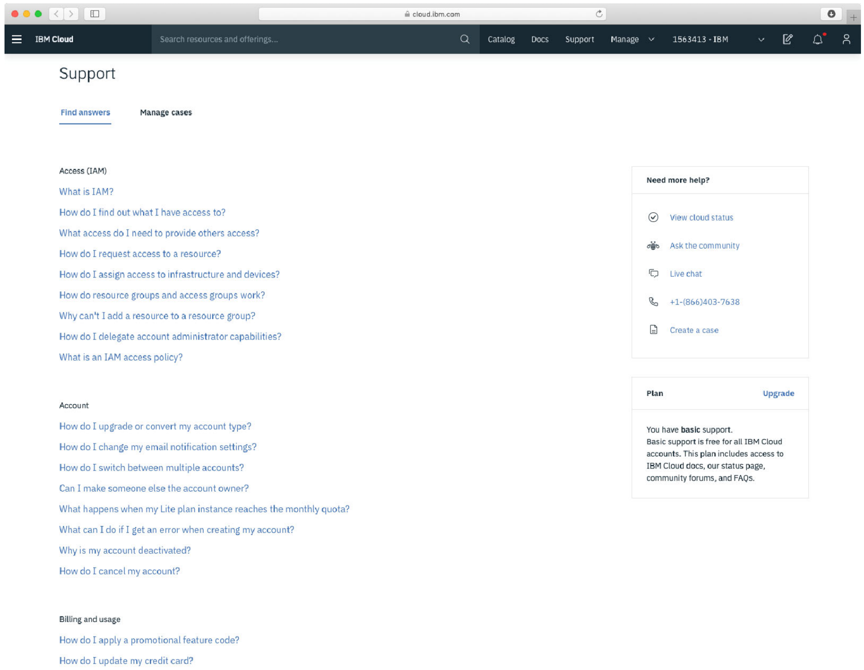
Figure 2-23. IBM Cloud: Docs

The IBM Cloud Docs section help developers get started with IBM Cloud by providing the following items:

- Service descriptions
- Use cases
- Solution tutorials
- Architecture Center articles
- Garage Method assets
- IBM Developer libraries

IBM Training

IBM Cloud: Support



Getting started with IBM Cloud

Figure 2-24. IBM Cloud: Support

The IBM Cloud Support section help developers get the right support and open ticket cases with IBM.

Developers can also browse FAQs, go to community-based forums, have a live chat with support, or even contact IBM directly over many channels to get support.

2.4. IBM Cloud catalog

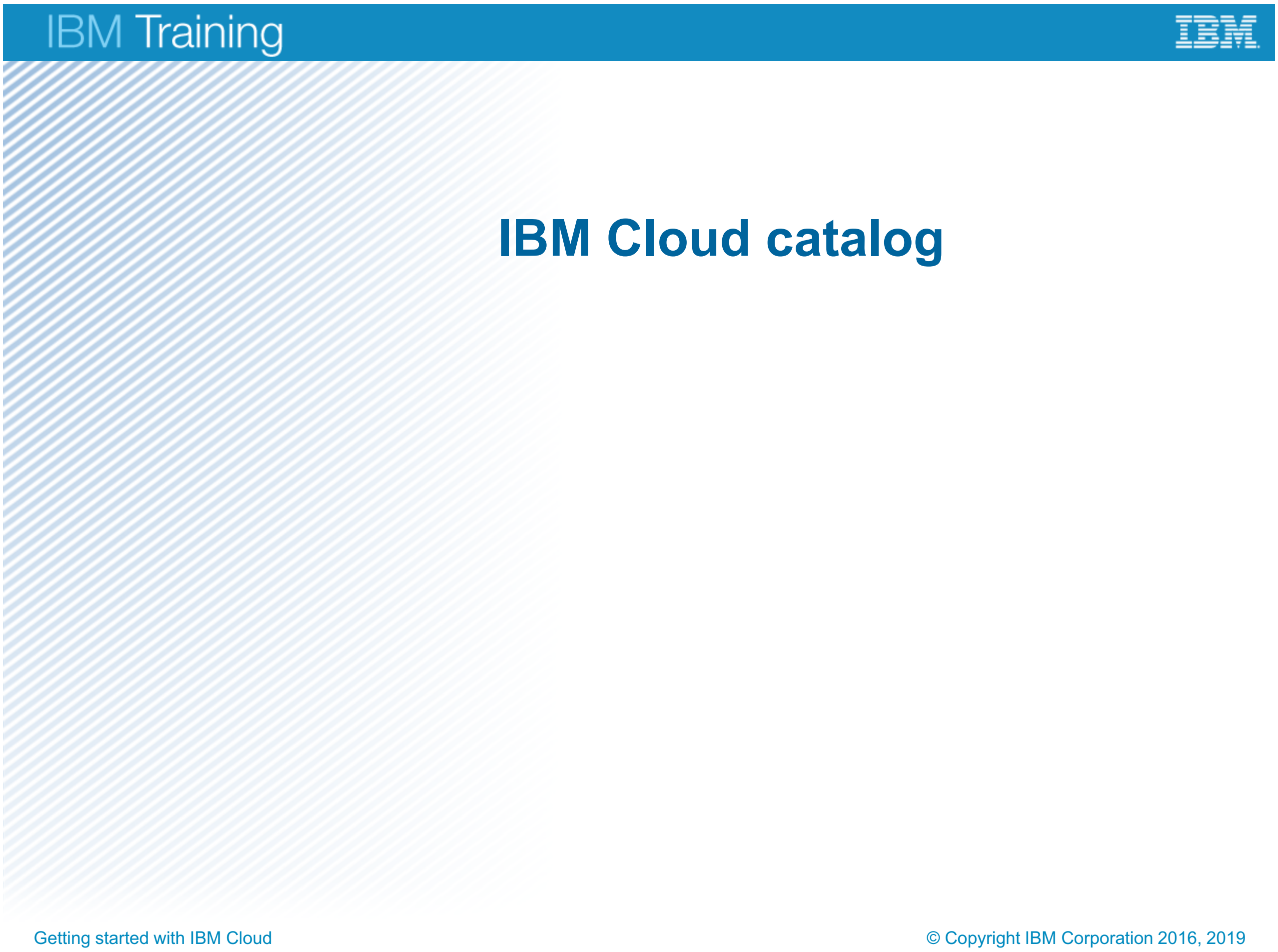


Figure 2-25. IBM Cloud catalog

IBM Training

Topics

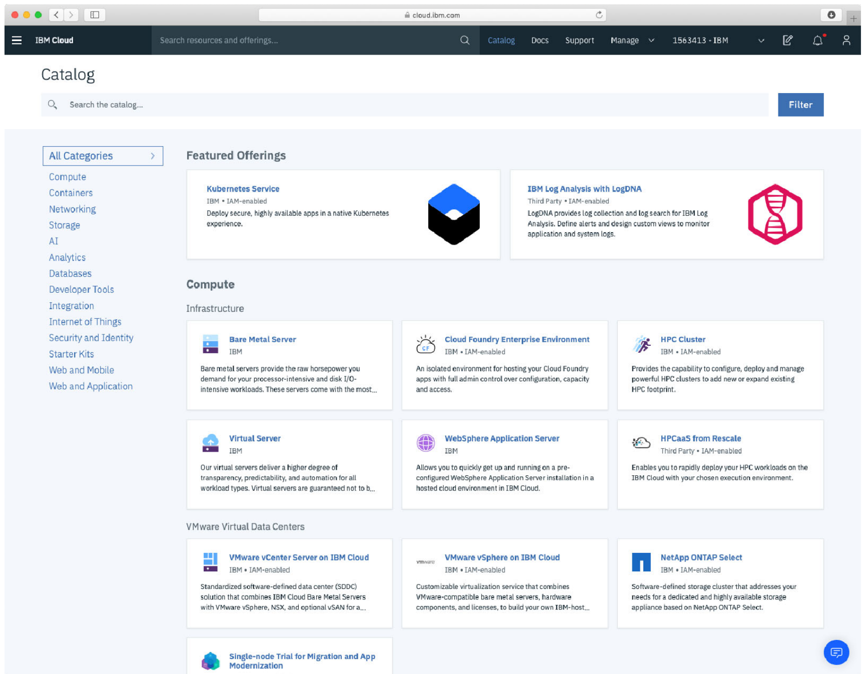
- Introduction to IBM Cloud
- IBM Cloud compute choices
- IBM Cloud console
- IBM Cloud catalog
- Creating and managing a Cloud Foundry application
- IBM Cloud regions and availability zones (optional)
- Managing your IBM Cloud users and resources (optional)

Getting started with IBM Cloud

Figure 2-26. Topics

IBM Training

IBM Cloud: Catalog



Getting started with IBM Cloud

Figure 2-27. IBM Cloud: Catalog

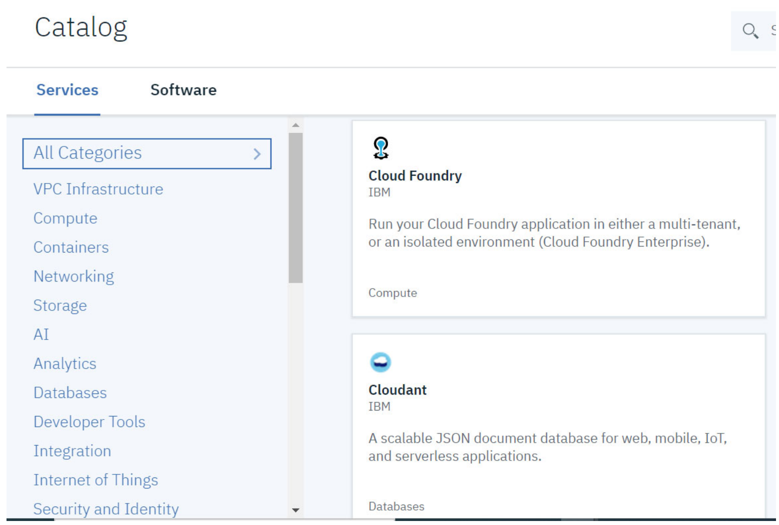
One of the main parts of the IBM Cloud console is the catalog that presents IBM Cloud offerings and services. The IBM Cloud catalog contains more than 190 unique services. You can check the current list of services in the catalog at the following link:

https://cloud.ibm.com/catalog

IBM Training

Working with resources

- A resource is anything that can be created, managed, and contained from the list of services, run times, and templates that are available on the IBM Cloud catalog.
- Some examples include:
  - Cloud Foundry Apps
  - Service instances
  - Container clusters
  - Storage volumes
  - Virtual servers



Getting started with IBM Cloud

Figure 2-28. Working with resources

IBM Cloud offerings are presented in the catalog as resources under different categories that are explained in the rest of this section. Resources include services, infrastructure, and templates in IBM Cloud catalog.

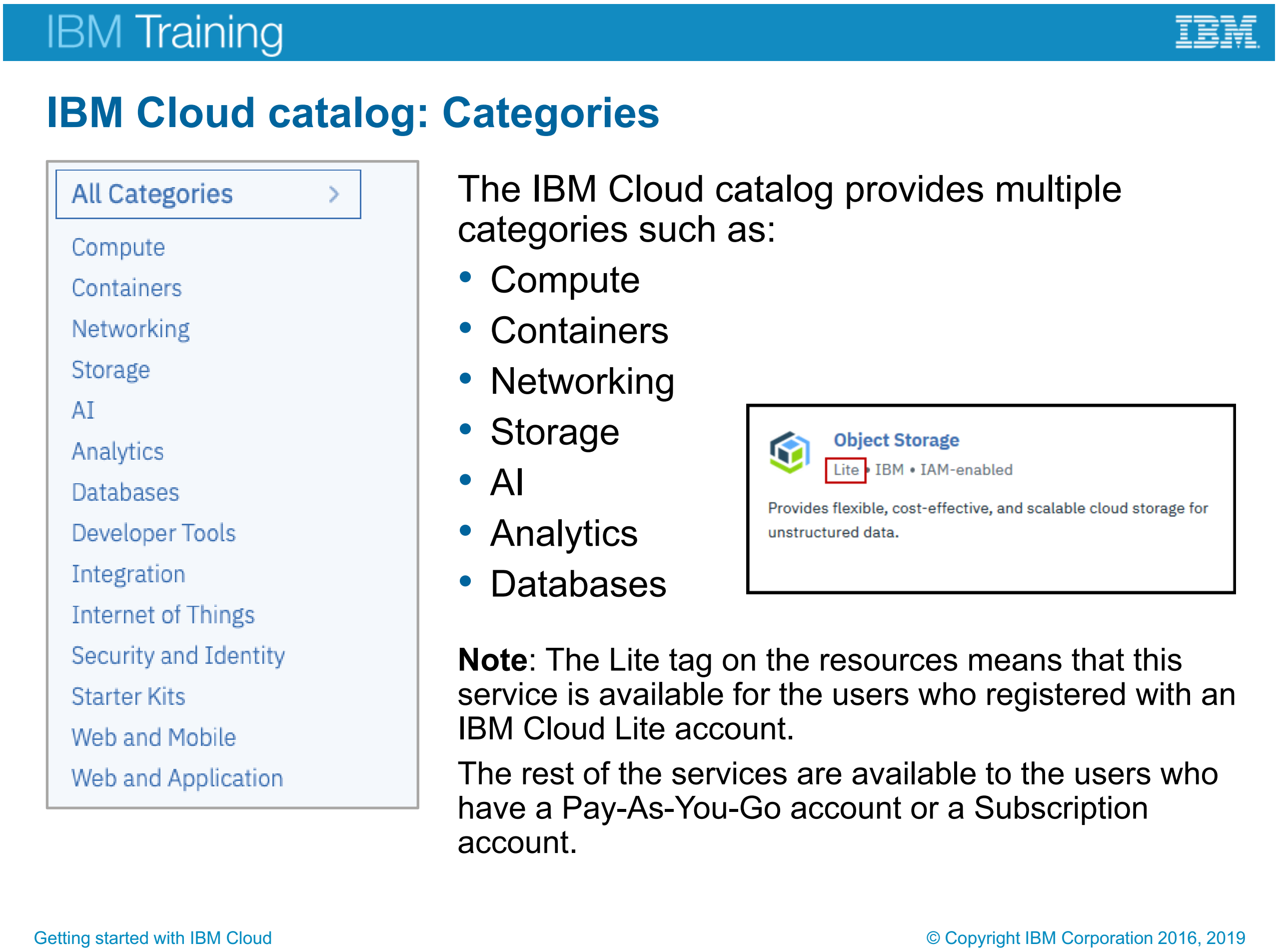


Figure 2-29. IBM Cloud catalog: Categories

The IBM Cloud catalog displays your IBM Cloud resources worldwide. Resources in the catalog can include services, infrastructure, and templates. You can use the IBM Cloud CLI to search the catalog, view details about catalog resources, and manage the visibility of your catalog resources.

To create a resource, click **Create** from your resource list. The catalog window appears. When you select a tile from the catalog, you can see where the resource is available. Not every resource that is listed in the catalog is available in every region.

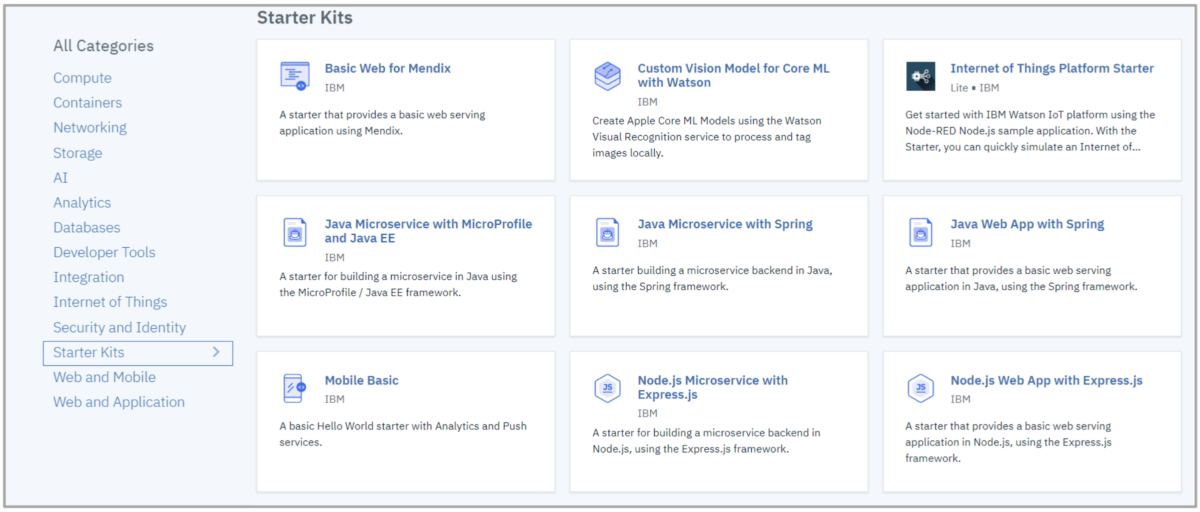
After you click the tile for the resource that you want create, you can select the location in which you want to deploy:

- For Cloud Foundry resources, you can select a specific region and then you select the org and space to which you assign to a service instance.
- For resources that are managed by IBM Cloud IAM, you select a location in which to deploy. Then, you select an Resource Group to which to assign the service instance.

IBM Training

IBM Cloud catalog: Starter kits

A *starter kit* contains a sample application and its associated run time, resource environment, and predefined services.



Getting started with IBM Cloud

Figure 2-30. IBM Cloud catalog: Starter kits

- A *starter kit* is a production-ready pattern that can be integrated with a set of services to generate a production-ready asset that can be deployed directly into a DevOps pipeline and a Kubernetes cluster. Starter kits are great for dynamically assembling a skeleton production application in the language of your choice that is ready for cloud deployment.
- A starter kit contains metadata that describes what the kit is and does. It also contains information that informs IBM Cloud what to produce. The output is production-ready and can be iterated on for further enhancements based on IBM Cloud best practices. Starter kit content is not as complex as a demonstration and not as trivial as a snippet or sample. Apps are dynamically created based on the developer's requirements.
- Each starter kit includes a language, a framework, and a pattern for a specific use case. You can reuse code rather than reinvent it. If a starter kit requires specific services, auto-provisioned services are available so that instances for those services are automatically created when you create your app.

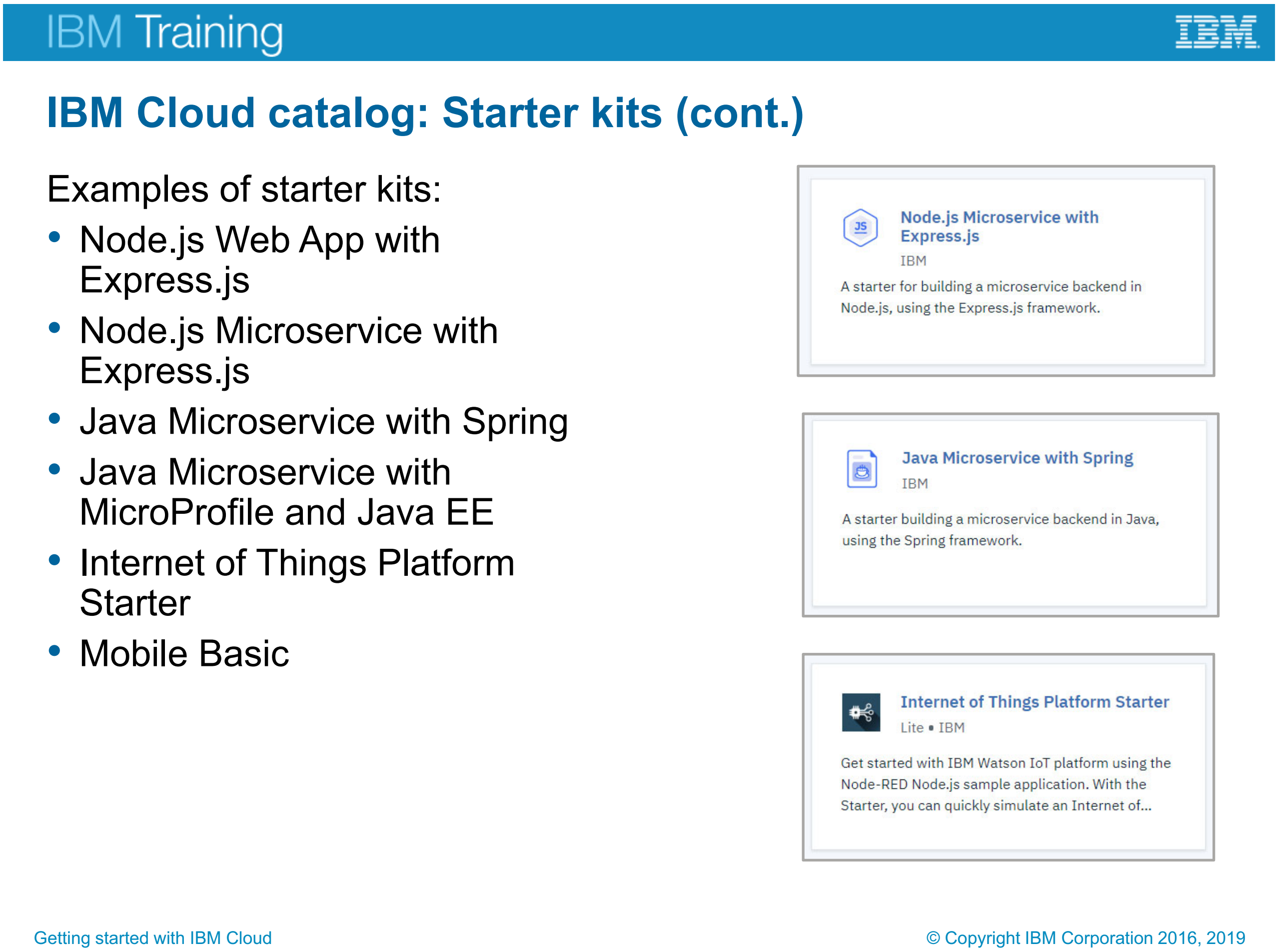


Figure 2-31. IBM Cloud catalog: Starter kits (cont.)

Examples of starter kits:

- Node.js Web App with Express.js

This starter kit comes pre-configured as a web app with Express.js that uses the Node.js run time.

- Node.js Microservice with Express.js

This starter kit comes pre-configured as a microservice with Express.js. It allows the user to use the IBM Cloud Developer Tools CLI to run and debug locally, then deploy to Kubernetes, Cloud Foundry, or a DevOps Pipeline.

- Java Microservice with Spring

This starter kit comes pre-configured as a Java microservice application that uses Spring Boot. It allows the user to use the IBM Cloud Developer Tools CLI to deploy to Kubernetes, Cloud Foundry, or a DevOps Pipeline.

- Java Microservice with MicroProfile and Java EE

This starter kit comes pre-configured as a Java microservice application with the needed tools to set up your development environment and deploy to Kubernetes through a DevOps Toolchain.

- Internet of Things Platform Starter

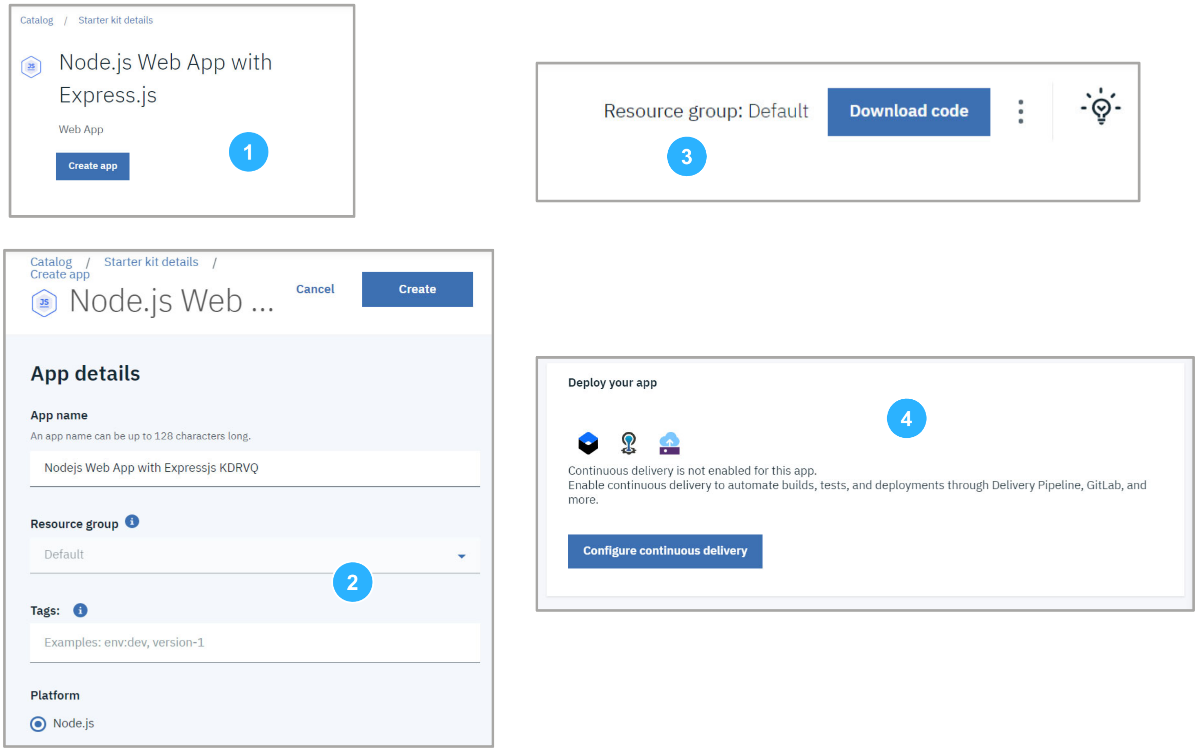
This starter kit includes the IBM Watson IoT Platform that uses the Node-RED Node.js sample application. With this starter kit, the user can quickly simulate an IoT device, create cards, generate data, and begin analyzing and displaying data in the Watson IoT Platform dashboard.

- Mobile Basic

This Mobile starter app pre-provisions push services, in addition to giving a code for the mobile app as a starting point.

IBM Training

IBM Cloud catalog: Creating a starter kit



Getting started with IBM Cloud

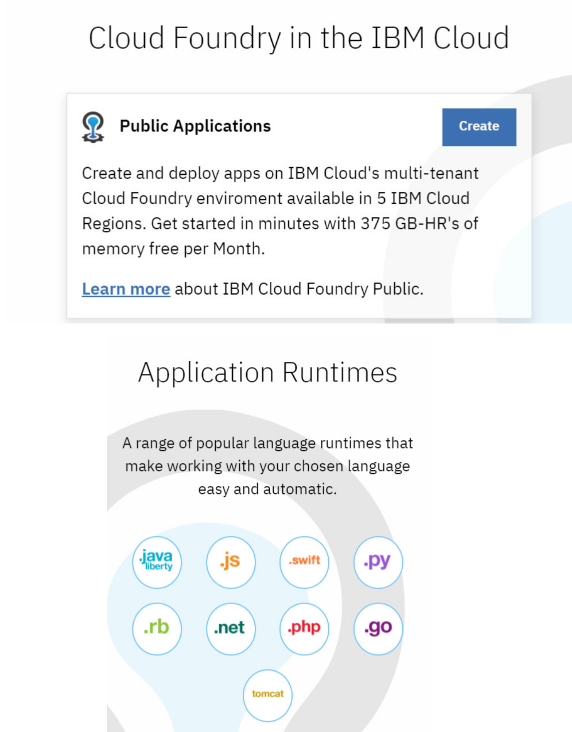
Figure 2-32. IBM Cloud catalog: Creating a starter kit

Here is an example of the Node.js Web App with Express.js starter kit:

1. Select the **Node.js Web App with Express.js** starter kit from the catalog, then click **Create app**.
2. Fill the fields that are required in the form as the app name and the resource group, which are set to **Default**. Then, click **Create**.
3. You are redirected to the starter kit details page. You can download the code by clicking **Download code**.
4. There is the option of deploying this application on the Cloud Foundry runtime, IBM Kubernetes Service, or Virtual Server, and then configure the continuous delivery pipeline.
5. The starter kit also can be connected to several services.

IBM Training

IBM Cloud catalog: Cloud Foundry Apps



IBM Cloud provides Cloud Foundry runtimes that support different programming languages, such as

- Liberty for Java
- SDK for Node.js
- ASP.NET core
- Runtime for Swift

Getting started with IBM Cloud

Figure 2-33. IBM Cloud catalog: Cloud Foundry Apps

IBM Cloud provides Cloud Foundry runtimes that are used to deploy applications of the corresponding language on top of the runtime.

What is Cloud Foundry

Cloud Foundry is the premier industry standard Platform-as-a-Service (PaaS) that ensures the fastest, easiest, and most reliable deployment of cloud-native applications. Cloud Foundry ensures that the build and deployment aspects of coding remain carefully coordinated with any attached services, which result in quick, consistent, and reliable iteration of applications.

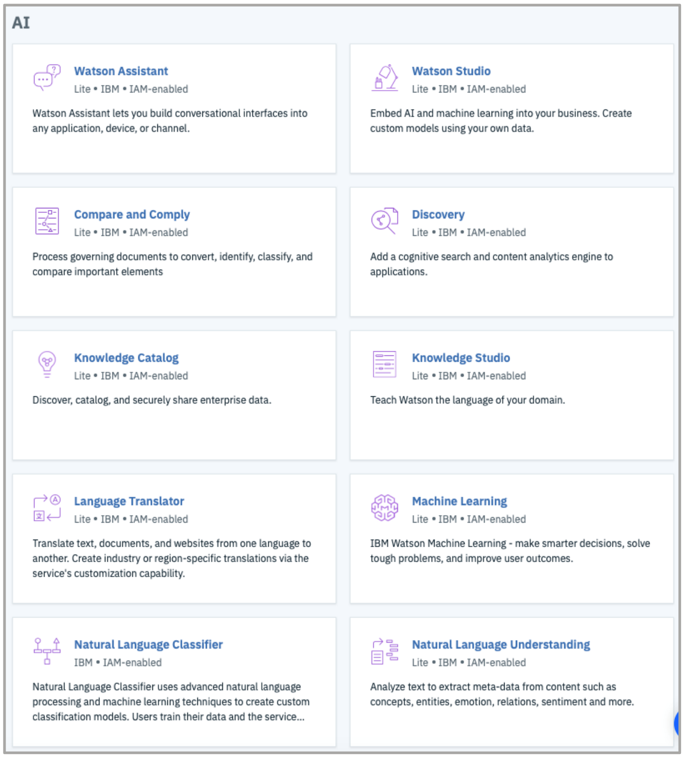
Key benefits of Cloud Foundry:

- Choose your own language: IBM Cloud Foundry includes run times for Java, Node.js, PHP, Python, Ruby, Swift, and Go. Cloud Foundry community build packs are also available. Combined with DevOps services, the application runtimes enable a delivery pipeline that automates much of the iterative development process.
- Fault tolerant: Runtimes facilitate developing applications as stateless processes that quickly start and stop, replicate if an instance fails, and duplicate if sustained or increased performance requires.

- Extend apps with services: Run times link IBM Cloud services to applications as endpoints, giving any instance of an application embedded knowledge of how to manage relevant calls and data. In fact, runtimes manage all linked resources this way: SDKs, APIs (whether made available as cloud services or exposed from within a traditional enterprise as custom services), and also applications themselves when used as resources by other applications.
- Access control: Fine-grained assignment and distribution of compute capacity to development teams.
- Automatic placement: Applications are automatically placed across multiple data center Pods for maximum reliability.
- Automatic Health Management: Crashing applications restart automatically.
- Automatic routing: Internet reachable routes are automatically created for your applications.
- High availability: Supports full high availability for high application availability.
- Automatic deployment scaling: The Auto-Scaling for IBM Cloud service enables you to increase or decrease automatically the compute capacity of your application to adjust rapidly to dynamic loading needs.

IBM Training

IBM Cloud catalog: AI



IBM Cloud provides access to dozens of AI services to enrich applications with cognitive features such as:

- Chat bots
- Machine learning
- Language translator
- Natural language understanding

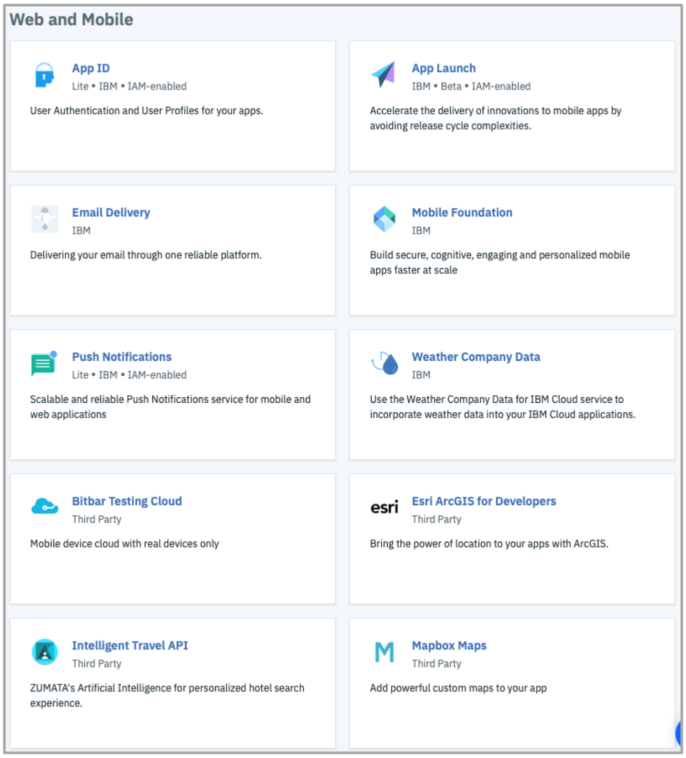
Getting started with IBM Cloud

Figure 2-34. IBM Cloud catalog: AI

IBM Cloud provides various AI services that are built on top of IBM Watson, which is the 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 Training

IBM Cloud catalog: Web and mobile



IBM Cloud offers a set of services that can ease the common tasks that are needed in building a mobile app.

These services take the drudgery of routine tasks away from the app developer and provide compact SDKs with which the developer can interact.

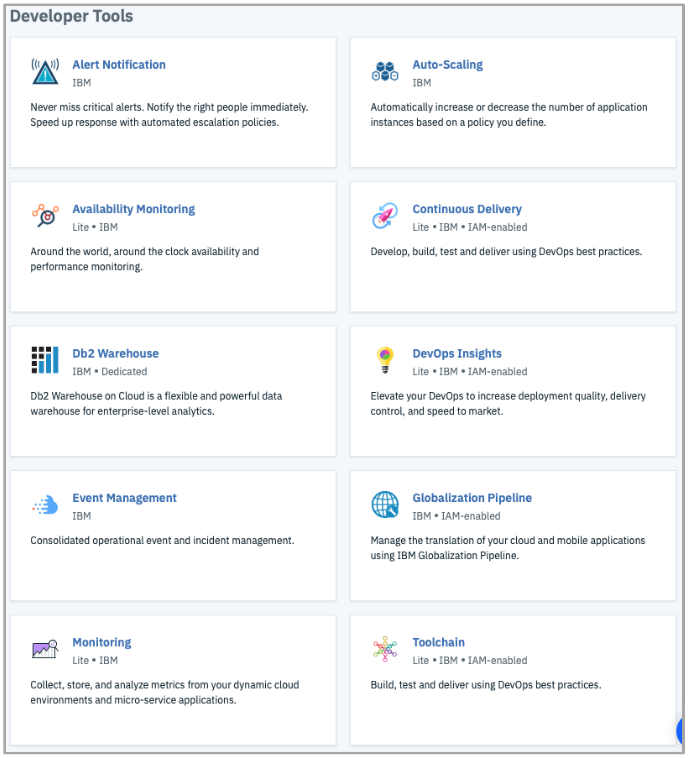
Getting started with IBM Cloud

Figure 2-35. IBM Cloud catalog: Web and mobile

IBM Cloud Mobile Foundation provides an integrated set of back-end capabilities for mobile, web, and progressive web apps (PWAs). Developers can choose to use front-end tools or frameworks of their choice and leverage the rich set of back ends that is provided by the Cloud Mobile Foundation service. Cloud Mobile Foundation SDKs are available for Cordova, iOS, Android, Xamarin, Windows 10, React Native, and Mobile Web.

IBM Training

IBM Cloud catalog: Developer tools



IBM Cloud provides developer tools to automate the tasks of developing and deploying applications.

Most of the services that are available in this section are DevOps related.

Getting started with IBM Cloud

Figure 2-36. IBM Cloud catalog: Developer tools

IBM Cloud Developer tools provide developer toolchains and APIs to help automate the tasks of developing and deploying your app. You can perform DevOps manually with simple apps, but the need for automation increases quickly as app complexity increases, and toolchain automation is necessary for continuous delivery.

The core component of a DevOps toolchain is a version control repository like GitHub. More tools might include backlog tracking, delivery pipelines, an integrated development environment (IDE), and monitoring like IBM Cloud DevOps Insights.

Examples of such tools and processes are the core set of tools that IBM Cloud Continuous Delivery provides for any DevOps toolchain: Git Repos and Issue Tracking, Delivery Pipeline, and Eclipse Orion Web IDE. Git Repos and Issue Tracking is based on the GitLab Community Edition, and offers planning boards and source code collaboration through merge requests. The Delivery Pipeline orchestrates build, test, and deployment jobs across multiple environments as changes progress from the developer to production. Applications can be deployed in minutes to the Cloud Foundry environment or to a Kubernetes cluster on IBM Cloud, to either public or private clouds.

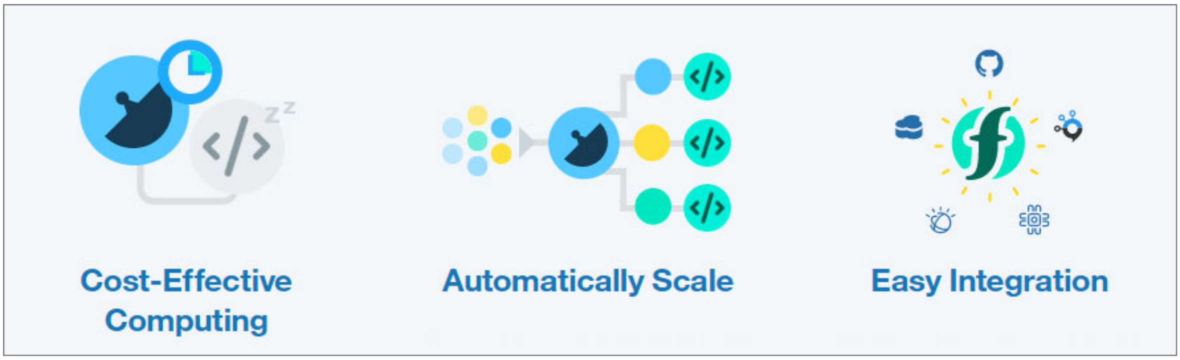
An open toolchain integrates more tools around Cloud Continuous Delivery, such as Slack, Atlassian JIRA, Sonatype Nexus, JFrog Artifactory, Sauce Labs, PagerDuty, IBM Cloud Availability Monitoring, IBM Cloud Alert Notification, IBM Vulnerability Advisor, and IBM Globalization Pipeline. You can also substitute other tools for the Cloud Continuous Delivery capabilities, including GitHub, GitHub Enterprise, and Jenkins. Developers can also use their favorite IDEs and editors, such as Visual Studio Code, Eclipse, and more.

Code repositories, issue tracking systems, build systems, and deployment systems represent a wealth of data that can be used to help you deliver apps more efficiently and effectively. IBM Cloud DevOps Insights uses big data analysis to provide valuable insights to executives, managers, and developers. DevOps Insights aggregates and analyzes data from your DevOps toolchain to advise you about the risk of deploying specific changes, and areas to improve both your codebase and team productivity. The Delivery Pipeline can automatically gate deployment to an environment based on the risk of a change.

IBM Training

IBM Cloud catalog: Functions

IBM Cloud Functions is a FaaS platform that runs functions in response to incoming events.



Getting started with IBM Cloud

© Copyright IBM Corporation 2016, 2019

Figure 2-37. IBM Cloud catalog: Functions

IBM Cloud Functions is an event-driven compute platform, also referred to as serverless computing or as FaaS, that runs code in response to events or direct invocations.

What is an action

An action is a small piece of code that can be invoked or set to run automatically in response to an event. In either case, each run results in a record that is identified by a unique activation ID. The input and the result of an action can be seen as key-value pairs. The key is a string and the value is a valid JSON value. An action can be written in the language of your choice and provided to the service as either source code or a Docker image. The action code runs when it is directly invoked by the Cloud Functions API, CLI, or iOS SDK. An action can automatically respond to events from IBM Cloud or third-party services.

Why would you use an action

By using actions, you limit the amount of time that your code is running, which lowers your overhead costs. For example, you can use actions to detect faces in an image, respond to changes in a database, aggregate a set of API calls, or even post a tweet.

Can you use more than one action at a time

Yes. You can use actions to call other actions, or you can string actions together to create sequences. To make this work, the output of one action is the input for another action, which provides an output that can be used to trigger another action, and so on. You can even bundle the group of actions that you create to form a package. With a package, you can reuse common actions or sequences by calling the package instead of configuring the action or sequence again.

2.5. Creating and managing a Cloud Foundry application

Figure 2-38. Creating and managing a Cloud Foundry application

IBM Training

Topics

- Introduction to IBM Cloud
- IBM Cloud compute choices
- IBM Cloud console
- IBM Cloud catalog
- Creating and managing a Cloud Foundry application
- IBM Cloud regions and availability zones (optional)
- Managing your IBM Cloud users and resources (optional)

Getting started with IBM Cloud

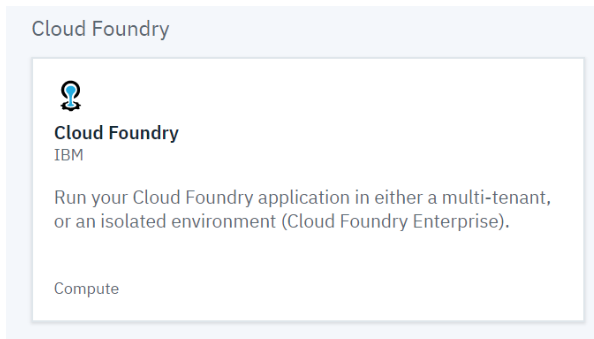
Figure 2-39. Topics

IBM Training

Creating an IBM Cloud Foundry application (1 of 3)

1. To create an IBM Cloud Foundry application, click **Create resource** from the dashboard.

2. From the catalog, click **Compute** and then scroll down and select **Cloud Foundry** service.



Getting started with IBM Cloud

Figure 2-40. Creating an IBM Cloud Foundry application (1 of 3)

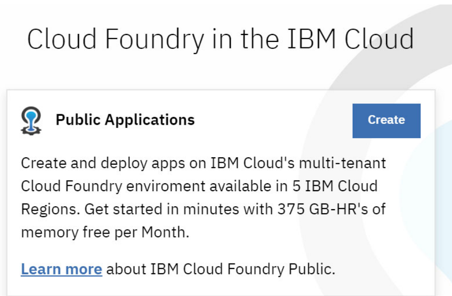
The following runtimes are now available as Cloud Foundry application runtimes:

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

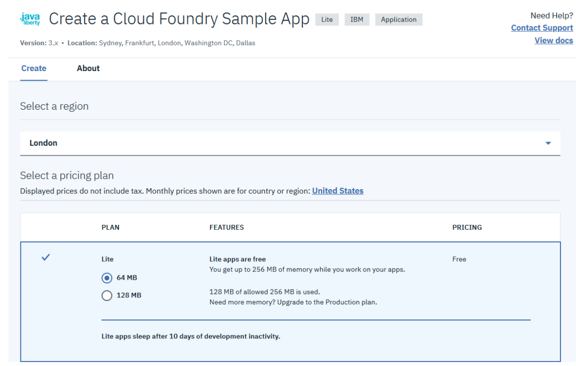
IBM Training

Creating an IBM Cloud Foundry application (2 of 3)

3. From Cloud Foundry Overview page, click **Create** under **Public Applications**.



4. Select region and pricing plan.



Getting started with IBM Cloud

© Copyright IBM Corporation 2016, 2019

Figure 2-41. Creating an IBM Cloud Foundry application (2 of 3)

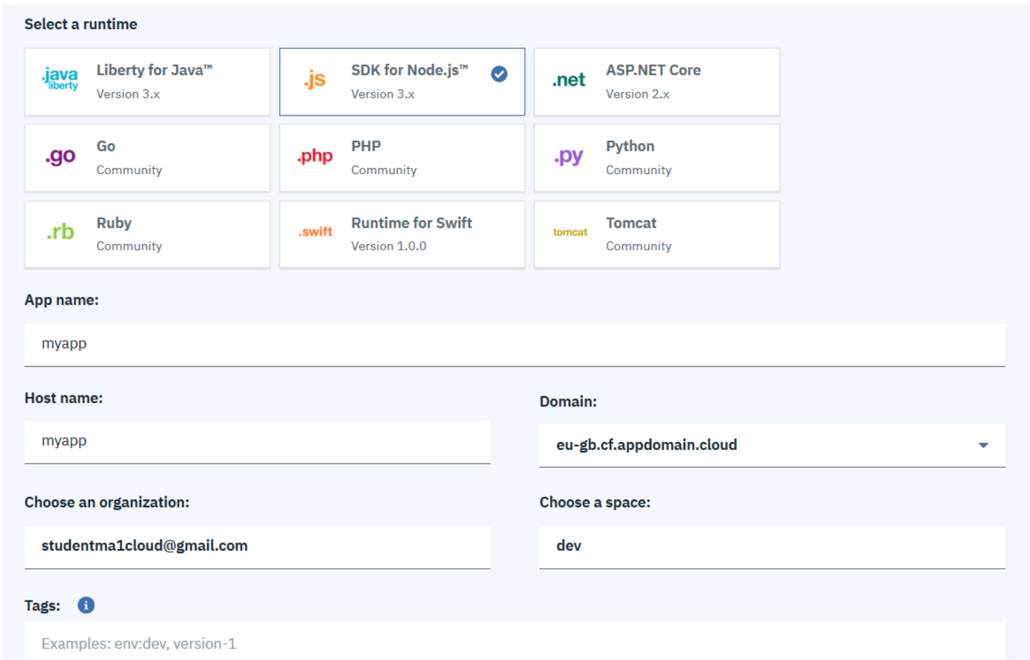
From the Cloud Foundry overview page, click **Create** for Public Applications

Select or confirm the region and select the pricing plan for this application. If you are using the IBM Cloud Lite account, you are not charged for usage.

IBM Training

Creating an IBM Cloud Foundry application (3 of 3)

4. Select the runtime and enter the app name and host name.



Getting started with IBM Cloud

Figure 2-42. Creating an IBM Cloud Foundry application (3 of 3)

Enter your application name and host name. By default, the host name is the same as the app name.

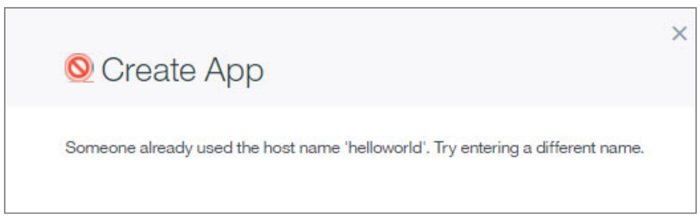
IBM Training

Host name must be unique

- IBM Cloud hosts your application on the domain that you select. It is preferable that you select `appdomain.cloud`.

  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 a host name that is used by another application.



Getting started with IBM Cloud

Figure 2-43. Host name must be unique

IBM Cloud hosts your application on the domain that you select. It is preferable that you select appdomain.cloud.

You do not share your memory and application instances with other IBM Cloud accounts.

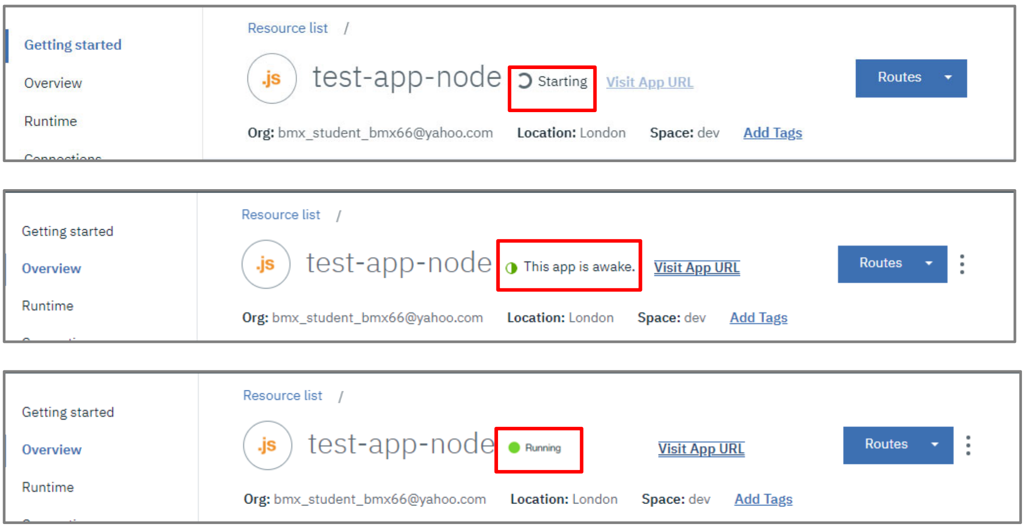
You must choose a host name that is unique across all applications from all IBM Cloud users. You cannot create an IBM Cloud application with the same host name that is used by another application.

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.

IBM Training

Creating an IBM Cloud Foundry application (3 of 3)

4. After waiting for your application to be deployed, the application is available and its status has changed from **Starting** to **Running** or **Awake**, based on the account type.



Getting started with IBM Cloud

Figure 2-44. Creating an IBM Cloud Foundry application (3 of 3)

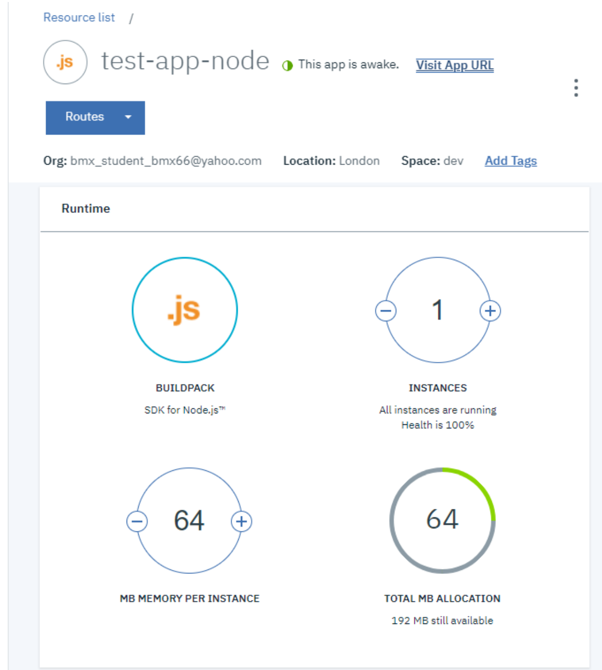
- The first figure shows the application with the status **Starting**.
- The second figure shows the application after it is successfully started in a Lite account with the status **This app is awake**.
- The third figure shows the application after it is successfully started in a non-Lite account (Pay-As-You-Go or Subscriber account) with the status **Running**.

IBM Training

IBM Cloud: Application Details page

Check and control the status of applications in the Application Details page in the Overview section:

- 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.



Getting started with IBM Cloud

Figure 2-45. IBM Cloud: Application Details page

By default, the application is created with one instance with 64 MB of memory per instance. You can adjust the number of instances and memory based on the type of your account. By default, the Lite account has a total of 256 MB allocated memory for all runtime applications.

IBM Training

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 or awake application only.
  - The application route must be unique across all applications in the domain.

Getting started with IBM Cloud

Figure 2-46. Testing applications through the application route

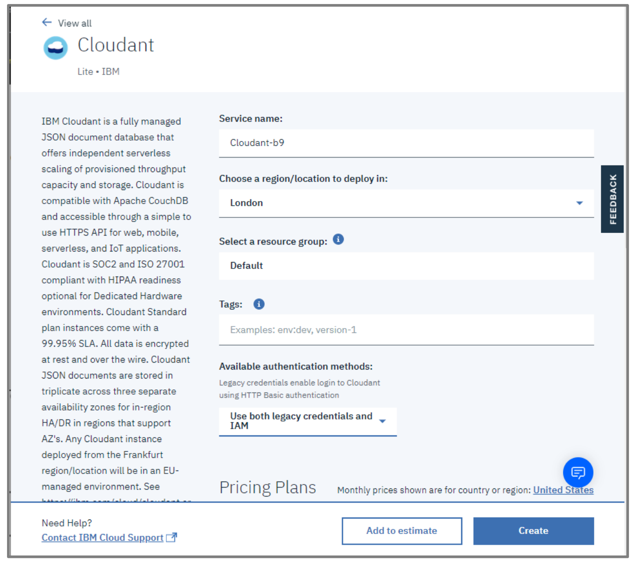
To view the application route, click **Visit App URL** from the Application Details page.

IBM Training

Adding an IBM Cloud service

Add a service by completing the following steps:

1. From the Dashboard, click **Create resource**.
2. Select a service from the catalog.
3. Enter a name for the service.
4. Select the authentication method.
5. Choose a Pricing Plan and then click **Create**.



Getting started with IBM Cloud

Figure 2-47. Adding an IBM Cloud service

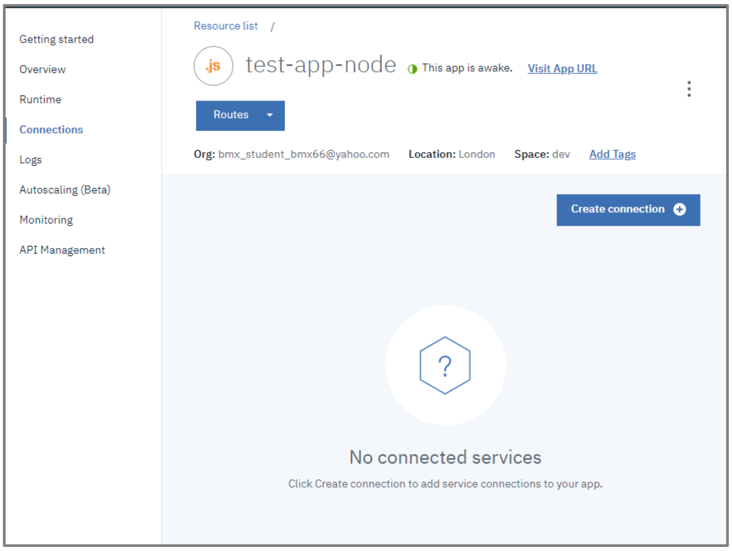
You can add an IBM Cloud service to your application by clicking **Create resource** from the application dashboard and then selecting a service from the IBM Cloud catalog. You must then enter a name for the service and select a pricing plan for your service. Plans allow you to choose different “sizes”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

IBM Training

Binding a service to an application (1 of 2)

1. From the Application Details page, select **Connections**. Then, click **Create connection**.



Getting started with IBM Cloud

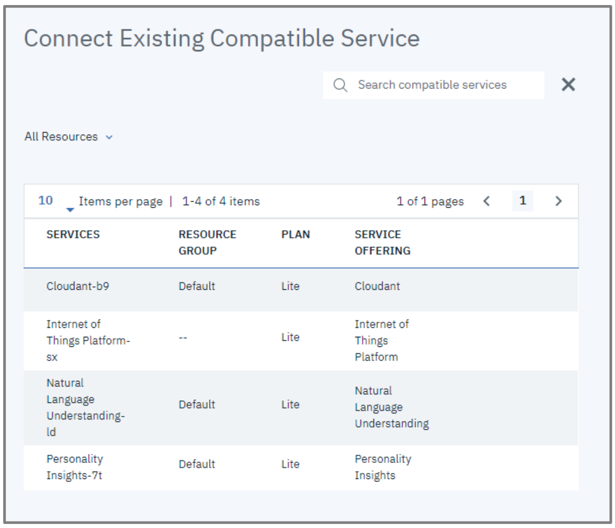
Figure 2-48. Binding a service to an application (1 of 2)

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**.

IBM Training

Binding a service to an application (2 of 2)

2. Select the service to bind to your application. Then, click **Connect**:
   - 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.



Getting started with IBM Cloud

© Copyright IBM Corporation 2016, 2019

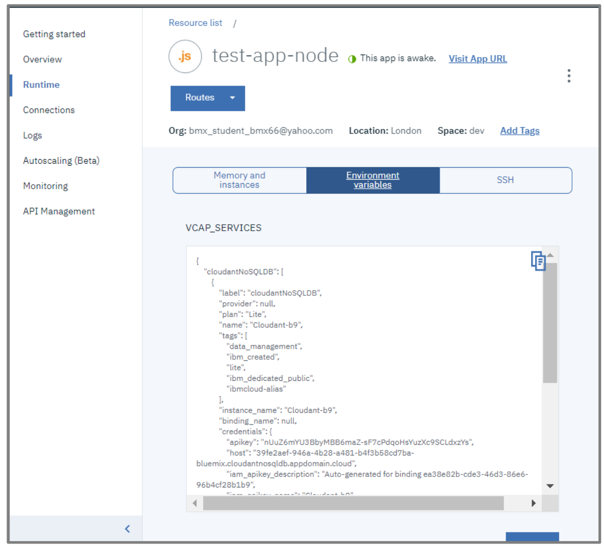
Figure 2-49. Binding a service to an application (2 of 2)

While you are connecting the application with the service, it prompts you to restage your application. Restaging the application makes the service available to be used within the application that uses its service credentials, which are also known as **VCAP_SERVICES**.

IBM Training

IBM Cloud: Environment variables

- Environment variables contain the environment information about a deployed application on IBM Cloud:
  - VCAP_SERVICES
  - User-defined
- You can refer to environment variables within the application code.



Getting started with IBM Cloud

Figure 2-50. IBM Cloud: Environment variables

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 hardcoding 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, and then clicking the **Environment variables** tab.

2.6. IBM Cloud regions and availability zones (optional)

IBM Training

IBM Cloud regions and availability zones (optional)

Getting started with IBM Cloud

Figure 2-51. IBM Cloud regions and availability zones (optional)

IBM Training

Topics

- Introduction to IBM Cloud
- IBM Cloud compute choices
- IBM Cloud console
- IBM Cloud catalog
- Creating and managing a Cloud Foundry application
- IBM Cloud regions and availability zones (optional)
- Managing your IBM Cloud users and resources (optional)

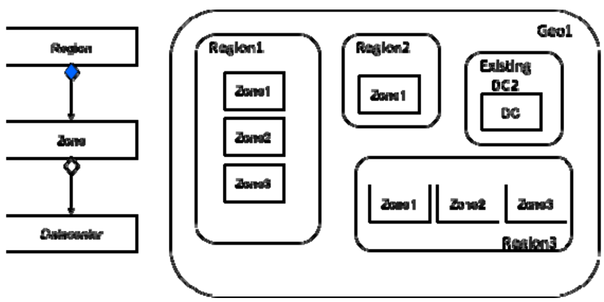
Getting started with IBM Cloud

Figure 2-52. Topics

IBM Training

IBM Cloud: Choice of regions and availability zones

- *Regions* are geographically separated from one another, and have physically distinct infrastructure from all other regions and unrelated zones.
- *Zones* are logically isolated data centers within a single campus, each having isolated electrical, mechanical, and network infrastructures. Zones are separated physically from one another. Within a zone, resources are connected by ultra-low latency, high-bandwidth networks.



MZR Name	Data Centers
us-south	dal10, dal12, dal13
us-east	wdc04, wdc06, wdc07
eu-gb	lon04, lon05, lon06
eu-de	fra02, fra04, fra05
jp-tok	tok02, tok04, tok05
au-syd	syd01, syd04, syd05

Getting started with IBM Cloud

Figure 2-53. IBM Cloud: Choice of regions and availability zones

IBM Cloud Region

A *region* is a geographically and physically separate group of one or more availability zones with independent electrical and network infrastructures that are isolated from other regions. Regions remove shared single points of failure with other regions and ensure low inter-zone latency within the region.

Regions across the world enable customers to support compliance with regulations, laws, and governance relating to data storage (at rest and in transit) and to locate applications based on market and business needs.

A major advantage over the current IBM Cloud data center infrastructure is that most regions across the globe are composed of multiple zones, which empower customers to implement applications and consume IBM Cloud services that are deployed across multiple zones to take advantage of the additional resiliency and high availability that are provided by the redundant infrastructure.

The network that connects different zones in a region delivers low-latency and high bandwidth to enable application designs that can do the following tasks:

- Replicate data across zones.
- Fail over between zones without interruption.
- Provide full business continuity.

IBM Cloud Availability Zone

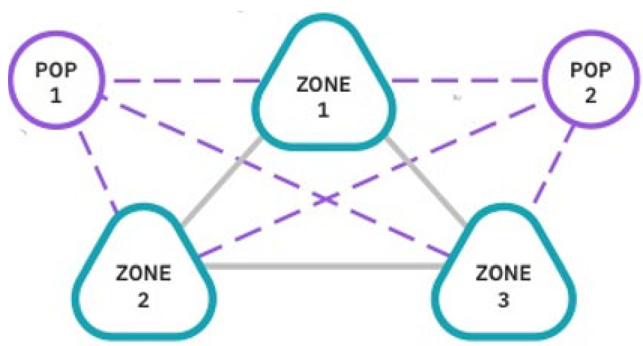
An availability zone is a logically and physically isolated location within an IBM Cloud region with independent power, cooling, and network infrastructures that are isolated from other zones to strengthen fault tolerance by avoiding single points of failure between zones while also ensuring high bandwidth and low inter-zone latency within a region.

IBM Training

IBM Cloud: Multi-Availability Zones

A *Multi-Zone Region (MZR)* has three or more zones that provide high availability and resiliency:

- Typically 3+ data centers within ~10 km of distance of each other with low latency between data centers.
- Each MZR offers a full and consistent set of services to support your customers' needs for highly available and enterprise-class workloads.



MZR example:

- Region: **us-south**
- Zones: **dal10, dal12, dal13**

Getting started with IBM Cloud

Figure 2-54. IBM Cloud: Multi-Availability Zones

MZRs enable highly available workloads:

- Geographical isolation of zones provides fault isolation to avoid natural disasters that might impact one zone only.
- Independent electrical and mechanical network components for each zone ensure high availability by having redundant resources across zones. Avoids single-point-of-hardware failures.
- Low latency and high bandwidth across zones (within a region) enables distributed services across zones to communicate efficiently.
- Regional services like Elastic Load Balancing and Auto-Scaling increase fault tolerance.
- Cross-region services, such as IBM Cloud Object Storage, keep instances across regions

2.7. Managing your IBM Cloud users and resources (optional)

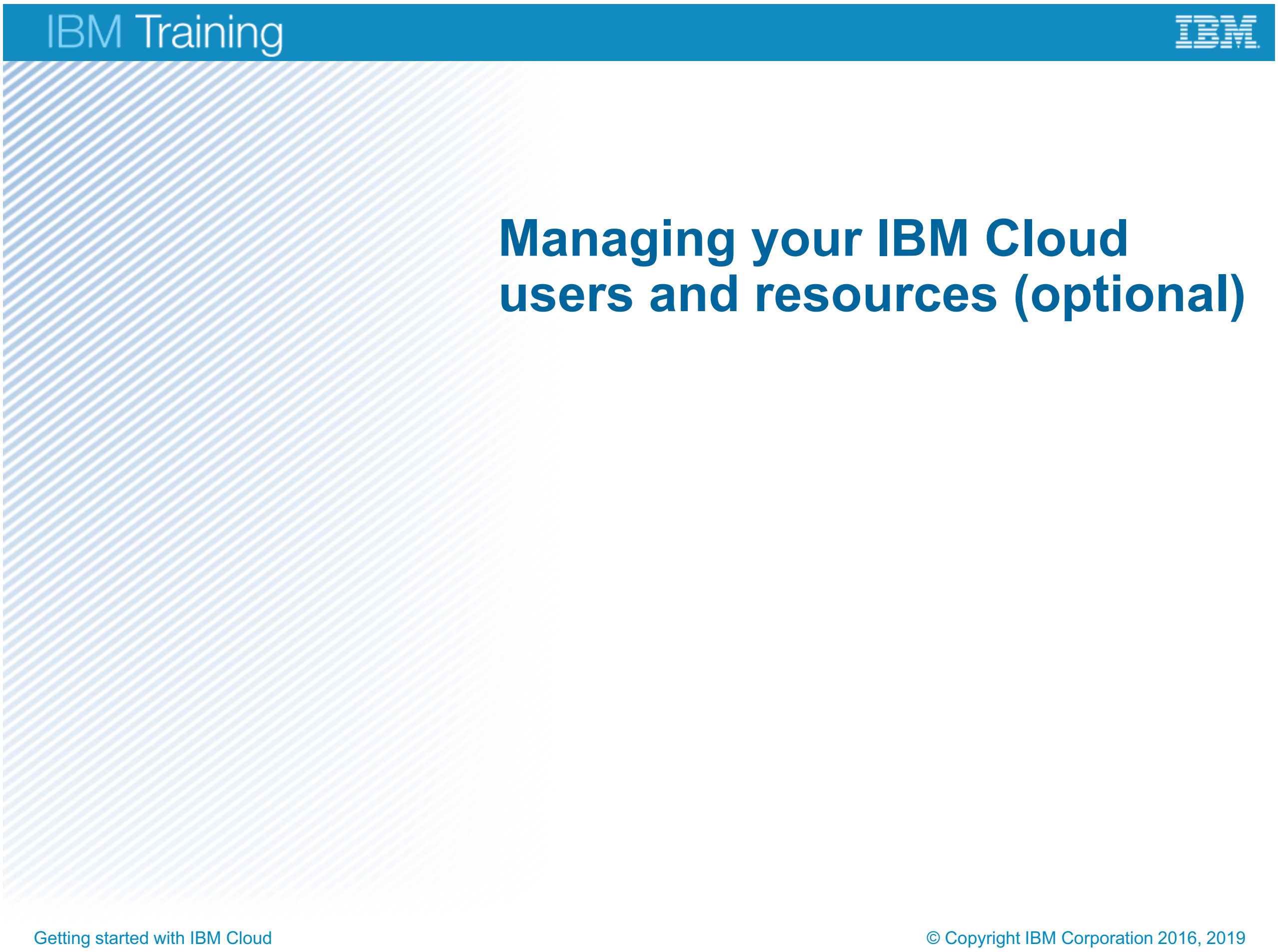


Figure 2-55. Managing your IBM Cloud users and resources (optional)

IBM Training

Topics

- Introduction to IBM Cloud
- IBM Cloud compute choices
- IBM Cloud console
- IBM Cloud catalog
- Creating and managing a Cloud Foundry application
- IBM Cloud regions and availability zones (optional)
- Managing your IBM Cloud users and resources (optional)

Getting started with IBM Cloud

Figure 2-56. Topics

IBM Training

IBM Cloud: Identity and Access Management

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

- Unified user management across IaaS and PaaS resources
- Enterprise federation
- Fine-grained access control

Getting started with IBM Cloud

Figure 2-57. IBM Cloud: Identity and Access Management

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 the 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 that they need.

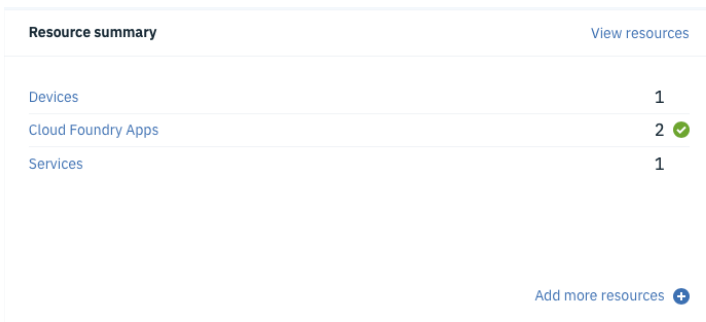
Reference:

https://cloud.ibm.com/docs/iam?topic=iam-userroles

IBM Training

IBM Cloud: Resources, users, and access control

- A resource is an entity in your account that you create from the IBM Cloud catalog. You can create multiple resources in an account.
- You can invite multiple users to an account and grant them access to resources.
- If the resources use Cloud IAM for access control, you can grant users access to the resources by using customizable *resource groups*.
- If the resources do not use Cloud IAM, you can use Cloud Foundry regions, organizations, and spaces for access control.



Getting started with IBM Cloud

Figure 2-58. IBM Cloud: Resources, users, and access control

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 VM, or a container are resources. Each account can have multiple resources.

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

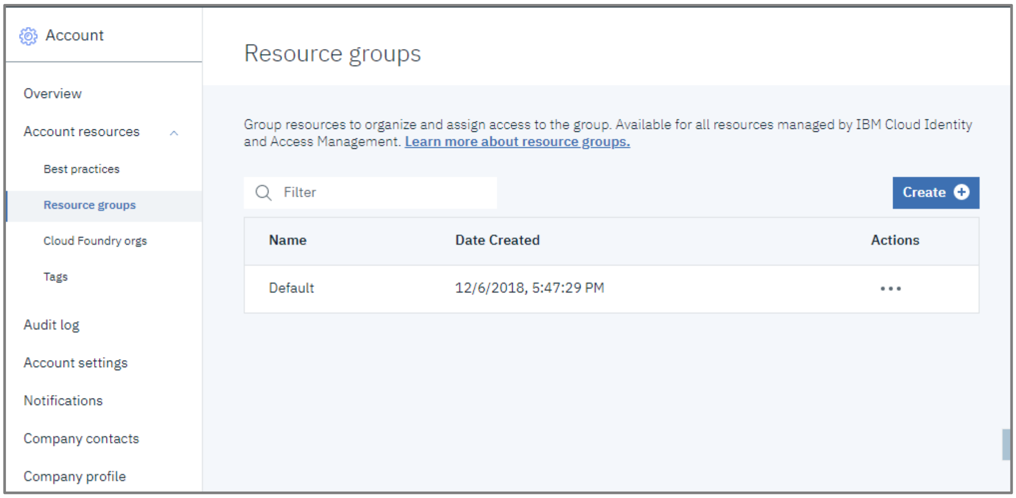
Users can be granted access to resources in an account in the following ways:

- 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 the Cloud Foundry apps and services that have not yet enabled the use of Cloud IAM. These resources cannot be added to a resource group

IBM Training

IBM Cloud: Resource groups

A resource group is a way for you to organize your account resources into customizable groups.



Getting started with IBM Cloud

Figure 2-59. IBM Cloud: Resource groups

A resource group is a way for you to organize your account resources into customizable groups so that you can quickly assign users access to more than one resource concurrently.

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. Select **Manage > Account** from the top navigation bar.
2. Click **Resource Groups**.
3. Click **Create**.
4. Specify the name of the resource group.
5. Click **Add**.

By default, there is a default resource group that is available, so by default any new resource is added to this group unless the user select a different one.

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

IBM Training

IBM Cloud: Resource controller

The *resource controller* is responsible for managing the lifecycle of resources in an account:

- Offers fine-grained access control to resources through IAM.
- Organizes resources by using resource groups:
  - Users are granted access to resources in a resource group.
  - Access to resources is not restricted to Cloud Foundry regions, organizations, and spaces.
- Resources that are based on Cloud Foundry remain assigned to Cloud Foundry regions, organizations, and spaces.

Getting started with IBM Cloud

Figure 2-60. IBM Cloud: Resource controller

The *resource controller* is the next-generation provisioning layer that manages the lifecycle of cloud resources. Previously, all services that were integrated into IBM Cloud used Cloud Foundry and an IBM Cloud version of the Cloud Foundry service broker. Although many similarities to the Cloud Foundry model still exist, the resource controller introduces several new concepts and changes to the Cloud Foundry model.

In general, resources that are tracked by the resource controller are intended to have associated usage metrics and billing, but that is not always the case. In some instances, the resource might be associated with the resource controller to ensure that the resource lifecycle can be managed along with the account lifecycle.

The resource controller is responsible for managing the lifecycle of resources in an account. It offers fine-grained access control to resources through IAM. Consider the following points:

- Resources are organized by using RGs.
- Users are granted access to resources in a resource group.
- Access to resources is not restricted to Cloud Foundry regions, organizations, and spaces.

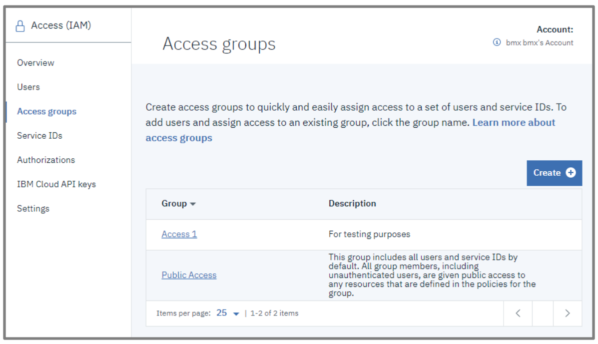
Another key objective of the resource controller is to make the IBM Cloud platform independent of Cloud Foundry. IBM Cloud is built on top of Cloud Foundry, so it inherits Cloud Foundry semantics, some of which are powerful, but others are limiting. For example, the Cloud Foundry service broker specification is a powerful concept that provides an extensible model for service providers to plug their services into the IBM Cloud platform.

However, coarse-grained access control and space-scoped service instances that are tied to a Cloud Foundry region are some of the inherited limitations. As IBM Cloud moves forward into the next generation of cloud, it will retain the service broker extensibility model while breaking away from Cloud Foundry organizations and space constructs.

IBM Training

IBM Cloud: Access group

An *access group* can be created to organize a set of users and service IDs into a single entity that makes it easy for you to assign access.



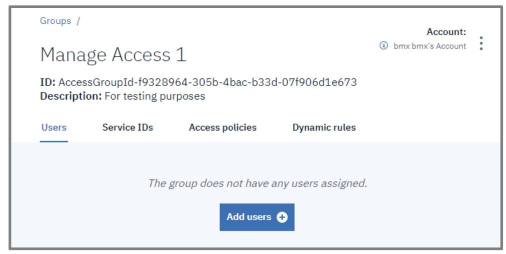


Figure 2-61. IBM Cloud: Access group

To create an access group, complete the following steps:

1. From the menu bar, click **Manage** > **Access (IAM)**, and select **Access Groups**.
2. Click **Create**.
3. Enter a name and optional description for your group, and click **Create**.

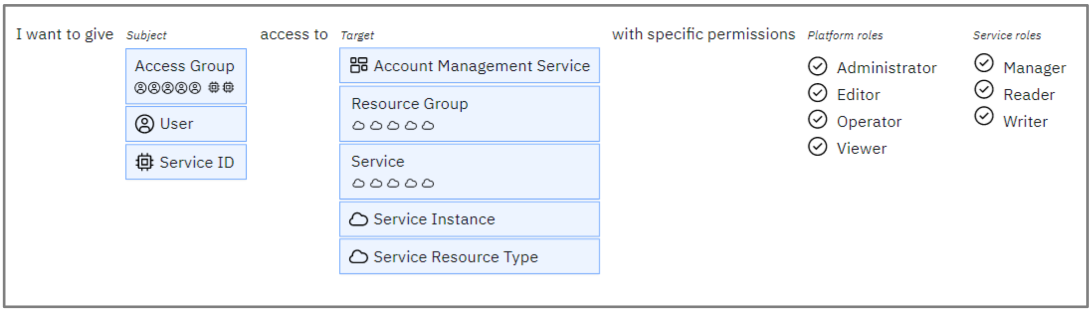
Next, continue to set up your group by adding users or service IDs:

1. Select the name of the group to which you want to add.
2. Click **Add users** on the **Users** tab.
3. Select the users that you want to add from the list, and click **Add to group**.
4. To add service IDs to the group, click the **Service IDs** tab, and click **Add service ID**.
5. Select the IDs that you want to add from the list, and click **Add to group**.

IBM Training

IBM Cloud: IAM policies

A policy grants a subject one or multiple roles in a set of resources so that specific actions can be taken within the context of the specified target resources.



Getting started with IBM Cloud

Figure 2-62. IBM Cloud: IAM policies

This slide explains how the IAM policy is created. Policies are always created by specifying the subject first. The subject is a specific user, service ID, or an access group. Next, the target of the policy is selected, which is what you are allowing the user to access, for example, all services in a resource group, all IAM-enabled services in the account, account management services, or a particular service instance. Finally, you complete your access policy by selecting from the available roles. These roles define exactly what actions a user can complete. More configuration options might be available, depending on the service you select.

