Course Guide

IoT Cloud Developer 2020

Course code SAIOT ERC 2.0

Islam Abd Alla

Karim E El-Bawab

Karen Medhat Fahmy

Mohamed Abdel Hafeez

Norhan Khaled

IBM Training

May 2020 edition

Notices

Trademarks

Contents

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:

BigInsights®	Cloudant®	Cognos®
DB™	Db2®	FileNet®
IBM Cloud™	IBM Edge™	IBM Watson®
Maximo®	OpenWhisk®	Rational®
Resource®	SPSS®	TRIRIGA®
Watson IoT™	Watson™	WebSphere®
With Watson®		

Linux® is a registered trademark used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Duration: 5 days

Purpose

The IoT Cloud Developer career path prepares students to apply IoT and cloud computing concepts to build solutions based on devices connected to a cloud-based IoT platform. This career path introduces students to basic concepts of IoT, device connectivity protocols, data generated by sensors and other devices, cloud computing, device simulators, flow-based programming tools, REST APIs, and analytics. Students apply the concepts they learn to practical use cases by using IBM Watson IoT Platform, Watson AI services, and tools on IBM Cloud.

Audience

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

Prerequisites

Before attending Module III IoT Cloud Developer classroom, students must meet the following prerequisites:

- Successful completion of Module I *Internet of Things Overview* (self-study).
- Successful completion of Module II *Node-RED: Basics to Bots* (self-study).

Objectives

After completing this course, you should be able to:

- Explain what IoT is.
- Explain the IBM perspective for Internet of Things (IoT) solution development.
- Describe the IoT reference architecture.
- Describe the characteristics of IBM Watson IoT Platform that simplify the development of IoT solutions.
- Describe the four quadrants of IBM Watson IoT Platform.
- Explain how to connect devices to the IBM Watson IoT Platform solution.
- List the communication protocols that are used in IoT solutions.
- Introduce common IoT devices.
- Describe the key capabilities of IBM Cloud.
- List services available in the IBM Cloud Catalog.
- List databases available on IBM Cloud.

- Manage IBM Cloud users and resources.
- Manage applications and services on IBM Cloud.
- Explain how to get started with the IBM Watson IoT Platform service on IBM Cloud.
- List the features of IBM Watson IoT Platform.
- Use the IBM Watson IoT Platform dashboard.
- Manage devices and set up security policies on IBM Watson IoT Platform.
- Manage user access on IBM Watson IoT Platform.
- Manage applications on IBM Watson IoT Platform.
- Describe the advanced features of IBM Watson IoT Platform.
- Use recipes to help jump-start application development.
- Explain the concept of flow-based programming.
- Explain what Node-RED is.
- Describe the Node-RED workspace and its components.
- Describe the Node-RED palette and the Palette Manager and explain how to use them to build flows.
- Describe the Hypertext Transfer Protocol (HTTP) protocol.
- Explain the concept of the publish and subscribe model.
- Describe the Message Queuing Telemetry Transport (MQTT) protocol.
- Compare the MQTT and HTTP protocols.
- Explain the concept of the application programming interface (API) and describe the types of APIs.
- Explain the characteristics of REST APIs.
- List the IBM Watson artificial intelligence (AI) services.
- Describe the IBM Cloudant database service on IBM Cloud.
- Describe how to access IBM Watson and IBM Cloudant services on IBM Cloud with REST APIs.
- Describe the features of IBM Analytics Engine and IBM Streaming Analytics.
- Describe IBM Geospatial Analytics as a use case of IBM Streaming Analytics.
- Explain the geospatial capabilities in IBM Cloudant database.

Agenda

Note

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

The exercise durations do not include the optional parts or sections. 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 the optional exercises after the class.

Day 1

(00:30) Welcome
(02:30) Unit 1 - Introduction to the Internet of Things and IBM Watson IoT Platform
(01:00) Exercise 1 - Designing an IoT solution
(01:00) Lunch break
(01:30) Unit 2 - Introduction to IBM Cloud
(01:00) Exercise 2 - Creating your first IoT application

Day 2

(02:00) Unit 3 - Introduction to IBM Watson IoT Platform
(00:45) Exercise 3 - Creating an application to monitor room temperature and humidity
(01:00) Lunch break
(01:00) Unit 4 - Introduction to Node-RED
(01:30) Exercise 4 - Extending IoT application functions by using Node-RED

Day 3

(01:00) Unit 5 - IBM Watson IoT Platform network protocols
(01:30) Exercise 5 - Monitoring server CPU usage and reporting statistics
(01:00) Lunch break
(02:30) Unit 6 - Enhancing Internet of Things solutions with REST APIs

Day 4

(01:00) Exercise 6 - Integrating IoT applications with IBM Watson Visual Recognition
(01:30) Unit 7 - Introduction to analytics services on IBM Cloud
(01:00) Lunch break
(02:00) Exercise 7 - Monitoring parking utilization with IBM Cloudant Geospatial

Day 5

(00:30) Course recap
(00:45) Feedback forms
(01:30) Practice test on the Skills Academy Portal
(00:30) Closing ceremony

Unit 1. Introduction to the Internet of Things and IBM Watson IoT Platform

Estimated time

02:30

Overview

This unit introduces the IBM perspective for a complete IoT solution. This unit provides an overview of Watson IoT Platform and explains how this platform facilitates IoT solution development. This unit provides an introduction to IoT devices and commonly used development electronic boards.

IBM Training

Unit objectives

- Explain the IBM perspective for Internet of Things (IoT) solution development.
- Describe the IoT reference architecture.
- Describe the characteristics of Watson IoT Platform that simplify the development of IoT solutions.
- Describe the four quadrants of Watson IoT Platform.
- Explain how to connect devices to the Watson IoT Platform solution.
- List the communication protocols that are used in IoT solutions.
- Introduce common IoT devices.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-1. Unit objectives

1.1. IBM IoT point of view

IBM Training

IBM IoT point of view

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-2. IBM IoT point of view

IBM Training

Topics

- IBM IoT point of view
- IoT reference architecture
- Watson IoT Platform overview
- Connecting devices to Watson IoT Platform
- Communication protocols
- Introduction to IoT devices

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-3. Topics

IBM Training

What is an IoT solution

IOT

- An IoT solution is more than an embedded system that has a connectivity feature.
- An IoT solution is a set of devices and sensors that are connected to a cloud platform through a gateway.

cloud Platform

- The cloud platform provides the infrastructure that is necessary to manage, store, secure, and analyze a large amount of data to extract valuable information and insights from it.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-4. What is an IoT solution

IoT is the result of the convergence of several technologies, including wireless communications, micro-electrical systems, and the internet.

A “thing” is any object with embedded electronics that can transfer data over a network without any human interaction. Some examples are wearable devices, environmental sensors, machinery in factories, devices in homes and buildings, or components in a vehicle.

IoT can make life easier for all. For example, smart homes and connected devices in your home can make your life easier. You can you can turn off lights, lock doors, turn on or off heat or air-conditioners, and do other actions through voice commands or mobile apps. HOW??

The truly transformative use of IoT is to combine structured and unstructured data with cognitive analytics.

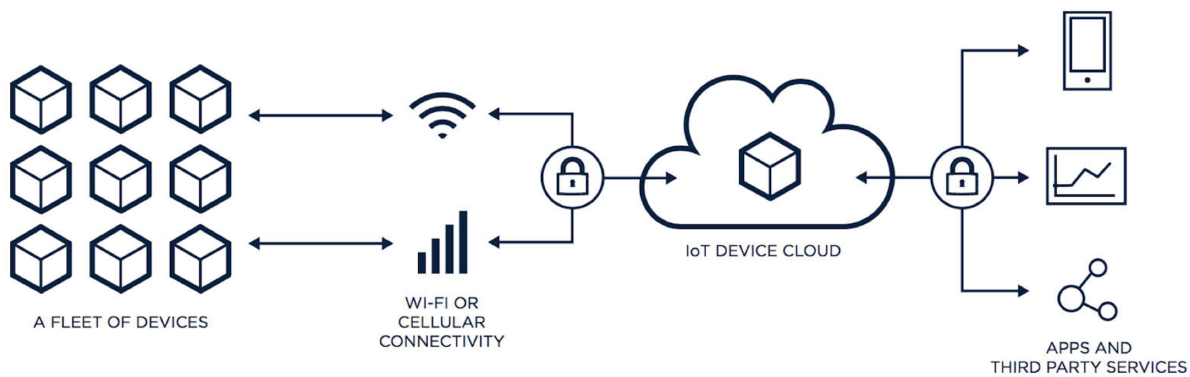
Gains of AI

IoT in the era of AI technologies makes it possible for you to make sense of the vast amounts of IoT data to understand your data better and deeper. Infusing intelligence into systems and processes can help you increase efficiency, improve customer satisfaction, uncover new business opportunities, and mitigate risks and threats proactively.

IBM Training

What is an IoT platform

- An IoT platform provides a set of ready-to-use features that greatly speed up the development of applications for connected devices and provide scalability and cross-device compatibility.
- An *IoT platform* is an integrated service that provides the capabilities that you need to bring physical objects online.



Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-5. What is an IoT platform

IoT platforms originated as IoT middleware, which functions as a mediator between the hardware and application layers. The primary tasks of this middleware include data collection from devices over different protocols and network topologies, remote device configuration and control, device management, and over-the-air firmware updates.

References:

- https://www.kaaproject.org/what-is-iot
- https://hackernoon.com/how-to-choose-the-right-iot-platform-the-ultimate-checklist-47b5575d4e20

IBM Training

Examples of IoT platforms

- Consumer IoT platforms:
  - IBM Watson IoT Platform
  - Amazon Web Services (AWS) IoT Platform
  - Microsoft Azure IoT Platform
  - Google IoT Cloud Platform
- Open-source IoT platforms
  - ThingsBoard
  - myDevices (Cayenne)
  - Blynk
- Industry-based IoT platforms
  - IoT platforms for vehicles
  - IoT platforms for assets management
  - IoT platforms for agriculture

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-6. Examples of IoT platforms

The available IoT platforms can be categorized in different ways. Here are the three main categories:

- Consumer IoT platforms: These platforms provide the basic features of device management, visualization, and connecting devices to the cloud, which can be achieved in different ways (directly or by using a gateway device). Other advanced features over the cloud include storing data services and analysis services. Some examples are IBM Watson IoT Platform, Amazon Web Services (AWS) IoT Platform, Microsoft Azure IoT Platform, and Google IoT Cloud Platform.
- Open-source IoT platforms: Open-source platforms provide basic functions, such as discovery and updating the source code of a platform to customize a user's features. Sometimes, they also provide a function to test the platform from a user's local machine by installing the platform locally. Some examples are ThingsBoard, myDevices (Cayenne), and Blynk.
- Industry IoT platforms: Some IoT platforms are made for specific industries, and they have extra features to support these industries. Some examples are IoT platforms for vehicles, IoT platforms for assets management, and IoT platforms for agriculture.

Reference:

https://hackernoon.com/how-to-choose-the-right-iot-platform-the-ultimate-checklist-47b5575d4e20

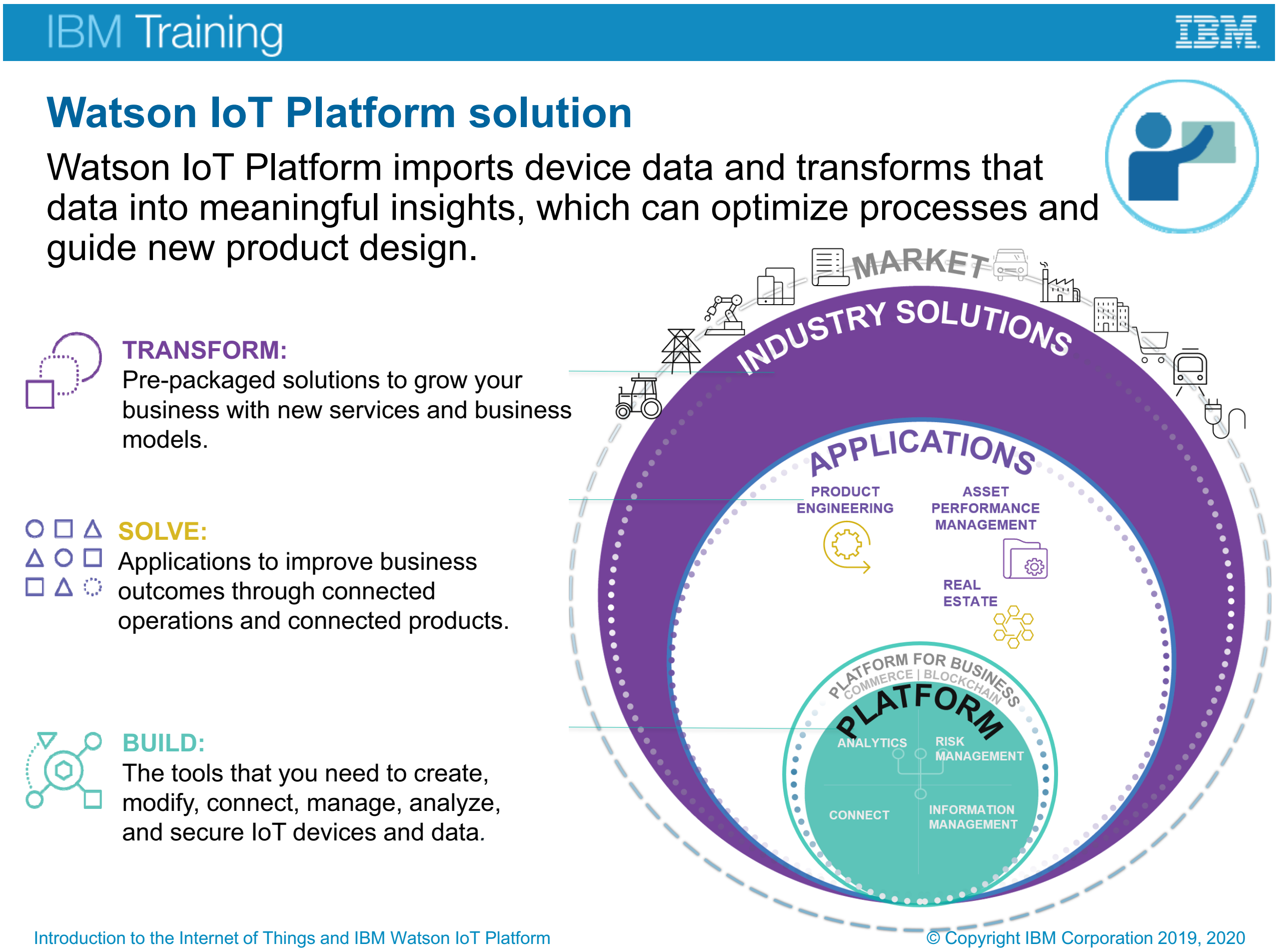


Figure 1-7. Watson IoT Platform solution

The goal of deploying an IoT solution is to discover new business opportunities from the insights that might be derived from the data that is continuously analyzed in real time or from the historical stored data.

Platforms

Watson IoT Platform helps developers to develop rapidly a robust IoT solution. It facilitates all the basic operations for the connecting devices. It manages and secures the data to extract meaningful insights.

You can deploy Watson IoT Platform with blockchain enabled, which can be ideal for a solution that involves multiple parties that must share data that is reported from IoT devices, such as cargo management.

Applications

With Watson IoT Platform, you can enable connectivity features on existing IBM applications. With it, you can empower these applications so that you can use them to adapt concepts like continuous engineering when Watson IoT Platform is integrated with IBM Rational tools. You can also use Watson IoT Platform to make the asset and facility management system more convenient and accurate when it is integrated with IBM Maximo or IBM TRIRIGA.

Industry solutions

It is possible to transform businesses by using prepackaged solutions that use the insights that are derived from Watson IoT Platform.

IBM Training

A connected building that uses Watson IoT Platform

Watch this video: https://youtu.be/EgwFD8ZDBjk

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-8. A connected building that uses Watson IoT Platform

This example shows how Watson IoT Platform converts an unconnected building to a smart building in a matter of hours. It also shows how IBM is uniquely positioned to transform data from that building into business insights.

Reference:

https://youtu.be/EgwFD8ZDBjk

1.2. IoT reference architecture

IBM Training

IoT reference architecture

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-9. IoT reference architecture

IBM Training

Topics

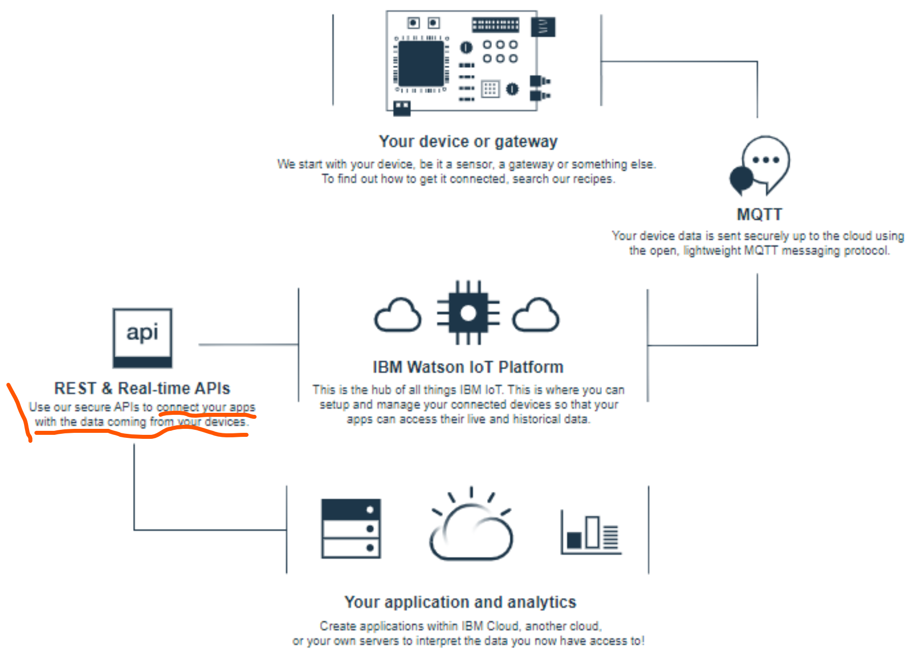
- IBM IoT point of view
- IoT reference architecture
- Watson IoT Platform overview
- Connecting devices to Watson IoT Platform
- Communication protocols
- Introduction to IoT devices

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-10. Topics

IBM Training

IBM IoT basic reference architecture



Introduction to the Internet of Things and IBM Watson IoT Platform

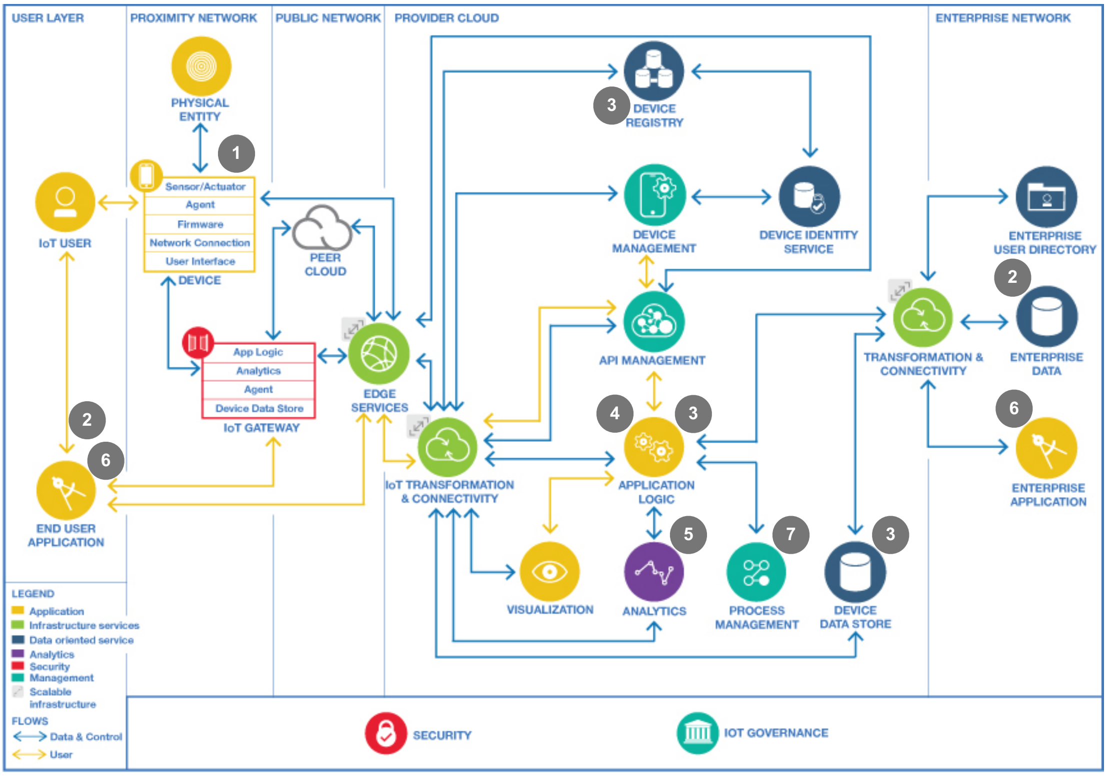
Figure 1-11. IBM IoT basic reference architecture

This slide shows a basic reference architecture that shows the flow of data from the IoT device to the cloud:

1. You connect your device or gateway to Watson IoT Platform.
2. The device communicates with Watson IoT Platform by using the Message Queuing Telemetry Transport (MQTT) protocol.
3. Watson IoT Platform communicates with external applications and services for device management, data visualization, data storage, data analysis, and other useful services that are provided by the cloud platform.

IBM Training

IBM Cloud Reference Architecture for IoT



Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-12. IBM Cloud Reference Architecture for IoT

You can use IBM Cloud IoT, data, and AI services to connect to IoT devices and quickly build scalable apps and visualization dashboards to gain insights from IoT data.