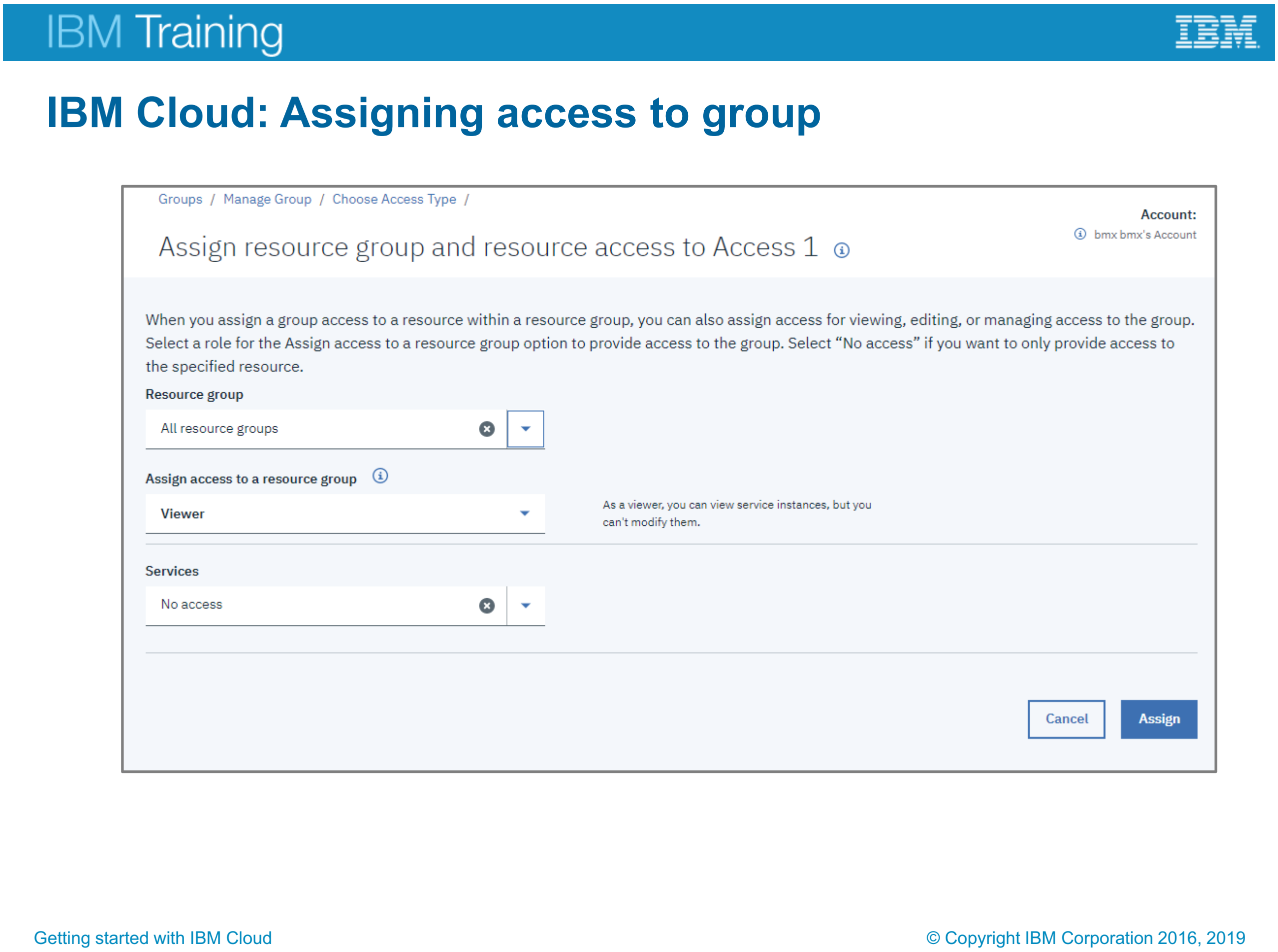


Figure 2-63. IBM Cloud: Assigning access to group

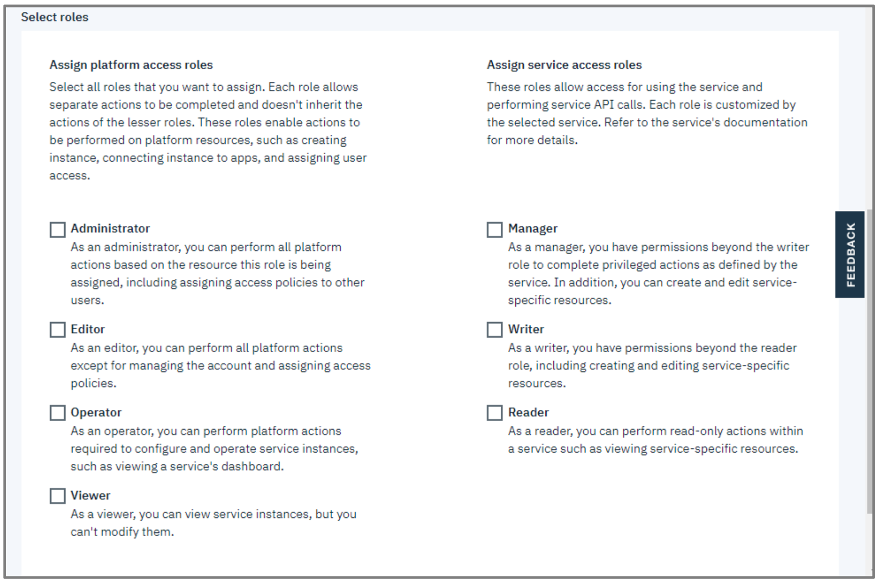
After you set up your group with users and service IDs, you can assign a common access policy to the group. Remember, any policy that you set for the group applies to all entities within the group.

1. From the menu bar, click **Manage** > **Access (IAM)**, and select **Access Groups**.
2. Select the name of the group to which you want to assign access.
3. Click **Access policies**.
4. Click **Assign access**.
5. Choose to assign access by resources within a resource group, individual resources available within the account, or account management services.

IBM Training

IBM Cloud: Cloud IAM roles

With Cloud IAM, you can manage and define access for users and resources in your account. Two types of roles can be assigned: platform management roles and service access roles.



Getting started with IBM Cloud

Figure 2-64. IBM Cloud: Cloud IAM roles

Platform management roles

Platform management roles cover a range of actions, including the ability to create and delete instances; manage aliases, bindings, and credentials; and manage access. The platform roles are administrator, editor, operator, viewer. Platform management roles also apply to account management services that enable users to invite users, manage service IDs, access policies, catalog entries, and track billing and usage depending on their assigned role on an account management service.

Service access roles

Service access roles define a user or service’s ability to perform actions on a service instance, such as accessing the console or performing API calls. The service access roles are manager, writer, and reader.

IBM Training

IBM Cloud: Organizing resources

Specify the resource group when creating a resource

Getting started with IBM Cloud

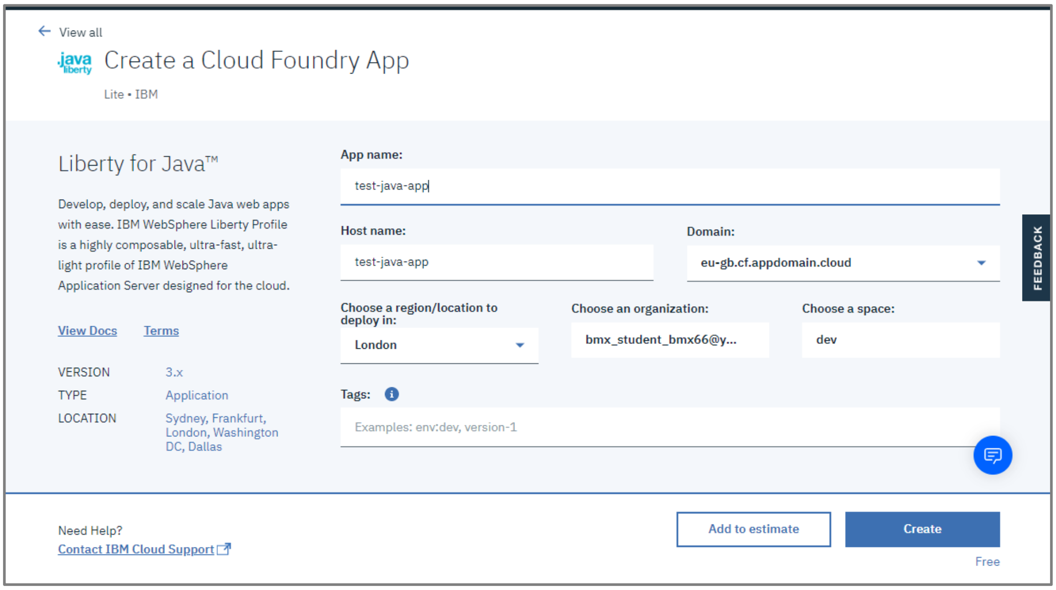
Figure 2-65. IBM Cloud: Organizing resources

When creating a resource, select the resource group, as shown in the slide. Most of the services in the IBM Cloud catalog use Cloud IAM for access control.

IBM Training

Cloud Foundry: Organizing resources

A Cloud Foundry resource is organized by region, organization, and space.



Getting started with IBM Cloud

Figure 2-66. Cloud Foundry: Organizing resources

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

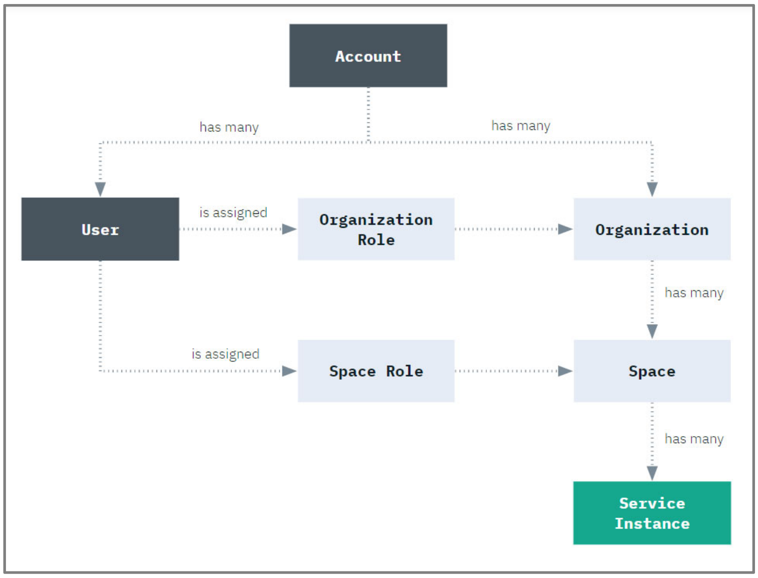
Services that use Cloud IAM have the following advantages over services that are based on Cloud Foundry:

- 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 group 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.

IBM Training IBM

Cloud Foundry: Access

Cloud Foundry roles are used for access to some service instances in IBM Cloud. Users are added to the org and space to which the instance belongs with a Cloud Foundry role assigned.



Getting started with IBM Cloud

Figure 2-67. Cloud Foundry: Access

This slide outlines how Cloud Foundry orgs, spaces, and roles relate within an account. An account can have many users, orgs, and spaces. Each user can be assigned to as many orgs and spaces as necessary, and when they are assigned to an org and space, you can set the level of access to work within each by assigning a Cloud Foundry role.

IBM Training

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).

Getting started with IBM Cloud

Figure 2-68. Cloud Foundry: Organizations

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.

IBM Training

Cloud Foundry: Domains and quota

- A *domain* provides a route on the internet that is allocated to an organization:
  - A *route* consists of a subdomain (also known as a host name) and a domain name.
  - 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

Getting started with IBM Cloud

Figure 2-69. Cloud Foundry: Domains and quota

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

- A route has a subdomain and a domain.
- A subdomain 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 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.

IBM Training

Cloud Foundry: Spaces

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

Getting started with IBM Cloud

Figure 2-70. Cloud Foundry: Spaces

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.

IBM Training

Cloud Foundry: User roles

Role	Permissions and tasks
Organization managers	• Create or delete spaces within the organization. • Invite users to the organization and manage users. • Manage 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	• Add users to the space and manage users. • Enable features for the space.
Space developers	• Create, delete, and manage applications and services within the space. • Have access to logs within the space.
Space auditors	Have read-only access to settings, logs, applications, and services.

Getting started with IBM Cloud

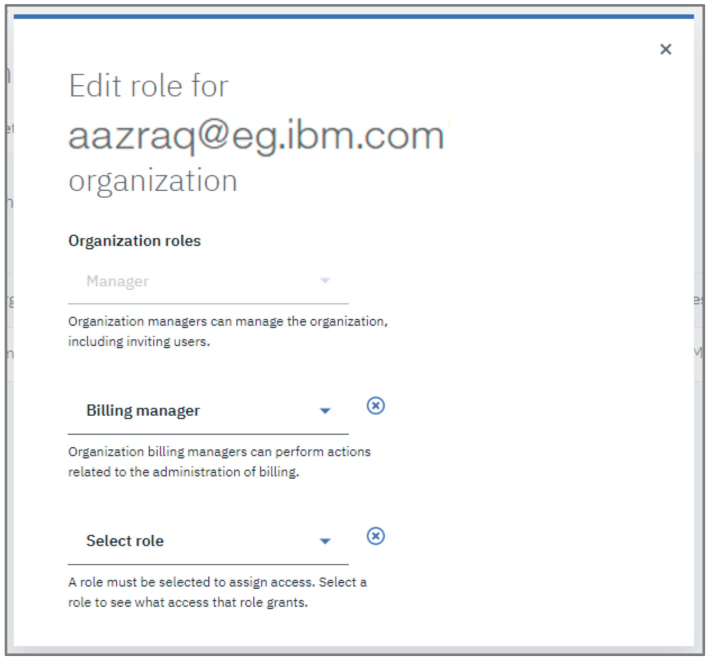
Figure 2-71. Cloud Foundry: User roles

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.
- The space manager can control who has access to the space.
- The space developer can create, delete, and manage applications and services within the space.
- The space auditors have read-only access to settings, logs, applications, and services.

IBM Training

Managing Cloud Foundry organizations



Getting started with IBM Cloud

Figure 2-72. Managing Cloud Foundry organizations

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, complete the following steps:

1. Click **Manage** on the top toolbar, click **Access (IAM)**, and click **Users** from the right menu.
2. Click the three dots in the selected user, and then select **Manage user details**.
3. From the top menu bar, select **Cloud Foundry access**, click Actions (the three dots icon) next to the organization, and click **Edit organization role**.

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

IBM Training

Inviting users to IBM Cloud

You can invite users, cancel invitations, and resend a pending invitation to an invited user. In addition, you can invite a single user or multiple users at once.

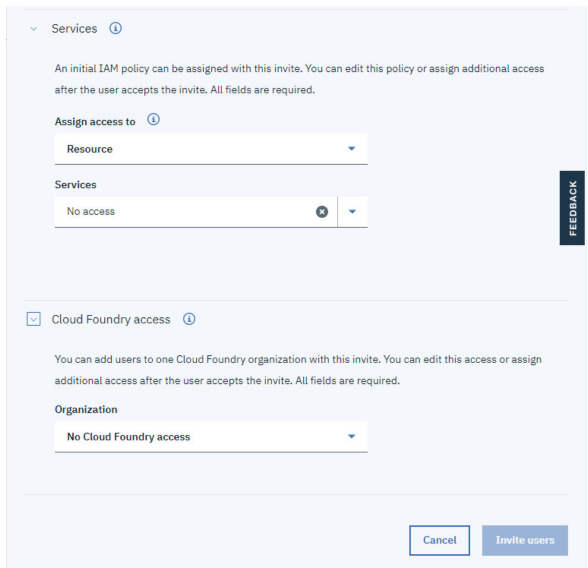


Figure 2-73. Inviting users to IBM Cloud

To invite users or manage user invitations in your account, complete the following steps:

1. From the menu bar, click **Manage** > **Access (IAM)**, and then select **Users**.
2. Click **Invite users**.
3. Specify the email address of the user. If you are inviting more than one user with a single invitation, they are all assigned the same access.
4. Add one or more of the access options that you manage. You must assign at least one access option. For any additional access options that you do not add and configure, the default value of **no access** is assigned. You might see one or all of the following access options, depending on the options that you are authorized to manage:
   - **Services**
   - **Cloud Foundry access**
   - **Classic infrastructure access**

IBM Training

Unit summary

- Describe IBM Cloud.
- Distinguish among the various compute options in IBM Cloud.
- Identify the runtimes and services that IBM Cloud offers.
- Describe IBM Cloud regions, zones, and multi-availability zones.
- Describe the IBM Cloud dashboard, catalog, and documentation features.
- Work with IBM Cloud resources.
- Explain starter kits and Cloud Foundry boilerplates.
- Describe how to manage your IBM Cloud users and resources (optional).
- Explain Identity and Access Management (IAM) and Resource Groups (RGs) (optional).
- Describe how the application route is used to test an application in the browser.
- Bind services to an application in IBM Cloud.
- Describe the environmental variables that are used with IBM Cloud services.
- Explain function as a service.

Getting started with IBM Cloud

Figure 2-74. Unit summary

IBM Training

Related resources

- The IBM Cloud documentation from the web interface provides an up-to-date reference and tutorial about building, deploying, and managing applications:
  http://cloud.ibm.com/docs/
- Join the IBM Cloud developers community to view videos, tutorials, and forums:
  https://developer.ibm.com/depmodels/cloud/
- Browse through a list of IBM Cloud solutions and sample applications:
  https://www.ibm.com/cloud/solutions/
- IBM Cloud Container Service:
  https://cloud.ibm.com/docs/containers?topic=containers-getting-started
- IBM Cloud Regions and Zones
  https://cloud.ibm.com/docs/containers?topic=containers-regions-and-zones#regions-and-zones
- IBM Cloud Services by Region
  https://cloud.ibm.com/docs/resources?topic=resources-services_region#services_region

Getting started with IBM Cloud

Figure 2-75. Related resources

IBM Training

Review questions

1. Which of the following statements about IBM Cloud is true (choose one)?
   - A. In IBM Cloud, anyone can manage your applications without permission.
   - B. Services are the primary means that enable users to share control over apps.
   - C. A user must belong to 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 host name.

Getting started with IBM Cloud

Figure 2-76. Review questions

IBM Training

Review questions (cont.)

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.

5. True or False: In IBM Cloud, the user that has a Pay-As-You-Go account cannot use Lite services.

Getting started with IBM Cloud

Figure 2-77. Review questions (cont.)

IBM Training

Review answers

1. Answers:
   A. No. You must authorize each person to access your organization and therefore see your applications.
   B. No. 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 only one space.
2. False. NoSQL databases are IBM Cloud *services*.
3. False. Only one app can use the host name for a specific domain name.
4. The *application* route in IBM Cloud is _____.
   c. The internet URL where users can access the application.
5. False. Lite services may be used by any type of IBM Cloud users.

Getting started with IBM Cloud

Figure 2-78. Review answers

Unit 3. Deploying applications to Cloud Foundry on IBM Cloud

Estimated time

01:00

Overview

This unit introduces Cloud Foundry and describes how to deploy applications to Cloud Foundry on IBM Cloud by using the IBM Cloud CLI. This unit describes the basic structure of Node.js apps.

IBM Training

Unit objectives

- Explain how to manage your IBM Cloud account with the IBM Cloud CLI.
- Describe how to create a Node.js application that runs on IBM Cloud.
- Deploy an application from a local workstation by using the IBM Cloud CLI.
- Describe the role of Node.js for server-side scripting.
- Deploy an application by using IBM Cloud App Service (Web Apps).

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-1. Unit objectives

3.1. Introduction to Cloud Foundry

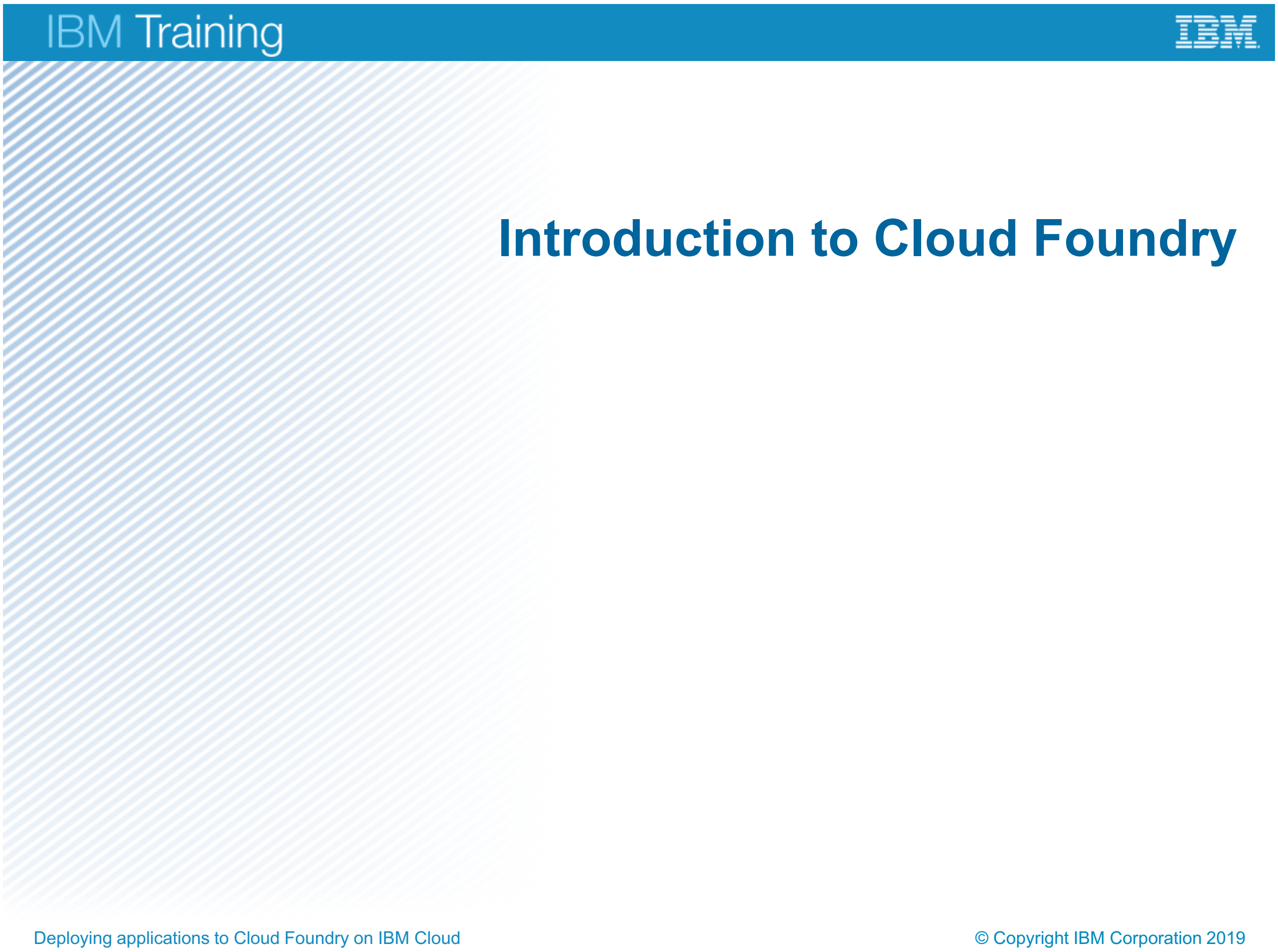


Figure 3-2. Introduction to Cloud Foundry

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-3. Topics

IBM Training

Cloud Foundry benefits

- Helps you deploy applications.
- Decouples applications from infrastructure.
- Language- and framework-neutral.
- Makes building, deploying, and scaling apps fast and easy.
- Is an open cloud-native platform.

IBM Cloud provides a **certified** Cloud Foundry platform.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-4. Cloud Foundry benefits

In today's world, services are the de-facto standard. Services can be hosted publicly on the internet, internally or on-premises. From computing, storage, networking to applications, every layer of the technology stack can be offered "as a service". This movement enables business to focus on delivery business values and move at the velocity that is required to be competitive.

Cloud native applications are infrastructure-unaware. They can be deployed onto any platforms.

Cloud Foundry provides:

- Automated infrastructure management and orchestration
- User management and security auditing for IT operations
- Load balancing and traffic routing
- Centralized log aggregation

IBM Cloud provides a certified Cloud Foundry platform. It offers multiple options to cater to your organization's needs, including:

- Flexible billing options: Per hour, pay-as-you-go for public cloud offerings
- Dedicated and isolated runtime environments
- Private cloud deployment with IBM Cloud Private

3.2. Deploying Cloud Foundry applications with IBM Cloud CLI

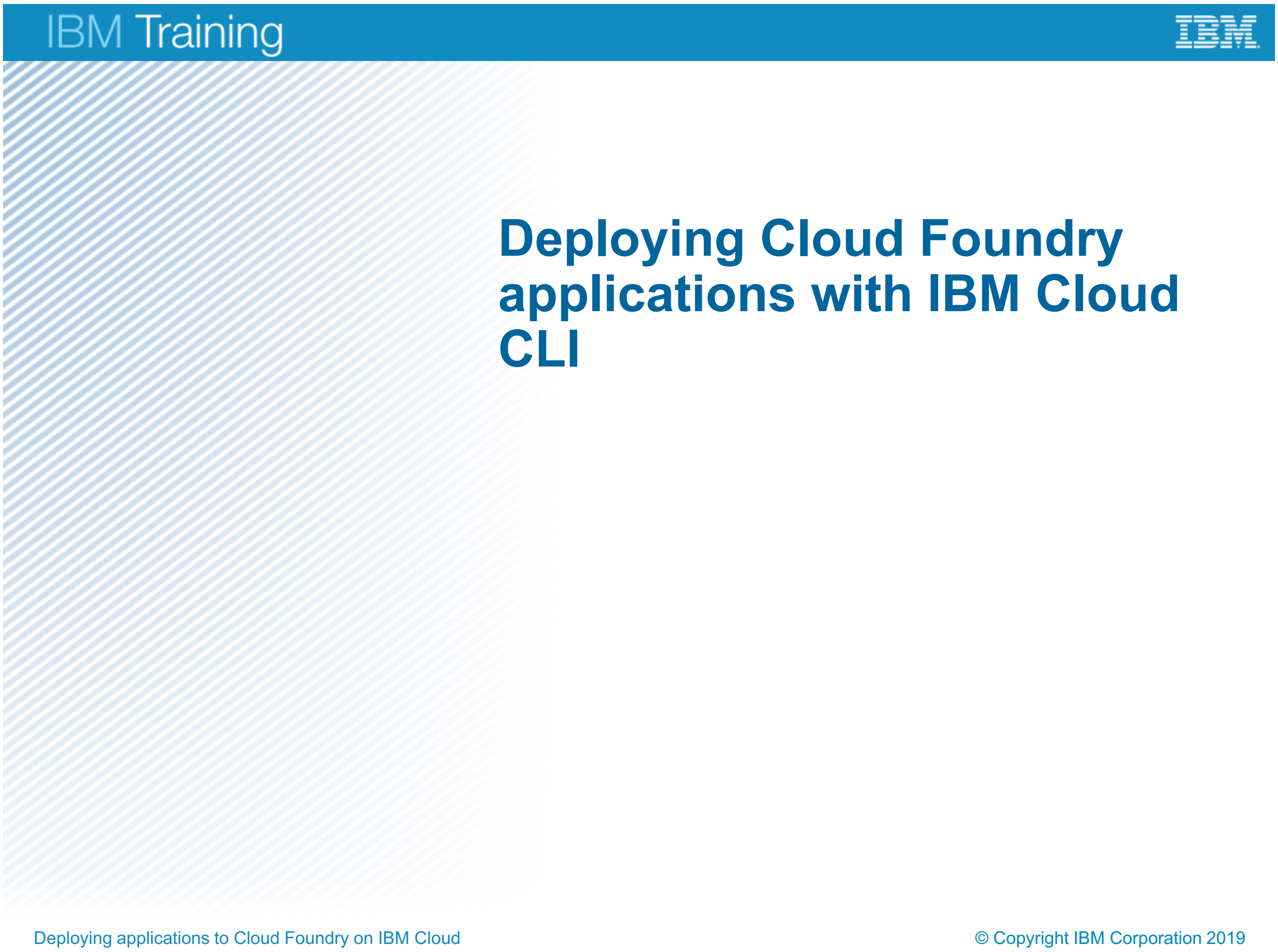


Figure 3-5. Deploying Cloud Foundry applications with IBM Cloud CLI

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-6. Topics

IBM Training

IBM Cloud CLI overview

- IBM Cloud CLI is a general-purpose developer tool that provides access to an IBM Cloud account and services through a command-line interface (CLI).
- Cloud Foundry commands are accessible by using **`ibmcloud cf`**.
- For a complete list of available commands, run `ibmcloud cf help -a`
- For the latest installer and instructions, go to: https://cloud.ibm.com/docs/cli

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-7. IBM Cloud CLI overview

- IBM Cloud CLI is a general-purpose developer tool that provides access to an IBM Cloud account and services through a command-line interface (CLI).
- Cloud Foundry commands are accessible by using **`ibmcloud cf`**.
- For a complete list of available commands, run `ibmcloud cf help -a`.
- For the latest installer and instructions, go to: https://cloud.ibm.com/docs/cli
- Working with a CLI gives you the same workflow across multiple operating systems.
- IBM Cloud CLI is optimized to work with services and manage resources in IBM Cloud.

Reference:

https://cloud.ibm.com/docs/cli?topic=cloud-cli-ibmcloud-cli

IBM Training

Deploying your first Node.js application – Prerequisites: Prepare IBM Cloud

- Before you begin, verify that:
  - You have access to an IBM Cloud account.
  - The IBM Cloud CLI installed is installed in your workstation.
- Run `ibmcloud login` and follow the instructions.
- You are prompted to enter your email, password, and a region to deploy your application.

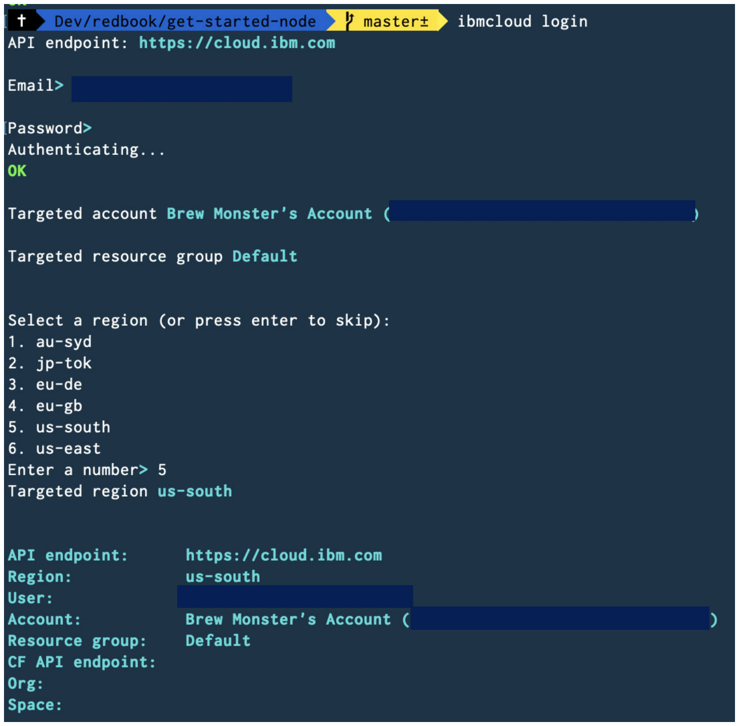
Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-8. Deploying your first Node.js application – Prerequisites: Prepare IBM Cloud

- Your IBM Cloud account uses the same IBM ID that you signed up with.
- To deploy in a region of your choice, make sure to choose the right region when you are creating your IBM ID.
- Before you begin, verify that:
  - You have access to an IBM Cloud account.
  - The IBM Cloud CLI is installed in your workstation.
- Run **`ibmcloud login`** by using the CLI and follow the instructions.
- You are prompted to enter your email, password, and a region to deploy your application.

IBM Training

Deploying your first Node.js application - Prerequisites: Prepare IBM Cloud (cont.)



Deploying applications to Cloud Foundry on IBM Cloud © Copyright IBM Corporation 2019

Figure 3-9. Deploying your first Node.js application - Prerequisites: Prepare IBM Cloud (cont.)

This slide shows an example of the login sequence.

IBM Training

Deploying your first Node.js application - Prerequisites: Prepare IBM Cloud (cont.)

- After you are logged in, set up the Cloud Foundry API endpoint by running the following command:
  `ibmcloud target --cf`
- To check the default organization and space to which you have access, go to https://cloud.ibm.com/account/cloud-foundry.

Figure 3-10. Deploying your first Node.js application - Prerequisites: Prepare IBM Cloud (cont.)

- After you are logged in, set up the Cloud Foundry API endpoint by running the following command:

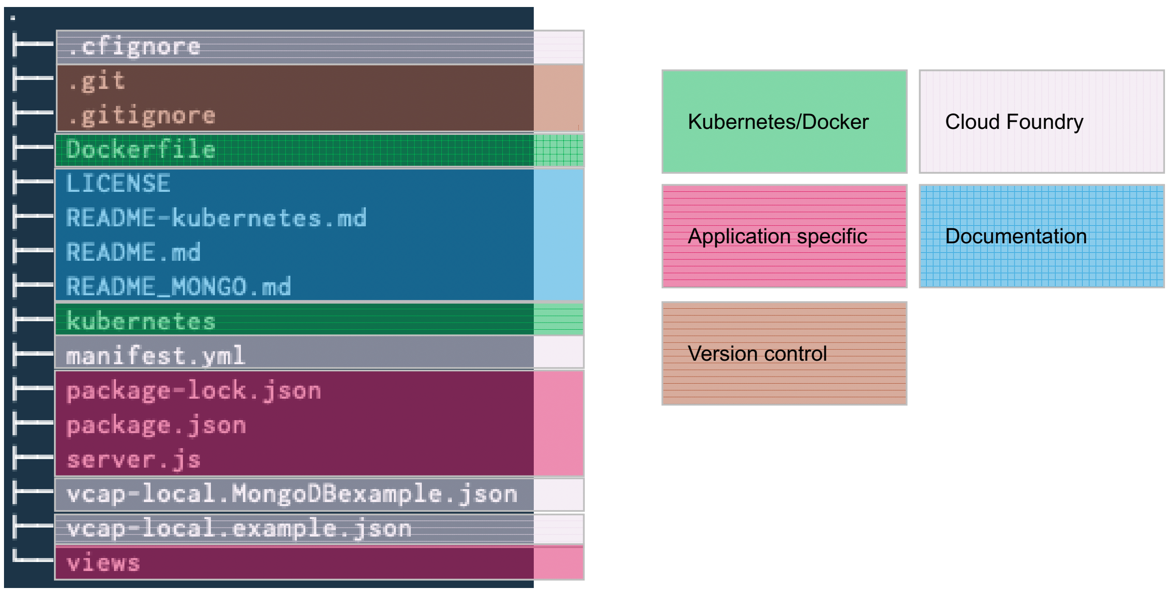
  ibmcloud target --cf

- To move the default organization to a different region, contact IBM Support.
- To check the default organization and space to which you have access, go to https://cloud.ibm.com/account/cloud-foundry.

IBM Training

Deploying your first Node.js application - Step 1: Understanding the sample application

- Clone the sample app by running the following command:
  `git clone https://github.com/IBM-Cloud/get-started-node`
- Go to the `get-started-node` folder, which contains the following files:



Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-11. Deploying your first Node.js application - Step 1: Understanding the sample application

Before we continue to explore the Cloud Foundry capabilities, we need to understand the sample Node.js application. There are multiple files and folders in the sample application, and each one is used for a different purpose. (The colors that highlight the text are described in the graphic to the right of the file structure.)

To ease the deployment when you use Kubernetes, we provide a Dockerfile that is named “`kubernetes deployment.yaml`”. Because you are focusing on Cloud Foundry, you ignore these files. Because you are working with a Node.js application, consider the following items:

- The package.json file provides metadata about the application and its dependencies, and how the application can be started, that is, the JavaScript file that you use in “npm start”.
- When you run “npm install”, Node.js pulls dependencies from NPM and places them into “node_modules”.
- In this project, server.js is used as the starting point of the application.

IBM Training

Deploying your first Node.js application - Step 1: Understanding the sample application (cont.)

- To deploy Cloud Foundry apps on IBM Cloud, you need a `manifest.yml` file.
- The sample application already includes an example of this manifest file:

```
---
applications:
 - name: GetStartedNode
   random-route: true
   memory: 256M
```

- This manifest file lists the deployment configurations, including the name of the app, how much memory the instance should have, and the routing.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-12. Deploying your first Node.js application - Step 1: Understanding the sample application (cont.)

- By declaring the resources configuration, you can deploy your applications automatically with consistency and reproducibility. Although you can change the configuration of your application dynamically by using the CLI, as a best practice, use the manifest file. By running “cf push”, the configurations that are provided in the `manifest.yml` file override the default values and values that are set by the CLI.
- To deploy an existing Node.js application, create a `manifest.yml file` within the directory that contains the app. Inside the manifest.yml file, at a minimum, you must specify the application’s name.
- The slide shows a deployment configuration for an an application that is named GetStartedNode, which requires 256 MB of memory to run. Using “random-route” generates a unique route on IBM Cloud to avoid host name collisions. You learn about routes later in this unit.
- For a complete list of manifest.yml attributes, go to https://docs.cloudfoundry.org/devguide/deploy-apps/manifest-attributes.html.

IBM Training

Deploying your first Node.js application - Step 1: Understanding the sample application (cont.)

- To prevent uploading non-essential files and folders as you deploy your application, create an ignore file.
- The `.cfignore` file prevents **`cf push`** from uploading all the files and folders that are listed in the file:

```
node_modules/
*.DS_Store
README.md
.github/
.git/
.gitignore
logs
*.log
```


Figure 3-13. Deploying your first Node.js application - Step 1: Understanding the sample application (cont.)

Typically, the content for the .cfignore file contains the same folders and files as the .gitignore file because they are the resources that you do not want to push to version control. For example, when you run “npm install” locally, it pulls many packages into “node_modules”. Not uploading this folder saves bandwidth and reduces deployment time.

The content of the .cfignore file should include local build outputs to your workstation’s environment, version-control files, any local logs, and local testing artifacts. This is not a complete list, and it depends on your application.

IBM Training

Deploying your first Node.js application - Step 2: Deploying the sample app

- Open the sample app folder and run **`ibmcloud cf push`**.

```
Waiting for app to start...

name:              GetStartedNode
requested state:   started
routes:            getstartednode-patient-elephant.mybluemix.net
last uploaded:     Wed 10 Apr 10:37:04 AEST 2019
stack:             cflinuxfs2
buildpacks:        SDK for Node.js(TM) (node.js-6.17.0, buildpack-v3.26-20190313-1440)

type:            web
instances:       1/1
memory usage:    256M
start command:   ./vendor/initial_startup.rb
     state     since                  cpu    memory      disk      details
#0   running   2019-04-10T00:37:21Z   0.0%   0 of 256M   0 of 1G
```

- Your application, which is named `GetStartedNode`, is running at a random route.

Deploying applications to Cloud Foundry on IBM Cloud

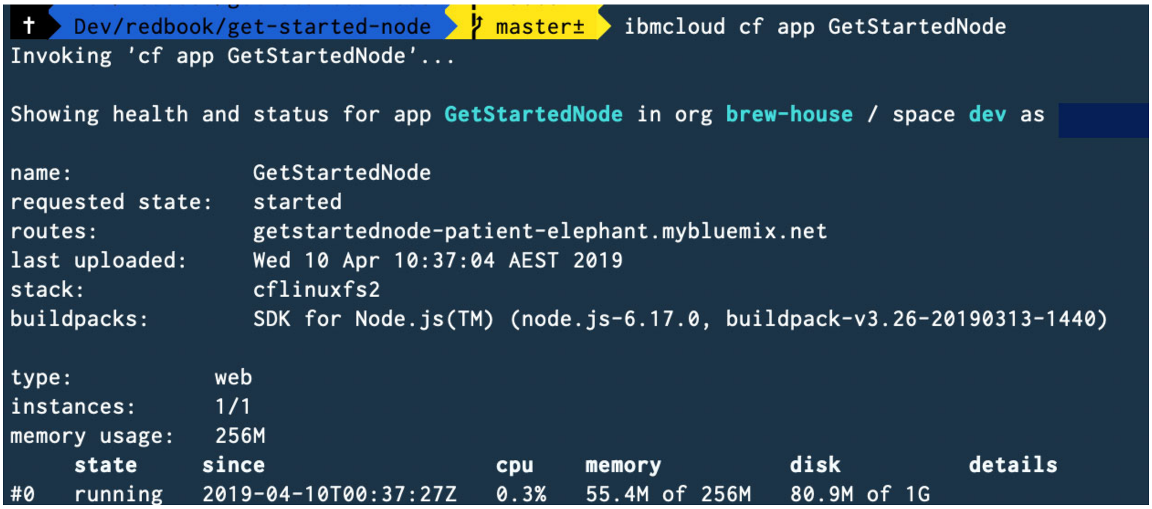
Figure 3-14. Deploying your first Node.js application - Step 2: Deploying the sample app

- The “ibmcloud cf push” command returns control of the CLI to the developer when it either deploys and starts at least one instance of the application or the application fails to start.
- As seen in the slide, Cloud Foundry deployments come with a range of defaults, including but not limited to the buildpack, stack, and number of instances.
- Because you specified “random-route: true” in manifest.yml, Cloud Foundry provides a random route that is named “getstartednode-patient-elephant.mybluemix.net”.
- You learn about buildpacks and routing in a later section in this unit.

IBM Training

Deploying your first Node.js application - Step 3: Checking whether your app is running

- You can view the details of your app by running the following command:
  `ibmcloud cf app GetStartedNode`



Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-15. Deploying your first Node.js application - Step 3: Checking whether your app is running

The basic status of an application includes:

- Name of application.
- Requested state.
- Route of the application. An application can have more than one route.
- When “cf push” was last ran.
- Base container image version (stack).
- Buildpack that is used for deployment.
- Number of running instances.
- Memory usage.
- Health and stats for each instance running for this application. Here, we use the default one instance for this deployment.

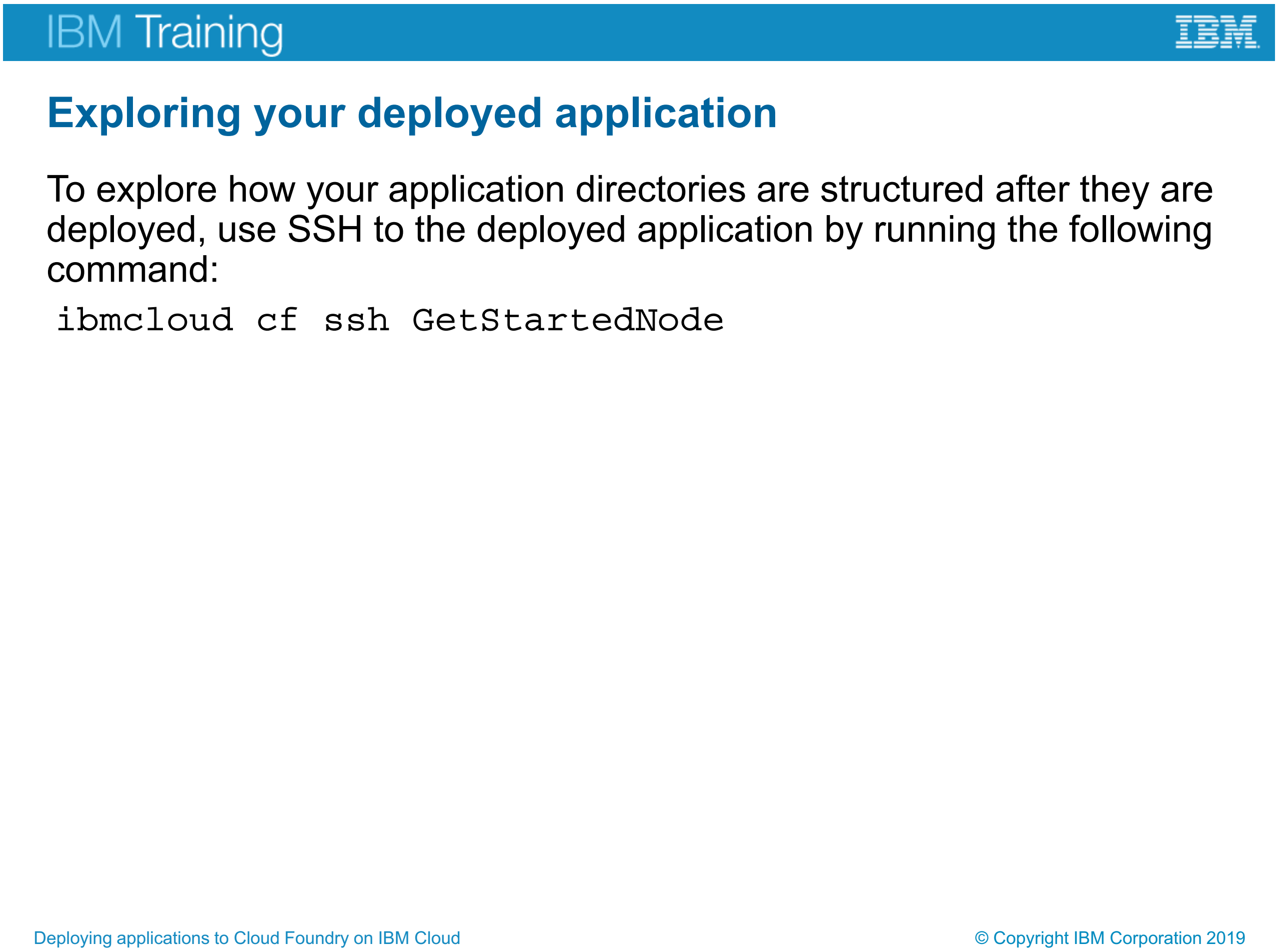


Figure 3-16. Exploring your deployed application

To explore how your application directories are structured after they are deployed, use SSH to the deployed application by running the following command:

```
ibmcloud cf ssh GetStartedNode
```

To exit from a ssh session, invoke `exit`.

Reference:

https://docs.cloudfoundry.org/devguide/deploy-apps/ssh-apps.html

3.3. Organizations and spaces

Figure 3-17. Organizations and spaces

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-18. Topics

IBM Training

Organizations and spaces

- Cloud Foundry is a virtualized layer (containers) on top of virtual machines (VMs).
- Developers do not have direct access to the VM or machines to which they are deploying. When you push your application, Cloud Foundry provisions a logical partition of resources for it.
- Cloud Foundry uses logical boundaries to allocate resources. These are known as *Orgs* (organizations) and *Spaces*.
- These boundaries provide:
  - Separation between Cloud Foundry resources
  - Separation between teams
  - Isolation of development, test, staging, and production environments
- A developer can belong to multiple orgs and spaces.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-19. Organizations and spaces

- Cloud Foundry is a virtualized layer (containers) on top of virtual machines (VMs).
- Developers do not have direct access to the VM or machines to which they are deploying. When you push your application, Cloud Foundry provisions a logical partition of resources for it.
- Cloud Foundry uses logical boundaries to allocate resources. These are known as *Orgs* (organizations) and *Spaces*.
- These boundaries provide:
  - Separation between Cloud Foundry resources
  - Separation between teams
  - Isolation of development, test, staging, and production environments
- A developer can belong to multiple orgs and spaces.

Reference:

https://docs.cloudfoundry.org/concepts/roles.html

IBM Training

Orgs

- *Org* is a level of abstraction to manage resources, such as service availability, quota plans, applications and custom domains for multiple users.
- With a free IBM Cloud account (Lite), you are entitled to a maximum of one org.
- When working within a team or a company, a logical mapping (of an org) might be to your business unit, an application, or to your team.
- Collaborators in an org share a resource quota plan, applications, services availability, and custom domains.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-20. Orgs

- *Org* is a level of abstraction to manage resources, such as service availability, quota plans, applications and custom domains for multiple users.
- With a free IBM Cloud account (ILte), you are entitled to a maximum of one org.
- When working within a team or a company, a logical mapping (of an org) might be to your business unit, an application, or to your team.
- Collaborators in an org share a resource quota plan, applications, services availability, and custom domains.

Reference:

https://docs.cloudfoundry.org/concepts/roles.html

IBM Training

Spaces

- A *space* provides a shared location for multiple users to deploy multiple applications.
- Every space belongs to one org. Each org can have multiple spaces.
- Any developer in a space can access and edit the configurations of an application.
- Environment variables (memory per app, routes, number of instances, and app-specific variables) are contained within a space.
- Example: You can have multiple spaces, each mapped to an environment, such as dev, test, user-acceptance testing (UAT), and production environments.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-21. Spaces

- A *space* provides a shared location for multiple users to deploy multiple applications.
- Every space belongs to one org. Each org can have multiple spaces.
- Any developer in a space can access and edit the configurations of an application.
- Environment variables (memory per app, routes, number of instances, and app-specific variables) are contained within a space.
- As an example, you can have multiple spaces, each mapped to an environment, such as dev, test, user-acceptance testing (UAT), and production environments.

Reference:

https://docs.cloudfoundry.org/concepts/roles.html

3.4. Buildpacks

IBM Training

Buildpacks

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-22. Buildpacks

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-23. Topics

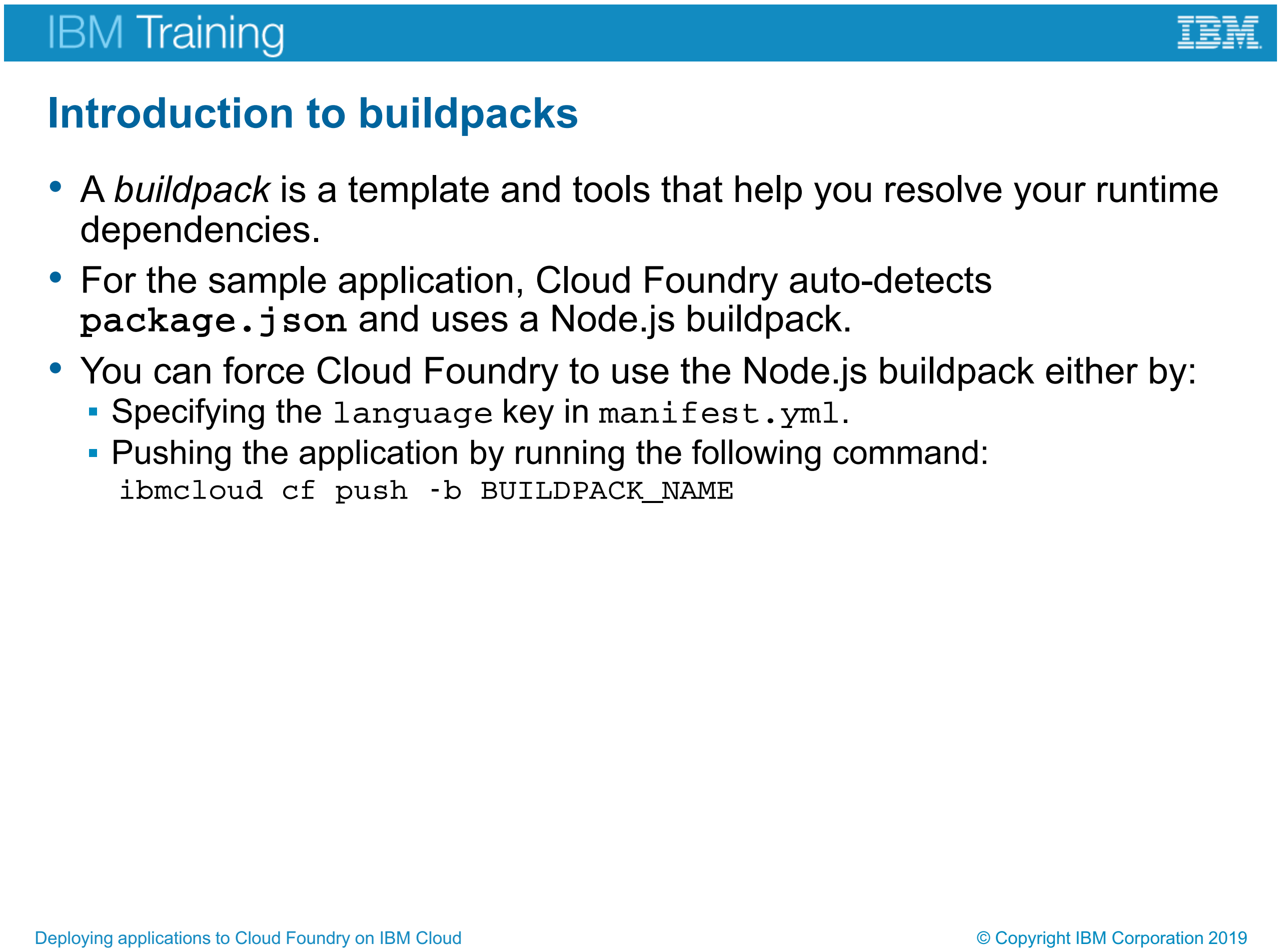


Figure 3-24. Introduction to buildpacks

- A *buildpack* is a template and tools that help you resolve your runtime dependencies.
- For the sample application, Cloud Foundry auto-detects **`package.json`** and uses a Node.js buildpack.
- You can force Cloud Foundry to use the Node.js buildpack either by:
  - Specifying the `language` key in `manifest.yml`.
  - Pushing the application by running the following command:

    ```
    ibmcloud cf push -b BUILDPACK_NAME
    ```
- Dependencies are your runtime dependencies, that is, the programs and libraries that Node.js requires to run.

- When deploying an application, a buildpack:
  - **Detects** what language and tools are needed to run the application.
  - **Supplies** the dependencies.
  - **Finalizes** the app for launch.
  - **Releases** the app by assigning the environment variables and running the start commands.

Reference:

https://docs.cloudfoundry.org/buildpacks/understand-buildpacks.html

IBM Training

IBM Cloud provided buildpacks

To get a list of the buildpacks that are available on IBM Cloud, run the following command:

`ibmcloud cf buildpacks`

```
Dev/redbook/get-started-node  master±  ibmcloud cf buildpacks
Invoking 'cf buildpacks'...

Getting buildpacks...

buildpack                              position   enabled   locked   filename                                                  stack
liberty-for-java                       1          true      false    buildpack_liberty-for-java_v3.30-20190325-1301.zip
sdk-for-nodejs                         2          true      false    buildpack_sdk-for-nodejs_v3.26-20190313-1440.zip
dotnet-core                            3          true      false    buildpack_dotnet-core_v2.2-20190327-1013.zip
swift_buildpack                        4          true      false    buildpack_swift_v2.0.18-20190303-1915.zip
noop-buildpack                         5          true      false    noop-buildpack-20140311-1519.zip
java_buildpack                         6          true      false    java-buildpack-v4.9.zip
ruby_buildpack                         7          true      false    ruby-buildpack-v1.7.15.zip
nodejs_buildpack                       8          true      false    nodejs-buildpack-v1.6.20.zip
go_buildpack                           9          true      false    go-buildpack-v1.8.20.zip
python_buildpack                       10         true      false    python-buildpack-v1.6.11.zip
xpages_buildpack                       11         true      false    xpages_buildpack_v1.2.2-20170112-1328.zip
php_buildpack                          12         true      false    php-buildpack-v4.3.51.zip
staticfile_buildpack                   13         true      false    staticfile-buildpack-v1.4.24.zip
binary_buildpack                       14         true      false    binary-buildpack-v1.0.17.zip
liberty-for-java_v3_17_1-20180131-1532 15         true      false    buildpack_liberty-for-java_v3.17.1-20180131-1532.zip
liberty_v3_14-20171013-1023            16         true      false    buildpack_liberty_v3.14-20171013-1023.zip
swift_buildpack_v2_0_17-20190212-2123  17         true      false    buildpack_swift_v2.0.17-20190212-2123.zip
swift_buildpack_v2_0_18-20190303-1915  18         true      false    buildpack_swift_v2.0.18-20190303-1915.zip
sdk-for-nodejs_v3_25_1-20190115-1637   19         true      false    buildpack_sdk-for-nodejs_v3.25.1-20190115-1637.zip
sdk-for-nodejs_v3_26-20190313-1440     20         true      false    buildpack_sdk-for-nodejs_v3.26-20190313-1440.zip
liberty-for-java_v3_29-20190223-2128   21         true      false    buildpack_liberty-for-java_v3.29-20190223-2128.zip
liberty-for-java_v3_30-20190325-1301   22         true      false    buildpack_liberty-for-java_v3.30-20190325-1301.zip
dotnet-core_v2_1-20181205-1536         23         true      false    buildpack_dotnet-core_v2.1-20181205-1536.zip
dotnet-core_v2_2-20190327-1013         24         true      false    buildpack_dotnet-core_v2.2-20190327-1013.zip
```

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-25. IBM Cloud provided buildpacks

To get a list of the buildpacks that are available on IBM Cloud, run the following command:

`ibmcloud cf buildpacks`

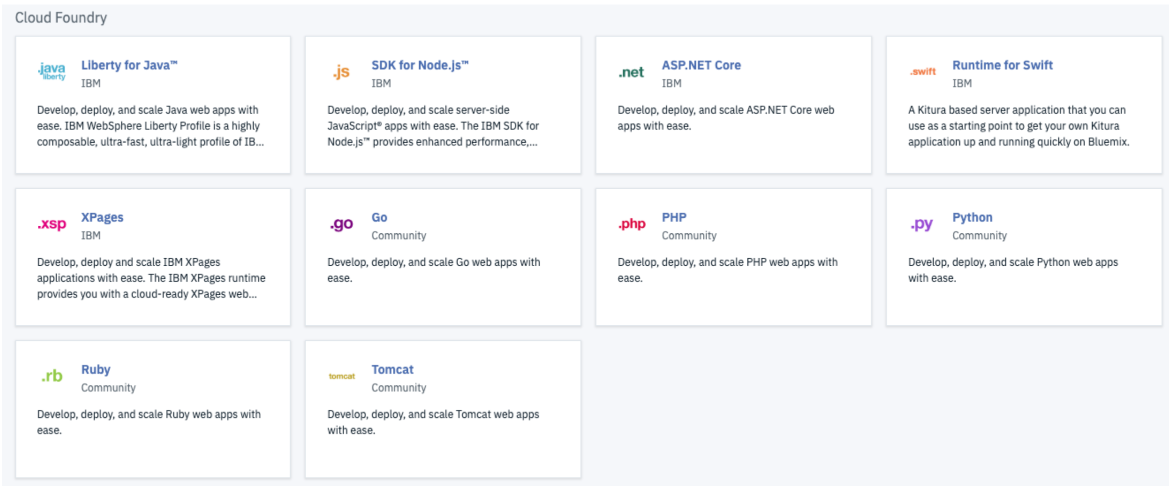
Reference:

https://docs.cloudfoundry.org/buildpacks/understand-buildpacks.html

IBM Training

IBM Cloud catalog view of buildpacks

IBM Cloud provides a UI catalog of the previously mentioned buildpacks.



Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-26. IBM Cloud catalog view of buildpacks

This slide shows a list of sample Cloud Foundry applications that is prepared from the list of buildpacks in the previous slide with more defaults, such as memory size and routing attributes.

For example, the Liberty for Java tile corresponds to the “liberty-for-java” buildpack.

IBM Training

Using custom buildpacks

- Aside from IBM buildpacks, you can also use community buildpacks and even create your own buildpacks.
- To use a custom buildpack, you can either:
  - Add the `buildpack` key into the application's `manifest.yml` file

```
---
applications:
- name:
  memory: 128M
  buildpack: GIT_BUILDPACK_URL
```

  - Specify a Git URL to the buildpack, for example:
    `ibmcloud cf push APP_NAME -b https://github.com/cloudfoundry/java-buildpack.git`

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-27. Using custom buildpacks

- Aside from IBM buildpacks, you can also use community buildpacks and even create your own buildpacks.
- To use a custom buildpack, you can either:
  - Add the buildpack key to the application's manifest.yml file
  - Specify a Git URL to the buildpack, for example:

    ibmcloud cf push APP_NAME -b https://github.com/cloudfoundry/java-buildpack.git
- You can use community buildpacks or a custom buildpack if IBM Cloud does not provide a buildpack that is specific to your needs.

Reference:

https://docs.cloudfoundry.org/buildpacks/understand-buildpacks.html

3.5. Resiliency

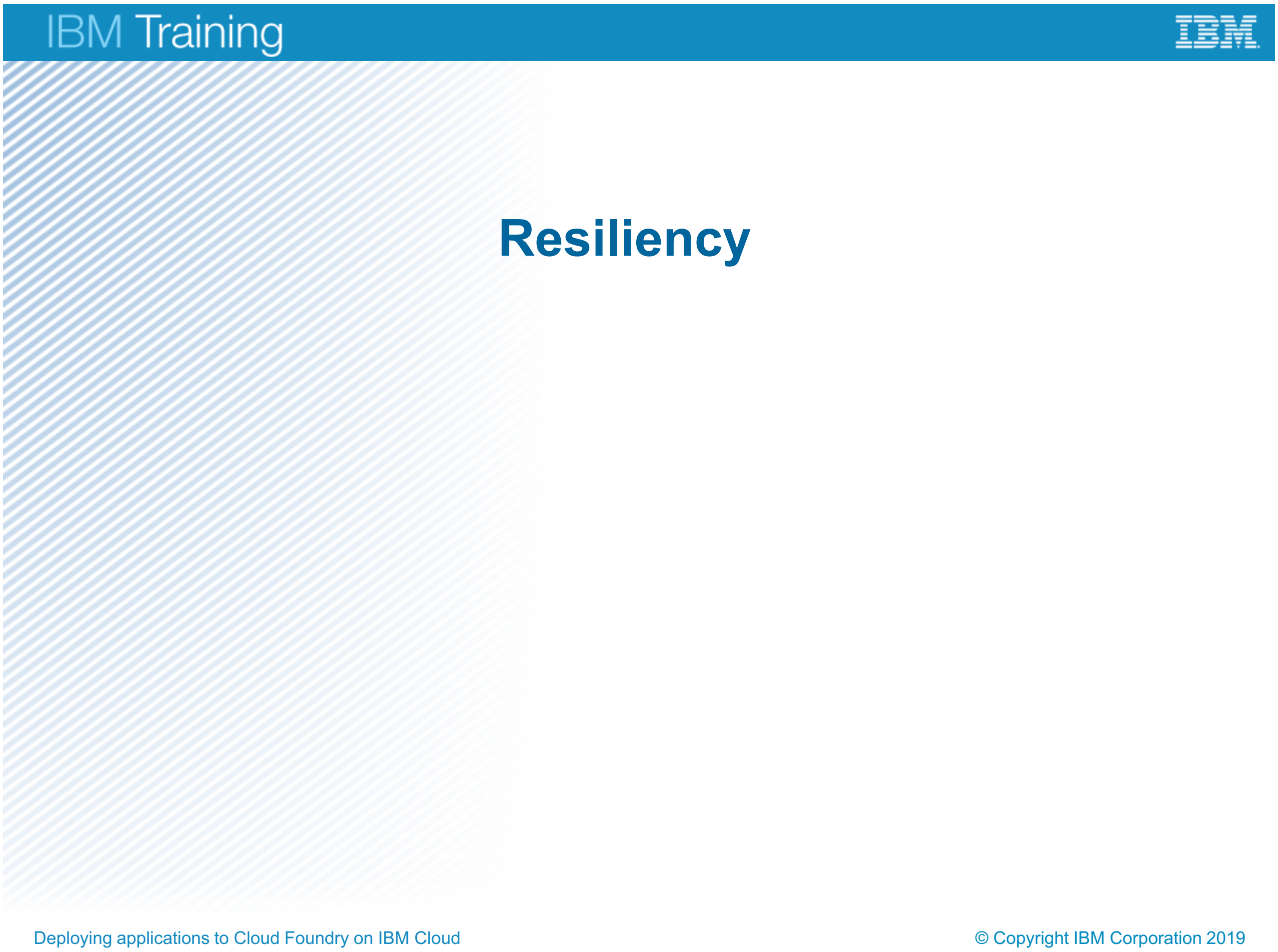


Figure 3-28. Resiliency

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-29. Topics

IBM Training

Making your app resilient

- Resiliency is about ensuring that the actual system state (number of running applications) matches the wanted state always.
- To configure the number of instances that you want in `manifest.yml`, specify the number of instances in the `instances` key.
- In the event of a failure (failed system process, unresponsive containers, and so on), Cloud Foundry kills or re-creates missing instances to match the wanted state.

```
applications:
- name: myCustomApp
  memory: 128M
  instances: 2
```

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-30. Making your app resilient

- The slide shows that an app that is named myCustomApp is created, requesting Cloud Foundry to provision two instances of this app running the same code, and allocating 128 MB of memory per instance.
- Resiliency is about ensuring that the actual system state (number of running applications) matches the wanted state always.
- To configure the number of instances that you want in `manifest.yml`, specify the number of instances in the `instances` key.
- In the event of a failure (failed system process, unresponsive containers, and so on), Cloud Foundry kills or re-creates missing instances to match the wanted state.

Reference:

https://docs.cloudfoundry.org/devguide/deploy-apps/cf-scale.html

IBM Training

Making your app resilient by using IBM Cloud CLI

- Aside from specifying the number of instances in `manifest.yml`, you can also use the IBM Cloud CLI to scale your application by running the following command:
  `ibmcloud cf scale APP_NAME -i NUMBER_OF_INSTANCES`
- However, it is recommended that you specify the number of wanted instances in your `manifest.yml` file to ensure that your configurations are stored as code.
- When you push your application, the value in your manifest file overrides any custom configuration.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-31. Making your app resilient by using IBM Cloud CLI

- Aside from specifying the number of instances in `manifest.yml`, you can also use the IBM Cloud CLI to scale your application by running the following command:

  `ibmcloud cf scale APP_NAME -i NUMBER_OF_INSTANCES`

- However, it is recommended that you specify the number of wanted instances in your `manifest.yml` file to ensure that your configurations are stored as code.
- When you push your application, the value in your manifest file overrides any custom configuration.

Reference:

https://www.ibm.com/blogs/bluemix/2018/11/infrastructure-as-code/

IBM Training

Making your app resilient by using IBM Cloud CLI (cont.)

- Example: `ibmcloud cf scale GetStartedNode -i 2`

```
† Dev/redbook/get-started-node  master± ibmcloud cf scale GetStartedNode -i 2
Invoking 'cf scale GetStartedNode -i 2'...

Scaling app GetStartedNode in org brew-house / space dev as [redacted]
OK
† Dev/redbook/get-started-node  master± ibmcloud cf app GetStartedNode
Invoking 'cf app GetStartedNode'...

Showing health and status for app GetStartedNode in org brew-house / space dev as [redacted]

name:              GetStartedNode
requested state:   started
routes:            getstartednode-patient-elephant.mybluemix.net
last uploaded:     Wed 10 Apr 10:37:04 AEST 2019
stack:             cflinuxfs2
buildpacks:        SDK for Node.js(TM) (node.js-6.17.0, buildpack-v3.26-20190313-1440)

type:           web
instances:      1/2
memory usage:   128M
     state      since                  cpu    memory        disk          details
#0   running    2019-04-10T23:53:33Z   0.5%   60M of 128M   80.9M of 1G
#1   starting   2019-04-10T23:55:13Z   0.0%   0 of 128M     0 of 1G
```

Deploying applications to Cloud Foundry on IBM Cloud © Copyright IBM Corporation 2019

Figure 3-32. Making your app resilient by using IBM Cloud CLI (cont.)

- Cloud Foundry manages application orchestration across multiple availability zones and infrastructure.
- If there are multiple instances of your application running, Cloud Foundry automatically load balances across instances.

Reference:

https://docs.cloudfoundry.org/devguide/deploy-apps/cf-scale.html

3.6. Logging and debugging

IBM Training

Logging and debugging

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-33. Logging and debugging

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-34. Topics

IBM Training

Debugging your application’s deployment

- The Cloud Foundry platform provides log aggregations.
- To view events and logs for the deployed application, run the following commands:
  - `ibmcloud cf events GetStartedNode`
  - `ibmcloud cf logs GetStartedNode`
- Additionally, you can use the IBM Cloud UI to look at the logs of your application.

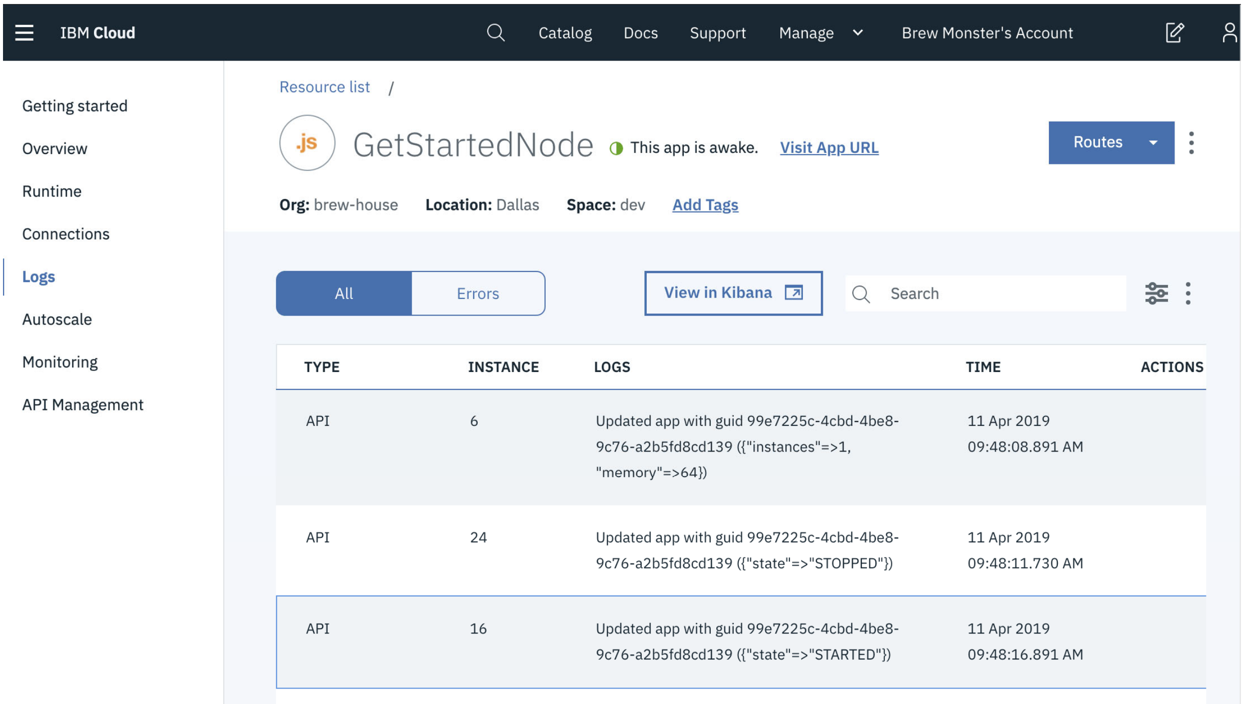
Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-35. Debugging your application’s deployment

- The Cloud Foundry platform provides log aggregations.
- To view events and logs for the deployed application, run the following commands:
  - ibmcloud cf events GetStartedNode
  - ibmcloud cf logs GetStartedNode
- Additionally, you can use the IBM Cloud UI to look at the logs of your application.

IBM Training

Viewing logs from the IBM Cloud dashboard



Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-36. Viewing logs from the IBM Cloud dashboard

- This screen shows the GetStartedNode Cloud Foundry app. This page can be navigated to from the dashboard by selecting **GetStartedNode** under **Cloud Foundry Apps** in the **Resource List**.
- This slide shows the logs that are generated by both the application and Cloud Foundry orchestrator.
- It is a GUI that uses the commands that are listed in the previous slide.

3.7. Domains and routes

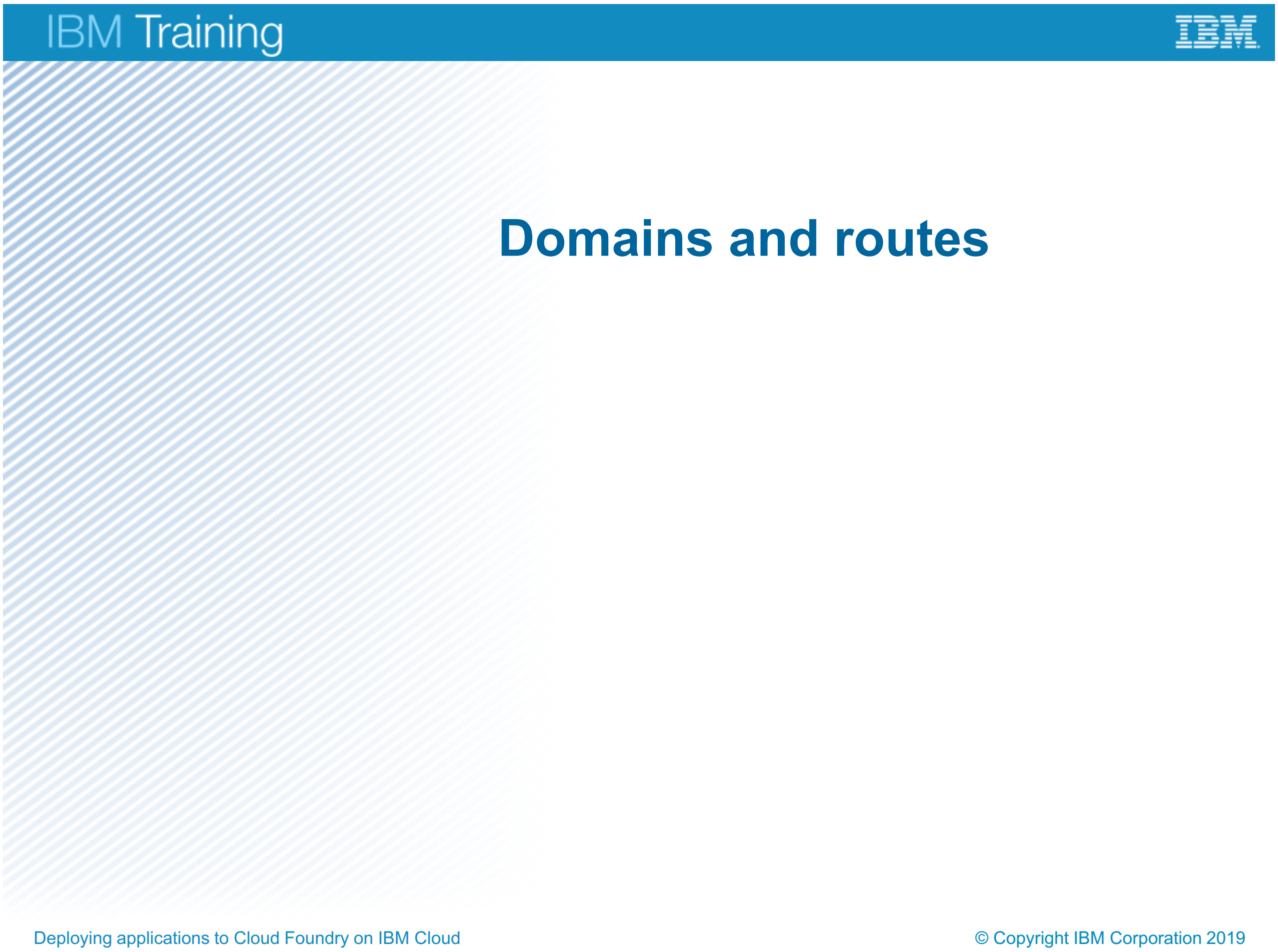


Figure 3-37. Domains and routes

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-38. Topics

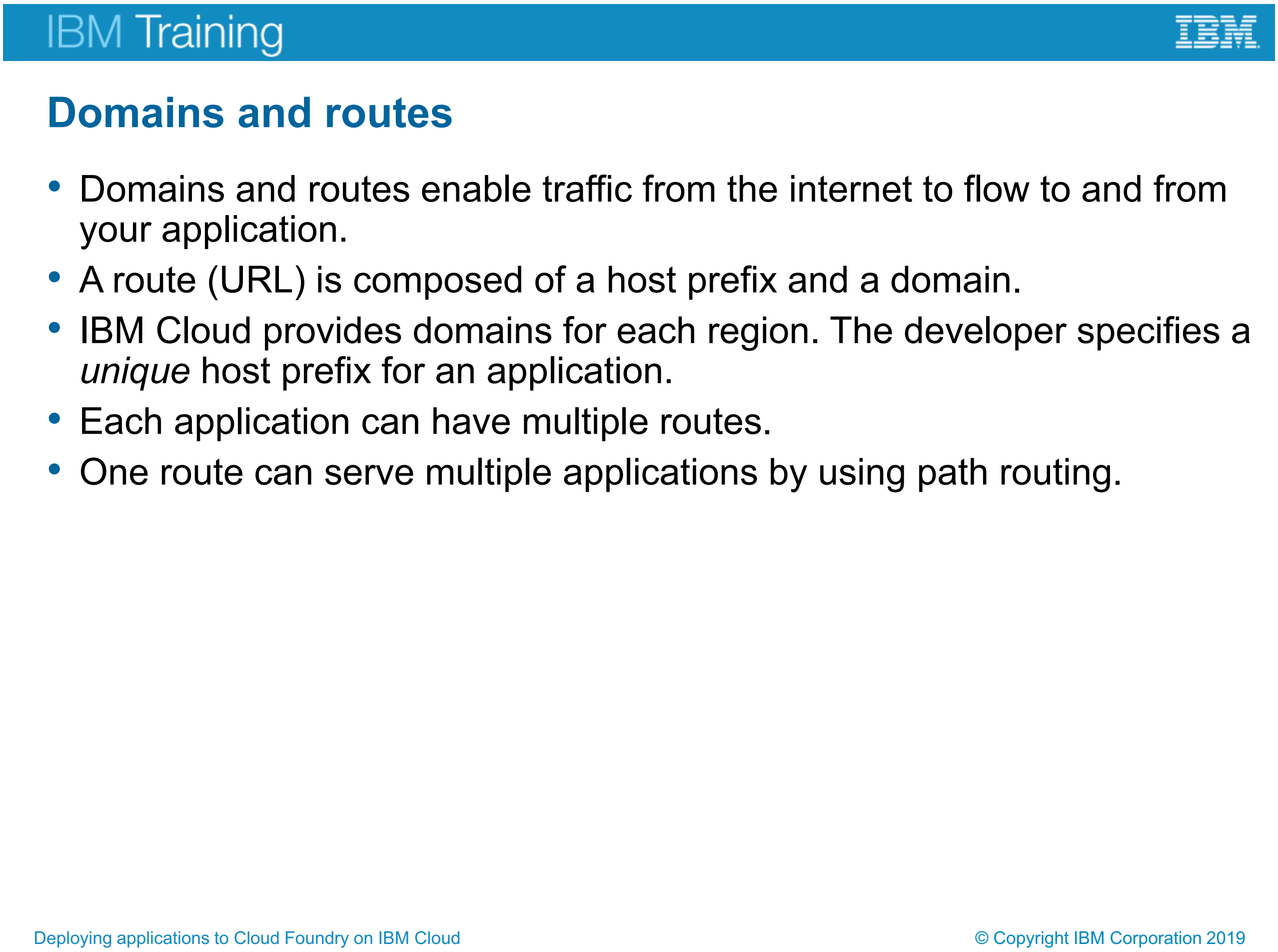


Figure 3-39. Domains and routes

- Domains and routes enable traffic from the internet to flow to and from your application.
- A route (URL) is composed of a host prefix and a domain.
- IBM Cloud provides domains for each region. The developer specifies a *unique* host prefix for an application.
- Each application can have multiple routes.
- One route can serve multiple applications by using path routing.

Here is an example of one application with two routes:

http://www.YourWebsite.com/calendar -> Calendar application

http://calendar.YourWebsite.com -> Calendar application

An example of path routing:

http://www.YourWebsite.com/calendar -> Calendar application

http://www.YourWebsite.com/mail -> Mail application

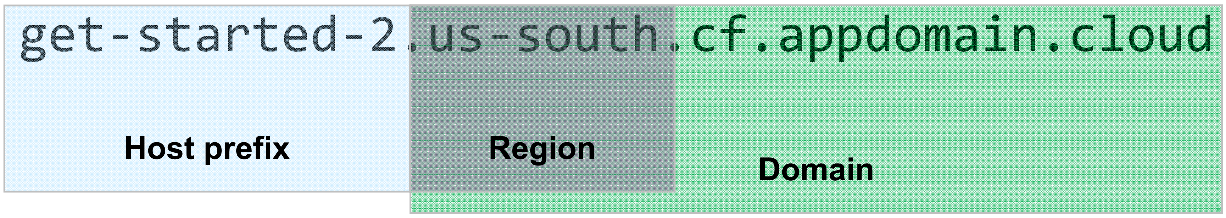
http://www.YourWebsite.com/todo -> Todo application

Reference:

https://docs.cloudfoundry.org/devguide/deploy-apps/routes-domains.html

IBM Training

Example of a route



Deploying applications to Cloud Foundry on IBM Cloud

© Copyright IBM Corporation 2019

Figure 3-40. Example of a route

IBM Training IBM

Setting up routes

- In the sample manifest, `random-route` is set to `true` to avoid duplicated host prefixes.
- To specify a route to an application, you can either:
  - Configure the route by using `manifest.yml`.

```
applications:
- name: GetStartedNode
  routes:
  - route: get-started-2.us-south.cf.appdomain.cloud
  memory: 128MB
```

  - Use the CLI with a unique host by running the following command:
    `ibmcloud app route-map APP_NAME DOMAIN -n HOST`

Deploying applications to Cloud Foundry on IBM Cloud © Copyright IBM Corporation 2019

Figure 3-41. Setting up routes

- In the sample manifest, `random-route` is set to `true` to avoid duplicated host prefixes.
- To specify a route to an application, you can either:
  - Configure the route by using `manifest.yml`.
  - Use the CLI with a unique host by running the following command:

    `ibmcloud app route-map APP_NAME DOMAIN -n HOST`
- Before creating a route with IBM Cloud CLI, you need:
  - The space in which the application is deployed.
  - The domain of your region (run “ibmcloud cf domains”).
- To check whether a route is available, run the following command:

  ibmcloud app route-check HOST DOMAIN [--path PATH]
- To create a route for later use, run the following command:

  ibmcloud app route-create SPACE DOMAIN -n HOST [--path PATH]
- To (create it if does not exist) map a route to an application, running the following command:

  ibmcloud app route-map APP_NAME DOMAIN -n HOST

- To unmap a route from an application, run the following command:
  ibmcloud app route-unmap APP_NAME DOMAIN -n HOST
- To delete a route, run the following command:
  ibmcloud app route-delete DOMAIN --hostname HOST [--path PATH] [-f]
- To delete all orphaned routes (not connected to any apps), run the following commands:
  ibmcloud app orphaned-routes-delete [-f]
  The -f flag is used to force deletion.

Reference:

https://docs.cloudfoundry.org/devguide/deploy-apps/manifest-attributes.html

3.8. Binding external services

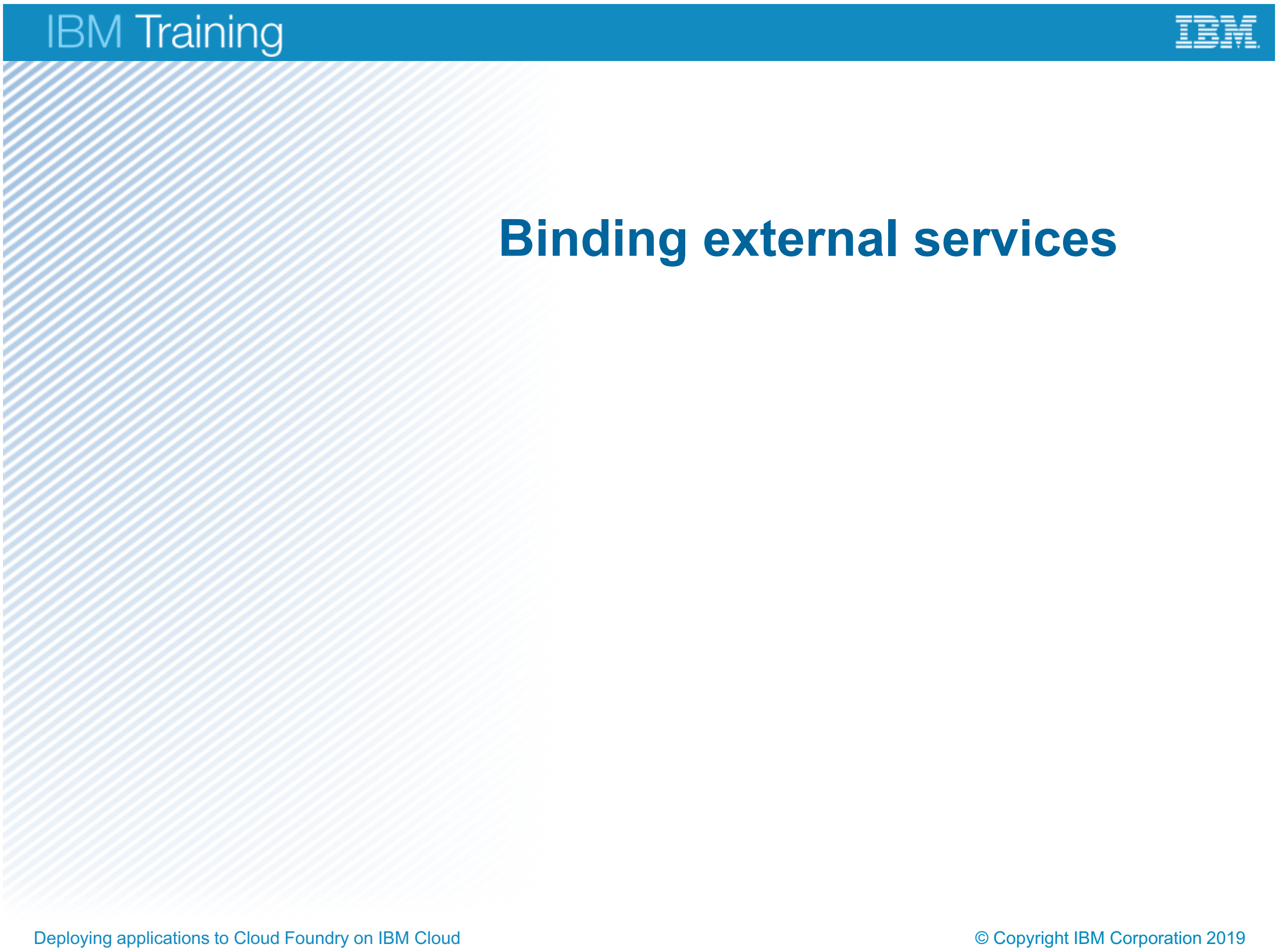


Figure 3-42. Binding external services

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-43. Topics

IBM Training

External services

- On Cloud Foundry, databases, file systems, messaging services, and any external systems with which your application interacts, are called *services*.
- These services enable you to do the following actions:
  - Horizontally scale.
  - Use existing marketplace services.
  - Reduce complexity and routing costs.
- For a list of services that are available on IBM Cloud, run the following command:
  - `ibmcloud service offerings`
  - Use the web UI to create services.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-44. External services

- On Cloud Foundry, databases, file systems, messaging services, and any external systems with which your application interacts, are called *services*.
- These services enable you to do the following actions:
  - Horizontally scale.
  - Use existing marketplace services.
  - Reduce complexity and routing costs.
- For a list of services that are available at IBM Cloud, run the following command:

  `ibmcloud service offerings`

  Use the web UI to create services.

References:

https://docs.cloudfoundry.org/devguide/services/

https://cloud.ibm.com/docs/cli/reference/ibmcloud?topic=cloud-cli-ibmcloud_commands_services#ibmcloud_commands_services.

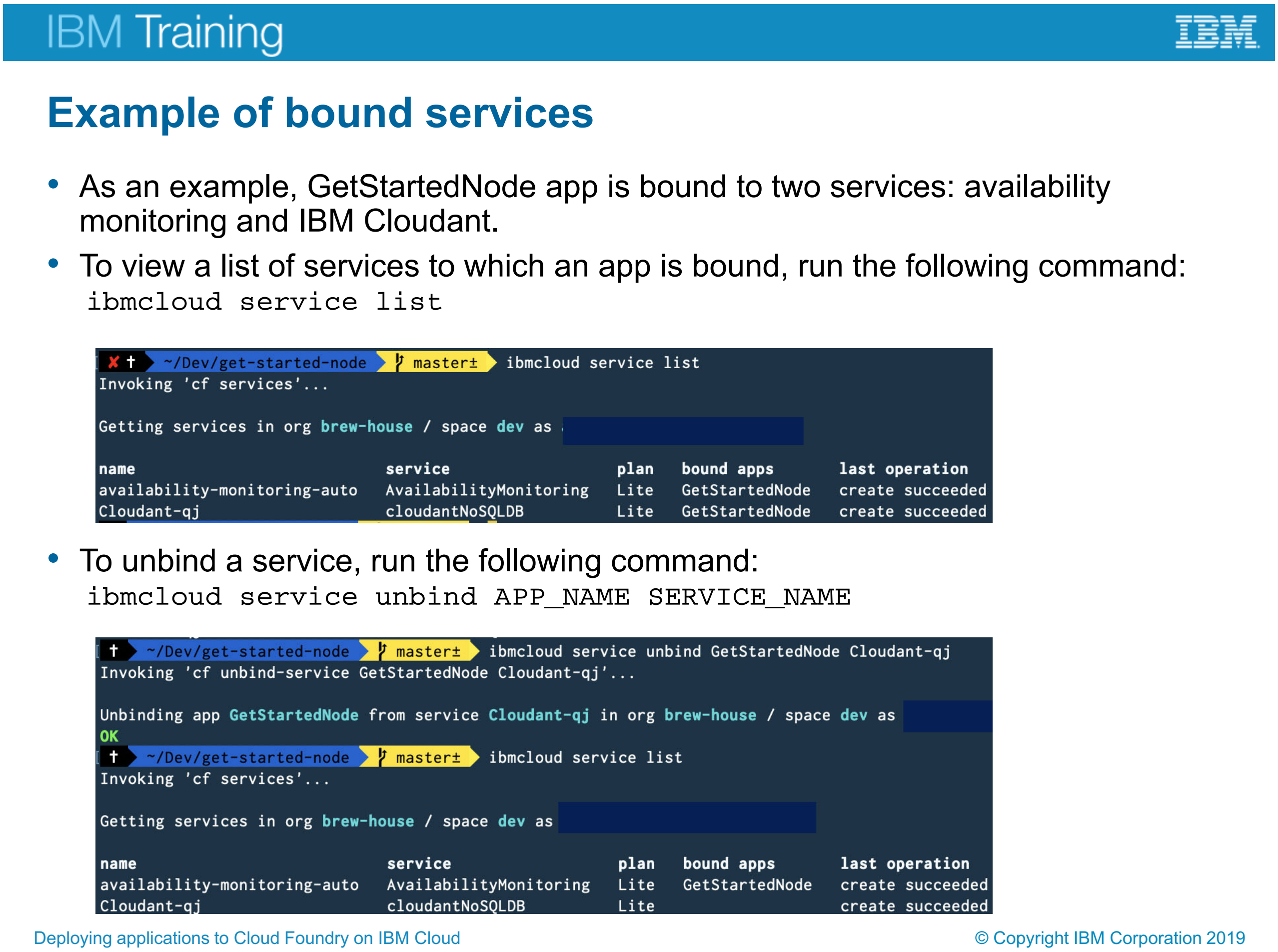


Figure 3-45. Example of bound services

- As an example, GetStartedNode app is bound to two services: availability monitoring and IBM Cloudant.
- To view a list of services to which an app is bound, run the following command:

  `ibmcloud service list`

- To unbind a service, run the following command:

  `ibmcloud service unbind APP_NAME SERVICE_NAME`

Reference:

https://docs.cloudfoundry.org/devguide/services/managing-services.html

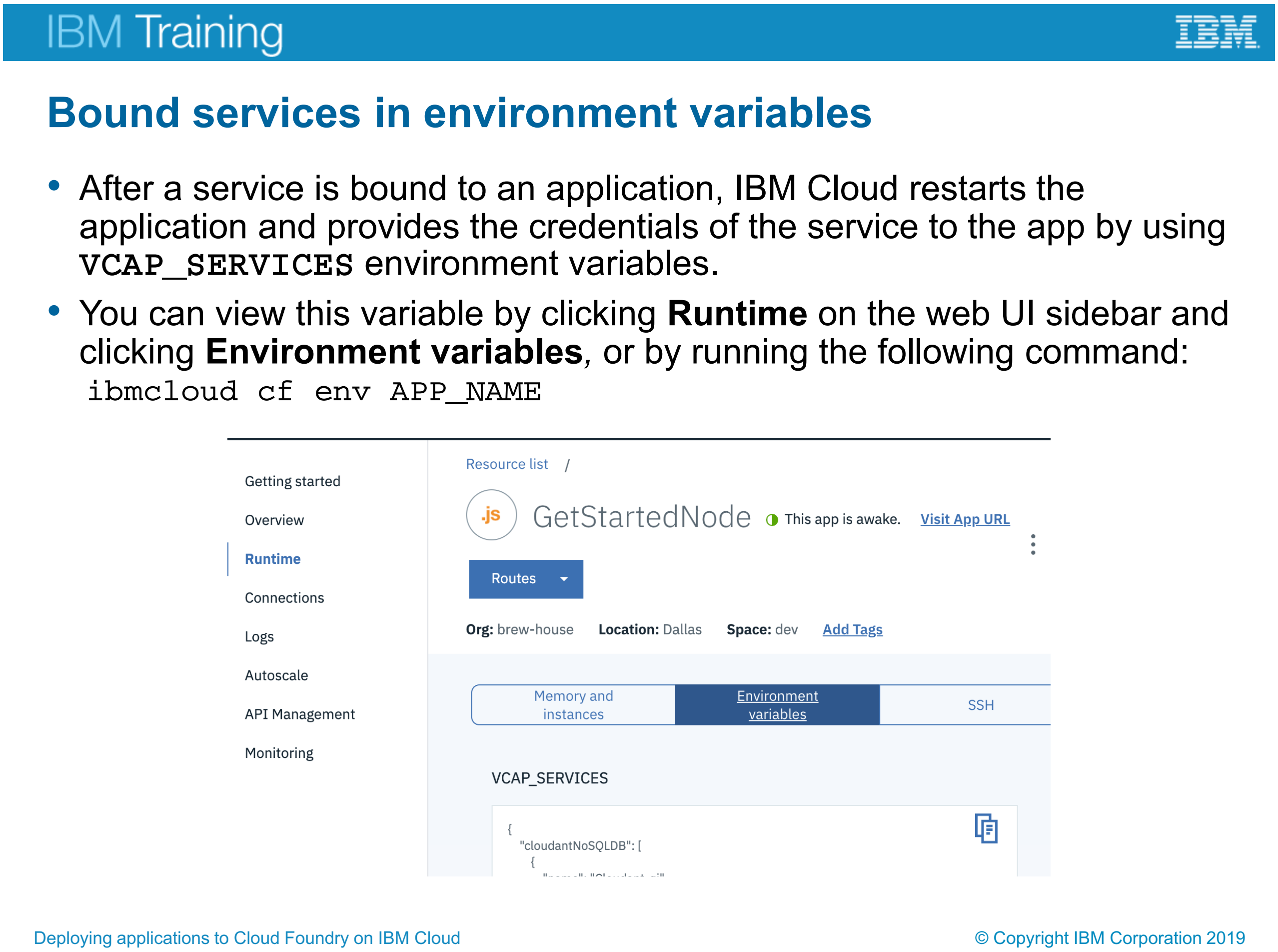


Figure 3-46. Bound services in environment variables

- After a service is bound to an application, IBM Cloud restarts the application and provides the credentials of the service to the app by using **VCAP_SERVICES** environment variables.
- You can view this variable by clicking **Runtime** on the web UI sidebar and clicking **Environment variables**, or by running the following command:

```
ibmcloud cf env APP_NAME
```

Reference:

https://docs.cloudfoundry.org/devguide/services/managing-services.html (search for VCAP_SERVICES)

3.9. Next steps

IBM Training

Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-47. Next steps

IBM Training

Topics

- Introduction to Cloud Foundry
- Deploying Cloud Foundry applications with IBM Cloud CLI
- Organizations and spaces
- Buildpacks
- Resiliency
- Logging and debugging
- Domains and routes
- Binding external services
- Next steps

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-48. Topics

IBM Training

Further reading

- In this unit, you learned the basic concepts and tenets of Cloud Foundry.
- To become better versed with this technology, see the following resources:
  - All attributes of `manifest.yml`: https://docs.cloudfoundry.org/devguide/deploy-apps/manifest-attributes.html
  - App container lifecycle: https://docs.cloudfoundry.org/devguide/deploy-apps/app-lifecycle.html
  - How to deploy Docker images onto Cloud Foundry: https://docs.cloudfoundry.org/devguide/deploy-apps/push-docker.html
  - Security and credential management with CredHub: https://docs.cloudfoundry.org/credhub/index.html
  - BOSH: https://www.cloudfoundry.org/bosh/

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-49. Further reading

IBM Training

Unit summary

- Explain how to manage your IBM Cloud account with the IBM Cloud CLI.
- Describe how to create a Node.js application that runs on IBM Cloud.
- Deploy an application from a local workstation by using the IBM Cloud CLI.
- Describe the role of Node.js for server-side scripting.
- Deploy an application by using IBM Cloud App Service (Web Apps).

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-50. Unit summary

IBM Training

Review questions

1. What information is required to log in to your IBM Cloud account by using the IBM Cloud CLI?
   A. IBM ID, password, API endpoint
   B. IBM ID, password, space, route, and API endpoint.
   C. IBM ID, password, organization, space, and application name.
   D. IBM ID, password, organization, space, and 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.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-51. Review questions

IBM Training

Review questions (cont.)

3. To deploy to IBM Cloud, you need a deployment configuration `package.json` file:
   A. True.
   B. False.
4. A developer can belong to:
   A. Many Orgs and many Spaces.
   B. A single Org and many Spaces.
   C. Many Orgs and a single Space.
   D. A single Org and a single Space.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-52. Review questions (cont.)

IBM Training

Review questions (cont.)

5. You can specify Cloud Foundry to use the Node.js buildpack by:
   A. Specifying the `language` key in `manifest.yml`.
   B. Pushing the application with **`ibmcloud cf push -b BUILDPACK_NAME`**.
   C. Selecting the buildpack by using the IBM Cloud UI catalog.
   D. All of the above.
6. Which statement on routing is correct:
   A. Each application can have a single route and one route can serve multiple applications.
   B. Each application can have multiple routes and one route can serve multiple applications.
   C. Each application can have multiple routes and one route can serve a single application.
   D. Each application can have a single route and one route can serve a single application.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-53. Review questions (cont.)

IBM Training

Review answers

1. **A**. To successfully log in, you must enter your IBM ID, password and API endpoint. Your user name 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.
3. **B (False)**. You require a `manifest.yml` file to deploy to IBM Cloud. The `package.json` file is the Node.js configuration file and is not necessary for deploying a different run time.
4. **A**. A developer can belong to many Orgs and many Spaces. An Org is the top level of abstraction, such as a team or company. A Space provides a shared location for multiple users to deploy multiple applications.
5. **D**. All of the above.
6. **B**. A route, also know as a URL, is composed of a host prefix and a domain. Each application can have multiple routes. One route can serve multiple applications by using path routing.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-54. Review answers

IBM Training

Exercise 1: Getting started with Cloud Foundry apps on IBM Cloud

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-55. Exercise 1: Getting started with Cloud Foundry apps on IBM Cloud

IBM Training

Exercise objectives

- This exercise describes how you can deploy a web application (app) without downloading or configuring a runtime environment or framework or setting up a server. This exercise also covers how to test and run the app when it is deployed.
- After completing this exercise, you should be able to:
  - Log in to IBM Cloud from a browser.
  - Create an IBM Cloud application by using one of the available run times.
  - Install the IBM Cloud command-line interface (CLI).
  - Sign on to IBM Cloud from the CLI.
  - Deploy an application from a local workstation by using the IBM Cloud CLI.
  - Test the application with its endpoint after the application is deployed and started.

Deploying applications to Cloud Foundry on IBM Cloud

Figure 3-56. Exercise objectives

Unit 4. Adopting a DevOps approach by using IBM Continuous Delivery

Estimated time

01:30

Overview

This unit introduces the features and functions of the DevOps services in the cloud development platform, IBM Cloud

IBM Training

Unit objectives

- Describe DevOps.
- Describe the capabilities of IBM Cloud Continuous Delivery.
- Identify the web-based integrated development environment (Web IDE) features in IBM Cloud Continuous Delivery.
- Describe how to use source code management (such as Git) and Issue tracking.
- Explain how to build and deploy applications using DevOps tools on IBM Cloud.

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-1. Unit objectives

4.1. Introduction to DevOps

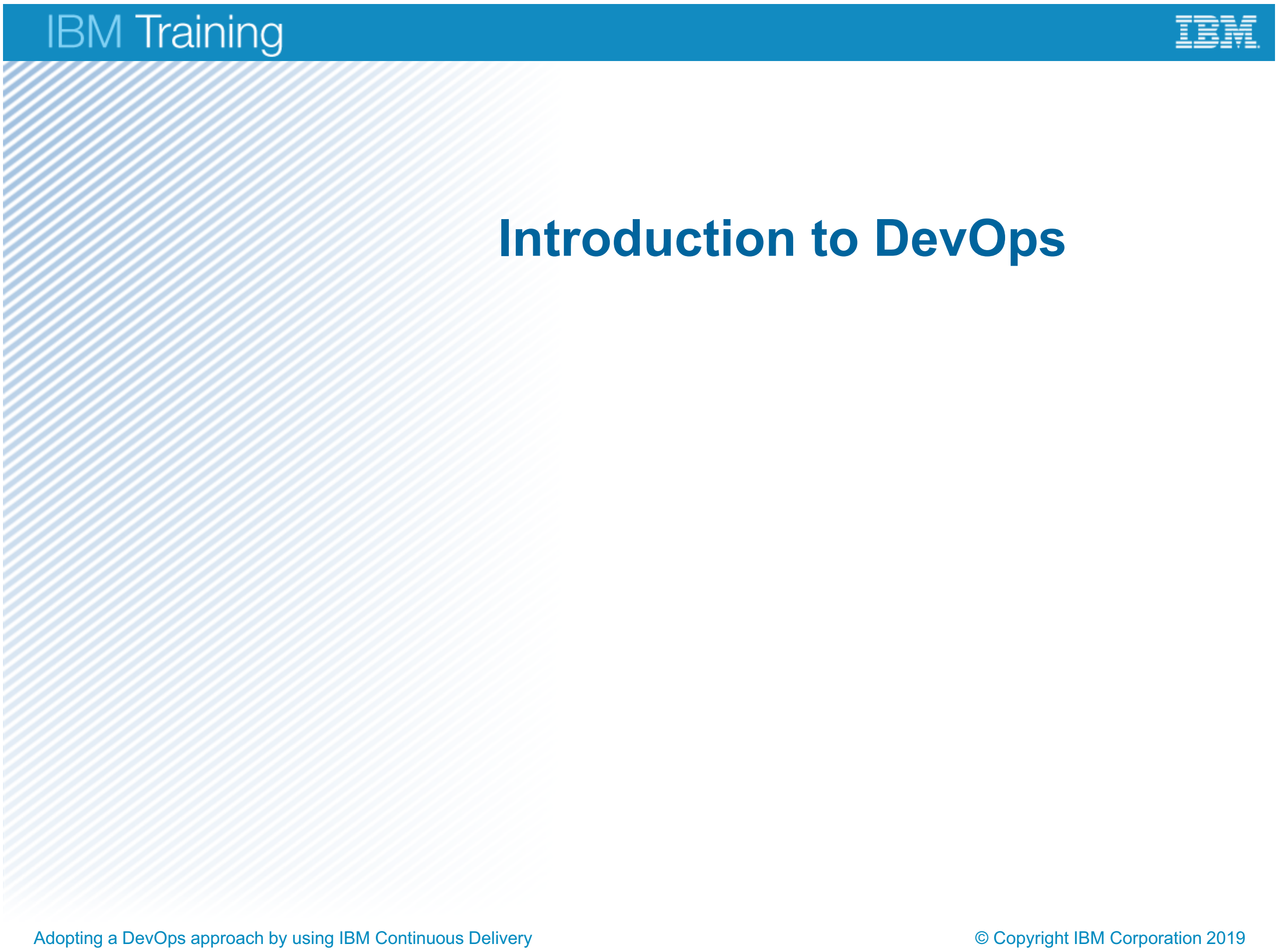


Figure 4-2. Introduction to DevOps

IBM Training

Topics

- Introduction to DevOps
- DevOps services on IBM Cloud
- IBM Cloud Continuous Delivery overview
- Web IDE (Edit Code)
- Source Code Management (Git repository) and Issue Tracker
- Automated build and deployment (Delivery Pipeline)

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-3. Topics

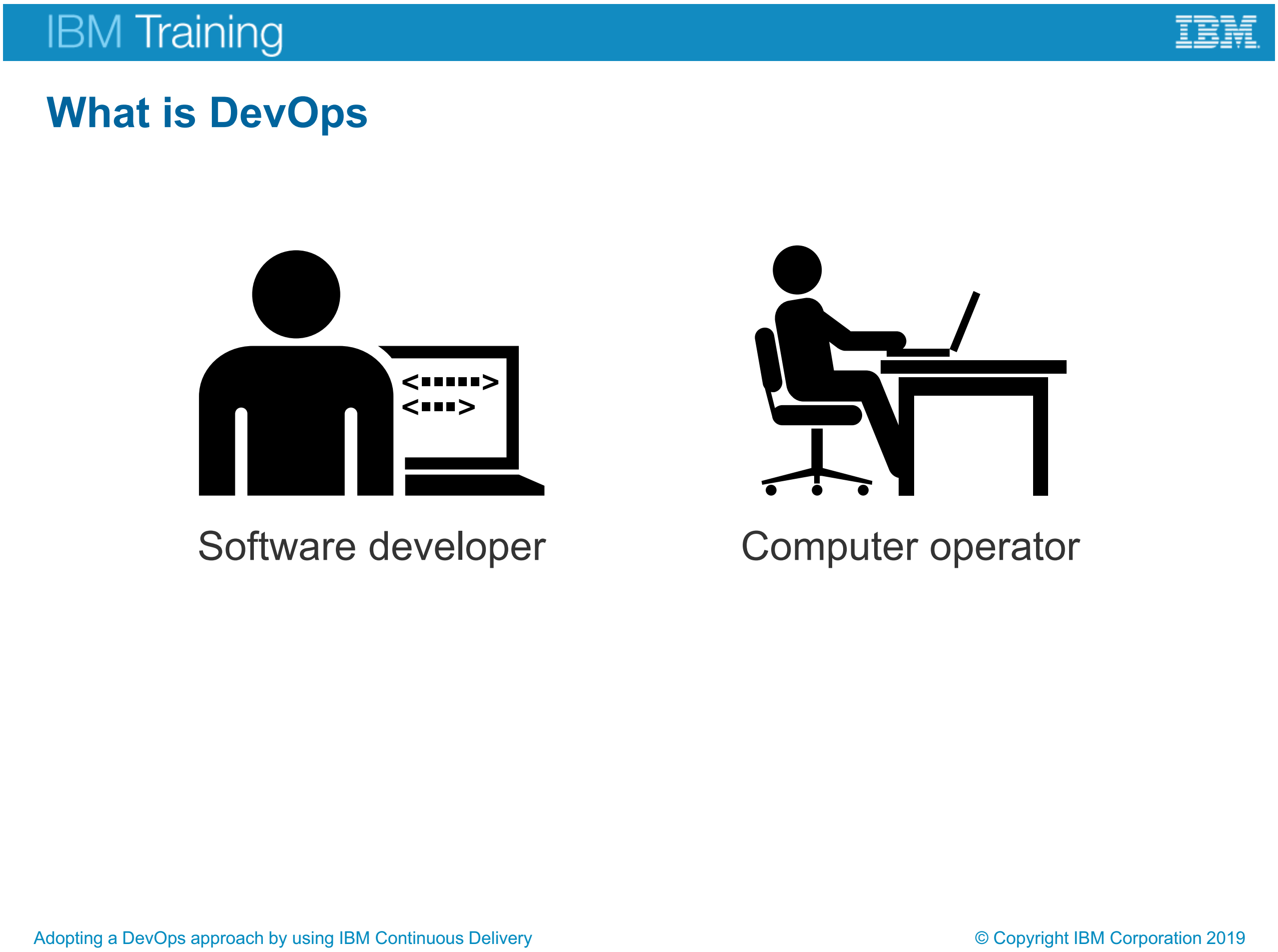


Figure 4-4. What is DevOps

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 production uptime, and diagnosed failures.

The process of delivering any business requirements from code to the production environment was a tough process that took days to months to complete. It might have taken a year or more to plan for major changes to the software. Both development and operation teams focused on their tasks separately without involving the other team, which caused bottlenecks and miscommunication between the two teams. This process led to a delayed delivery of the business requirements, and made it hard to maintain competitive parity with competitors.

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. Automation plays a great role in DevOps, to make the collaboration easy.

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 improve them. DevOps thus allows organizations to deliver products which are better, easier to maintain and can be delivered faster to the market.

References:

- DevOps For Dummies®, 3rd IBM Limited Edition, Chapter 1, found at:

  https://www.ibm.com/downloads/cas/P9NYOK3B
- Understanding DevOps, found at:

  https://www.youtube.com/watch?v=EIvOZA1HgHU

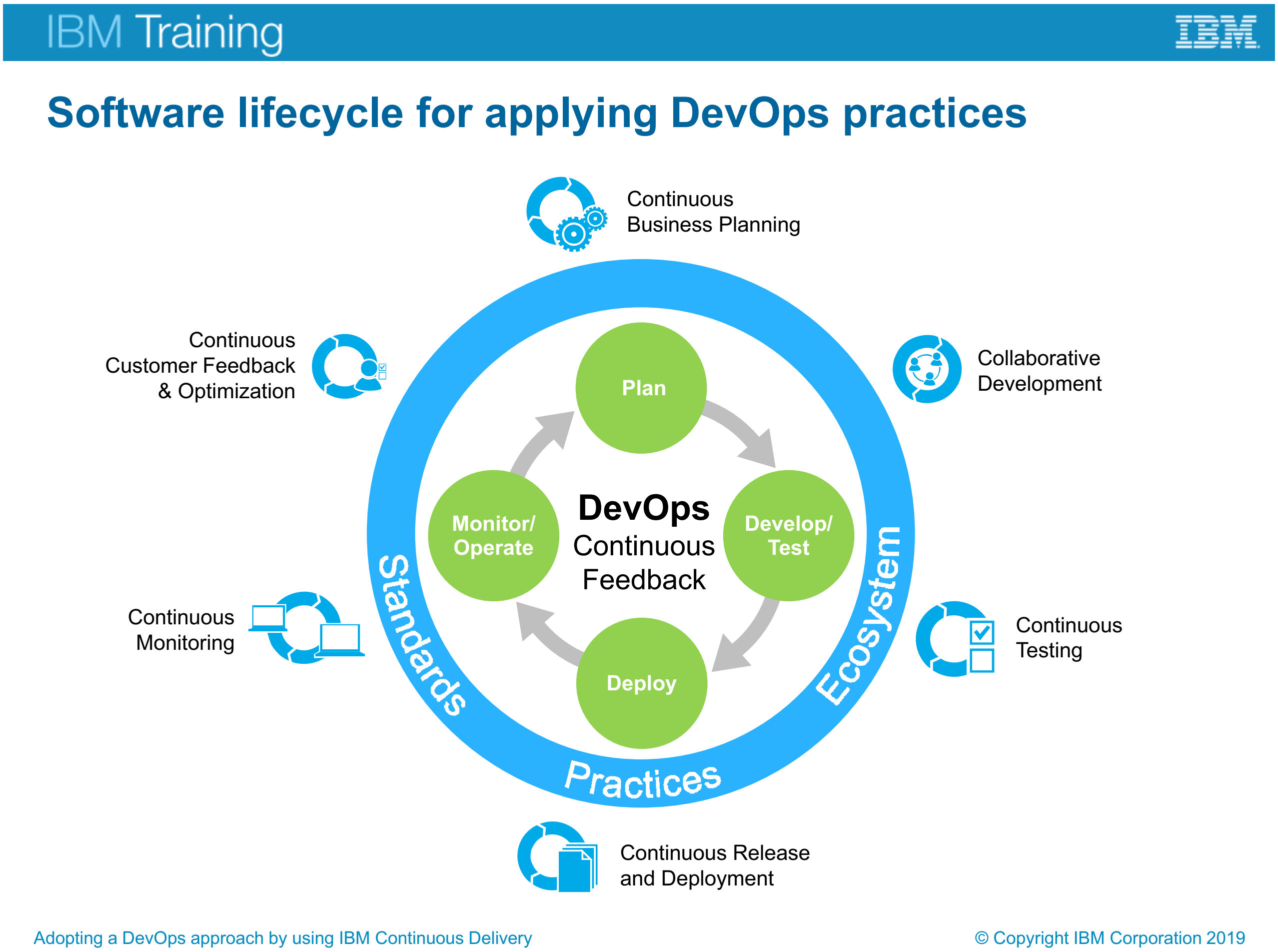


Figure 4-5. Software lifecycle for applying DevOps practices

The figure shows the software lifecycle for applying DevOps practices:

1. Planning: In this phase, the team is responsible of defining the business requirements, and then defining the required resources and timeline (sprint planning and releases).
2. Development and testing: This phase is the main implementation phase that may be divided into different sprints under different releases in agile development. Automated and continuous test management occurs by using integrated tools.
3. Deployment: This phase is when you deliver the software to the client by deploying it to the production environment. This phase is tightly linked with both Development and testing and Monitoring because of the DevOps focus on continuous and rapid delivery with automated deployment.
4. Monitoring: After deploying the application to production, the quality of the application is being analyzed to capture the enhancements and fixes for the delivered application in the monitoring phase. This phase helps to identify issues and to provide early warnings and feedback to the next planning and development cycles.

IBM Training

DevOps practices

- Continuous improvement
- Release planning
- Continuous integration
- Continuous delivery
- Continuous testing
- Continuous monitoring and feedback

Figure 4-6. DevOps practices

- *Continuous improvement* (continuous operations) for the operation processes is considered an important goal for organizations. Many businesses have process improvement teams that work on improving processes based on observations and lessons learned. Other businesses allow the teams that adopt the processes to self-assess and determine their own process improvement paths.
- *Release planning* (continuous business planning) is a critical business function that is driven by business needs to offer capabilities to customers and the timelines of these needs. Well-defined processes and automation enable frequent releases with better quality.
- *Continuous integration* enables developers to integrate their code frequently in an agile manner, which reduces risk and identifies issues earlier in the software development lifecycle.
- *Continuous delivery* is achievable by automating the deployment of software to the testing, system testing, staging, and production environments. Continuous integration leads to the practice of continuous delivery, which helps the business to receive frequent feedback and deliver frequent updates to maintain competitive parity. Automated builds, tests, and deployments are required for a *continuous delivery pipeline*.

- *Continuous testing* is achieved by using automated unit testing and integration testing, which increase the quality and the speed of development. Continuous testing must cover the following areas: *test environment provisioning and configuration*, *test data management and test integration*, *function*, *performance*, and *security*.
- *Continuous monitoring and feedback* are available because of continuous delivery. This frequent feedback is captured by using well-defined processes. These processes must be agile enough to adapt to market and regulatory changes.

Reference:

DevOps For Dummies®, 3rd IBM Limited Edition, Chapter 3, found at:

https://www.ibm.com/downloads/cas/P9NYOK3B

IBM Training

IBM Cloud Garage Method

Figure 4-7. IBM Cloud Garage Method

IBM Cloud Garage Method is a set of practices that can speed-up DevOps adoption by providing how-to guides on culture, best practices, tools, self-guided or hands-on training, and even sample code and reference architectures. Its practices are the IBM approach to enable business, development, and operations to design, deliver, and validate continuously new solutions. The practices and workflows cover the entire product lifecycle from inception through capturing and responding to customer feedback and market changes.

The practices are summarized as follows:

Culture

Culture is key to the success of the modern software development transformation such as agile methodology which is an iterative method of development where development life cycle is progressed through the collaboration between self organizing cross functional teams. The culture must support small and collocated teams that are autonomous and able to make decisions that are based on efficiency and knowledge.

Discover

Discover practices help teams to dig deep into problem domains, align everyone to common goals, and identify potential problems and bottlenecks.

Think

Think practices provide development teams with a repeatable approach to deliver rapidly innovative user experiences. To deliver a minimum viable product (MVP), teams must understand the user and identify the highest priority items.

Develop

Adopt DevOps development practices to help the team collaborate and produce high-quality code that can be confidently delivered to production. Many teams use pair programming to ensure real-time code review with the added advantage of cross-training the development team.

Reason

Build a solid information architecture and extract, cleanse, and curate data so that data can be turned into knowledge, regardless of the source of the data.

Operate

Ensure operational excellence by continuously monitoring app status and performance. Focus on building automation that enables high availability and resiliency while reducing the cost of the supporting infrastructure and the necessary resources to manage apps in production.

Learn

Learn from your customers and make better decisions. Learn how your team works together and how customers use the apps that you deliver by studying analytics data.

References

https://www.ibm.com/cloud/devops/get-started

https://www.ibm.com/cloud/garage/practices/overview

https://cloud.ibm.com/docs/services/ContinuousDelivery?topic=ContinuousDelivery-devops_intro

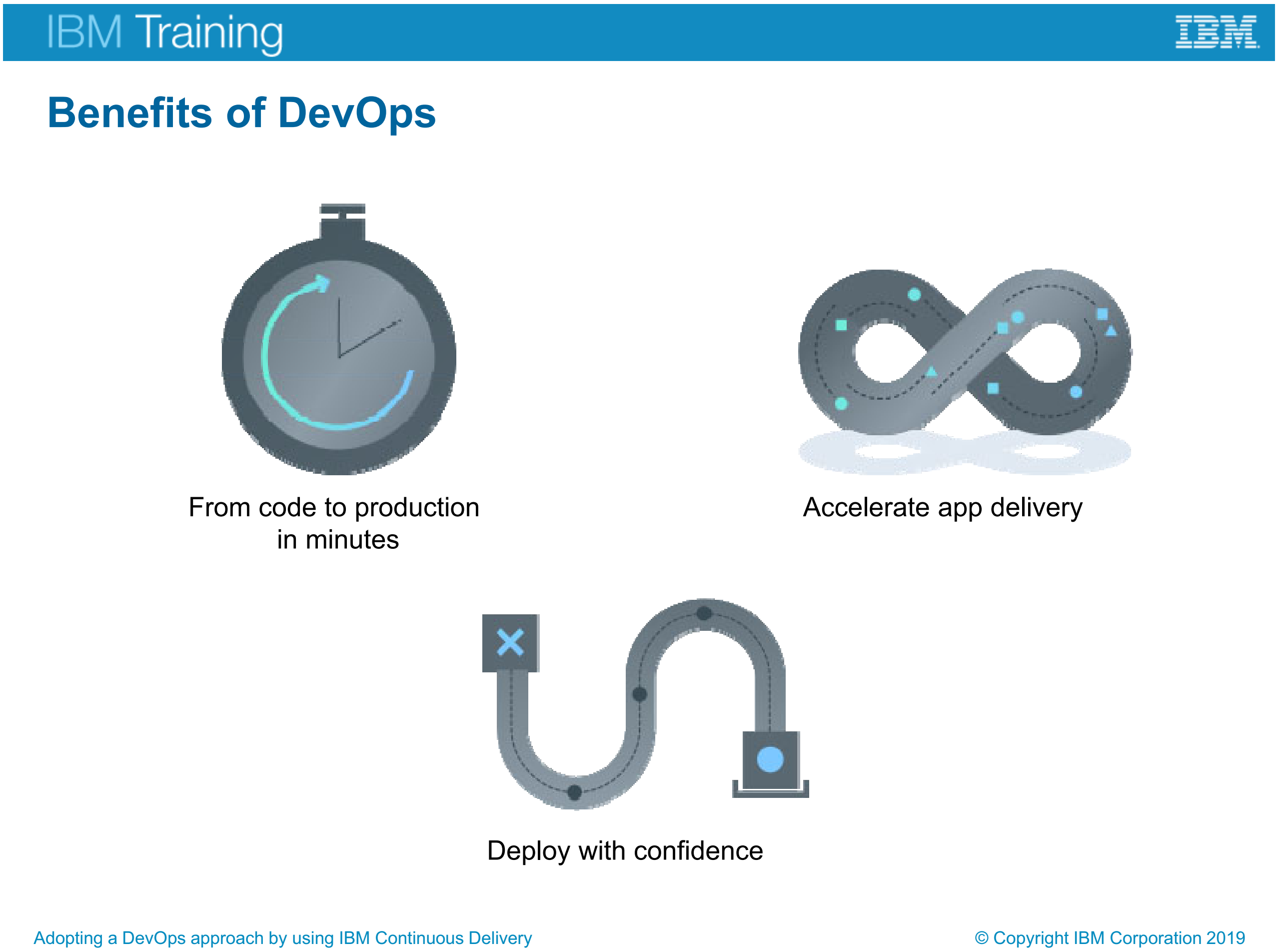


Figure 4-8. Benefits of DevOps

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:
  - Use the repeatability of the process by using automation, which ensures that the same steps are done every time that you deploy.
  - Use 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, and in many other cases automated tests could be even applied in the development phase before delivering to source code repositories.

IBM Training

Deployment strategies

- Blue/green deployment
- Canary deployment
- A/B testing

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-9. Deployment strategies

Due to the DevOps practices and agile methodologies, some new techniques to deploy new applications to production appeared recently, such as:

- Blue/green deployment:
- Canary deployment:
- A/B testing

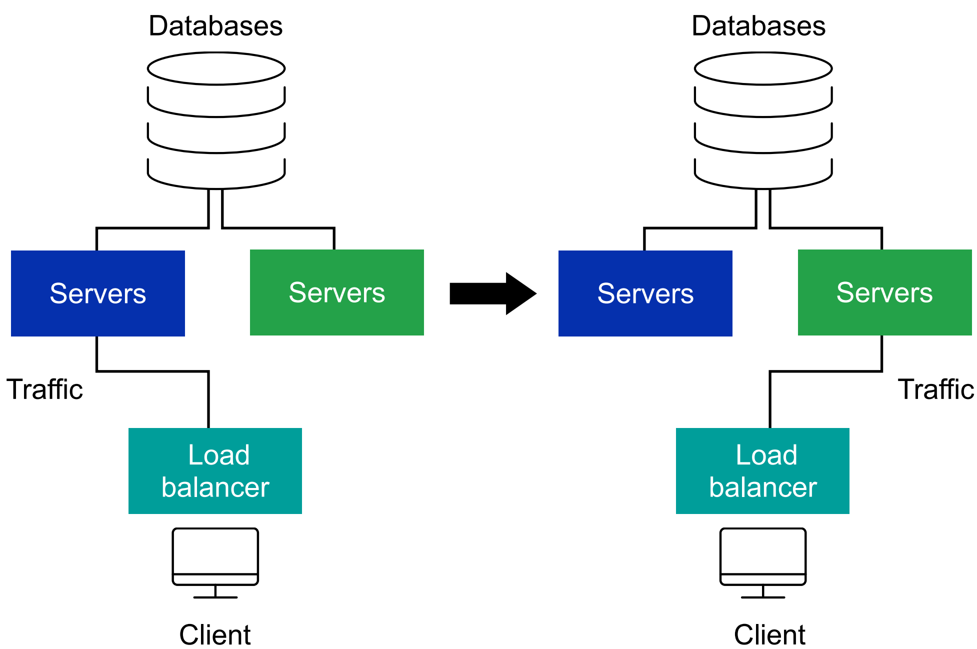
There are many tools that are provided by IBM to be able to apply these strategies such as **IBM UrbanCode Deploy**, **IBM Cloud Kubernetes Service**, **Istio** and **Jenkins**.

Check the following tutorials:

- Blue Green Zero Downtime Continuous Deployment with IBM UrbanCode and IBM Cloud Private https://www.ibm.com/cloud/garage/dte/tutorial/blue-green-zero-downtime-continuous-deployment-ibm-urbancode-and-ibm-cloud-private/
- Canary deployment with Jenkins and Istio https://developer.ibm.com/tutorials/use-jenkins-and-istio-for-canary-deployment/

IBM Training

Deployment strategies – Blue/green deployment



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-10. Deployment strategies – Blue/green deployment

Blue/green deployment

a. Deploy an instance of your updated application “green” while leaving the production or old environment “blue” as is.

b. Run smoke tests on the new environment “green” to verify that it is working.

c. Switch over the production traffic while keeping the old environment as a hot backup.

The new environment should be pointed to the same database and use the same data set to avoid data loss or inconsistency.

The figure shows how the blue/green deployment strategies are applied and how the traffic is switched from the old “blue” environment to the new “green” environment.

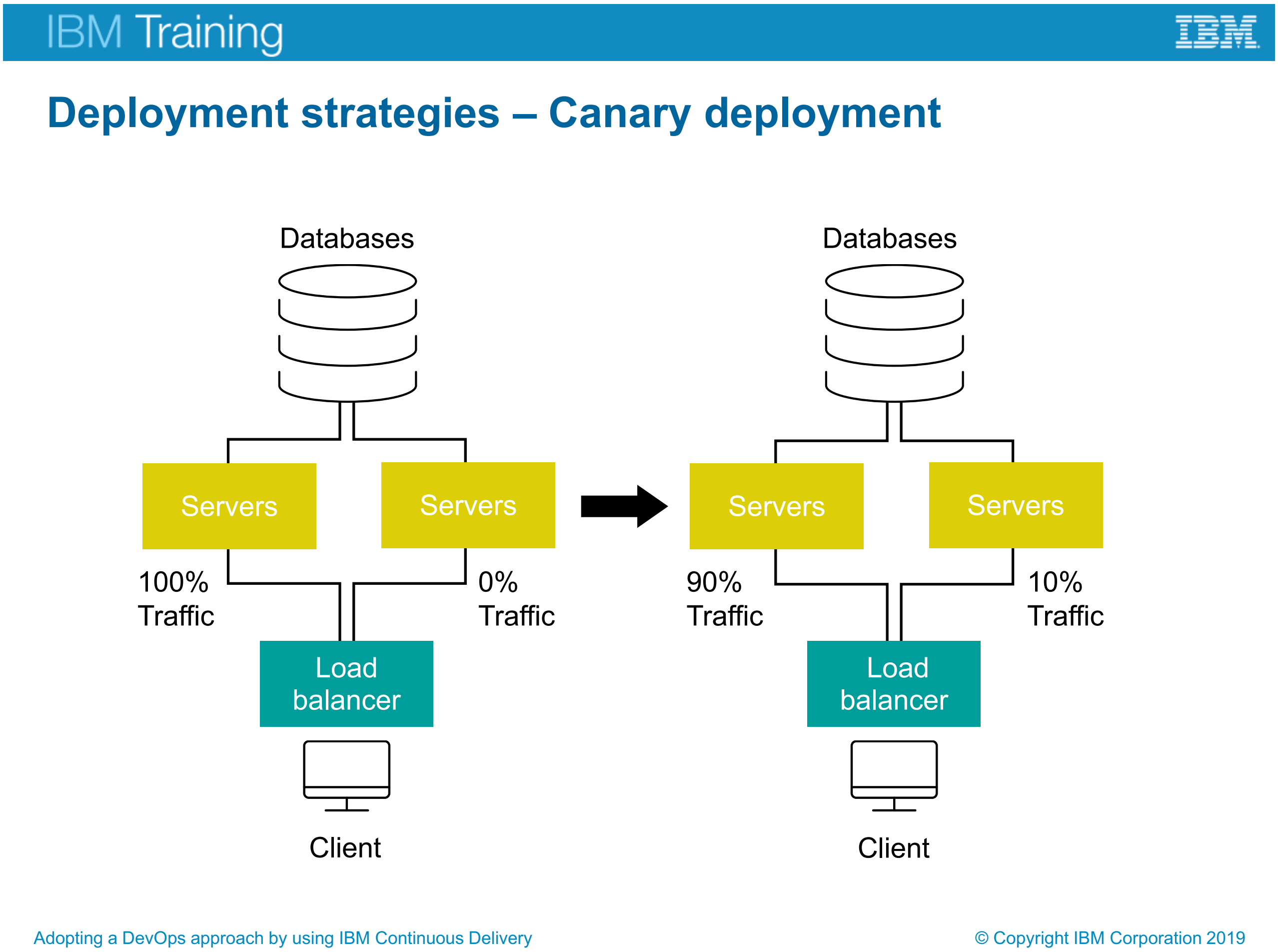


Figure 4-11. Deployment strategies – Canary deployment

Canary deployment

a. Deploy an instance of your application, and split the traffic based on the weight between the old and the new environment.

b. Run smoke tests on the new environment to verify that it is working.

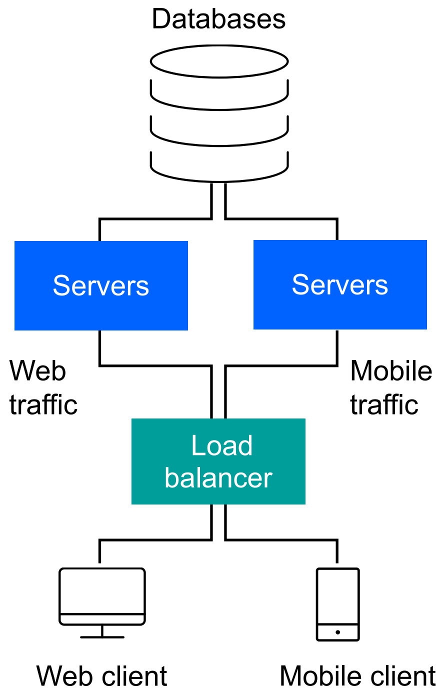
c. Gradually shift production traffic from the old environment to the new one.

Similar to the blue/green deployment, the two environments point to the same database and use the same data set to avoid data loss or inconsistency.

The figure shows how to split the traffic in the canary deployment. In this example, the old servers start by handling all the traffic, and then gradually, the traffic is shifted from the old to the new servers.

IBM Training

Deployment strategies – A/B testing



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-12. Deployment strategies – A/B testing

A/B testing

Routing a subset of users to a new version based on marketing strategies. This strategy is used to test various versions and determine which ones to keep and which ones to roll out.

This figure shows an example of the A/B testing strategy. In this example, the mobile requests are switched to target the new environment and the web requests keep targeting the old environment.

4.2. DevOps services on IBM Cloud

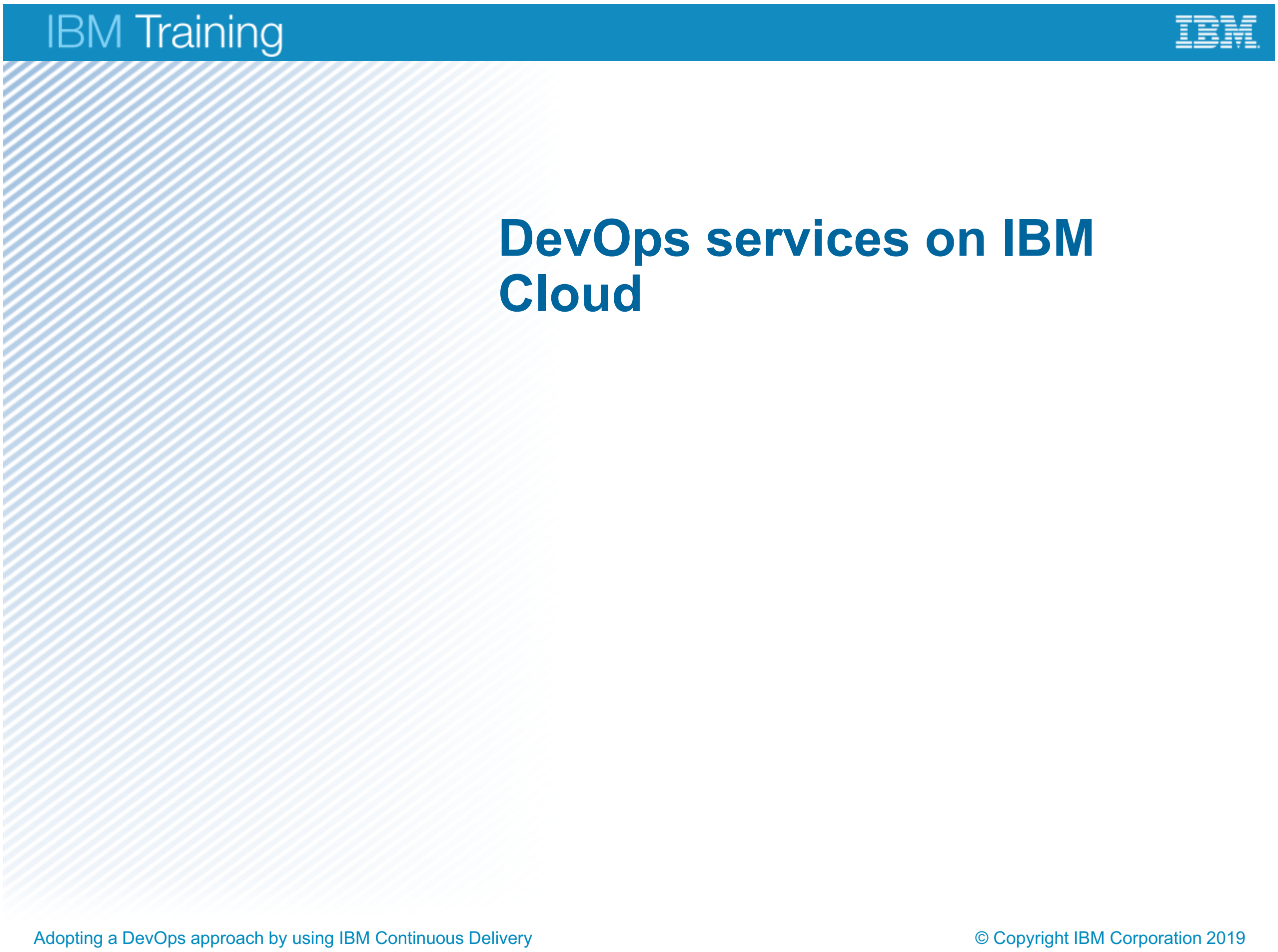


Figure 4-13. DevOps services on IBM Cloud

IBM Training

Topics

- Introduction to DevOps
- DevOps services on IBM Cloud
- IBM Cloud Continuous Delivery overview
- Web IDE (Edit Code)
- Source Code Management (Git repository) and Issue Tracker
- Automated build and deployment (Delivery Pipeline)

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-14. Topics

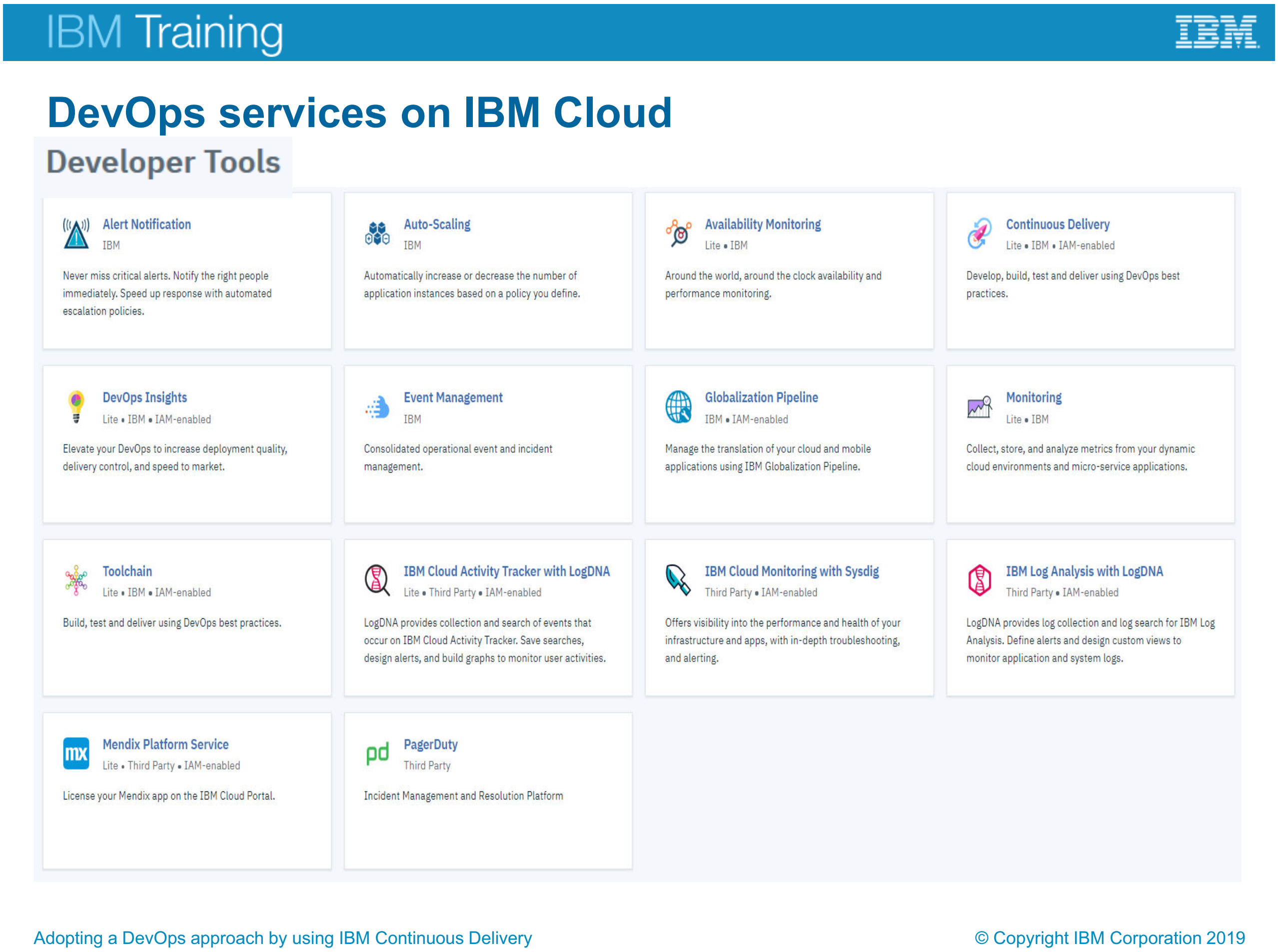


Figure 4-15. DevOps services on IBM Cloud

DevOps on IBM Cloud provides concrete practices and architectures for cloud development. It enables developers to get started quickly with new projects from the rich catalog of services on IBM Cloud. DevOps also provides developers an open and integrated set of tools for automating delivery with speed and control. These services are available on the IBM Cloud Catalog under the **Developer Tools** section.

Here are examples of the DevOps services that are available on IBM Cloud:

Alert Notification

This service is used to send early notifications of application or service issues before they affect the users. It can filter events to domain experts or support teams.

Auto-Scaling

The Auto-Scaling for IBM Cloud service enables the compute capacity of the applications to be automatically increased or decreased as demands on the application changes. The number of application instances are adjusted dynamically based on the Auto-Scaling policy that is defined by the developer.

Availability Monitoring

IBM Cloud Availability Monitoring helps DevOps teams ensure that their applications are always available and meeting user expectations for response time as they roll out continuous updates. The service, which is tightly integrated into the DevOps toolchain, runs synthetic tests from locations around the world and around the clock to detect and fix proactively performance issues before they impact users.

Continuous Delivery

Automate builds, unit tests, deployments, and more. Edit and push code by using Git Repos and Issue Tracking and the rich web-based IDE. Create toolchains to enable tool integrations that support the development, deployment, and operation tasks. This service is explained in more detail in the next section.

DevOps Insights

DevOps Insights provides comprehensive insights from popular continuous integration and continuous delivery tools to increase the speed and control of the application delivery. View the pipeline test results for every build and from every deployment and environment. Create policies to ensure that only quality code is delivered to production, and review trends to understand improvements over time.

Event Management

Restore service and resolve operational incidents fast. Empower the DevOps teams to correlate different sources of events into actionable incidents, synchronize teams, and automate incident resolution. The service sets you on course to achieve efficient and reliable operational health, service quality, and continuous improvement.

Globalization Pipeline

IBM Globalization Pipeline is a DevOps integrated application translation management service that users can use to translate and release rapidly cloud and mobile applications to their global customers. IBM Globalization Pipeline capabilities are accessible through its dashboard, a RESTful API, or by integrating it seamlessly into the application's Delivery Pipeline.

Monitoring

Provides insight into how the apps are performing and consuming resources. Quickly identifies trends, and detects and diagnoses problems with immediate time to value and low total cost of ownership (TCO).

Toolchain

A toolchain is a set of integrated tools for development, deployment, and monitoring. This service is explained in more detail in the next section.

There are also third-party DevOps services that are available, such as IBM Cloud Activity Tracker with LogDNA, IBM Cloud Monitoring with Sysdig, IBM Log Analysis with LogDNA, Mendix Platform Service, and PagerDuty.

4.3. IBM Cloud Continuous Delivery overview

Figure 4-16. IBM Cloud Continuous Delivery overview

IBM Training

Topics

- Introduction to DevOps
- DevOps services on IBM Cloud
- IBM Cloud Continuous Delivery overview
- Web IDE (Edit Code)
- Source Code Management (Git repository) and Issue Tracker
- Automated build and deployment (Delivery Pipeline)

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-17. Topics

IBM Training

What is IBM Cloud Continuous Delivery

- IBM Cloud Continuous Delivery provides the capabilities, practices and industry-leading tools:
  - Create open and integrated toolchains that support your development, deployment, and operations tasks.
  - Use automated pipelines to deliver continuously
  - Use the Web IDE to edit and push your code from anywhere
  - Collaborate with your team by using the Git repository and issue tracker

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-18. 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.

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 open and integrated DevOps 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 develop, build, deploy, test, and manage collaboratively 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 PagerDuty, Sauce Labs, 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, debug, and complete source-control tasks in Git. 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.

To have a healthy toolchain, it needs to meet the needs of your development pipeline. So developers must include tools for the following:

- Planning and collaboration such as project management tools and communication tools like Slack
- Source control tools like Git
- Tracking and escalating tools like issue tracking (GitLab)
- Continuous integration
- Monitoring tools
- Testing and automation tools
- Deployment tools

To check IBM Cloud Continuous Delivery service check the following link:

https://cloud.ibm.com/catalog/services/continuous-delivery

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 (including Git repos and Issue Tracking), see the following resources:

- Develop a Kubernetes app toolchain:

  https://www.ibm.com/cloud/garage/toolchains/develop-kubernetes-app-toolchain

- Develop a Cloud Foundry app toolchain:

  https://www.ibm.com/cloud/garage/toolchains/develop-cloud-foundry-app-toolchain

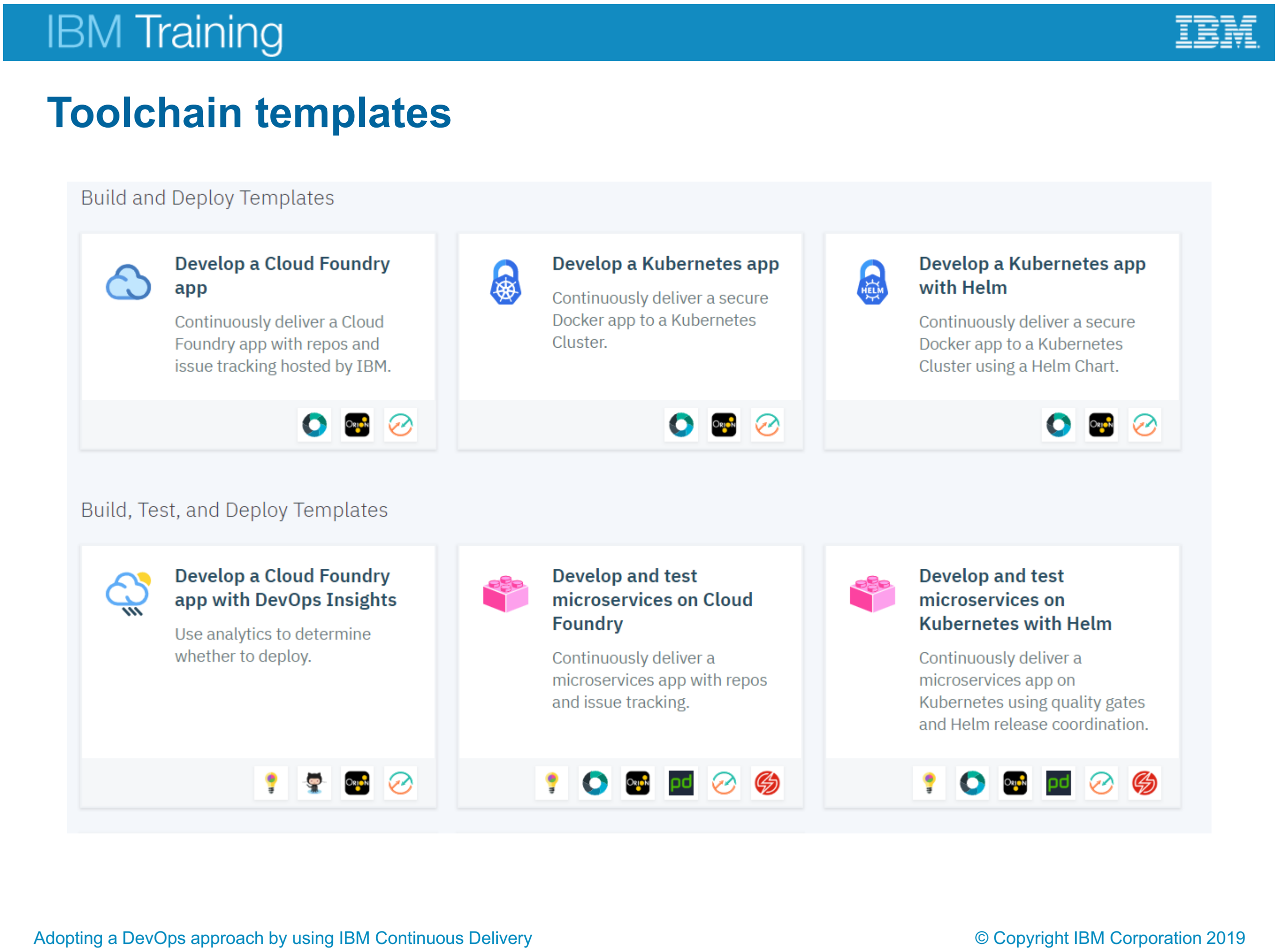


Figure 4-19. Toolchain templates

- A *toolchain* is a set of tool integrations that supports development, deployment, and operations tasks. The collective power of a toolchain is greater than the sum of its individual tool integrations.
- Open toolchains are available in the Public and Dedicated environments on IBM Cloud. You can create a toolchain in two ways:
  - Use a template to create a toolchain
  - Create a toolchain from an app.
- IBM Cloud provides the most commonly used DevOps patterns as toolchain templates which can be easily extended. You can select any template of the available ones and add more tool integrations to it as needed. You also can create a custom toolchain by using the “Build your own toolchain” template.

Reference

https://cloud.ibm.com/devops/getting-started

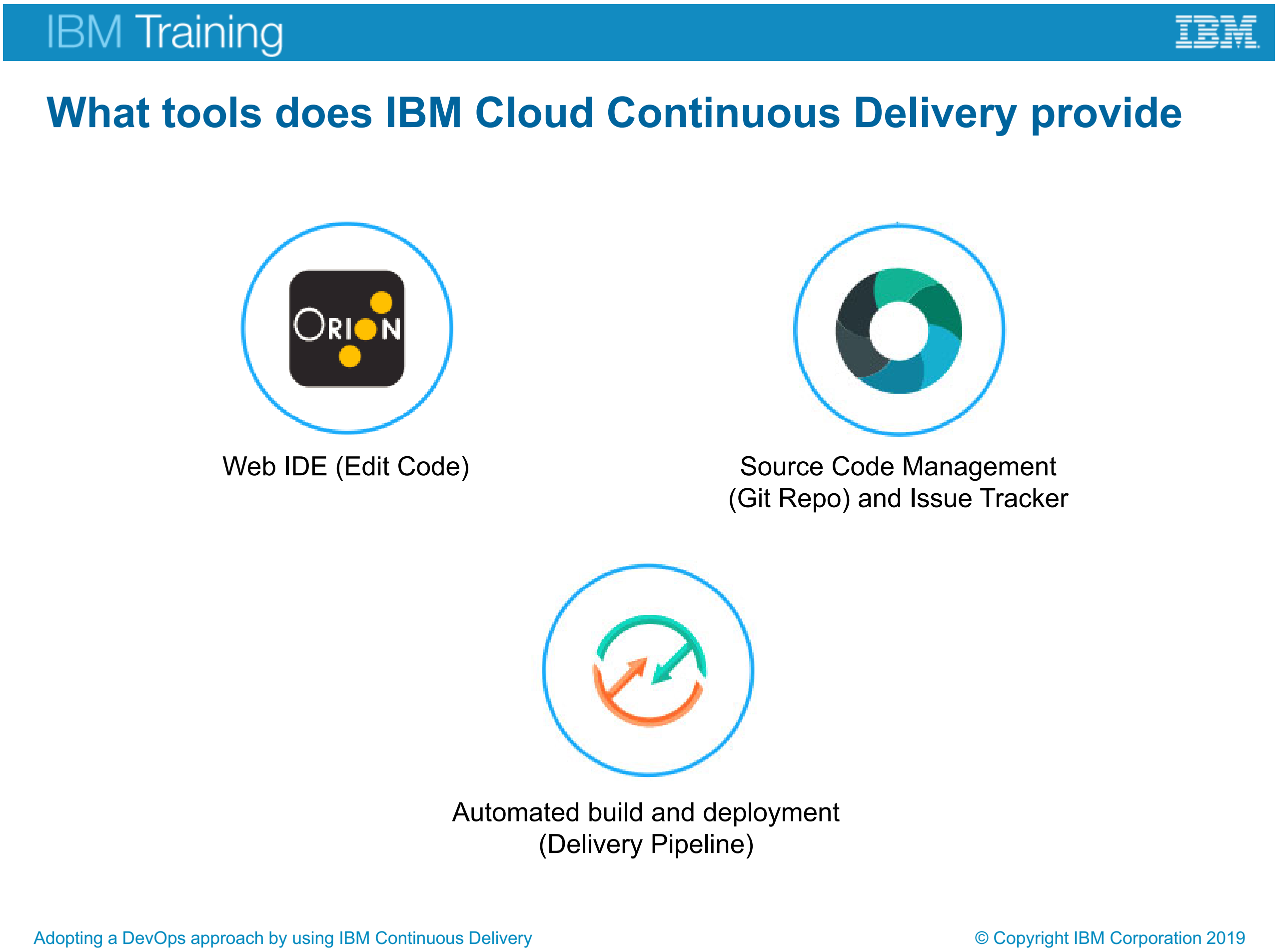


Figure 4-20. What tools does IBM Cloud Continuous Delivery provide

IBM Cloud Continuous Delivery provides the following tools:

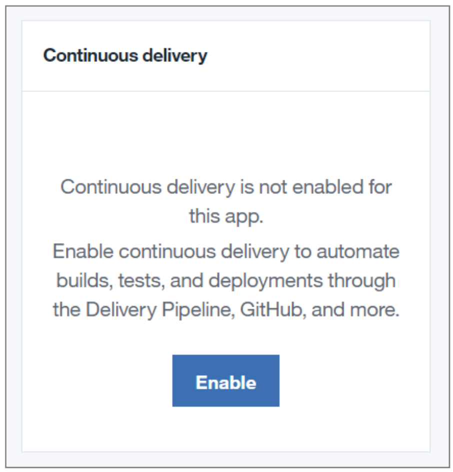
- Web IDE: You can develop your code (for example, Node.js, Java, or other languages) in the Web IDE. No software other than your web browser is necessary. The Web IDE saves your current work in a cloud-based file directory, which is known as the *local repository*. IBM Cloud provides Eclipse Orion as the Web IDE. You can also use a desktop IDE, such as Eclipse with DevOps services.
- Source control management (Git repository) 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 an SCM system. The Git repository, which is known as the *remote repository*, is hosted on the IBM Cloud infrastructure. Issue Tracker is part of the functions 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.

IBM Training

Adding Continuous Delivery

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

1. In the IBM Cloud resource list, select your application.
2. Click **Overview** in the left navigation bar.
3. In the Continuous Delivery pane, click **Enable**.



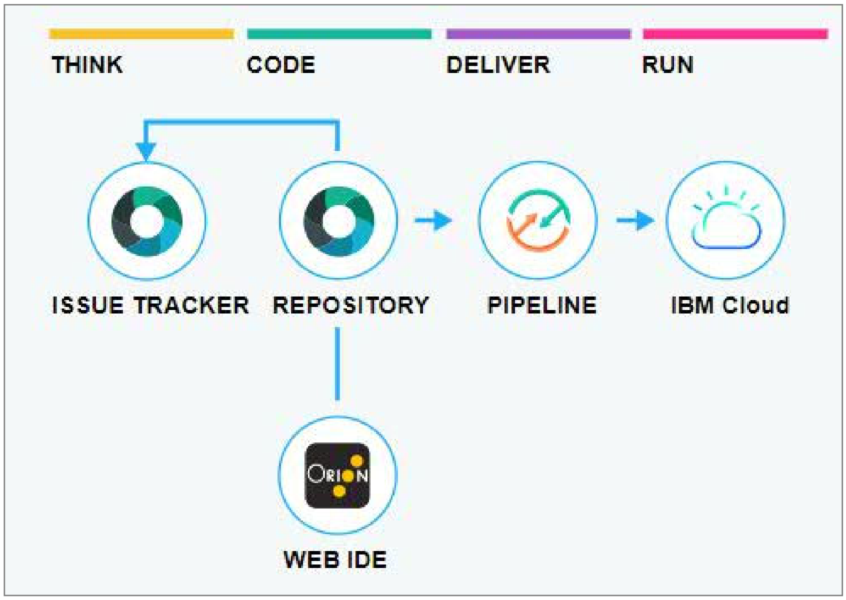
Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-21. Adding Continuous Delivery

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

IBM Training

Creating Continuous Delivery Toolchains



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-22. Creating Continuous Delivery Toolchains

A *toolchain* is a set of tool integrations that supports 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 ideas, tasks, or bugs 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 a source code management (SCM) 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. By using it, you can 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.

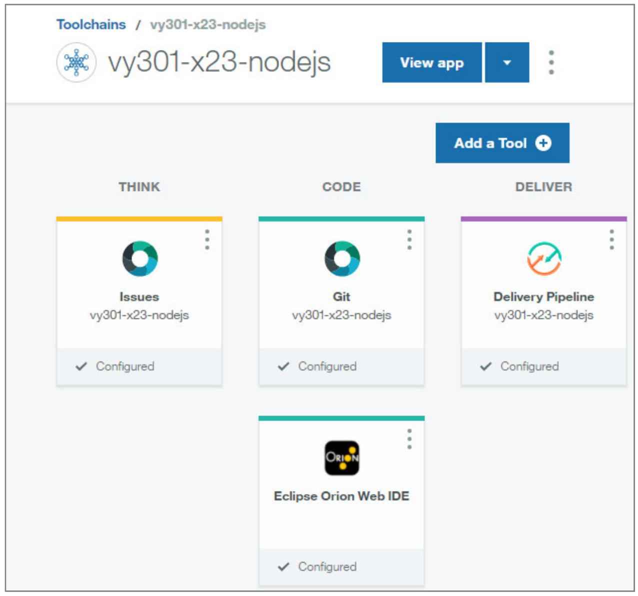
For more information about the phases of developing a Cloud Foundry app toolchain, see the IBM Cloud Garage method:

https://www.ibm.com/cloud/garage/toolchains/develop-cloud-foundry-app-toolchain

IBM Training

Adding Tools to the toolchain

You can add tools into your continuous delivery toolchain.