Flow

Smart homes with connected devices and sensors enable insurance companies to improve service for their policy holders while providing insight into risks in the home. By connecting home infrastructure partners, insurers, and services such as weather reporting, the IBM IoT for Insurance solution leverages key components of the IBM Cloud Reference Architecture. With this solution, the policy holder receives notification of potential danger to the home and engages with the insurer in a more proactive manner. For example, leak-detection sensors and valves can monitor for water leaks and protect the home from the resulting damage. The device maker is responsible for the lifecycle of the devices, and the insurance company benefits from access to the device data so that it can provide an improved experience to its policy holders.

Setting up this solution requires the following steps:

1. Sensors and actuators are deployed in the home and attached to the device maker's cloud service. The sensors might include water leak detection, water flow, and temperature sensors, and the actuators can include automatic water shutoff valves.
2. The homeowner logs in to the insurance mobile application and authorizes the insurance service to access the device maker's (peer) cloud and their device data. The mobile application sends the authorization token and insurance company identifier to the cloud service.
3. This information is used to map the user, devices, and insurance policy within the cloud service. The device cloud service is used because the device makers have already deployed their own cloud and own the lifecycle of the device and the user experience with the devices.
4. The insurance service receives information such as authorization, device details, and the insurance ID from the insurance mobile application and processes the data in several nodes (application logic, device registry, and device data store). The devices are registered with the device registry, and data mapping is updated in the application logic component.
5. The insurance service application connects to the device maker (peer) cloud by using the authorization token and requests the data. The application is set up to pull data at a configured interval. In addition to device data, the application can be configured to access other data sources, such as a weather data service for use in analysis.
6. Data from devices and other sources such as the weather service are continually updated and sent to analytics to determine whether a potential risk threshold was exceeded. This data is analyzed to determine whether there is a potential for damage to the home (including water damage, freeze potential, and more). After it is determined that there is a problem, notifications are sent to the homeowner and to the insurance company by using the analysis from step 5. Then, the homeowner can respond to the notification and determine whether damage occurred, and the insurance company can initiate a claim process.
7. If damage occurred, the insurance business process of claims management is initiated. The insurance business process can be accomplished in the cloud service, enterprise applications, or mobile applications, depending on how and where the insurance company decides to perform the business logic.

Reference:

https://www.ibm.com/cloud/garage/architectures/iotArchitecture/reference-architecture

1.3. Watson IoT Platform overview

Figure 1-13. Watson IoT Platform overview

IBM Training

Topics

- IBM IoT point of view
- IoT reference architecture
- Watson IoT Platform overview
- Connecting devices to Watson IoT Platform
- Communication protocols
- Introduction to IoT devices

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-14. Topics

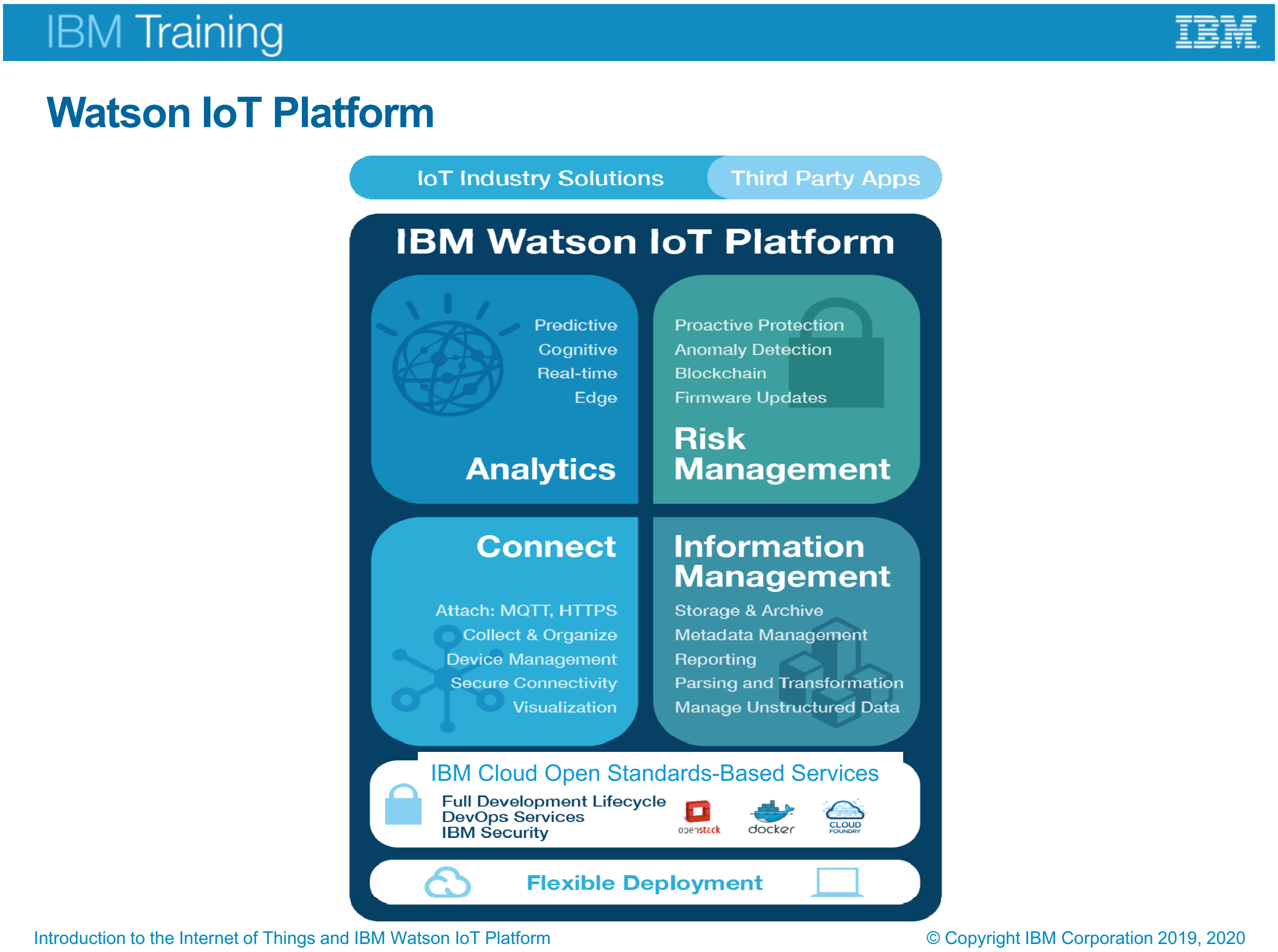


Figure 1-15. Watson IoT Platform

The Watson IoT Platform consists of four logical quadrants:

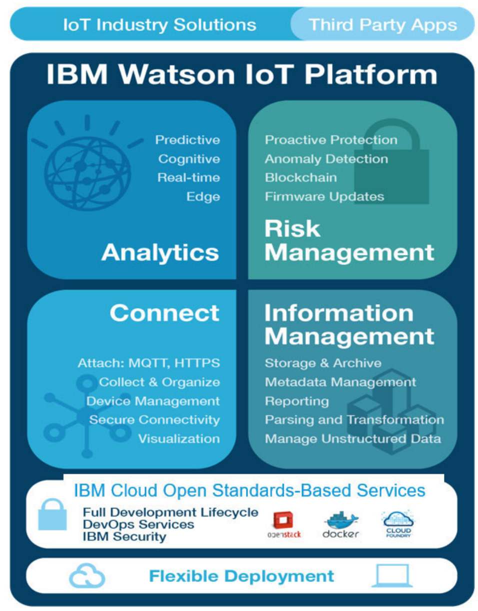
1. Analytics
2. Connect
3. Information Management
4. Risk Management

In the next slides, you explore the role of each quadrant and how they facilitate IoT solution development.

IBM Training

The four Watson IoT Platform quadrants

1. Connect: Connect and manage devices, networks, and gateways.
2. Information Management: Integrating structured and unstructured data from devices, people, and the world around us.
3. Analytics: Analyze and visualize real-time device data by using the Watson IoT Platform dashboards. Gain insights from data by using real-time data, predictive analytics, and AI by leveraging platform services.
4. Risk Management: You ensure that you leverage the correct information from trusted sources, that the correct software runs where you need it, and that your solution is visible end to end.



Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-16. The four Watson IoT Platform quadrants

Here are descriptions of the four Watson IoT Platform quadrants:

1. Connect: This quadrant's main concern is to manage the connection of IoT devices, networks, and gateways. There are many connection settings that you configure to achieve a robust and efficient connection of the devices. You might want to connect the devices directly to the Watson IoT Platform, or you might need the devices connected through a gateway. Because the growth of the network in some cases might be large or you need edge analytics, you probably make the connections through gateways; in other cases, connecting devices directly might be an effective choice.
2. Information Management: This quadrant is responsible for integrating structured and unstructured data from devices, people, and the world around us. In this quadrant, you design the way that the extracted information is managed. For example, according to your use case you might want to make the information persistent in an unstructured database like MongoDB or Cloudant databases, and in other cases you must make this information persistent in structured databases like IBM Db2 or Oracle. In many cases, these decisions are governed by processing or analytics that you perform later.
3. Analytics: This quadrant is where the value of deploying a Watson IoT Platform solution is gained because you can analyze and visualize real-time device data by using the Watson IoT Platform dashboards. Gain insights from data by using real-time data, predictive analytics, and AI by leveraging platform services.

4. Risk Management. You ensure that you leverage the correct information from trusted sources, that the correct software runs where you need it, and that your solution is visible end to end. Otherwise, your cloud might receive the wrong information and provide misleading insights. So, making sure that IoT devices are running well is vital to ensuring the correct operation of the system. In cases that include IoT devices with actuation capabilities, it might be dangerous if these devices start to actuate while they are malfunctioning.

1.4. Connecting devices to Watson IoT Platform

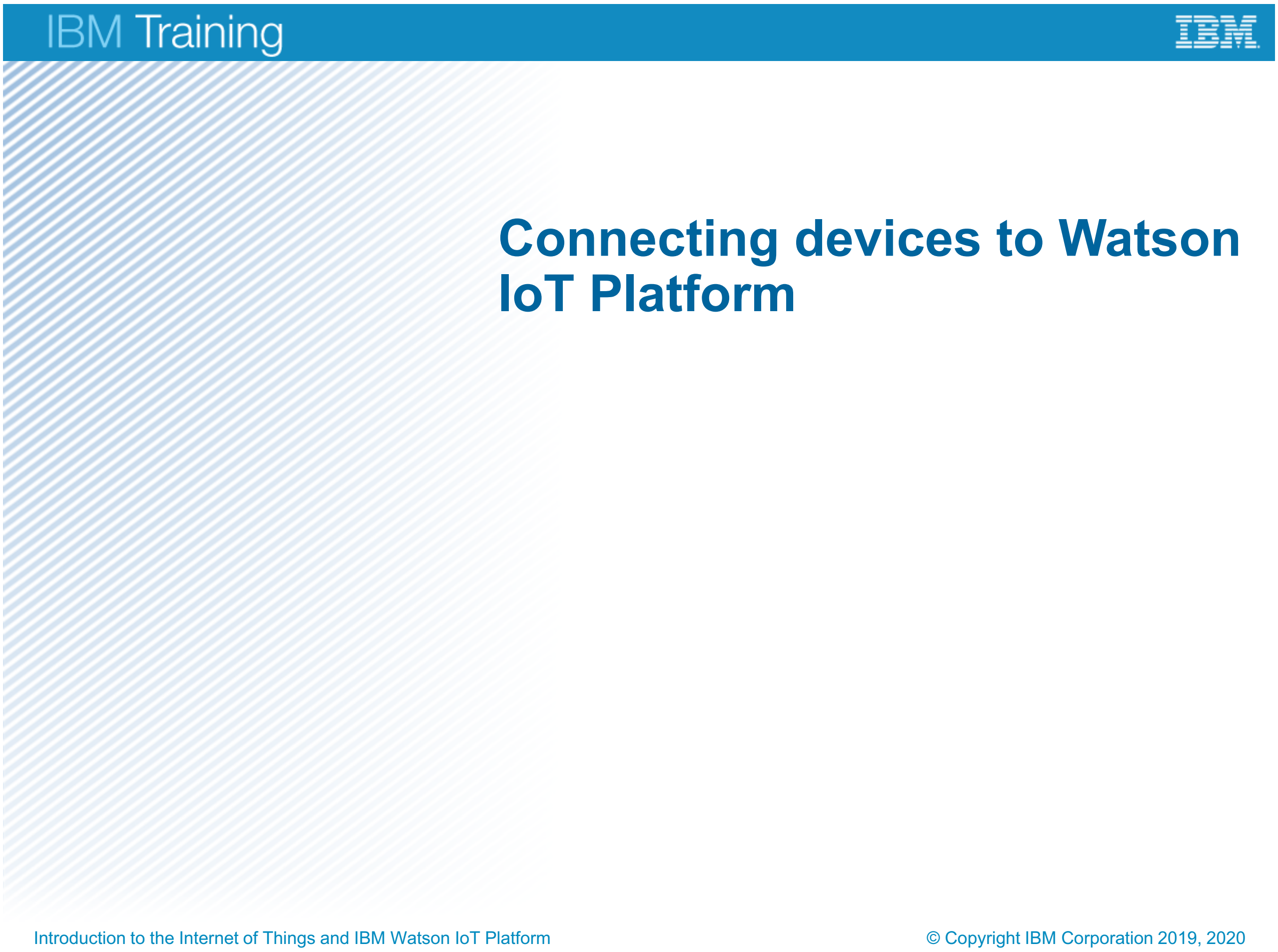


Figure 1-17. Connecting devices to Watson IoT Platform

IBM Training

Topics

- IBM IoT point of view
- IoT reference architecture
- Watson IoT Platform overview
- Connecting devices to Watson IoT Platform
- Communication protocols
- Introduction to IoT devices

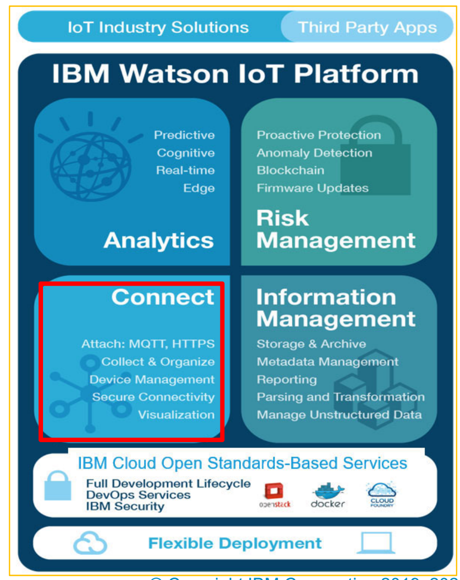
Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-18. Topics

IBM Training

Connecting

- Connect and manage your IoT devices, gateways, and networks by using a broad and growing infrastructure.
- Use open standards-based communication (MQTT and HTTPS).
- Secure your communication and management.
- Globally scalable starting with a single device.



The Hub for IoT devices

Figure 1-19. Connecting

Connect

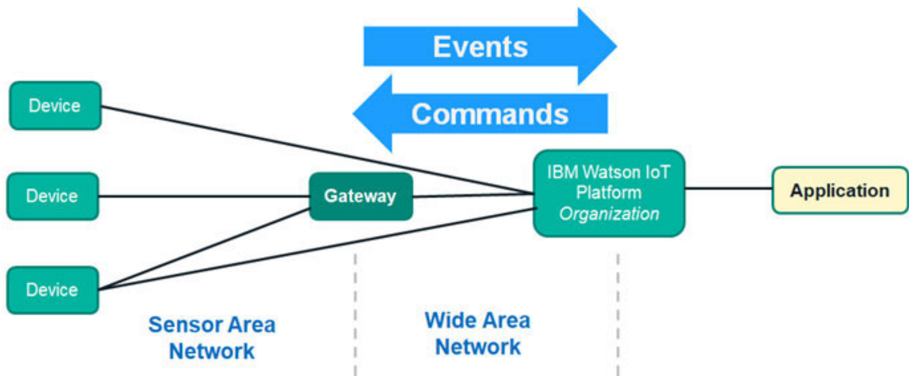
Import IoT data from any source by using Watson IoT Platform Service to register, connect, and control your IoT devices. Enable secure IoT data consumption and device control by using the Watson IoT Platform Registration Service.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/overview/overview.html

IBM Training

Connecting (cont.)



Bluetooth	**MQTT**
ZigBee	**HTTP**

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-20. Connecting (cont.)

The devices can be connected directly to the platform or though a gateway, depending on the use case:

- The *Sensor Area Network* is the network protocols that are used to communicate the IoT sensor devices to the gateway. Examples include Bluetooth or ZigBee.
- *Events* represent certain sensor data value ranges that are measured. For example, an event can be a temperature sensor reading above 30 degrees. In many cases, devices do not have to send the sensor data to the cloud continuously but only when it matters, such as when an event occurs. Usually, the events are sent from the devices to the cloud.
- *Commands* are sent from Watson IoT Platform to the devices. Commands might be actuation commands that turn off some devices or they might be diagnostic commands so that devices can report malfunctions.

1.5. Communication protocols

IBM Training

Communication protocols

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-21. Communication protocols

IBM Training

Topics

- IBM IoT point of view
- IoT reference architecture
- Watson IoT Platform overview
- Connecting devices to Watson IoT Platform
- Communication protocols
- Introduction to IoT devices

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-22. Topics

IBM Training

Watson IoT Platform communication protocols

- *Message Queuing Telemetry Transport (MQTT)* is a machine-to-machine (M2M) IoT connectivity protocol:
  - It follows the publish and subscribe protocol.
  - It runs over TCP/IP.
- *Hypertext Transfer Protocol (HTTP)* is an application protocol that you use to enable communication between physically dispersed systems. It is one of the most used protocols to transfer data over the internet.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-23. Watson IoT Platform communication protocols

MQTT is the primary protocol that devices and applications use to communicate with Watson IoT Platform. MQTT is a publish and subscribe messaging transport protocol that enables the efficient exchange of real-time data between sensor and mobile devices.

MQTT runs over TCP/IP. Although it is possible to code directly to TCP/IP, you also may choose to use a library that handles the details of the MQTT protocol for you. A wide range of MQTT client libraries are available. IBM contributes to the development and support of several client libraries.

HTTP is also available to connect devices to Watson IoT Platform by using the HTTP REST API. HTTP APIs provide the following features:

- Device management
- Application management
- User management
- Access Management

References:

- http://mqtt.org/
- https://cloud.ibm.com/docs/services/IoT/reference/mqtt/index.html#ref-mqtt

IBM Training

Other IoT communication protocols

- Advanced Message Queuing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware.
- Constrained Application Protocol (CoAP) is a specialized web transfer protocol that you use with constrained nodes and constrained networks for IoT devices.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-24. Other IoT communication protocols

Advanced Message Queuing Protocol (AMQP) enables encrypted and interoperable messaging between organizations and applications. The protocol is used in client/server messaging and IoT device management. It is used in some IoT platforms such as Microsoft Azure IoT Platform.

CoAP is used by different IoT platforms to transfer data. CoAP is used for M2M applications, such as smart energy and building automation.

References:

- https://www.amqp.org/
- http://coap.technology/

1.6. Introduction to IoT devices

IBM Training

Introduction to IoT devices

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-25. Introduction to IoT devices

IBM Training

Topics

- IBM IoT point of view
- IoT reference architecture
- Watson IoT Platform overview
- Connecting devices to Watson IoT Platform
- Communication protocols
- Introduction to IoT devices

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-26. Topics

IBM Training

Introduction to IoT devices

- The “thing” in IoT is any device with embedded electronics that can transfer data over a network without any human interaction.
- The device that can be used in IoT is a computer system contains the following components:
  - Processor
  - Memory
  - Input/output peripheral devices
  - Internet connection device
- Examples of the most used embedded system boards for experimental and educational purposes are:
  - Arduino boards
  - Raspberry Pi boards

Introduction to the Internet of Things and IBM Watson IoT Platform

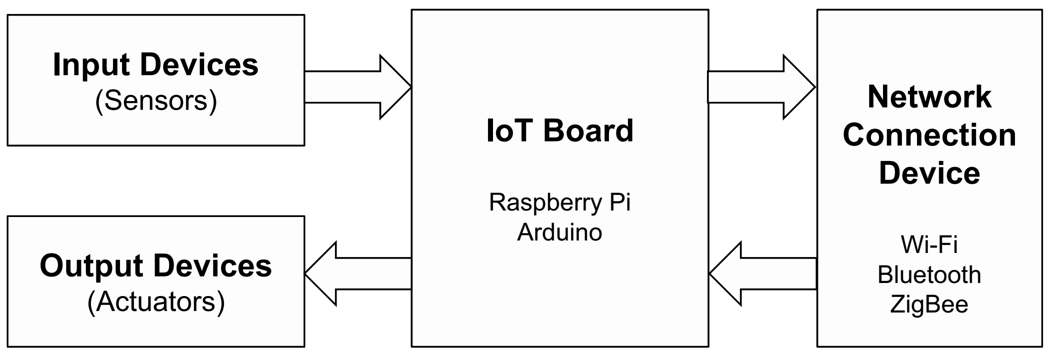
Figure 1-27. Introduction to IoT devices

Reference:

https://en.wikipedia.org/wiki/Embedded_system

IBM Training

IoT device overview



Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-28. IoT device overview

An IoT device has four main components:

1. Input sensors to measure the needed physical environment variables.
2. Output actuators to control the physical environment, such as moving objects and raising the temperature.
3. The network connection device to allow the device to share its collected data with the cloud. This device may be part of the IoT board like Raspberry Pi boards or an external add-on board like Arduino boards.
4. The IoT board collects the signals from the sensors, controls the actuators, and sends collected data and the device status to the network.

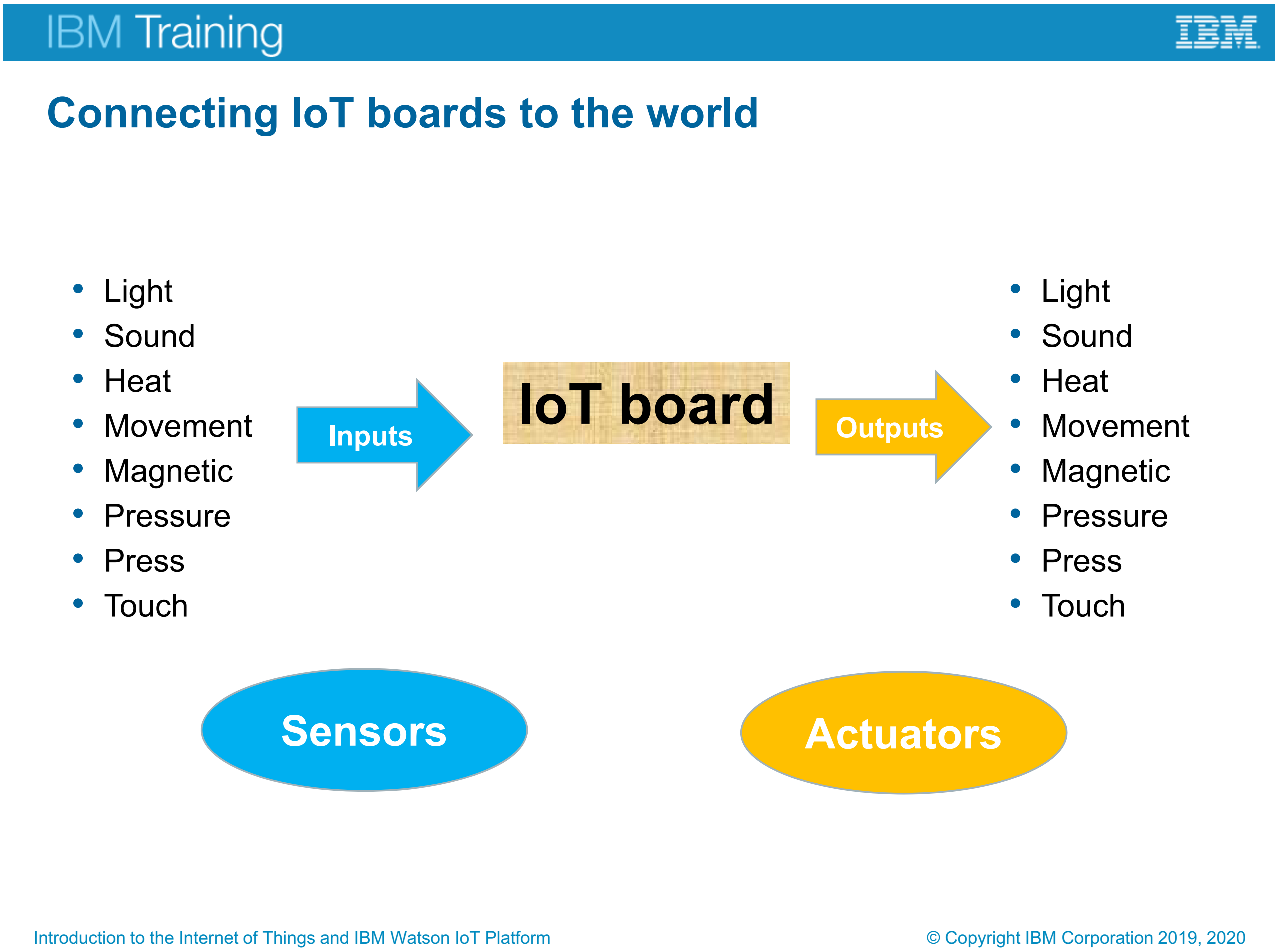


Figure 1-29. Connecting IoT boards to the world

IoT boards can collect different types of signals from the environment by using different types of sensors, and they can make similar changes to the environment by using the actuators.

IBM Training

Analog and digital signals

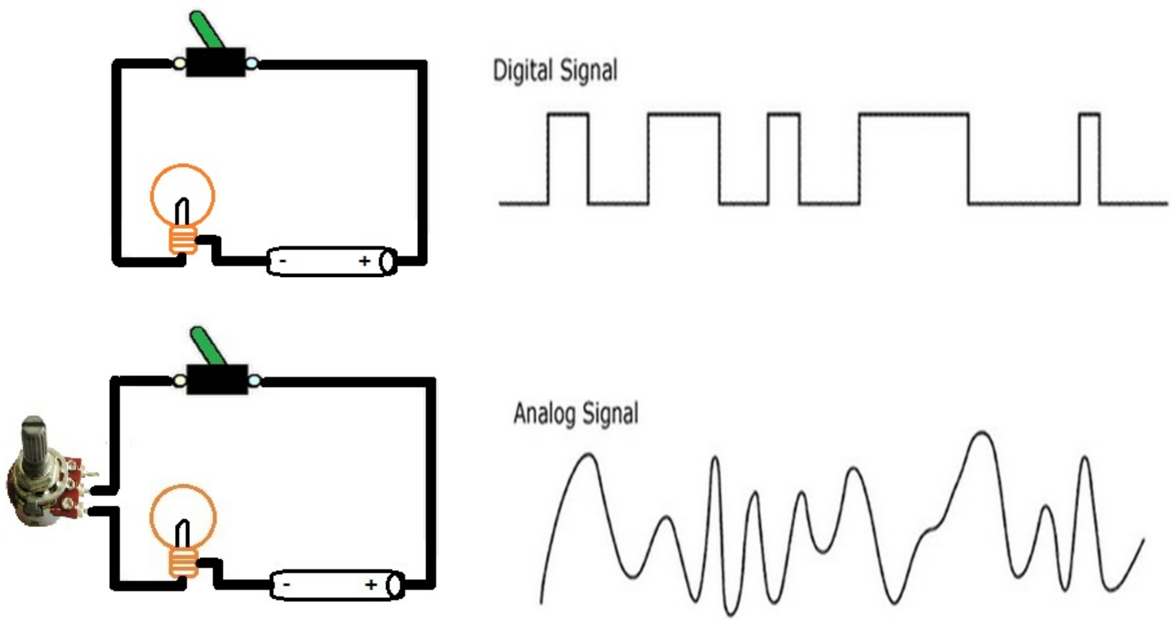


Figure 1-30. Analog and digital signals

Digital signal: We can represent the digital signal as the variations that happen in the voltage in a simple electric circuit when we turn it on or off.

Analog signal: If we add a potentiometer to the basic electric circuit and there is a change in the amount of voltage over time according to the position of the potentiometer, then this sign is an analog signal.

Nearly all physical phenomena around us are analog because they vary over time, like sound, temperature, and pressure.

IBM Training

Input devices: Sensors

There are many devices of different types that can be connected to an Arduino board to changes its inputs.

Switch sensors

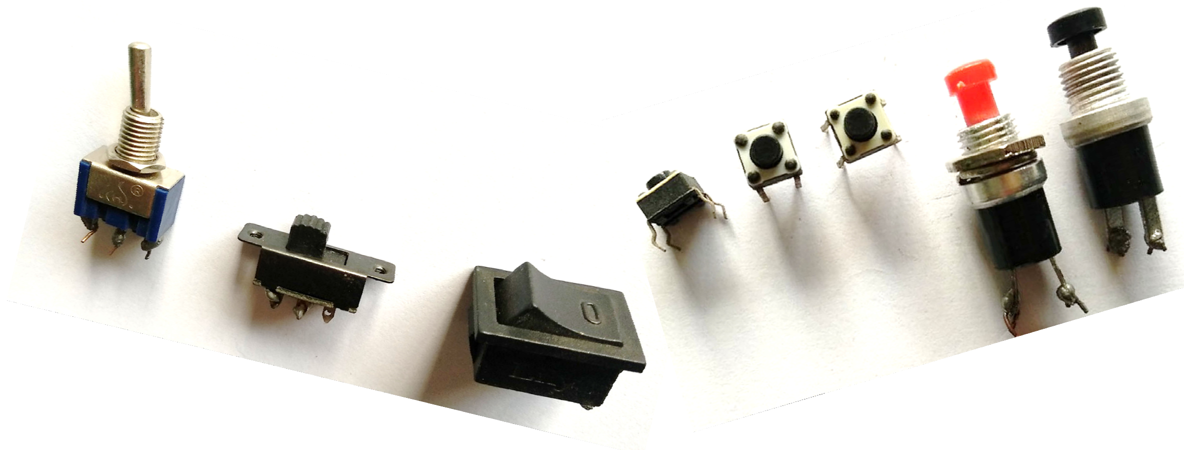


Figure 1-31. Input devices: Sensors

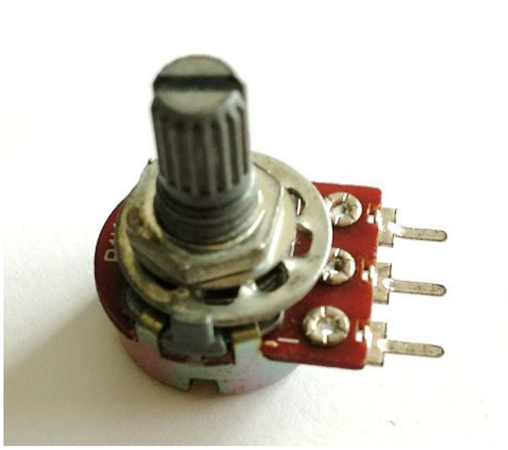
There are many types of devices that can connect to an Arduino board to change its inputs.

With *switch sensors*, the board can the sense the availability (on) or absence (off) of an electric voltage or current.

IBM Training

Input devices: Sensors (cont.)

Position sensors (variable resistors)



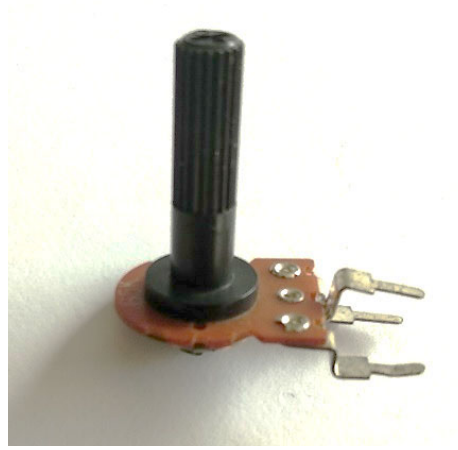


Figure 1-32. Input devices: Sensors (cont.)

With a *variable resistor*, an Arduino board can receive inputs as an analog signal, which simulates normal physical environment signals.

IBM Training

Input devices: Sensors (cont.)

Light sensors

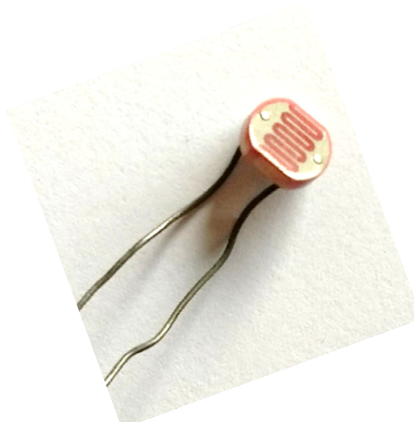


Figure 1-33. Input devices: Sensors (cont.)

With a light sensor, the Arduino board can detect the light or colors of the environment and act to control the devices.

IBM Training

Input devices: Sensors (cont.)

Sound sensors

Introduction to the Internet of Things and IBM Watson IoT Platform

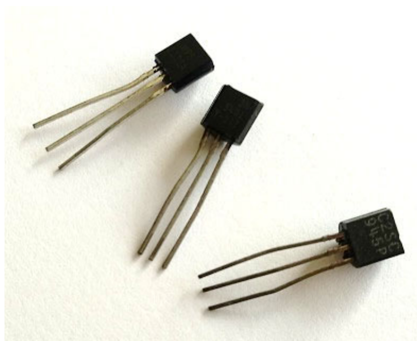
Figure 1-34. Input devices: Sensors (cont.)

With sounds sensors, the Arduino board can sense voice and sounds around the device.

IBM Training

Input devices: Sensors (cont.)

Heat sensors



Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-35. Input devices: Sensors (cont.)

With heat sensors, the Arduino board can sense the temperature of the device and the environment around it to control the temperature of the environment or protect the device itself.

IBM Training

Output devices: Actuators

There are many different types of devices that can be connected to Arduino boards to show changes in Arduino outputs.

Light sources

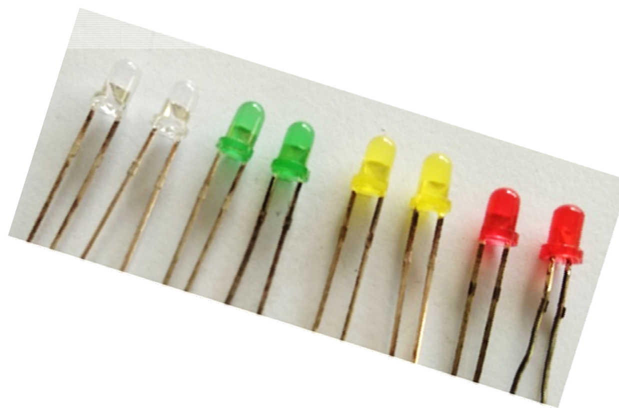


Figure 1-36. Output devices: Actuators

By using an Arduino board, you can control light sources and indicators.

IBM Training

Output devices: Actuators (cont.)

Display screens: Seven-segment and dot matrix

Introduction to the Internet of Things and IBM Watson IoT Platform

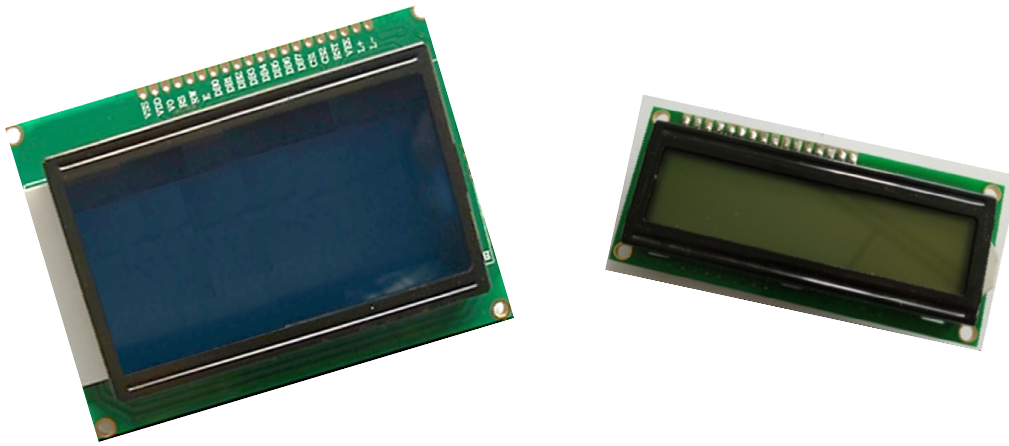
Figure 1-37. Output devices: Actuators (cont.)

By using an Arduino board, you can display numbers or even show simple graphics.

IBM Training

Output devices: Actuators (cont.)

Display screens: Text and graphics LCD



Introduction to the Internet of Things and IBM Watson IoT Platform

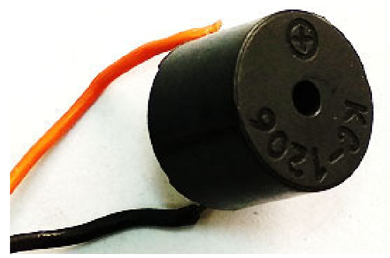
Figure 1-38. Output devices: Actuators (cont.)

By using an Arduino board, you can display text or graphics in full color.

IBM Training

Output devices: Actuators (cont.)

Sound: Speakers and buzzers



Introduction to the Internet of Things and IBM Watson IoT Platform

© Copyright IBM Corporation 2019, 2020

Figure 1-39. Output devices: Actuators (cont.)

By using an Arduino board, you can generate tones and sounds and replay voice or music.

IBM Training

Output devices: Actuators (cont.)

Movement: DC, steppers, and servo motors

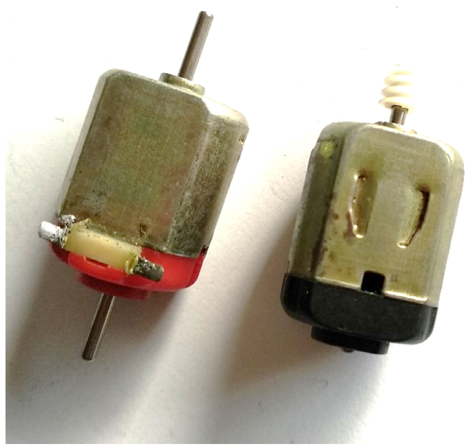


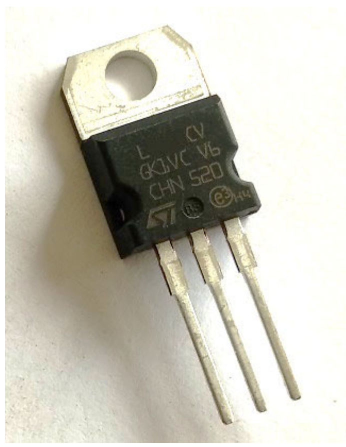
Figure 1-40. Output devices: Actuators (cont.)

By using an Arduino board, you can control moving objects and make robots move.

IBM Training

Output devices: Actuators (cont.)

Electricity: Transistor or relay



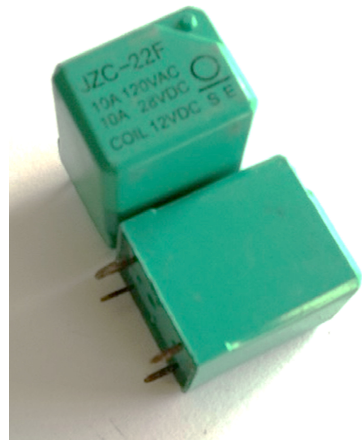


Figure 1-41. Output devices: Actuators (cont.)

By using an Arduino board, you can control electrical devices.

IBM Training

IoT boards: Arduino boards

- *Arduino* is an open source electronics hardware and software platform.
- Arduino boards can read inputs from sensors (such as light, sound, or pressure) and turn them into outputs (motor movement, turning on LEDs, and publishing something online).
- Arduino was created at the Ivrea Interaction Design Institute as a means for students without a background in electronics and programming to do fast prototyping.
- For more information about the Arduino board shape and specs, see the following website:
  https://store.arduino.cc/usa/arduino-uno-rev3

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-42. IoT boards: Arduino boards

Reference:

www.arduino.cc

IBM Training

IoT boards: Why Arduino boards

- Simple and powerful: Easy to use; anyone can learn to use it. It can be used in powerful applications.
- Inexpensive: Arduino boards are relatively inexpensive compared to other microcontroller platforms.
- Cross-platform: The Arduino integrated development environment (IDE) runs on Windows, Macintosh OSX, and Linux operating systems.
- Open source and extensible software: As open source tools, Arduino tools are available for extension by experienced programmers.
- Open source and extensible hardware: As open hardware, experienced circuit designers can improve and extend the hardware design.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-43. IoT boards: Why Arduino boards

Reference:

www.arduino.cc

IBM Training

IoT boards: Why Arduino boards (cont.)

- Compatibility and scalability: You can easily expand your code, memory, processor, and I/Os by scaling up the hardware board according to your project needs due to the compatibility between the different Arduino boards.
- Built in power supply: No need for an external power supply for most applications.
- Built in ROM flasher: No need for an external programming circuit to flash (download) your code into the Arduino board.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-44. IoT boards: Why Arduino boards (cont.)

Reference:

www.arduino.cc

IBM Training

IoT boards: Arduino boards types

- Arduino UNO
  - A microcontroller board with 14 digital I/O pins
  - Six analog inputs
  - 32 KB flash memory
  - USB connection
  - Power jack
  - https://store.arduino.cc/usa/arduino-uno-rev3

- Arduino NANO
  - A small, complete, and breadboard (a temporary work board for electronic circuits) friendly board
  - 22 digital I/O pins
  - Eight analog inputs
  - 32 KB flash memory
  - https://store.arduino.cc/usa/arduino-nano

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-45. IoT boards: Arduino boards types

Reference:

www.arduino.cc

IBM Training

IoT boards: Arduino boards types (cont.)

- Arduino MEGA
  - A microcontroller board with 54 digital I/O pins
  - Sixteen analog inputs
  - 256 KB flash memory
  - USB connection
  - Power jack
  - https://store.arduino.cc/usa/mega-2560-r3

- For more information, go to the following website: https://www.arduino.cc/en/Main/Products

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-46. IoT boards: Arduino boards types (cont.)

Reference:

www.arduino.cc

IBM Training

IoT boards: Arduino shields

- Arduino shields are elements that can be plugged onto a board to give it extra features.
- There are many Arduino shields that allow Arduino boards to interact with the real world through different physical sensors.
- By using Arduino shields, we can have Arduino connectivity to the internet to use the Arduino boards in IoT (most Arduino boards not have internet connectivity by default).

Figure 1-47. IoT boards: Arduino shields

Reference:

www.arduino.cc

IBM Training

IoT Boards: Arduino shield examples

- Arduino Ethernet Shield: By using this shield, you can connect your Arduino board to the internet by plugging this module onto an Arduino board, connecting the shield to your network by using an RJ45 cable, and completing a few simple instructions to start controlling your world through the internet. For more information, see the following website:
  https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1

- Arduino Motor Shield: By using this shield, your Arduino board can activate and use DC and stepper motors, relays, and solenoids. For more information, see the following website:
  https://store.arduino.cc/usa/arduino-motor-shield-rev3

- Arduino LCD Screen: This shield is a backlit LCD screen with headers that can draw text, images, and shapes to the screen. For more information, see the following website:
  https://store.arduino.cc/usa/arduino-lcd-screen

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-48. IoT Boards: Arduino shield examples

Reference:

www.arduino.cc

IBM Training

IoT Boards: Arduino shield examples (cont.)

- Arduino GSM Shield: By using this shield, Arduino boards can make phone calls, send SMS messages, and connect to the internet. For more information, go to the following website:
  https://www.arduino.cc/en/Main.ArduinoGSMShieldV1
- Arduino Wi-Fi Shield: By using this shield, you can connect an Arduino board to the internet wirelessly. For more information, see the following website:
  https://store.arduino.cc/usa/arduino-wifi-shield
- Arduino 4 Relays Shield: By using this shield, your Arduino board can drive high-power loads. For more information, see the following website:
  https://store.arduino.cc/usa/4-relays-shield

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-49. IoT Boards: Arduino shield examples (cont.)