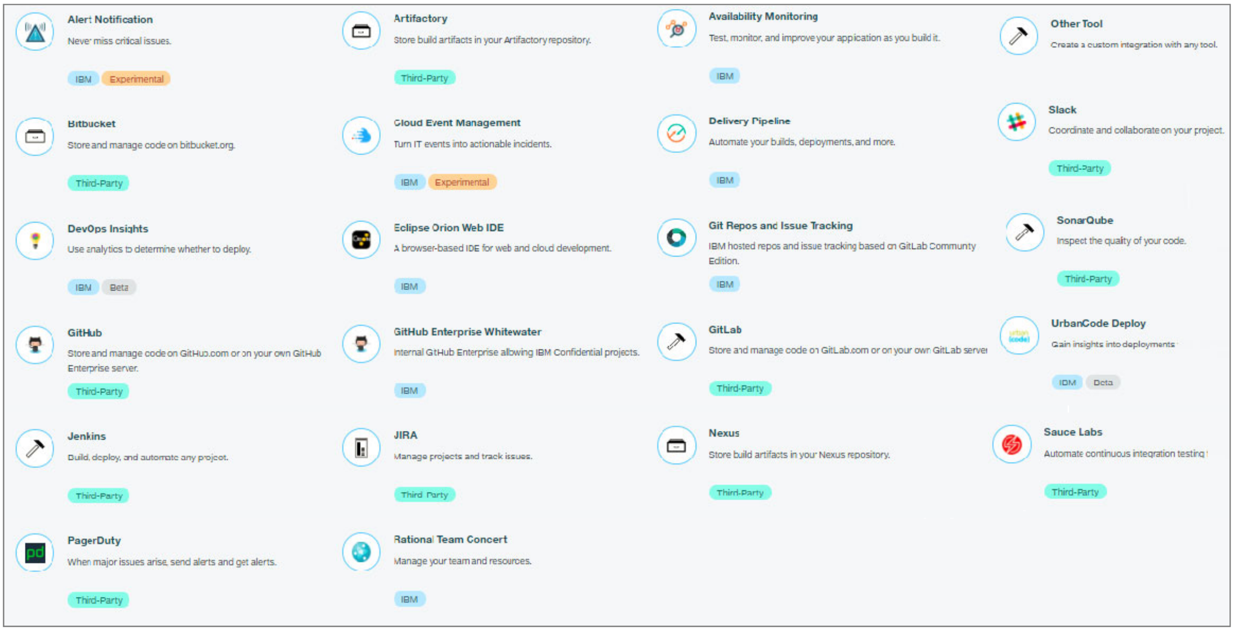
Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-23. Adding Tools to the toolchain

To add tools, click **Add a Tool** from within a toolchain. With this feature, you have the flexibility to integrate and add new tools or services to your toolchain, such as Alert Notifications, Availability Monitoring, DevOps insights, or many of the other DevOps services that are available on IBM Cloud (some of these services are explained in the next slide).

IBM Training

Adding Tool Integration to a toolchain



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-24. Adding Tool Integration to a toolchain

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.

4.4. Web IDE (Edit Code)

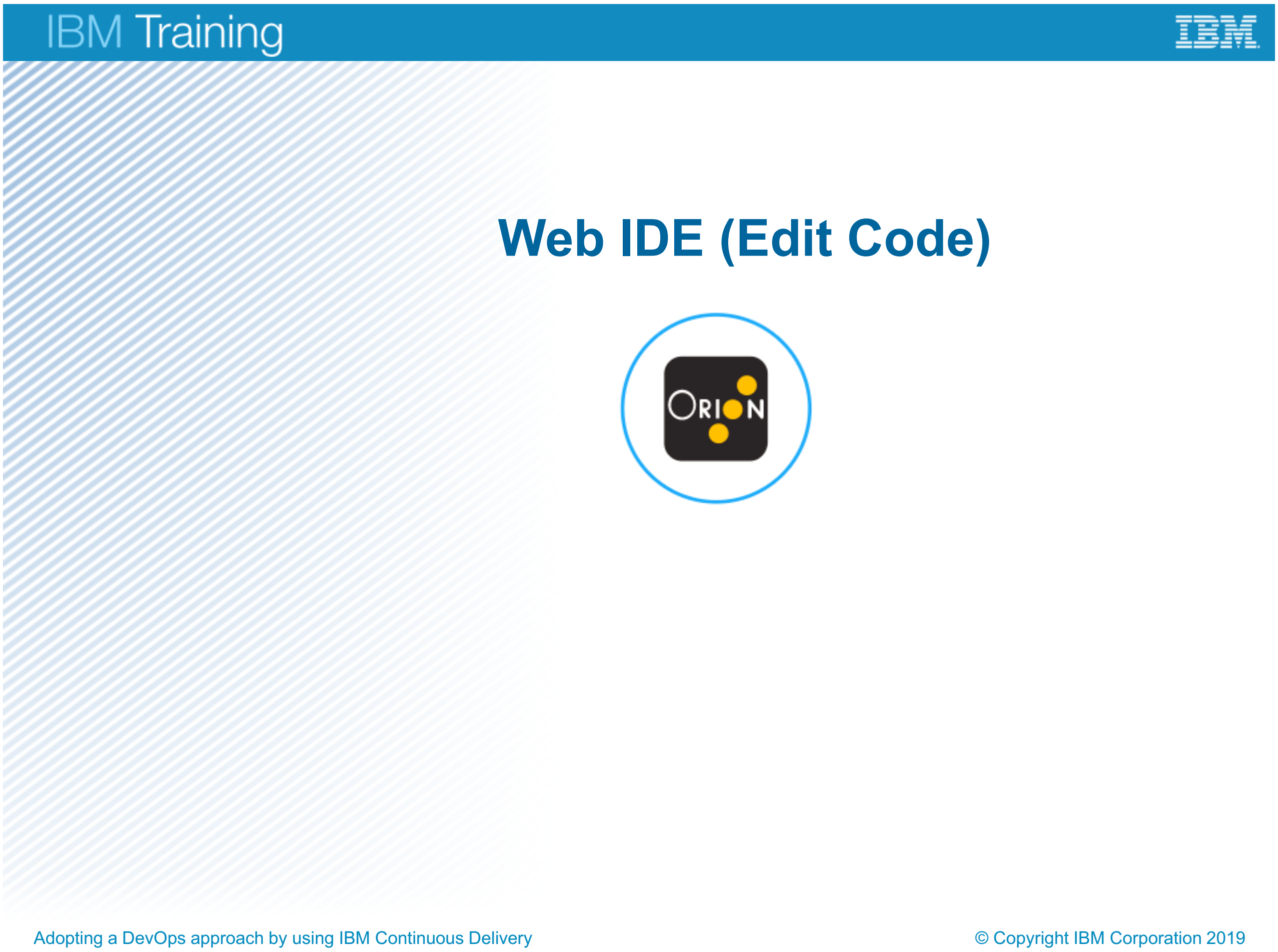


Figure 4-25. Web IDE (Edit Code)

IBM Training

Topics

- Introduction to DevOps
- DevOps services on IBM Cloud
- IBM Cloud Continuous Delivery overview
- Web IDE (Edit Code)
- Source Code Management (Git repository) and Issue Tracker
- Automated build and deployment (Delivery Pipeline)

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-26. Topics

IBM Training

Web IDE: Edit code features

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

WEB IDE

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-27. Web IDE: Edit code features

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

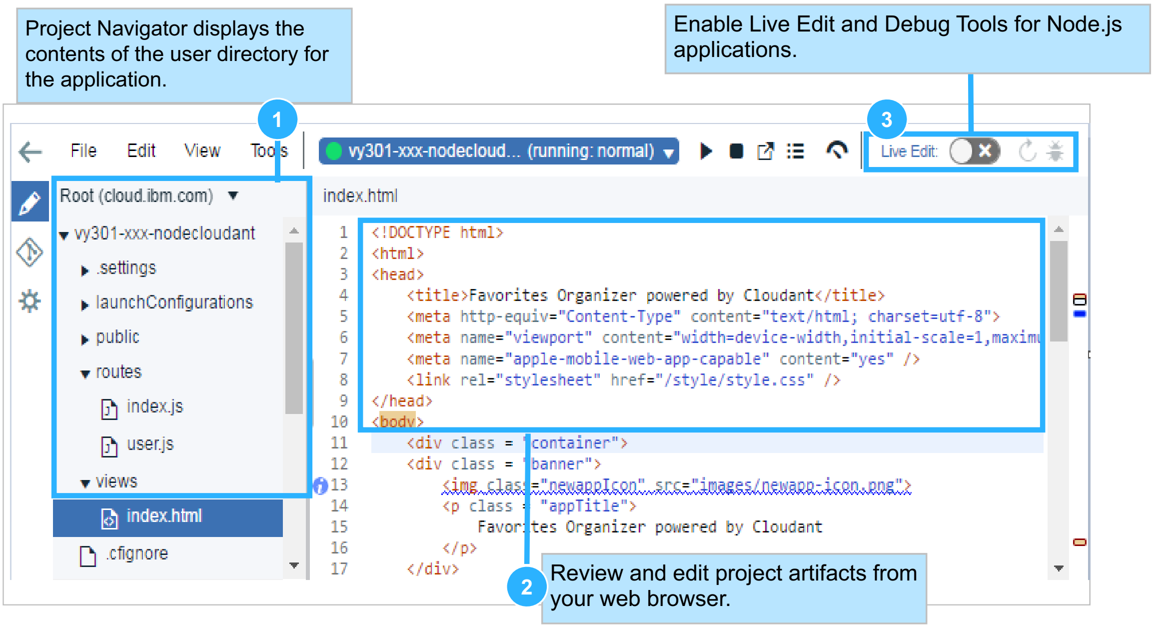
- Uses Eclipse Orion Web IDE.
- No installations are required. Use a browser to code.
- The Edit Code feature provides a workspace to develop source code and configuration files.
- Full-featured environment for writing your application code by using your web browser.
- Rich code completion capabilities 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, they are available for Node.js applications only):

- Live Edit: You can change your application from the Web IDE without redeploying 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 run time, and more by using the Node Inspector debugger.

IBM Training

Web-based integrated development environment



Adopting a DevOps approach by using IBM Continuous Delivery

© Copyright IBM Corporation 2019

Figure 4-28. Web-based integrated development environment

The Edit Code perspective for Eclipse Orion Web IDE is shown in this slide.

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, which is described in the next section.

IBM Training

Editing source code

- 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.

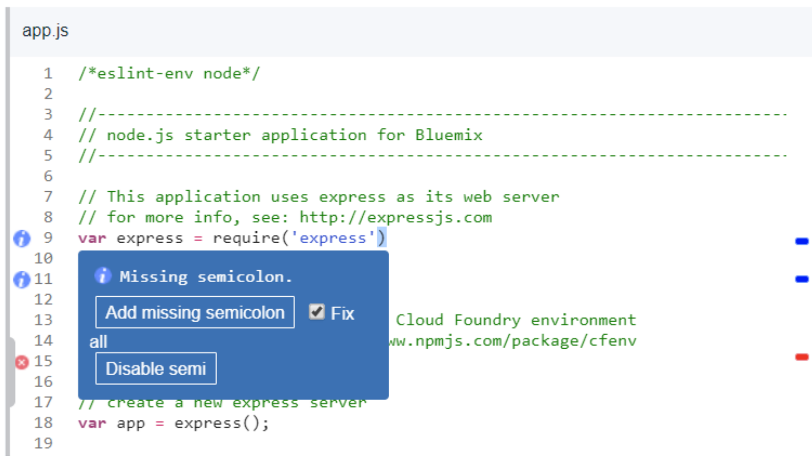


Figure 4-29. Editing source code

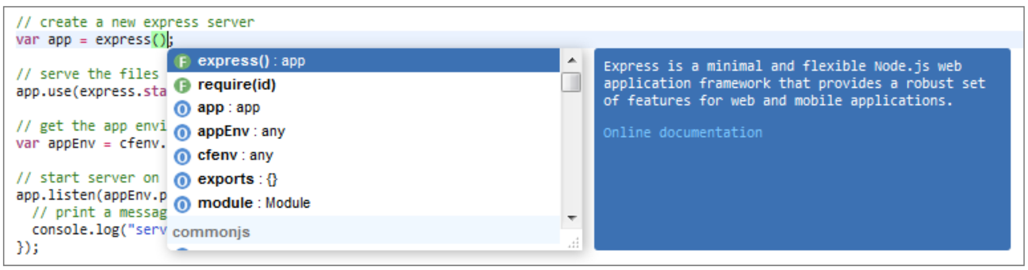
When you open a Node.js script file, the editor provides real-time validation and syntax checking of the source code. It uses tools, such as JSHint, which is a JavaScript code quality tool that helps detect errors and potential problems in the code. A preview window flags warnings and errors.

IBM Training

Editor features: Code completion

The code completion feature suggests functions and parameters in the editor:

- This feature dynamically displays all available libraries at the cursor point.
- Press Ctrl + Spacebar to activate the code completion menu.



Adopting a DevOps approach by using IBM Continuous Delivery

© Copyright IBM Corporation 2019

Figure 4-30. Editor features: Code completion

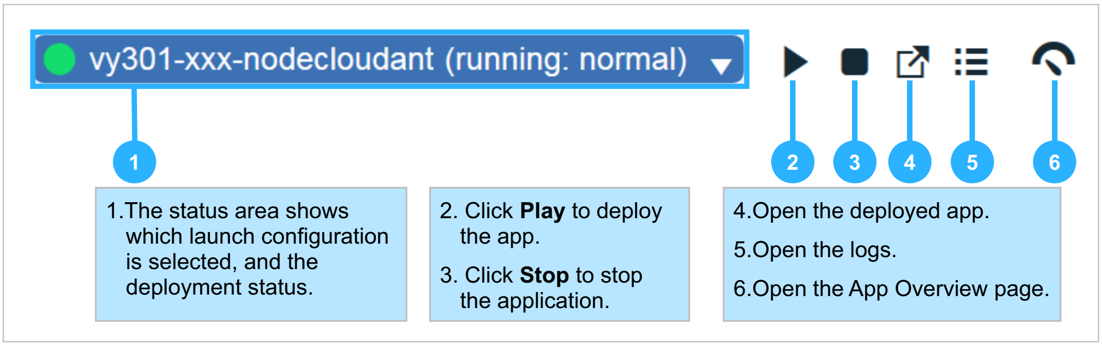
To use the code completion shortcut, place your cursor within the editor and press Ctrl + Spacebar. This feature displays the libraries that are available at the cursor point, including third-party modules that you imported in the script. For example, as shown in the slide, the code completion feature 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.

IBM Training

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**.

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-31. Editor features: Run bar

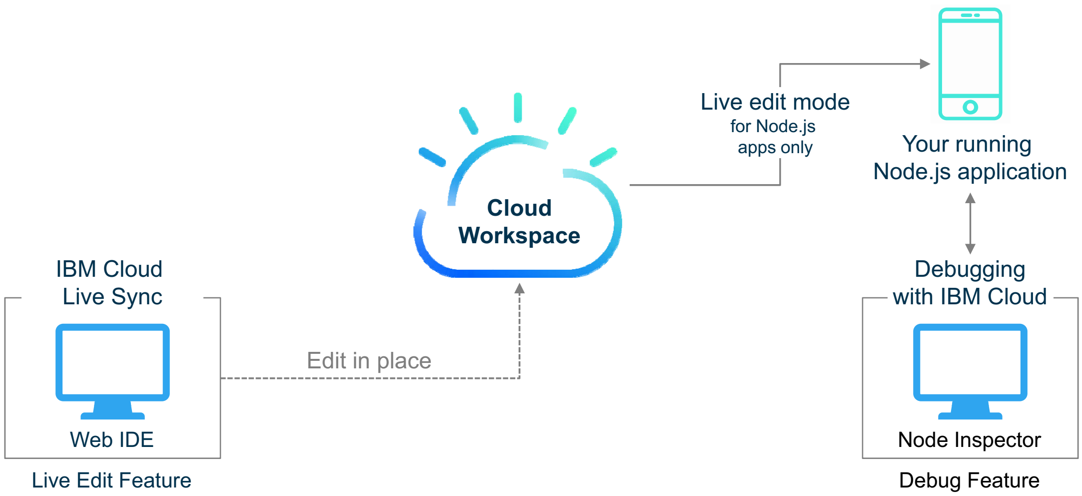
You can quickly build and deploy your application to a test environment on IBM Cloud without committing your code to the Git repository. The following features are highlighted in the slide:

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

IBM Training

IBM Cloud Live Sync features

If you are building a Node.js application, you can change your application on IBM Cloud as you would on the desktop without the need to redeploy it.



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-32. IBM Cloud Live Sync features

If you are building a Node.js application, you can use IBM Cloud Live Sync to update quickly 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 it 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 run time, and more.

  Note: To use debug features, you must use the Chrome browser and Node.js V4 because IBM Cloud Live Sync Debug uses Node Inspector to provide debug features, which are not available in the later versions of Node.js.

4.5. Source Code Management (Git repository) and Issue Tracker

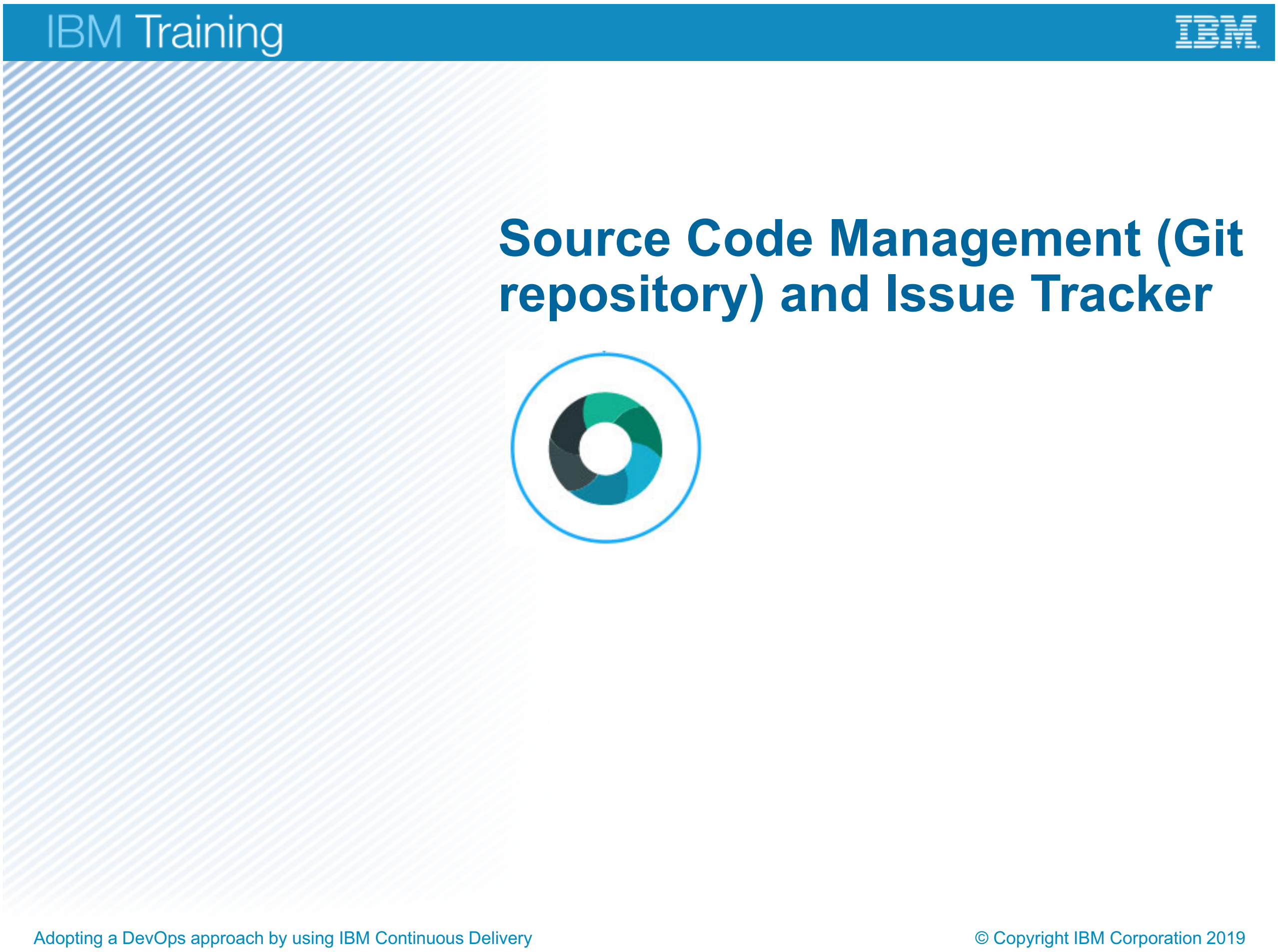


Figure 4-33. Source Code Management (Git repository) and Issue Tracker

IBM Training

Topics

- Introduction to DevOps
- DevOps services on IBM Cloud
- IBM Cloud Continuous Delivery overview
- Web IDE (Edit Code)
- Source Code Management (Git repository) and Issue Tracker
- Automated build and deployment (Delivery Pipeline)

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-34. Topics

IBM Training

Git repository and Issue Tracking

- 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.

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-35. Git repository and Issue Tracking

IBM Training

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 at this website:

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

Figure 4-36. Source control with a Git repository

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. GitLab is the web based Git repository. Github is another example of the web based git repositories.

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:

https://www.git-scm.com/

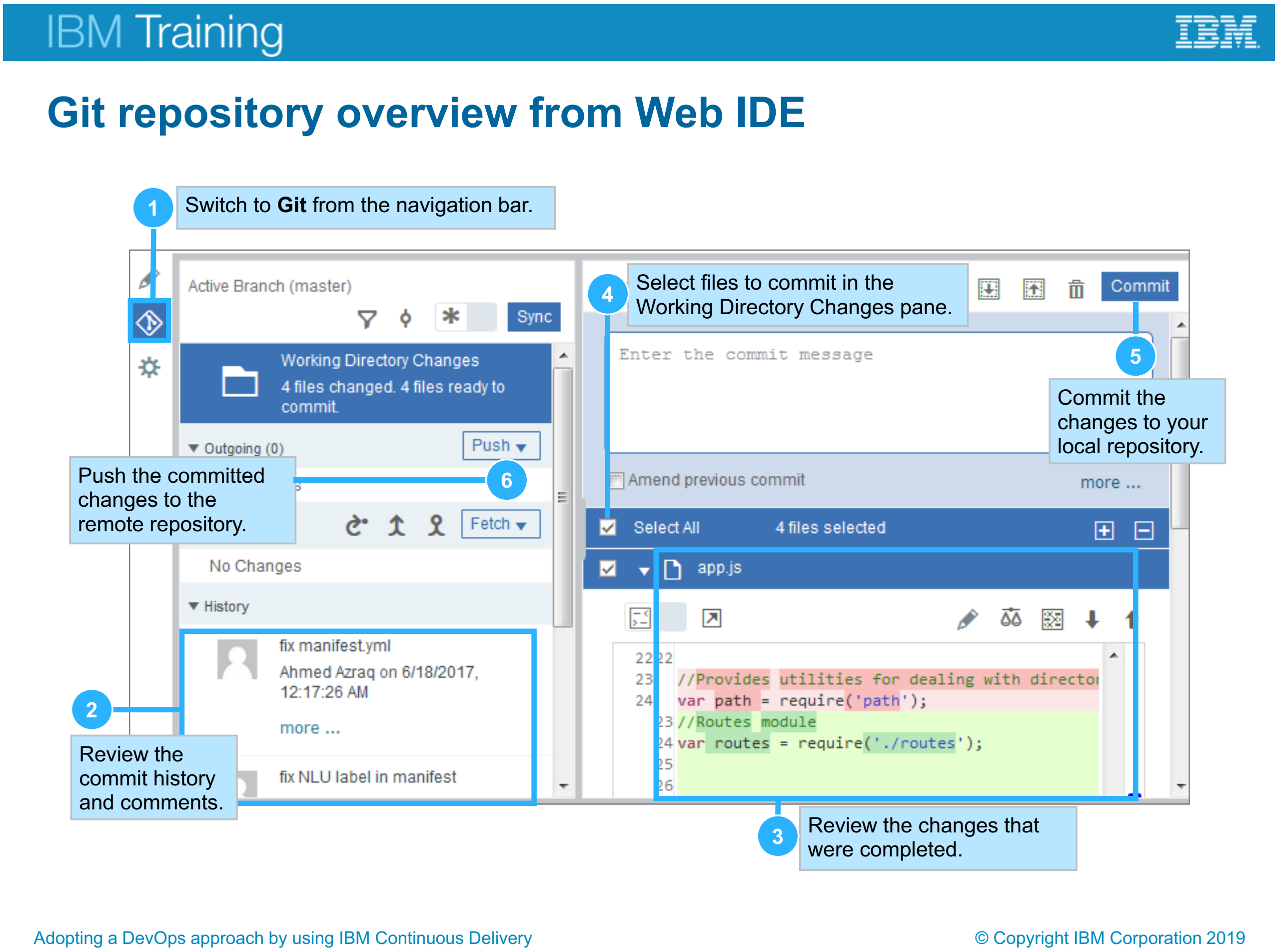


Figure 4-37. Git repository overview from Web IDE

The numbers that are shown in the slide 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 and date for each commit. You can perform actions for each commit, such as view the files that are committed and revert changes that were introduced by any commit.
3. The Working Directory Changes pane detects any updated files in the user directory. Click a file to review the changes that were made in that file.
4. Select the files that you want to commit and 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, and 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.

IBM Training

Connecting a Git client to your repository on IBM Cloud

- 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://github.com/IBMRedbooks/Cloud-
Application-Developer.git
```

- 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
```

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-38. Connecting a Git client to your repository on IBM Cloud

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

If you do not have a Git client that is installed, download and install the latest version of the Git client from 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 and 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-line interface (CLI). After you have a copy of the source code, use a text editor or IDE 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.

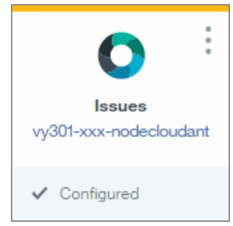
Then, run the “`git commit`” command and enter a message for the history log to commit all the added files to your local repository.

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

IBM Training

Issue Tracking tool

IBM Cloud Continuous Delivery toolchain includes a tool to track issues.



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-39. Issue Tracking tool

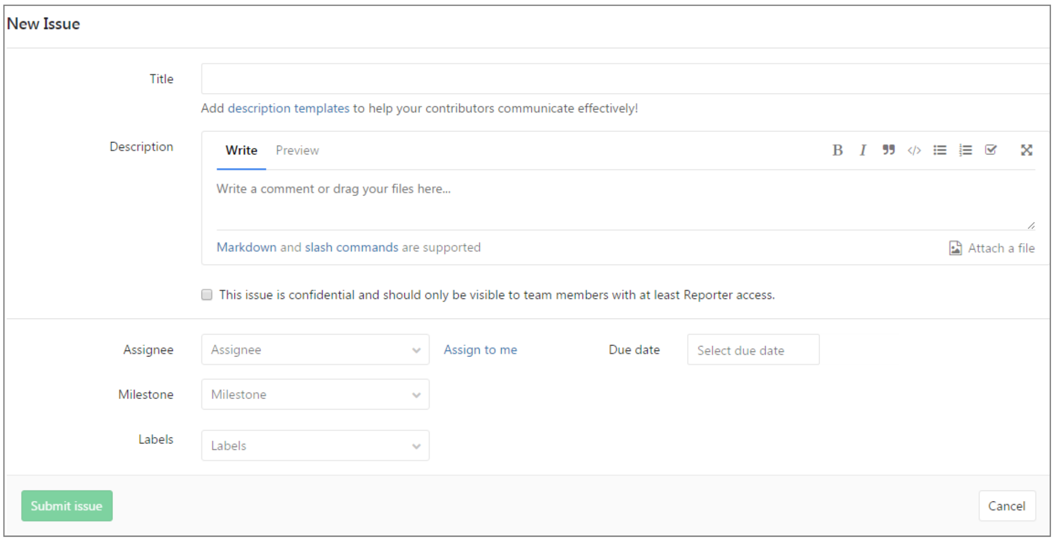
The Issue Tracking tool (*Issue Tracker* in the slide) is used to track items (issues) that require resolution or improvements in a project.

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

IBM Training

Creating an issue

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



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-40. Creating an issue

Complete the form that is shown in the slide. Then, click **Submit issue** to create the bug or task.

4.6. Automated build and deployment (Delivery Pipeline)

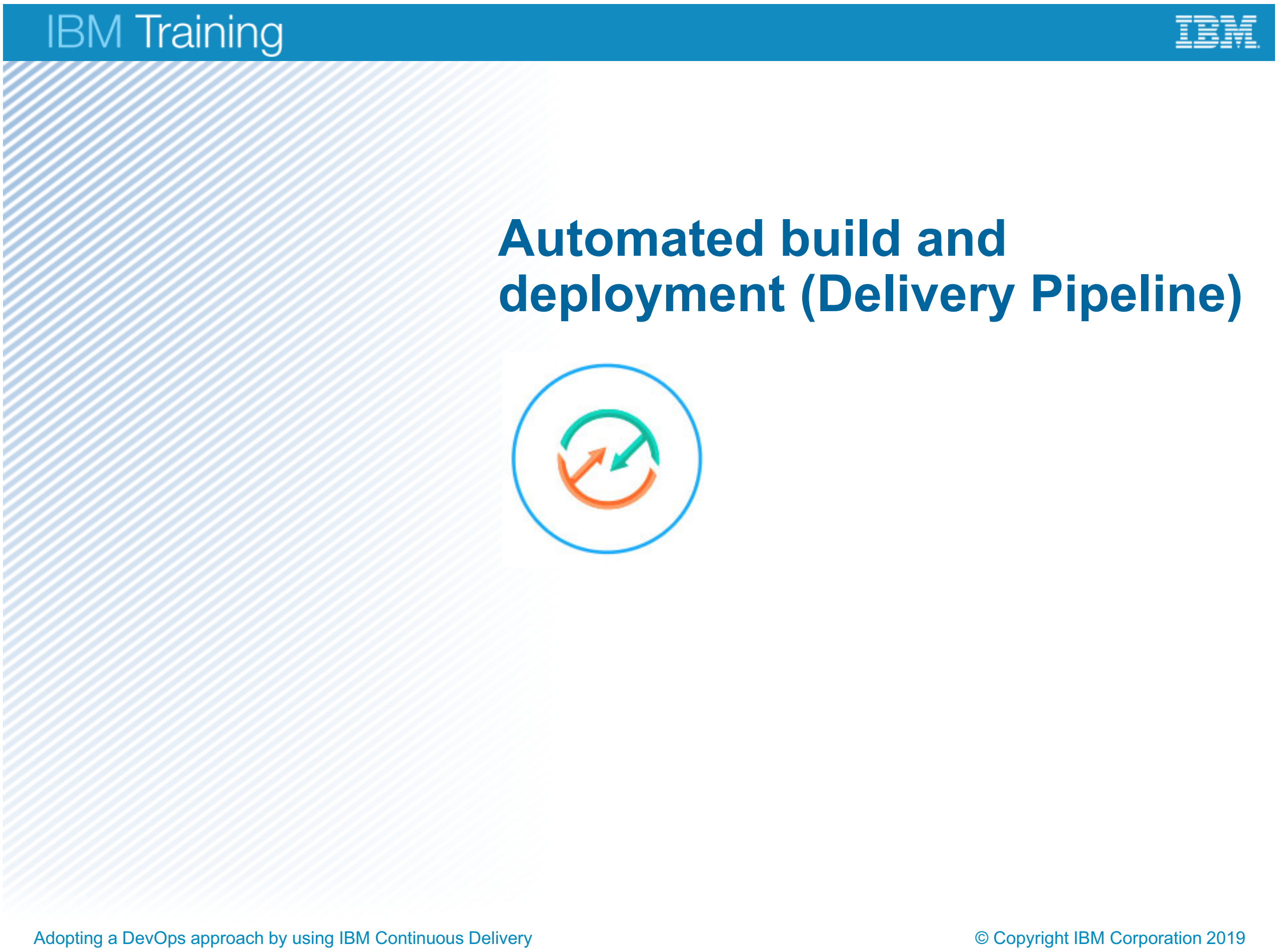


Figure 4-41. Automated build and deployment (Delivery Pipeline)

IBM Training

Topics

- Introduction to DevOps
- DevOps services on IBM Cloud
- IBM Cloud Continuous Delivery overview
- Web IDE (Edit Code)
- Source Code Management (Git repository) and Issue Tracker
- Automated build and deployment (Delivery Pipeline)

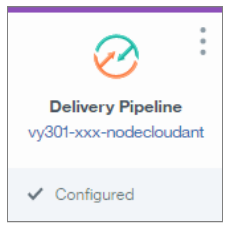
Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-42. Topics

IBM Training

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 this function is configured to be triggered automatically.
  - When you click **Play** from the run bar.
- Jobs (Build, Deploy, and Test) are grouped into stages.



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-43. Delivery pipeline

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 runs on only committed code or builds, depending on the stage.

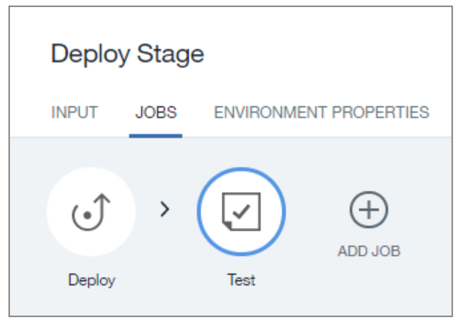
You can access the Continuous Delivery toolchain by clicking **View toolchain** from the Application Details page. To browse the Delivery Pipeline (or any other tool), click its tool card in the toolchain overview page.

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

IBM Training

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.



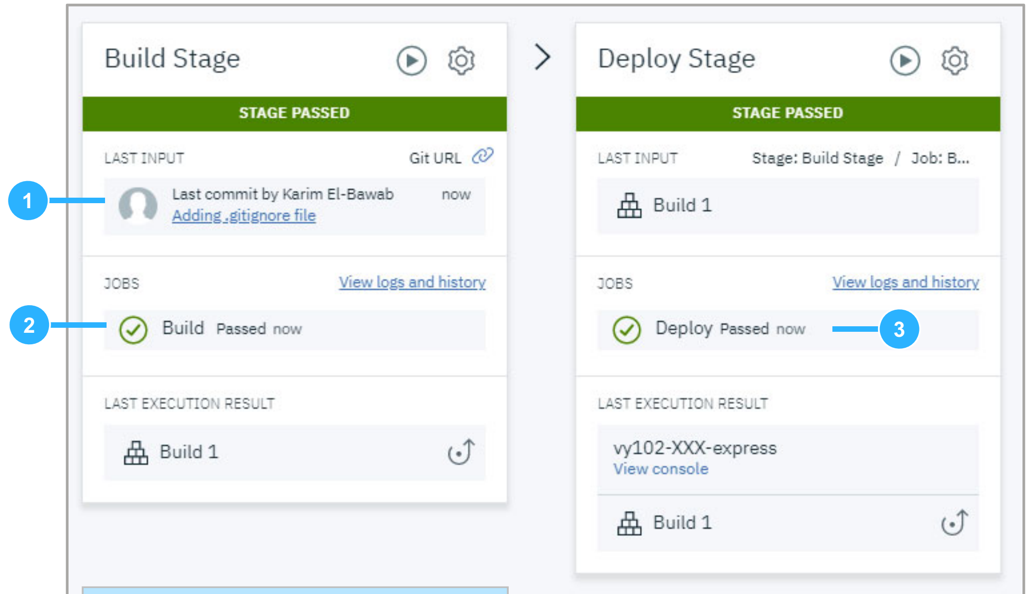
Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-44. Customizing Delivery Pipeline

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.

IBM Training

Example: Default delivery pipeline



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

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-45. Example: Default delivery pipeline

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 run time.

You can customize the settings for the server run time through the manifest.yml file.

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

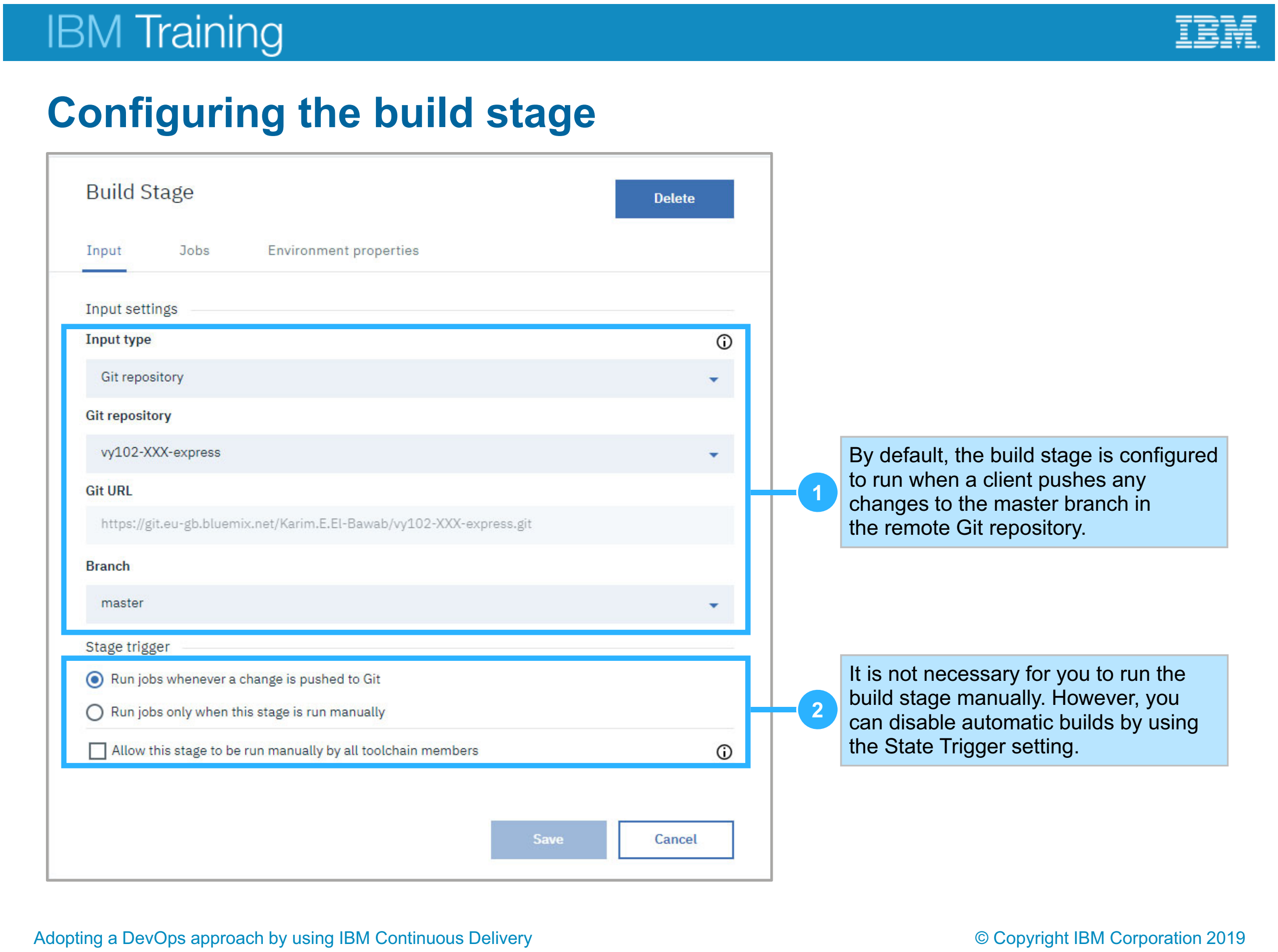


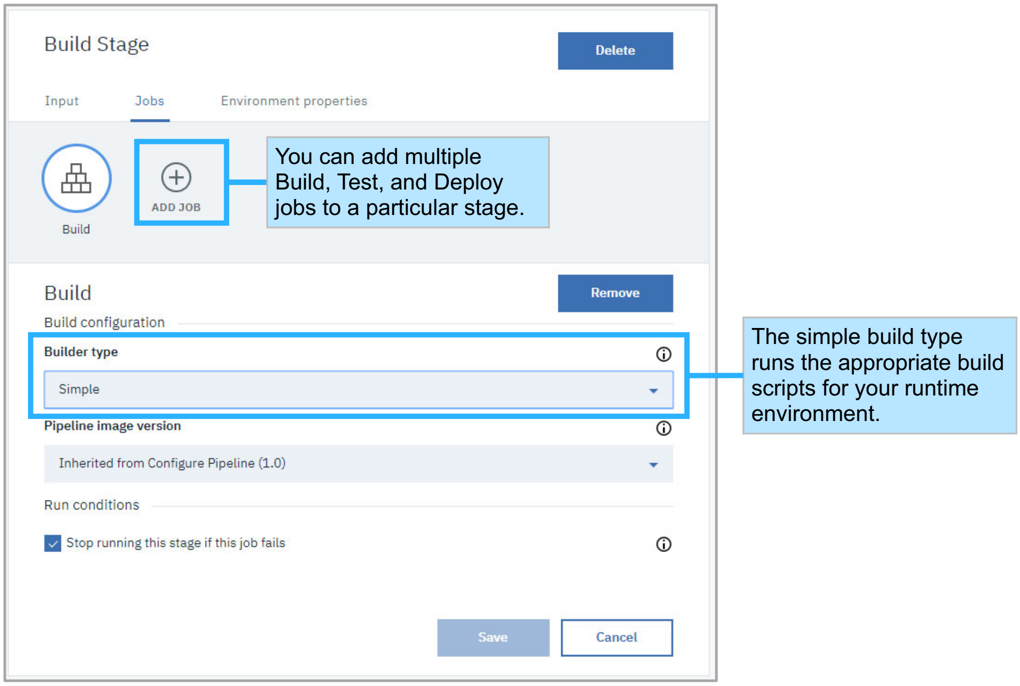
Figure 4-46. Configuring the build stage

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**.

IBM Training

Configuring build jobs



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-47. Configuring build jobs

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 writing, Builder Type supports the following types:

- Simple
- Ant
- Container Registry
- Custom Docker Image
- Gradle
- Gradle (Artifactory, Nexus, or SonarQube)
- Grunt

- Maven
- Maven (Artifactory, Nexus, or SonarQube)
- npm
- npm (Artifactory or Nexus)
- Shell Script

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

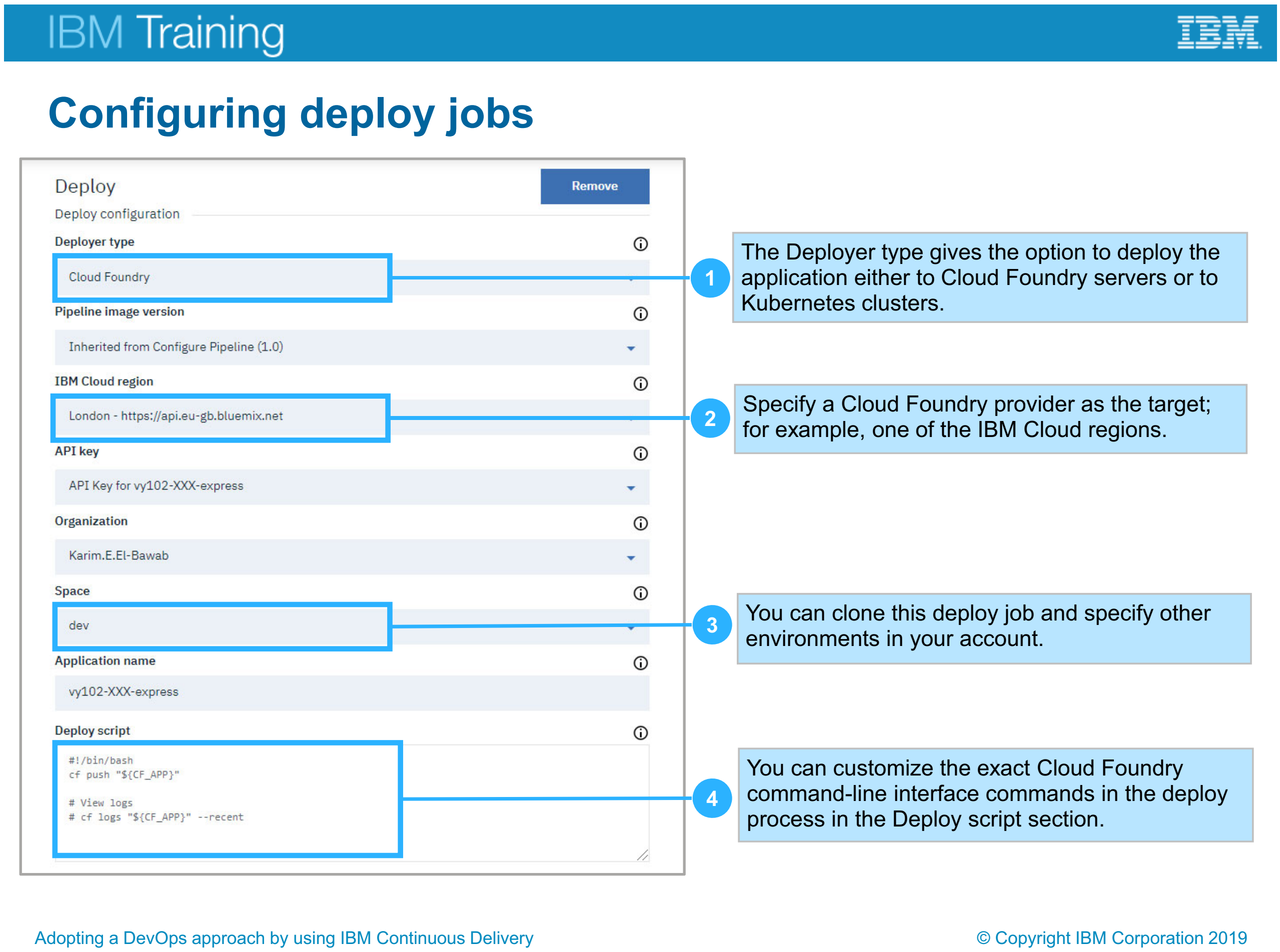


Figure 4-48. Configuring deploy jobs

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:

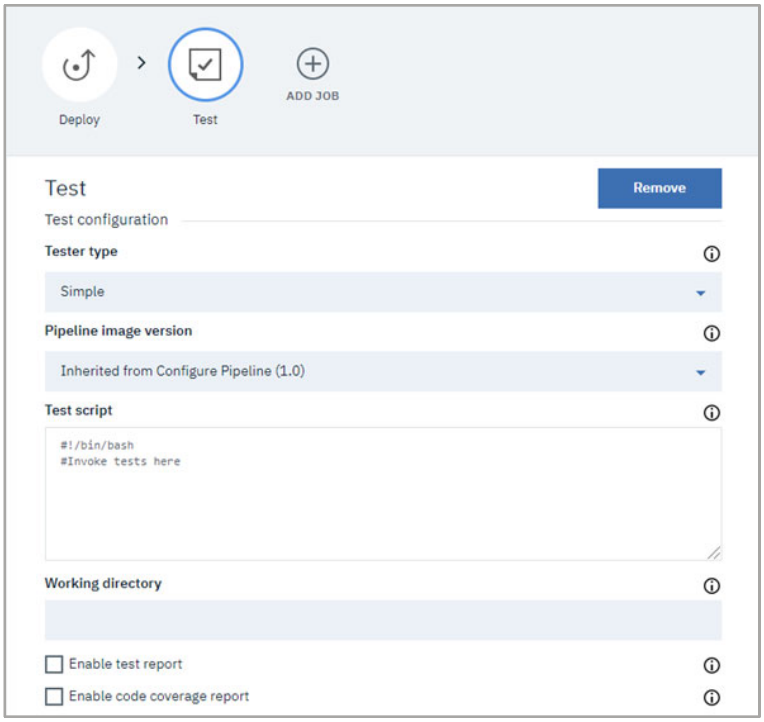
- Cloud Foundry: Deploys applications to Cloud Foundry servers. This type is the default deployer type for the deploy job.
- Kubernetes: Deploys applications to Kubernetes clusters, such as those found within the IBM Cloud Kubernetes Service.
- Custom Docker Image: Deploy by using your custom Docker image with fine-grained control over the versions of node, Java, or other tools.

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 the slide refers to the application name).

IBM Training

Configuring test jobs



Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-49. Configuring test jobs

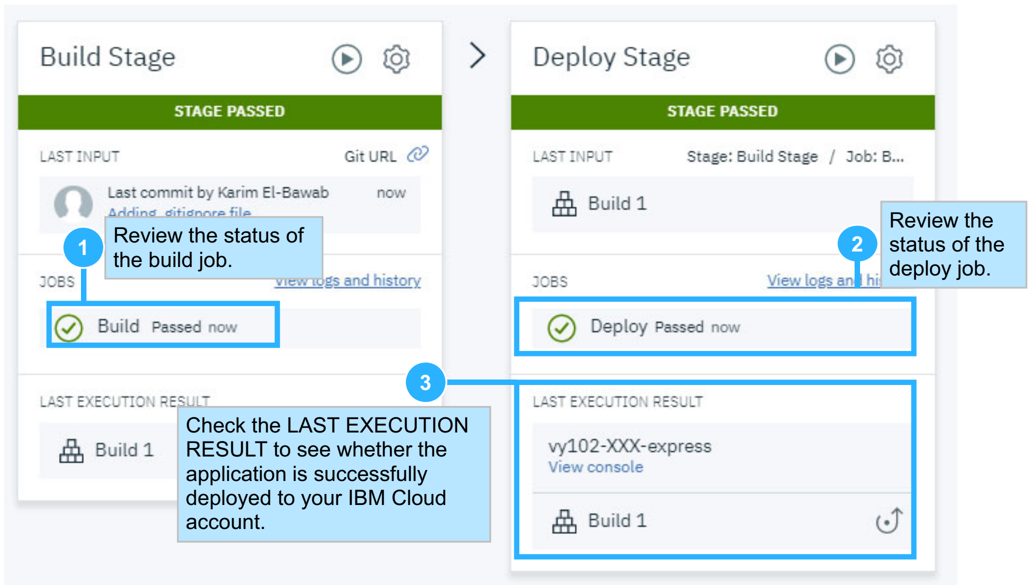
After the deploy job is completed, you can add and configure a test job to test your deployed application.

The following Tester types are available:

- Simple
- Custom Docker Image
- DevOps Insights Gate (DEPRECATED)
- Sauce Labs
- Simplified Cloud Foundry org and space shell
- Vulnerability Advisor

IBM Training

Example: A successful build and deploy result



Adopting a DevOps approach by using IBM Continuous Delivery

© Copyright IBM Corporation 2019

Figure 4-50. Example: A successful build and deploy result

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 completes 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 complete successfully, as shown in the JOBS pane of the Build Stage and Deploy Stage.

The LAST EXECUTION RESULT pane shows that the application 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 the 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.

IBM Training

Unit summary

- Describe DevOps.
- Describe the capabilities of IBM Cloud Continuous Delivery.
- Identify the web-based integrated development environment (Web IDE) features in IBM Cloud Continuous Delivery.
- Describe how to use source code management (such as Git) and Issue tracking.
- Explain how to build and deploy applications using DevOps tools on IBM Cloud.

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-51. Unit summary

IBM Training

Review questions

1. _____ is a DevOps practice that allows applications to have automated deployments.
   A. Continuous Improvement
   B. Continuous Integration
   C. Continuous delivery
   D. Release Planning

2. _____ is one of the DevOps services available on IBM Cloud.
   A. Load Balancers
   B. Cloudant
   C. Natural Language Understanding
   D. Auto-Scaling

3. **True or False**. To create Continuous Delivery Toolchain for your IBM Cloud application, select **Enable** in the Continuous Delivery pane.

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-52. Review questions

IBM Training

Review questions (cont.)

4. **True or False**. A toolchain template can be integrated with other tools and services only during creation.

5. **True or False**. In the delivery pipeline, only one job can be configured per each stage.

6. ____ is one of the job types that is available during the Delivery Pipeline stages.
   A. Test
   B. Monitor
   C. Feedback
   D. Delivery

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-53. Review questions (cont.)

IBM Training

Review answers

1. **C**.
2. **D**.
3. **True**.
4. **False**. You can add tools and services either during creation or after the creation of the toolchain.
5. **False**. You can configure one or more jobs per stage in the delivery pipeline.
6. **A**.

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-54. Review answers

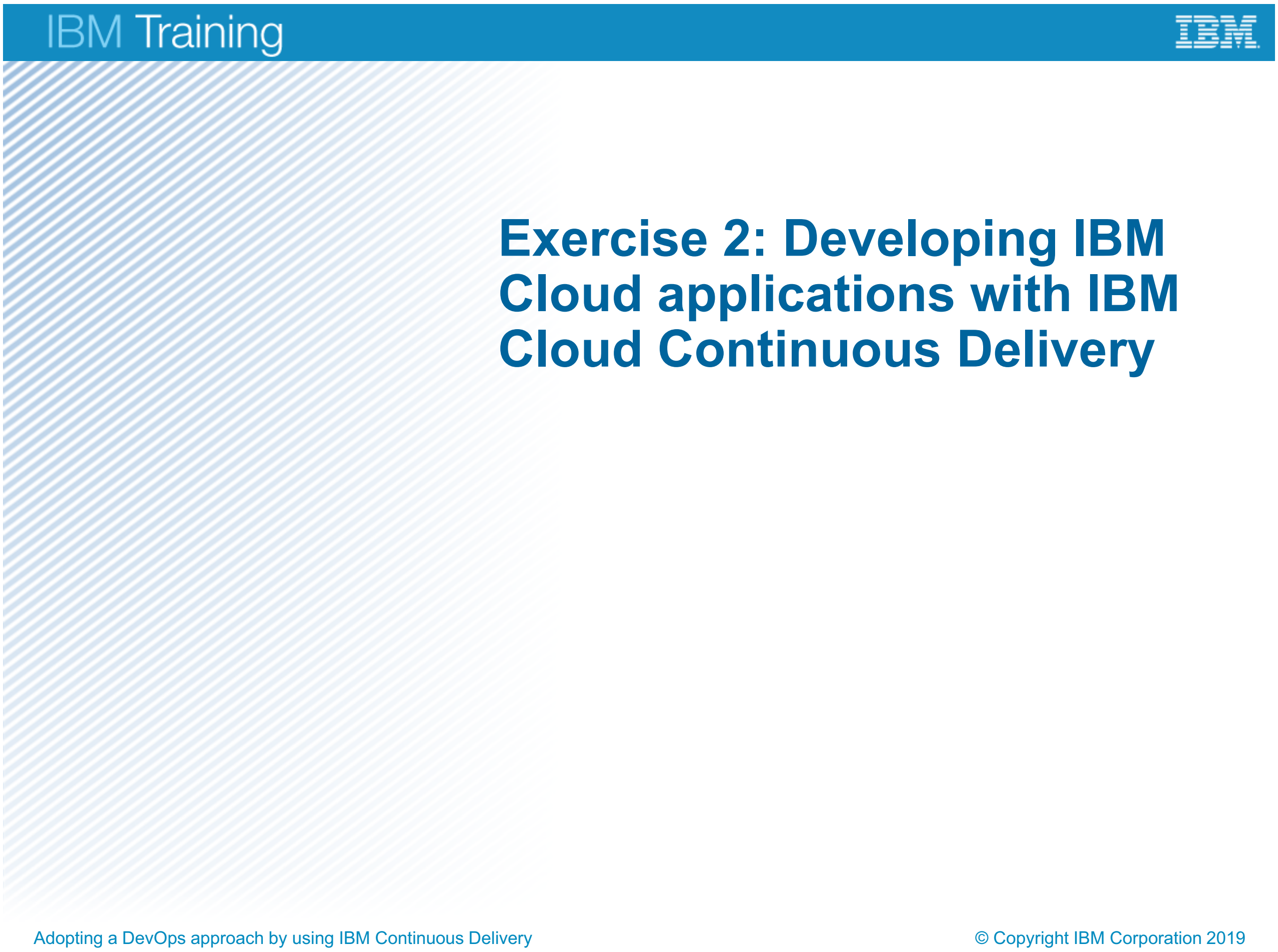


Figure 4-55. Exercise 2: Developing IBM Cloud applications with IBM Cloud Continuous Delivery

IBM Training

Exercise objectives

- In this exercise, you work with the IBM Cloud Continuous Delivery services to explore, develop, build, and deploy IBM Cloud applications.
- After completing this exercise, you should be able to perform the following tasks:
  - Enable your application to use IBM Cloud Continuous Delivery.
  - Create a Git repository to manage your source code.
  - View and edit code in the Eclipse Orion Web integrated development environment (IDE).
  - Build and deploy code to IBM Cloud.
  - Test the application in IBM Cloud.

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-56. Exercise objectives

IBM Training

References

- DevOps For Dummies®, 3rd IBM Limited Edition:
  https://www.ibm.com/downloads/cas/P9NYOK3B
- Open-source Git project:
  www.git-scm.com
- Continuous Delivery Docs:
  https://cloud.ibm.com/docs/services/ContinuousDelivery?topic=ContinuousDelivery-cd_getting_started#cd_getting_started
- IBM Cloud Garage Method:
  https://www.ibm.com/cloud/garage/
- DevOps toolchains:
  https://www.ibm.com/cloud/garage/toolchains/

Adopting a DevOps approach by using IBM Continuous Delivery

Figure 4-57. References

Unit 5. REST architecture and Watson APIs

Estimated time

01:30

Overview

This unit introduces Representational State Transfer (REST) and Resources Representation and JavaScript Object Notation (JSON). It describes how to apply REST architecture concepts to server-side applications. This unit introduces Watson services and provides examples that show how to call Watson services by using REST APIs.

IBM Training

Unit objectives

- Describe the architecture of Representational State Transfer (REST).
- List best practices for using REST in your applications.
- Describe the representation format of data in REST.
- Explain the advantages of the JavaScript Object Notation (JSON) data format.
- List the IBM Watson services on IBM Cloud.
- Provide examples of Watson services REST APIs.

REST architecture and Watson APIs

Figure 5-1. Unit objectives

5.1. REST concepts, architecture, and characteristics

IBM Training

REST concepts, architecture, and characteristics

REST architecture and Watson APIs

Figure 5-2. REST concepts, architecture, and characteristics

IBM Training

Topics

- REST concepts, architecture, and characteristics
- Applying REST to server-side applications
- Resources representation and JSON
- Example: Using REST APIs with IBM Watson

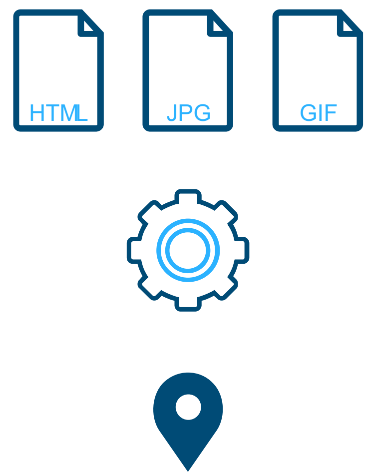
REST architecture and Watson APIs

Figure 5-3. Topics

IBM Training

What is REST

- REST is an architecture style for accessing and manipulating resources on the web.
- HTML documents, images, and script files are considered resources.
- REST can retrieve, update, or delete a resource.
- REST uses a Uniform Resource Identifier (URI) to describe the network location of the resource.



REST architecture and Watson APIs

Figure 5-4. What is REST

REST is a term that was first used by Roy Fielding (one of the creators of the HTTP protocol) in his doctoral thesis, which was published in the year 2000. This style of software architecture for distributed systems (such as World Wide Web) is not applicable only for the development of web services. For example REST principles are intended to apply to other protocols such as FTP.

REST is an architecture style for accessing and manipulating resources on the web, such as HTML documents, images, and script files.

To retrieve or update a resource, perform an action by using 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.

Reference:

https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

IBM Training

REST characteristics

- REST is an architecture style for designing distributed systems. 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.

REST architecture and Watson APIs

Figure 5-5. REST characteristics

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 true. Although the most common scenarios for using REST are 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.

IBM Training

REST architecture constraints

- Uses a client/server model.
  - The client initiates a request by using HTTP. The server processes the request and returns the responses. Separation of concerns is the key principle, for example separating user interface from data storage layer.
- Stateless:
  - Each request from a client must contain all information to understand the request.
- Resource:
  - Any available information is a resource. Each resource must have a unique identifier.
- Uniform interface:
  - REST can retrieve, create, update, or delete a resource by using the following HTTP methods:
    - `GET`
    - `POST`
    - `PUT`
    - `DELETE`
  - Uses URIs to expose resources:
    - A URI identifies the resource to retrieve or update. The URI describes the network location of the resource and allowed actions.

REST architecture and Watson APIs

Figure 5-6. REST architecture constraints

- Uses a client/server model

  The client initiates a stateless request by using the HTTP protocol for destination servers. Servers process the requests and return the responses based on these requests. These requests and responses are built around the transfer of representations of resources. Separation of concerns is the principle behind the client-server constraints. For instance, by separating the user interface concerns from the data storage concerns, we improve the user interface portability across multiple platforms and improve scalability by decoupling and simplifying the server components.

- Stateless

  In client/server interaction, the communication must be stateless natively, which means that each request from client to server must contain all of the information that is necessary to understand the request, and not take advantage of any stored context on the server. Therefore, the session state is kept on the client.

- Resource

  In the REST architectural style, any available information is a resource. The registration of a person, an image, a document, and the quotation of a currency are examples of resources. Each resource must have a unique identifier. This identifier is used so that the resource can be accessed. On the internet or in an intranet, a web resource is identified by a URI.

  A resource is an object with a type, associated data, relationship to other resources, and a set of methods that operate the resource.

- Uniform Interface.

  The architecture specification defines a uniform interface for the object domain. This uniform representation is expressed in terms of:

  - Uses a URL format that uses standard HTTP methods (GET, POST, and DELETE)
  - Uses standard and custom HTTP headers.
  - Uses default codes for HTTP return (200, 404, 403, and others).
  - Uses standard representations by using a URI.

This basic REST design principle establishes a one-to-one mapping between create, read, update, and delete operations and HTTP methods:

- GET
- POST
- PUT
- DELETE

For example, HTTP GET is defined as a data-producing method that is intended to be used by a client application to retrieve a resource, fetch data from a web server, or run a query with the expectation that the web server looks for and responds with a set of matching resources. The GET method is used to retrieve information from the server. When you use your browser to go to any URI, you use the GET method to get the HTML of that website. The query string containing 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.

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

The PUT method is used to change the state of current data in the server. In this case, the parameters are posted in the body of the request, and not in the URI. Using PUT to replace the original resource provides a much cleaner interface that is consistent with REST principles and the definition of HTTP methods.

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

REST uses default HTTP return codes, such as

- 200 – OK
- 404 – NOT FOUND
- 403 – FORBIDDEN
- 500 – INTERNAL SERVER ERROR

A URI identifies the resource to retrieve or update. The URI describes the network location of the resource.

Other architecture constraints such as Layered system, Cache and Code on demand can be found REST dissertation link in the References section.

References:

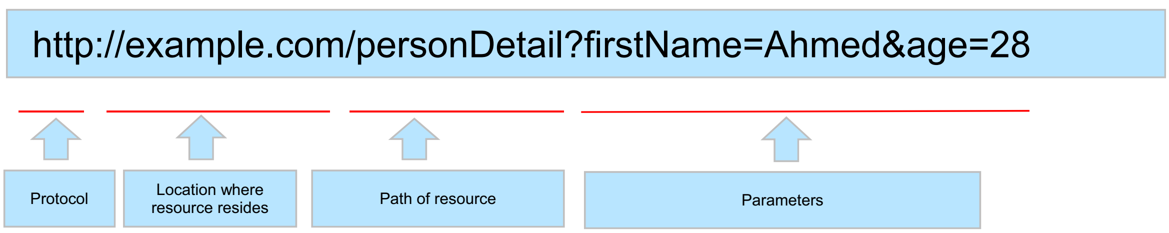
https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

https://developer.ibm.com/articles/ws-restful/

IBM Training

Resource Identifier and URIs

- In the REST world, URIs are the resources identifiers and used to create, update, retrieve, search for, and delete resources.
- The design of URIs is important because it defines how consistent your model is in accessing resources.
- A URI includes the following parts:
  - Base URL (protocol and the location where the resource is)
  - Context-Root (can be /)
  - Path of resource
  - Operation and parameters (optional)
- URI, URL, URN, and location of resource and parameters example:



URI: http://example.com/personDetail
URL: http://example.com/
URN: urn:example:persondetail:ahmed:28

REST architecture and Watson APIs

© Copyright IBM Corporation 2019

Figure 5-7. Resource Identifier and URIs

Differences between URI, URL, and URN

Every URL and URN are a URI because a URI is the superset of both a URL and URN.
URI is a text that is used to identify any resource or name. In the example, it is the location of the resource http://example.com/personDetail.

A Uniform Resource Location (URL) is a subset of a URI. A URL includes the location and protocol to retrieve the resource. In the example, it is http://example.com/.

A Uniform Resource Name (URN) is the subset of a URI that is required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable. A URN is a URI that uses the URN scheme, and **does not imply availability of the identified resource**. Both URNs (names) and URLs (locators) are URIs, and a particular URI may be both a name and a locator concurrently. A URN is similar to a person’s name, and a URL is like a street address. A URN must have the form <URN> ::= "urn:" <NID> ":" <NSS>, where <NID> is the Namespace Identifier and <NSS> is the Namespace Specific String. In the example, it is urn:example:persondetail:ahmed:28.

A URI can be simple, such as http://example.com/index.html. When the protocol is http, the location where the resource resides is example.com, the context root is /, and the path of the resource is index.html. The URL is http://example.com.

Another example of a URI is http://example.com/personDetail?firstName=Ahmed&age=28, where the protocol is http, the URL is http://example.com/ with the context root as / and the resource as personDetail. The parameters are firstName and age.

Each parameter is represented as a name-value pair. The parameters are separated by “&”. The URI for a GET request can be formatted as shown in the following example:

http://example.com/customer/customerDetail?firstName=Ahmed&age=28

A more complex example of URI could be:

http://example.com/services/customer/customerDetail?firstName=Ahmed&age=28.

This example is considered more complex because we have all elements in the URI, such as the protocol, base URL, the context root, the resource name, resource method, and parameters. The http is the protocol, the URL is http://example.com/services with the context root /services, and the customer is the resource path with operation customerDetail on the resource customer and the parameters firstName and age.

In a possible implementation/structure of the underlying backend service, the operation customerDetail could be mapped to the language method customerDetail in the resource customer.

References:

https://tools.ietf.org/html/rfc3986

https://quintupledev.wordpress.com/2016/02/29/difference-between-uri-url-and-urn/

http://www.java67.com/2013/01/difference-between-url-uri-and-urn.html#ixzz5lxzOH0Xu

IBM Training

Best practices to model URIs in REST

- URI should be simple, intuitive, easy to read, and consistent.
- Avoid having your API start the root domain. Consider using a URI like http://example.com/api instead of http://api.example.com.
- Resource endpoints should be a plural and not a singular name. For example: http://example.com/api/products
- Access a specific instance of a resource to use as an identifier in HTTP GET call. For example: http://example.com/api/products/prod123
- Do not use privileged user information as unencrypted parameters. For example: http://example.com/api/user?ssn=039456337-87
- To make the REST service more portable and consumable, allow multiple results representations such as JSON and XML. For example:
  - http://example.com/api/products?format=json
  - http://example.com/api/products?format=pdf
- HTTP POST should be used to create a resource
- Searches should not return a data overload. If required, always schedule a paging mechanism. For example: http://example.com/api/products?pageSize=10&pageNo=4

REST architecture and Watson APIs

Figure 5-8. Best practices to model URIs in REST

- A URI should be simple, intuitive, easy to read, and consistent.

  Think of a URI as a kind of self-documenting interface that requires little, if any, explanation or reference for a developer to understand what it points to and to derive related resources. To this end, the structure of a URI should be straightforward, predictable, and easily understood.

  REST interfaces are resource-based. The most important aspect of the interface design is the URI structure. It allows the consumer of REST resources to access the properties and change the state of the resource.. Using a REST approach, the create customer example looks like ***POST /customers (payload)***. Here, “customers” is an object collection that is part of the URI structure, and the action that is taken on the collection is implied by the REST verb. Were you to create an interface to retrieve a customer record, this would look like ***GET /customers/{customer-id}***, with “customer-id” now representing a particular member of the “customers” collection.

- Avoid having your API start the root domain. Consider using a URI like http://example.com/api instead of http://api.example.com.

- Resource endpoints should be a plural and not a singular name, for example: http://example.com/api/products. In this case, if you are using an HTTP GET, you collect all the products. If you are using HTTP POST, you might be creating a product, for example http://example.com/api/products.

- Access a specific instance of a resource to use as an identifier. For example, http://example.com/api/products/prod123, where prod123 is the ID of a collection of products to be retrieved.
- To make the REST service more portable and consumable, allow multiple representations of the results, such as JSON or XML. For example:

  http://example.com/api/products?format=json

  http://example.com/api/products?format=pdf
- REST via HTTP use Media-Types/MIME-Type, like "application/json" in the Content-Type HTTP Header and moreover a REST endpoint expose in the Accept Header the supported Media-Types of the service

  Header name

  - Content-Type

  Examples:

  application/json --- > Indicates that the request body format is JSON

  application/xml --- > Indicates that the request body format is XML.

  application/x-www-form-urlencoded --- > Indicates that the request body is URL encoded.

  - Accept

  Examples:

  application/json --- > Sets output type to JSON

  application/xml --- > Sets output type to XML.
- HTTP POST should be used to create a resource
- Searches should not return a data overload. If required, always schedule a paging mechanism. For example: http://example.com/api/products?pageSize=10&pageNo=4

More best practices:

- Instead of using the “404 Not Found” code if the request URI is for a partial path, always provide a default page or resource as a response.
- Hide the server-side scripting technology file extensions (.jsp, .php, and .asp) so you can port to something else without changing the URIs.
- GET without resource ID should deliver a listing.
- GET with resource ID should return the identified resource.
- GET should never change resources.

References:

https://developer.ibm.com/apiconnect/2015/01/26/basics-good-api-design/

https://www.ibm.com/support/knowledgecenter/en/SSRASJ_8.8.0/com.ibm.ima.ug_soa/soa/InfoSphere/restful/rest_intro.html

https://www.ibm.com/support/knowledgecenter/en/SS8PJ7_9.1.2/com.ibm.xtools.rest.doc/topics/t_trans_overview.html?view=kc&origURL=SS8PJ7_9.1.2/com.ibm.xtools.rest.doc/topics/t_trans_overview.html

https://developer.ibm.com/articles/ws-restful/

https://www.ibm.com/support/knowledgecenter/en/SS6PEW_9.4.0/com.ibm.help.custom.restapis.doc/c_SpecifyingHTTPHeaders.html

5.2. Applying REST to server-side applications

Figure 5-9. Applying REST to server-side applications

IBM Training

Topics

- REST concepts, architecture, and characteristics
- Applying REST to server-side applications
- Resources representation and JSON
- Example: Using REST APIs with IBM Watson

REST architecture and Watson APIs

Figure 5-10. Topics

IBM Training

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 your server applications available 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.

REST architecture and Watson APIs

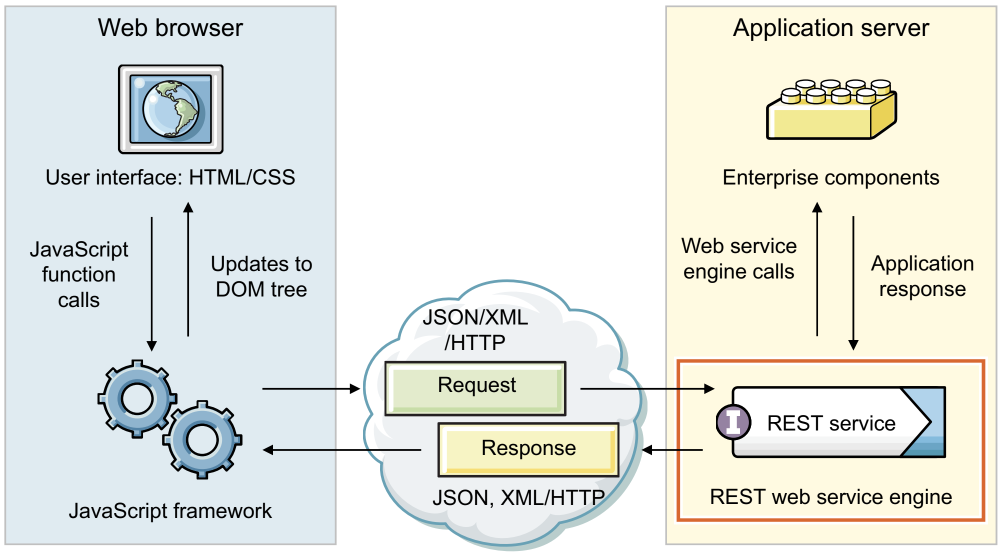
Figure 5-11. Applying REST to server-side applications

In a more general sense, web resources represent a source of information. For example, HTML documents define the structure of a web page. Cascading Stylesheet (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.

IBM Training

Example: Application Model Architecture for REST services



REST architecture and Watson APIs

Figure 5-12. Example: Application Model Architecture for REST services

In this example, enterprise 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 the HTTP protocol.

The server-side code can be in Java, NodeJS, Python, Scala, Groovy, Ruby, or other languages. They all can use REST. The client can be an other service or machine; it is not restricted to a web browser. The client can be in JavaScript, Java, Node.js, and other languages.

IBM Training

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, which are applications and sources of information. For example, 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`.

REST architecture and Watson APIs

Figure 5-13. What is a RESTful web service

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 (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 this example, send a GET operation to "/stock/IBM" to retrieve the current IBM stock price.

IBM Training

Best practices for RESTful web services

- The implementation of a REST web service follows some basic design principles:
  - Use HTTP methods explicitly:
    - To retrieve a resource, use `GET`.
    - To create a resource on the server, use `POST`.
    - To change the state of a resource or to update it, use `PUT`.
    - To remove or delete a resource, use `DELETE`.
  - Be stateless.
  - Expose directory structure-like URIs.
  - Expose a resource in some data format, such as XML, JSON, or both.

REST architecture and Watson APIs

Figure 5-14. Best practices for RESTful web services

- Use HTTP methods explicitly.

  One of the key characteristics of a RESTful web service is the explicit use of HTTP methods in a way that follows the protocol that is defined by RFC 2616. For example, HTTP GET is defined as a data-producing method that is intended to be used by a client application to retrieve a resource, fetch data from a web server, or run a query with the expectation that the web server looks for and responds with a set of matching resources.

  Developers should use HTTP methods explicitly and in a way that is consistent with the protocol definition. This basic REST design principle establishes a one-to-one mapping between create, read, update, and delete operations and HTTP methods. According to this mapping:

  - To create a resource on the server, use POST.
  - To retrieve a resource, use GET.
  - To change the state of a resource or to update it, use PUT.
  - To remove or delete a resource, use DELETE.

- Be stateless.

  REST web services must scale to meet increasingly high-performance demands. Clusters of servers with load-balancing and failover capabilities, proxies, and gateways are typically arranged in a way that forms a service topology, which allows requests to be forwarded from one server to the other as needed to decrease the overall response time of a web service call. Using intermediary servers to improve scale requires REST web service clients to send complete, independent requests, that is, to send requests that include all the data that is needed to be fulfilled so that the components in the intermediary servers may forward, route, and load balance without any state being held locally in between requests.

  A complete, independent request does not require the server while processing the request to retrieve any kind of application context or state. A REST web service application (or client) includes within the HTTP headers and body of a request all of the parameters, context, and data that is needed by the server-side component to generate a response. Statelessness in this sense improves web service performance and simplifies the design and implementation of server-side components because the absence of state on the server removes the need to synchronize session data with an external application.

- Expose directory structure-like URIs.

  From the standpoint of client applications addressing resources, the URIs determine how intuitive the REST web service is going to be and whether the service is going to be used in ways that the designers can anticipate. A RESTful web service characteristic is all about the URIs.

- Expose a resource in some data format, such as XML, JSON, or both

  A resource representation typically reflects the current state of a resource and its attributes when a client application requests it. Resource representations in this sense are snapshots in time. They might be as simple as a representation of a record in a database that consists of a mapping between column names and XML tags where the element values in the XML contain the row values. If the system has a data model, then according to this definition a resource representation is a snapshot of the attributes of one of the things in your system's data model. These are the things that you want your REST web service to serve.

  The last set of constraints that goes into a RESTful web service design has to do with the format of the data that the application and service exchange in the request and response payload or in the HTTP body. This is where it really pays to keep things simple, human-readable, and connected.

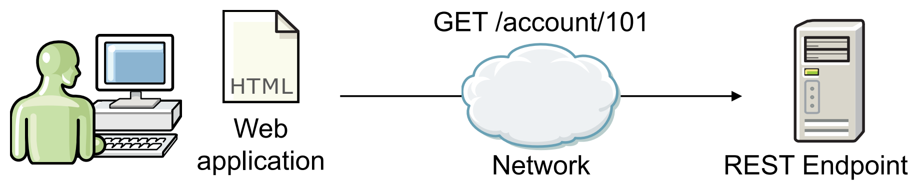
  The objects in your data model are usually related in some way, and the relationships between data model objects (resources) should be reflected in the way they are represented for transfer to a client application. In the discussion-threading service, an example of connected resource representations might include a root discussion topic and its attributes, and embed links to the responses given to that topic.