Reference:

www.arduino.cc

IBM Training

IoT boards: Raspberry Pi boards

- Raspberry Pi is a series of small, single-board computers that were developed to promote the teaching of basic computer science in schools and developing countries.
- Several generations of Raspberry Pi were released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).
- The processor speed is 700 MHz - 1.5 GHz for the Pi 4, and the onboard memory is 256 MB - 4 GB on the Pi 4.
- Secure Digital (SD) cards are used to store the operating system and program memory.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-50. IoT boards: Raspberry Pi boards

References:

- https://en.wikipedia.org/wiki/Raspberry_Pi
- www.raspberrypi.org

IBM Training

IoT boards: Raspberry Pi boards (cont.)

- The boards have 1 - 5 USB ports.
- For video output, HDMI and composite video are supported, with a standard 3.5 mm tip-ring-sleeve jack for audio output.
- Lower-level output is provided by several General Purpose Input/Output (GPIO) pins, which support common protocols like I^2C.
- The B-models have an Ethernet port, and the Pi 3, Pi 4, and Pi Zero W have onboard Wi-Fi 802.11n and Bluetooth.
- Prices are $5 - $55.
- For more information, see the following website: https://www.raspberrypi.org/products/

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-51. IoT boards: Raspberry Pi boards (cont.)

References:

- https://en.wikipedia.org/wiki/Raspberry_Pi
- www.raspberrypi.org

IBM Training

IoT boards: Raspbian

- *Raspbian* is the official operating system for *all* models of Raspberry Pi.
- Raspbian is based on Debian Linux that is optimized for the Raspberry Pi hardware. Raspbian contains the set of basic programs and utilities that make Raspberry Pi run.
- Raspbian comes with over 35,000 packages: Precompiled software that is bundled in a nice format for easy installation on Raspberry Pi.
- For more information about Raspberry Pi boards, go to the following website:
  https://www.raspberrypi.org/documentation/raspbian/

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-52. IoT boards: Raspbian

Reference:

https://www.raspberrypi.org/documentation/raspbian/

IBM Training

IoT boards: Raspberry Pi Sense HAT

- *Sense HAT* is an add-on board for Raspberry Pi that was made especially for the Astro Pi mission, which was sent to the International Space Station in December 2015. Sense HAT is now available to buy.

- Sense HAT has an 8 × 8 RGB LED matrix, a five-button joystick, and includes the following sensors:
  - Gyroscope
  - Accelerometer
  - Magnetometer
  - Temperature
  - Barometric pressure
  - Humidity

- For more information about Raspberry Pi Sense HAT, go to the following website:
  https://www.raspberrypi.org/products/sense-hat/

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-53. IoT boards: Raspberry Pi Sense HAT

References:

- https://en.wikipedia.org/wiki/Raspberry_Pi
- www.raspberrypi.org

IBM Training

Unit summary

- Explain the IBM perspective for Internet of Things (IoT) solution development.
- Describe the IoT reference architecture.
- Describe the characteristics of Watson IoT Platform that simplify the development of IoT solutions.
- Describe the four quadrants of Watson IoT Platform.
- Explain how to connect devices to the Watson IoT Platform solution.
- List the communication protocols that are used in IoT solutions.
- Introduce common IoT devices.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-54. Unit summary

IBM Training

Review questions

1. True or False: IBM focuses on only embedded systems when it comes to providing an IoT solution.
2. A complete IoT solution can help in which of the following situations:
   A. Improve business outcomes through connected operations and connected products.
   B. Grow a business by using new services and business models.
   C. Create and connect IoT devices and analyze data.
   D. A, B, and C are correct.
3. The Information Management quadrant in Watson IoT Platform is responsible for which of the following items:
   A. Managing the connection of IoT devices.
   B. Integrating structured and unstructured data from devices, people, and the world around us.
   C. Analyzing and visualizing real-time device data by using the Watson IoT Platform dashboards.
   D. Ensuring that you leverage the correct information from trusted sources.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-55. Review questions

IBM Training

IBM

Review questions (cont.)

4. The Risk Management quadrant in Watson IoT Platform is responsible for which of the following items:
   A. Managing the connection of IoT devices.
   B. Integrating structured and unstructured data from devices, people, and the world around us.
   C. Analyzing and visualizing real-time device data by using the Watson IoT Platform dashboards.
   D. Ensuring that you leverage the correct information from trusted sources.
5. True or False: Watson IoT Platform does not support open standards like HTTPS and MQTT.
6. True or False: MQTT is the primary protocol that devices and applications use to communicate with Watson IoT Platform.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-56. Review questions (cont.)

IBM Training

Review answers

1. True or False: IBM focuses on only embedded systems when it comes to providing an IoT solution.
2. A complete IoT solution can help in which of the following situations:
   A. Improve business outcomes through connected operations and connected products.
   B. Grow a business by using new services and business models.
   C. Create and connect IoT devices and analyze data.
   D. A, B, and C are correct.
3. The Information Management quadrant in Watson IoT Platform is responsible for which of the following items:
   A. Managing the connection of IoT devices.
   B. Integrating structured and unstructured data from devices, people, and the world around us.
   C. Analyzing and visualizing real-time device data by using the Watson IoT Platform dashboards.
   D. Ensuring that you leverage the correct information from trusted sources.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-57. Review answers

1. False
2. D
3. B

IBM Training

Review answers (cont.)

4. The Risk Management quadrant in Watson IoT Platform is responsible for which of the following items:
   A. Managing the connection of IoT devices.
   B. Integrating structured and unstructured data from devices, people, and the world around us.
   C. Analyzing and visualizing real-time device data by using the Watson IoT Platform dashboards.
   D. Ensuring that you leverage the correct information from trusted sources.
5. True or False: Watson IoT Platform does not support open standards like HTTPS and MQTT.
6. True or False: MQTT is the primary protocol that devices and applications use to communicate with Watson IoT Platform.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-58. Review answers (cont.)

4. D
5. False
6. True

IBM Training

Exercise: Designing an IoT solution

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-59. Exercise: Designing an IoT solution

IBM Training

Exercise objectives

- In this exercise, you design a complete IoT solution by using IBM Watson IoT Platform.
- After completing this exercise, you should be able to:
  - Choose IoT sensors that best match the IoT overall solution.
  - Identify the best possible ways to connect an IoT device to Watson IoT Platform.
  - Explain how data that is sent from an IoT device can be managed and used by Watson IoT Platform.

Introduction to the Internet of Things and IBM Watson IoT Platform

Figure 1-60. Exercise objectives

Unit 2. Introduction to IBM Cloud

Estimated time

01:30

Overview

This unit provides a high-level overview of IBM Cloud. This unit introduces basic concepts such as the service catalog, cloud applications, and cloud services. This unit describes the Watson IoT Platform service and its components.

IBM Training

Unit objectives

- Describe the key capabilities of IBM Cloud.
- List services available in the IBM Cloud Catalog.
- List databases available on IBM Cloud.
- Manage IBM Cloud users and resources.
- Manage applications and services on IBM Cloud.
- Explain how to get started with the IBM Watson IoT Platform service on IBM Cloud.

Introduction to IBM Cloud

Figure 2-1. Unit objectives

2.1. IBM Cloud overview

Figure 2-2. IBM Cloud overview

IBM Training

Topics

- IBM Cloud overview
- Databases on IBM Cloud
- Managing users and resources on IBM Cloud
- Managing applications and services on IBM Cloud
- Getting started with the Watson IoT Platform service on IBM Cloud

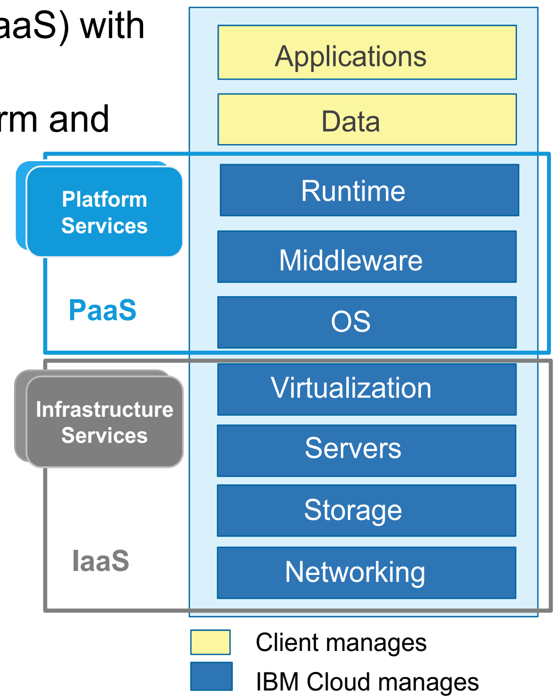
Introduction to IBM Cloud

Figure 2-3. Topics

IBM Training

What is IBM Cloud

- An open, standards-based cloud computing platform.
- Combines platform as a service (PaaS) with infrastructure as a service (IaaS).
- Includes a catalog of diverse platform and infrastructure services.
- Used to build, deploy, and run rapidly business applications, infrastructures, or both.



Introduction to IBM Cloud

Figure 2-4. What is IBM Cloud

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

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

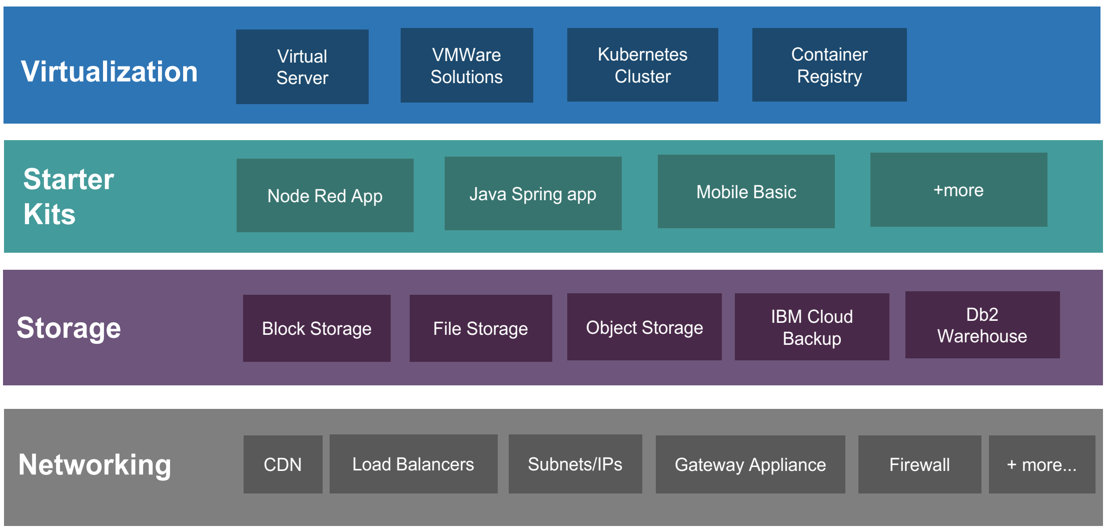
The application types can be web, mobile, big data, smart devices, and IoT.

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

IBM Training

IaaS 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, and networking.
- Provides services to deploy, access, and manage the infrastructure.



Introduction to IBM Cloud

Figure 2-5. IaaS from IBM Cloud

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

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

With IBM Cloud, you also can 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.

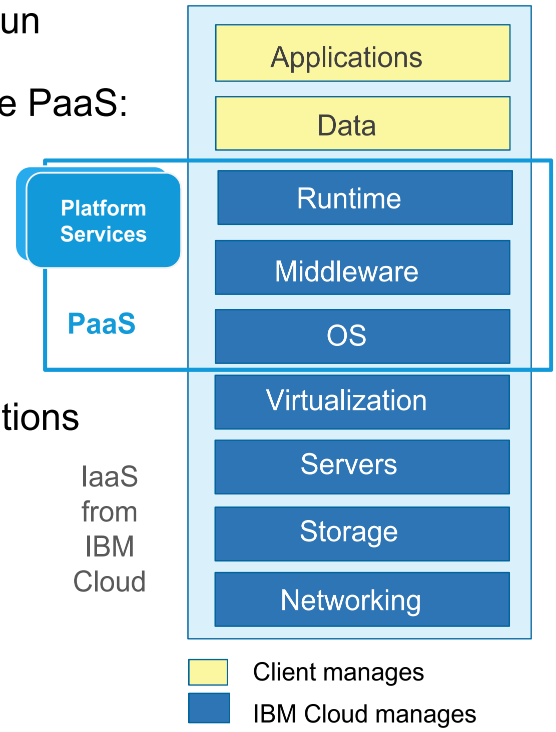
Note

The services that are available in IBM Cloud can change.

IBM Training

PaaS from IBM Cloud

- Enables you to build, manage, and run applications.
- Uses Cloud Foundry, an open source PaaS:
  - Extends Cloud Foundry with services from IBM and IBM Business Partners.
  - Provides a scriptable command-line interface (CLI).
  - Integrates with development tools, such as Eclipse, to ease the deployment process.
- Runs on IBM Cloud data center locations worldwide.



Introduction to IBM Cloud

Figure 2-6. PaaS 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 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:

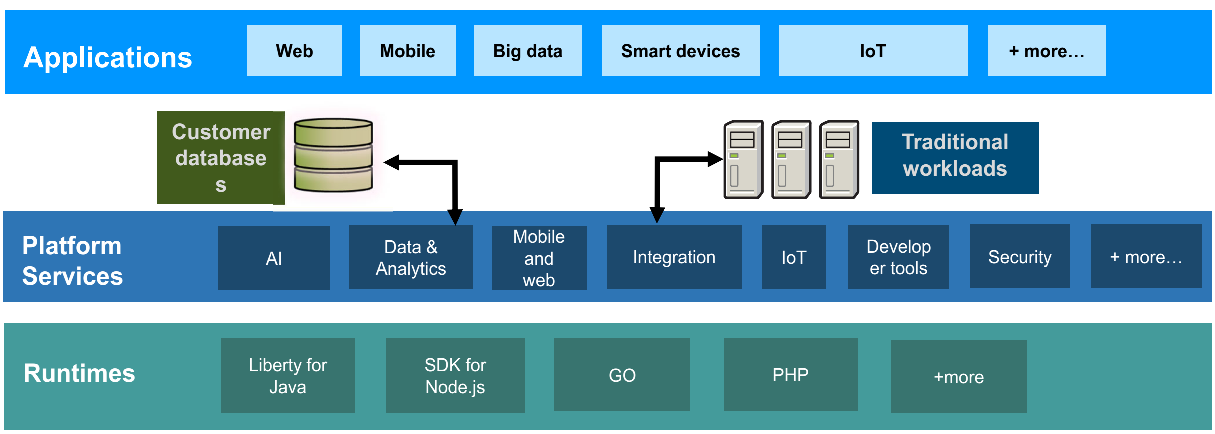
- Extends Cloud Foundry with services from IBM and IBM Business Partners.
- Provides a scriptable command-line interface (CLI).
- Provides integration with development tools to ease the deployment process. DevOps services provide an online code editor, a build pipeline, and a version control system.

IBM Cloud runs on IBM Cloud data centers locations worldwide.

IBM Training

PaaS 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.
- Integration with data and traditional workloads in on-premises systems.
- DevOps capabilities and tools.



Introduction to IBM Cloud

Figure 2-7. PaaS from IBM Cloud (cont.)

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

- Runtimes on which to run Cloud Foundry applications, such as Liberty for Java, SDK for Node.js, Go, and PHP.
- A catalog of selectable services that are used to build applications:
  - Databases: CloudantDB and Databases for PostgreSQL
  - Mobile and web support: Mobile Foundation
  - Analytics: Streaming analysis and SQL Query
  - Artificial intelligence (AI): Watson Studio and Watson Assistant
  - Security: SSL certificates and AppID
- 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.

IBM Training

What is 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 deployment aspects of coding remain carefully coordinated with any attached services, which result in quick, consistent, and reliable iterating of applications.

Cloud Foundry
IBM • Compute

Run your Cloud Foundry application in either a multi-tenant, or an isolated environment (Cloud Foundry Enterprise).

Lite

Figure 2-8. What is Cloud Foundry

Cloud Foundry has a container-based architecture that runs apps in any programming language. You can deploy apps to Cloud Foundry by using your existing tools with zero modification to the code.

Cloud Foundry is an open source project with an open contribution and open governance model that gives users maximum flexibility to avoid vendor lock-in.

IBM Cloud offers the Cloud Foundry service to create and deploy Public Applications and Enterprise Environment with popular language runtimes.

References:

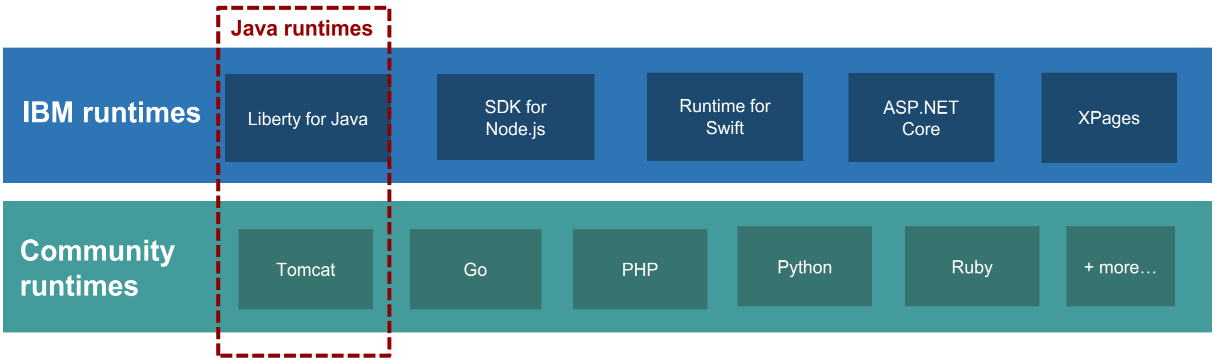
https://www.cloudfoundry.org/

https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-what-is-cloud-foundry

IBM Training

IBM Cloud: Choice of runtimes

- With IBM Cloud, you can run your app by using a runtime without needing to manage the underlying infrastructure.
- A *runtime* is a set of resources that is used to run an application:
  - You can choose the runtime on which to run your application (for example, Node.js or Swift).
  - For each runtime, an IBM or community buildpack provides the scripts to prepare your code to run on IBM Cloud.



Introduction to IBM Cloud

Figure 2-9. IBM Cloud: Choice of runtimes

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

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

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

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

The following runtimes are available for Java:

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

Note

The available runtimes in IBM Cloud can change.

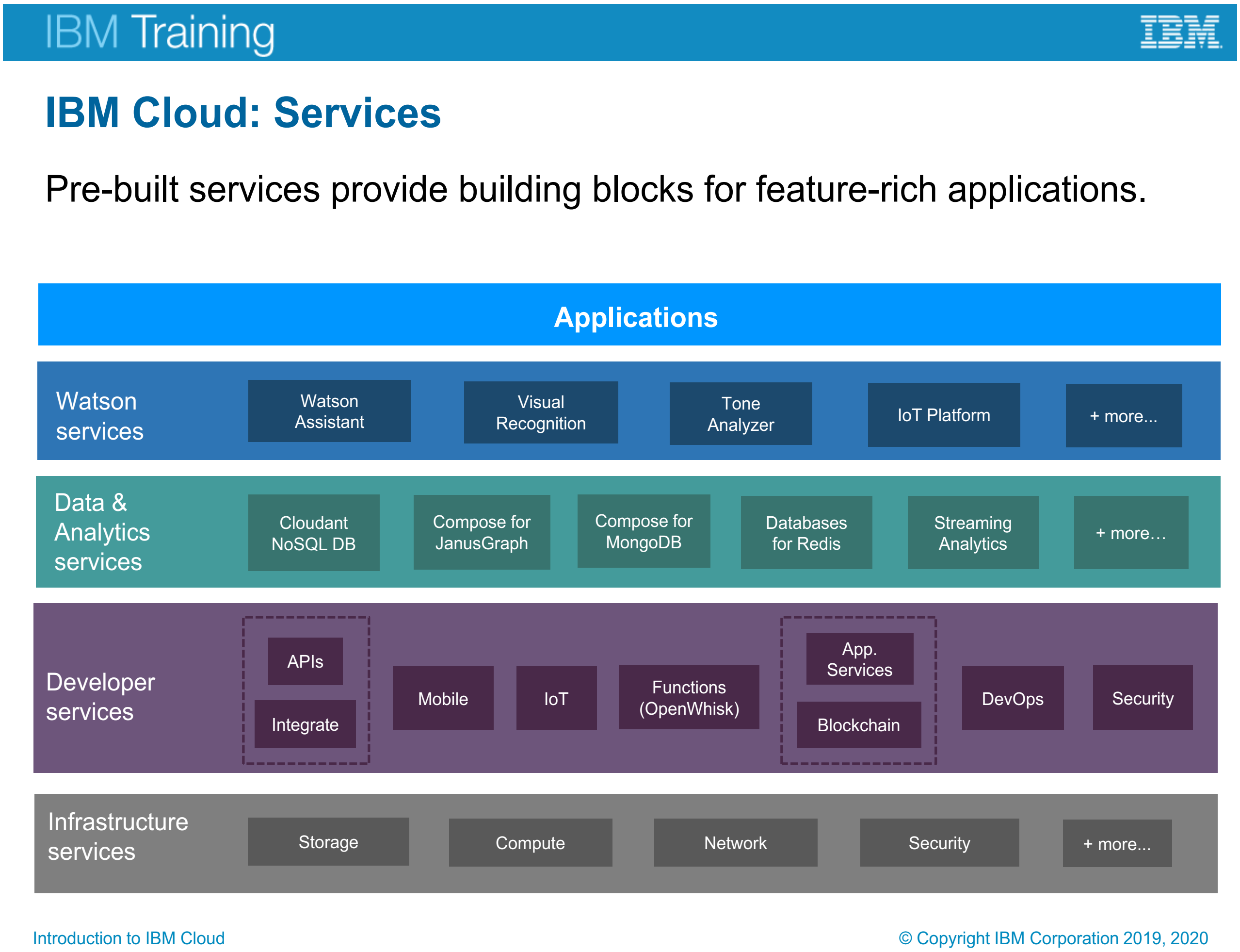


Figure 2-10. IBM Cloud: Services

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

With Watson services, you can add the power of AI to your application with speech, vision, and natural language processing (NLP) APIs.

Data & Analytics services help you 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 that are 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 blockchain, Message Hub, WebSphere Application Server, Business Rules, and other application services, are on the cloud.
- DevOps: Tools to help innovate new applications faster and cheaper.
- Security: Build security into your application design.

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

IBM Training

IBM Cloud: Regions and locations

- IBM Cloud is hosted worldwide.
- A region is a geographic area where your application is hosted:
  - Locations are data centers in a region where the hardware is deployed.
- Select a region to deploy your application:
  - The region nearest to users can provide better performance.
  - A specific region can meet data security requirements.
  - Multiple regions provide high availability.

IBM Cloud Kubernetes Service region	Corresponding IBM Cloud location	Single zone Location
AP North (standard clusters only)	Tokyo	Singapore, Chennai, Seoul, and Hong Kong S.A.R. of the PRC
AP South	Sydney	Melbourne
EU Central	Frankfurt	Amsterdam, Milan, Oslo, and Paris
UK South	London	
US East (standard clusters only)	Washington DC	Montreal and Toronto
US South	Dallas	San Jose and Sao Paulo

Introduction to IBM Cloud

Figure 2-11. IBM Cloud: Regions and locations

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

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

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

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

Note

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

Within the IBM Cloud console, the region is automatically set to the closest region. To switch to another region, click the user account link, expand the **Region** menu, and select the region. If you use a Lite account, you can use only one region for your applications and services.

You can organize your resources across IBM Cloud services by using IBM Cloud locations, which are also called regions. For example, you can create a Kubernetes cluster by using a private Docker image that is stored in your IBM Cloud Container Registry of the same location.

Note

Not all IBM Cloud services are available in all regions.

For more information, see:

https://cloud.ibm.com/docs/containers?topic=containers-regions-and-zones

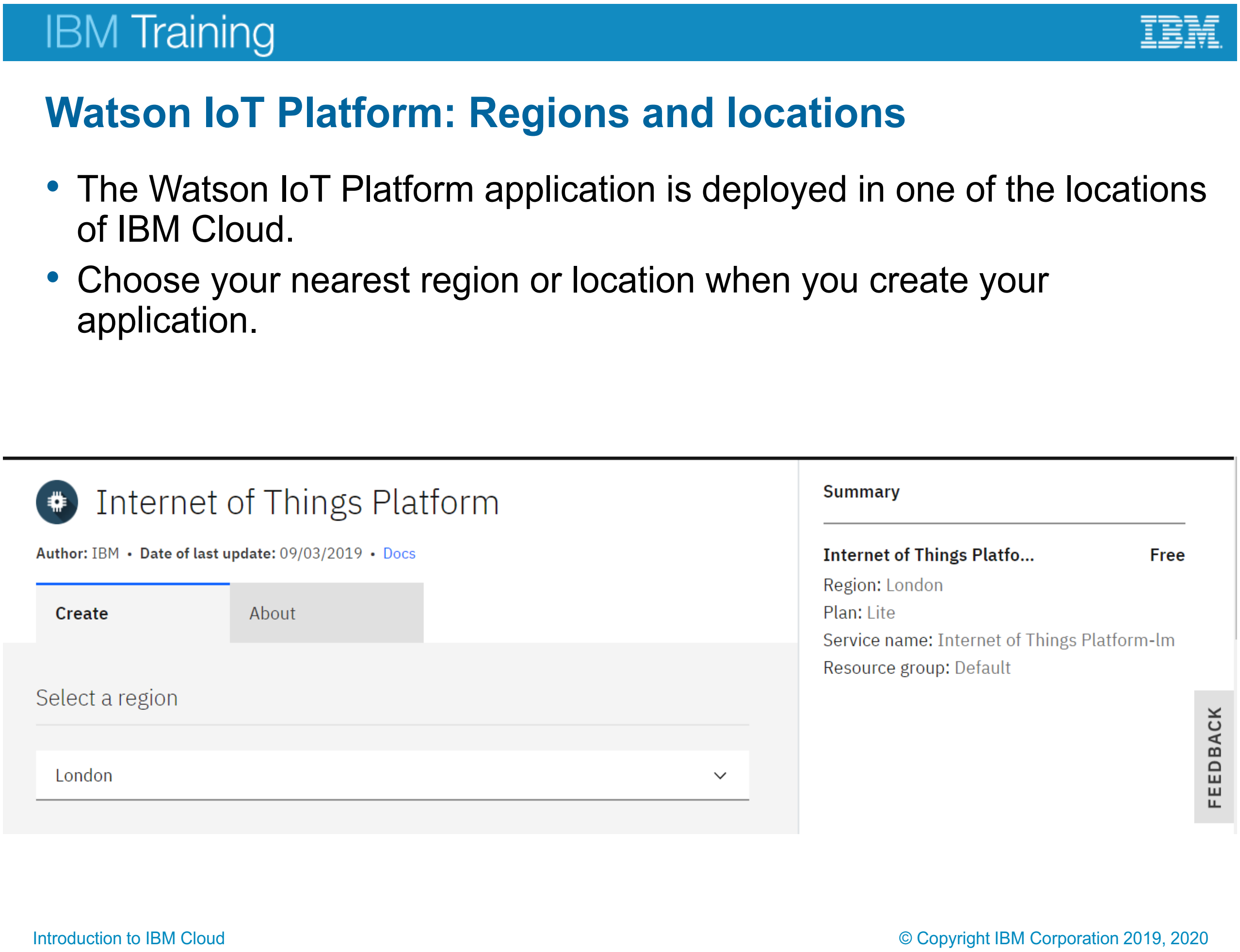


Figure 2-12. Watson IoT Platform: Regions and locations

You can choose from four different regions where you can deploy your Watson IoT application (Dallas, Frankfurt, Washington, D.C., and London).

Organizations can span multiple regions, and they are defined by the following items:

- Users: The role with basic permission in organizations and spaces. You must be assigned to an organization before you can be granted other permissions to the spaces within the organization.
- Domains: Provides the route on the internet that is allocated to the organization. A route has a subdomain and a domain. A subdomain is typically the application name. A domain might be a system domain or a custom domain that you registered for your application.
- Quota: Represents the resources that are available to an organization, including the number of services and the amount of memory that can be allocated for use by the organization. Quotas are assigned when organizations are created. Any application or service in a space within an organization contributes to the usage of the quota. With Pay-As-You-Go or Subscription accounts, you can adjust your quota for Cloud Foundry applications and containers as the needs of your organization change.

Adding spaces

Within an organization, you can use spaces to group a set of applications, services, and users. Spaces are tied to a specific region in IBM Cloud. You can create spaces in an organization based on the delivery lifecycle. For example, you can create a dev space as a development environment, a test space as a testing environment, and a production space as a production environment. Then, you can associate your apps with spaces. Each Cloud Foundry application can be assigned to only one space.

Reference:

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

2.2. Databases on IBM Cloud

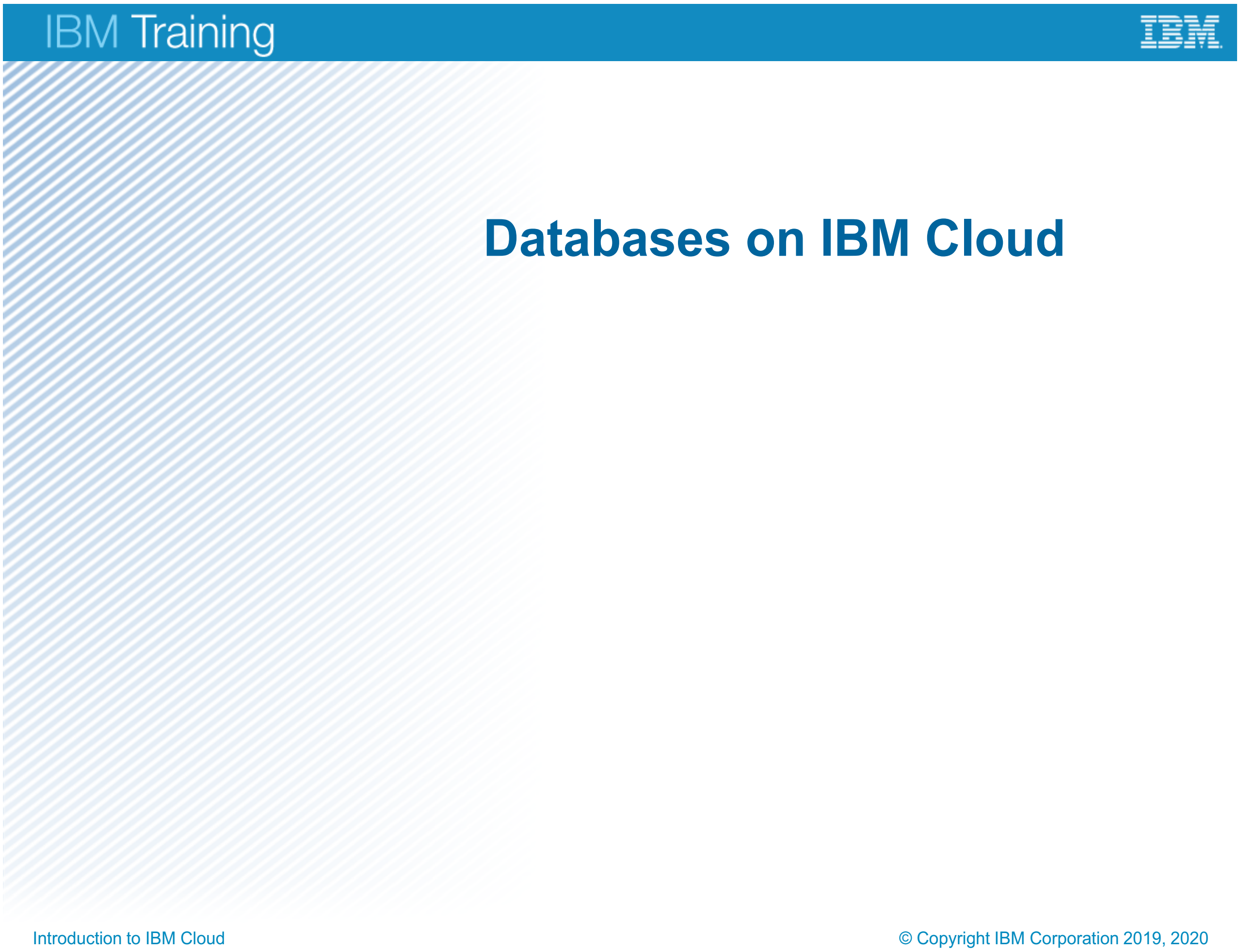


Figure 2-13. Databases on IBM Cloud

IBM Training

Topics

- IBM Cloud overview
- Databases on IBM Cloud
- Managing users and resources on IBM Cloud
- Managing applications and services on IBM Cloud
- Getting started with the Watson IoT Platform service on IBM Cloud

Introduction to IBM Cloud

Figure 2-14. Topics

IBM Training

Database choices on IBM Cloud

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

Introduction to IBM Cloud

Figure 2-15. Database choices on IBM Cloud

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

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

- Each column is stored in the following database records:
  - Ahmed: 1, Ben: 2, John: 3
  - 29:1, 34:2, 32:3
  - 65:1, 70:2, 73:3
- *Key-value pair data services* allow efficient storage of key-value pair data.

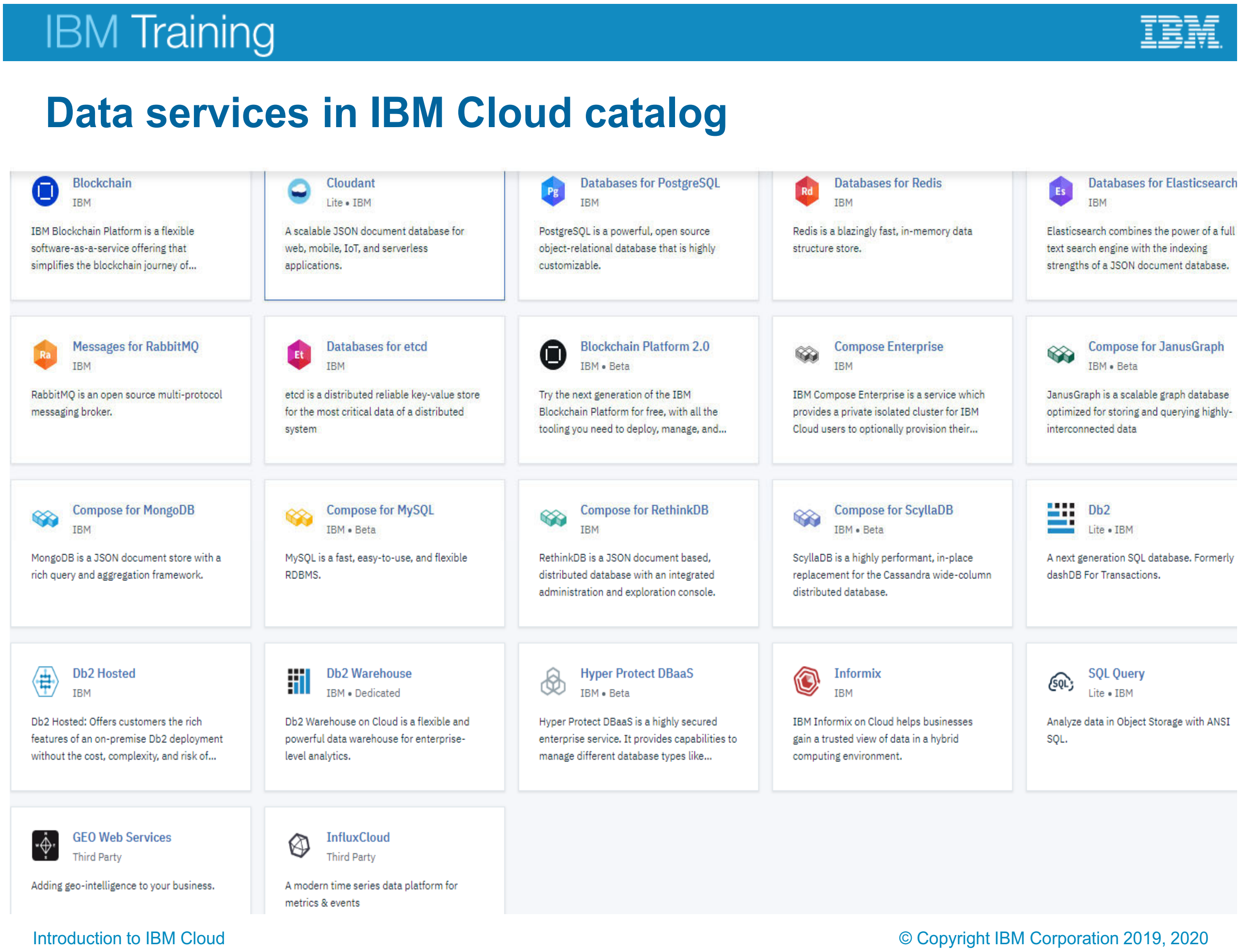


Figure 2-16. Data services in IBM Cloud catalog

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

- NoSQL database services, including Cloudant NoSQL DB and Compose for MongoDB.
- SQL database services. including Db2 Hosted, Compose for PostgreSQL, ElephantSQL, and Databases for PostgreSQL.
- Columnar database services, including Compose for ScyllaDB.
- Key-value pair in-memory data services, including Compose for Redis, which is an open source, fast key-value, and low-maintenance store.

2.3. Managing users and resources on IBM Cloud

Figure 2-17. Managing users and resources on IBM Cloud

IBM Training

Topics

- IBM Cloud overview
- Databases on IBM Cloud
- Managing users and resources on IBM Cloud
- Managing applications and services on IBM Cloud
- Getting started with the Watson IoT Platform service on IBM Cloud

Introduction to IBM Cloud

Figure 2-18. Topics

IBM Training

Managing users and resources on IBM Cloud

IBM Cloud offers two ways for access control for users:

- Identity and Access Management (IAM)
- Cloud Foundry access

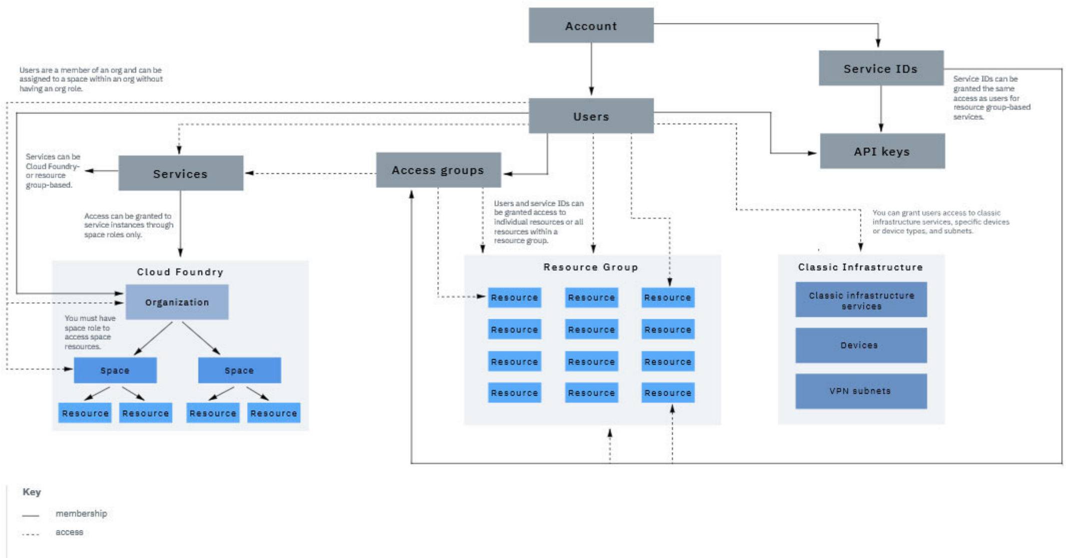


Figure 1. Account components and systems

Introduction to IBM Cloud

Figure 2-19. Managing users and resources on IBM Cloud

IBM Cloud offers to ways to grant access to resources

- Identity and Access Management (IAM): All services that are organized in a resource group in your account are managed by using IBM Cloud IAM. Account owners are automatically assigned the account administrator role for Cloud IAM. As the account administrator, you can assign and manage access for users, create resource groups, create access groups, view billing details and track usage, and create service instances. You provide access for users, service IDs, and access groups by creating policies that set a target for the subject of the policy to access and a role that defines what type of access that is allowed.
- Cloud Foundry access: Currently, not all services can be managed by using Cloud IAM. You can continue to use Cloud Foundry roles for access to these service instances. Users are added to the org and space to which the instance belongs with a Cloud Foundry role assigned.
  - Orgs enable collaboration among users and facilitate the logical grouping of project resources in the following ways:
    - You can group a set of spaces, apps, services, domains, routes, and users together in orgs.
    - You can manage the user access to the orgs and spaces on an individual basis.

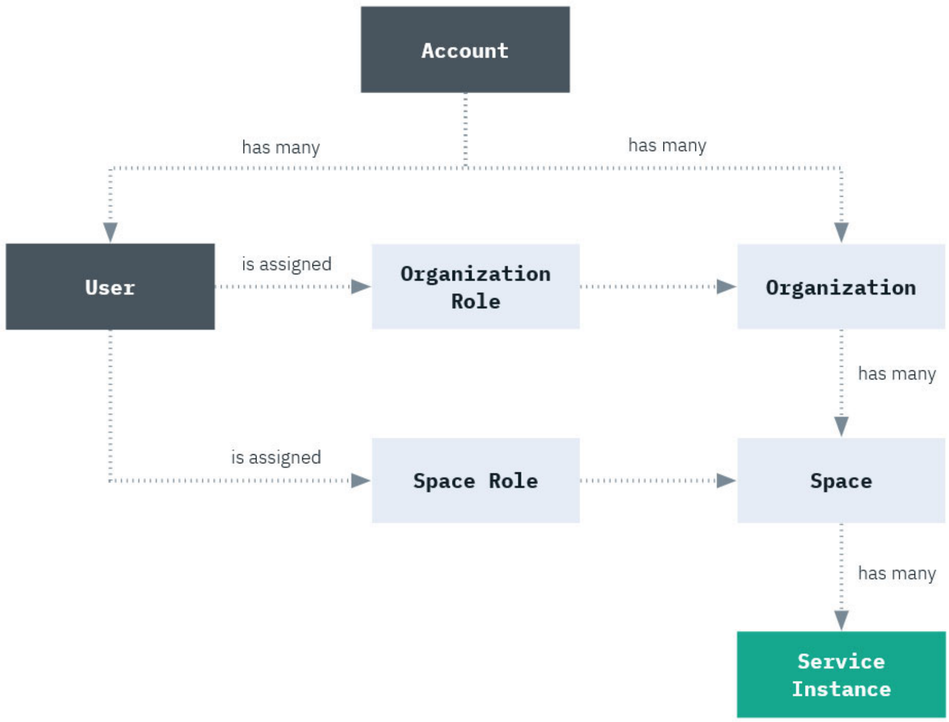
- Orgs can span multiple regions, and they are defined by the following items:
  - Spaces: A subgroup within an org that you can use to organize applications, services, and users. Spaces are tied to a specific region in IBM Cloud.
  - Users: The role with basic permission in orgs and spaces. You must be assigned to an org before you can be granted other permissions to the spaces within the org.
  - Domains: Provide the route on the internet that is allocated to the org. A route has a subdomain and a domain. A subdomain is typically the application name. A domain might be a system domain or a custom domain that you registered for your application.
  - Quota: Represents the resources that are available to an org, including the number of services and the amount of memory that can be allocated for use by the org. Quotas are assigned when orgs are created. Any application or service in a space within an org contributes to the usage of the quota. With Pay-As-You-Go or Subscription accounts, you can adjust your quota for Cloud Foundry applications and containers as the needs of your org change.