Reference:

https://developer.ibm.com/articles/ws-restful/

IBM Training

Example: Sending an HTTP GET request to a REST service




```
GET /account/101 HTTP/1.1
Host: example.com
Accept: application/json,
        application/xml
```

1. To retrieve a web resource from the server, the client web application sends an HTTP GET request.
2. The name of the server resource is `/account/101`. This resource path represents an account record with an ID value of 101.

REST architecture and Watson APIs

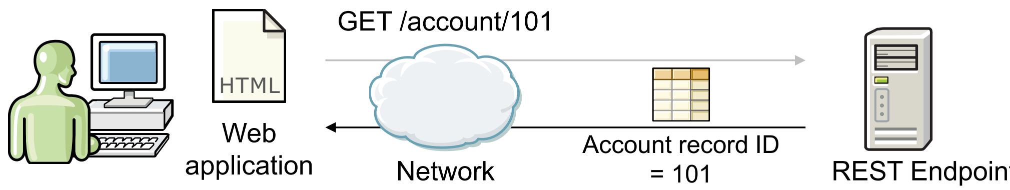
Figure 5-15. Example: Sending an HTTP GET request to a REST service

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 go to a URL on your browser, your browser automatically sends a GET request to retrieve the requested page.

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

IBM Training

Example: Receiving an HTTP GET response from a REST service




```
HTTP/1.1 200 OK
Content-Length: 81
Content-Type: application/json
Accept: application/json
Date: Wed, 19 Jun 2013 13:19:23 GMT
{
 "name": "John",
 "accountId": "101",
 "type": "savings",
 "balance": 50.00
}
```

3 The HTTP status code of 200 indicates that the REST service operation completed successfully.

4 The body of the HTTP response message contains information about the account in a JSON data format.

REST architecture and Watson APIs

Figure 5-16. Example: Receiving an HTTP GET response from a REST service

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 and version, HTTP status code, Content-Length, Content-Type, Date, and Response Body.

In this example, the protocol type and 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.

Other HTTP status codes might be the following ones:

- 404 – NOT FOUND
- 403 – FORBIDDEN
- 500 – INTERNAL SERVER ERROR

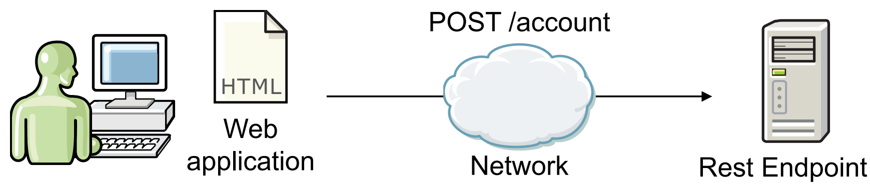
Content-Length contains the length of the response message, which in this example is 81 characters.

Content-Type describes the data type of the response, which in this example is JSON. Different content-types are supported, such as XML, Text/Plain, and HTML. The request also has an Accept Header which request a specific MIME-type for response.

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

IBM Training

Example: Sending an HTTP POST request to a REST service




```
POST /account/ HTTP/1.1
Host: example.com
Content-Type: application/json
Accept: application/json,
        application/xml
{
 "name": "John",
 "type": "savings",
 "ssn": "12345678"
}
```

1. To create a web resource from the server, the client web application sends an HTTP POST request.
2. The name of the server resource is `/account`. The user completes the HTTP form body with required values to create an account.

REST architecture and Watson APIs

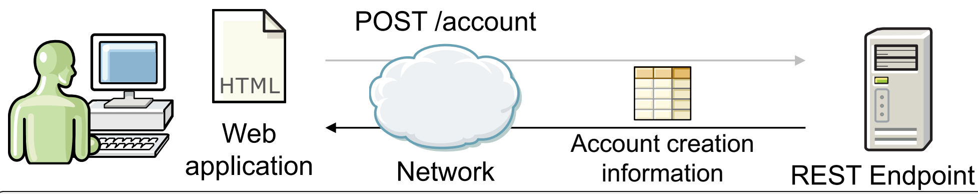
Figure 5-17. Example: Sending an HTTP POST request to a REST service

In this example, a client application running in the web browser sends an HTTP POST request to the resource on the server. Notice that the HTTP POST is used to create a record. Conversely, HTTP GET is used to retrieve data.

In most cases, an HTTP form is used to send information in a single request. Some fields in the form can be private data and they must be encrypted by the HTTPS request. In this example, the request is to create a bank account. The name of the server resource is “/account” without extra parameters because all the data is sent in the HTTP form. The format of the POST request can be in XML or JSON.

IBM Training

Example: Receiving an HTTP POST response from a REST service




```
HTTP/1.1 200 OK
Content-Length: 81
Content-Type: application/json
Date: Wed, 19 Jun 2019 13:19:23 GMT
{
 "name": "John",
 "accountId": "101",
 "type": "savings",
 "balance": 0.00,
 "activeDate" : "2019-04-19"
}
```

3 The HTTP status code of 200 indicates that the REST service operation completed successfully.

4 The body of the HTTP response message can contain some information about the create account process in a JSON data format.

REST architecture and Watson APIs

Figure 5-18. Example: Receiving an HTTP POST response from a REST service

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

In this example, the protocol type and 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.

Other HTTP status codes might be:

- 404 – NOT FOUND
- 403 – FORBIDDEN
- 500 – INTERNAL SERVER ERROR

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

Content-Type describes the data type of the response, which in this example is JSON. Different content-types are supported like XML, Text/Plain, and HTML.

Response Body is a JSON object that contains four name-value pairs that contain the values of account creation confirmation, such as name, the account ID that was created by using some server processing, the initial balance, and the activeDate of account.

5.3. Resources representation and JSON

IBM Training

Resources representation and JSON

REST architecture and Watson APIs

Figure 5-19. Resources representation and JSON

IBM Training

Topics

- REST concepts, architecture, and characteristics
- Applying REST to server-side applications
- Resources representation and JSON
- Example: Using REST APIs with IBM Watson

Figure 5-20. Topics

IBM Training

Resource representation in REST

- A resource representation typically reflects the state of a resource and its attributes when a client application requests it. Resource representations in this sense are mere snapshots in time.
- The resource representation might be as simple as a representation of a record in a database that consists of a mapping between column names and a format.
- If the system has a data model, then according to this definition a resource representation is a snapshot of the attributes of one of the things in your system's data model.
- The more common data representations in REST are:
  - JSON
  - XML
  - HTML
  - Text/Plan
- JSON is one of most popular formats to represent resource data.

REST architecture and Watson APIs

Figure 5-21. Resource representation in REST

Reference:

https://developer.ibm.com/articles/ws-restful/

IBM Training

Introduction to JSON

- *JSON* is a text format for structured data:
  - Its syntax is derived from the object literals of JavaScript, according to *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 and 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

REST architecture and Watson APIs

Figure 5-22. Introduction to JSON

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 the scripting language standard.

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

The main design goal of JSON is to provide a minimal and 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 built on two structures: A collection of name-value pairs that are known as *objects* and a list of values that are known as *arrays*.

IBM Training

JSON data types

`"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.

REST architecture and Watson APIs

Figure 5-23. JSON data types

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. They 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.

IBM Training

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"
}
```

REST architecture and Watson APIs

Figure 5-24. JSON data type: Objects

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

IBM Training

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]
]
```

REST architecture and Watson APIs

Figure 5-25. JSON data type: Arrays

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 this slide. 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.

5.4. Example: Using REST APIs with IBM Watson

IBM Training

Example: Using REST APIs with IBM Watson

REST architecture and Watson APIs

Figure 5-26. Example: Using REST APIs with IBM Watson

IBM Training

Topics

- REST concepts, architecture, and characteristics
- Applying REST to server-side applications
- Resources representation and JSON
- Example: Using REST APIs with IBM Watson

REST architecture and Watson APIs

Figure 5-27. Topics

IBM Training

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.
- You do not need to know the details of every associated AI subfield.
- You must have a high-level understanding of each subfield.
- You must know how to apply the correct AI technology to the problem by using AI application programming interfaces (APIs) or a ready-to-use AI framework.

REST architecture and Watson APIs

Figure 5-28. IBM Watson

IBM Watson is an AI system that enables a new partnership between people and computers. It is the AI offering from IBM. Watson combines five core capabilities:

- Interacts with people more naturally based on the person's preference.
- Quickly imports key industry materials by partnering with experts to scale and elevate expertise.
- Enables new products and services to sense, reason, and learn about users and the world around them.
- Uses data to improve business processes and forecasting, which increases operational effectiveness.
- Enhances exploration and discovery, which uncovers unique patterns, opportunities, and actionable hypotheses.

Data, information, and expertise create the foundation for working with Watson. Watson works with structured and unstructured data. Structured data examples are CSVs; Excel files; and databases, and unstructured data examples are Word documents; PDFs; and HTML files.

IBM Watson includes AI technologies. Those technologies are complex, and to understand them fully requires many years of study. However, to add AI capabilities to your applications, you do not have to know the details that are associated with each AI subfield. You can add AI capabilities by gaining a high-level understanding of each AI subfield, for example, NLP, computer vision, and machine learning. Then, you must know how to apply the appropriate AI technology to a problem by using AI APIs or AI frameworks.

IBM Training

How Watson works

What is Watson, and how is it helping businesses across the globe to build a smarter future?

Watch the following video to learn about Watson technology or discover the different parts that make up Watson:

https://www.youtube.com/watch?v=r7E1TJ1HtM0

REST architecture and Watson APIs

Figure 5-29. How Watson works

IBM Watson is available as:

- A set of AI services on IBM Cloud
- Industry solutions for various industries
- Tools, documentation, and samples for developers

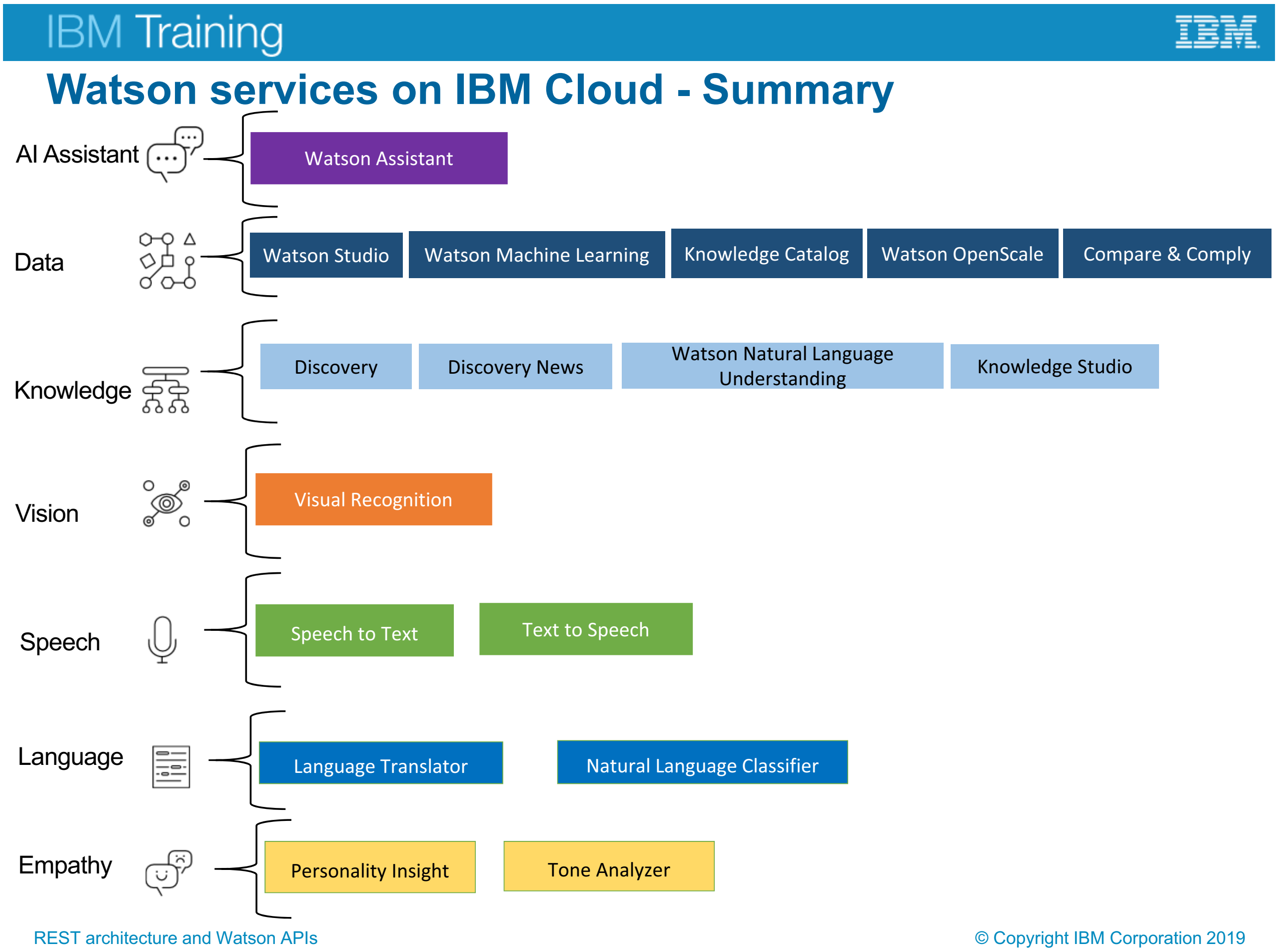


Figure 5-30. Watson services on IBM Cloud - Summary

The figure shows the Watson services on IBM Cloud at the time of writing. These services will continue to be enhanced, and new services will be introduced in the future.

The following services are available as REST APIs and software as a solution (SaaS) tools that developers can use to build AI solutions or add AI features to their applications. For more information, see https://www.ibm.com/watson/products-services/.

AI Assistant (Chatbot): Integrate diverse conversation technologies into your application:

- Watson Assistant: Quickly build a chat bot by using intents, entities and dialog trees. It uses Natural Language Processing and pattern recognition techniques.

Data: Collect, organize, and analyze your data, and then achieve trust, scale, and automation across your full AI lifecycle:

- Watson Studio: A collaborative environment with AI tools that a team can use to collect and prepare training data, and to design, train, and deploy machine learning models.
- Watson Machine Learning: Enables users to perform two fundamental operations of machine learning: training and scoring.
- Watson Knowledge Catalog: Machine learning data catalog that enables you to access, curate, categorize, and share data, knowledge assets, and their relationships, wherever they are.
- Watson OpenScale: Operate and automate AI at scale by infusing it with trust and transparency, explaining its outcomes, and eliminating harmful bias.
- Watson Compare & Comply: Extract data from contracts and governing documents to increase productivity, reduce costs, and minimize exposure.

Knowledge: Get insights through accelerated data optimization capabilities:

- Discovery: Unlock hidden value in data to find answers, monitor trends, and surface patterns.
- Discovery News: Explore news and blogs with smarter news from Watson that includes concepts, sentiment, relationships, and categories.
- Watson Natural Language Understanding: NLP for advanced text analysis.
- Knowledge Studio: Teach Watson to discover meaningful insights in unstructured text.

Vision: Identify and tag content, and then analyze and extract detailed information that is found in an image:

- Visual Recognition: Tag and classify visual content by using machine learning.

Speech: Converts text and speech with the ability to customize models:

- Speech to Text: Easily converts audio and voice into written text.
- Text to Speech: Converts written text into natural-sounding audio.

Language: Analyzes text and extracts metadata from unstructured content:

- Language Translator: Translates text from one language to another.
- Natural Language Classifier: Interprets and classifies natural language. Applies NLP and machine learning techniques to return the best matching classes for a sentence or phrase.

Empathy: Understands tone, personality, and emotional state:

- Personality Insights: Predicts personality characteristics through text.
- Tone Analyzer: Understands emotions and communication style in text.

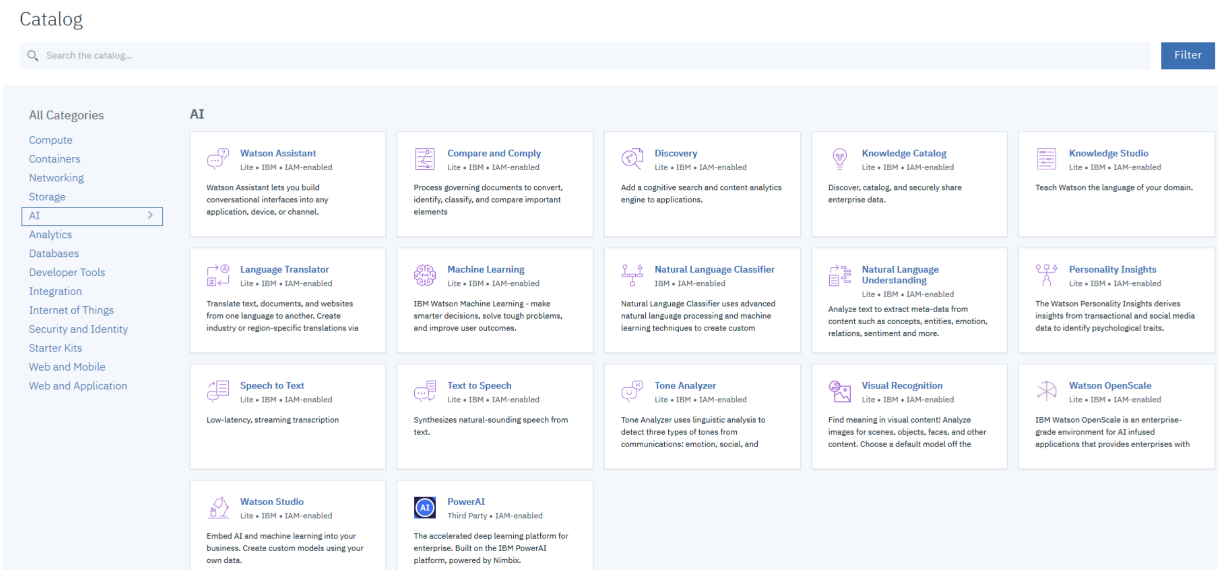
References:

https://www.ibm.com/watson/products-services/

https://www.ibm.com/cloud/compare-and-comply

IBM Training

AI services on IBM Cloud catalog



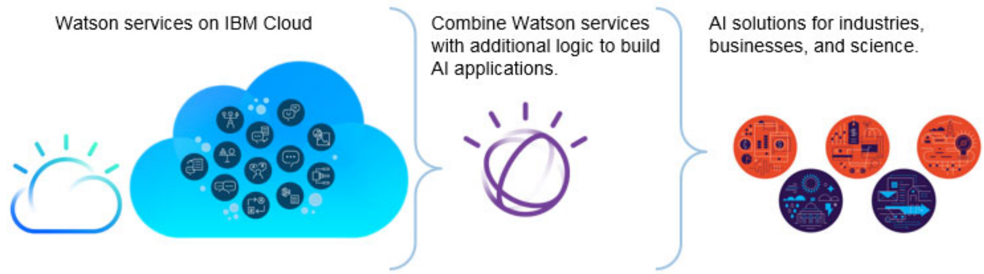
REST architecture and Watson APIs

Figure 5-31. AI services on IBM Cloud catalog

You can access the Watson services from the IBM Cloud catalog at https://cloud.ibm.com/catalog?category=ai.

IBM Training

Building AI solutions with IBM Watson services on IBM Cloud



REST architecture and Watson APIs

Figure 5-32. Building AI solutions with IBM Watson services on IBM Cloud

Watson services on IBM Cloud provide a cloud-hosted marketplace where application providers of all sizes and industries can tap into resources for developing applications that are powered by Watson services. Developers can combine the Watson services (and other services that are available in IBM Cloud) with extra logic to build applications with AI capabilities.

The goal of Watson services on the IBM Cloud is to provide a flexible platform for building AI applications in industry domains. The microservices architecture enables developers to envision a broad range of potential applications by mixing and matching services.

The available resources include developer toolkits, educational materials, and access to Watson APIs and SaaS tools. This approach makes IBM Watson technology available as a development platform in the cloud to enable a worldwide community of software application providers to build a new generation of applications that is infused with Watson AI capabilities.

IBM Training

Getting started with Watson services on IBM Cloud

You must complete these steps to use the Watson services on IBM Cloud:

1. Set up your IBM Cloud account.
2. Find and select a Watson service in the IBM Cloud catalog.
3. Create the service instance.
4. Get the service credentials to authenticate to your service from your application.
5. Start coding your application and calling the Watson APIs to infuse AI capabilities into your app.

REST architecture and Watson APIs

Figure 5-33. Getting started with Watson services on IBM Cloud

To get started with Watson services on IBM Cloud, you must complete these steps before you can use the service in your applications:

1. Set up your IBM Cloud account.
2. Find and select a Watson service that you need from the IBM Cloud catalog.
3. Create a service instance. IBM Cloud provides resources to your applications through a service instance.
4. Get service credentials. You need the service credentials to authenticate to your service instance from your application.
5. Start coding your application and calling the Watson APIs to infuse AI capabilities into your app.

IBM Training

Options to call Watson APIs

- There are basically two options to call APIs:
  - Using any REST API client like cURL or Postman.
    By using `curl` or the Postman plug-in in your browser, you can run commands to Watson services endpoints. For example, using cURL:

```
curl -X POST \ -H "Content-Type: application/json" \ -u
"apikey:{apikey}" \ -d @parameters.json \
"{url}/v1/analyze?version=2018-11-16“
```

  - Using SDKs
    You can use the native language wrappers that call Watson Services. There are several options, such as Node.js, Python, Java, Go, dotnet, Swift, and Unit. For more information, see:
    https://github.com/watson-developer-cloud
- Data input and output on Watson API calls
  - APIs support JSON for non-binary data input and output, and might also support CSV or XML. Formats default to JSON when not specified.

REST architecture and Watson APIs

Figure 5-34. Options to call Watson APIs

Options to call Watson APIs:

- Using any REST API client like cURL or Postman
  By using curl.exe or Postman in your browser, you can run commands to call Watson services endpoints. This approach is a programming language neutral way to call Watson services and use the REST API architecture. For example, you can use cURL by running the following command:

  ```
  curl -X POST \ -H "Content-Type: application/json" \ -u "apikey:{apikey}" \ -d
  @parameters.json \ "{url}/v1/analyze?version=2018-11-16“
  ```

  In this case, you are running an HTTP POST to analyze text that is inside the “@parameters.json” file by calling the Watson Natural Language Understanding service. The end point is “analyze” and “{apikey}” and “{url}” must be completed with information that is provided after you create the Watson Natural Language Understanding service.

- Using SDKs

  You can use native language wrappers that call Watson services. There are several options, such as Node.js, Python, Java, Go, dotnet, Swift, and Unit. For more information, see the following website:

  https://github.com/watson-developer-cloud

- Data input and output in Watson API calls

  APIs support JSON format for non-binary data input and output, and might also support CSV or XML, by using the Content-Type (for input) and accept (for output) headers to specify formats. Formats default to JSON when not specified. For more information, see the following website:

  https://github.com/watson-developer-cloud/api-guidelines#json-vs-csv-vs-xml

IBM Training

Watson service example: Watson Natural Language Understanding

- Analyze semantic features of text input, including the following items:

  Categories, concepts, emotions, entities, keywords, metadata, relations, semantic roles, and sentiment.

- Categorize content.
- Develop custom annotation models to identify domain-specific entities and relations in unstructured text by using Knowledge Studio.
- Example applications: Categorize news articles and blog posts and sort them based on general concepts, keywords, and entities.

REST architecture and Watson APIs

Figure 5-35. Watson service example: Watson Natural Language Understanding

This slide shows just one example of a Watson service.

Watson Natural Language Understanding

This service analyzes text to extract metadata from content, such as categories, concepts, emotions, entities, keywords, metadata, relations, semantic roles, and sentiment.

Here are summaries of the metadata that is returned by Watson Natural Language Understanding:

- Categories

  Identify high-level concepts that are not necessarily directly referenced in the text.

- Concepts

  Categorize your content by using a five-level classification hierarchy. You can view the complete list of categories at the following website:

  https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-categories-hierarchy

- Emotions

  Analyze emotions that are conveyed by specific target phrases or by the document as a whole. You can also enable emotion analysis for entities and keywords that are automatically detected by the service.

- Entities

  Find people, places, events, and other types of entities that are mentioned in your content. You can view the complete list of entity types and subtypes at the following website:

  https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-entity-type-systems

  For example:

  Input

  Text: "I love apples, but I hate oranges."
  Targets: "apples", and "oranges"

  Response

  "apples": joy
  "oranges": anger

- Keywords

  Search your content for relevant keywords. For example:

  Input

  URL:"http://www-03.ibm.com/press/us/en/pressrelease/51493.wss"

  Response

  Australian Open
  Tennis Australia
  IBM SlamTracker analytics

- Metadata

  For HTML and URL input, get the author of the webpage, the page title, and the publication date.

- Relations

  Recognize when two entities are related, and identify the type of relation. For example:

  Input

  Text: "The Nobel Prize in Physics 1921 was awarded to Albert Einstein."

  Response

  "awardedTo" relation between "Noble Prize in Physics" and "Albert Einstein"
  "timeOf" relation between "1921" and "awarded"

- Semantic roles

  Parse sentences into subject-action-object form, and identify entities and keywords that are subjects or objects of an action. For example:

  Input

  Text: "In 2011, Watson competed on Jeopardy!"

  Response

  Subject: Watson
  Action: competed
  Object: on Jeopardy

- Sentiment

  Analyze the sentiment toward specific target phrases and the sentiment of the document as a whole. You can also get sentiment information for detected entities and keywords by enabling the sentiment option for those features. For example:

  Input

  Text: "Thank you and have a nice day!"

  Response

  Positive sentiment (score: 0.91)

- Custom annotation models
  Custom annotation models are developed by using Watson Knowledge Studio to identify industry- and domain-specific entities and relations in unstructured text.

- Example of applications

  Categorize news articles and blog posts and sort them based on general concepts, keywords, and entities.

Reference:

https://cloud.ibm.com/apidocs/natural-language-understanding

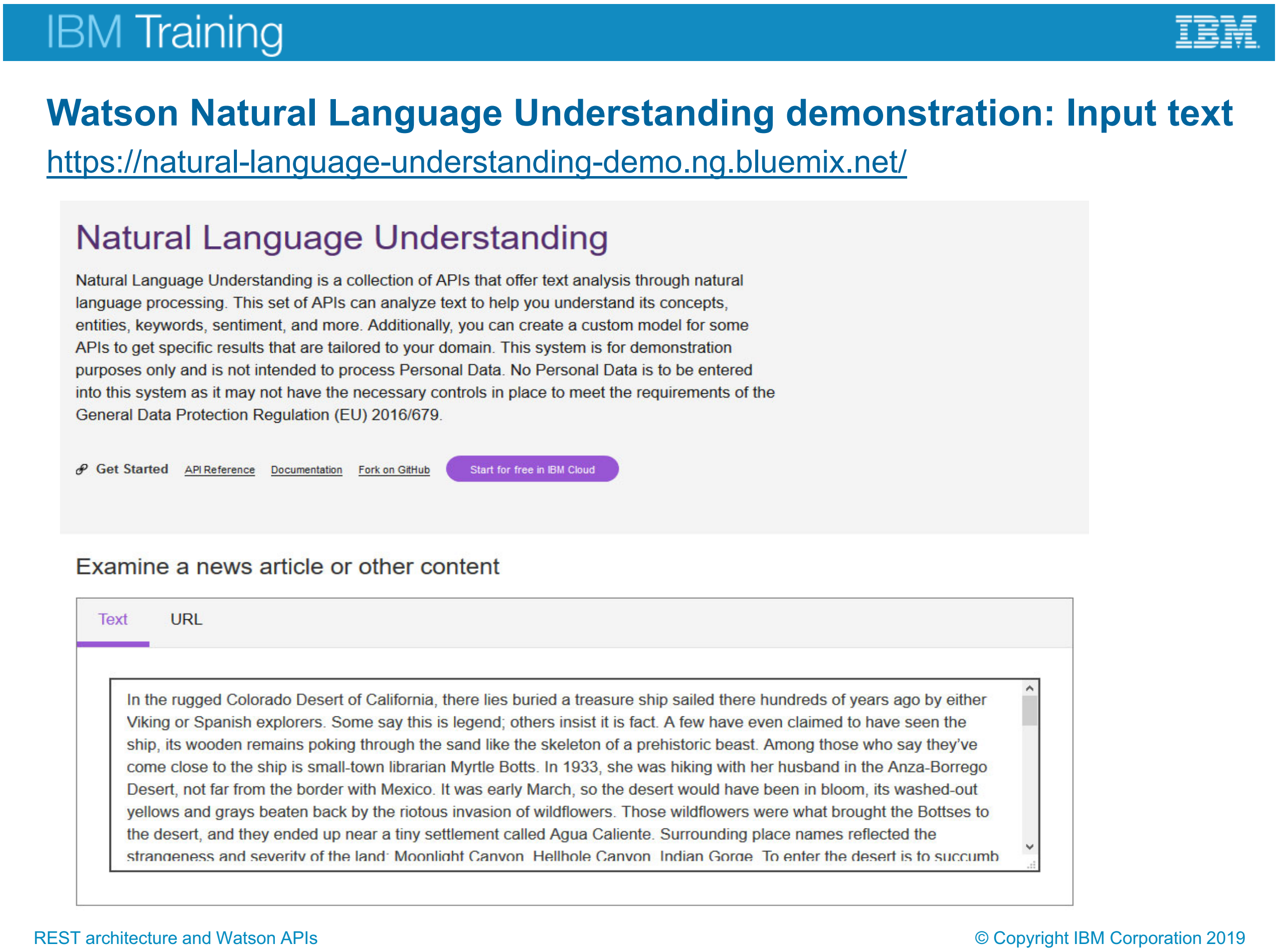


Figure 5-36. Watson Natural Language Understanding demonstration: Input text

For a demonstration of the Watson Natural Language Understanding API, go to this demonstration: https://natural-language-understanding-demo.ng.bluemix.net/

This demonstration runs the *analyze* function of the Watson Natural Language Understanding REST API on text that is provided by a user or from a URL link. The analyze function extracts the information from a text or URL and provides some advanced text analytics results, such as categories, concepts, emotions, entities, keywords, metadata, relations, semantic roles, and sentiment.

Note that *syntax* is an experimental output in Watson Natural Language Understanding that returns texts as *tokens* with a structure of the token as it appears in the analyzed text:

- *Part of speech* of the token, such as ADJ (Adjective)
- NOUN (name),
- PRON (pronoun)
- *Location* indicates the character offset that shows the beginning and end of the token
- *Lemma* is the canonical form, dictionary form, or citation form of a word.

Supported languages

Support for these text analytics elements depends on the language in which you are providing the text. For more information about supported languages and features, see the following website:

https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-language-support

Reference:

https://cloud.ibm.com/apidocs/natural-language-understanding#analyze-text

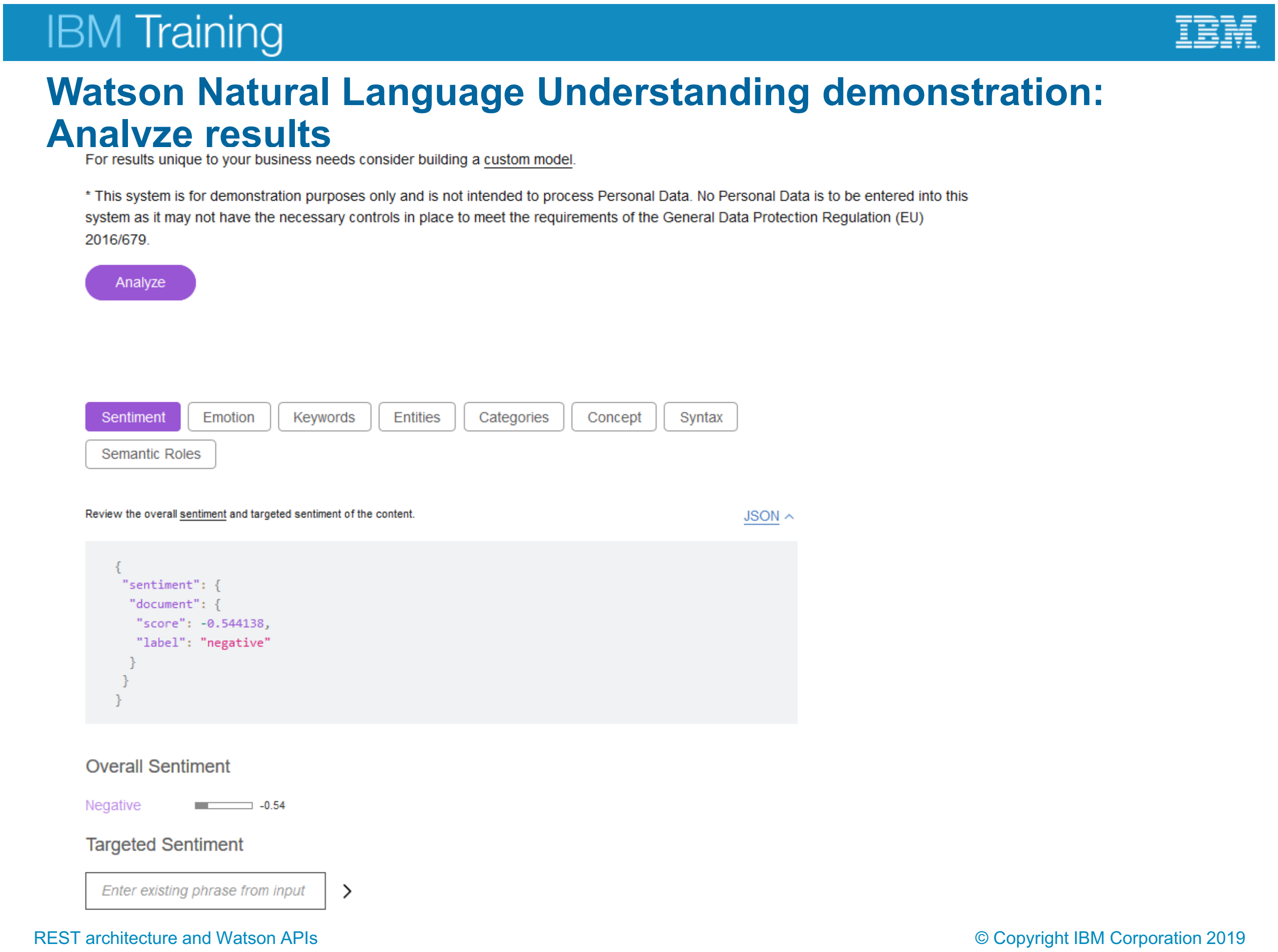


Figure 5-37. Watson Natural Language Understanding demonstration: Analyze results

Watson Natural Language Understanding includes a set of text analytics features that you can use to extract meaning from unstructured data. The Watson Natural Language Understanding API Reference documentation includes examples in several programming languages that help you get started. This slide shows a sentiment feature request in a node.

This example shows the sentiment view of the demonstration application, and in this case sentiment analyzes the general sentiment of the content. When you click **JSON**, you see the score of the sentiment. The example shows a negative number score, which means a negative sentiment.

Reference:

https://cloud.ibm.com/apidocs/natural-language-understanding?code=node#sentiment

IBM Training

Additional resources for developers

For more information, see the following resources:

- Explore the complete list of Watson APIs: Watson Products and Services
- Get started on IBM Cloud: IBM Cloud essentials
- Access developer's resources:
  - Build with Watson
  - Documentation and API Reference
  - Watson SDKs
  - Building Cognitive Applications with IBM Watson Services
  - AI articles and tutorials
  - Watson webinars
  - Building with Watson: Application Starter Kits for developers
  - Watson Starter Kits

REST architecture and Watson APIs

Figure 5-38. Additional resources for developers

- Watson products and services: https://www.ibm.com/watson/products-services/
- IBM Cloud essentials: https://cognitiveclass.ai/courses/ibm-cloud-essentials/
- Build with Watson: https://cloud.ibm.com/developer/watson/dashboard
- Documentation and API Reference: https://cloud.ibm.com/developer/watson/documentation
- Watson SDKs: https://cloud.ibm.com/docs/services/watson?topic=watson-using-sdks#sdks
- Building Cognitive Applications with IBM Watson Services: http://www.redbooks.ibm.com/redbooks.nsf/pages/cognitiveapps?Open
- AI articles and tutorials: https://developer.ibm.com/technologies/artificial-intelligence/
- Watson webinars: https://www.ibm.com/watson/webinars/
- Building with Watson: Application Starter Kits for developers: https://developer.ibm.com/tv/building-with-watson-application-starter-kits-for-developers/
- Watson Starter Kits: https://cloud.ibm.com/developer/watson/starter-kits

IBM Training

Documentation and other information sources

- REST APIs:
  - *RESTful web services* - The basics at IBM Developer: https://www.ibm.com/developerworks/library/ws-restful/
  - *ECMA-262 ECMAScript Language Standard, Third Edition*: http://www.ecma-international.org/publications/standards/Ecma-262.htm
- IBM CEO Ginni Rometty describes a new era in technology and business: https://www.youtube.com/watch?v=bMLYKhiZCVI

REST architecture and Watson APIs

Figure 5-39. Documentation and other information sources

IBM Training

Unit summary

- Describe the architecture of Representational State Transfer (REST).
- List best practices for using REST in your applications.
- Describe the representation format of data in REST.
- Explain the advantages of the JavaScript Object Notation (JSON) data format.
- List the IBM Watson services on IBM Cloud.
- Provide examples of Watson services REST APIs.

REST architecture and Watson APIs

Figure 5-40. Unit summary

IBM Training

Review questions

1. **True or false**. The REST architecture recommends the HTTP `GET` method to retrieve data.
2. **True or false**. A JSON array is an unordered set of values.
3. 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.
4. **True or false.** You can combine Watson services with extra logic to build AI applications by using REST APIs calls.
5. Which of these statements is **false** about Watson:
   a. IBM Watson is the AI offering from IBM.
   b. Data, information, and expertise create the foundation for working with Watson.
   c. Watson works with structured data only.
   d. Watson uses machine learning technologies to reveal insights from large amounts of unstructured data.

REST architecture and Watson APIs

Figure 5-41. Review questions

IBM Training

Review answers

1. **True.** The HTTP `GET` method is preferred way to retrieve resource data.
2. **False**. A JSON array is an *ordered* collection of values.
3. **d**.
4. **True.** You can use Watson Services by calling those services as REST APIs, and with extra logic you can build AI applications.
5. **c**.

REST architecture and Watson APIs

Figure 5-42. Review answers

Unit 6. Introduction to data services on IBM Cloud

Estimated time

01:30

Overview

This unit provides an overview of the types of data stores that are used in cloud computing. You will also learn about the data services offerings that are available through the cloud development platform.

IBM Training

Unit objectives

- Describe different databases types and capabilities
- Describe the main types of data services in IBM Cloud.
- Explain the benefits of IBM Cloudant.
- Access Cloudant databases and documents on IBM Cloud.
- Use HTTP APIs to interact with Cloudant database.

Introduction to data services on IBM Cloud

Figure 6-1. Unit objectives

6.1. Introduction to databases

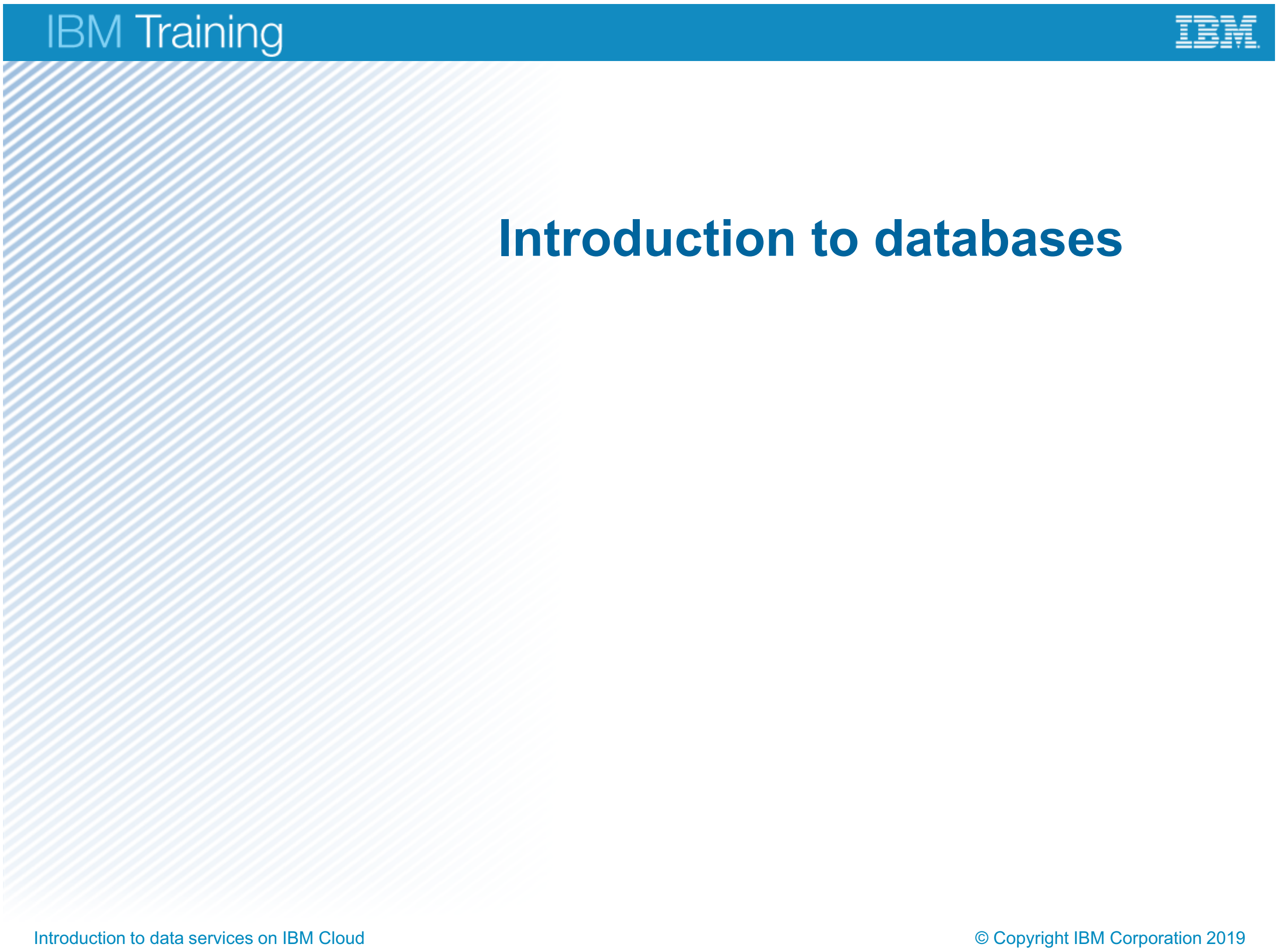


Figure 6-2. Introduction to databases

IBM Training

Topics

- Introduction to databases
- Data services in IBM Cloud
- IBM Cloudant

Introduction to data services on IBM Cloud

Figure 6-3. Topics

IBM Training

Importance of data

- Data is a set of facts, statistics, or figures.
- Raw data is processed to produce useful information.
- Structured data versus unstructured data.
- Leading organizations excel at capitalizing on data.

Introduction to data services on IBM Cloud

Figure 6-4. Importance of data

Data is defined as a set of facts, statistics, or figures. It can be in many formats, such as text documents, images, audio, or videos.

Raw data, which is the most basic format of data, is processed to produce useful information. Data processing and analysis helps modern organizations increase their productivity and make better business decisions.

Data can be categorized into two main categories: structured data and unstructured data. Structured data is the formatted and highly organized data that can fit easily into data models with fixed fields. An example is a list of students or employees data, including their names, ages, and addresses. Unstructured data is the opposite of the structured data. It is unorganized, raw and has no formal structure and it is considered as loosely structured data. For example unstructured text and multimedia like email messages, webpages, documents, photos, audio files and videos.

There is a popular saying that data is the new oil because the data and the information that is obtained by processing the data play an important role in modern organizations and contribute to the development of new business models. The organizations that are considered the most successful ones are those that can capture, manage, and derive key insights from their corporate data. Cloud technologies enable small organizations to design, set up data platforms, and use data analysis services on the cloud quickly and receive benefits from the scalability, reliability, and quality of service that is provided by the cloud. These factors help these organizations to evolve quickly and grow up faster in the market.

IBM Training

How data is stored

- Flat files (including XML files)
- Excel spreadsheets
- Relational databases (for example, Db2, MySQL, and PostgreSQL)
- NoSQL databases (Cloudant, MongoDB, and Redis)
- Object-based storage (IBM Cloud Object Storage)

Introduction to data services on IBM Cloud

Figure 6-5. How data is stored

Capitalizing on data requires storage for that data. There are many formats for storing this data. For example:

- Flat files (including XML files)
- Excel spreadsheets
- Relational databases (for example, Db2, MySQL, and PostgreSQL)
- NoSQL databases (Cloudant, MongoDB, and Redis)
- Object-based storage (IBM Cloud Object Storage)

In enterprise systems (systems that are used to run businesses), data is stored by using a database.

IBM Training

Database and data model

- A database is a collection of information that is organized so that the data can easily be accessed, managed, and updated.
- Modern organizations use various databases types to organize and store their data.
- A data model is a conceptual representation of the data structures that are required by the database.

Introduction to data services on IBM Cloud

Figure 6-6. Database and data model

Databases are used by modern organizations to organize and store their data so that the data can easily be accessed, managed, and updated. A *database management system (DBMS)* is system software for creating and managing databases. A DBMS provides users and programmers with a systematic way to create, retrieve, update, and manage data.

A *data model* focuses on the type of data that is required, the way it must be organized, and the manipulation process that is performed on the data to provide a complete and accurate structure for data within the information system. A data model organizes data elements and standardizes how the data elements relate to each other.

Two data models that have many database implementations are the *relational* and *NoSQL* data models. The following slides describe these data models through their databases.

IBM Training

Relational databases

A *relational database* is a persistent storage mechanism whose data is stored in tables with a well-defined relationship between database tables.

Field	Definition	Value
user_id	int	1
first_name	varchar2(32)	Peter
last_name	Varchar2(32)	Smith
dept_id	int	1
location_id	Int	1
start_date	date	01-01-2010

Field	Definition	Value
dept_id	int	1
manager_id	int	1
dept_name	Varchar2(24)	Sales
cost_code	Varchar2(6)	S001

Field	Definition	Value
location_id	int	1
Country	Varchar2(20)	UK
city	Varchar2(20)	Slough
address	Varchar2(512)	1 High St

Introduction to data services on IBM Cloud

Figure 6-7. Relational databases

A *relational database* is a database that works on the relational model that was described by Edgar F Codd in 1969. A relational database stores the data in a set of tables so that the data can be easily accessed by using *Structured Query Language (SQL)* statements, which is why a relational database is sometimes called an SQL database.

The key features of the relational database are as follows:

- Data is split over many tables to avoid duplication.
- Primary and unique keys are defined to prevent duplicate rows.
- Integrity is maintained by using foreign keys to prevent rows from referring to locations and departments that no longer exist.
- Follow the *Atomicity – Consistency – Isolation – Durability (ACID)* properties for all transactions in the relational database models.
- Relational databases cannot work with unstructured data, so unstructured data must be analyzed, organized, and transformed to structure data to be stored in relational databases.

IBM Training

NoSQL databases

- A NoSQL database provides a mechanism for storage and retrieval of data that is modeled by means other than the tabular relations that are used in relational databases.
- Main key characteristics:
  - Highly scalable
  - Flexible data schema
- Different types of NoSQL databases:
  - Key-Value
  - Document
  - Columnar
  - Graph

Introduction to data services on IBM Cloud

Figure 6-8. NoSQL databases

A NoSQL database (originally referred to as "non-SQL" or "nonrelational") provides a mechanism for storage and retrieval of data that is modeled by means other than the tabular relations that are used in relational databases. Its origin comes from the 21st century website architectures that faced challenges from the relational model.

The approach that is followed by a NoSQL database is as follows:

- Unlike the relational database, a NoSQL database is eventually consistent and can provide read/write operations at a lower latency.
- Renormalize the data model: Data can be stored as it is used in the application.

The key features of a NoSQL database are the following ones:

- Flexibility and scalability: IT provides flexibility because some NoSQL databases are tuned for speed at the expense of data consistency, and other NoSQL databases sacrifice speed for scalability.
- Schema-less: The data is not stored in strictly defined schemas like those in a relational database.
- It does not require a predefined data model for storage, such as specific row and column names and sizes.
- It is optimized to work on distributed hardware.
- It uses relatively simple queries that can be processed quickly across much larger data sets.
- It can process unstructured data and store it in its original format.

IBM Training

Different types of NoSQL databases: Key-value

- A simplistic data schema with a simple list of keys and values. The key is a pointer to the value.
- The key can be a hash value or a real value, such as an email address or other unique reference number. The contents of the value are not formatted.
- Key-value stores allow fast access.

Key	Value
Name	Peter Smith
Department	Sales
Location	{country: ‘UK’, city: ‘Slough’}

Introduction to data services on IBM Cloud

Figure 6-9. Different types of NoSQL databases: Key-value

Here are the pros and cons of a key-value:

Pros

- The data model is flexible (no predefined structure).
- The data structure is more application-oriented and simplifies the application design (no mapping between code and the relational database).

Cons

- There are no association between attributes that can construct a real data model.
- The data model is more application-oriented and less reusable by different applications.
- The database does not support complex queries (no join or aggregation queries, and update and delete can be done only by primary key).

Different types of NoSQL databases: Document

Document NoSQL databases pair each key with a complex data structure that is known as a *document*. Documents can contain many different key-value pairs, key-array pairs, or nested documents.

Figure 6-10. Different types of NoSQL databases: Document

Here are the main characteristics of a document NoSQL database:

- Data is stored as documents in the JSON or XML formats. The structure of each document is not imposed; each document has a flexible schema (schema-less).
- The database supports add, update, or deletion operations for some fields in a document and the ability to create an index of the document fields, which enables fast access without using a primary key and more complex requests than a key-value pair.
- Indexing and analytical queries are available by using paradigms like MapReduce. MapReduce is a way of processing huge amounts of data in parallel by breaking it into small blocks or pieces across many nodes and then combining or reducing the results of those nodes.
- Does not support object relational mappings like relational databases.

IBM Training

Different types of NoSQL databases: Columnar

- A *columnar* database is a database that stores data in columns instead of rows. For example:
  - 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

Introduction to data services on IBM Cloud

Figure 6-11. Different types of NoSQL databases: Columnar

A column-oriented database uses keys and dynamic groupings of columns to store data across distributed servers. To improve speed, similar data is grouped. Column-oriented databases are designed to have millions of columns and billions of rows.

Main characteristics:

- Keys include row, column family, and columns, so queries can be done on these keys.
- Multiple types of keys can be created, such as row or composite keys.
- Support for timestamping during inserts enables automatic versioning.
- Strong support of MapReduce for analytics for large amounts of data.
- Similar data is grouped to improve speed, especially for querying data by using aggregate functions such as count, sum, avg, max, and min.
- Compared to row-oriented databases, columnar databases can better compress data and save storage space, which permits aggregate functions to be performed rapidly.

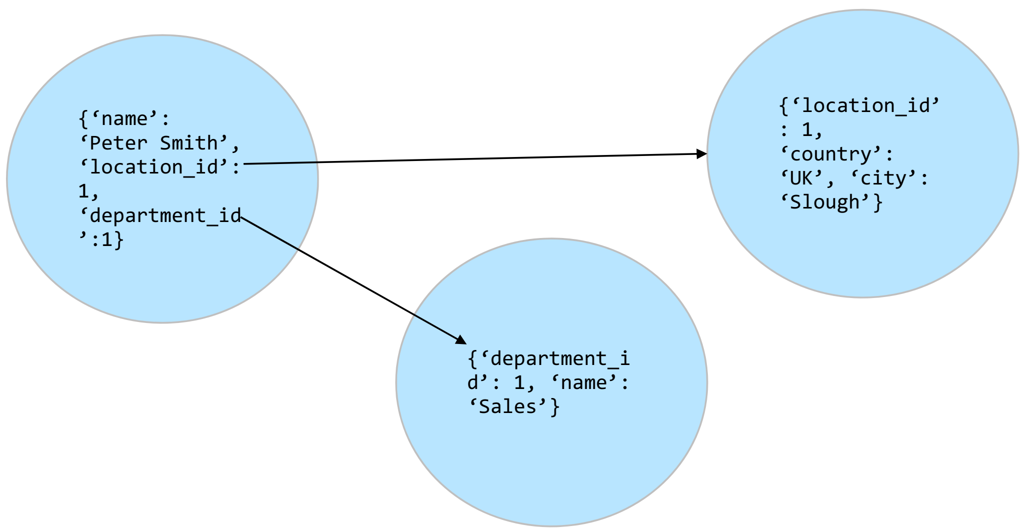
Here are some key use cases:

- Due to the column-base structure that is used to store the data in the columnar database, it performs better with aggregate function queries such as the queries that are used to get maximum, minimum and average values. Therefore, columnar databases are a good option for applications that rely on this kind of queries such as data-processing applications.
- Columnar databases can keep homogeneous data in a single block, so they can apply strategies to compress the data in a block and can optimize storage space which makes it a perfect option for big data. Big data is the large volume of data that might be analyzed computationally to extract information from.

IBM Training

Different types of NoSQL databases: Graph

- Graph-based databases can process complex queries more easily than the relational data model because they use an intuitive data model with a simple modeling of the nodes and the relationship as the edges.
- Queries generally use the *graph traversal process* or algorithms



Introduction to data services on IBM Cloud

Figure 6-12. Different types of NoSQL databases: Graph

Graph databases store information in entities, nodes and relationships, or edges. Graph databases are ideal when determining or representing data for multiple degrees of data and their distances.

The main difference between the graph model and the relational model is that the graph model has a more flexible structure. A graph model relies on nodes and relationships to describe the relations. In the relational model, the relations among the table can be described by using special properties, such as foreign keys.

Here are some key use cases:

- A powerful approach when the data is highly connected and related: A graph database can show relationships among various data items that are critical, and it can find inferences and rules among data.
- Social data analysis: For example, social networking sites can benefit by quickly locating friends, friends of friends, and likes.
- Filter mapping, pattern determination, and optimization problems in graph applications: For example, routing, spatial, and map applications can use graphs to model easily their data for finding close locations or building the shortest routes for directions.

6.2. Data services in IBM Cloud

IBM Training

Data services in IBM Cloud

Introduction to data services on IBM Cloud

Figure 6-13. Data services in IBM Cloud

IBM Training

Topics

- Introduction to databases
- Data services in IBM Cloud
- IBM Cloudant

Introduction to data services on IBM Cloud

Figure 6-14. Topics

IBM Training

Data services in the IBM Cloud catalog

- Databases services
- Storage services
- Analytics services

Introduction to data services on IBM Cloud

Figure 6-15. Data services in the IBM Cloud catalog

IBM Cloud offers various data services that are supported by IBM and third parties to use with cloud computing. Application developers, data scientists, and data engineers can use a combination of these services to complement their existing architecture and compose a business solution.

- **Database services**

  IBM Cloud offers a range of databases for various use cases. It provides relational databases for mission-critical transactional workloads, non-relational databases for flexible and extensible mobile and web apps, and data warehousing for blazing analytics.

- **Storage services**

  The IBM Cloud Storage family provides a flexible storage approach with rapid storage growth that is driven by new data sources and evolving technologies. It offers stand-alone or auxiliary storage for a server with either a storage solution that is built by using standard tools or the components to create your own storage solution.

- **Analytics services**

  With analytics services, you can make fast and accurate data-based decisions. They help you find new and unexpected insights quickly and deliver business-changing results. You can easily analyze the data and build machine learning models that can be deployed in cognitive applications.

References:

https://www.ibm.com/cloud/data

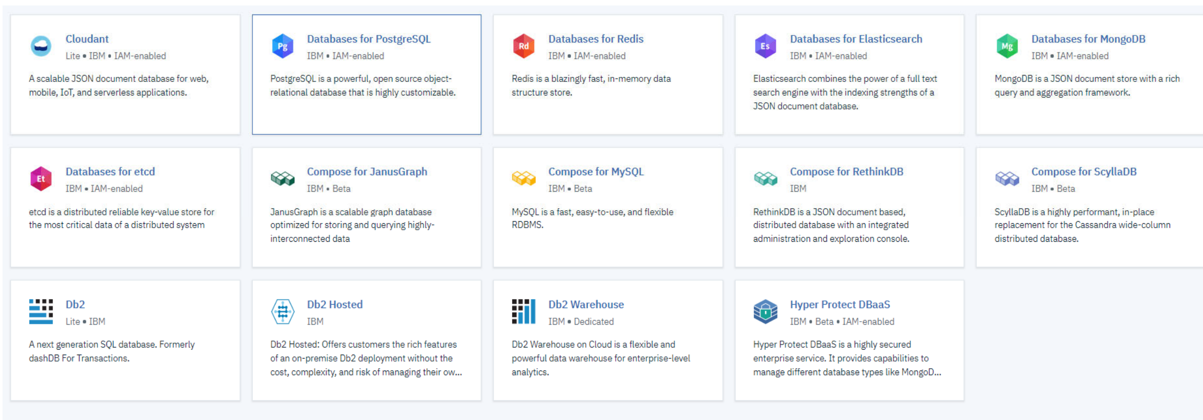
https://www.ibm.com/cloud/databases

https://www.ibm.com/cloud/storage

https://www.ibm.com/cloud/analytics

IBM Training

Databases services



Introduction to data services on IBM Cloud

Figure 6-16. Databases services

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

- Relational databases (SQL databases):
  - IBM Db2 Hosted

    Db2 is a Relational Database Management System (RDBMS). Db2 stores, analyzes, and retrieves data efficiently. With Db2 Hosted, you can run Db2 with full administrative access on a cloud infrastructure. It eliminates the cost, complexity, and risk of managing your own infrastructure.

  - Databases for PostgreSQL

    Postgres is a powerful, open source object-relational database that is highly customizable. It is a feature-rich enterprise database with JSON support that has the best of both the SQL and NoSQL worlds.

  - Compose for MySQL (Beta)

    MySQL is a fast, easy-to-use, and flexible RDBMS. As the central component of the Linux, Apache, MySQL, and PHP (LAMP) web service model, it has many connectors, including Python, PHP, and C++ for development needs.

- Key-value pair in-memory databases:
  - Databases for Redis

    Redis is an open source, in-memory data structure store that is used as a database, cache, and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperlogs, and geospatial indexes with radius queries.
  - Databases for etcd

    etcd is a key-value store that developers can use to coordinate and manage server clusters or provide lightning-fast metadata storage.
- Document databases:
  - Cloudant

    A highly scalable and performant document store that excels at pushing and retrieving data to and from the edge. It is a fully managed JSON document database that offers independent serverless scaling of provisioned throughput capacity and storage. Cloudant is compatible with Apache CouchDB and accessible through a simple to use HTTPS API for web, mobile, serverless, and IoT applications. It is explained in detail in the following slides.
  - Databases for MongoDB

    MongoDB is a JSON document store with a rich query and aggregation framework. Its features include high availability, automated backup orchestration, and de-coupled scaling of storage and RAM.
  - Databases for Elasticsearch

    Elasticsearch combines the power of a full text search engine with the indexing strengths of a JSON document database to create a powerful tool for rich data analysis on large volumes of data. Its features include high availability, automated backup orchestration, and de-coupled scaling of storage and RAM.
- Columnar databases:
  - Compose for ScyllaDB (Beta)

    ScyllaDB is a highly performant, in-place replacement for the Cassandra wide-column distributed database. ScyllaDB is written in C++ instead of Java like Cassandra's for better resource usage that can result in 10 times better performance in benchmarks.
- Graph databases:
  - Compose for JanusGraph (Beta)

    JanusGraph is scalable graph database that is optimized for storing and querying highly interconnected data that is modeled as millions or billions of vertices and edges.

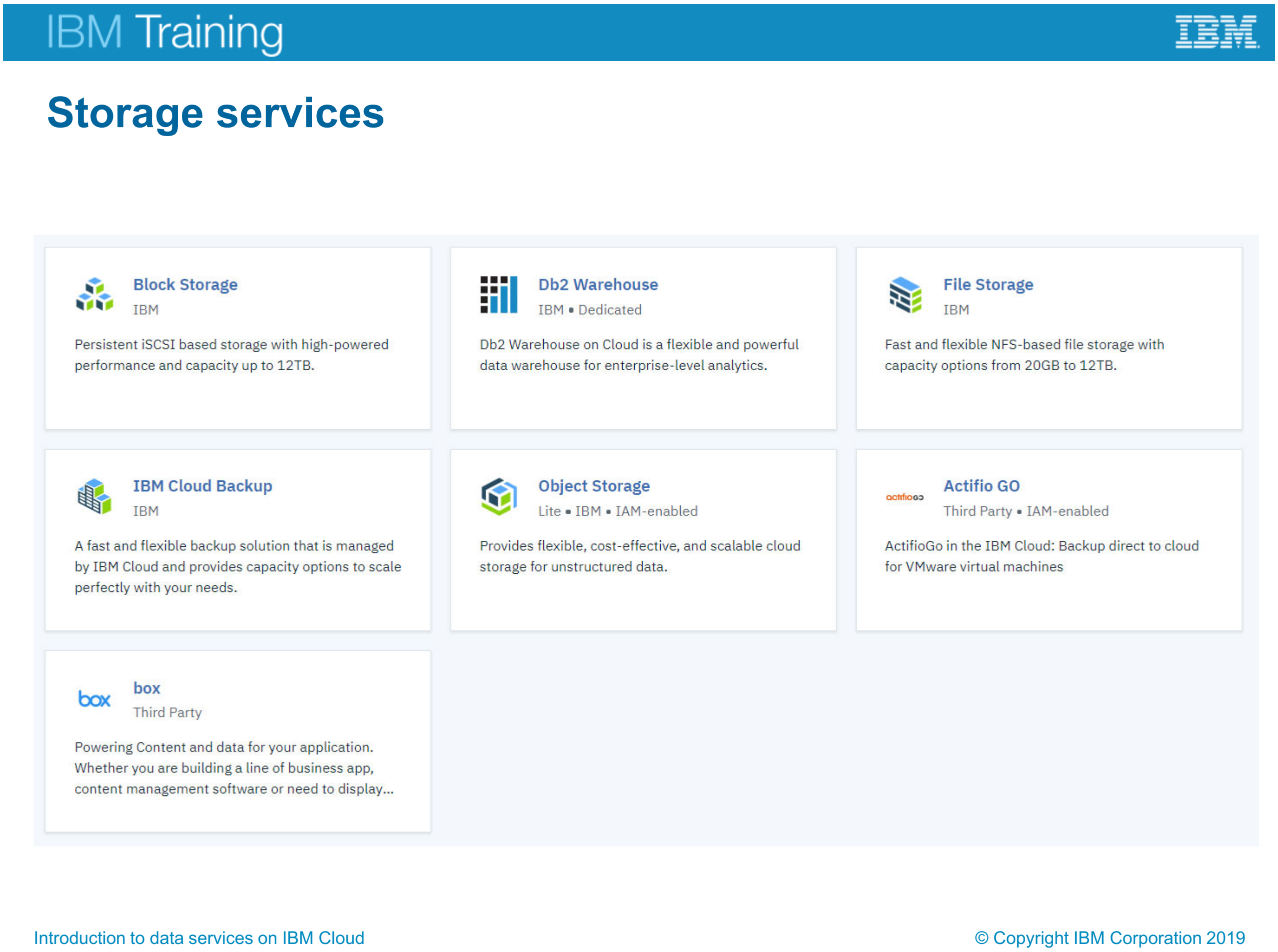


Figure 6-17. Storage services

IBM Cloud offers a flexible storage approach with the following robust and durable object, block, and file storage services:

- Cloud Object Storage

  Cloud Object Storage is a highly scalable cloud storage service for high durability, resiliency, and security. By using a self-service portal and RESTful APIs, a user can store, manage, and access the data.

- IBM Cloud Block Storage

  Provides persistent iSCSI-based storage with high-powered performance and capacity up to 12 TB. Flexible customization, durability, snapshot and replication, volume duplication, and high availability with concurrent access are the main key features of Cloud Block Storage.

- IBM Cloud File Storage

  A fast and flexible NFS-based file storage with capacity options of 20 GB – 12 TB. It has the same key features as block storage, such as flexible customization, durability, snapshot and replication, volume duplication and high availability with concurrent access.

- IBM Cloud Backup

  A fast and flexible backup solution that is managed by IBM Cloud and provides capacity options to scale perfectly with user needs.

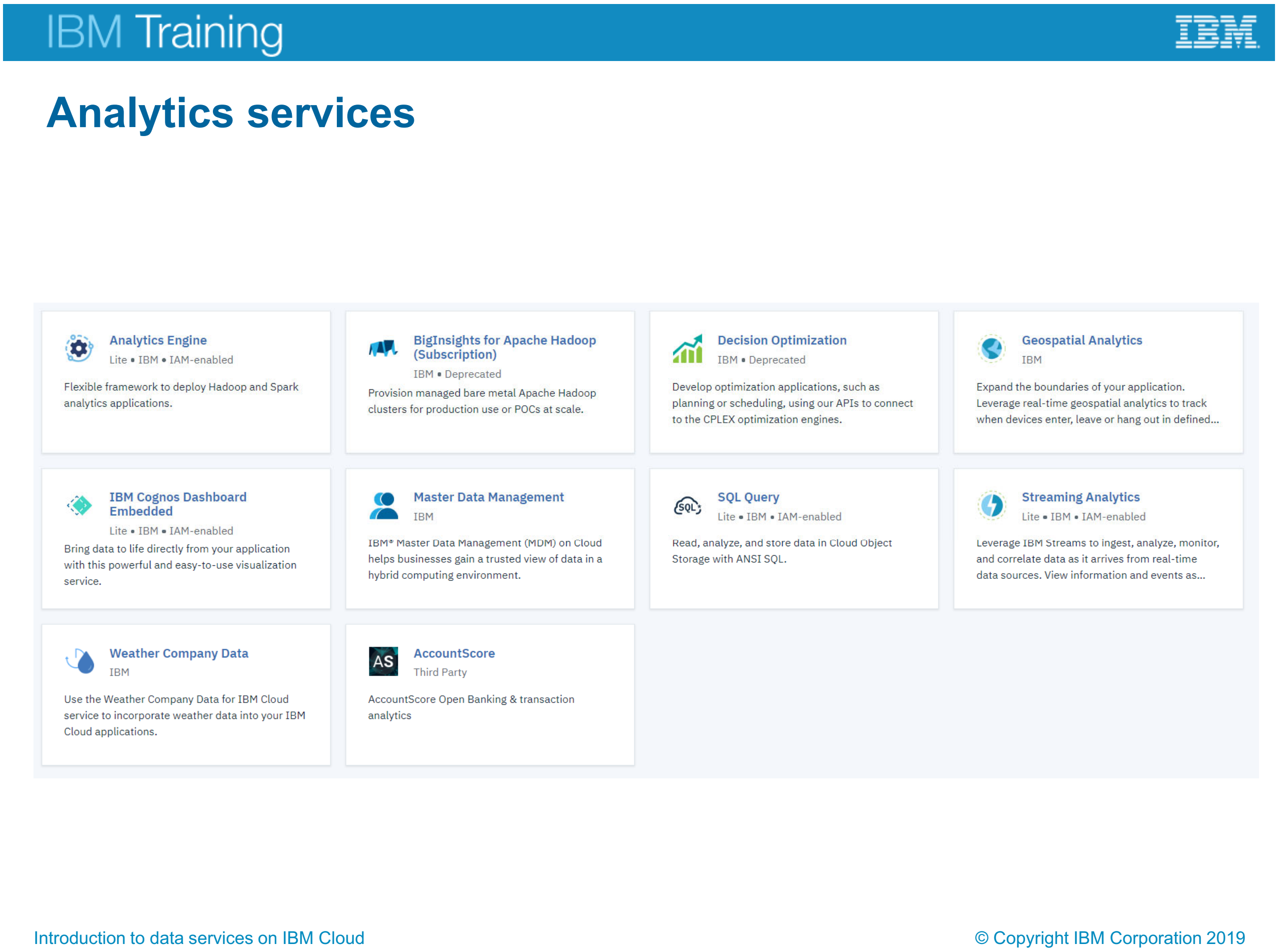


Figure 6-18. Analytics services

IBM Cloud offers the following analytics services to analyze easily the data and build machine learning models that can be deployed in AI applications:

- Analytics Engine

  It allows the development and the deployment of analytics applications by using open source Apache Spark and Apache Hadoop. The Analytics Engine key feature is its ability to customize the cluster by using your own analytics libraries and open source packages and the possibility to integrate with IBM Watson Studio or third-party applications to submit jobs to the cluster.

- Streaming Analytics

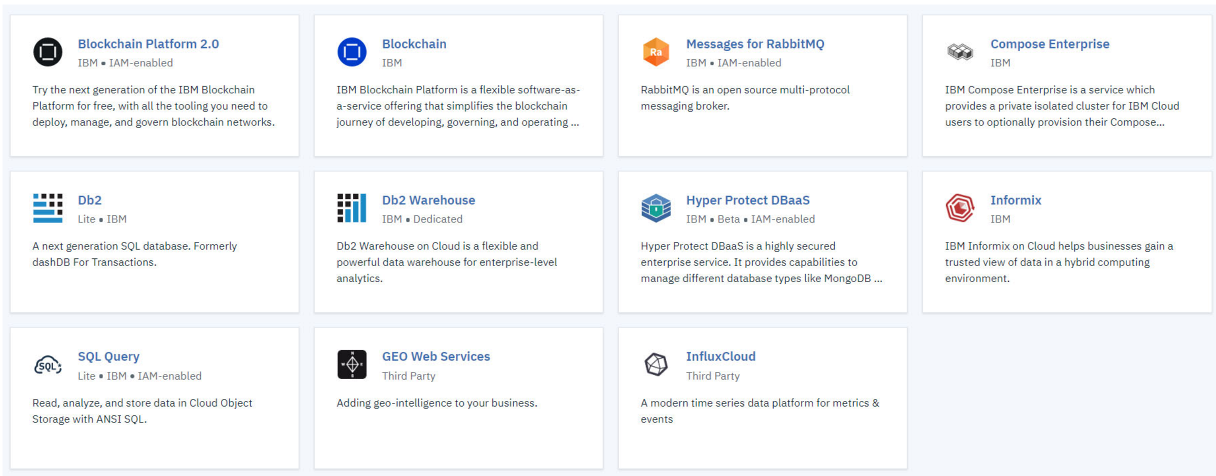
  With IBM Streams, you can ingest, analyze, monitor, and correlate data as it arrives from real-time data sources. You use it to view information and events as they unfold.

- IBM Geospatial Analytics

  IBM Geospatial Analytics provides real-time geospatial analytics to track when devices enter, leave, or hang out in defined regions. Powered by IBM Streaming Analytics on IBM Cloud.

IBM Training

Other related data services



Introduction to data services on IBM Cloud

Figure 6-19. Other related data services

IBM Cloud offers many other related data services, and here are some of them:

- IBM Blockchain Platform

  IBM Blockchain Platform is a flexible software as a service (SaaS) offering that is delivered by IBM Cloud. It enables network members to get started quickly developing and move easily to a collaborative environment. The platform simplifies your blockchain journey of developing, governing, and operating a network.

- Messages for RabbitMQ

  RabbitMQ is an open source multi-protocol messaging broker. It routes messages between microservices for modern applications.

- IBM Db2 Warehouse

  IBM Db2 Warehouse on Cloud is a fully managed and enterprise-class cloud data warehouse service. Powered by IBM BLU Acceleration, Db2 Warehouse on Cloud provides unmatched query performance.

6.3. IBM Cloudant

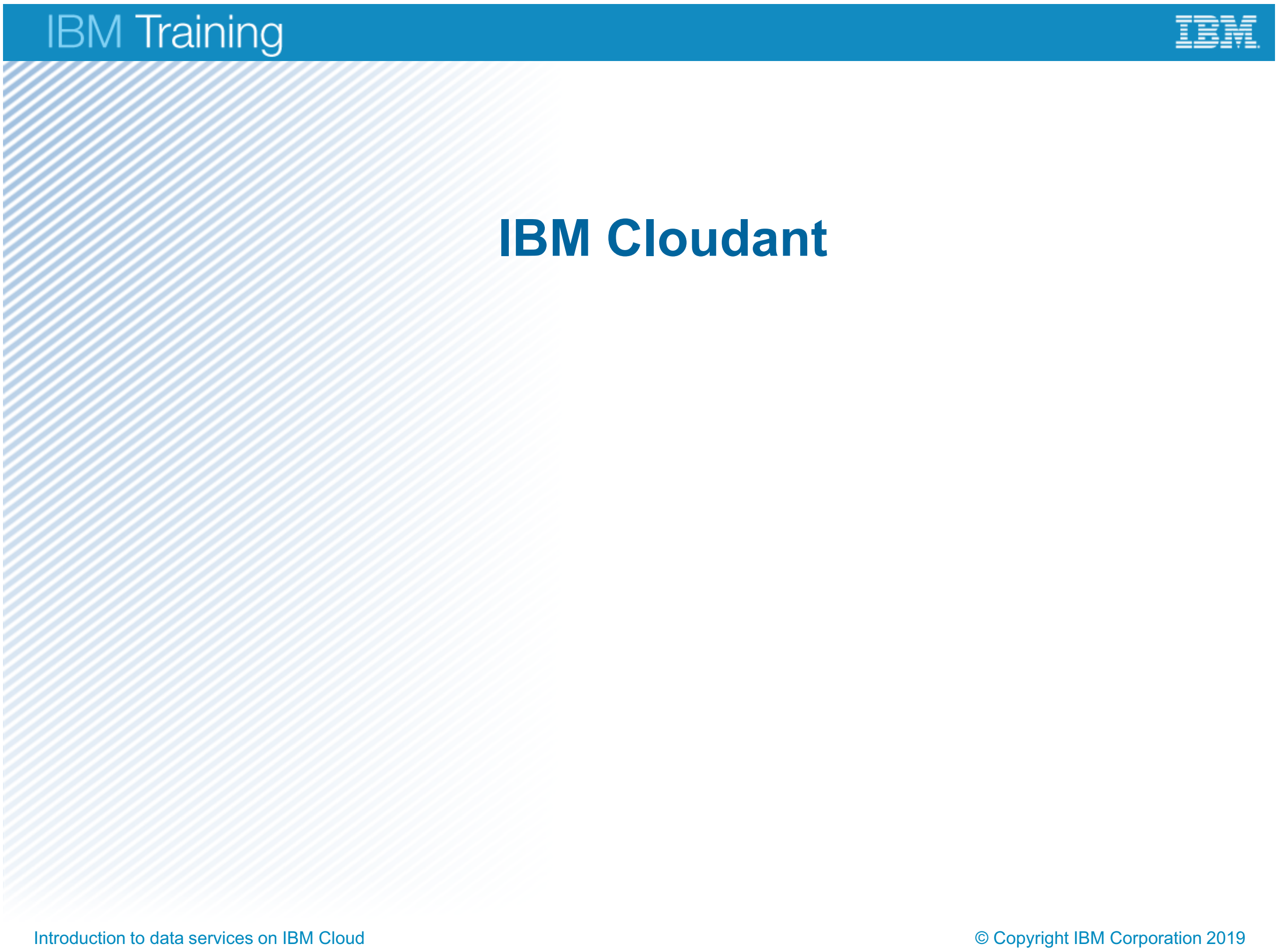


Figure 6-20. IBM Cloudant

IBM Training

Topics

- Introduction to databases
- Data services in IBM Cloud
- IBM Cloudant

Introduction to data services on IBM Cloud

Figure 6-21. Topics

IBM Training

Cloudant capabilities

- Database as a service (DBaaS): 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.
- IBM Cloudant Geo.
- Offline First mobile web apps capabilities.
- Synchronization feature for disconnected Android and Apple apps.
- Client libraries for developing your own application.

Introduction to data services on IBM Cloud

Figure 6-22. Cloudant capabilities

IBM acquired Cloudant, a Boston-based cloud database startup, 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 ensures 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 that are used in the applications:

- The Cloudant search function is powered by Apache Lucene, which is the most popular open source search library that indexes and searches 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 a GeoJSON storage with built-in spatial querying and map visualization that is powered by Mapbox. GeoJSON is a format that is 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.

Reference:

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

IBM Training

Benefits of IBM Cloudant

- High scalability
- High availability, including data replication worldwide
- Satisfied by eventually consistent results (”stale” reads are better than no reads)
- High performance at large (> 1 TB) scale
- ACID transactions at the document level
- Fully managed database as a service
- Powerful serverless API

Introduction to data services on IBM Cloud

Figure 6-23. Benefits of IBM Cloudant

IBM Cloudant is available in all IBM Cloud regions and 55+ data centers around the world. Cloudant can easily support disaster recovery between continents. In addition, Cloudant can scale an app for a global release through a horizontal scaling architecture that can handle millions of users and terabytes of data to grow seamlessly alongside the business. All Cloudant instances are deployed on clusters that span availability zones in regions that support them for added durability at no extra cost.

IBM Cloudant provides a fully managed and distributed JSON document database. Instantly deploy an instance, create databases, and independently scale throughput capacity and data storage to meet the application requirements.

Enhance applications with built-in key value, MapReduce, full-text search, and geospatial querying that goes beyond simple bounding boxes. Stream the changes feed for seamless integration with event-driven applications and IBM Cloud Functions.