References:

- https://cloud.ibm.com/docs/account?topic=account-overview
- https://cloud.ibm.com/docs/iam?topic=iam-iamoverview

IBM Training

IBM Cloud: Cloud Foundry access



Introduction to IBM Cloud

Figure 2-20. IBM Cloud: Cloud Foundry access

Currently, not all services can be managed by using Cloud IAM. You can continue to use Cloud Foundry roles for access to these service instances. Users are added to the org and space to which the instance belongs with a Cloud Foundry role assigned.

This figure 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.

Cloud Foundry access is assigned by adding a user to an org and space, and then assigning an org role and space role. Depending on the type of role that is assigned, that user can complete specific actions for service instances that are added to a space.

References:

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

IBM Training

IBM Cloud: Identity and Access Management

- Identity:
  - The identity concept consists of:
    - User identities
    - Service and app identities
    - API keys
    - Resources
- Access Management:
  - Access management enables you to control which users see, create, use, and manage resources in your account. To grant access, you can assign roles that allow users levels of access for completing platform management tasks and accessing account resources.

Introduction to IBM Cloud

Figure 2-21. IBM Cloud: Identity and Access Management

The identity concept consists of the following aspects:

- User identities: Users are identified by their IBMid or IBM Cloud account ID.
- Service and app identities: Service IDs are a second type of identity that is used in an account. Service IDs are used to provide a separate identity for services and applications. You can create a service ID that is used by an application that needs access to your IBM Cloud services so that individual user credentials do not have to be used.
- API keys: API keys are provided through IBM Cloud IAM and cannot be used generally to authenticate with IBMid outside of IBM Cloud. You can also use a single classic infrastructure API key to access classic infrastructure APIs; however, this infrastructure is not required because you can use IBM Cloud API keys to access the same APIs.
- Resources: The final piece of the identity concept in IAM is IBM Cloud resources, which are identified by their cloud resource names (CRNs).

References:

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

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

IBM Training

IBM Cloud: Identity and Access Management (cont.)

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

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

Introduction to IBM Cloud

Figure 2-22. IBM Cloud: Identity and Access Management (cont.)

Some of the key IBM Cloud IAM features are:

- Unified user management across IBM Cloud PaaS and IaaS: A unified user management console is used to manage your users across IBM Cloud platform and infrastructure services.
- Enterprise federation: A federated ID can be used to sign up for IBM Cloud only if your company worked with IBM to register. Registering a company's domain with IBM enables users to log in to IBM products and services by using their company's 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.

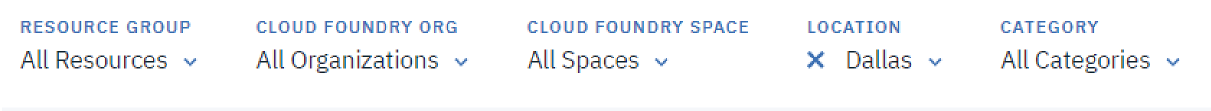
Reference:

https://cloud.ibm.com/docs/iam?topic=iam-iamoverview#features

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 IAM for access control, you can grant users access to the resources by using customizable resource groups.
- If the resources do not use IAM, you can use Cloud Foundry regions, organizations, and spaces for access control.



Introduction to IBM Cloud

Figure 2-23. 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: Cloudant, a Cloud Foundry application, a virtual machine (VM), or a container. Each account can have multiple resources.

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

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

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

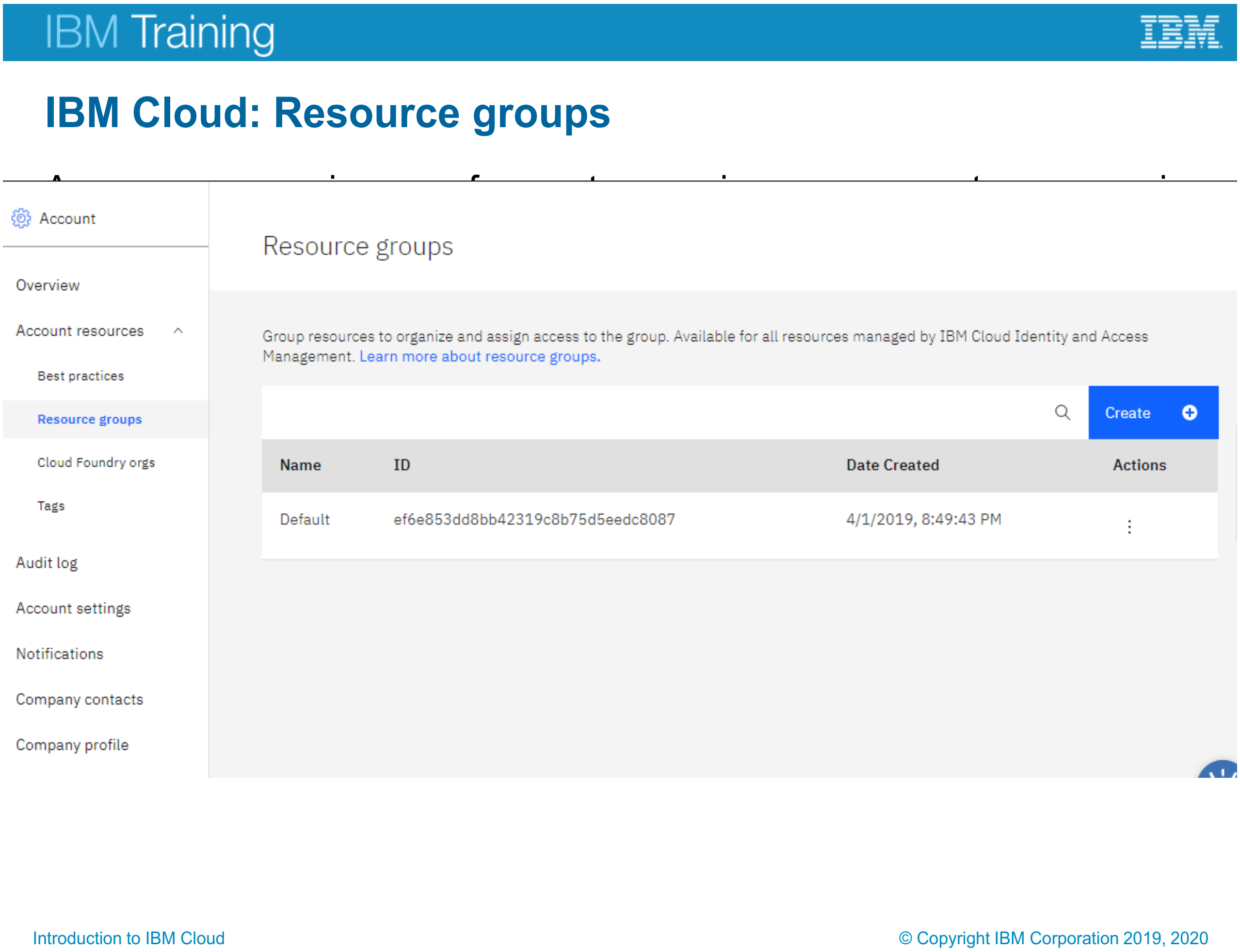


Figure 2-24. IBM Cloud: Resource groups

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

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

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

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

You must be assigned an IAM policy with the Administrator role on All Account Management services to create extra resource groups. If you have a Lite account or 30-day trial, you cannot create extra resource groups, but you can rename your default resource group.

To create a resource group, complete the following steps:

1. Click **Manage** > **Account** > **Account resources**.
2. Click **Resource Groups**.
3. Click **Create a resource group**.
4. Specify the name of the resource group.
5. Click **Add**.

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

Reference:

https://cloud.ibm.com/docs/resources?topic=resources-rgs

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.
- Cloud Foundry -based resources remain assigned to Cloud Foundry regions, organizations, and spaces

Introduction to IBM Cloud

Figure 2-25. 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 -like 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 to 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 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.

Another key objective of the resource controller is to make the IBM Cloud platform independent of Cloud Foundry. 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 retains the service broker extensibility model while breaking away from Cloud Foundry organizations and space constructs.

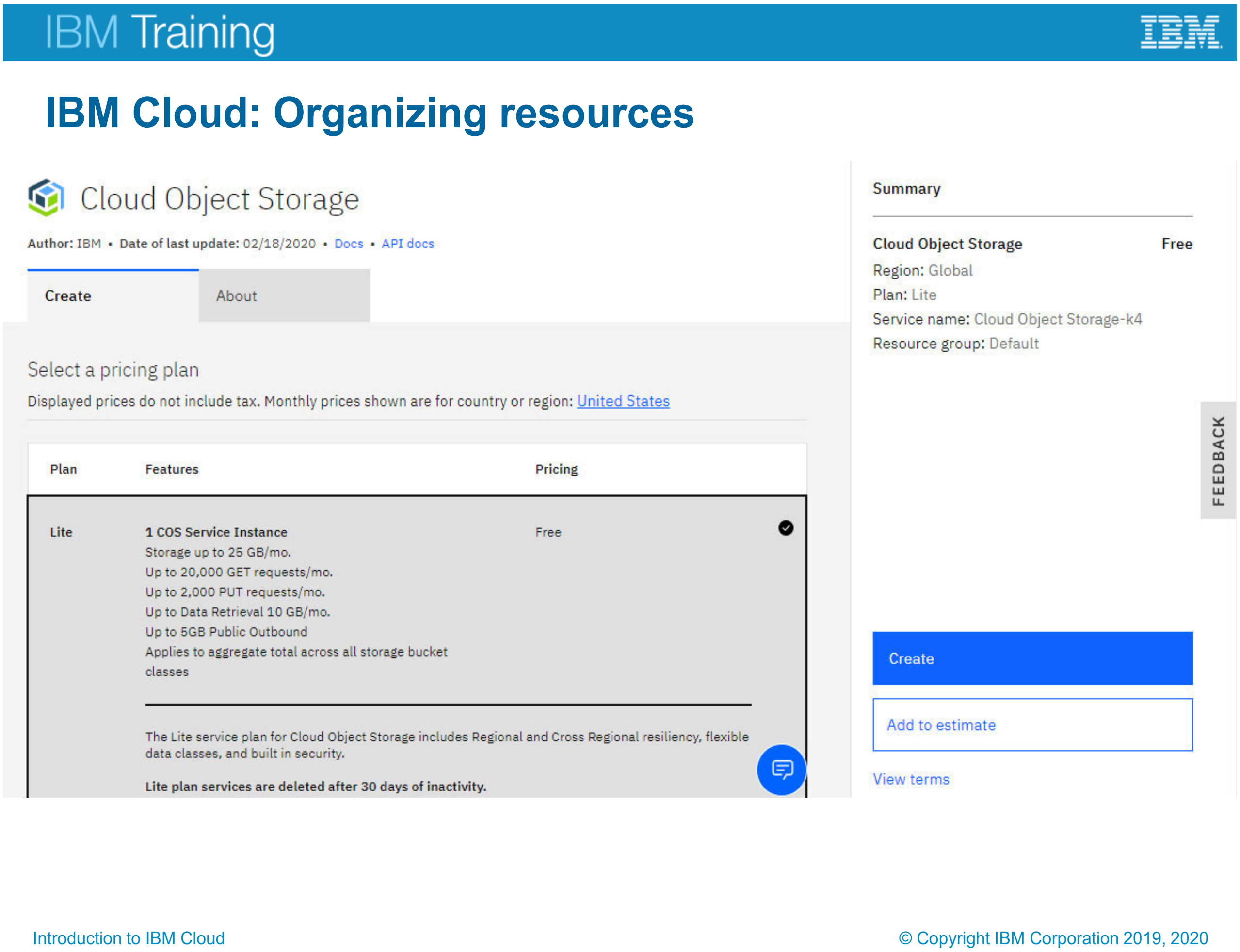


Figure 2-26. IBM Cloud: Organizing resources

When creating a resource, the resource group is added as shown in the right pane of the slide.

Note

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

Cloud Foundry-based services are expected to move to Cloud IAM and resource groups for Access Management.

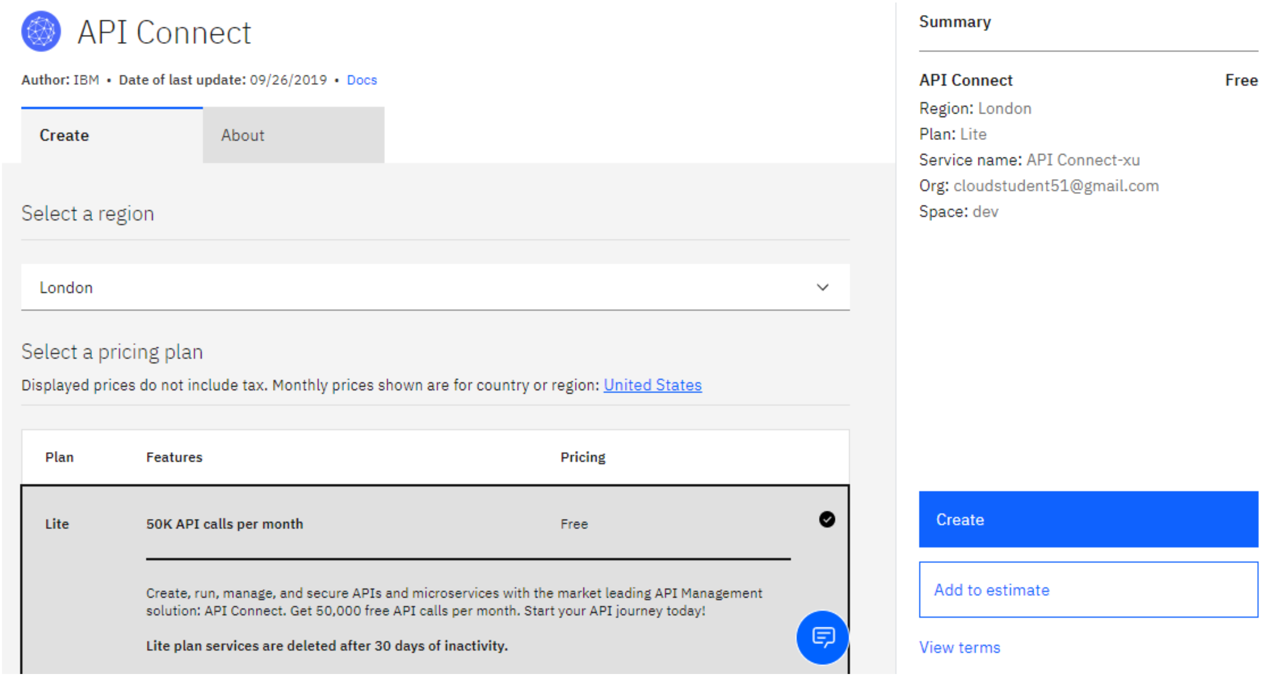
Any services that are introduced in the IBM Cloud catalog use Cloud IAM and resource groups for access control.

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

https://www.ibm.com/blogs/bluemix/2017/11/organize-navigate-manage-resources-ibm-cloud/

IBM Training

Cloud Foundry: Organizing resources



Introduction to IBM Cloud

Figure 2-27. Cloud Foundry: Organizing resources

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

When you create a Cloud Foundry resource (application or service) you can select the region. The organization and space to which the resource is assigned are added by default as shown in the slide. You can create a resource only in a space for which you have *developer* access.

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

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

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.

Introduction to IBM Cloud

Figure 2-28. 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 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: 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

Introduction to IBM Cloud

Figure 2-29. 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.
- Space auditors have read only access to settings, logs, applications, and services.

2.4. Managing applications and services on IBM Cloud

Figure 2-30. Managing applications and services on IBM Cloud

IBM Training

Topics

- IBM Cloud overview
- Databases on IBM Cloud
- Managing users and resources on IBM Cloud
- Managing applications and services on IBM Cloud
- Getting started with the Watson IoT Platform service on IBM Cloud

Introduction to IBM Cloud

Figure 2-31. Topics

IBM Training

IBM Cloud: Applications and services

- Applications: Programs that developers build in the Cloud Foundry environment:
  - Mobile applications
  - Web applications
- Services: Cloud extensions that are hosted by IBM Cloud:
  - Database, messaging, push notifications for mobile apps, and elastic caching for web apps

Cloud Foundry Applications

Name	Region	CF Org
MEAGalaxy-Agora	Dallas	karenfahmy@eg.ibm
Visual-Recognition-App1	Dallas	karenfahmy@eg.ibm
ex2-karen-nodejs	Dallas	karenfahmy@eg.ibm
honda-test-2	Dallas	karenfahmy@eg.ibm
ibmclinictest	Dallas	karenfahmy@eg.ibm

Services

Name	Location	Resource Group
Cloudant-wd	Dallas	default
Watson Assistant (formerly Conversation)-92	Sydney	default

Introduction to IBM Cloud

Figure 2-32. IBM Cloud: Applications and services

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

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

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

You can create your own services in IBM Cloud. They can be simple utilities, such as the functions 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.

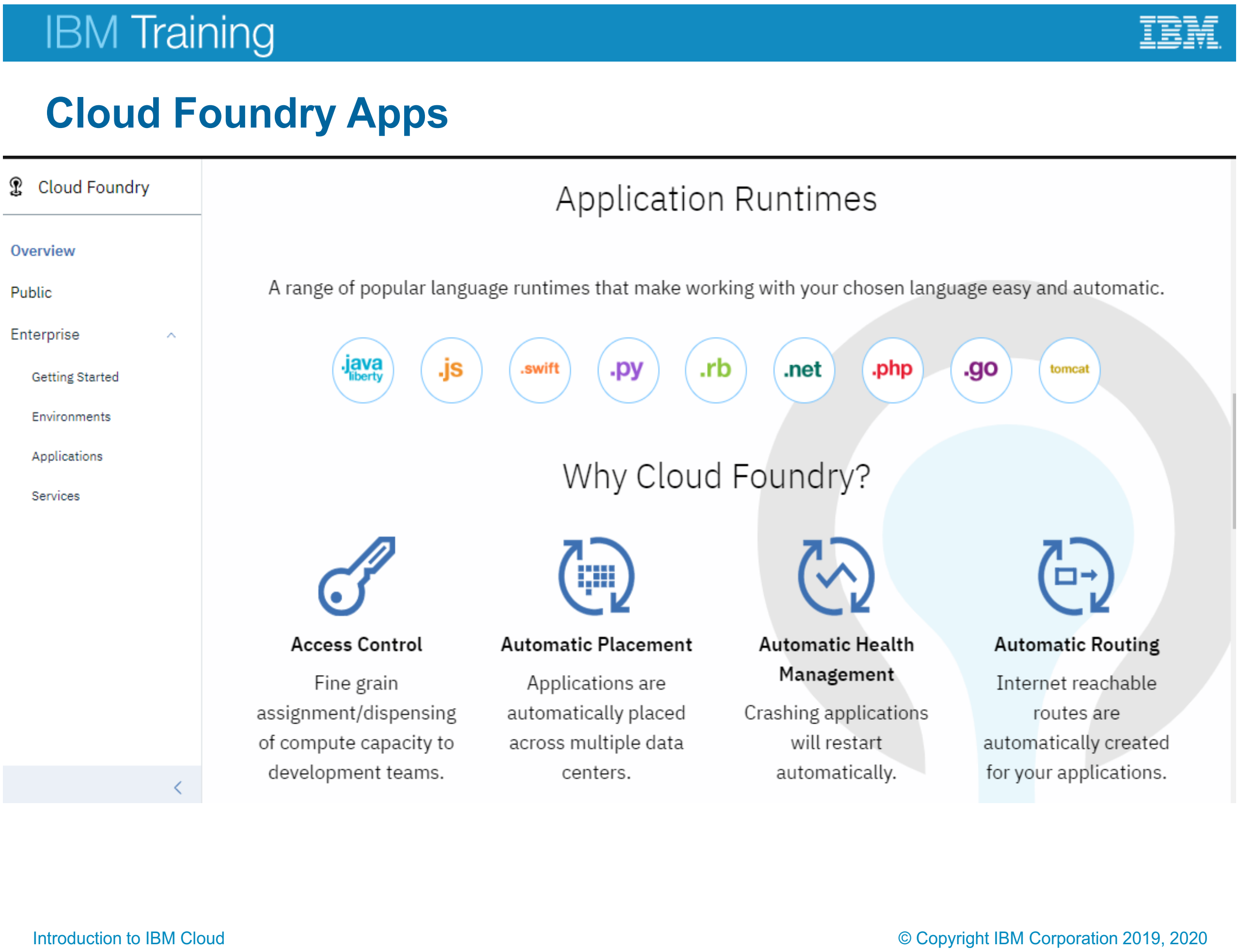


Figure 2-33. Cloud Foundry Apps

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

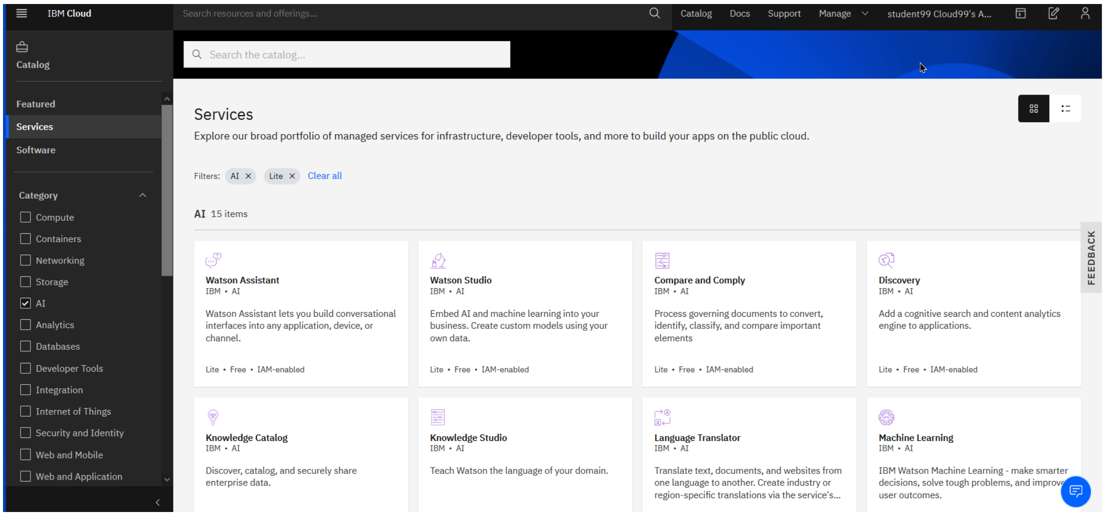
- IBM runtime environments, which are supported by IBM buildpacks, such as Liberty for Java.
- Community runtimes, which rely on open source and third-party buildpacks, such as Tomcat.