IBM Training

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.

Introduction to data services on IBM Cloud

Figure 6-24. Documents in Cloudant

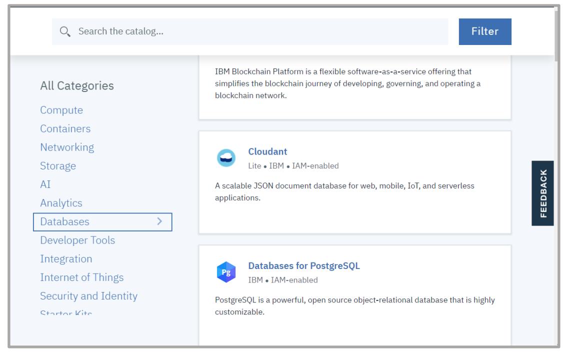
Cloudant documents are containers for data, and the documents are JSON objects. All documents in Cloudant must contain the following unique fields:

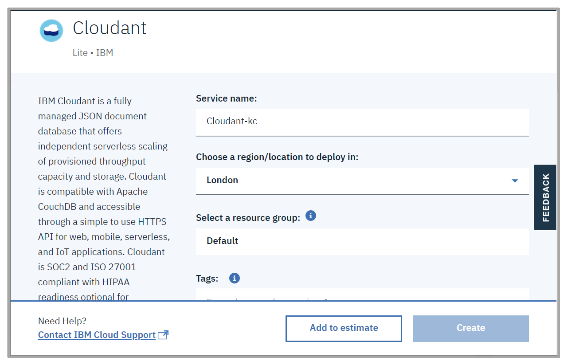
- An identifier _id field serves as the document key. It can be created by the application or generated automatically by Cloudant.
- A revision number _rev field is automatically generated and used internally by the Cloudant database as a revision number. A revision number is added to your documents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

IBM Training

Getting started with Cloudant in IBM Cloud

1. From the IBM Cloud Dashboard, click **Create resource**.
2. In the Catalog page, select **Databases** under categories and then select **Cloudant**.
3. Enter a descriptive name in the **Service name** field.
4. Select a **region**, **resource group**, and **authentication method**.
5. Select a **pricing plan** that fits your needs. You can always start with the free plan and upgrade later through the Cloudant Dashboard.
6. Click **Create**.





Introduction to data services on IBM Cloud

Figure 6-25. Getting started with Cloudant in IBM Cloud

A resource group is a way to organize the user account resources in customizable grouping for access control and billing purposes. For more information, see the following resources:

https://cloud.ibm.com/docs/resources?topic=resources-rgs

https://cloud.ibm.com/docs/resources?topic=resources-bp_resourcegroups#bp_resourcegroups

The Cloudant service has two authentication methods that are available:

- **Use both legacy credentials and IAM**: Both *IBM Cloud Identity and Access Management (IAM)* and legacy credentials can be used to access the account.
- **Use only IAM**: Only IAM credentials are provided through the “Service binding and credential generation” menu.

Legacy credentials enable login to Cloudant by using HTTP Basic authentication and providing a user name and password that is used for authentication.

IAM provides a unified approach to managing user identities, services, and access control. IAM authentication requires the exchange of an IAM API key for a time-limited access token before you can make a request to Cloudant. When the access token expires, the client must get a new one from the IAM token service.

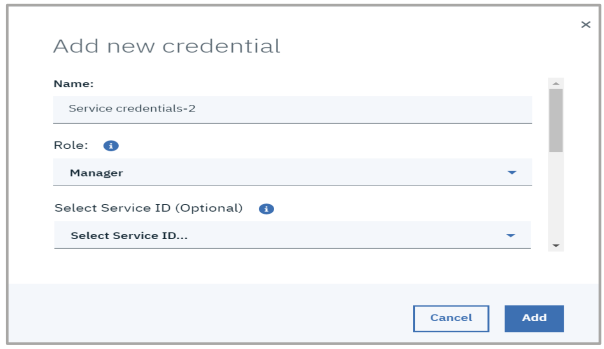
References

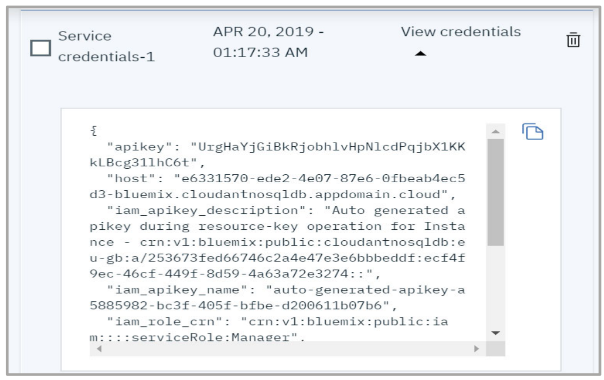
https://cloud.ibm.com/docs/services/Cloudant/guides?topic=cloudant-ibm-cloud-identity-and-access-management-iam-#enabling-iam-with-ibm-cloudant

IBM Training

Cloudant with IBM Cloud: Creating credentials

1. From the resource list, select the Cloudant service instance to open it.
2. Select the **Credentials** tab, and click **New Credential +**.
3. From the Add New Credential dialog, provide a **Name**, **Role**, **Service ID (optional), and Inline Configuration Parameters (Optional)**.
4. Click **Add** to generate the new service credential.





Introduction to data services on IBM Cloud

Figure 6-26. Cloudant with IBM Cloud: Creating credentials

Credentials are created for cases where you want to connect manually an app or external consumer to a Cloudant service on IBM Cloud. A credential might contain a user name, password, host name, port, and a URL.

Because the Cloudant service is managed by IAM, it should include an IAM Service access role.

Role defines the permitted action when accessing the service. This field currently allows the Manager role only.

Service ID identifies a service or application similar to how a user ID identifies a user. There are two options for creating service ID:

- Auto-generating one.
- Creating a customized one.

Add Inline Configuration Parameters (Optional): This optional field provides service-specific configuration parameters in a valid JSON object.

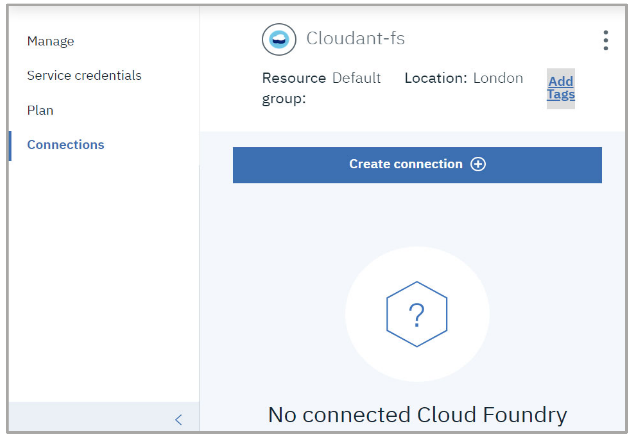
Reference:

https://cloud.ibm.com/docs/resources?topic=resources-service_credentials

IBM Training

Cloudant with IBM Cloud: Creating connections

1. From the resource list, select the Cloudant service instance to open it.
2. Select the **Connections** tab, and click **Create connection**.
3. Click **Connect** for the row of the app for which you want to create the connection.
4. Select an access **Role**, **Service ID** and optionally add configuration parameters in JSON format.
5. Click **Connect & restage app**.



Introduction to data services on IBM Cloud

Figure 6-27. Cloudant with IBM Cloud: Creating connections

To use a Cloudant service instance with a Cloud Foundry application, you must create a connection or bind that service to the app.

While creating the connection with the application, a Cloud Foundry service is generated as an alias of the original Cloudant service to allow the connection with the application. After restaging the application, you can find the service credentials in the application side that contains all the required credentials, such as the URL, host, port number, and API key that belong to this service with this application. These services credentials are considered the environment variables of the Cloudant database (**VCAP_SERVICES**).

To connect, you can customize the **ServiceID** and access role that is used for this binding. Restaging the application is required to connect with the Cloudant service, which might result in application downtime.

Role defines the permitted action when accessing the service. This field currently allows the Manager role only.

Service ID identifies a service or application similar to how a user ID identifies a user. There are two options for creating service ID:

- Auto-generating one.
- Creating a customized one.

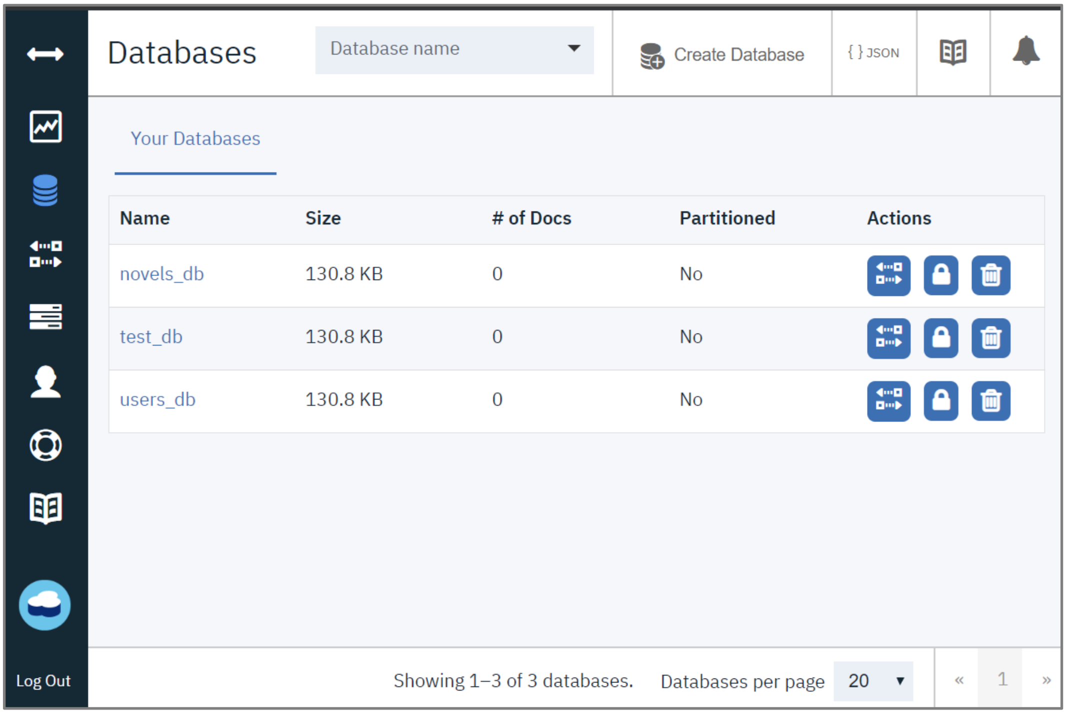
Add Inline Configuration Parameters (Optional): This field is optionally used to provide service-specific configuration parameters in a valid JSON object.

Reference:

https://cloud.ibm.com/docs/resources?topic=resources-s2s_binding

IBM Training

Cloudant Dashboard



Introduction to data services on IBM Cloud

Figure 6-28. Cloudant Dashboard

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.
- View capacity usage (reads/second, write/second, storage limit, and so on).
- Manage the plan settings (upgrade plan, raise throughput capacity, and so on).

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.

IBM Training

Cloudant HTTP 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`.

Introduction to data services on IBM Cloud

Figure 6-29. Cloudant HTTP API

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

The following HTTP Request methods can be used to apply the create, read, update, and delete operations on Cloudant documents by directly referencing the document ID:

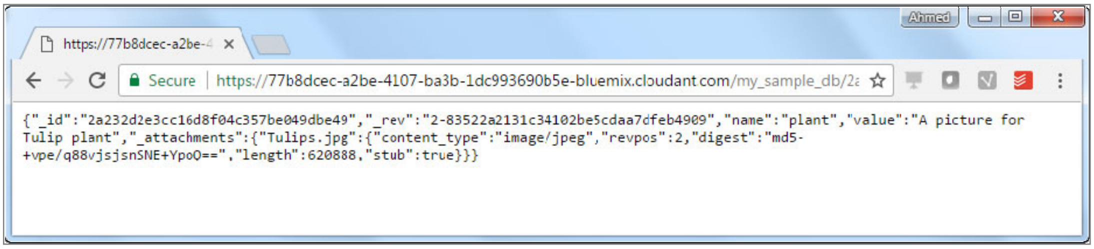
- **GET**: Request a specific JSON document.
- **POST**: Set values, and create documents.
- **PUT**: Create databases and documents.
- **DELETE**: Delete a specific document.

IBM Training

Reading a document in Cloudant

To access a document with the Cloudant API, issue a **GET** request to the following URL:

https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID



Introduction to data services on IBM Cloud

Figure 6-30. Reading a document in Cloudant

This figure in this slide shows accessing a document from the browser. From your application, you can access the document through HTTP APIs.

To access a document with the Cloudant HTTP 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 by using a **GET** HTTP API 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.

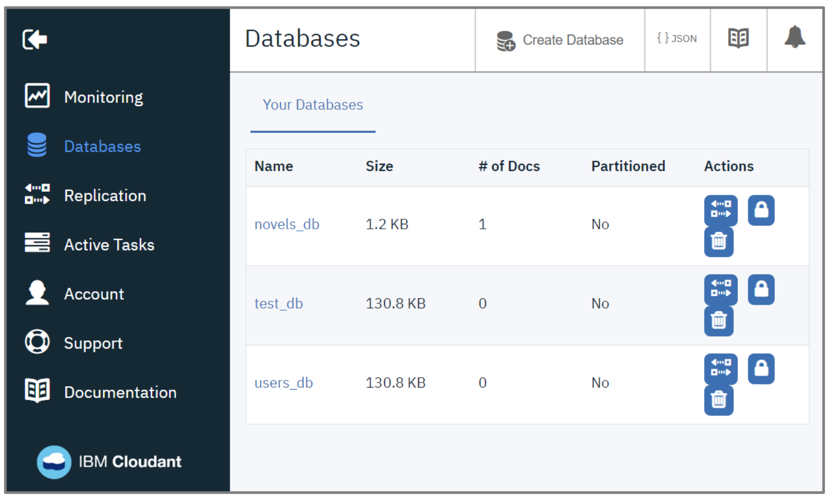
The sample document in the Cloudant Dashboard that is shown in the slide includes three fields (name, value, and _attachments) in addition to the mandatory fields _id and _rev.

IBM Training

View all documents

To view all documents at a database, issue a **GET** request to the following URL:

`https://$USERNAME.cloudant.com/$DATABASE/_all_docs?include_docs=true`



Introduction to data services on IBM Cloud

Figure 6-31. View all documents

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.

As shown in the slide, 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.

IBM Training

More Cloudant HTTP APIs

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

Introduction to data services on IBM Cloud

Figure 6-32. More Cloudant HTTP APIs

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

To update (or create) a document, you can send a PUT request to 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, you can send a DELETE request to 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.

IBM Training

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 with a selector in the body.
- A *selector* is a JSON object describing the criteria that is used to select documents.
- Example of a Cloudant query body:

```
{
  "selector": {
    "lastname": "Brown",
    "location": "New York City, NY"
  },
  "fields": [
    "firstname",
    "lastname",
    "location"
  ]
}
```

Introduction to data services on IBM Cloud

Figure 6-33. Cloudant Query

Before you query for a specific field, it is a best practice to create an index for each field in the selector to optimize query performance.

The JSON body that is provided shows an example of a Cloudant query request body. In this example, the response of this request returns Cloudant documents that have **lastname** = '**Brown**' and **location** = '**New York City, NY**'. The document fields that are shown are only **firstname**, **lastname, and location**. Some advanced operators can be used in the Cloudant query, such as the $eq (equal) and $gt (greater than) operators that are used to search for documents.

IBM Training

Cloudant indexes

- Indexes enable quick access to a portion of the data.
- To create an index, issue a `POST` request to https://$USERNAME.cloudant.com/$DATABASE/_index with a body that contains index field names, an index name, and an index type.
- Example of creating an index body:

```
{
      "index": {
          "fields": ["foo"]
      },
      "name": "Movie_name-text",
      "type": "text"
  }
```

Introduction to data services on IBM Cloud

Figure 6-34. Cloudant indexes

A database index is a sorted data structure that enables quick access to a portion of the data. By default, IBM Cloudant generates a primary index for the _id field so that it can retrieve data by _id.

A user can create secondary indexes for other fields if there are many queries that run on these fields.

After you create an index, a design document is generated on Cloudant to describe the index that is created. Design documents are used to build indexes, validate updates, and format query results.

The *index type* is either text or JSON. Text indexes are powered by Cloudant search indexes, which enable you to query a database by using Lucene Query Parser. JSON indexes are powered by MapReduce.

You can use Cloudant endpoints to create, list, update, and delete indexes in a database, and to query data by using these indexes.

The JSON body that is provided shows an example of a create index request body. In this example, an index of type text is created for a field that is called “foo”. After the creation of this index, the Cloudant query that is used to search for Cloudant documents by using the “foo” field in the query are more efficient and faster.

References:

https://cloud.ibm.com/docs/services/Cloudant/api?topic=cloudant-query

https://cloud.ibm.com/docs/services/Cloudant/api?topic=cloudant-design-documents

https://developer.ibm.com/clouddataservices/docs/cloudant/indexes/

IBM Training

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"
}
```

Introduction to data services on IBM Cloud

Figure 6-35. HTTP status codes

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 other information about the error is returned in the response as JSON, as shown in the slide.

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

IBM Training

Unit summary

- Describe different databases types and capabilities
- Describe the main types of data services in IBM Cloud.
- Explain the benefits of IBM Cloudant.
- Access Cloudant databases and documents on IBM Cloud.
- Use HTTP APIs to interact with Cloudant database.

Introduction to data services on IBM Cloud

Figure 6-36. Unit summary

IBM Training

Review questions

1. _____ is one of the document type databases that are available in IBM Cloud.
   A. Databases for etcd
   B. Db2 Hosted
   C. Databases for MongoDB
   D. ScyllaDB

2. **True or False**. You should have an account on cloudant.com before creating a Cloudant service on IBM Cloud.

3. **True or False**. Cloudant databases and documents are accessed by using HTTP APIs.

4. **True or False**. You can issue PUT or `POST` HTTP requests to create a document.

Introduction to data services on IBM Cloud

Figure 6-37. Review questions

IBM Training

Review questions (cont.)

5. _____ is automatically generated in a Cloudant document and used internally by the Cloudant database as a revision number.
   A. _id
   B. _rev
   C. _index
   D. _revision_number

Introduction to data services on IBM Cloud

Figure 6-38. Review questions (cont.)

IBM Training

Checkpoint answers

1. **C**.
2. **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.
3. **True**. Cloudant databases and documents are accessed by using HTTP APIs.
4. **True**. You can issue `PUT` or `POST` HTTP requests to create a document.
5. **B**.

Introduction to data services on IBM Cloud

Figure 6-39. Checkpoint answers

IBM Training

Exercise 3: IBM Cloud with Cloudant

Introduction to data services on IBM Cloud

Figure 6-40. Exercise 3: IBM Cloud with Cloudant

IBM Training

Exercise objectives

- This exercise demonstrates how you can create a Cloudant database service on IBM Cloud.
- After completing this exercise, you should be able to perform the following tasks:
  - Create an instance of the Cloudant service on IBM Cloud.
  - Create service credentials by using IBM Cloud Identity and Access Management (IAM).
  - Access the Cloudant documentation.
  - Explore the features of the Cloudant Dashboard.
  - Create, read, update, and delete Cloudant documents by using HTTP APIs.
  - Verify the data that is stored in the database from the Cloudant Dashboard.
  - Create indexes and query Cloudant documents by using HTTP APIs.

Introduction to data services on IBM Cloud

Figure 6-41. Exercise objectives

IBM Training

Documentation and other information sources

- Cloudant documentation:
  https://cloud.ibm.com/docs/services/Cloudant?topic=cloudant-getting-started
- IBM Cloud Data Services documentation:
  https://developer.ibm.com/clouddataservices/
- IBM Cloudant videos on YouTube:
  https://www.youtube.com/channel/UCSMx6Fgq1RJLq58em2mJeKQ
- IBM Developer 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/category/databases/

Introduction to data services on IBM Cloud

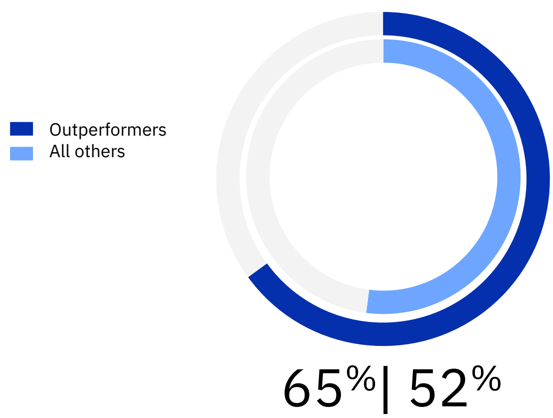
Figure 6-42. Documentation and other information sources

IBM Training

Importance of data (backup slide)

Leading organizations excel at capitalizing on data.

Capture, manage, and derive insights from key corporate data



• Source: 2018 AI Survey: A8.5: For the following statement, indicate the level of rigor with which your organization performs the following activities - Capture, manage, and access business, technology, and operational information on key corporate data, N= 4992

Introduction to data services on IBM Cloud

Figure 6-43. Importance of data (backup slide)

There is a popular saying that data is the new oil because the data and the information that is obtained by processing the data play an important role in modern organizations and contribute to the development of new business models. The organizations that are considered the most successful ones are those that can capture, manage, and derive key insights from their corporate data. Cloud technologies enable small organizations to design, set up data platforms, and use data analysis services on the cloud quickly and receive benefits from the scalability, reliability, and quality of service that is provided by the cloud. These factors help these organizations to evolve quickly and grow up faster in the market.

IBM Training

Atomicity – Consistency – Isolation – Durability (backup slide)

- **Atomicity:** Either all tasks in a transaction are performed or none of them are. If one element of a transaction fails, the entire transaction fails.
- **Consistency:** The transaction does not violate any protocols or rules that are defined in the system and the database must remain in a consistent state at the beginning and end of a transaction; there are never any half-completed transactions.
- **Isolation:** No transaction has access to any other transaction that is in an intermediate or unfinished state. This is required for both performance and consistency of transactions within a database.
- **Durability:** After the transaction is complete, it persists as complete and cannot be undone; it survives system failure, power loss, and other types of system breakdowns.

Introduction to data services on IBM Cloud

Figure 6-44. Atomicity – Consistency – Isolation – Durability (backup slide)

IBM Training

Basically Available – Soft State – Eventual consistency (backup slide)

- **Basically Available:** There is a response to any request, *but* that response might still “fail” to obtain the requested data or the data might be in an inconsistent or changing state, much like waiting for a check to clear in your bank account.
- **Soft state:** The state of the system can change over time, so even during times without input there might be changes going on due to *eventual consistency*. Thus, the state of the system is always *soft*.
- **Eventual consistency:** The system *eventually* becomes consistent after it stops receiving input. The data propagates to everywhere it should sooner or later, but the system continues to receive input and is not checking the consistency of every transaction before it moves onto the next one.

Introduction to data services on IBM Cloud

Figure 6-45. Basically Available – Soft State – Eventual consistency (backup slide)

IBM Training

Features of NoSQL databases (backup slide)

- Key characteristics of NoSQL technologies:
  - Highly scalable.
  - Flexible data schema.
- Why is it more flexible and scalable?
  - It does not require a predefined data model for storage, such as specific row and column names and sizes.
  - It is optimized to work on distributed hardware.
  - It uses relatively simple queries that can be processed quickly across much larger data sets.
- Therefore, these databases are well-suited to applications that are characterized by:
  - Large amounts of data.
  - Low latency requirements.
  - Non-relational data.
  - Unstructured data.
  - Simple data queries (does not require multistep transactions).
- Follow the BASE design properties.

Introduction to data services on IBM Cloud

Figure 6-46. Features of NoSQL databases (backup slide)

IBM Training

Cloudant Index and other types of queries (backup slide)

CRUD – Document	Primary Index	Secondary Index (view)	Search Index	Geospatial Index	Cloudant Query
• Direct document lookup by _id	• Exists "OOTB" • Stored in a b-tree • Primary key > doc._id	• Built by using MapReduce • Stored in a b-tree • Key > user-defined fields	• Built by using Lucene • FTI: Any or all fields can be indexed	• Stored in R* tree • Lat/Long coordinates in GeoJSON	• "Mongo-style" querying • Built natively in erlang
• Use when you want a single document and can find by its _id	• Use when you can find documents based on their _id • Pull back a range of keys	• Use when you need to analyze data or get a range of keys • Examples: count data fields, sum/average numeric results, advanced stats, group by date, and so on.	• Ad hoc queries • Find documents based on their contents • Can do groups, facets, and basic geo queries (bbox and sort by distance)	• Complex geometries (polygon, circularstring, and so on) • Advanced relations (intersect, overlaps, and so on)	• Ad hoc queries • Many operators (>, <, IN, OR, AND, and so on) • Intuitive for people who come from Mongo or SQL backgrounds

Introduction to data services on IBM Cloud

Figure 6-47. Cloudant Index and other types of queries (backup slide)

You can perform basic create, read, update, and delete operations on documents by directly referencing the document ID.

Additionally, Cloudant creates the Primary Index for every database by using standard features and stores it in a b-tree data structure. Cloudant uses the document ID as the primary key.

The search index is built by using Lucene search.

The geospatial index is stored in the most efficient way to allow for 4D querying.

Cloudant Query uses "Mongo-style" querying and provides you with a declarative way to define and query indexes on your database.

If your query involves more than lookups by the primary key, then you must build an index. Secondary indexes are built by using the MapReduce paradigm.

So, if you need to find a document by its _id, use the direct document lookup.

To get a list of documents by their _ids, use the primary index.

For analytics such as counts, sums, averages, or other mathematical functions, use a secondary index, also called a view, which is built by using MapReduce.

For *ad hoc* queries on one or more fields, searches involving large blocks of text, or queries that require more Lucene syntax like wildcards, fuzzy search, and facets, use a search index.

If your application requires advanced geospatial queries beyond a bounding box or 4D, use a geospatial index.

Lastly, if you prefer Mongo-style querying syntax for searching containing multiple logical operators, use Cloudant Query.

Search Index, Geospatial Index, and Cloudant Query are features that are unique to Cloudant and not currently available with Apache CouchDB.

IBM Training

Cloudant best practices (backup slide)

- Performance considerations.
- Large attachments.
- Replication with `_replicate` endpoint does not persist.
- Use show and list functions sparingly.
- Data design considerations:
  - Eventually Consistent system.
  - Doc updates and immutable data.
  - Balancing the following needs:
    - Denormalizing data to minimize HTTP requests
    - Using fine-grained documents to avoid conflicts
  - Migrating from relational or SQL to Cloudant.
  - Simulating transactions and ACID compliance.
  - MVCC is not a version control system.
  - Organizing docs into databases.

Introduction to data services on IBM Cloud

Figure 6-48. Cloudant best practices (backup slide)

Unit 7. Enriching your applications with IBM Cloud services

Estimated time

01:00

Overview

This unit presents an example of a cloud application to solve a business problem. It introduces functional and non-functional requirements, application architecture, and services on IBM Cloud that you can integrate with your application logic to implement a solution.

IBM Training

Unit objectives

- Define your business problem and goals
- Identify functional and non-functional requirements
- Select the technical components that best fit your solution
- Design a simple architecture for a cloud application
- Identify services in the IBM Cloud catalog that you can use to enrich your cloud apps
- Describe App ID, Watson Natural Language Understanding, Watson Tone Analyzer, LogDNA, and IBM Cloud Monitoring services and their integration in the sample use case

Enriching your applications with IBM Cloud services

Figure 7-1. Unit objectives

7.1. Business problem and requirements

IBM Training

Business problem and requirements

Enriching your applications with IBM Cloud services

Figure 7-2. Business problem and requirements

IBM Training

Topics

- **Business problem and requirements**
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

Enriching your applications with IBM Cloud services

Figure 7-3. Topics

IBM Training

Define your business problem and goals

- Define your problem domain and establish common goals
  - You cannot solve a problem without first understanding it.
- Define your business objectives
  - Define the “who”, “what”, and “why”.
  - A business objective must center on a measurable outcome, or the "why.“
  - State the benefits that the stakeholders expect from the system.
- Identify potential problems and bottlenecks
  - Surface the big issues that might negatively impact or prevent the desire outcomes.

Enriching your applications with IBM Cloud services

Figure 7-4. Define your business problem and goals

Before you start designing the solution, you must understand the problem. In the *discovery* phase you must:

- Dig deep into your problem domain.
- Establish common goals.
- Identify potential problems and bottleneck.
- Defining your problem domain and establishing common goals:

  You cannot solve a problem without first understanding it. Your team must dig deep into your problem domain and align everyone on common goal.

- Defining your business objectives

  The objectives frame a problem in terms of a user outcome, but they do not describe or prescribe any type of implementation. Identify who will benefit from your system (for example, users, departments in the organization) and how will they be rewarded (for example, new functions, savings, new business opportunities) and describe the end state that you want to achieve. The user and the end state are the "who" and the "what.“ Finally, include a statement of differentiation. The differentiation is either a qualitative or quantitative measure and represents “why” you are embarking in this journey.

  A business objective must center on a measurable outcome, or the "why."

  While a business objective does not need to specify or be oriented toward a specific user or community of interest, the business objective must have a sponsor.

- Identify potential problems and bottlenecks

  Discovering serious issues late might lead to wasting a lot of resources and to the failure of the project. Identifying problems and bottlenecks early helps the entire organization see the biggest issues and how they negatively affect getting value to customers. The bigger bottlenecks have impact on downstream processes and tasks. If the bigger bottleneck is understood by everyone in the organization, it is clear that the organization must focus on reducing that delay or issue to the benefit of everyone, especially customers.

You must ensure that you define clear objectives to guide what is being delivered, when, and by whom. You must also ensure that you have a clear understanding of the total cost of ownership for your cloud platform.

Reference:

https://www.ibm.com/cloud/garage/content/course/explore-garage-method-for-cloud/9

IBM Training

Cognitive Tweets Analyzer: Business problem and objectives

- Business problem
  - A movie production company wants to know how the audience reacts to an episode of their popular movie series. They also want to identify the type of scenes that cause the most excitement and positive reaction from the audience.
  - Business objectives
    - Who: Screenplay writers, movie trailer editors
    - What: Guidelines for screenplays and movie trailers
    - Why: Attract larger audiences and new advertisers, sell future episodes
  - Potential problems and bottlenecks
    - Cannot identify duplicate tweets that are retrieved with tweets replies or retweets.
    - Users post emoticons, GIFs, or videos to express their opinions and emotions, which cannot be analyzed

Enriching your applications with IBM Cloud services

Figure 7-5. Cognitive Tweets Analyzer: Business problem and objectives

This unit presents the Cognitive Tweets Analyzer app as an example. This use case is based on real life cloud applications that are developed for the movie industry. However, this use case is an oversimplification to keep the scope of the problem and solution easy to describe in this course.

Notice that the business problem and objectives are not described in technical terms. As a solution architect and also an app developer you are required to translate the client’s description of problems and objectives from business to technical terms.

Business problem

This use case presents a movie production company that wants to know how the audience reacts to an episode of their popular movie series. They also want to identify the type of scenes that cause the most excitement and positive reaction from the audience.

Business objectives

The objective is to provide guidelines to screenplay writers and movie trailer editors for their screenplays and the scenes to be included in the movie trailers. The expectation is that these guidelines will improve the quality of the screenplay and movie trailers and therefore, attract larger audiences and new advertisers and sell future episodes more effectively.

Potential problems and bottlenecks

One potential problem is identifying duplicate tweets that can be retrieved with tweets replies or retweets. Another problem is when viewers expressed their emotions and opinions by using emoticons, GIFs, or videos, which cannot be analyzed by the proposed solution. In this case, Watson Visual Recognition might have to be added but it is outside the scope for this use case.

References:

https://www.ibm.com/blogs/watson/2017/04/hollywood-taps-build-ibm-watson/

https://www.ibm.com/blogs/think/2016/08/cognitive-movie-trailer/

IBM Training

Functional and Non-functional requirements

Functional requirements (FRs)	Non-functional requirements (NFRs)
Specify what the system should do or accomplish	Define system attributes such as security, reliability, performance, maintainability, scalability, availability, and usability.
Describe the functions and services that should be provided by the system	Guide how the system should fulfill the functional requirements
Provide information on how business needs and goals will be delivered through a specific project	Cover requirements that are not covered by the functional requirements. They specify the criteria that judge the working of a system.
Are implemented by the application capabilities, functions and features	Apply to a system as a whole

Enriching your applications with IBM Cloud services

Figure 7-6. Functional and Non-functional requirements

The first step to translate business requirements into technical terms is to identify the technical requirements. It is a good practice to document requirements of any system to be engineered by referring to functional and non-functional requirements. These requirements give the development team a clear and concise view of the system that is expected by the client.

Whereas a *business requirement* states the “why” for a project, a *functional requirement* outlines the “what”. Functional requirements are detailed and provide information on how business needs and goals will be delivered through a specific project. Functional requirements (FRs) specify what the system should do or accomplish. FRs describe the functions and services that should be provided by the system at a component level.

Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. Mainly, NFRs guide how the system should fulfill the functional requirements. NFRs cover requirements that are not covered by the functional requirements. Further, they specify the criteria that judge the working of a system, for example, availability, response time, maintainability.

Other examples of NFRs that are used frequently are:

- Portability
- Performance
- Documentation
- Certification
- Backup
- Fault Tolerance and high availability.

References:

https://en.wikipedia.org/wiki/Non-functional_requirement

https://en.wikipedia.org/wiki/Functional_requirement

IBM Training

Cognitive Tweets Analyzer : Functional requirements

Functional requirements (FRs)	Selection
Use a social media platform to capture viewers' reactions to the movie in real time.	**Twitter**.
Retrieve the data from the social media platform, orchestrate the services integration, and provide the logic of the solution.	**Node.js app** **Twitter API** **REST APIs**
The web UI can be accessed from mobile devices and desktop. The analyzed data results are displayed in the web UI with graphics and charts.	**Angular app**
Analyze the social media data to classify the information, detect the viewers' sentiment over time, extract the viewers' emotions, and the keywords that best describe the topics that they are talking about.	**Watson Natural Language Understanding** **Watson Tone Analyzer**
Store the data that is retrieved from the social media platform and the metadata (analyzed data) in a database.	**Cloudant**

Enriching your applications with IBM Cloud services

Figure 7-7. Cognitive Tweets Analyzer : Functional requirements

The slide shows a simplified version of the functional requirements for the sample Cognitive Tweets Analyzer app and the corresponding technical selection to implement the requirement.

- Use a social media platform to capture viewers' reactions to the movie in real time.

  Twitter is selected as the social media platform. Twitter is used every day by people to express their feelings or thoughts, especially about something that it is happening at the moment or just occurred.

  Analyzing Twitter and tweets is a powerful tool that can provide a sense on the public opinion about a topic that you are interested in.

- Retrieve the data from the social media platform, orchestrate the services integration, and provide the logic of the solution.

  Twitter provides an **API** that can be used in a cloud app to access and filter public tweets. The **cloud app**, that is written in **Node.js**, retrieves the tweets, orchestrates the integration with the other services in the system, and provides the logic of the solution. Node.js is a popular JavaScript runtime used to build server-side apps. The integration of services is implemented through **REST APIs**.

- Improve app's usability by providing a web UI that can be accessed from mobile devices and desktop. The analyzed data results are displayed in the web UI with graphics and charts.

  Angular is selected as the framework to build the UI for mobile and desktop apps. The results of the tweets analysis are presented as graphics and charts so the users can quickly get an overview of the reviews.

- Analyze the social media data to classify the information, detect the viewers' sentiment over time, extract the viewers' emotions, and keywords that best describe the topics that they are talking about.

  The **Watson Natural Language Understanding** service is selected to analyze the tweets text to extract metadata from the content such as concepts, keywords, categories, sentiment, emotion, and more.

  The **Watson Tone Analyzer** service is selected to perform linguistic analysis of the text in the tweets and detect joy, fear, sadness, anger, analytical, confident, and tentative tones found in text.

- Store the data that is retrieved from the social media platform and the metadata (analyzed data) in a database.

  The Twitter API returns tweets encoded using JSON. **Cloudant** is selected because it is a NoSQL database that is conceived as a native JSON database. The advantages of this type of database are that they are designed to be agile and scalable and they use dynamic schemas without defining the structure first.

IBM Training

Cognitive Tweets Analyzer : Non-functional requirements

Non-functional requirements (NFRs)	Selection
Users must authenticate to the app with their Google account to protect back-end resources and simplify the management of user access.	**App ID**
Monitor the health of the app to ensure its availability.	**IBM Cloud Monitoring**
Manage system and application logs in the cloud. Provide capabilities to view, monitor, and manage logs.	**LogDNA**

Enriching your applications with IBM Cloud services

Figure 7-8. Cognitive Tweets Analyzer : Non-functional requirements

The slide shows a simplified version of the non-functional requirements for the sample Cognitive Tweets Analyzer app and the corresponding technical selection to implement the requirement.

- Users must authenticate to the app with their Google account to protect back-end resources and simplify the management of user access.

  In all web and mobile applications, security is important to protect sensitive data, APIs and back ends running on the cloud. The **App ID** service is selected to add user's authentication through trusted identity providers, in this use case, Google.

- Monitor the health of the app to ensure its availability.

  The **IBM Cloud Monitoring** service is selected to define rules and alerts that notify the operations team of conditions that require attention. You can use the Monitoring service in the IBM Cloud to automatically collect and measure key metrics from your environment and applications.

- Manage system and application logs in the cloud. Provide capabilities to view, monitor, and manage logs.

  LogDNA was selected to add log management capabilities.

7.2. Solution architecture

IBM Training

Solution architecture

Enriching your applications with IBM Cloud services

Figure 7-9. Solution architecture

IBM Training

Topics

- Business problem and requirements
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

Enriching your applications with IBM Cloud services

Figure 7-10. Topics

IBM Training

IBM Cloud Architecture Center

- Architectures provide a roadmap to build, extend, and deploy an application.
- IBM Cloud Architecture Center provides:
  - Practices and reference architectures for building apps on the cloud
  - Editable architecture diagram templates
  - Solutions, templates, case studies, and samples in each architecture
- Examples of architecture types in IBM Cloud Architecture Center:
  - Multicloud
  - Hybrid
  - Built for AI
  - Popular application styles
  - Aspects for scaling applications
  - Industry solutions
- Access the IBM Cloud Architecture Center at https://www.ibm.com/cloud/garage/architectures

Enriching your applications with IBM Cloud services

Figure 7-11. IBM Cloud Architecture Center

Solving complex business problems with cloud technologies is much easier if you have a roadmap to the solution.

The IBM Cloud Garage Architecture Center is an excellent resource for designing and building solutions for the cloud. It provides practices and reference architectures for building apps on the cloud. The reference architectures define the basic pattern, while implementations provide specific technology, practices, and tool choices to build and deploy that pattern.

The architecture center provides editable architecture diagram resources. You can use architecture diagram templates to create your own architectures with simple icons to represent architecture components.

Examples of architecture types in IBM Cloud Architecture Center include:

- Multicloud

  Examples of architectures that provide multicloud solutions include private cloud, public cloud, and multicloud management.

- Hybrid

  Examples of architectures that provide hybrid solutions include application modernization, Digital Business Automation, and hybrid integration.

- Built for AI

  Includes examples of architectures that focus on AI solutions and analytics.

- Popular application styles

  Build a specific type of application, such as cloud-native apps with microservices, blockchain, IoT, mobile, e-commerce, and event-driven.

- Aspects for scaling applications

  Nonfunctional perspectives are the domains that you must consider regardless of the application style. This type includes service management, security, resilience, transformation with APIs, DevOps.

- Industry solutions

  This type includes architectures, solutions, and artifacts that apply a specific industry. Examples include banking and financial markets, automotive, insurance, retail, energy, and utilities and healthcare and life sciences.

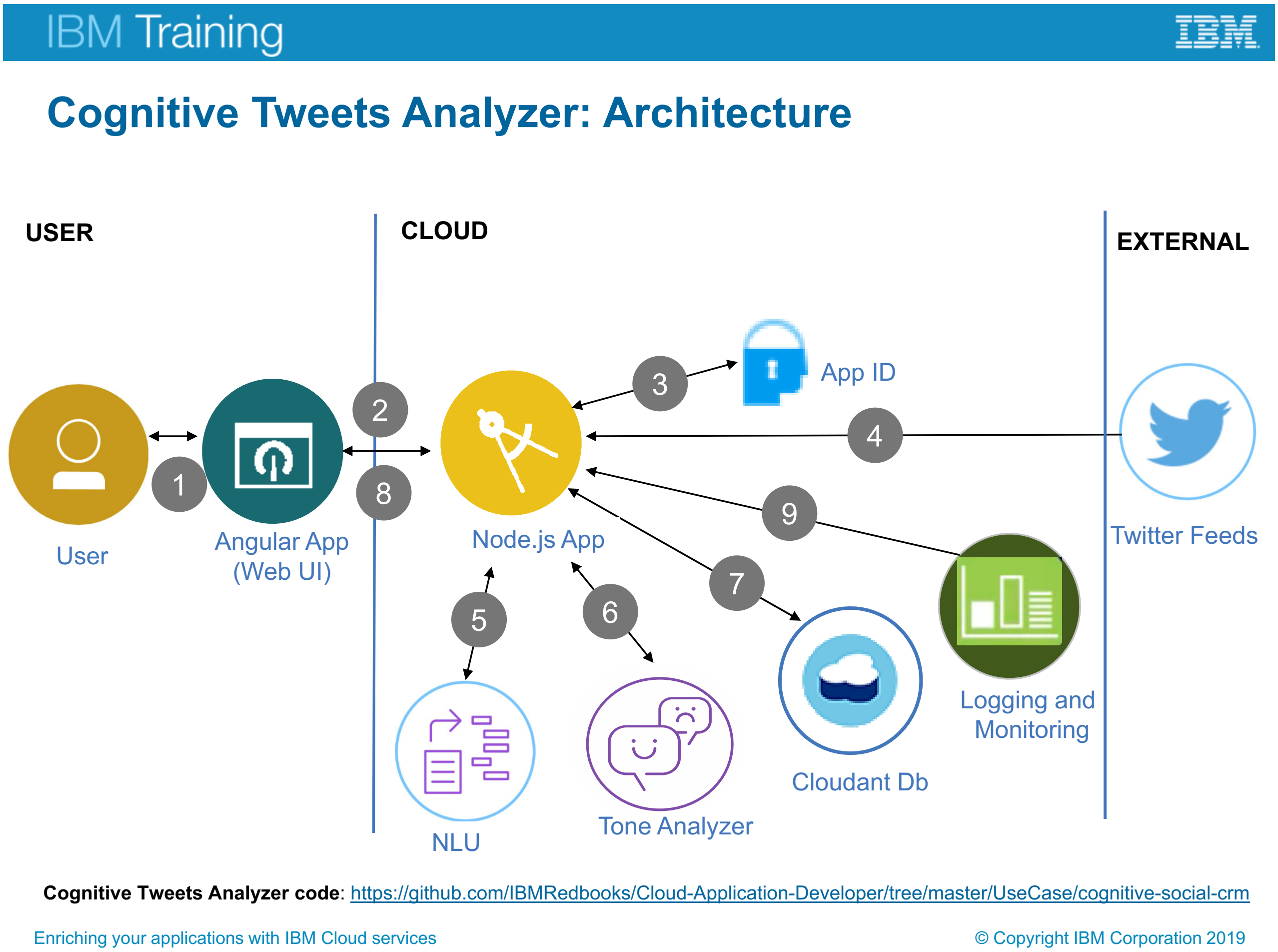


Figure 7-12. Cognitive Tweets Analyzer: Architecture

The cloud app that is featured in this unit provides a typical example of a cloud application design and implementation. The Node.js application represents an example of back-end service. It orchestrates the cloud services integration and provides the logic of the solution.

In this code pattern, the server application subscribes to a Twitter feed that is configured by the user. Each tweet that is received, is analyzed for emotional tone and sentiment. All data is stored in a Cloudant database, including historical data. The resulting analysis is displayed in a web UI as a series of graphs and charts.

The slide shows the architecture of the Cognitive Tweets Analyzer application:

1. The user opens the web UI app on the browser.
2. The user logs in using a Google account.
3. The web UI calls the Node.js application to check whether the user is authenticated. The Node.js app calls the App ID service to authenticate with the identity provider (Google in this use case).

Note

For details of this flow see the App ID service slides.

4. After the user is authenticated, tweets are pushed out by Twitter by using the Streaming Twitter API and the Node.js app processes the tweets.

Note

See the “Retrieving and processing the tweets” slides for details.

5. The Node.js app calls the Watson Natural Language Understanding service and passes the tweet information in the request. NLU returns keywords, entities, emotions and sentiments in the response.
6. The Node.js app calls the Watson Tone Analyzer service and passes the tweet information in the request. Tone Analyzer returns the emotional tone of the user in the response.
7. The Node.js app stores tweets and metadata (analyzed data) in the Cloudant database.
8. The user visualizes the results on the web UI. The web UI displays the analysis results as charts and graphs. It also shows the tweets details.
9. The logging service, LogDNA, logs the Node.js app data and the Monitoring service monitors the health of the app. In this use case, the Third-Party LogDNA service is integrated as a user-provided service. IBM Log Analysis with LogDNA from the IBM Cloud catalog is *not* used.

Note

See the LogDNA slides for details.

The Cognitive Tweets Analyzer app represents a simple cloud app that uses just a few services from the IBM Cloud catalog and a Third-Party services (LogDNA). The objective is to give you an example of creating rich cloud applications by using existing services on IBM Cloud, on other clouds, and even on premises.

Note

You can find the code for this sample application and directions to deploy and run the app at https://github.com/IBMRedbooks/Cloud-Application-Developer/tree/master/UseCase/cognitive-social-crm

References:

https://github.com/IBMRedbooks/Cloud-Application-Developer/tree/master/UseCase/cognitive-social-crm

https://github.com/IBM/cognitive-social-crm

7.3. Components

Figure 7-13. Components

This section describes the main components (programs and services) used in the Cognitive Tweets Analyzer sample application. For simplicity, this use case presents only a few apps and services but there are dozens of services available in the IBM Cloud catalog that you can integrate to enrich your cloud app.

IBM Training

Topics

- Business problem and requirements
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

Enriching your applications with IBM Cloud services

Figure 7-14. Topics

IBM Training

Node.js server app and Angular web UI

- Node.js server application
  - Retrieves the tweets, orchestrates the integration with the other services in the system, and provides the logic of the solution.
  - Uses the Twitter API to access and filter public tweets.
  - The integration of cloud services is implemented through REST API calls.
  - Implements standardized logging and monitoring.
  - Node.js is a popular Javascript runtime for server-side apps.
- Angular web UI
  - Provides a web UI that can be accessed from mobile devices and desktop.
  - Angular is a JavaScript framework that makes it easy to build front-end apps for the web. Some features of the Angular framework are:
    - Cross platform: Progressive web apps, native, and desktop
    - Speed and performance: Code generation, universal, code splitting
    - Productivity: Templates, Angular CLI, IDEs
    - Full development story: Testing, animation, accessibility

Enriching your applications with IBM Cloud services

Figure 7-15. Node.js server app and Angular web UI

Node.js server application

- Retrieves the tweets, orchestrates the integration with the other services in the system, and provides the logic of the solution.
- Uses the Twitter API to access and filter public tweets.
- The integration of cloud services is implemented through REST API calls.
- Implements standardized logging and monitoring.
- Node.js is a popular programming language for JavaScript server-side apps.

Note

You will learn about Node.js programming in the course *Developing Cloud Applications with SDK for Node.js*.

Angular web UI

- Provides a web UI that can be accessed from mobile devices and desktop.
- Angular is a JavaScript Framework that makes it easy to build front-end apps for the web. Some features of the Angular framework are:
  - Cross platform: Progressive web apps, native, and desktop
  - Speed and performance: Code generation, universal, code splitting
  - Productivity: Templates, Angular CLI, IDEs
  - Full development story: Testing, animation, accessibility

Angular features

- Cross platform
  - Progressive Web Apps

    Use modern web platform capabilities to deliver app-like experiences. High performance, offline, and zero-step installation.
  - Native

    Build native mobile apps with strategies from Cordova, Ionic, or NativeScript.
  - Desktop

    Create desktop-installed apps across Mac, Windows, and Linux using the same Angular methods you've learned for the web plus the ability to access native OS APIs.
- Speed and Performance
  - Code generation

    Angular turns your templates into code that is highly optimized for today's JavaScript virtual machines, giving you all the benefits of hand-written code with the productivity of a framework.
  - Universal

    Serve the first view of your application on Node.js, .NET, PHP, and other servers for near-instant rendering in just HTML and CSS. Also paves the way for sites that optimize for SEO.
  - Code splitting

    Angular apps load quickly with the new Component Router, which delivers automatic code-splitting so users only load code required to render the view they request.

- Productivity
  - Templates

    Quickly create UI views with simple and powerful template syntax.
  - Angular CLI

    Command line tools: start building fast, add components and tests, then instantly deploy.
  - IDEs

    Get intelligent code completion, instant errors, and other feedback in popular editors and IDEs.
- Full Development Story
  - Testing

    With Karma for unit tests, you can know if you've broken things every time you save. And Protractor makes your scenario tests run faster and in a stable manner.
  - Animation

    Create high-performance, complex choreographies and animation timelines with very little code through Angular's intuitive API.
  - Accessibility

    Create accessible applications with ARIA-enabled components, developer guides, and built-in a11y test infrastructure.

Reference:

https://angular.io/features

IBM Training

App ID service: Overview

- Secures resources and adds authentication to mobile or web applications with few lines of code.
- By requiring users to sign in to your app, you can:
  - Store user data such as app preferences.
  - Store information from public social profiles.
  - Use the user’s data to customize each user's experience within the app.
- Provides a log-in framework, but you can also bring your own branded screens to use with Cloud Directory.
- Based on industry standard protocols and specifications:
  - OAuth 2.0 Authorization Framework
  - Open ID Connect.
  - OAuth 2.0
- App ID successfully completed several certifications, audits, and standards.

Enriching your applications with IBM Cloud services

Figure 7-16. App ID service: Overview

Application security can be incredibly complicated. For most developers, it is one of the hardest parts of creating an app. By integrating IBM Cloud App ID into your apps, you can secure resources and add authentication, even if you do not have much security experience.

Benefits

App ID helps developers to easily add authentication to their web and mobile apps with few lines of code, and secure their Cloud-native applications and services on IBM Cloud. By requiring users to sign in to your app, you can store user data such as app preferences, or information from public social profiles, and then use that data to customize each user's experience within the app. App ID provides a log-in framework for you, but you can also bring your own branded screens to use with Cloud Directory.

Compliance and standards

App ID successfully completed several certifications, audits, and standards.

App ID is based on a set of well-known, industry standard protocols and specifications that are frequently found in both enterprise and consumer facing applications, the OAuth 2.0 Authorization Framework and Open ID Connect. OAuth 2.0 is used to obtain and verify authorization for accessing protected resources. On top of that, Open ID Connect adds a layer of an authentication and identity protection to your application.

References:

Check out this video to learn more about the different ways that you can use the service.

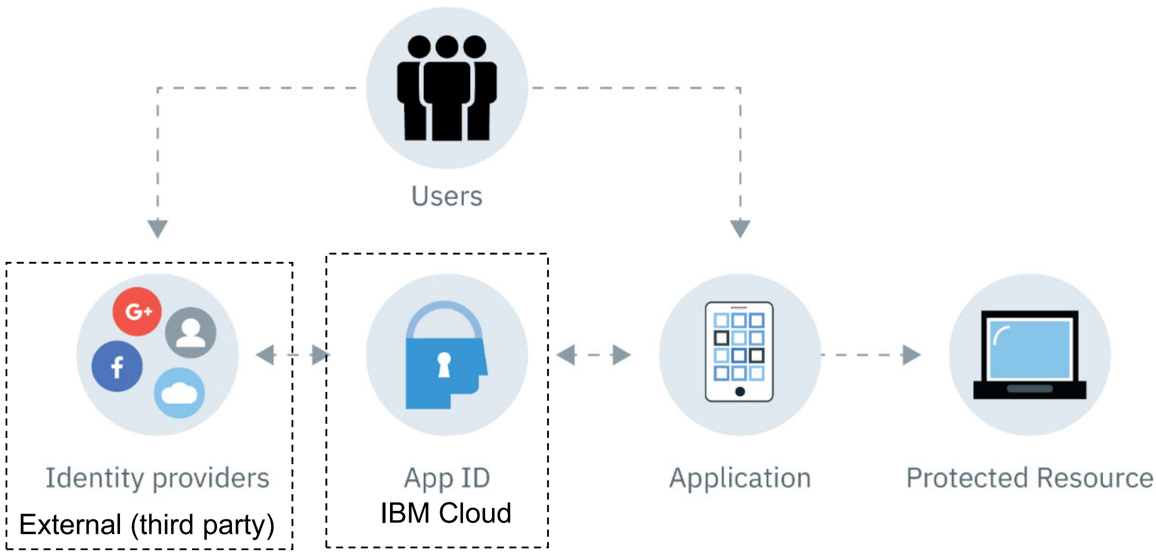
https://youtu.be/XIrCjHdK43Q

https://cloud.ibm.com/docs/services/appid?topic=appid-about

https://cloud.ibm.com/docs/services/appid?topic=appid-key-concepts#key-concepts

IBM Training

App ID service: How it works



Enriching your applications with IBM Cloud services

Figure 7-17. App ID service: How it works

With App ID, you can add a level of security to your apps by requiring users to sign in. You can also use the server SDK or APIs to protect your back-end resources.

Application

- Server SDK: You can protect your back-end resources that are hosted on IBM Cloud and your web apps by using the server SDK. It extracts the access token from a request and validates it with App ID.
- Client SDK: You can protect your mobile apps with the Android or iOS client SDK. The client SDK communicates with your cloud resources to start the authentication process when it detects an authorization challenge.

IBM Cloud

- App ID: After successful authentication, App ID returns access and identity tokens to your app.
- Cloud directory: Users can sign up for your service with their email and a password. You can then manage your users in a list view through the UI. With cloud directory, App ID functions as your identity provider.

External (third party)

Social and enterprise identity providers: App ID supports Facebook, Google+, and SAML 2.0 Federation as identity provider options. The service arranges a redirect to the identity provider and verifies the returned authentication tokens. If the tokens are valid, the service grants access to your app without ever having access to the actual passphrase.

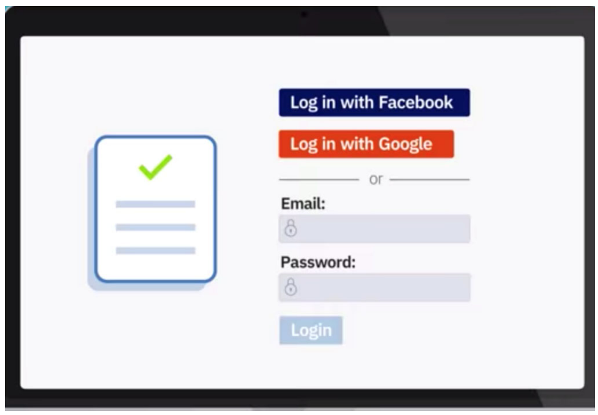
Reference:

https://cloud.ibm.com/docs/services/appid?topic=appid-about

IBM Training

App ID service: Authentication options

Identity providers	Type
Cloud Directory	Managed registry
SAML	Enterprise
Facebook	Social
Google+	Social
Custom	



Enriching your applications with IBM Cloud services

Figure 7-18. App ID service: Authentication options

There are several identity providers that the App ID service can be configured to use:

Cloud Directory: It is a managed registry. You can maintain your own user registry in the cloud. When a user signs up for your app, they are added to your directory of users. This option gives your users more freedom to manage their own account within your app.

SAML: It is of enterprise type .You can create a single sign-on experience for your end users.

Facebook: It is of social type . Users can sign in to your app by using their Facebook credentials.

Google+: It is of social type. Users can sign in to your app by using their Google credentials. This identity provider is used in the Cognitive Tweets Analyzer app.

Custom: If none of the provided options fit your specific need, you can configure your own identity flow to work with App ID.

To sign in with their Facebook and Google accounts, users need their Facebook App ID and Google Client ID respectively.

- Facebook
  - Users log in to the 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. 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
  - Users log in to the 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. You must create a project in Google Developer Console at https://console.developers.google.com/apis/library to receive the Google Client ID

Reference:

https://cloud.ibm.com/docs/services/appid?topic=appid-managing-idp

IBM Training

App ID service: Scenarios and solutions (optional)

Scenario	Solution
You need to add authorization and authentication to your mobile and web apps but do not have a background in security.	App ID makes it easy to add an authentication step to your apps. You can add email or username sign in, social sign in, or enterprise sign in to your apps with APIs, SDKs, prebuilt UIs, or your own branded UIs.
You want to limit access to your apps and back-end resources.	You can secure your apps, back-end resources, and APIs easily by using the standards based authentication provided by App ID.
You want to build personalized app experiences for your users.	With App ID, you can store user data such as app preferences or information from their public social profiles, and then use that data to customize each experience of your app.
You want to manage users in a scalable way.	App ID allows you to create a Cloud Directory, which makes it possible for you to add user sign-up and sign-in to your apps. Cloud Directory provides you with the framework to maintain a user registry that can scale with your user base. With the pre-built functionality for self service, such as email verification and password resets, you can be sure that your app is authenticating users securely.

Enriching your applications with IBM Cloud services

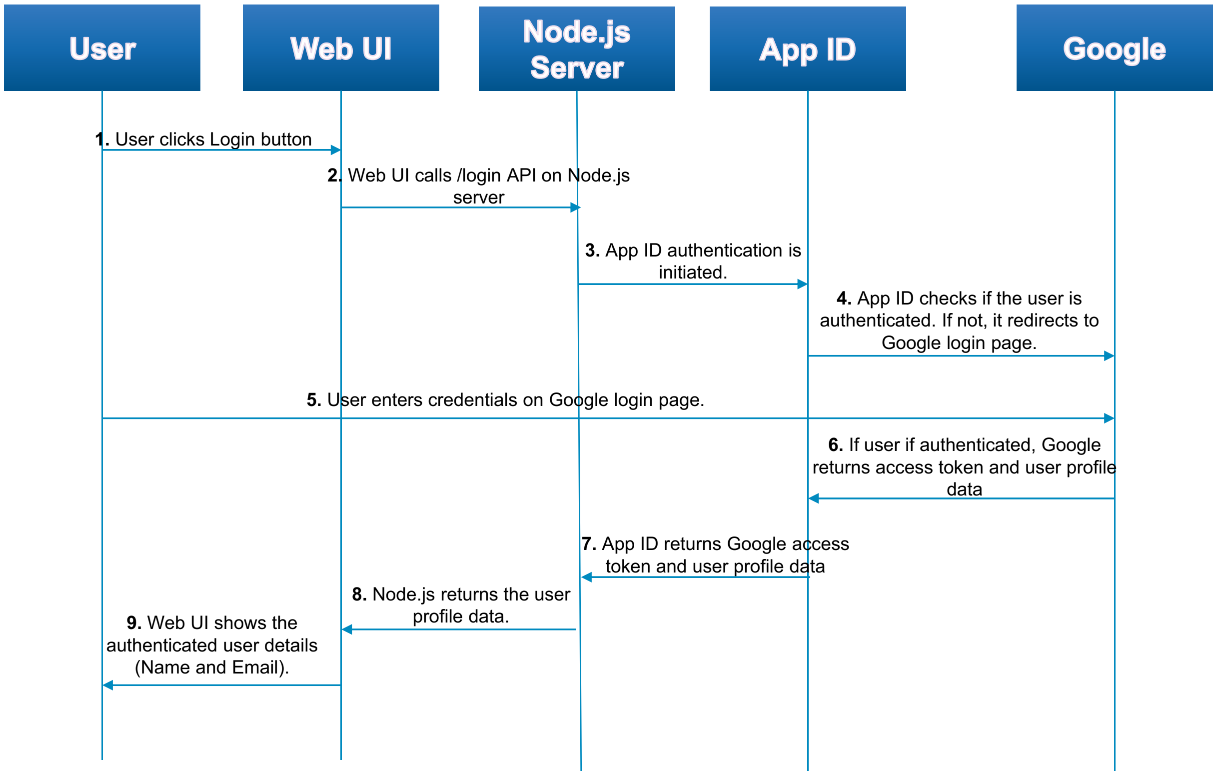
Figure 7-19. App ID service: Scenarios and solutions (optional)

Why would you want to use App ID? Check out the scenarios in the slide to see whether any of them apply to you.

In the Cognitive Tweets Analyzer App, App ID is used for Authorization and Authentication by using Google account.

IBM Training

How is App ID used in Cognitive Tweets Analyzer



Enriching your applications with IBM Cloud services

Figure 7-20. How is App ID used in Cognitive Tweets Analyzer

App ID authentication flow for Cognitive Tweets Analyzer:

1. The user clicks Login on the web UI.
2. The web UI calls the /login API on Node.js app to check if the user is authenticated
3. The Node.js login API initiates the App ID authentication flow.
4. App ID contacts Google identity provider to authenticate the user, which redirects to the Google login page.
5. The user enters credentials on Google login page.
6. If the user is authenticated successfully, Google returns the results (access token + user profile data) to App ID.
7. App ID returns results (access token + user profile data) to Node.js.
8. Node.js returns the user profile data to web UI.
9. Web UI shows the authenticated user's information (name and email) and the user is logged in.

IBM Training

Retrieving and processing tweets

1. Get Twitter API keys
   - API key, API secret, Access token, and Access token secret
2. Install the Twitter library
   - **twit (**a Twitter API client for Node.js) is used in the Cognitive Tweets Analyzer app
3. Connect to Twitter Streaming APIs

```
//initializing Twit Library
const twitOptions = {};
 twitOptions.consumer_key = config.consumer_key || '';
 twitOptions.consumer_secret = config.consumer_secret || '';
 twitOptions.access_token = config.access_token;
 twitOptions.access_token_secret = config.access_token_secret;
 twitOptions.timeout_ms = 60 * 1000; // optional HTTP request timeout to apply to all requests.
 this.twitterClient = new Twit(twitOptions);

//Calling Twit stream API
 const twitParams = {};
 twitParams.lang = 'en';
 twitParams.follow = this.options.userIds;
 this.status.listening = this.options.listenTo;
 this.stream = this.twitterClient.stream('statuses/filter', twitParams);
```


Figure 7-21. Retrieving and processing tweets

The slide shows the high level steps for retrieving and processing tweets and a sample code snippet.

Getting Twitter API keys

To start with, you need to have a Twitter developer account and obtain credentials (that is, API key, API secret, Access token, and Access token secret) to access the Twitter API. Follow these steps:

1. Create a Twitter developer account if you do not already have one from https://developer.twitter.com/
2. Go to https://developer.twitter.com/en/apps and log in with your Twitter account.
3. Click **Create an app**.
4. Complete the form, and click **Create**.
5. At the pop up window that opens, review the Developer Terms. Click **Create** again.
6. In the next page, select the **Keys and Access Tokens** tab, and copy your *API key* and *API secret* from the Consumer API keys section.
7. Scroll down to the **Access token & access token secret** section and click **Create**. Then, copy your *Access token* and *Access token secret*.

Installing the Twitter library

There are many libraries in various programming languages that let you use the Twitter API. See https://developer.twitter.com/en/docs/developer-utilities/twitter-libraries

In this use case, the **twit** library, which is a Twitter API client for Node.js, is used. See https://www.npmjs.com/package/twit

Connecting to Twitter Streaming APIs

The Streaming APIs give access to (usually a sample of) all tweets as they are published on Twitter. On average, about 6,000 tweets per second are posted on Twitter and developers get a small proportion (<=1%) of it. The Streaming APIs are one of the two types of Twitter APIs. The other type is REST APIs. In this use case, the Streaming API is used. Streaming API sends out real-time tweets only.

Reference:

https://developer.twitter.com/

https://developer.twitter.com/en/docs

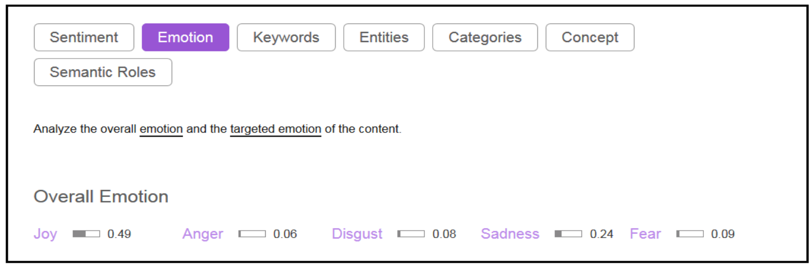
http://socialmedia-class.org/twittertutorial.html

https://developer.twitter.com/en/docs/developer-utilities/twitter-libraries

IBM Training

Watson Natural Language Understanding service

- NLU extracts meaning from unstructured data.
- NLU processes input provided as text, HTML, or a public URL.
- NLU analyzes semantic features of input text and provides an output that includes:
  - Categories
  - Emotion
  - Entities and relationships
  - Sentiment analysis
  - Keywords



- It can be trained by creating a custom model by using IBM Watson Knowledge Studio.
- Example applications: Categorize news articles and blog posts and sort them based on general concepts, keywords, and entities.

Enriching your applications with IBM Cloud services

Figure 7-22. Watson Natural Language Understanding service

Watson Natural Language Understanding (NLU) includes a set of text analytics features that you can use to extract meaning from unstructured data. NLU accepts user input and provides an output that includes entities and relationships that are found within the text and performs sentiment analysis of the input text.

Input can be in the form of text, raw HTML, or a public URL. The service cleans HTML content before analysis by default so the results can ignore most advertisements and other unwanted content.

You can create custom models by using Watson Knowledge Studio to detect custom entities and relations in NLU.

Custom annotation models are developed by using Watson Knowledge Studio to identify industry- and domain-specific entities and relations in unstructured text

Here are summaries of the metadata that is returned by Watson Natural Language Understanding:

- Categories

  Identify high-level concepts that are not necessarily directly referenced in the text.

- Concepts

  Categorize your content by using a five-level classification hierarchy. You can view the complete list of categories at the following website:

  https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-categories-hierarchy

- Emotions

  Analyze emotions that are conveyed by specific target phrases or by the document as a whole. You can also enable emotion analysis for entities and keywords that are automatically detected by the service.

- Entities

  Find people, places, events, and other types of entities that are mentioned in your content. You can view the complete list of entity types and subtypes at the following website:

  https://cloud.ibm.com/docs/services/natural-language-understanding?topic=natural-language-understanding-entity-type-systems

Examples:

- **Entities**

  Input

  Text: "I love apples, but I hate oranges."
  Targets: "apples", and "oranges"

  Response

  "apples": joy
  "oranges": anger

- Keywords

  Search your content for relevant keywords. For example:

  Input

  URL: "http://www-03.ibm.com/press/us/en/pressrelease/51493.wss"

  Response

  Australian Open
  Tennis Australia
  IBM SlamTracker analytics

- Metadata

  For HTML and URL input, get the author of the webpage, the page title, and the publication date.

- Relations

  Recognize when two entities are related, and identify the type of relation. For example:

  Input

  Text: "The Nobel Prize in Physics 1921 was awarded to Albert Einstein."

  Response

  "awardedTo" relation between "Noble Prize in Physics" and "Albert Einstein"
  "timeOf" relation between "1921" and "awarded"

- Semantic roles

  Parse sentences into subject-action-object form, and identify entities and keywords that are subjects or objects of an action. For example:

  Input

  Text: "In 2011, Watson competed on Jeopardy!"

  Response

  Subject: Watson
  Action: competed
  Object: on Jeopardy

- Sentiment

  Analyze the sentiment toward specific target phrases and the sentiment of the document as a whole. You can also get sentiment information for detected entities and keywords by enabling the sentiment option for those features. For example:

  Input

  Text: "Thank you and have a nice day!"

  Response

  Positive sentiment (score: 0.91)

- Custom annotation models
  Custom annotation models are developed by using Watson Knowledge Studio to identify industry- and domain-specific entities and relations in unstructured text.

- Example of applications

  Categorize news articles and blog posts and sort them based on general concepts, keywords, and entities.

Reference:

https://cloud.ibm.com/apidocs/natural-language-understanding?code=node#introduction

IBM Training

How is Natural Language Understanding used in Cognitive Tweets Analyzer

1. Configure Natural Language Understanding API with the required features:

```
//configuring nlu parameters to be sent to the nlu analyze api
  nluParams = {
    features: {
      emotion: {},
      sentiment: {},
      entities: {
        emotion: false,
        sentiment: false,
        limit: 2
      },
      keywords: {
        emotion: false,
        sentiment: false,
        limit: 2
      }
    }
  };
```

2. Call Natural Language Understanding **analyze** API

```
try {
    this.nluParams.text = text;
    this.nluParams.language = 'en';

this.nlu.analyze(this.nluParams, (err, success) => {
 if (err) {
    this.LOGGER.error('NLU: ' + err);
    return reject('NLU: ' + err);
 }
 resolve({ nlu: success });
 });
 } catch (err) {
    reject(err);
 }
```

Enriching your applications with IBM Cloud services

Figure 7-23. How is Natural Language Understanding used in Cognitive Tweets Analyzer

1. To call the Natural Language Understanding *analyze* API, configure the API with the required **features** to be returned from the API. The Cognitive Tweets Analyzer use case requires the following features:
   - Emotion
   - Sentiment
   - Entities
   - Keywords
2. Add also the **text** to be analyzed and the **language** of the text to the params. Then, call the *analyze* API.

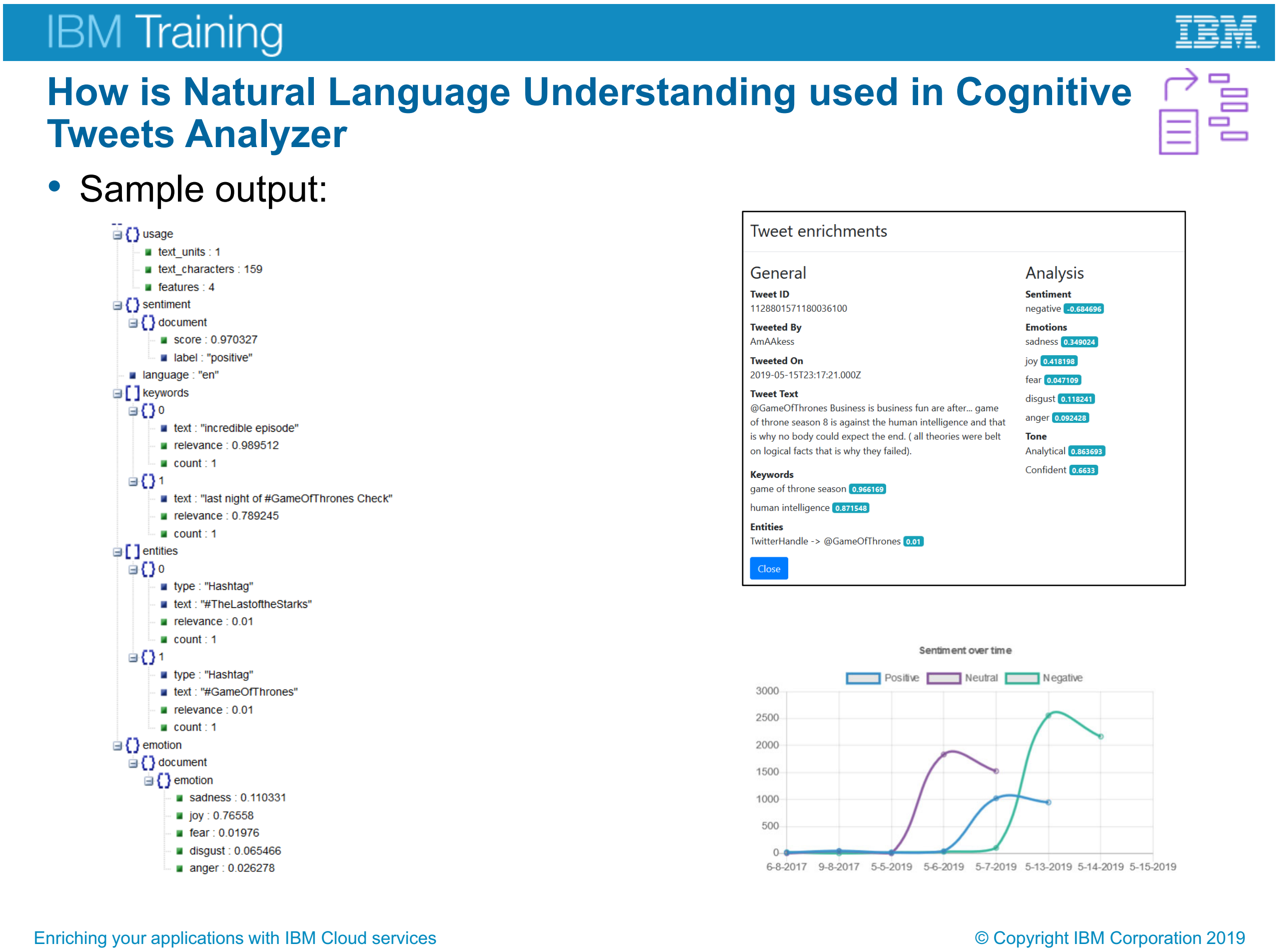


Figure 7-24. How is Natural Language Understanding used in Cognitive Tweets Analyzer

The response from the *analyze* API contains the following objects:

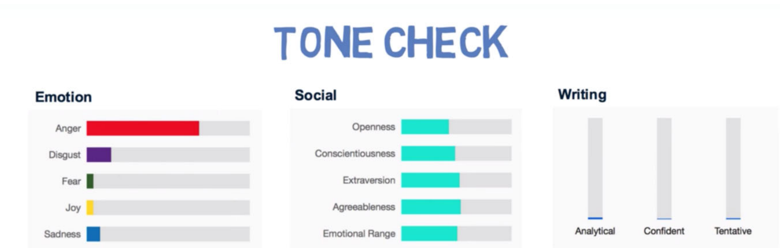
- usage
- sentiment
- language
- keywords
- entities
- emotion.

The resulting data from *analyze* API, is shown on the Web UI in the tweets analysis details and represented also as graphs and charts.

IBM Training

Watson Tone Analyzer service

- Watson Tone Analyzer uses linguistic analysis to identify various tones. It detects three types of tones from text:
  - Emotion (anger, disgust, fear, joy, and sadness)
  - Social tendencies (openness, conscientiousness, extroversion and introversion, agreeableness, and emotional range)
  - Language/Writing styles (analytical, confident, and tentative)
- The service offers two endpoints:
  - General-purpose endpoint: Used to analyze shorter web data, such as email messages or tweets, or longer documents, such as articles or blog posts.
  - Customer-engagement endpoint: Used to monitor customer service and support conversations.



Enriching your applications with IBM Cloud services

Figure 7-25. Watson Tone Analyzer service

Tone Analyzer

Tone impacts the effectiveness of communication in different contexts.

People show various tones, such as joy, sadness, anger, and agreeableness, in daily communications. Such tones can impact the effectiveness of communication in different contexts.

Watson Tone Analyzer uses linguistic analysis to identify various tones at both the sentence and document level. This insight can then be used to refine and improve communications.

It detects three types of tones:

- Emotion (anger, disgust, fear, joy, and sadness)
- Social propensities (openness, conscientiousness, extroversion and introversion, agreeableness, and emotional range)
- Language/Writing styles (analytical, confident, and tentative) from text

Use the Watson Tone Analyzer API in your applications to understand emotions, social tendencies, and perceived writing style.

The service offers two endpoints:

General-purpose endpoint

Use the Tone Analyzer general-purpose endpoint to analyze shorter web data, such as email messages or tweets, or longer documents, such as articles or blog posts. Monitor social media to understand what customers are saying about a brand and to determine whom to target with specific messaging. The endpoint accepts JSON, plain text, or HTML input. For more information about the method and the tones that it returns.

Customer-engagement endpoint

Use the Tone Analyzer customer-engagement endpoint to monitor customer service and support conversations. Escalate customer conversations when they turn sour or find opportunities to improve customer service scripts, dialogue strategies, and customer journeys. The endpoint accepts JSON input. For more information about the method and the tones that it returns

References:

https://cloud.ibm.com/docs/services/tone-analyzer?topic=tone-analyzer-gettingStarted

https://cloud.ibm.com/apidocs/tone-analyzer

https://cloud.ibm.com/docs/services/tone-analyzer?topic=tone-analyzer-about

https://cloud.ibm.com/docs/services/tone-analyzer?topic=tone-analyzer-utgpe

https://cloud.ibm.com/docs/services/tone-analyzer?topic=tone-analyzer-utco

IBM Training

How is Tone Analyzer used in Cognitive Tweets Analyzer

- Call Tone Analyzer **tone** API passing the text to be analyzed

```
try {
        this.toneParams.text = text;
        this.toneParams.sentences = false;
        this.toneAnalyzer.tone(this.toneParams, (err, success) => {
          if (err) {
            this.LOGGER.error('Tone: ' + err);
            return reject('Tone: ' + err);
          }
          resolve({ tone: success });
        });
} catch (err) {
  reject(err);
}
```


Figure 7-26. How is Tone Analyzer used in Cognitive Tweets Analyzer

To call the Tone Analyzer *tone* API, send the following items to the API as parameters:

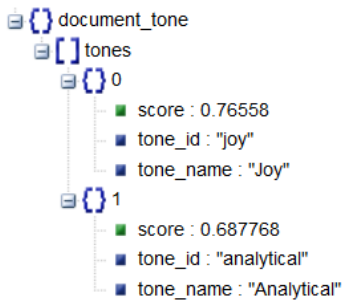
- text (to be analyzed)
- sentences

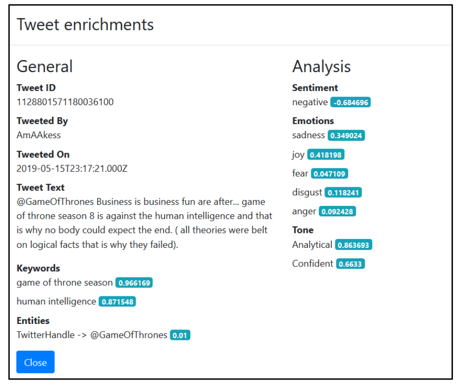
In Cognitive Tweets Analyzer, the *sentences* param is set to *false*, as we do not need the API to return the analysis of each sentence in addition to the analysis of the full text.

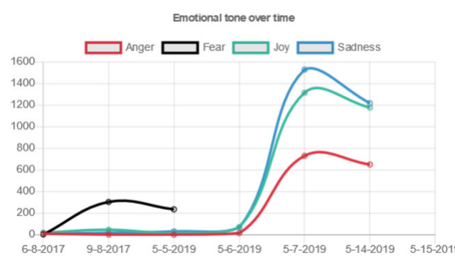
IBM Training

How is Tone Analyzer used in Cognitive Tweets Analyzer

- Sample output:







Enriching your applications with IBM Cloud services

Figure 7-27. How is Tone Analyzer used in Cognitive Tweets Analyzer

The response from the *tone* API contains the tones that are detected in the text. Each tone object contains:

- score
- tone_id
- tone_name

The resulting data from the *tone* API, is shown on the web UI in the tweets analysis details and it is represented also as graphs.

IBM Training

Cloudant database

- Fully managed
  - NoSQL database that is conceived as a native JSON database.
    - The Twitter API returns tweets encoded using JSON.
  - Provides a fully managed, distributed JSON document database.
  - Easy to deploy and create databases on IBM Cloud.
  - Scale throughput capacity and data storage to meet application’s requirements.
- Secure
  - ISO27001, SOC 2 Type 2 compliant, and HIPAA ready. All data is encrypted over the wire and at rest.
- Globally available
  - Available in all IBM Cloud regions and 55+ data centers across the world.
  - Easy to set up for disaster recovery between continents.
  - Easy to set up for scaling an app through a horizontal scaling architecture that can handle millions of users and terabyte.
- Data flexibility
  - Flexible JSON schema and powerful API that is compatible with Apache CouchDB
- Durable replication
  - Move application data closer to all the places it needs to be, for uninterrupted data access, offline or on.

Enriching your applications with IBM Cloud services

Figure 7-28. Cloudant database

Fully managed

The IBM Cloud™ provides a fully managed, distributed JSON document database. Instantly deploy an instance, create databases, and independently scale throughput capacity and data storage to meet your application requirements. IBM expertise takes away the pain of hardware and software provisioning, patching, and upgrades, while it offers a 99.95 percent SLA (service-level agreement).

Secure

IBM Cloudant® is ISO27001, SOC 2 Type 2 compliant, and HIPAA ready. All data is encrypted over the wire and at rest with optional user-defined key management through IBM Key Protect. Cloudant also offers an EU-managed service that ensures all data and operations are handled solely by EU citizens.

Globally available

Available in all IBM Cloud regions and 55+ data centers across the world, Cloudant can easily be set up for disaster recovery between continents or scaling an app for a global release through a horizontal scaling architecture that can handle millions of users and terabytes of data to grow seamlessly alongside your business. All Cloudant instances are deployed on clusters that span availability zones in regions that support them, for added durability at no extra cost.

Data flexibility

Leverage a flexible JSON schema and powerful API that is compatible with Apache CouchDB™, enabling you to access an abundance of language libraries and tools to rapidly build new applications and features.

Durable replication

Move application data closer to all the places it needs to be, for uninterrupted data access, offline or on. Cloudant helps teams build progressive web apps, develop with an offline-first architecture, or manipulate data on edge devices.

Note

Cloudant is discussed in detail in the Cloud Database course.

Reference:

https://www.ibm.com/cloud/cloudant/details

IBM Training

How is Cloudant used in Cognitive Tweets Analyzer

1. Initialize Cloudant instance by using Cloudant credentials.