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

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

IBM Training

IBM Cloud catalog: Services



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

Introduction to IBM Cloud

Figure 2-34. IBM Cloud catalog: Services

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

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

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

Many IBM Cloud services are available, and more are being added. The slide shows a sample of the available Watson services.

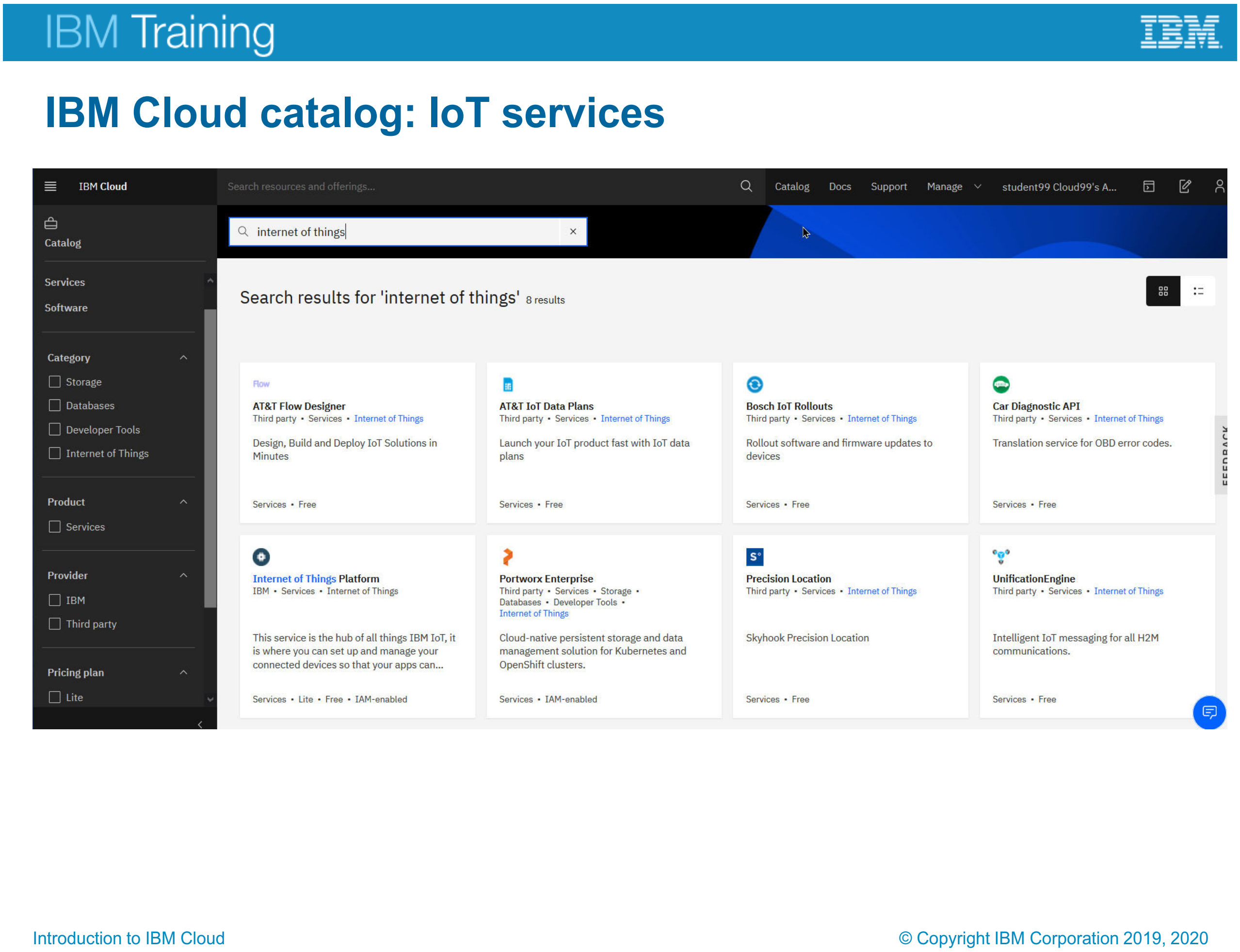


Figure 2-35. IBM Cloud catalog: IoT services

The following IBM Cloud services for IoT are available:

- Internet of Things Platform

  This service is the hub for Watson IoT Platform and lets you communicate with and use data from connected devices and gateways. Use the built-in web console dashboards to monitor your IoT data and analyze it in real time. Then, enhance and customize your Watson IoT Platform experience by building and connecting your own apps by using messaging and REST APIs.

- AT&T Flow Designer

  Unlock the power of the IoT by prototyping, building, and hosting IoT applications with AT&T Flow Designer. AT&T Flow Designer is a robust web-based development environment where data-driven applications can be designed and deployed with ease. Flow Designer makes it easy to prototype IoT solutions. Flow nodes are open source and available at GitHub. The Design Flow infrastructure is built to AT&T exacting security standards.

- UnificationEngine (UE)

  The world's first truly intelligent IoT messaging platform, the Unified Inbox Pte. Ltd. (UIB) patented device and platform-neutral UE technology uses AI and NLP to enable human-to-machine (H2M) communications on over 20 of the world's most popular global communications channels, including email, SMS, Facebook, WhatsApp, Viber, Skype, and many others, with new channels being added monthly. Dedicated to the IoT, the UE AI-powered unified messaging and intelligent chatbot technology platform provides users, developers, and manufacturers with increased security, interoperability, and convenience for smart home, smart enterprise, and smart city use cases and solutions. Learn more about how UE can enable you to communicate simply by going to the following websites:

  - https://www.unificationengine.com
  - https://www.unifiedinbox.com

2.5. Getting started with the Watson IoT Platform service on IBM Cloud

Figure 2-36. Getting started with the Watson IoT Platform service on IBM Cloud

IBM Training

Topics

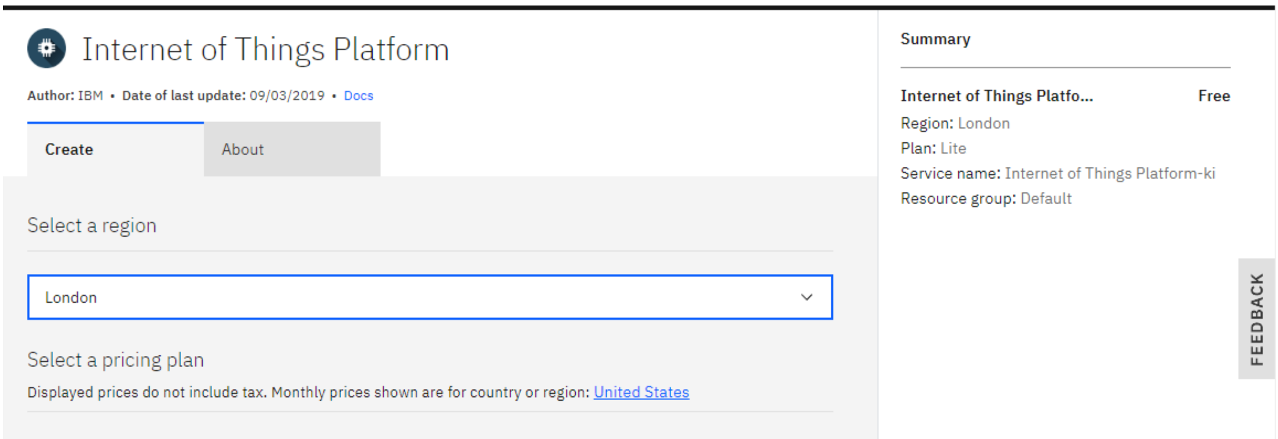
- IBM Cloud overview
- Databases on IBM Cloud
- Managing users and resources on IBM Cloud
- Managing applications and services on IBM Cloud
- Getting started with the Watson IoT Platform service on IBM Cloud

Figure 2-37. Topics

IBM Training

Watson IoT Platform service

- Watson IoT Platform for IBM Cloud gives you a versatile toolkit that includes gateway devices, device management, and powerful application access.
- By using the Watson IoT Platform service, you can collect connected device data and perform analytics on real-time data from your organization.



Introduction to IBM Cloud

Figure 2-38. Watson IoT Platform service

Watson IoT Platform is a fully managed, cloud-hosted service that makes it simple to derive value from IoT devices.

Register and connect your device, whether it is a sensor, a gateway, or something else, to Watson IoT Platform and start sending data securely up to the cloud by using the open, lightweight MQTT messaging protocol. You can set up and manage your devices by using your online dashboard or our secure APIs so that your apps can access and use your live and historical data.

Reference:

https://cloud.ibm.com/docs/services/IoT?topic=iot-platform-about_iotplatform

IBM Training

Data storage solutions on the Watson IoT Platform service

- Cloudant NoSQL DB
- IBM Event Streams for IBM Cloud
- IBM Db2 Warehouse on Cloud-Analytics Service data lake
- IBM Cloud Object Storage

Introduction to IBM Cloud

Figure 2-39. Data storage solutions on the Watson IoT Platform service

The Watson IoT Platform solution includes several separate data storage solutions for retaining IoT data for different purposes and periods.

The following databases are used:

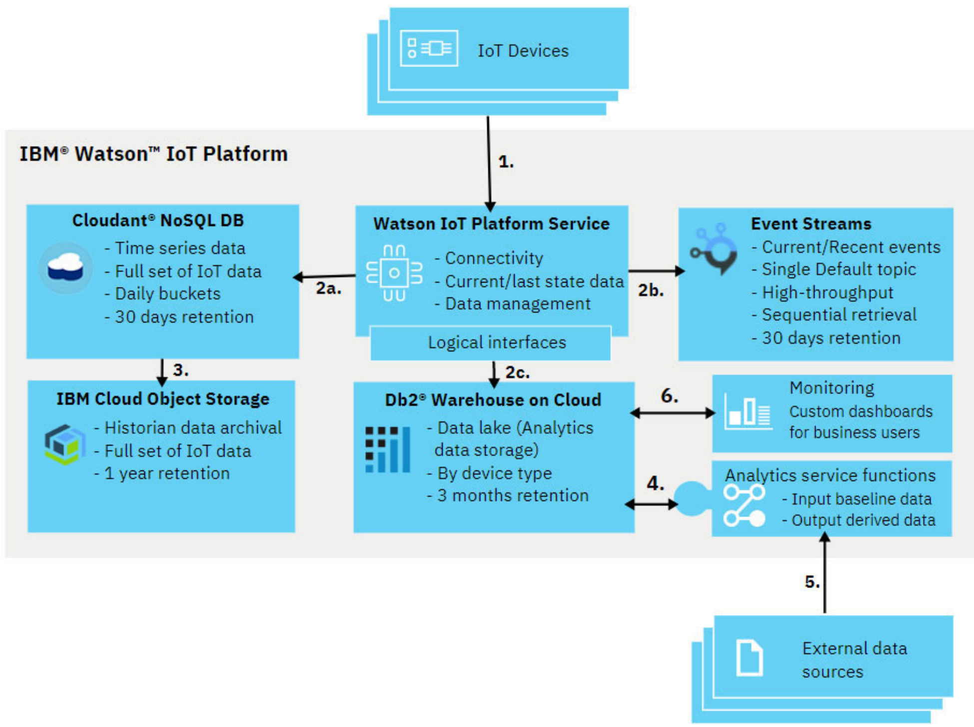
- Cloudant NoSQL DB: Contains the recent historical data for your IoT devices. You can view trends in Platform Service boards and cards.
- IBM Event Streams for IBM Cloud: Contains the current and recent historical data for your IoT devices.
- Db2 Warehouse on Cloud-Analytics Service data lake: Contains the long-term historical data for analytics access. Use analytics tools to extract knowledge from your data. If your plan includes the Analytics Service add-on, Db2 Warehouse on Cloud constitutes the data lake from which entity data is used and to which function output is written.
- IBM Cloud Object Storage: The complete set of historical data for archival purposes. Download data locally for offline processing.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/overview/iot_data_lifecycle.html

IBM Training

How data storage handles data in Watson IoT Platform



Introduction to IBM Cloud

Figure 2-40. How data storage handles data in Watson IoT Platform

Watson IoT Platform uses the following IoT device data flow:

1. Your IoT devices send data to Platform Service, which acts as a message broker and real-time handler of IoT data. For your IoT device data to be accessible to Watson IoT Platform you must register, connect, and create a device twin for your device in Platform Service.
2. Device data is written to the following databases:
    a. Cloudant NoSQL DB for short-term historian access.
    b. IBM Event Streams for IBM Cloud for real-time sequential retrieval of current and recent appliance data.
    c. Db2 Warehouse on Cloud for intermediate length storage and analytics.
3. The full data set is also written to IBM Cloud Object Storage for long-term storage.
4. Analytics Service add-on: The analytics component uses Db2 Warehouse on Cloud as the data lake from which entity data is used and to which function output is written.
5. Analytics Service add-on: Import entity data from external sources.
6. Use entity data with custom monitoring dashboards.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/overview/iot_data_lifecycle.html

IBM Training

Watson IoT Platform add-ons

The Watson IoT Platform solution can be expanded by using add-ons:

- Watson IoT Platform on Blockchain (add-on)
- Analytics Service (add-on)

Introduction to IBM Cloud

Figure 2-41. Watson IoT Platform add-ons

The Watson IoT Platform solution can be expanded by using add-ons:

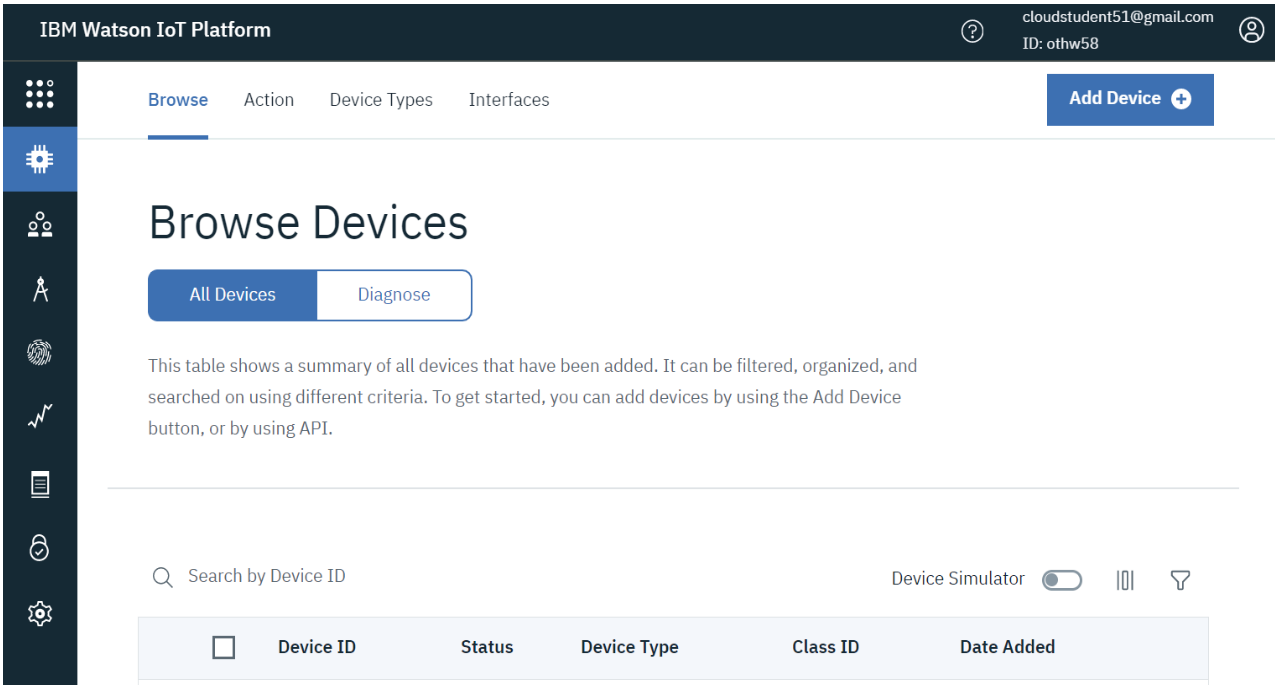
- Watson IoT Platform on Blockchain (add-on): Watson IoT Platform on Blockchain is an add-on component of Watson IoT Platform that enables your IoT resources to participate in blockchain business networks.
- Analytics Service (add-on): Analytics Service is an extension and an add-on component to Watson IoT Platform. Analytics Service enables line-of-business users to easily enrich and interact with the raw metrics data coming from IoT entities by using built-in configurable analytics functions.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/overview/overview.html

IBM Training

Watson IoT Platform landing page



Introduction to IBM Cloud

Figure 2-42. Watson IoT Platform landing page

This slide shows an overview of the Watson IoT Platform landing page. On this page, Watson IoT Platform provides a clean and simple UI (dashboard) where you can simply and easily add and manage your devices.

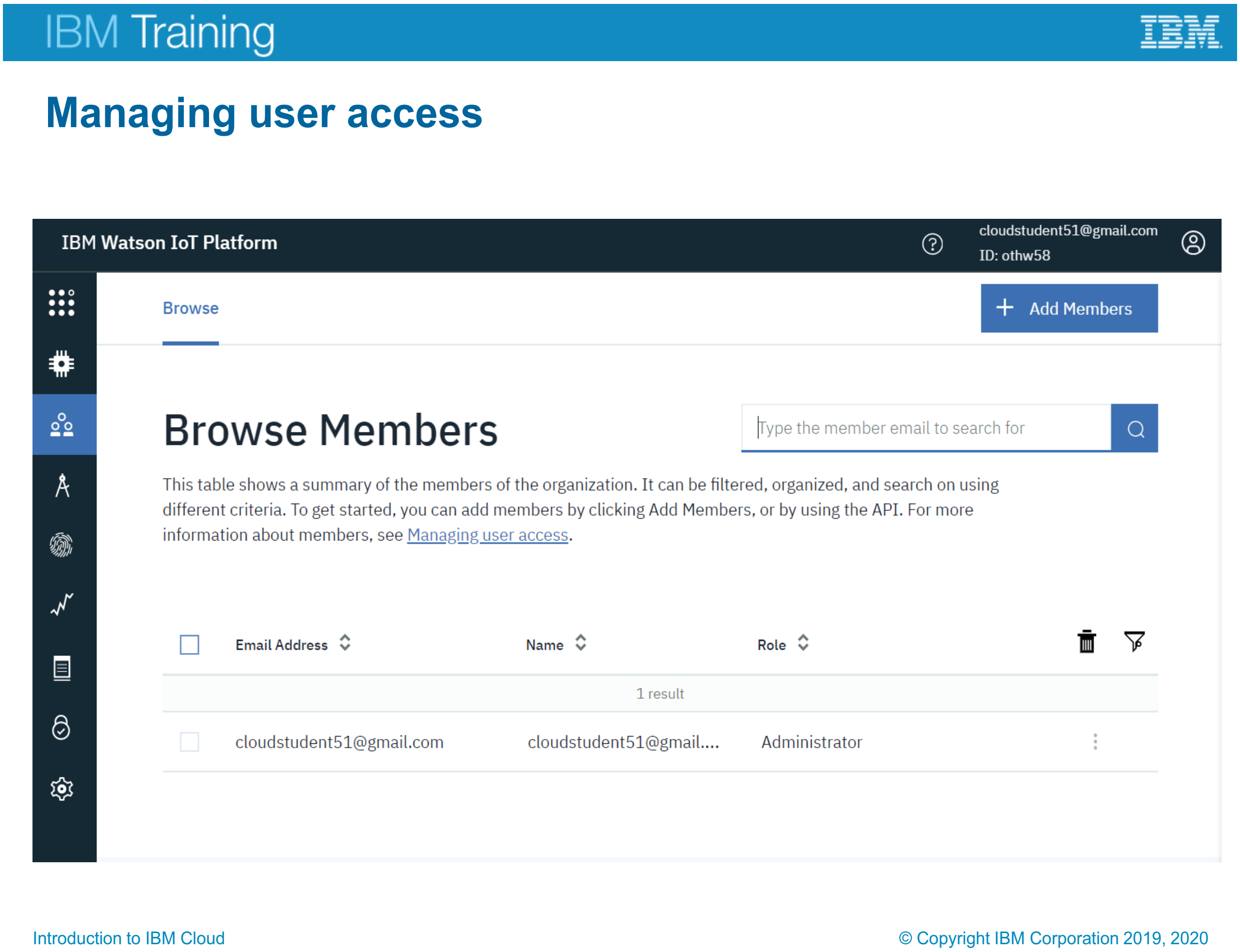


Figure 2-43. Managing user access

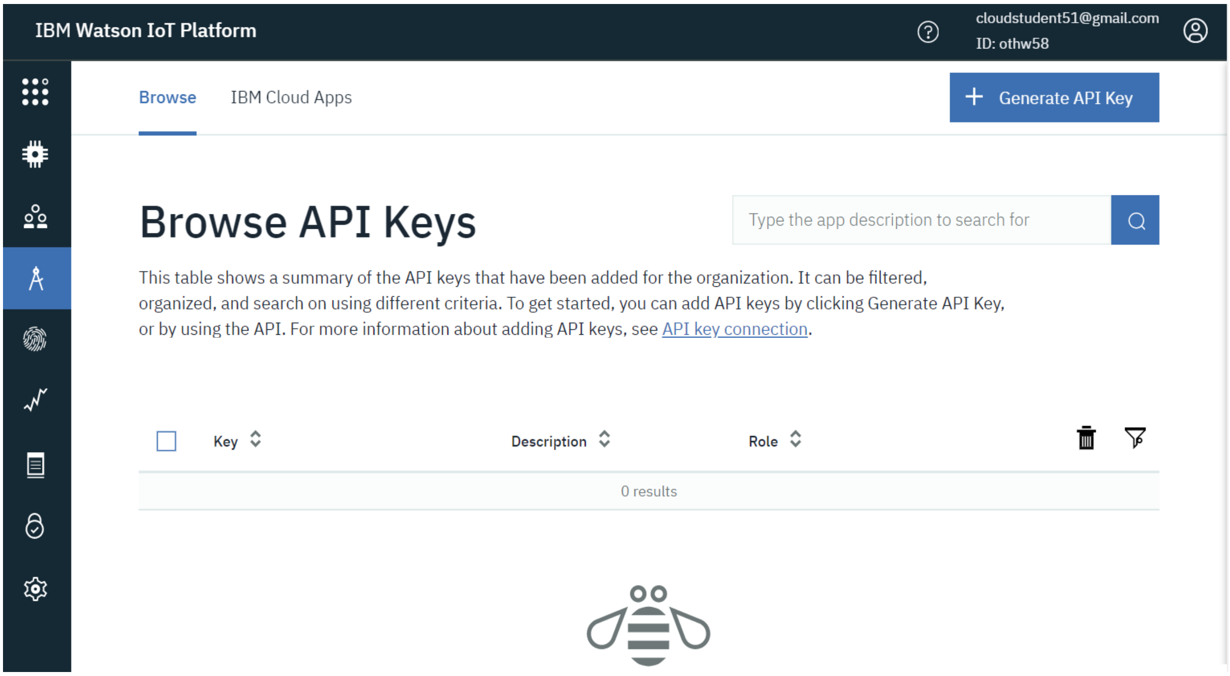
From the Members dashboard, you can control and manage access to your Platform Service organization. You can add users by adding, inviting, or importing them. You can also give different levels of access to your users by assigning roles.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/add_users.html#managing-user-access

IBM Training

Adding new application APIs



Introduction to IBM Cloud

Figure 2-44. Adding new application APIs

The application is the most powerful class of things that is supported by Platform Service. Use applications to do the following tasks:

- Send events on behalf of devices.
- Send commands to devices.
- Receive commands that are sent to devices.
- Interact with Platform Service through the REST APIs.