```
const cloudant = Cloudant({
   account: config.cloudant_username,
   password: config.cloudant_password,
   plugins: { retry: { retryErrors:
false, retryStatusCodes: [429] } }
    });
this.cloudant = cloudant;
```

2. Call Bulk Save to Cloudant

```
this.LOGGER.debug('Saving to
Cloudant...');
this.cloudantDB.bulk(this.bulkSaveBuffer,
(err, result) => {
   if (err) {
    this.LOGGER.error('Error while saving
to database::' + err);
   reject(err);
} else {
  this.LOGGER.debug('Successfully saved '
+
  this.bulkSaveBuffer.docs.length +
 ' docs to Cloudant.');
this.bulkSaveBuffer.docs = [];
  resolve();
 }
});
```

Enriching your applications with IBM Cloud services

Figure 7-29. How is Cloudant used in Cognitive Tweets Analyzer

1. To call Cloudant APIs you must initialize the Cloudant instance first using the Cloudant service credentials (username and password).
2. In Cognitive Tweets Analyzer, the Bulk Save of analyzed tweets is used to save multiple tweets at the same time, with one call to Cloudant *bulk* API.

IBM Training

LogDNA service

- Centralized cloud log management software
  - Aggregates all system and application logs in one centralized logging system
  - Third-Party service from LogDNA https://logdna.com/
- Features:
  - Troubleshoot logs in real time to diagnose issues and identify problems
  - Automatic parsing and indexing of log sources
  - Keyword-based log search and graphing
  - Get alert notifications of important events and errors
  - Provides easy and fast integration of various log sources

IBM Cloud Log Analysis with LogDNA Service

- Based on the LogDNA Third-Party product
- Operated by LogDNA in partnership with IBM
- Offers administrators, DevOps teams, and developers advanced features to filter, search, and tail log data, define alerts, and design custom views to monitor application and system logs.
- Does *not* support Cloud Foundry

Note: In the Cognitive Tweets Analyzer application, LogDNA (not IBM Cloud Log Analysis with LogDNA) is used due to the lack of Cloud Foundry support in the IBM Cloud offering

Enriching your applications with IBM Cloud services

Figure 7-30. LogDNA service

LogDNA is a centralized cloud log management software that you can integrate in your cloud app to add log management capabilities. It aggregates all system and application logs in one centralized logging system. It is provided by a Third-Party, LogDNA .

Features

- Troubleshoot logs in real time to diagnose issues and identify problems

  By using LogDNA's streaming "live tail", developers and DevOps teams can diagnose issues, analyze stack traces and exceptions, identify the source of errors, and monitor different log sources through a single view. This feature is available through the command line and through the web interface.

- Automatic parsing and indexing of log sources

  Automatically parse and index all log types and formats that are ingested. LogDNA intelligently detects log types, parses them to display beautifully, and more importantly indexes them in a way that makes keyword search fast and easy to use. Custom parsing rules can be configured for any proprietary formats as well.

- Keyword-based log search and graphing

  Use simple to use keyword-based search to search across your logs instead of fiddling with custom query languages. Apply the same keyword search to build beautiful time series graphs instantly.

- Get alert notifications of important events and errors

  Act promptly on application and service logs that you identify as critical. LogDNA users can configure multi-channel alert notifications based on pattern matching to various direct integrations such as email, Slack, PagerDuty, Opsgenie, VictorOps, or your own custom webhooks.

- Provides easy and fast integration of various log sources

  LogDNA offers various integrations for log ingestion:

  - Operating system logs (Windows, Linux, MacOS)
  - Platform logs (Kubernetes, Docker)
  - Syslog
  - APIs (REST, cURL)
  - Code libraries (Node.js, Ruby, Rails, Python, Java, PHP, Go, iOS)

IBM Cloud Log Analysis with LogDNA Service

IBM Log Analysis with LogDNA is operated by LogDNA in partnership with IBM. You can use IBM Log Analysis with LogDNA to manage system and application logs in the IBM Cloud.

IBM Log Analysis with LogDNA offers administrators, DevOps teams, and developers advanced features to filter, search, and tail log data, define alerts, and design custom views to monitor application and system logs.

To add logging features with LogDNA in the IBM Cloud, you must provision an instance of IBM Log Analysis with LogDNA service.

Before you provision an instance of IBM Log Analysis with LogDNA service, consider the following information:

- Log data is hosted on the IBM Cloud.
- Log data is sent to a third party.
- Your users must have permissions to create, view, and delete an instance of a service in the IBM Cloud.
- Your users must have permissions to create resources within the context of the resource group where you plan to provision the LogDNA instance.

IBM Cloud Log Analysis with LogDNA can be used to manage Kubernetes clusters and Linux Ubuntu or Debian servers logs only but not Cloud Foundry apps.

Note

In the Cognitive Tweets Analyzer application, LogDNA (not IBM Cloud Log Analysis with LogDNA) is used due to the lack of Cloud Foundry support in the IBM Cloud offering.

References:

https://logdna.com/

https://www.ibm.com/cloud/log-analysis

https://cloud.ibm.com/docs/services/Log-Analysis-with-LogDNA?topic=LogDNA-about

IBM Training

Integrating LogDNA with Cognitive Tweets Analyzer

- The Cognitive Tweets Analyzer Cloud Foundry application integrates LogDNA through a user-provided service instance.
- User-provided service instances enable developers to use services that are not available in IBM Cloud with their apps running on Cloud Foundry.
  - You can bind your application to services outside of IBM Cloud.
  - Once created, user-provided service instances behave like service instances that are created through the IBM Cloud catalog.
- To use a user-provided service with your Cloud Foundry application on IBM Cloud, you need:
  - A service outside of IBM Cloud (LogDNA in this use case)
  - Host and port to the external service (endpoint URL)
  - Credentials to access the user-provided service (if required by the service)
  - Create the user-provided service instance
  - Your application binds to the user-provided service

Enriching your applications with IBM Cloud services

Figure 7-31. Integrating LogDNA with Cognitive Tweets Analyzer

User-provided service instances enable developers to use services that are not available in IBM Cloud with their apps running on Cloud Foundry. An example of such a service is LogDNA. Once created, user-provided service instances behave like service instances created through the IBM Cloud catalog.

To use a user-provided service with your Cloud Foundry application on IBM Cloud, you need:

- A service outside of IBM Cloud (LogDNA in this use case)
- Host and port to the external service (endpoint URL)
- Credentials to access the user-provided service (optional)
- Create the user-provided service instance
- Your application binds to the user-provided service

References:

https://www.ibm.com/support/knowledgecenter/en/SSBS6K_2.1.0/cloud_foundry/buildpacks/buildpacks_userprovidedservices.html

https://docs.cloudfoundry.org/devguide/services/user-provided.html

https://logdna.com/blog/logging-your-cloud-foundry-apps-to-logdna/

IBM Training

Integrating LogDNA with Cognitive Tweets Analyzer (cont.)

- Summary of integration steps. Ensure that you have the IBM Cloud CLI installed in your workstation:

```
1. Login to LogDNA at https://app.logdna.com
2. Create your organization
3. Choose to connect Via Platform > Cloud Foundry
4. Select Provision a syslog port for Cloud Foundry.
5. You will receive a new LogDNA syslog endpoint, which includes the
   host and port that you need to connect to the LogDNA service.
   Example: syslog-a.logdna.com:15177

6. Run the following ibmcloud CLI command to create the user-provided
   services, name it my-logs and link it to the syslog url:
   ibmcloud cf cups my-logs -l syslog://syslog.logdna.com:<syslog-port>
   where <syslog-port> is the syslog port you provisioned in step 5.

7. Run the following command to bind the user-provided service to
   your Cloud Foundry application:

   ibmcloud cf bind-service <CF APP NAME> my-logs
   where <CF APP NAME>, is the Cloud Foundry application name.

8. Run the following command, to restart the application after
   binding it to the LogDNA service.
    ibmcloud cf restart <CF APP NAME>
      where <CF APP NAME>, is the Cloud Foundry application name
```

Enriching your applications with IBM Cloud services

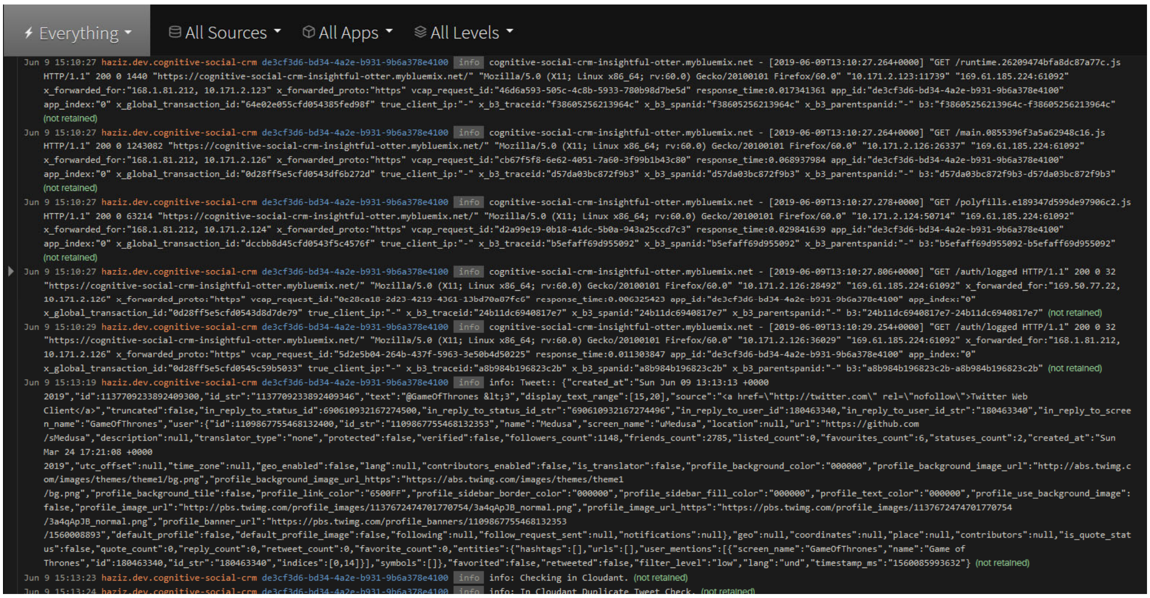
Figure 7-32. Integrating LogDNA with Cognitive Tweets Analyzer (cont.)

The slide summarizes the steps to integrate the Cloud Foundry Cognitive Tweets Analizer app with LogDNA through a user-provided service.

IBM Training

Integrating LogDNA with Cognitive Tweets Analyzer (cont.)

- Sample logs on LogDNA Dashboard



Enriching your applications with IBM Cloud services

© Copyright IBM Corporation 2019

Figure 7-33. Integrating LogDNA with Cognitive Tweets Analyzer (cont.)

This slide shows a sample output of Cognitive Tweets Analyzer app logs on LogDNA Dashboard.

IBM Training

IBM Cloud Monitoring service

- Expand your collection and retention capabilities when working with metrics.
- Define rules and alerts that notify you of conditions that require attention.
- Gain insight into how your apps are performing and consuming resources.
- Identify trends, detect and diagnose problems.
- Integrate your monitoring data into your applications and operations through the Monitoring service APIs.
- Send metrics for your Cloud Foundry applications and Virtual Machines (VMs) into the Monitoring service.
- Provision the Monitoring service through the IBM Cloud catalog.
- View and analyze metrics collected by the Monitoring service through the IBM Cloud dashboard.

Enriching your applications with IBM Cloud services

Figure 7-34. IBM Cloud Monitoring service

The IBM Cloud by default, collects and displays metrics for CPU usage, memory utilization, and network I/O for the Kubernetes Service. You can use the Monitoring service on IBM Cloud to automatically collect and measure key metrics from your environment and applications. No special instrumentation is required to collect metrics.

For example, you can use information provided by performance metrics to monitor how a service is running in the cloud, detect resource bottlenecks, and keep an eye on the service level agreement (SLA).

When you analyze performance data for a service, you can detect situations that can lead to a resource bottleneck and consequently affect your service SLA to your clients. By taking early action, you can prevent situations that can impact your business negatively.

The slide lists some of the benefits provided by IBM Cloud Monitoring:

- The IBM Cloud Monitoring service automatically collects metric data from IBM Cloud services, eliminating the need for agents. APIs make it easy to add custom metrics and to query your monitoring data.
- You can define rules to notify you of conditions requiring attention. The IBM Cloud Monitoring service offers an API that you can use to set performance thresholds, and to be notified when those thresholds are breached. Define alert rules for a single service instance or app instance, and alert rules that report on a set of instances. When an alert is triggered, get a notification through an e-mail, a PagerDuty event, a webhook notification, or any combination of the three.
- You can visualize how your apps and infrastructure are performing and consuming resources with user customizable dashboards. IBM Cloud Monitoring offers Grafana to quickly build and adapt your dashboard to your application needs.
- You can integrate your monitoring data into your applications and operations through the Monitoring service APIs. Use the APIs to add relevant application and business metrics to your Cloud monitoring data. You can also use the APIs to send metric data from outside the IBM Cloud into the Monitoring service.
- You can send metrics for your Cloud Foundry (CF) applications and Virtual Machines (VMs) to the Monitoring service. For more information on how to send metrics, see Sending metrics to the Monitoring service at https://cloud.ibm.com/docs/services/cloud-monitoring?topic=cloud-monitoring-send_retrieve_metrics_ov#send_retrieve_metrics_ov
- You can provision the Monitoring service through the IBM Cloud catalog.
- You can view and analyze metrics collected by the Monitoring service through the IBM Cloud dashboard.

References:

https://cloud.ibm.com/docs/services/cloud-monitoring?topic=cloud-monitoring-monitoring_ov

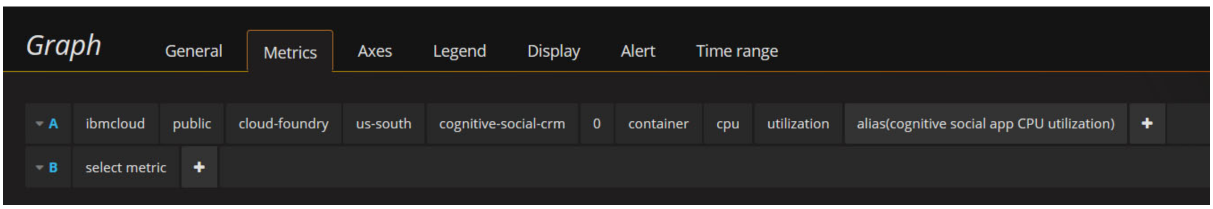
https://cloud.ibm.com/docs/services/cloud-monitoring?topic=cloud-monitoring-send_retrieve_metrics_ov#send_retrieve_metrics_ov

IBM Training

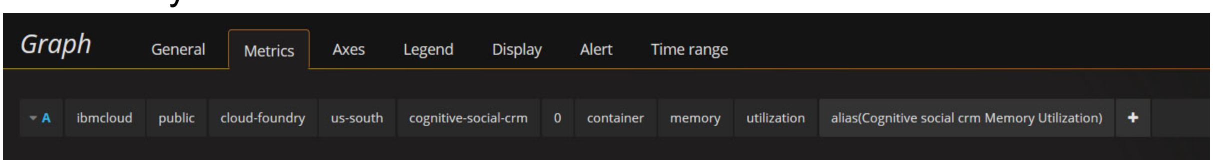
Integrating Monitoring with Cognitive Tweets Analyzer

- On the Monitoring service dashboard, create a Grafana dashboard to monitor the metrics that you want to display.
- On the Monitoring service dashboard, create a Grafana dashboard to monitor the metrics that you want to display.

- CPU utilization metric



- Memory utilization metric



Enriching your applications with IBM Cloud services

Figure 7-35. Integrating Monitoring with Cognitive Tweets Analyzer

On the Monitoring service dashboard, create a Grafana dashboard to monitor the metrics you want to display.

On the Grafana dashboard, define the query that filters the data to show the metrics that you selected.

For the Cognitive Tweets Analyzer, two graph metrics are defined, with two queries that monitor the percentage of CPU and memory utilization within the limits of the container in which the app is deployed.

References:

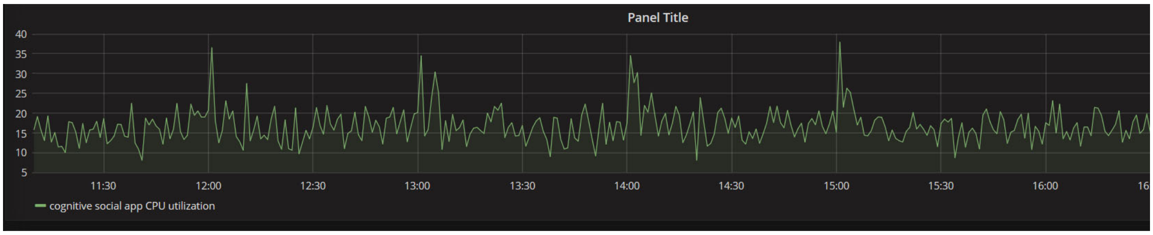
https://cloud.ibm.com/docs/services/cloud-monitoring?topic=cloud-monitoring-getting-started#getting-started

https://cloud.ibm.com/docs/services/cloud-monitoring/tutorials?topic=cloud-monitoring-cfapps_metrics#cfapps_step4

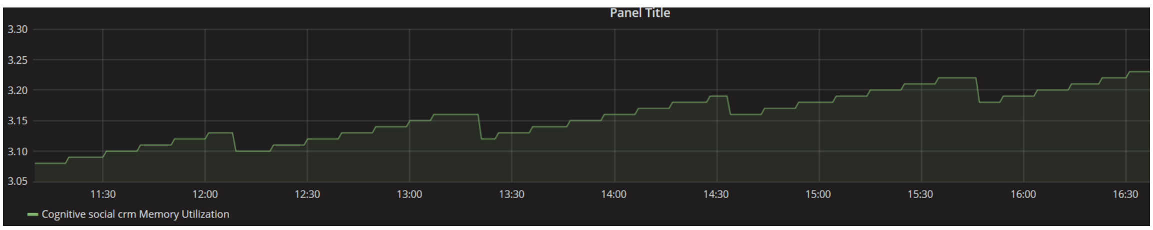
IBM Training

Integrating Monitoring with Cognitive Tweets Analyzer

- CPU utilization metric: Sample output



- Memory utilization metric: Sample output



Enriching your applications with IBM Cloud services

Figure 7-36. Integrating Monitoring with Cognitive Tweets Analyzer

The sample outputs from the Monitoring service that is integrated with Cognitive Tweets Analyzer, shows:

- One graph metric with the application's CPU utilization.
- One graph metric with the application's memory utilization.

7.4. Cognitive Tweets Analyzer: Demonstration

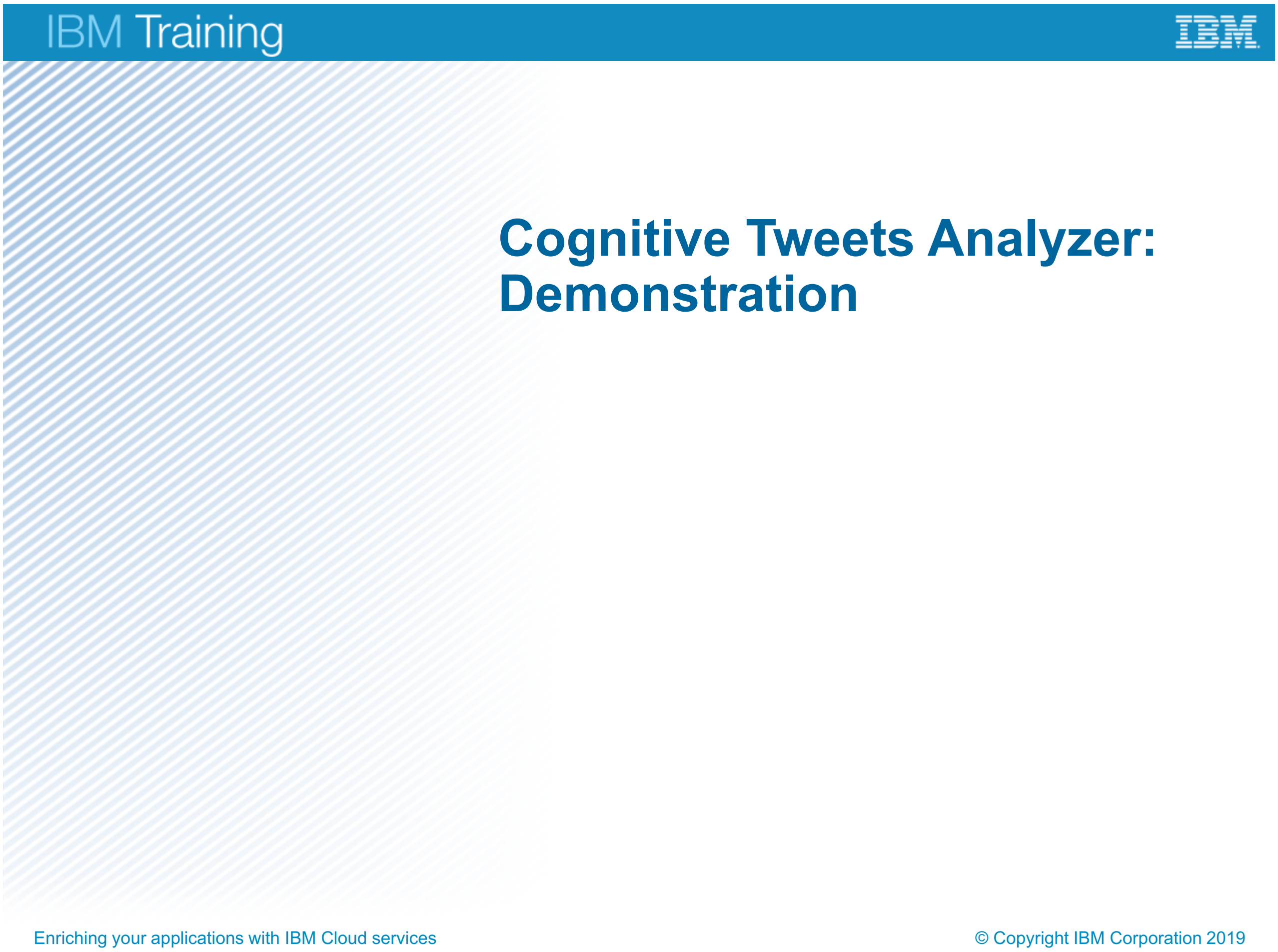


Figure 7-37. Cognitive Tweets Analyzer: Demonstration

IBM Training

Topics

- Business problem and requirements
- Solution architecture
- Components
- Cognitive Tweets Analyzer: Demonstration

Enriching your applications with IBM Cloud services

Figure 7-38. Topics

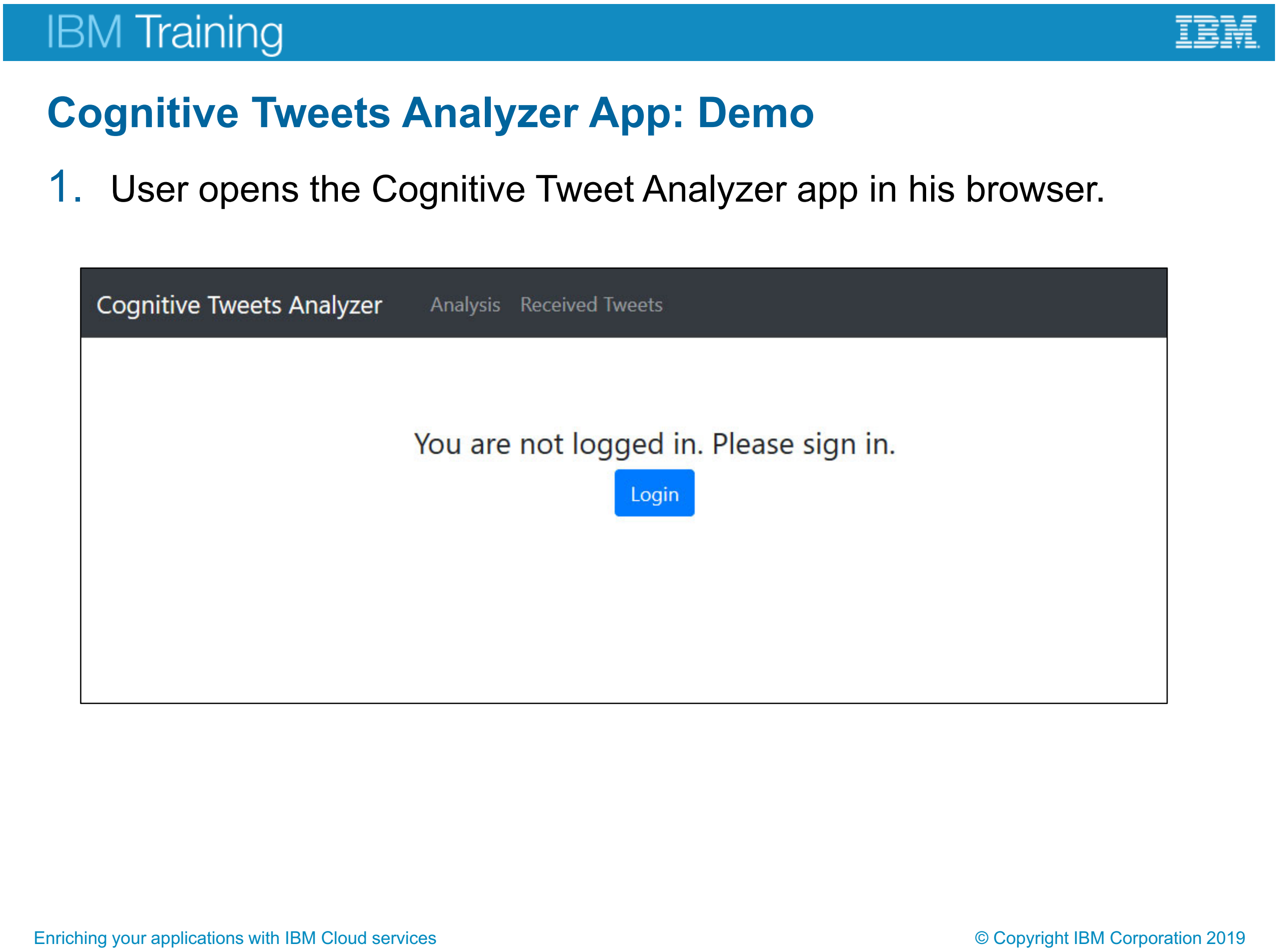


Figure 7-39. Cognitive Tweets Analyzer App: Demo

Cognitive Tweets Analyzer App:

Here is the github link to deploy the application. Follow the steps in the README to deploy the application locally or on IBM Cloud.

https://github.com/IBMRedbooks/Cloud-Application-Developer/tree/master/UseCase/cognitive-social-crm

---

Note

In the Node.js application configurations, you add the Twitter hashtag you wish to run the analysis on. For this demo, the hashtag being used is "*GameOfThrones*".

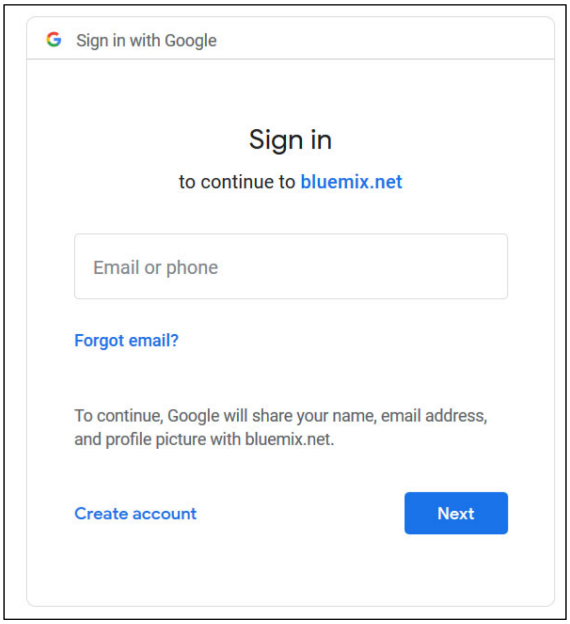
User scenario:

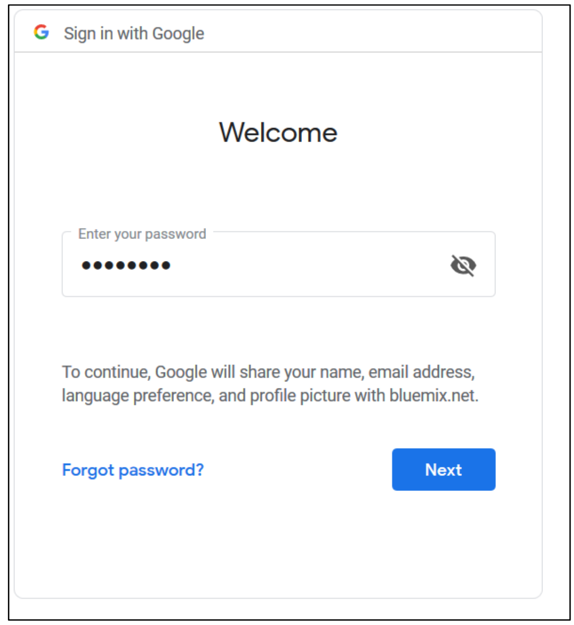
1. The user opens the application in a browser. The Log in page is displayed.

IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

2. User logs in with a Google account





Enriching your applications with IBM Cloud services

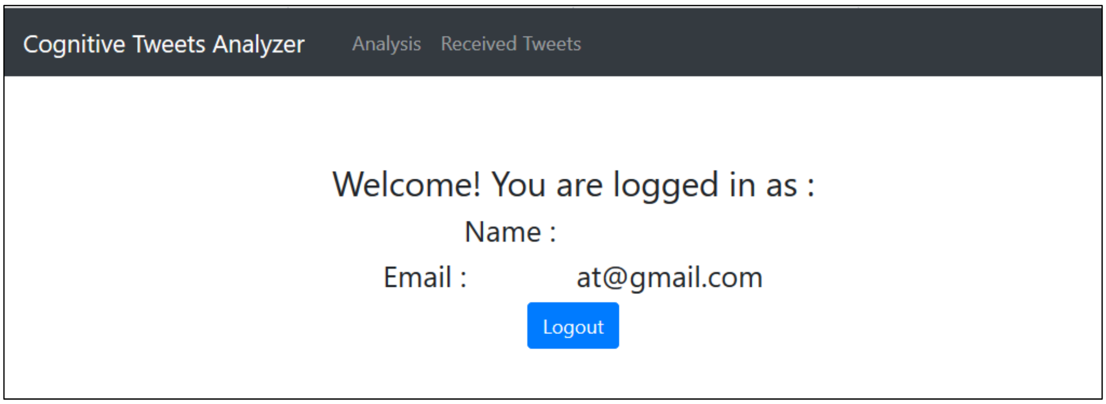
Figure 7-40. Cognitive Tweets Analyzer App: Demo (cont.)

2. The Google sign in windows are displayed. The user logs in with a Google account.

IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

3. After successful log in, the user’s name and email are displayed on the page.



Enriching your applications with IBM Cloud services

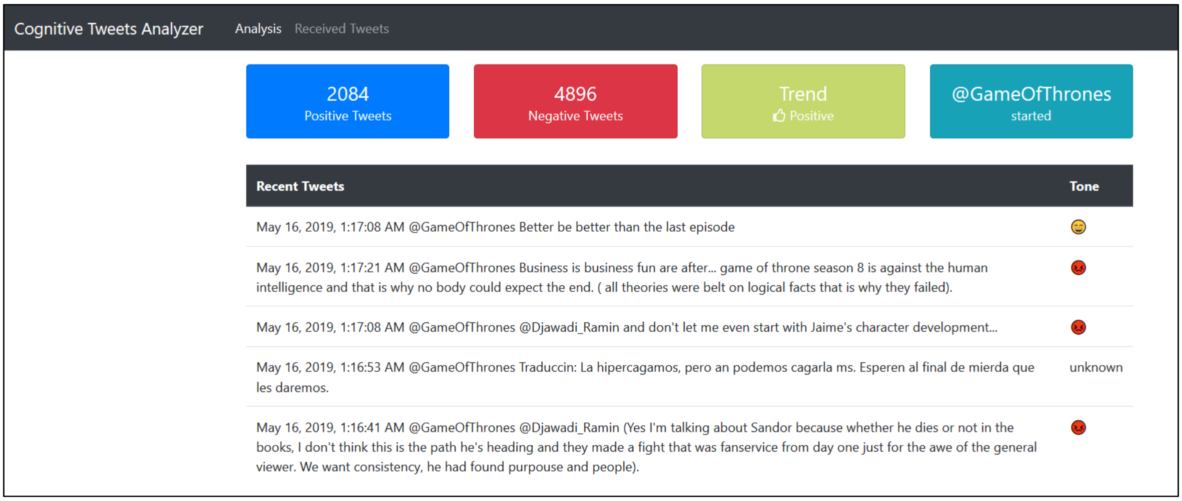
Figure 7-41. Cognitive Tweets Analyzer App: Demo (cont.)

3. After successful log in, the user’s name and email are displayed on the page.

IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

4. The user opens the Analysis tab, to visualize the analysis data on "GameOfThrones" hashtag.



Enriching your applications with IBM Cloud services

© Copyright IBM Corporation 2019

Figure 7-42. Cognitive Tweets Analyzer App: Demo (cont.)

4. When users open the *Analysis* tab, they can see the data visualizations and charts that show the emotional tones and sentiments on Twitter about the selected hashtag: GameOfThrones.
    - The analyzed data shows: number of positive tweets, number of negative tweets, average trend (positive or negative), recent tweet texts.

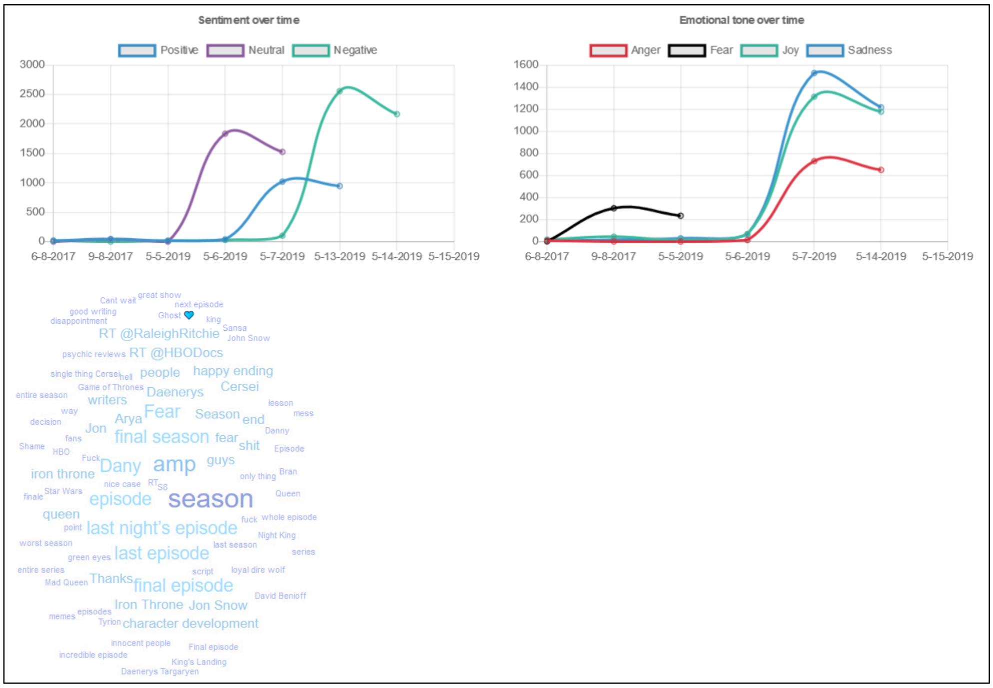
Note

The sample app is configured to analyze tweets for the "GameOfThrones" hashtag. The hashtag is added to the configuration files of the Node.js application.

IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

5. Analyze data graphs in the Analysis tab.



Enriching your applications with IBM Cloud services

Figure 7-43. Cognitive Tweets Analyzer App: Demo (cont.)

5. The Sentiment over time and Emotional tone over time graphs are also displayed along with a chart with the keywords that are detected in the tweets.

IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

6. The user opens the Tweets tab to display the list of tweets that are pushed to the application, which is sorted by posted date.

Received	Text	Tone
May 16, 2019, 1:17:08 AM	@GameOfThrones Better be better than the last episode	joy 😄
May 16, 2019, 1:17:21 AM	@GameOfThrones Business is business fun are after... game of throne season 8 is against the human intelligence and that is why no body could expect the end. (all theories were belt on logical facts that is why they failed).	joy 😄
May 16, 2019, 1:17:08 AM	@GameOfThrones @Djawadi_Ramin and don't let me even start with Jaime's character development...	sadness 😦
May 16, 2019, 1:16:53 AM	@GameOfThrones Traduccin: La hipercagamos, pero an podemos cagarla ms. Esperen al final de mierda que les daremos.	joy 😄
May 16, 2019, 1:16:41 AM	@GameOfThrones @Djawadi_Ramin (Yes I'm talking about Sandor because whether he dies or not in the books, I don't think this is the path he's heading and they made a fight that was fanservice from day one just for the awe of the general viewer. We want consistency, he had found purpouse and people).	sadness 😦
May 16, 2019, 1:16:36 AM	@GameOfThrones No need to watch. Jon will off Daenerys and go north. Arya will wander off hither and yon. Sansa will rule the North. And Bran will become the ruler of the seven kingdoms -- which is absurd and disappointing beyond belief.	sadness 😦
May 16, 2019, 1:15:06 AM	@GameOfThrones @Djawadi_Ramin The worst season ever	sadness 😦
May 16, 2019, 1:15:21 AM	@GameOfThrones This season was wack as hell, I hope to God someone was dreaming or something please! Make it make sense	anger 😡
May 16, 2019, 1:15:19 AM	@GameOfThrones @Djawadi_Ramin Redo this. Nonsensical character arcs, nonsensical character outcomes and even some nonsensical deaths and a fanservice fight made exclusively for the general viewer's awe without respecting his char	sadness 😦

Enriching your applications with IBM Cloud services

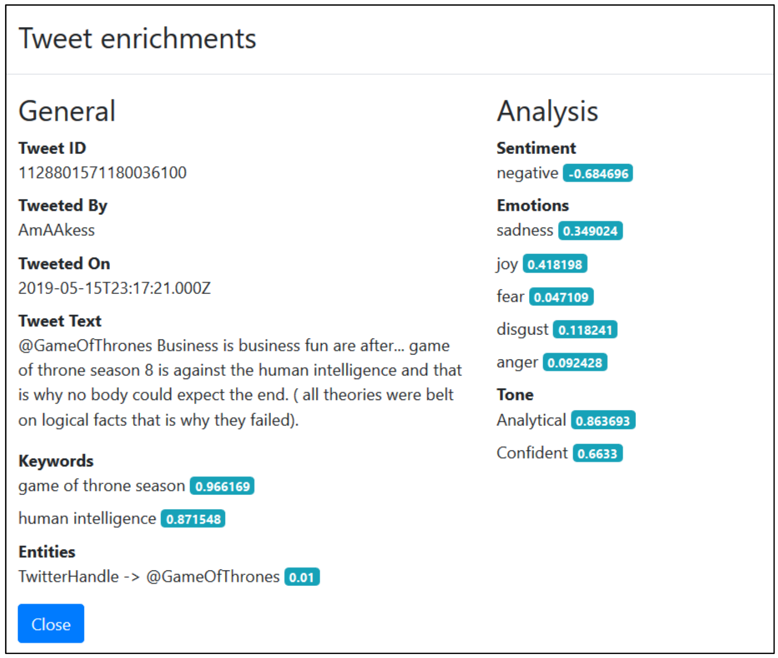
Figure 7-44. Cognitive Tweets Analyzer App: Demo (cont.)

6. The user opens the Tweets tab to display the list of tweets that are pushed to the application, which is sorted by posted date.

IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

7. When the user clicks one of the tweets that are listed in the Tweets tab, the sentiment and emotional tone analysis data is shown in the details view.



Enriching your applications with IBM Cloud services

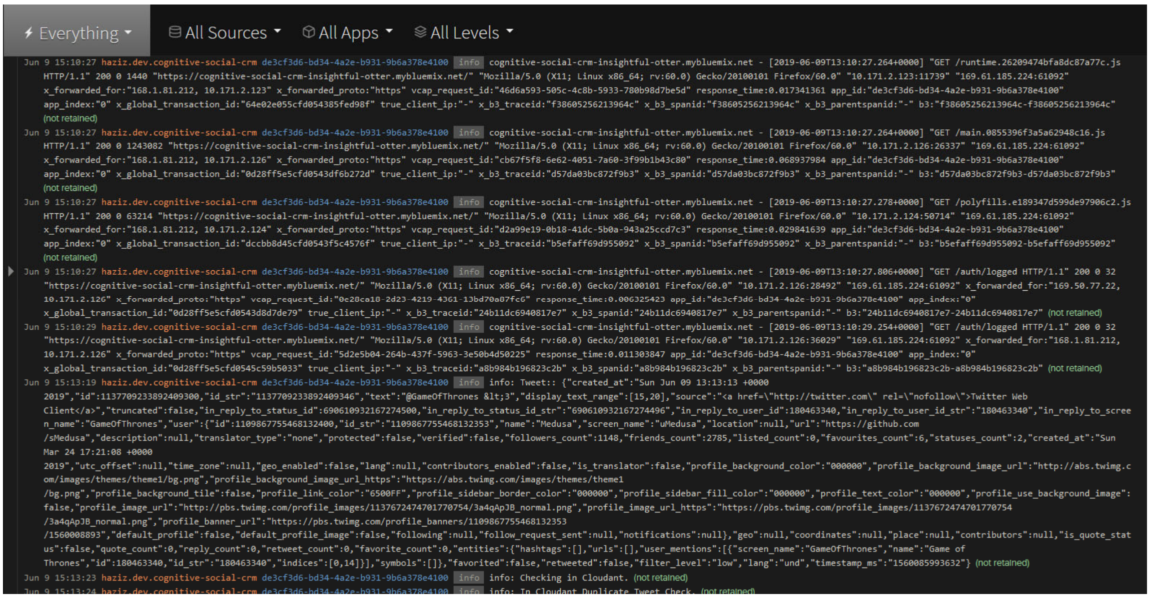
Figure 7-45. Cognitive Tweets Analyzer App: Demo (cont.)

7. When the user clicks one of the tweets, a dialogue opens showing the tweet details, and analysis data: sentiments scores, emptions scores, and emotional tone scores for that tweet.

IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

- Sample app logs displayed in the LogDNA dashboard



Enriching your applications with IBM Cloud services

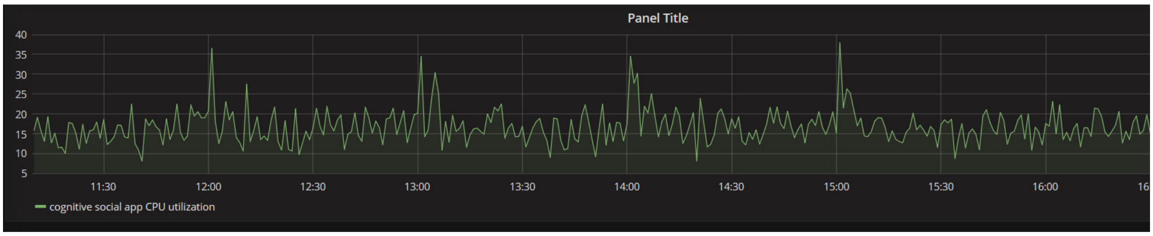
Figure 7-46. Cognitive Tweets Analyzer App: Demo (cont.)

This slide shows an example of logs displayed in the LogDNA dashboard.

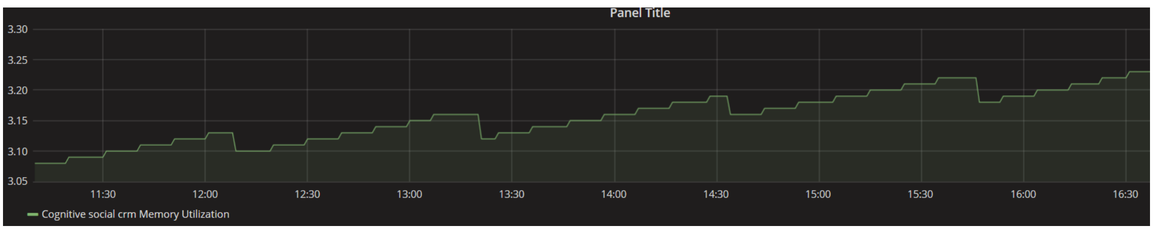
IBM Training

Cognitive Tweets Analyzer App: Demo (cont.)

- CPU utilization metric: Sample output



- Memory utilization metric: Sample output



Enriching your applications with IBM Cloud services

Figure 7-47. Cognitive Tweets Analyzer App: Demo (cont.)

The sample outputs from the Monitoring service that is integrated with Cognitive Tweets Analyzer, shows:

- One graph metric with the application’s CPU utilization.
- One graph metric with the application’s memory utilization.

IBM Training

Unit summary

- Define your business problem and goals
- Identify functional and non-functional requirements
- Select the technical components that best fit your solution
- Design a simple architecture for a cloud application
- Identify services in the IBM Cloud catalog that you can use to enrich your cloud apps
- Describe App ID, Watson Natural Language Understanding, Watson Tone Analyzer, LogDNA, and IBM Cloud Monitoring services and their integration in the sample use case

Enriching your applications with IBM Cloud services

Figure 7-48. Unit summary

IBM Training

Review questions

1. Functional requirements:
   A. Define system attributes such as security, reliability, performance, maintainability, scalability, availability, and usability.
   B. Describe the functions and services that should be provided by the system at a component level.
   C. Define the basic pattern of a solution
   D. Provide a roadmap to build, extend, and deploy an application.
2. The Third Party LogDNA service can be integrated with Cloud Foundry apps on IBM Cloud through:
   A. Kubernetes clusters
   B. User-provided service
   C. Swagger framework
   D. REST APIs
3. True or False: The Cloudant database is compatible with Apache CouchDB

Enriching your applications with IBM Cloud services

Figure 7-49. Review questions

IBM Training

Review questions

4. Non-functional requirements:
   A. Define system attributes such as security, reliability, performance, maintainability, scalability, availability, and usability.
   B. Describe the functions and services that should be provided by the system at a component level.
   C. Define the basic pattern of a solution
   D. Provide a roadmap to build, extend, and deploy an application.
5. App ID supports the following identity providers:
   A. Facebook, Google+, and SAML 2.0 Federation, Cloud Directory, Custom
   B. LinkedIn, Facebook, Google+, and SAML
   C. AWS, Facebook, Google+, SAML, Cloud Directory
   D. Yahoo, Facebook, Google+, Instagram, PayPal, LinkedIn

Enriching your applications with IBM Cloud services

Figure 7-50. Review questions

IBM Training

Review questions

6. Watson Tone Analyzer can detect in the input text:
   A. Sentiment, entities, relations
   B. Emotion, social tendencies, language/writing styles
   C. Keywords, categories, language/writing styles
7. One of the following services can be used to define rules and alerts that notify you of conditions that require attention:
   A. IBM Cloud Log Analysis with LogDNA
   B. IBM Cloud Monitoring
   C. App ID
   D. Watson Natural Language Understanding
8. The main benefit provided by Cloudant to store tweets that are returned by the Twitter API is:
   A. It is a relational database
   B. It supports XML documents
   C. It has well defined schemas
   D. It is a native JSON database.

Enriching your applications with IBM Cloud services

Figure 7-51. Review questions

IBM Training

Review answers

1. Functional requirements:
   - A. Define system attributes such as security, reliability, performance, maintainability, scalability, availability, and usability.
   - **B. Describe the functions and services that should be provided by the system at a component level.**
   - C. Define the basic pattern of a solution
   - D. Provide a roadmap to build, extend, and deploy an application.
2. The Third Party LogDNA service can be integrated with Cloud Foundry apps on IBM Cloud through:
   - A. Kubernetes clusters
   - **B. User-provided service**
   - C. Swagger framework
   - D. REST APIs
3. **True** or False: The Cloudant database is compatible with Apache CouchDB

Enriching your applications with IBM Cloud services

Figure 7-52. Review answers

IBM Training

Review answers

4. Non-functional requirements:
   - A. **Define system attributes such as security, reliability, performance, maintainability, scalability, availability, and usability.**
   - B. Describe the functions and services that should be provided by the system at a component level.
   - C. Define the basic pattern of a solution
   - D. Provide a roadmap to build, extend, and deploy an application.
5. App ID supports the following identity providers:
   - A. **Facebook, Google+, and SAML 2.0 Federation, Cloud Directory, Custom**
   - B. LinkedIn, Facebook, Google+, and SAML
   - C. AWS, Facebook, Google+, SAML, Cloud Directory
   - D. Yahoo, Facebook, Google+, Instagram, PayPal, LinkedIn

Enriching your applications with IBM Cloud services

Figure 7-53. Review answers

IBM Training

Review answers

6. Watson Tone Analyzer can detect in the input text:
   - A. Sentiment, entities, relations
   - **B. <u>Emotion, social tendencies, language/writing styles</u>**
   - C. Keywords, categories, language/writing styles
7. One of the following services can be used to define rules and alerts that notify you of conditions that require attention:
   - A. IBM Cloud Log Analysis with LogDNA
   - **B. <u>IBM Cloud Monitoring</u>**
   - C. App ID
   - D. Watson Natural Language Understanding
8. The main benefit provided by Cloudant to store tweets that are returned by the Twitter API is:
   - A. It is a relational database
   - B. It supports XML documents
   - C. It has well defined schemas
   - **D. <u>It is a native JSON database</u>**

Enriching your applications with IBM Cloud services

Figure 7-54. Review answers

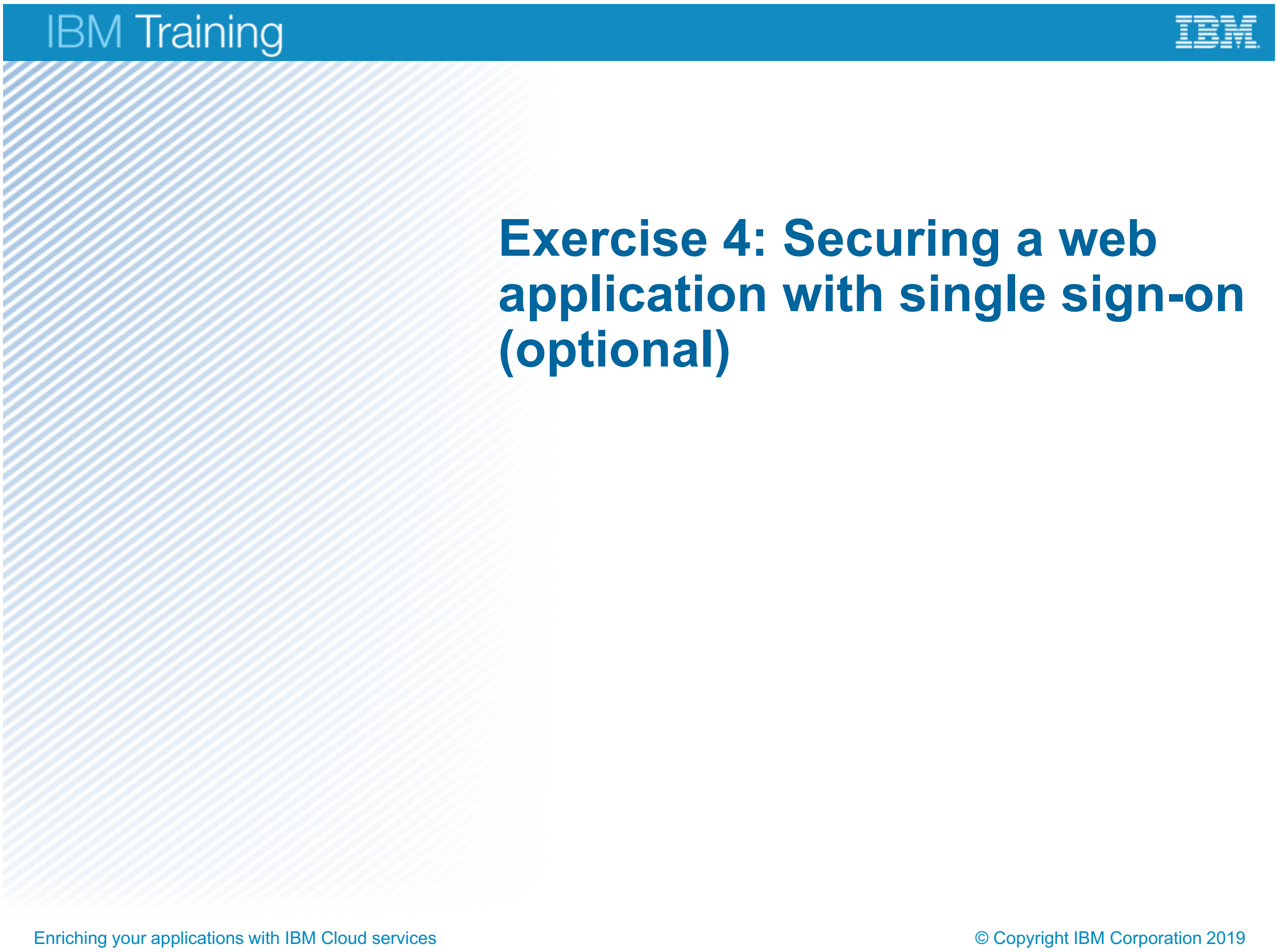


Figure 7-55. Exercise 4: Securing a web application with single sign-on (optional)

IBM Training

Exercise objectives

- In this exercise, you secure an application by using the App ID service for single sign-on by authenticating your application through trusted server providers.
- After completing this exercise, you should be able to:
  - Create an App ID service.
  - Bind the App ID service to an application to add single sign-on capability to the application.
  - Describe different configurations in the App ID service.

Enriching your applications with IBM Cloud services © Copyright IBM Corporation 2019

Figure 7-56. Exercise objectives

Unit 8. Developing containerized applications on Kubernetes

Estimated time

02:00

Overview

This unit introduces containers and containers orchestration. It provides an overview of the Kubernetes platform and describes basic concepts such as Kubernetes architecture, Kubernetes objects, and management of Kubernetes objects.

IBM Training

Unit objectives

- Explain containers and the difference between containers and virtual machines (VMs).
- Describe container orchestration (Kubernetes).
- List the key capabilities of Kubernetes.
- Articulate the importance of using Kubernetes to prevent vendor lock-in.
- Describe the Kubernetes building blocks: Pod, Deployment, and Service.
- Scale and auto-scale your Deployment for high availability.

Developing containerized applications on Kubernetes

Figure 8-1. Unit objectives

8.1. Containers

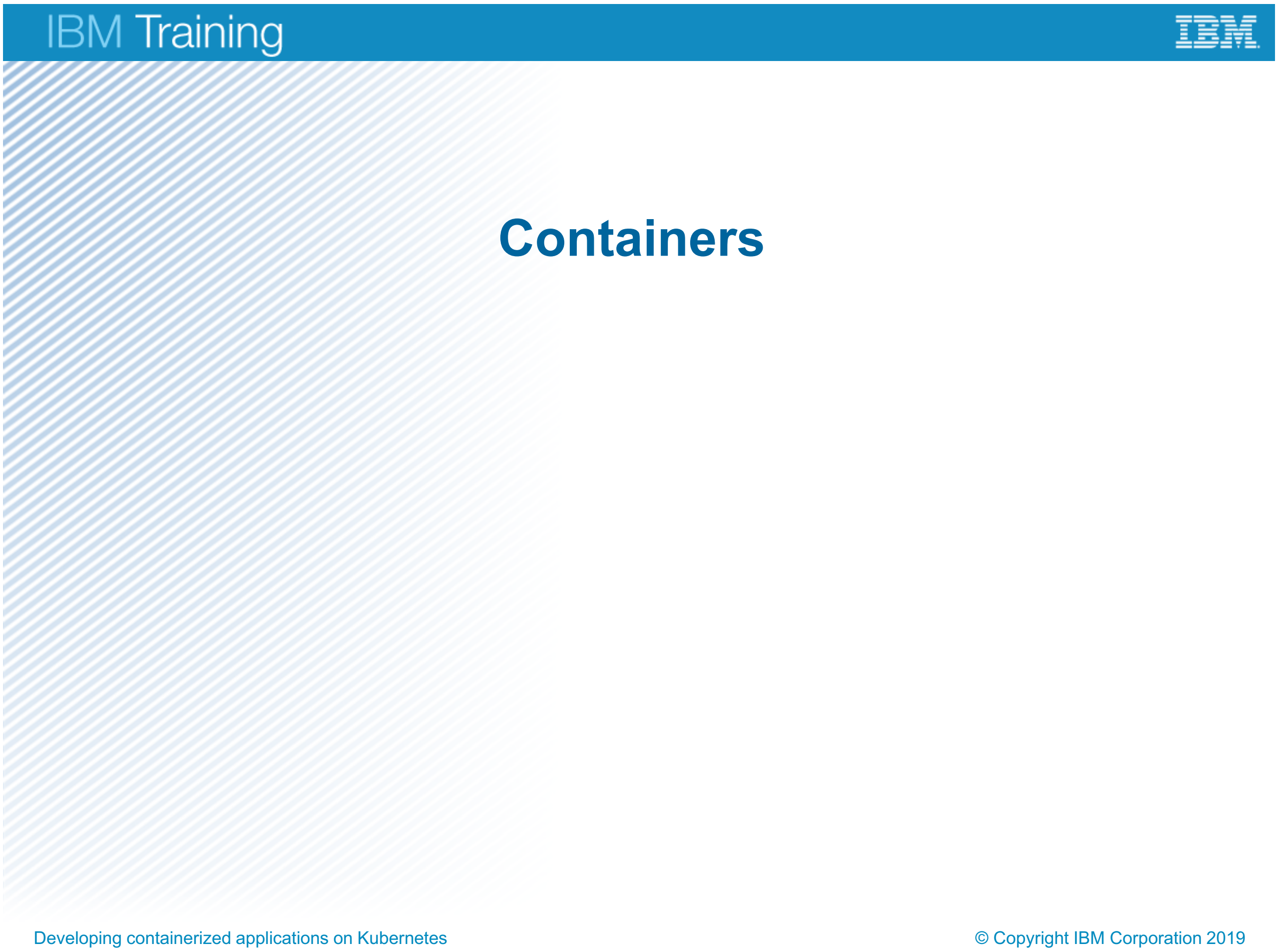


Figure 8-2. Containers

IBM Training

Topics

- Containers
- Container orchestration
- Introducing Kubernetes
- Kubernetes architecture
- Kubernetes objects
- Next steps

Developing containerized applications on Kubernetes

Figure 8-3. Topics

IBM Training

Containers

- Containers provide operating system (OS)-level virtualization.
- Containers are isolated from each other and package the application, code, tools, and libraries together.
- Containers run from a single OS kernel.

Linux + Container Engine

Figure 8-4. Containers

With containers, you can run securely isolated applications with quotas on system resources. Containers started as an individual feature that was delivered with the Linux kernel. Docker started with making containers easy to use and developers quickly latched onto that idea. Containers also sparked an interest in the microservice architecture, which is a design pattern for developing applications in which complex applications are pared down into smaller, composable pieces that work together.

References:

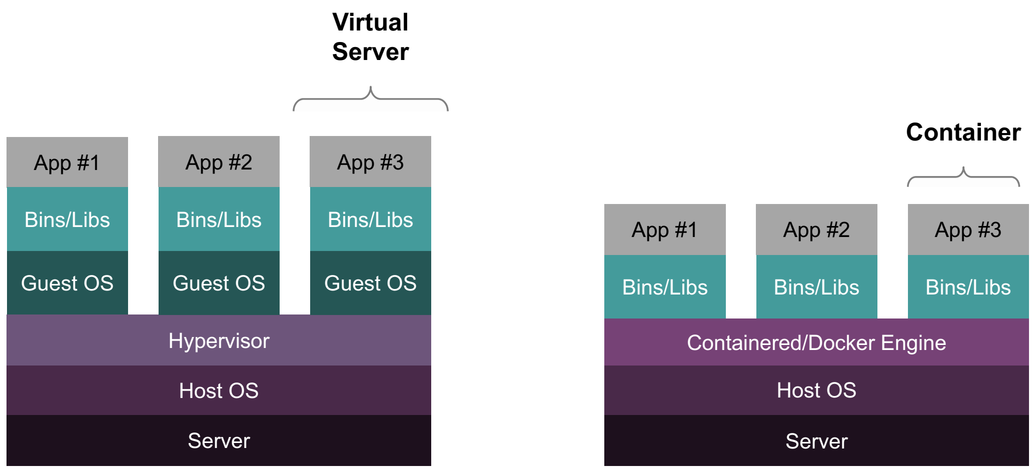
https://www.ibm.com/cloud/garage/practices/run/tool_ibm_container/

https://www.ibm.com/support/knowledgecenter/en/linuxonibm/com.ibm.linux.z.ldvd/ldvd_r_plan_container_vm.html

IBM Training

Containers versus virtual machines and the benefits of containers

- Containers are isolated, but share the kernel.
- Containers are isolated by hiding information (such as namespaces).
- Containers offer speed, agility, and portability.



Developing containerized applications on Kubernetes

Figure 8-5. Containers versus virtual machines and the benefits of containers

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 tight dependency on physical hardware, which allows for more efficient use of computing resources.

Virtual servers, which are also known as VMs, are created by using a hypervisor. The hypervisor virtualizes the physical hardware to create a software-defined computer that runs its own OS. Unless special software is installed for cloud management, an OS that is running on a VM is unaware that it is running on VM.

Unlike a VM that provides hardware virtualization, a container provides OS-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 OS 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.

Key difference between containers and VMs

- VMs

  Before containers, most infrastructure ran on hypervisors that managed multiple virtualized OSes. This arrangement allowed isolation of applications at a higher level than what was provided by the OS. These virtualized OSes see what looks like their own exclusive hardware. However, this also means that each of these virtual OSes is replicating an entire OS, taking up disk space.

- Containers

  Containers provide isolation similar to VMs, except that it is provided by the OS and at the process level. Each container is a process or group of processes that run in isolation. Typical containers explicitly run only a single process because they have no need for the standard system services. What they usually need can be provided by system calls to the base OS kernel.

Benefits of containers

Traditional applications run on native hardware. A single app does not typically use the full resources of a single machine. Most organizations try to run multiple apps on a single machine to avoid wasting resources. You could run multiple copies of the same app, but to provide isolation, you can use VMs to run multiple app instances on the same hardware. These VMs have full OS stacks that make them relatively large and inefficient due to duplication, both at run time and on disk.

Containers allow you to share the host OS hypervisor. This reduces duplication while still providing the isolation.

Containers also allow you to drop unneeded files such as system libraries and binary files to save space and reduce your attack surface.

References:

https://github.com/IBM/container-service-getting-started-wt

https://www.ibm.com/cloud/garage/practices/run/tool_ibm_container/

IBM Training

Dockerfile: Building a container?

A Dockerfile is a text document that contains collections of commands and instructions that are automatically run in sequence in the Docker environment for building a new Docker image.

```
# Simple nginx web server image
FROM nginx:alpine

# Metadata
LABEL maintainer "Mihai Criveti"

# Serving static HTML - copy com
COPY html /usr/share/nginx/html
```

The `docker build` command builds an image from a Dockerfile.

Developing containerized applications on Kubernetes

Figure 8-6. Dockerfile: Building a container?

Details about the Dockerfile

- The first line starts with # and it is a comment. You can add comments to the Dockerfile with the help of the # mark.
- The `FROM` **nginx:alpine** line tells Docker that the image is based on the nginx alpine version image.
- You can add labels to your image to help organize images by project and record licensing information, to aid in automation, or for other reasons. For each label, add a line beginning with `LABEL` and with one or more key-value pairs.
- MAINTAINER contains the name of the maintainer of the image.
- **COPY** line. When you run **COPY**, it copies the files from the local source to the image. In this case, the files in the html directory are copied to the location /usr/share/nginx/html within the image.

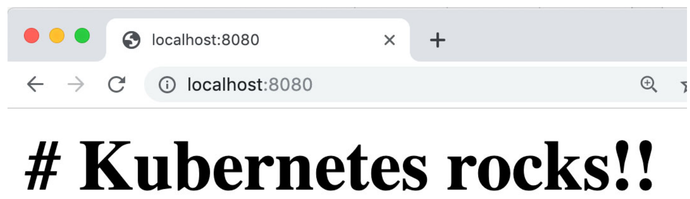
IBM Training

Building a container image from a Dockerfile

The following code shows how to build a simple webserver image and test it locally:

```
$ vi Dockerfile
FROM nginx:alpine
COPY index.html /usr/share/nginx/html
$ vi index.html
<html> <body>
<h1> # Kubernetes rocks!! </h1>
</body> </html>
$ docker build -t cloud-course/helloweb:v1 .
$ docker run -p 8080:80 -d cloud-course/helloweb:v1
```

This graphic shows the result of the code.



Developing containerized applications on Kubernetes

Figure 8-7. Building a container image from a Dockerfile

1. The first command creates the Dockerfile that is used to build the image.
2. The second command creates the index.html file that displays "`# Kubernetes rocks!!`".
3. The third command builds the image by using the **docker build** command, and gives it the tag "cloud-course/helloweb:v1".
4. The fourth command runs the image and forwards the port from port 8080 to port 80. The **run** command creates a container out of the image and then runs it.

To run this code, you need to have Docker locally. In this unit, you learn how to build Docker containers on IBM Cloud without having Docker locally. You will be able to build your images directly on the IBM Cloud Container Registry service.

8.2. Container orchestration

IBM Training

Container orchestration

Developing containerized applications on Kubernetes

Figure 8-8. Container orchestration

IBM Training

Topics

- Containers
- Container orchestration
- Introducing Kubernetes
- Kubernetes architecture
- Kubernetes objects
- Next steps

Developing containerized applications on Kubernetes

Figure 8-9. Topics

IBM Training

Container orchestration

You can manage the deployment, placement, and lifecycle of containers at scale.

Container orchestration performs the following functions:

- Cluster management
- Self-healing
- Replication
- Service discovery
- Scheduling
- Scaling and workload auto-scaling
- Persistent storage
- Blue-green deployments
- Same API
- Managing stateful and stateless applications.

Common container orchestration platforms include *Kubernetes*, *Docker Swarm*, and *Apache Mesos*. In this course, the focus is on Kubernetes.

Developing containerized applications on Kubernetes

Figure 8-10. Container orchestration

Container orchestration performs the following functions:

- Cluster management: Federate hosts and manage them.
- Self-healing: Detect and replace unhealthy container Pods and hosts. Attempt to put the cluster back to the wanted state.
- Replication: Ensure that the wanted number of Pod replicas is running.
- Service discovery: Locate and distribute client requests across running containers.
- Scheduling: Distribute containers across the worker nodes.
- Scaling: Adding or removing containers to match workload.
- Persistent Storage: Manage persistent storage.
- Blue-green deployments: Blue-green deployments use two identical environments. While clients are using one active environment, you can update the other environment without interrupting the active environment
- Same API: The same API crosses multiple clouds or infrastructure providers so that you can avoid vendor lock-in and deploy your application across a consistent platform.
- Managing stateful and stateless applications: Both stateful and stateless applications can be managed by container orchestration.

References:

https://www.ibm.com/cloud/garage/content/course/kubernetes-101/1

https://developer.ibm.com/solutions/container-orchestration-and-deployment/

https://www.ibm.com/blogs/bluemix/2019/04/container-orchestration-explained/

8.3. Introducing Kubernetes

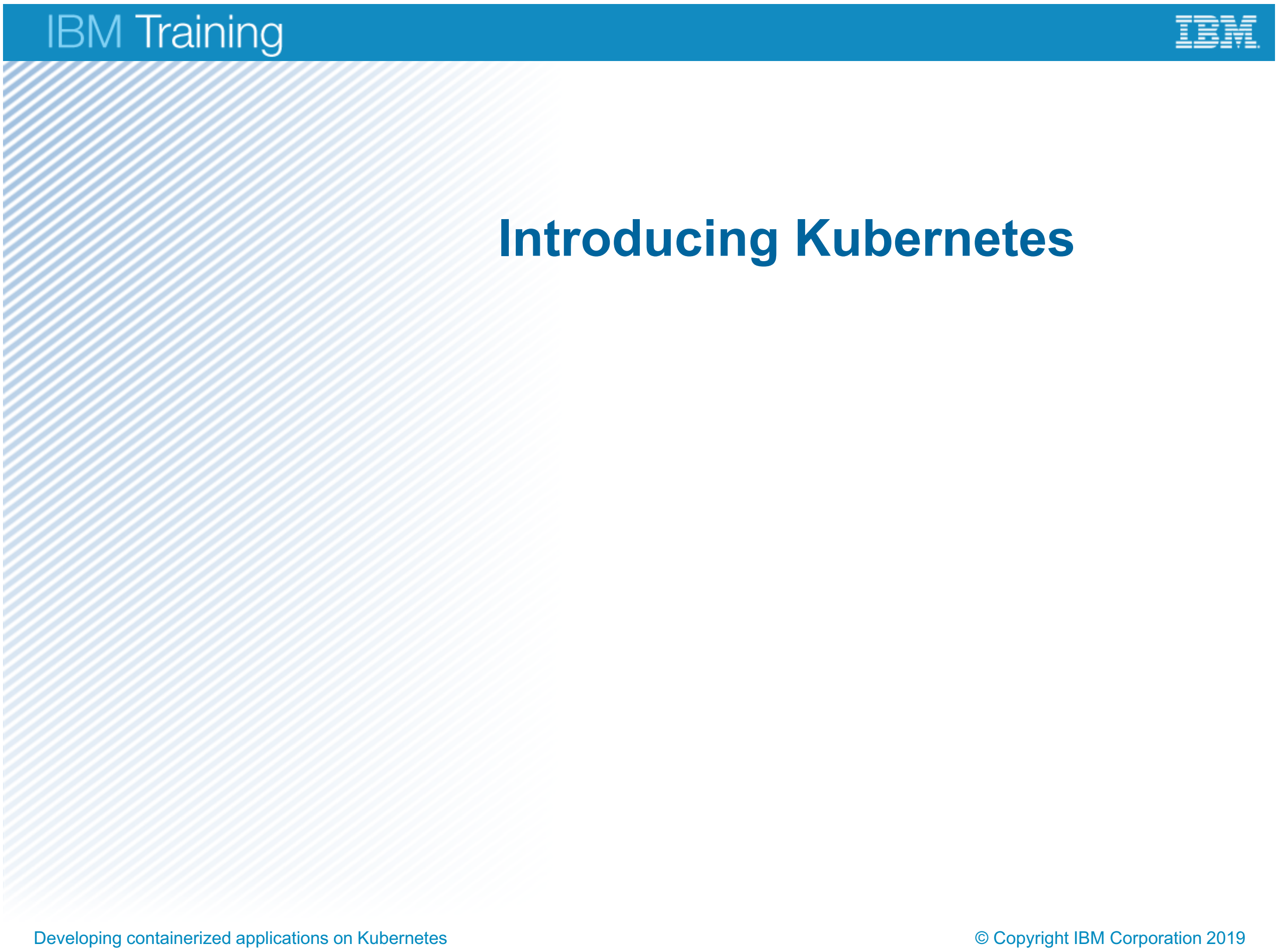


Figure 8-11. Introducing Kubernetes

https://www.ibm.com/cloud/garage/content/course/kubernetes-101/2

IBM Training

Topics

- Containers
- Container orchestration
- Introducing Kubernetes
- Kubernetes architecture
- Kubernetes objects
- Next steps

Developing containerized applications on Kubernetes

Figure 8-12. Topics

IBM Training

What is Kubernetes

- According to kubernetes.io, Kubernetes has the following features:
  - It is a portable and extensible open source platform for managing and scaling containerized workloads and services.
  - It facilitates both declarative configurations and automation.
  - It has a large and rapidly growing infrastructure.
  - Kubernetes services, support, and tools are widely available.
- Kubernetes is an open source project that is hosted by the *Cloud Native Computing Foundation* (CNCF).
- Kubernetes prevents vendor lock-in because it is offered by major cloud providers, such as IBM Cloud Kubernetes Service, IBM Cloud Private, Red Hat OpenShift, Azure Kubernetes Service, and Google Kubernetes Engine.

Developing containerized applications on Kubernetes

Figure 8-13. What is Kubernetes

Kubernetes comes from the Greek word for "helmsman" (of a ship) and it is commonly abbreviated "K8s".

Kubernetes is a container and microservices platform that enables portability across infrastructure providers. It has a container orchestration system for automating application deployment, automating scaling, and managing the operations of containers across a cluster of worker nodes.

You can use Kubernetes to provision, manage, and scale containerized applications on a cluster of worker nodes, which can be physical machines or VMs.

The key paradigm in Kubernetes is the *declarative model*: You determine the necessary state, and Kubernetes makes it happen.

Kubernetes is an open source project that is hosted by the *Cloud Native Computing Foundation* (CNCF). The CNCF is a vendor-neutral governance group and a subfoundation of the *Linux Foundation*. It is one of the largest open source projects, with over 2000 individual contributors, 75,000 commits, and 50,000 stars on GitHub.

Kubernetes prevents vendor lock-in because it is offered by major cloud providers, such as IBM Cloud Kubernetes Service, IBM Cloud Private, Red Hat OpenShift, Azure Kubernetes Service, and Google Kubernetes Engine, which gives a client the flexibility to port their workload from a vendor to another one.

References:

https://kubernetes.io/

8.4. Kubernetes architecture

IBM Training

Kubernetes architecture

Developing containerized applications on Kubernetes

Figure 8-14. Kubernetes architecture

IBM Training

Topics

- Containers
- Container orchestration
- Introducing Kubernetes
- Kubernetes architecture
- Kubernetes objects
- Next steps

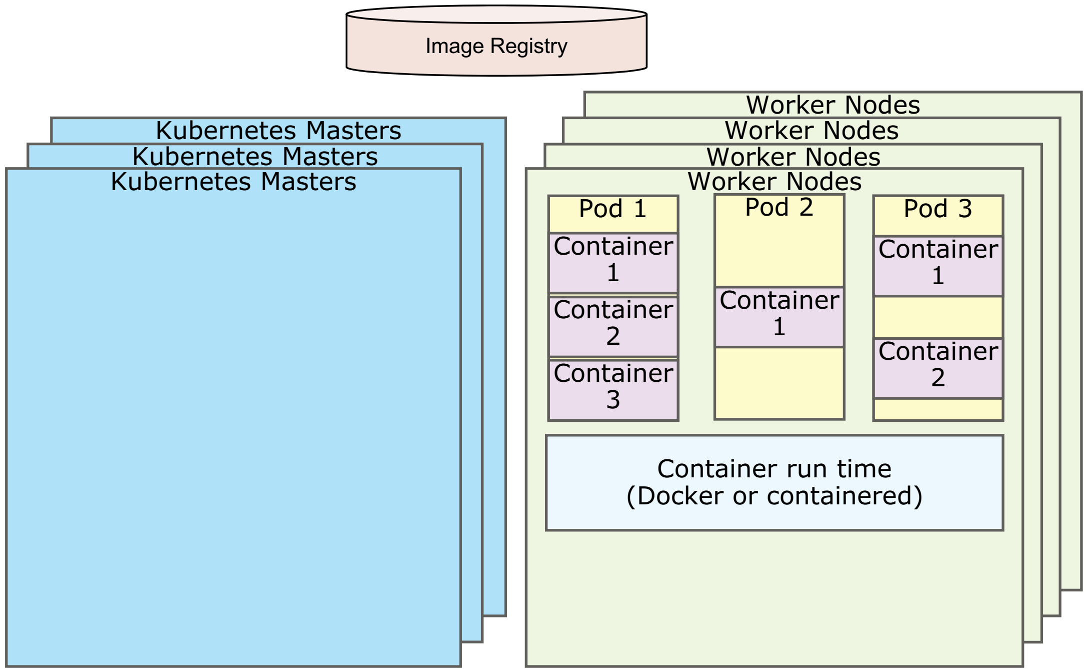
Developing containerized applications on Kubernetes

Figure 8-15. Topics

IBM Training

Kubernetes architecture

Multiple master and worker nodes provide scaling and high availability.



Developing containerized applications on Kubernetes

Figure 8-16. Kubernetes architecture

These slide shows an example of a Kubernetes cluster:

- Master nodes: Make global decisions about the cluster (for example, scheduling), and detecting and responding to cluster events (starting a new Pod when a replication controller's replicas field is unsatisfied). They provide the cluster's control window.
- Worker nodes: Maintain running Pods and provide the Kubernetes runtime environment.
- Pod: The smallest object model that you can create and run. A Pod typically represents a process in your cluster.
- Image registry: Stores the images.

8.5. Kubernetes objects

IBM Training

Kubernetes objects

Developing containerized applications on Kubernetes

Figure 8-17. Kubernetes objects

IBM Training

Topics

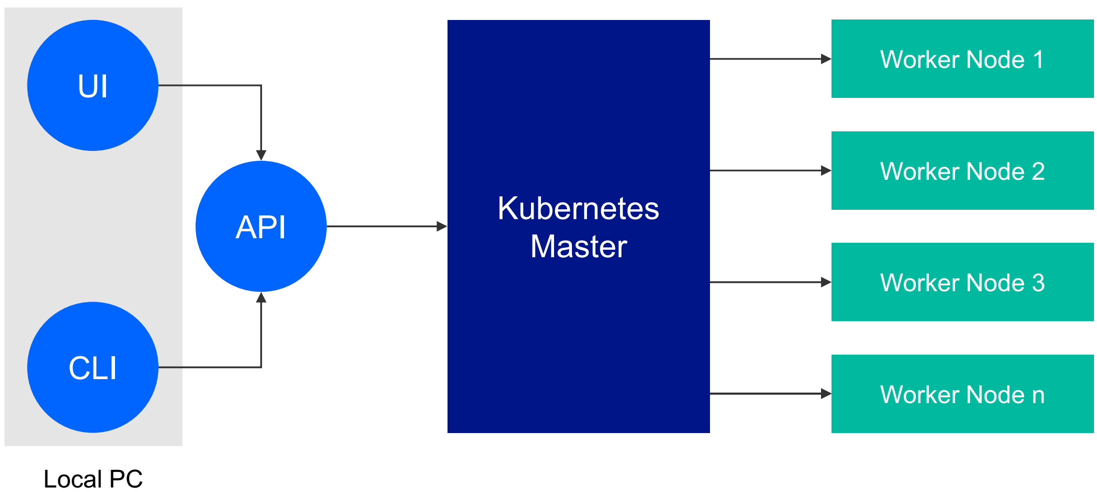
- Containers
- Container orchestration
- Introducing Kubernetes
- Kubernetes architecture
- Kubernetes objects
- Next steps

Developing containerized applications on Kubernetes

Figure 8-18. Topics

IBM Training

Interaction with a Kubernetes cluster



Developing containerized applications on Kubernetes

Figure 8-19. Interaction with a Kubernetes cluster

Kubernetes provides a client interface by using the following interfaces:

- Web UI: You can manage your cluster by using the Kubernetes Dashboard web user interface.
- Command-line interface (CLI): You can manage your cluster by using the **kubectl** CLI. Kubectl commands allow you to manage your apps and manage cluster and cluster resources by modifying the model in the data store. You directly manipulate resources through YAML, which is a human-readable serialization language, as shown in the following command:

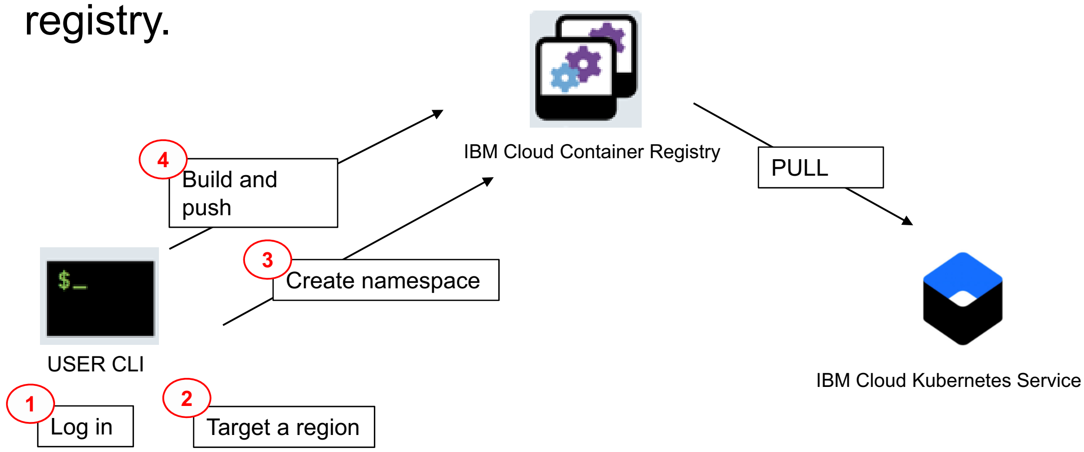
```
$ kubectl (create|get|apply|delete) -f myResource.yaml
```

Both deal with Kubernetes cluster through APIs that are exposed by the Kubernetes master nodes.

IBM Training

Building a container and storing it in the container registry

- A container registry stores and distributes container images.
- Examples for Container Registry are IBM Cloud Container Registry, and Docker Hub.
- The flow shows the steps of building an image directly into the IBM Cloud Container Registry as an example container registry.



Developing containerized applications on Kubernetes

Figure 8-20. Building a container and storing it in the container registry

You can build a Docker image directly in the IBM Cloud Container Registry or create your own Docker image on your local computer and upload (push) it to your namespace in IBM Cloud Container Registry.

A Docker image is the basis for every container that you create. An image is created from a Dockerfile, which is a file that contains instructions to build the image. A Dockerfile might reference build artifacts in its instructions that are stored separately, such as an app, the app's configuration, and its dependencies.

If you want to take advantage of IBM Cloud compute resources or Docker is not installed on your workstation, build your image directly in IBM Cloud. If you must access resources in your build that are on servers that are behind your firewall, build your image locally.

Here are the commands for creating images in a container registry.

1. `ibmcloud login`
2. `ibmcloud target –r <region>`
3. `ibmcloud cr namespace-add <namespace name>`
4. `ibmcloud cr build -t <region domain>/<namespace>/<application name>:<tag> <directory>`

References:

https://cloud.ibm.com/docs/services/Registry?topic=registry-registry_overview

https://cloud.ibm.com/docs/services/Registry?topic=registry-registry_images_

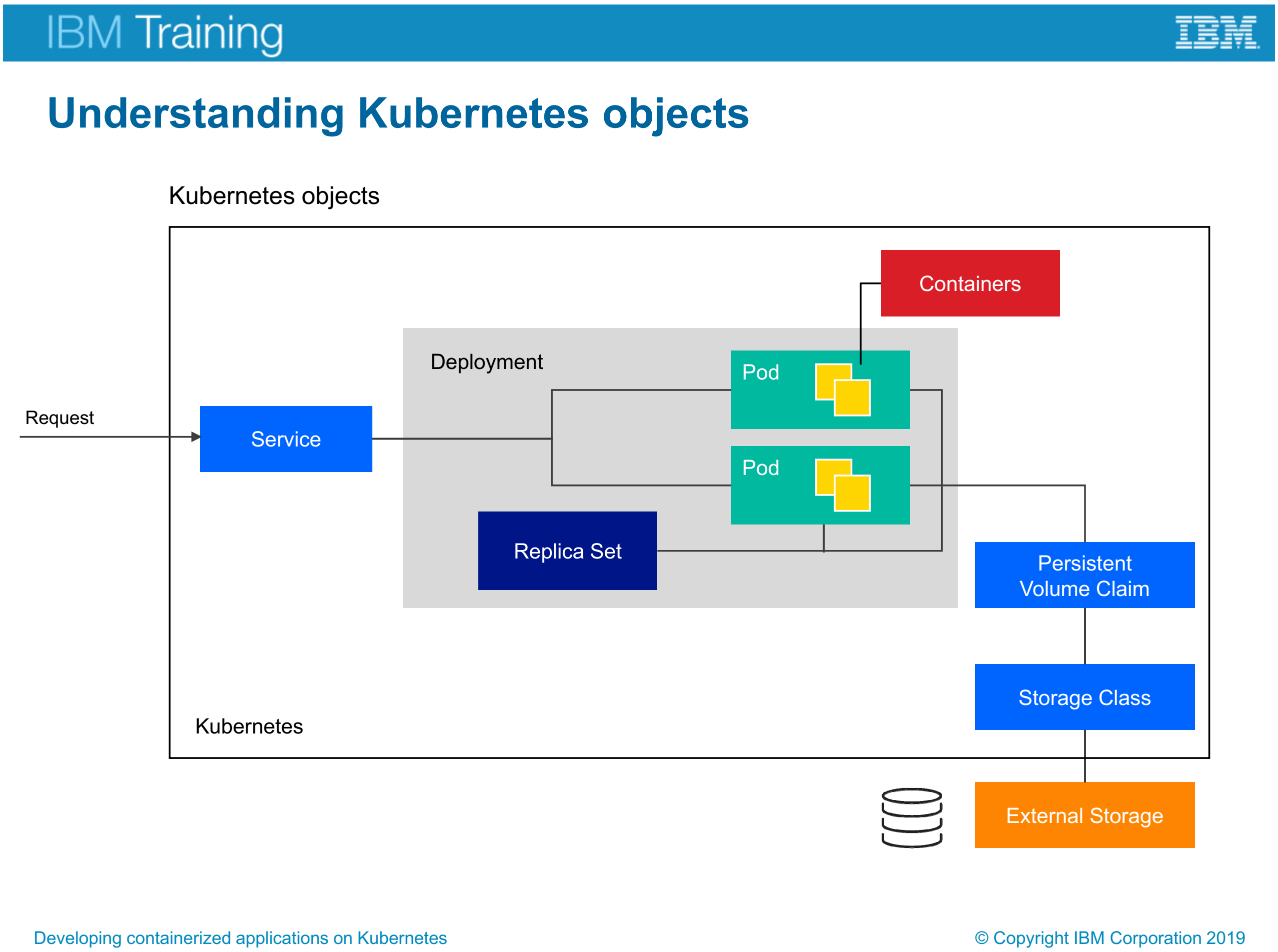


Figure 8-21. Understanding Kubernetes objects

Kubernetes objects are persistent entities in Kubernetes that are used to represent the state of your cluster. They can describe what *containerized applications* are running and on which nodes, the *resources* that are available for those applications, and the *policies* that dictate their behavior.

To create, modify, or delete Kubernetes objects, you must use the Kubernetes API (or **kubectl**, which is a command-line tool that makes the necessary API calls for you).

The basic Kubernetes objects include the following ones:

- Pod: The smallest unit of Deployment that can be managed by Kubernetes.
- Service: Describes a logical set of Pods and the policies to access it.

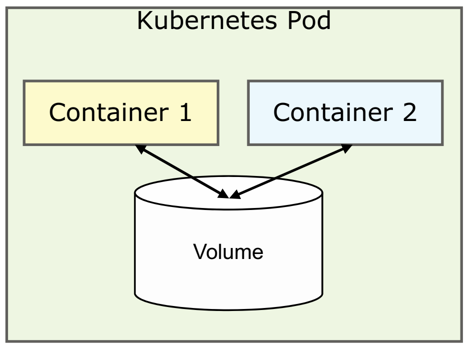
Reference:

https://kubernetes.io/docs/concepts/overview/working-with-objects/kubernetes-objects/

IBM Training

Pods

- A Pod is the smallest unit of a Deployment that can be managed by Kubernetes.
- It consists of a group of one or more containers with a shared network, storage, and a specification for how to run containers.
- It contains one or more application containers that are tightly coupled.



Developing containerized applications on Kubernetes

Figure 8-22. Pods

A Pod is the smallest and simplest Kubernetes object. A Pod represents a set of running containers on your cluster. A Pod is typically set up to run a single primary container. It can also run optional sidecar containers that add supplementary features like logging. Pods are commonly managed by a Deployment.

You have already seen how to deploy containers in slide "Building a container image from a Dockerfile". In Kubernetes, if you want to deploy it, it is deployed as a container in a Pod.

Unlike containers in separate Pods, containers within the same Pod share an IP address and port space, and can locate each other through localhost, and communicate by using standard inter-process communication (semaphores and shared memory).

Containers within the same Pod have access to shared volumes.

Reference:

https://kubernetes.io/docs/concepts/workloads/Pods/Pod/

IBM Training

Deployment

The difference between kind: Pod, and Deployment is that Deployment maintains the replicas, so when you delete a Pod, it is rescheduled automatically by the master so that the current replicas match the wanted one.

Here is an example of a Deployment manifest file in YAML format:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-student-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: hello-student
  template:
    metadata:
      labels:
        app: hello-student
    spec:
      containers:
      - name: hello-student
        image: us.icr.io/cr-student/hello-student:v1
        ports:
        - containerPort: 3000
```

Developing containerized applications on Kubernetes

Figure 8-23. Deployment

A Deployment runs multiple replicas of your application and automatically replaces any instances that fail or become unresponsive.

The main job of a Deployment is to provide declarative updates to both Pods and ReplicaSet. Deployment maintains the replicas so that when you delete a Pod, it is rescheduled automatically by the master so that the current replicas match the wanted one.

Both Pod and Deployment are full-fledged objects in the Kubernetes API.

Deployment manages to create Pods through ReplicaSets.

Deployment creates Pods by using a spec that is taken from the template. It is unlikely that you will ever need to create Pods directly for a production use case.

Here is an example of a Deployment operation:

1. A Deployment that is named hello-student-deployment is created and indicated by the “metadata: name” field.
2. The Deployment creates three replicated Pods, which are indicated by the “replicas” field.
3. The Pod template, or “spec: template” field, indicates that its Pods are labeled “app: hello-student”.
4. The Pod template's specification, or “template: spec field”, indicates that the Pods run one container, hello-student, which is at the “us.icr.io/cr-student/hello-student” IBM Cloud Container Registry image at version “v1”.
5. The Deployment opens port 3000 for use by the Pods.

References:

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

https://developer.ibm.com/tutorials/yaml-basics-and-usage-in-kubernetes/

IBM Training

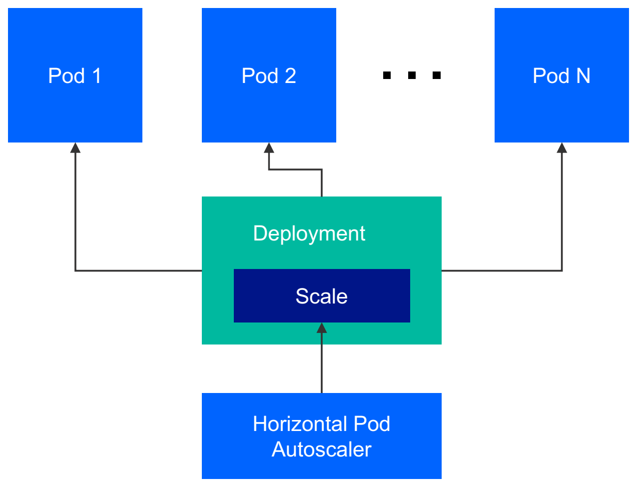
Scaling the Deployment

- Scale in and out the Deployment by changing the number of replicas by running the following command:

```
kubectl scale deployment <deployment name> --replicas=4
```

- For example, you can use Horizontal Pod Autoscaler to increase or decrease automatically the number of instances of your apps based on CPU by running the following command:

```
kubectl autoscale deployment <deployment name> --cpu-percent=80 --
min=4 --max=10
```



Developing containerized applications on Kubernetes

Figure 8-24. Scaling the Deployment

A *replica* is how Kubernetes scales out a Deployment. A replica is a copy of a Pod that already contains a running Service. By having multiple replicas of a Pod, you can ensure that your Deployment has the available resources to handle the increasing load on your app.

Horizontal Pod Autoscaler automatically scales the number of Pods in a replication controller, Deployment, or replica set based on observed CPU utilization.

References:

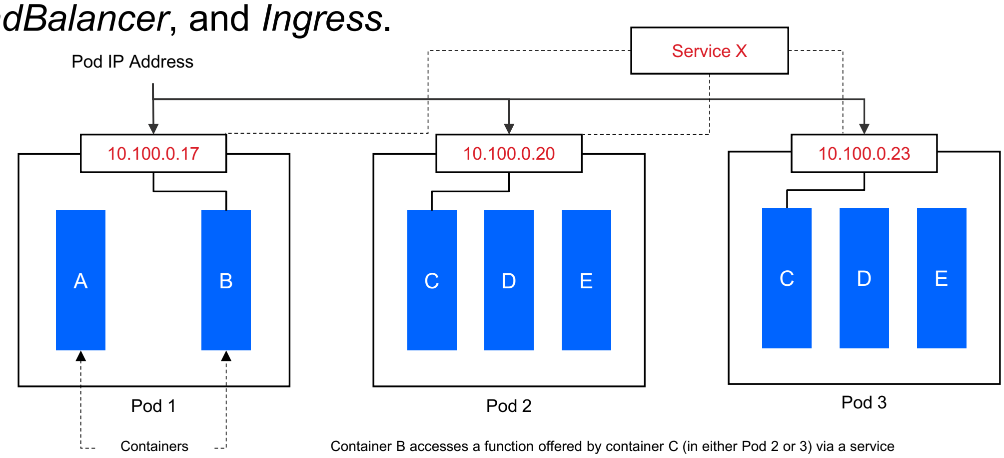
https://cloud.ibm.com/docs/containers?topic=containers-app

https://kubernetes.io/docs/tasks/run-application/horizontal-Pod-autoscale/

IBM Training

Services

- *Services* describe a logical set of Pods and a policy to access them.
- The set of Pods that is targeted by a Service is usually determined by a *Label Selector*.
- The Service propagates state and networking information to all the worker nodes.
- Various Service exposure types exist, such as *ClusterIP*, *NodePort*, *LoadBalancer*, and *Ingress*.



Developing containerized applications on Kubernetes

Figure 8-25. Services

A Service provides a single access point to a set of Pods that is identified by Labels and selected by Selectors. Labels tag objects (like Pods) with attributes (key-value pairs), and are used to organize the objects. Selectors allow filtering a list of resources based on Labels, and are applied when querying a list of resources.

You must have the Service object because the Pods from the Deployment object can be killed or scaled up and down, and you cannot rely on their IP addresses because they are not persistent.

Various Service exposure types exist:

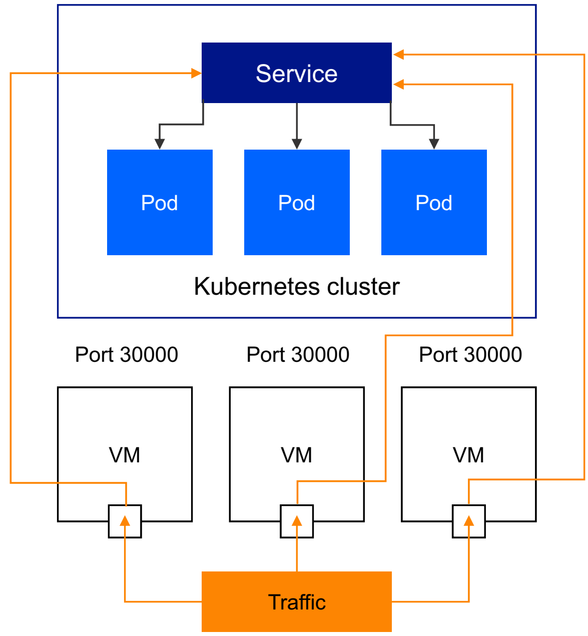
- ClusterIP services make your apps accessible internally to allow communication between Pods in your cluster only.
- NodePort, LoadBalancer, and Ingress services make your apps externally accessible from the public internet or a private network.
- NodePort exposes your app in a specific port on all worker nodes.
- LoadBalancer exposes your apps in an externally facing load-balancer.
- Ingress is a collection of routing rules that govern how external users access services that are running in a Kubernetes cluster. You can use Ingress to expose multiple app services to the public or to a private network by using a unique public or private route.

IBM Training

Services example: NodePort

- NodePort exposes a Service on all the worker nodes on a specific port.

```
apiVersion: v1
kind: Service
metadata:
  name: hello-student-service
spec:
  selector:
    app: hello-student
  ports:
  - name: http
    protocol: TCP
    port: 80
    nodePort: 30000
  type: NodePort
```



Developing containerized applications on Kubernetes

Figure 8-26. Services example: NodePort

A NodePort is an open port on every node of your cluster. Kubernetes transparently routes incoming traffic on the NodePort to your Service, even if your application is running on a different node.

When you expose apps with a NodePort Service, a NodePort in the range of 30000 - 32767 and an internal cluster IP address is assigned to the Service. To access the Service from outside the cluster, you use the public or private IP address of any worker node and the NodePort in the format ”<IP_address>:<nodeport>”. However, the public and private IP addresses of the worker node are not permanent. When a worker node is removed or re-created, new public and private IP addresses are assigned to the worker node. NodePorts are ideal for testing public or private access or providing access for only a short amount of time.

In this example, the service name is “`hello-student-service`” and specifies all Pods with a Label “app: hello-student” to be exposed as part of this Service. The port that is exposed by the Pod is 80, and the application is exposed on all worker nodes on port 30000. The type of the Service is identified in the last line as NodePort.

IBM Training

Storage: Persistence Volumes

- By default, the file systems in Kubernetes offer ephemeral storage. As such, data does not survive a container restart.
- Kubernetes volumes provide persistent storage.
- This storage can also be used as a shared disk space across containers within a Pod.

Example types of volumes:

- CephFS
- GlusterFS
- NFS
- vSphere Volumes

Developing containerized applications on Kubernetes

Figure 8-27. Storage: Persistence Volumes

The container file system lives only while the container does. So, if your app's state must survive relocation, restarts, and crashes, you must configure persistent storage.

You can choose from several options to store your app data and share data across Pods in your cluster. However, not all storage options offer the same level of persistence and availability in situations where a component in your cluster or a whole site fails.

Links for reference:

https://kubernetes.io/docs/concepts/storage/volumes/

https://kubernetes.io/docs/concepts/storage/persistent-volumes/

8.6. Next steps

IBM Training

Next steps

Developing containerized applications on Kubernetes

Figure 8-28. Next steps

IBM Training

Topics

- Containers
- Container orchestration
- Introducing Kubernetes
- Kubernetes architecture
- Kubernetes objects
- Next steps

Developing containerized applications on Kubernetes

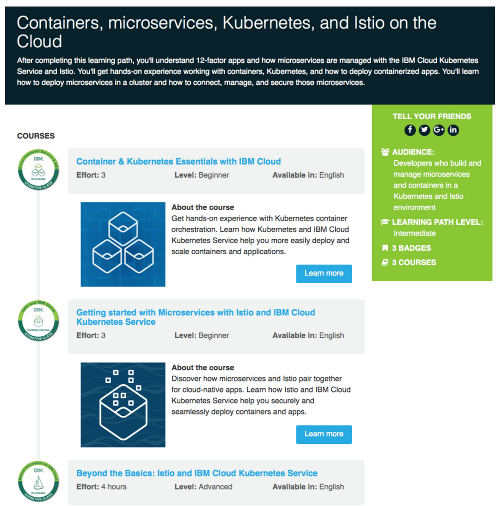
Figure 8-29. Topics

IBM Training

Further reading

For more information, see the following resources:

- https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/
- https://www.ibm.com/cloud/garage/content/course/kubernetes-101
- https://cognitiveclass.ai (for free badges and training)



Developing containerized applications on Kubernetes

Figure 8-30. Further reading

IBM Training

Unit summary

- Explain containers and the difference between containers and virtual machines (VMs).
- Describe container orchestration (Kubernetes).
- List the key capabilities of Kubernetes.
- Articulate the importance of using Kubernetes to prevent vendor lock-in.
- Describe the Kubernetes building blocks: Pod, Deployment, and Service.
- Scale and auto-scale your Deployment for high availability.

Developing containerized applications on Kubernetes

Figure 8-31. Unit summary

IBM Training

Review questions

1. True or False: You can have only one container per Kubernetes Pod.

2. The Kubernetes Scheduler:
   A. Determines which node is going to host a Pod.
   B. Checks etcd for the actual state.
   C. Returns the Pod status.
   D. Holds the wanted state of the Pod.

3. What is the smallest Deployment unit in Kubernetes?
   A. Container
   B. Image
   C. Deployment
   D. Pod

Developing containerized applications on Kubernetes

Figure 8-32. Review questions

IBM Training

Review questions (cont.)

4. True or False: A container registry stores and distributes container images.

5. True or False: **Services** describe a logical set of Pods and a policy to access them.

6. In Kubernetes, the ___ controls the _____.
   A. Master, nodes
   B. Docker registry, kubelet
   C. Server, client
   D. Master, YAML

Developing containerized applications on Kubernetes

Figure 8-33. Review questions (cont.)

IBM Training

Review answers

1. True or **False**: You can have only one container per Kubernetes Pod.

2. The Kubernetes Scheduler:
   A. Determines which node is going to host a Pod.
   B. Checks etcd for the actual state.
   C. Returns the Pod status.
   D. Holds the wanted state of the Pod.

3. What is the smallest Deployment unit in Kubernetes?
   A. Container
   B. Image
   C. Deployment
   D. Pod

Developing containerized applications on Kubernetes

Figure 8-34. Review answers

IBM Training

Review answers (cont.)

4. **True** or False: A container registry stores and distributes container images.

5. **True** or False: Services describe a logical set of Pods and a policy to access them.

6. In Kubernetes, the ___ controls the _____.
   - A. **Master, nodes**
   - B. Docker registry, kubelet
   - C. Server, client
   - D. Master, YAML

Figure 8-35. Review answers (cont.)

Unit 9. IBM Cloud Kubernetes Service overview

Estimated time

01:00

Overview

This unit introduces IBM Cloud Kubernetes Service and describes how a Kubernetes cluster lets you securely manage the resources that you need to deploy, update, and scale applications.

IBM Training

Unit objectives

- Create a Kubernetes cluster by using the IBM Cloud Kubernetes Service.
- Create containers and build on the IBM Cloud Container Registry.
- Work with helm charts.
- Deploy an application to IBM Kubernetes Service.

IBM Cloud Kubernetes Service overview

Figure 9-1. Unit objectives

9.1. IBM Cloud Kubernetes Service

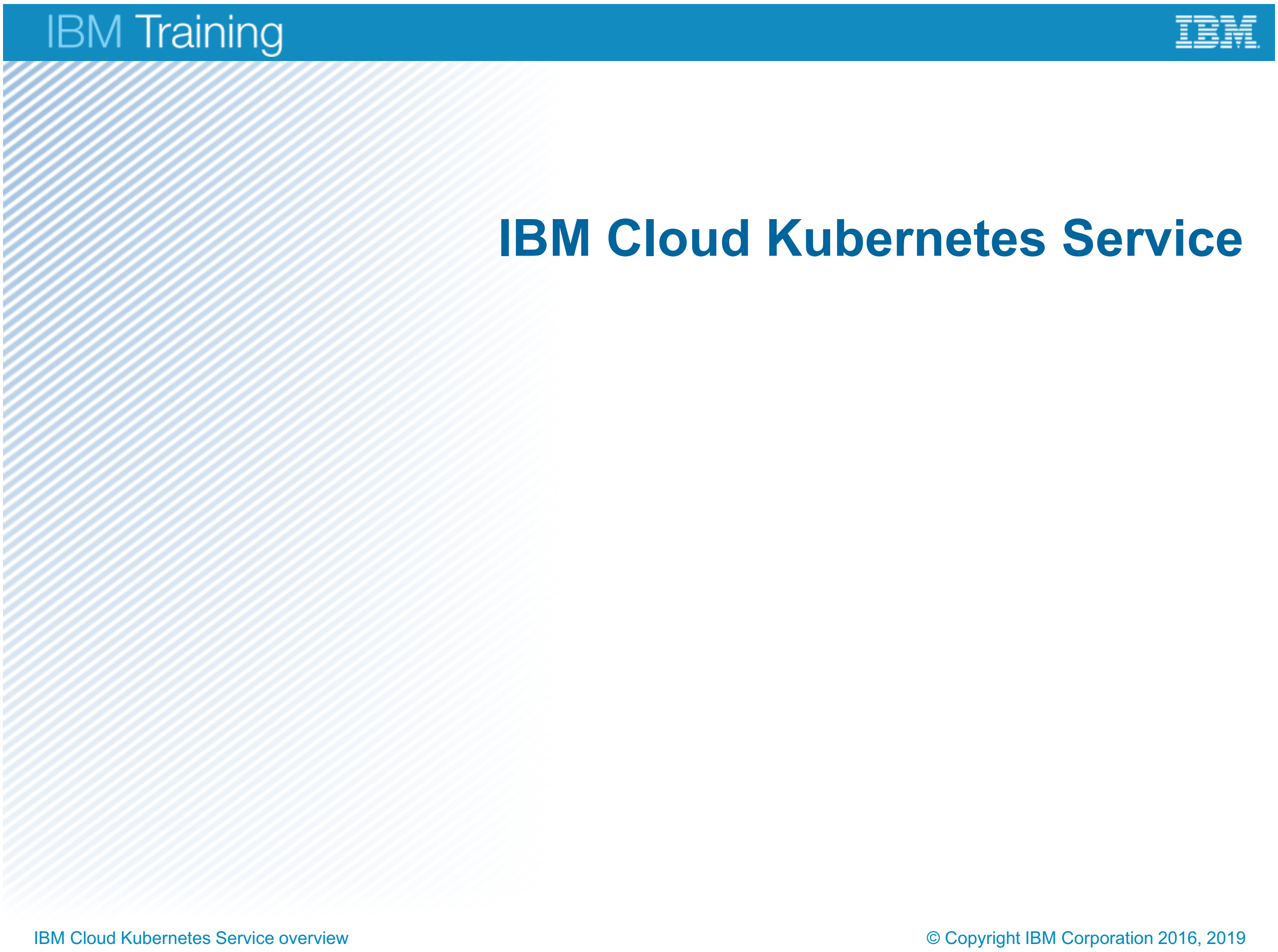


Figure 9-2. IBM Cloud Kubernetes Service

IBM Training

Topics

- IBM Cloud Kubernetes Service
- Deploying an application
- Summary and further reading

IBM Cloud Kubernetes Service overview

Figure 9-3. Topics

IBM Training

IBM Cloud Kubernetes Service

A certified, managed Kubernetes service that provides an intuitive user experience with on-going cluster management. It has built-in security and isolation to enable rapid delivery of apps while using IBM Cloud Services.

Features:

- Secure.
- Automated lifecycle management.
- Fully integrated with IBM Cloud and third-party services.
- Fully integrated with cognitive solutions with various Watson APIs.
- Different worker node types.
- Supports community-certified Kubernetes and Red Hat OpenShift (RHOS) Kubernetes.

IBM Cloud Kubernetes Service overview

Figure 9-4. IBM Cloud Kubernetes Service

You can choose the type of cluster, such as Red Hat Open Shift and Kubernetes, for IBM Cloud Kubernetes Service.

Kubernetes

Kubernetes is an open source project that is hosted by the Cloud Native Computing Foundation (CNCF). The CNCF is a vendor-neutral governance group and a subfoundation of the Linux Foundation.

Kubernetes prevents vendor lock-in because it is offered by major cloud providers. For example, IBM, Amazon, Google, and Microsoft offer a certified Kubernetes distribution on their Cloud platform, such as IBM Cloud Kubernetes Service, IBM Cloud Private, Azure Kubernetes Service, and Google Kubernetes Engine. This flexibility enables clients to port their workload from one vendor to another one.

IBM Cloud Kubernetes Service continues to be the first public-managed Kubernetes service to support the latest upstream versions from the community.

Red Hat Open Shift on IBM Cloud

IBM Cloud Kubernetes Service adds support for users that require access to a fully managed Red Hat OpenShift (RHOS) environment. IBM Cloud Kubernetes Service manages 20,000+ clusters by using upstream Kubernetes software, and it now uses that experience to provide users with RHOS clusters. RHOS is a platform as a service (PaaS) environment that is built on Kubernetes and containers, which provides users with various packages of open source tools, such as your application runtimes, frameworks, databases, and more in one place.

IBM Cloud Kubernetes Service RHOS clusters integrate with IBM Cloud services, such as Identity and Access Manager (IAM), monitoring, logging, IBM Watson, Internet of Things (IoT), IBM Cloud Databases, and DevOps tools. IBM Cloud Kubernetes Service RHOS clusters support the same compute isolation choices as the traditional Kubernetes clusters with hourly Virtual Server Instance (VSI) worker nodes. It also integrates with IBM Cloud Container Registry and Vulnerability Advisor to provide security insights into your static images and live containers. IBM Cloud Kubernetes Service RHOS clusters can run on Red Hat Enterprise Linux, which provides metered billing for RHOS licensing.

References:

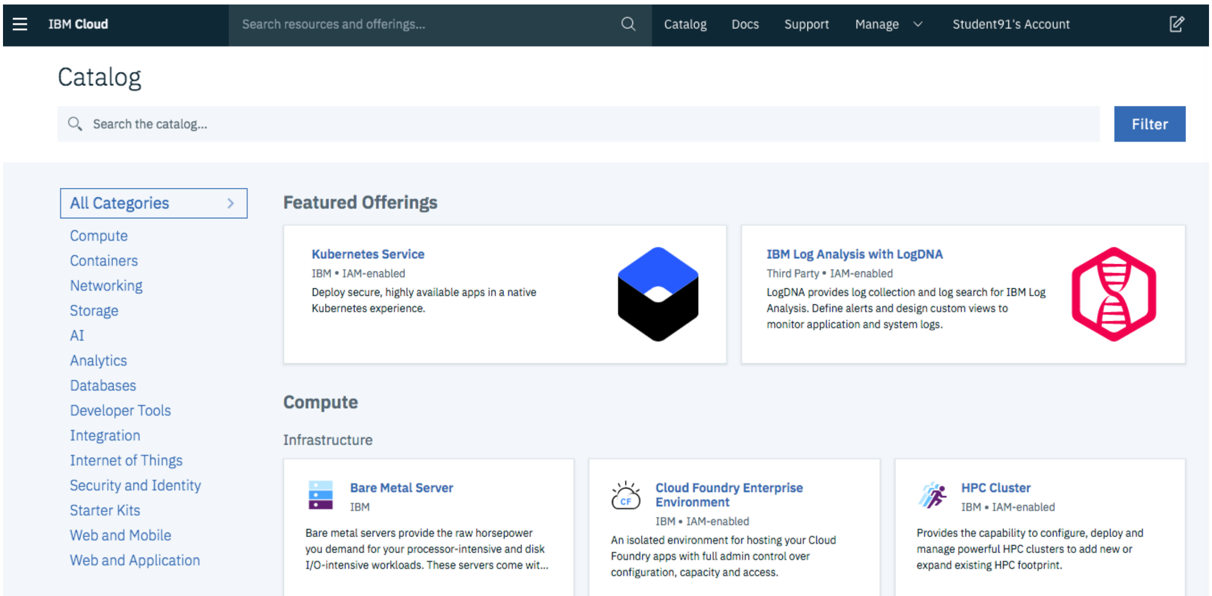
Certified Kubernetes: https://github.com/cncf/k8s-conformance

Deploy OpenShift: https://cloud.ibm.com/docs/terraform?topic=terraform-redhat#deploy_openshift

IBM Training

IBM Cloud Kubernetes Service demonstration

Access the IBM Service catalog and select **Kubernetes Service**.



IBM Cloud Kubernetes Service overview

Figure 9-5. IBM Cloud Kubernetes Service demonstration

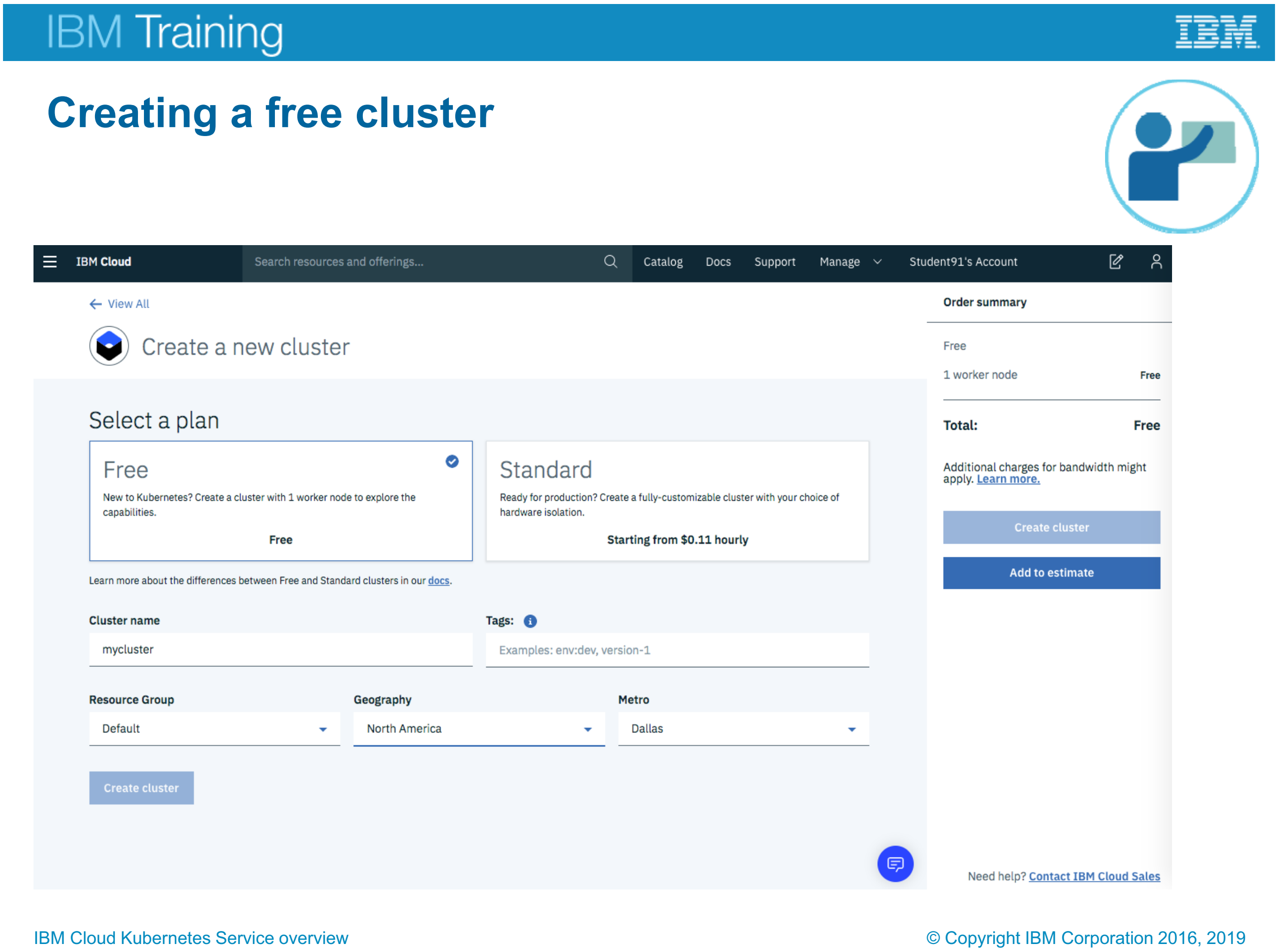


Figure 9-6. Creating a free cluster

You can create one free cluster or any number of standard clusters.

Try out free clusters to become familiar with some of the Kubernetes capabilities. Free clusters are automatically deleted after 30 days.

You can create standard clusters to use the full capabilities of Kubernetes to deploy apps, but in this course you use only the free cluster capabilities.

To create a free cluster, click **Create Cluster** and give your cluster a unique name. A worker pool is created that contains one worker node. The worker node can take a few minutes to provision, but you can see its progress in the **Worker nodes** tab. When the status reaches "Ready", you can start working with your cluster.

The free cluster has one virtual worker node that is grouped into a worker pool with two processors, 4 GB of memory, and a single 100 GB SAN disk that is available for your apps to use. When you create a standard cluster, you can choose between physical (bare metal server) or virtual machines (VMs) of various machine sizes.

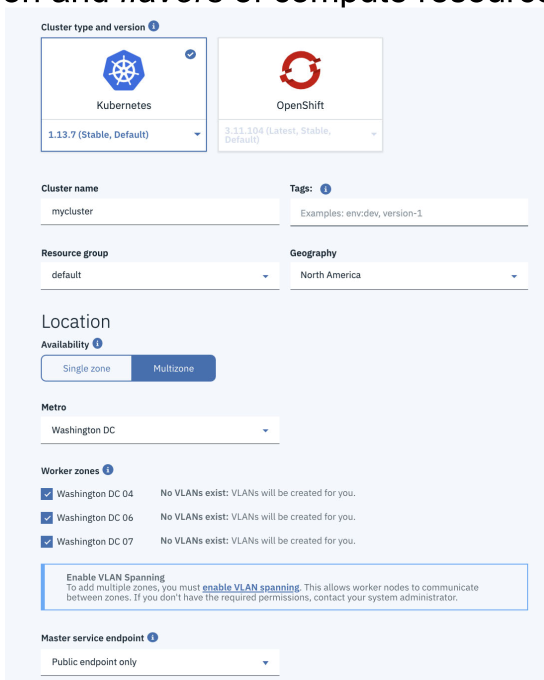
Reference:

https://cloud.ibm.com/docs/containers?topic=containers-getting-started

IBM Training

Instructor demonstration

If you choose to use a standard cluster, you can select the Kubernetes version and *flavors* of compute resources.



IBM Cloud Kubernetes Service overview

Figure 9-7. Instructor demonstration

If you choose to use a standard cluster, you can select the Kubernetes version and *flavors* of compute resources.

Note

The flavor defines the amount of virtual CPU, memory, and disk space that is set up in each worker node and made available to the containers.

Worker nodes are available in selected flavors of compute resources.

Each worker node is a physical machine (bare metal server) or a VM that runs on physical hardware in the cloud environment.

What is a worker pool
A worker pool is a collection of worker nodes with the same flavor, such as machine type, processor, and memory. When you create a cluster, a default worker pool is automatically created for you. To spread the worker nodes in your pool across zones, add worker nodes to the pool, or update worker nodes, you can use the **ibmcloud ks worker-pool** commands.

References:

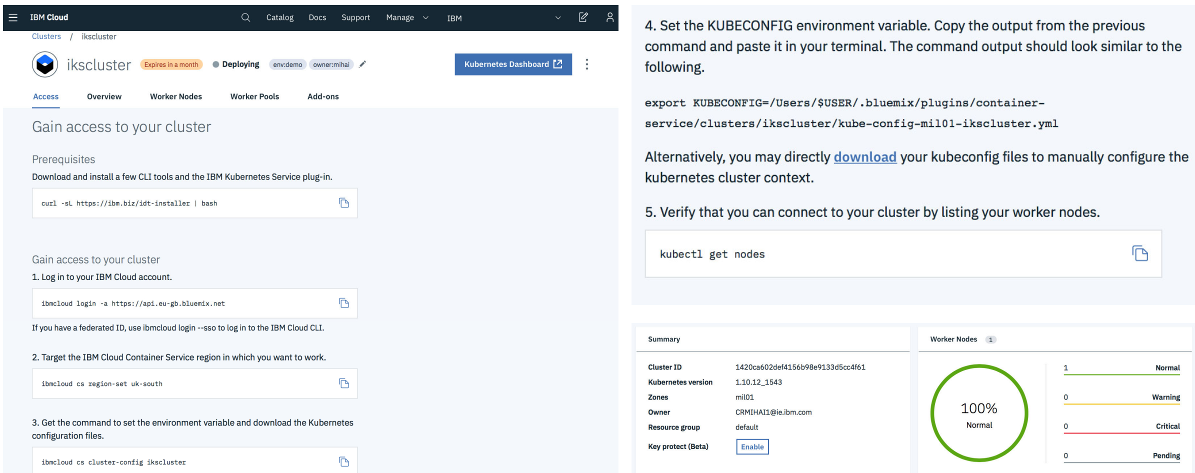
For more information about system limitations, see the following website:

https://cloud.ibm.com/docs/containers?topic=containers-ibm-cloud-kubernetes-service-technology&locale=en

IBM Training

Accessing your cluster

Wait for your cluster to provision, and then access it by using the **`kubectl`** command.



IBM Cloud Kubernetes Service overview

© Copyright IBM Corporation 2016, 2019

Figure 9-8. Accessing your cluster

You can access your cluster by using the IBM Cloud CLI.

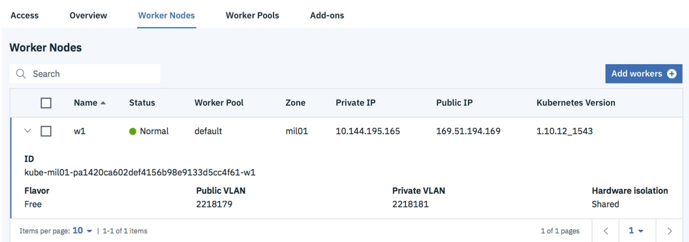
First, set the context for your Kubernetes cluster in your CLI. Then, every time that you log in to the IBM Cloud Kubernetes Service CLI to work with clusters, you must run commands to set the path to the cluster's configuration file as a session variable. The Kubernetes CLI uses this variable to find a local configuration file and certificates that are necessary to connect with the cluster in IBM Cloud.

For more information about these commands, see Slide, “Accessing IBM Cloud Kubernetes Service by using kubectl”, of this unit.

IBM Training

Scaling out

The cluster can scale out by adding more worker nodes.



IBM Cloud Kubernetes Service overview

Figure 9-9. Scaling out

To increase the availability of your apps, you can add worker nodes to an existing zone or multiple existing zones to your cluster. To help protect your apps from zone failures, you can add zones in your cluster.

You can add more worker nodes to a pool by resizing it or by adding more worker pools.

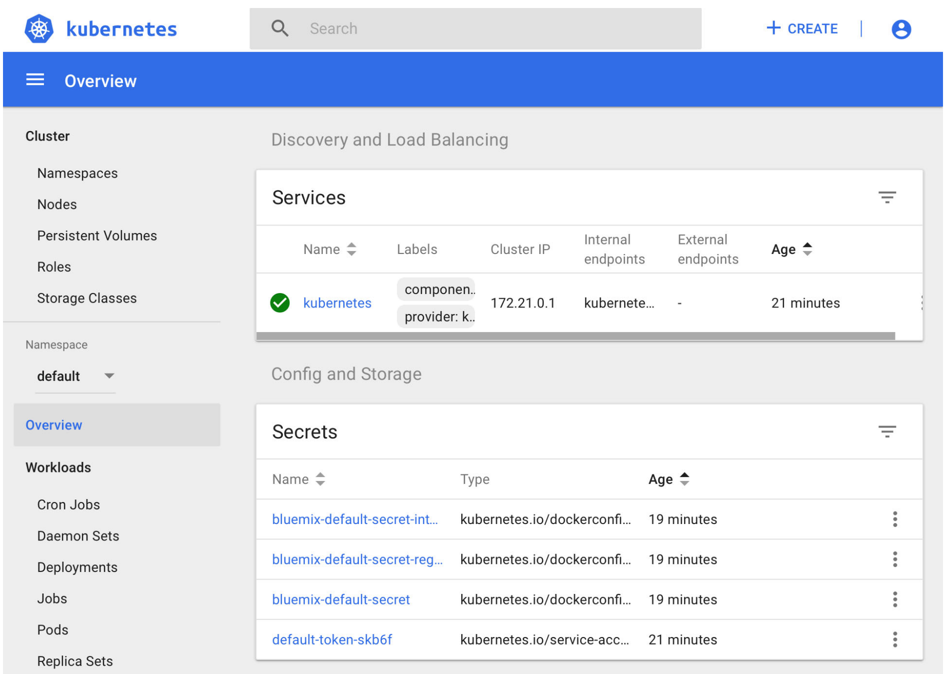
Reference:

https://cloud.ibm.com/docs/containers?topic=containers-add_workers

IBM Training

Kubernetes Dashboard

You can also access the Web UI (Dashboard).



IBM Cloud Kubernetes Service overview

Figure 9-10. Kubernetes Dashboard

Dashboard is a web-based Kubernetes user interface. You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster resources. You can use Dashboard to get an overview of applications running on your cluster, and to create or modify individual Kubernetes resources (such as Deployments, Jobs, and DaemonSets).

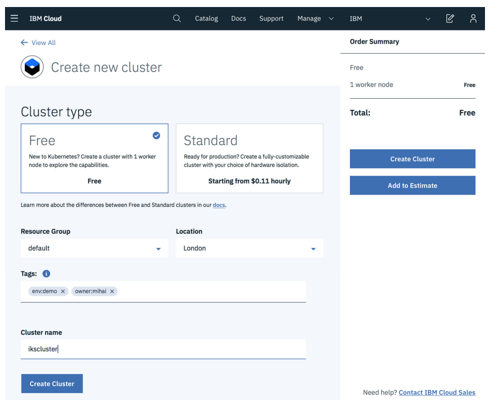
Reference:

https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/

IBM Training

Exercise 5: Part 1

- For exercise 5, you must create a new Kubernetes cluster by clicking **Free Cluster** and then clicking **Create Cluster** on the IBM Cloud console.
  - It takes 30 minutes to provision the cluster.
- Perform Exercise 5, *Part 1. Creating an IBM Cloud Kubernetes Service cluster* **now**.
- Notify the instructor after you click **Create Cluster** to proceed.



IBM Cloud Kubernetes Service overview

Figure 9-11. Exercise 5: Part 1

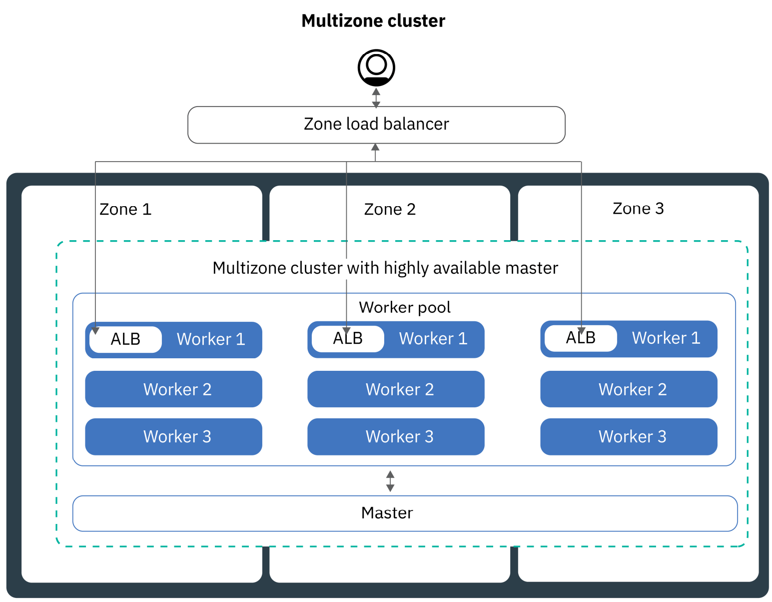
Important

Students must perform Exercise 5, *Part 1. Creating an IBM Cloud Kubernetes Service cluster* **now**. It takes 30 minutes to provision a Kubernetes cluster. By the time this lecture is over, the cluster should be created and you will be able to proceed with Exercise 5. Part 2.

IBM Training

Planning your cluster for high availability

IBM Cloud Kubernetes Service can also deploy across multiple availability zones.



IBM Cloud Kubernetes Service overview

Figure 9-12. Planning your cluster for high availability

- IBM Cloud Region: A region is a geographically and physically separate group of one or more availability zones with independent electrical and network infrastructures that are isolated from other regions. Regions remove shared single points of failure with other regions and ensure low interzone latency within the region.
- IBM Cloud Availability Zone: An availability zone is a logically and physically isolated location within an IBM Cloud Region with independent power, cooling, and network infrastructures that is isolated from other zones to strengthen fault tolerance by avoiding single points of failure between zones while ensuring high bandwidth and low interzone latency within a region.
- Multizone: If you create a cluster in a multizone metro location, the replicas of your highly available Kubernetes master are automatically spread across zones. You may spread your worker nodes across zones to protect your apps from a zone failure.
- Single zone: If you create a cluster in a single data center location, you can create multiple worker nodes, but you cannot spread them across zones. The highly available master includes three replicas on separate hosts, but is not spread across zones.

References:

https://cloud.ibm.com/docs/containers?topic=containers-ha_clusters

https://cloud.ibm.com/docs/containers?topic=containers-regions-and-zones

IBM Training

Accessing IBM Cloud Kubernetes Service by using kubectl

Access the Kubernetes cluster by using the `kubectl` command.

```
# 1. Log in to your IBM Cloud account:
ibmcloud login -a https://cloud.ibm.com

# 2. Target the IBM Cloud Kubernetes Service region:
ibmcloud ks region-set uk-south

# 3. Get the command to set the environment variables and download
the Kubernetes configuration files:
ibmcloud ks cluster-config ikscluster


# 4.  Set the KUBECONFIG environment variable:
export KUBECONFIG=/Users/cmihai/.bluemix/plugins/container-
service/clusters/ikscluster/kube-config-mil01-ikscluster.yml

# 5. Verify that you can connect to your cluster:
kubectl get nodes
kubectl get all
kubectl get pods --all-namespaces
```

IBM Cloud Kubernetes Service overview

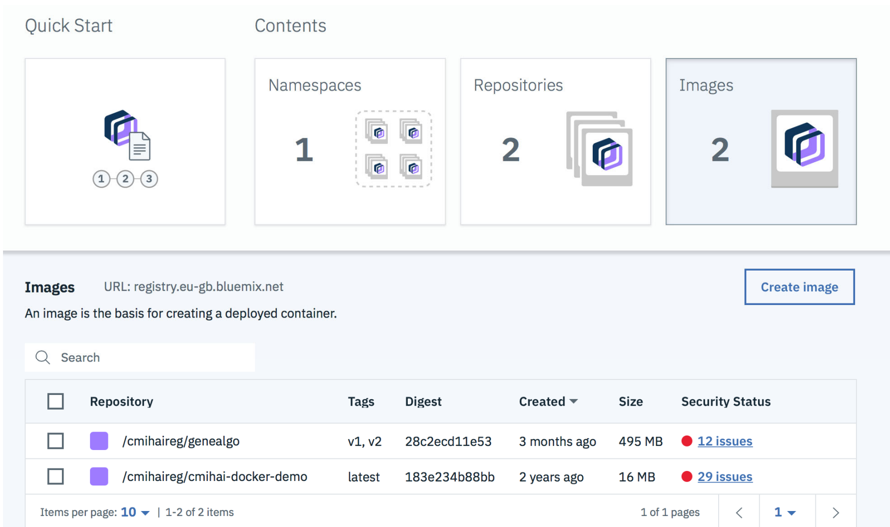
Figure 9-13. Accessing IBM Cloud Kubernetes Service by using kubectl

You can access your cluster by using IBM Cloud CLI and **kubectl** and completing the steps in the slide.

IBM Training

IBM Cloud Container Registry

- Multi-tenant private image registry with an integrated Vulnerability Advisor.
- Check images for known vulnerabilities and create deployment rules to prevent using vulnerable images.



IBM Cloud Container Registry

IBM Cloud Kubernetes Service overview

Figure 9-14. IBM Cloud Container Registry

IBM Cloud Container Registry

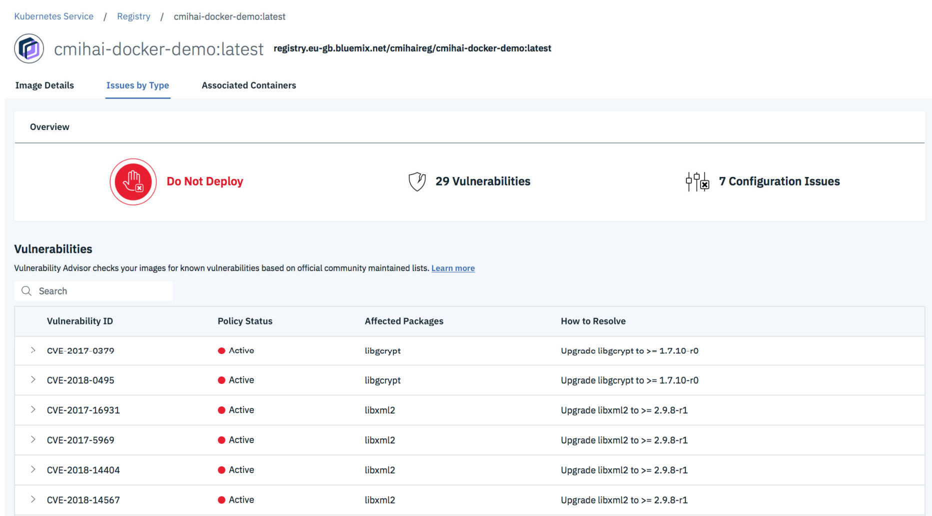
Store and distribute container images in a fully managed private registry. Push private images to conveniently run them in the IBM Cloud Kubernetes Service and other runtime environments. Images are checked for security issues so that you can make informed decisions about your deployments.

IBM Cloud Container Registry provides a multi-tenant, highly available, and scalable private image registry that is hosted and managed by IBM. You can use the private registry by setting up your own image namespace and pushing Docker images to your namespace.

IBM Training

Vulnerability Advisor

- Detailed view of detected vulnerabilities.
- Vulnerability Advisor provides security management for IBM Cloud Container Registry and provides functions to help you secure your images.



IBM Cloud Kubernetes Service overview

Figure 9-15. Vulnerability Advisor

Vulnerability Advisor checks the security status of container images that are provided by IBM, third parties, or added to your organization's registry namespace. If you install the container scanner in each cluster, Vulnerability Advisor also checks the status of running containers.

When you add an image to a namespace, the image is automatically scanned by Vulnerability Advisor to detect security issues and potential vulnerabilities. If security issues are found, instructions are provided to help fix the reported vulnerability.

IBM Training

Using the Private Image Registry

Command-line usage

```
# List images
ibmcloud cr login
ibmcloud cr image-list

# Creating a namespace
ibmcloud cr namespace-list
ibmcloud cr namespace-add <my_namespace>

# Tag and push an image
ibmcloud cr build -t
us.icr.io/<my_namespace>/<my_repository>:<my_tag> .
```

IBM Cloud Kubernetes Service overview

Figure 9-16. Using the Private Image Registry

- **ibmcloud cr login**

  Run this command to log in to IBM Cloud Container Registry.

- **ibmcloud cr images-list**

  Run this command to check all the images that are built on IBM Cloud Container Registry.

- **ibmcloud cr namespace-list**

  Run this command to display all the namespaces that are owned by your IBM Cloud account.

- **ibmcloud cr namespace-add <my_namespace>**

  Run this command to add a namespace to a container. In this unit, it is “oneibm”.

- **ibmcloud cr build -t us.icr.io/<my_namespace>/<my_repository>:<my_tag> .**

  Run this command to build the image by using a Dockerfile into IBM Cloud Registry and give it the tag “registry.ng.bluemix.net/oneibm/hellohub:v1”. (You do not need to have Docker installed on the machine.)

9.2. Deploying an application

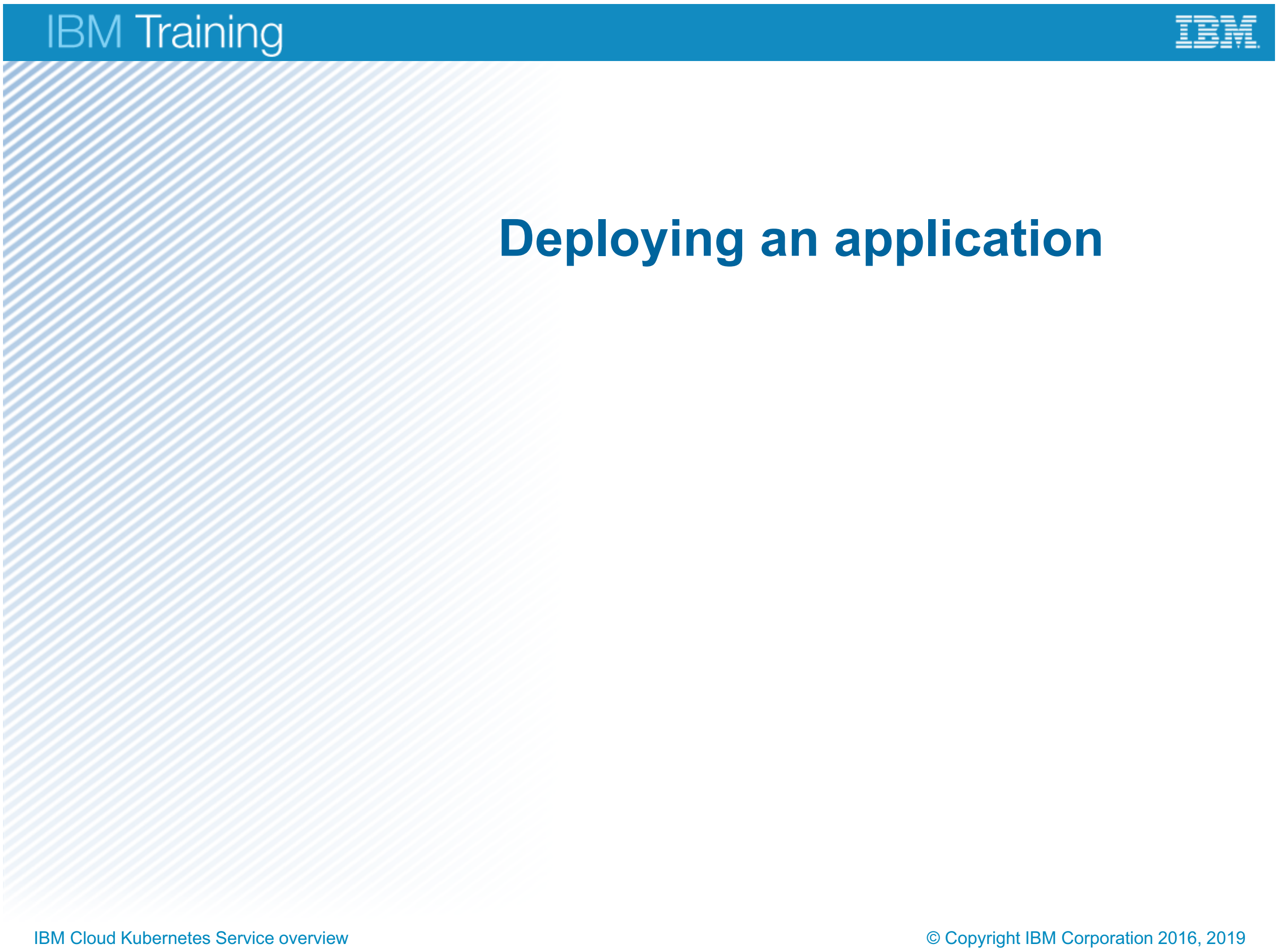


Figure 9-17. Deploying an application

IBM Training

Topics

- IBM Cloud Kubernetes Service
- Deploying an application
- Summary and further reading

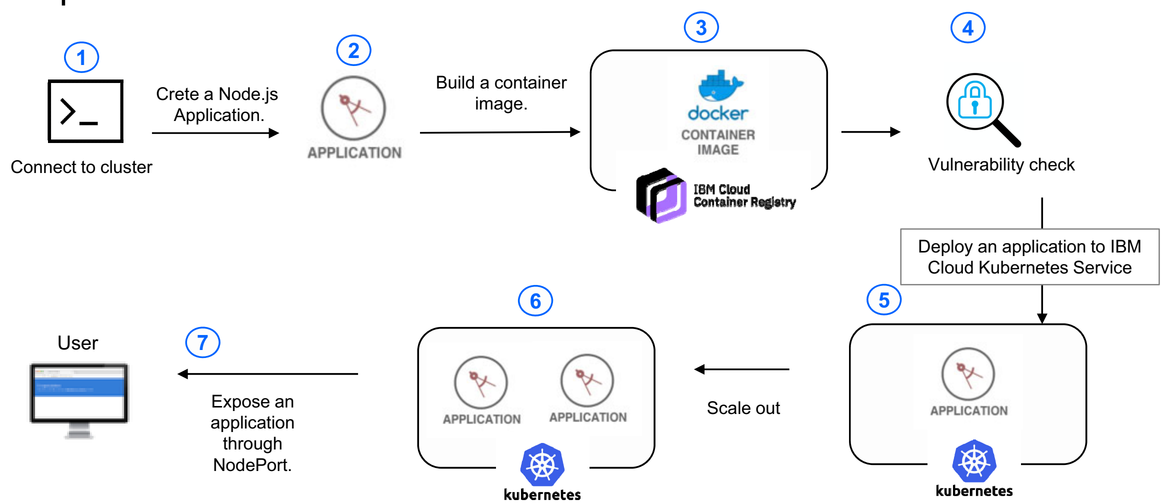
IBM Cloud Kubernetes Service overview

Figure 9-18. Topics

IBM Training

Deploying an application on IBM Cloud Kubernetes Service

You can deploy an application according to the steps that are shown in this slide. The circled numbers are matched with each step of the next slide.



IBM Cloud Kubernetes Service overview

© Copyright IBM Corporation 2016, 2019

Figure 9-19. Deploying an application on IBM Cloud Kubernetes Service

IBM Training

Deploy an application on IBM Cloud Kubernetes Service

You deploy an application in exercise 6 by completing the following steps:

1. Connect to your cluster by using the CLI.
2. Create a Node.js sample application.
   - IBM Cloud App Service starter kits provide a pre-configured sample application.
3. Build a container image.
   - You build your images directly on the IBM Cloud Container Registry service.
4. Check the security status of container images.
   - IBM Cloud Container Registry provides the Vulnerability Advisor service.
5. Deploy an application to IBM Cloud Kubernetes Service.
   - You can directly manipulate deployment through YAML.
6. Scale out an application.
   - You can scale an application manually or set up the autoscale function.
7. Expose the app over the internet.
   - You expose the service by using NodePort.

IBM Cloud Kubernetes Service overview

Figure 9-20. Deploy an application on IBM Cloud Kubernetes Service

9.3. Summary and further reading

IBM Training

Summary and further reading

IBM Cloud Kubernetes Service overview

Figure 9-21. Summary and further reading

IBM Training

Topics

- IBM Cloud Kubernetes Service
- Deploying an application
- Summary and further reading

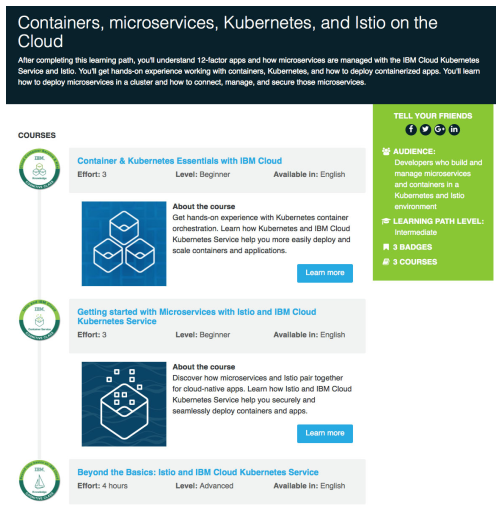
IBM Cloud Kubernetes Service overview

Figure 9-22. Topics

IBM Training

Further reading

- https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/
- https://www.ibm.com/cloud/garage/content/course/kubernetes-101
- https://cognitiveclass.ai



IBM Cloud Kubernetes Service overview

Figure 9-23. Further reading

IBM Training

Unit summary

- Create a Kubernetes cluster by using the IBM Cloud Kubernetes Service.
- Create containers and build on the IBM Cloud Container Registry.
- Work with helm charts.
- Deploy an application to IBM Kubernetes Service.

IBM Cloud Kubernetes Service overview

Figure 9-24. Unit summary

IBM Training

Review questions

1. True or False. : You cannot scale out the worker nodes after you create a Kubernetes cluster on IBM Cloud Kubernetes Service.
2. Which of the following services checks images for known vulnerabilities?
   A. Vulnerability Advisor
   B. Key Protect
   C. AppID
   D. FortiGate Security Appliance
3. Which of the following items is supported by IBM Cloud Kubernetes Service? (Choose two items)
   A. Red Hat OpenShift
   B. Kubernetes
   C. Apache Mesos
   D. Rocket

IBM Cloud Kubernetes Service overview

Figure 9-25. Review questions

IBM Training

Review questions (cont.)

4. True or False: You can deploy an application to a multizone in IBM Cloud Kubernetes Service.

5. What is the CLI command to view details of the nodes?
   - A. `kubectl get nodes`
   - B. `ibmcloud ks nodes`
   - C. `kubectl get pods`
   - D. `ibmcloud ks pods`

IBM Cloud Kubernetes Service overview

Figure 9-26. Review questions (cont.)

IBM Training

Review answers

1. True or **False**. : You cannot scale out the worker nodes after you create a Kubernetes cluster on IBM Cloud Kubernetes Service.
2. Which of the following services checks images for known vulnerabilities?
   A. **Vulnerability Advisor**
   B. Key Protect
   C. AppID
   D. FortiGate Security Appliance

3. Which of the following items is supported by IBM Cloud Kubernetes Service? (Choose two items)
   A. **Red Hat OpenShift**
   B. **Kubernetes**
   C. Apache Mesos
   D. Rocket

IBM Cloud Kubernetes Service overview

Figure 9-27. Review answers

IBM Training

Review answers(cont.)

4. **True** or False: You can deploy an application to a multizone in IBM Cloud Kubernetes Service.

5. What is the CLI command to view details of the nodes?
   A. **`kubectl get nodes`**
   B. `ibmcloud ks nodes`
   C. `kubectl get pods`
   D. `ibmcloud ks pods`

IBM Cloud Kubernetes Service overview

Figure 9-28. Review answers (cont.)

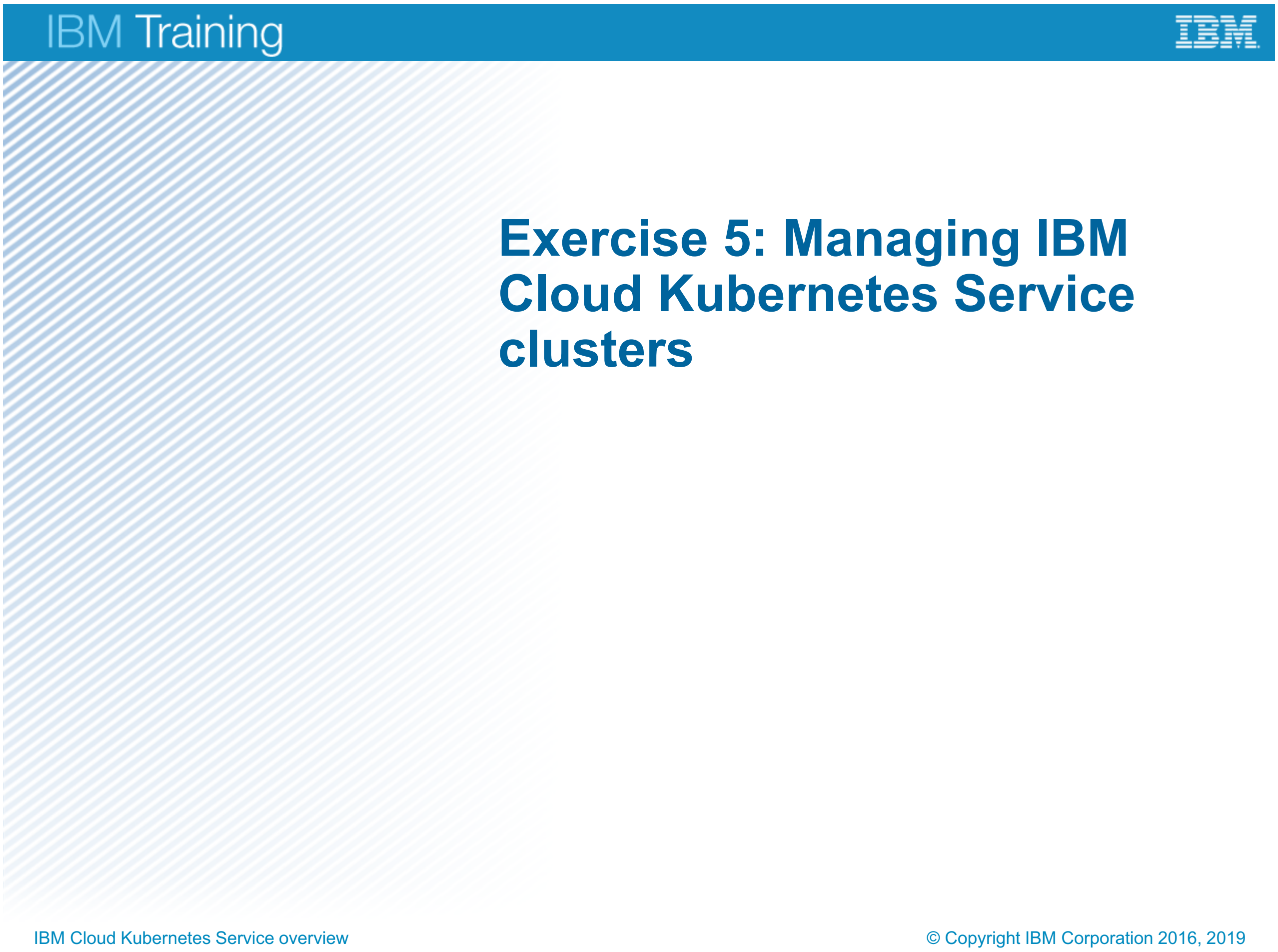


Figure 9-29. Exercise 5: Managing IBM Cloud Kubernetes Service clusters

IBM Training

Exercise objectives

- This exercise demonstrates how to create an IBM Kubernetes Service cluster and manage it by using the **kubectl** CLI.
- After completing this exercise, you should be able to:
  - Create an IBM Cloud Kubernetes Service cluster.
  - Connect to a cluster on IBM Cloud Kubernetes Service.
  - List the worker nodes in a cluster.

IBM Cloud Kubernetes Service overview

Figure 9-30. Exercise objectives

IBM Training

Exercise 6: Deploying an application on Kubernetes

IBM Cloud Kubernetes Service overview

Figure 9-31. Exercise 6: Deploying an application on Kubernetes

IBM Training

Exercise objectives

- In this exercise, you build a containerized application and deploy it to IBM Cloud Kubernetes Service.
- After completing this exercise, you should be able to:
  - Create a containerized Node.js application and build it on IBM Cloud Container Registry.
  - Explain how the container security analysis capability of Vulnerability Advisor can identify the security vulnerabilities by scanning an image.
  - Create a deployment and scale it.
  - Expose your application on the internet.

IBM Cloud Kubernetes Service overview

Figure 9-32. Exercise objectives

IBM Training