Applications authenticate to Watson IoT Platform by using an API key and authentication token. You use the access section of the dashboard to create API keys.

To create an API key and authentication token pair, complete the following steps:

1. In the Platform Service dashboard, click **Apps > API Keys**.
2. Click **Generate API Key**. Make a note of the API key and token pair. Authentication tokens are unrecoverable. If you lose or forget this token, you must reregister the API key to generate a new authentication token.
    - An example of an API key is a-organization_id-a84ps90Ajs.
    - An example of a token is MP$08VKz!8rXwnR-Q*.
3. Add a comment to identify the API key in the dashboard, for example: "Key to connect my application".
4. Click **Finish**.

IBM Training

Unit summary

- Describe the key capabilities of IBM Cloud.
- List services available in the IBM Cloud Catalog.
- List databases available on IBM Cloud.
- Manage IBM Cloud users and resources.
- Manage applications and services on IBM Cloud.
- Explain how to get started with the IBM Watson IoT Platform service on IBM Cloud.

Introduction to IBM Cloud

Figure 2-45. Unit summary

IBM Training

Review questions

1. True or False: An IaaS provides direct access to the VMs and operating systems that you use to run your application.
2. Which one of the following areas is not a key area of Watson IoT Platform:
   A. Connect.
   B. Information Management.
   C. Analytics.
   D. Data Management.
3. Which of the following statements about IBM Cloud is true (choose one):
   A. In IBM Cloud, anyone can manage your application without permission.
   B. Services are the primary means that enable users to share control over apps.
   C. A user must belong to only one space in IBM Cloud.
   D. Each Cloud Foundry -based application or service is associated with exactly one space.

Introduction to IBM Cloud

Figure 2-46. Review questions

IBM Training

Review questions (cont.)

4. Which of the following regions is NOT one of deployment regions of Watson IoT Platform:
   A. Tokyo
   B. London
   C. Frankfurt
   D. Washington, D.C.
5. True or False: NoSQL databases are a good example of an IBM Cloud starter application, which is sometimes referred to as an app template.

Introduction to IBM Cloud

Figure 2-47. Review questions (cont.)

IBM Training

Review answers

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

2. Which one of the following areas is not a key area of Watson IoT Platform:
   A. Connect.
   B. Information Management.
   C. Analytics.
   D. Data Management.
   The answer is D.

Introduction to IBM Cloud

Figure 2-48. Review answers

IBM Training

Review answers (cont.)

3. Which of the following statements about IBM Cloud is true (choose one):
   A. In IBM Cloud, anyone can manage your application without permission.
   B. Services are the primary means that enable users to share control over apps.
   C. A user must belong to only one space in IBM Cloud.
   D. Each Cloud Foundry -based application or service is associated with exactly one space.

   The answer is D.

4. Which of the following regions is NOT one of deployment regions of Watson IoT Platform:
   A. Tokyo
   B. London
   C. Frankfurt
   D. Washington, D.C.

   The answer is A.

Introduction to IBM Cloud

Figure 2-49. Review answers (cont.)

IBM Training

Review answers (cont.)

5. True or False: NoSQL databases are a good example of an IBM Cloud starter application, which is sometimes referred to as an app template.
   The answer is False.
   NoSQL databases are IBM Cloud *services*.

Introduction to IBM Cloud

Figure 2-50. Review answers (cont.)

Figure 2-51. Exercise: Creating your first IoT application

This exercise shows how you can set up an environment on IBM Cloud to deploy your first IoT application by using some of the services that are provided by IBM Cloud.

IBM Training

Exercise objectives

- This exercise shows how you can set up an environment on IBM Cloud to deploy your first IoT application by using some of the services that are provided by IBM Cloud.
- After completing this exercise, you should be able to:
  - Set up an IBM Cloud Lite account.
  - Log in to IBM Cloud from a browser.
  - Describe the IoT related services in the IBM Cloud catalog.
  - Deploy the IoT application that is based on Node-RED.
  - Set up Node-RED configurations to start developing on Node-RED Editor.

Introduction to IBM Cloud

Figure 2-52. Exercise objectives

Unit 3. Introduction to IBM Watson IoT Platform

Estimated time

02:00

Overview

This unit takes a closer look at the building blocks of an IoT architecture and at the capabilities that are provided by Watson IoT Platform.

IBM Training

Unit objectives

- List the features of Watson IoT Platform.
- Use the Watson IoT Platform dashboard.
- Manage devices and set up security policies on Watson IoT Platform.
- Manage user access on Watson IoT Platform.
- Manage applications on Watson IoT Platform.
- Describe the advanced features of Watson IoT Platform.
- Use recipes to help jump-start application development.

Introduction to IBM Watson IoT Platform

Figure 3-1. Unit objectives

3.1. Watson IoT Platform features overview

IBM Training

Watson IoT Platform features overview

Introduction to IBM Watson IoT Platform

Figure 3-2. Watson IoT Platform features overview

IBM Training

Topics

- Watson IoT Platform features overview
- Watson IoT Platform dashboard
- Device management and security policies
- Application management
- Access management
- Advanced features in Watson IoT Platform
- Watson IoT Platform recipes

Introduction to IBM Watson IoT Platform

Figure 3-3. Topics

IBM Training

What is Watson IoT Platform

- Watson IoT Platform is a fully managed and cloud-hosted service that makes it simple to derive value from Internet of Things (IoT) devices.
- Watson IoT Platform has a versatile toolkit that includes gateway devices, device management, and powerful application access.
- By using Watson IoT Platform, connected device data can be collected, and analytics on real-time data can be performed over the cloud.

Introduction to IBM Watson IoT Platform

Figure 3-4. What is Watson IoT Platform

Watson IoT Platform is a fully managed and cloud-hosted service that makes it simple to derive value from Internet of Things (IoT) devices.

Simply register and connect your device, whether it is a sensor, gateway, or something else to Watson IoT Platform and start sending data securely to the cloud by using the open and lightweight Message Queuing Telemetry Transport (MQTT) messaging protocol. You can set up and manage your devices by using your online dashboard or the Watson IoT Platform secure APIs so that your apps can access and use your live and historical data.

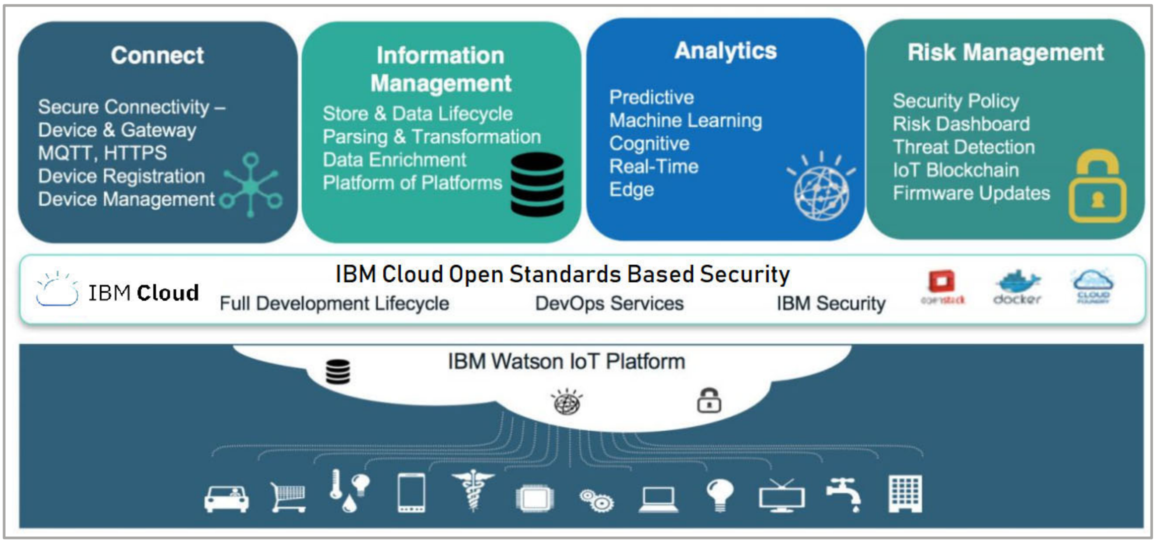
IBM Watson IoT Platform for IBM Cloud gives you a versatile toolkit that includes gateway devices, device management, and powerful application access. By using Watson IoT Platform, you can collect connected device data and perform analytics on real-time data from your organization.

Reference:

https://cloud.ibm.com/docs/services/IoT?topic=iot-platform-about_iotplatform

IBM Training

Watson IoT Platform key features



Introduction to IBM Watson IoT Platform

Figure 3-5. Watson IoT Platform key features

The following key features are included with the Watson IoT Platform solution:

Connect

Quickly and securely register and connect your devices and gateways. You can find simple step-by-step instructions for connecting popular devices, sensors, and gateways in the recipes site.

Information Management

Control what happens to the data that is received from your connected devices. Manage data storage, configure data transformation actions, and integrate with other data services and device platforms.

Analyze in real time

Monitor your real-time device data through rules, analytics, and dashboards. Define rules to monitor conditions and trigger automatic actions that include alerts, email, If This Then That (IFTTT), Node-RED flows, and external services to react quickly to critical changes.

Risk and Security management

The secure-by-design control capabilities protect the integrity of your IoT solution through secure connectivity and access control for users and applications. Extend the base security with threat intelligence for IoT to visualize critical risks and automate operational responses with policy-driven mitigation actions.

Reference:

https://cloud.ibm.com/catalog/services/internet-of-things-platform#about

3.2. Watson IoT Platform dashboard

IBM Training

Watson IoT Platform dashboard

Introduction to IBM Watson IoT Platform

Figure 3-6. Watson IoT Platform dashboard

IBM Training

Topics

- Watson IoT Platform features overview
- Watson IoT Platform dashboard
- Device management and security policies
- Application management
- Access management
- Advanced features in Watson IoT Platform
- Watson IoT Platform recipes

Introduction to IBM Watson IoT Platform

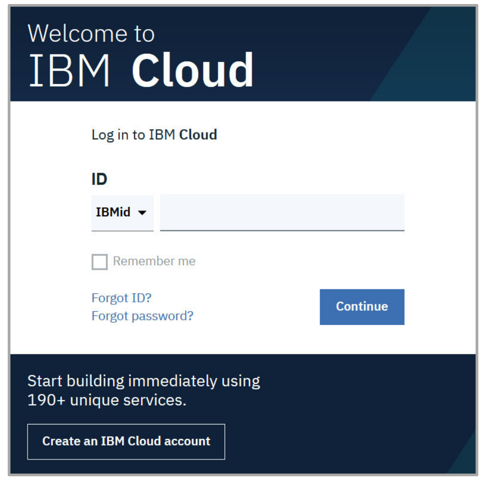
Figure 3-7. Topics

IBM Training

Getting started: Accessing Watson IoT Platform

You must register for an IBM Cloud account to access Watson IoT Platform:

1. Open the IBM Cloud page at http://cloud.ibm.com.
2. Click **Create an IBM Cloud account**:
   - If you do not have an IBM ID, you must create one.
   - Creating an IBM Cloud account automatically creates an IBM ID for you.
   - There is no charge to create an IBM ID or an IBM Cloud Lite account.
   - After your IBM Cloud account is created, you may create an instance of the Watson IoT Platform service.



Introduction to IBM Watson IoT Platform

Figure 3-8. Getting started: Accessing Watson IoT Platform

Before you work on any of the exercises in this course, you must sign up for an IBM Cloud account. You can register for a free IBM Cloud Lite account.

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

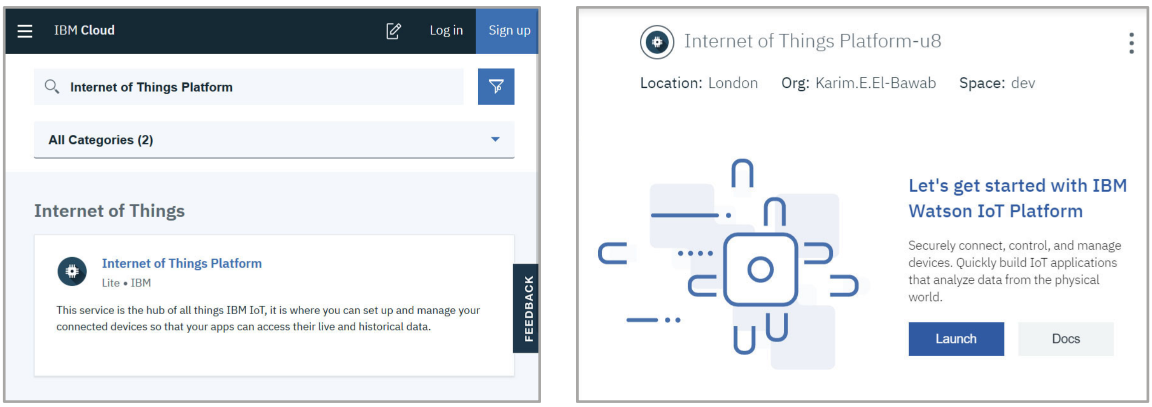
https://www.youtube.com/watch?v=0rMYXcbpHbI

IBM Training

Getting started: Accessing Watson IoT Platform (cont.)

To create an instance of the Watson IoT Platform service, complete the following steps:

1. Open the IBM Cloud catalog by going to https://cloud.ibm.com/catalog.
2. Search for the Internet of Things Platform service and open it.
3. Create an instance of the service by clicking **Create**.
4. After you create the service instance, click **Launch**.



Introduction to IBM Watson IoT Platform

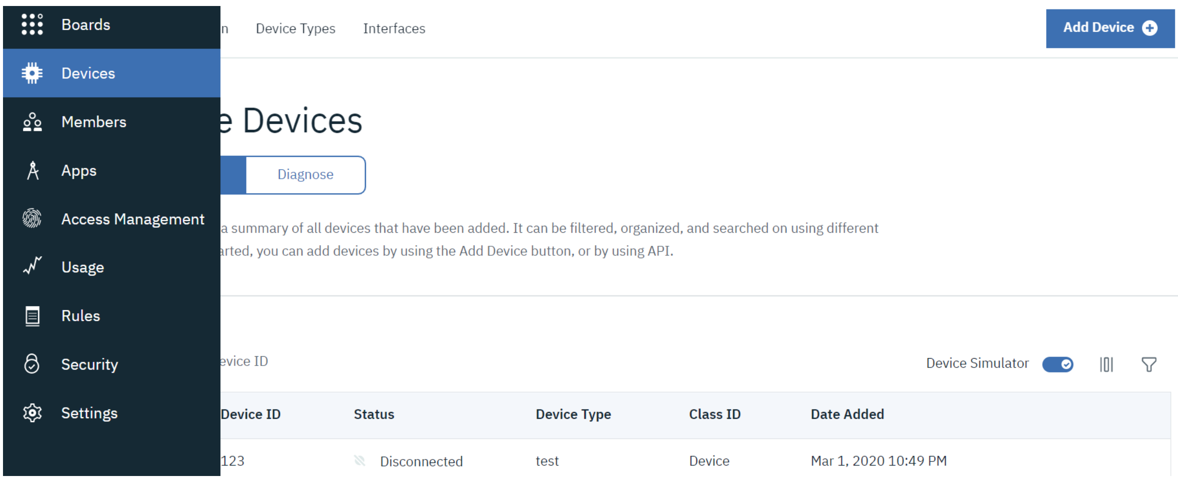
Figure 3-9. Getting started: Accessing Watson IoT Platform (cont.)

The Watson IoT Platform instance is considered your *organization*, where you can connect, manage, and control your devices. After you create the Watson IoT Platform service instance, IBM Cloud generates an organization ID to identify the organization. The organization ID might be required as one of the configuration parameters to connect the instance with devices, gateways, and applications.

IBM Training

Watson IoT Platform dashboard

- Watson IoT Platform has a simple dashboard structure that you can use to explore its features.
- The left navigation pane includes the main key features of the platform, which are organized under different sections.
- The top bar contains the account settings, documentation, terms, and privacy policies.



Introduction to IBM Watson IoT Platform

Figure 3-10. Watson IoT Platform dashboard

To add a device from the Watson IoT Platform dashboard, complete the following steps:

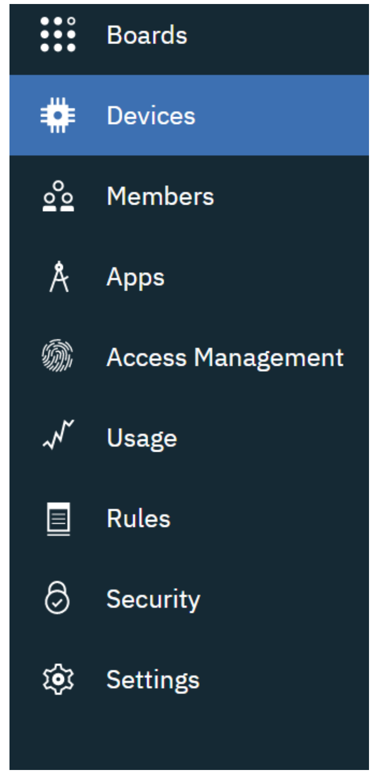
1. On the IBM Cloud console, select **IoT** from the menu and click **Watson IoT Platform**.
2. Log in or click **Sign up to Create**.
3. On the Watson IoT Platform page, choose a region, organization, and space, and click **Create**.
4. On the service page, click **Launch** to start administering your Watson IoT Platform organization.
5. In the Overview dashboard, from the menu pane select **Devices** and click **Add Device**.

6. Select or create a device type for the device that you are adding. Each device that is connected to the Watson IoT Platform must be associated with a device type. Device types are groups of devices that share common characteristics. When you add your first device to your Watson IoT Platform organization, no device types are available in the **Device type** menu. You must create a device type:
   a. Click **Create device type**.
   b. Enter a device type name, such as *my_device_type* and a description for the device type. Important: The device type name must be no more than 36 characters and can contain only the following characters:
   c. Alpha-numeric characters (a-z, A-Z, and 0-9)
   d. Hyphens (-)
   e. Underscores (_)
   f. Periods (.)
   g. Click **Create** to add the new device type.
7. Click **Next** to begin the process of adding your device with the selected device type.
8. Enter a device ID, such as my_first_device. The device ID is used to identify the device in the Watson IoT Platform dashboard. It is a required parameter for connecting your device to Watson IoT Platform.

IBM Training

Watson IoT Platform dashboard: Left navigation pane

- Boards: Visualizing real-time data.
- Devices: Connecting, managing, and controlling devices and gateways.
- Members: Controlling and managing users' access to the organization.
- Apps: Building and managing applications that interact with and use information from connected devices in Watson IoT Platform.
- Access Management: Granting or restricting access to specific operations by using roles to manage permissions for groups of users, applications, and gateways.
- Usage: Showing how many devices and applications use Watson IoT Platform.
- Rules: Creating embedded rules that trigger when an event occurs. Setting up an action in response to the trigger.



Introduction to IBM Watson IoT Platform

Figure 3-11. Watson IoT Platform dashboard: Left navigation pane

The left navigation pane of Watson IoT Platform dashboard contains the main key features of the platform, which are organized under different sections:

- Boards:
  - Create boards and cards to create and share dashboards that visualize device data in real time. Data set values from one or more devices can be graphically visualized by using boards and cards to provide a quick overview and enhance understanding of the data.
  - Create boards and add cards that display the data as raw numbers, real-time graphs, gauges, and more.
  - Add members to boards to share them with other users in the organization.
  - Arrange the cards and add explanatory text dividers to fine-tune the presentation.

    The Watson IoT Platform dashboard has some default boards, such as Usage Overview, Rule-Centric Analytics, Device-Centric Analytics, and Risk and Security Overview. The cards that are added in the board have different types:

    - Generic visualization.
    - Line chart.
    - Bar chart.

- Donut chart.
- Value.
- Gauge.
- Device properties.
- All device prosperities.
- Device list.
- Device information.
- Device map.
- Rules.
- Rules alerts.
- Alert information.
- Policy compliance.
- Blacklist and whitelist compliance.
- Connection security.
- Device types.
- Data transferred.
- Separator.

You can expand the default set of cards by creating custom cards.

- Devices

  Connecting a device to Watson IoT Platform involves registering the device with Watson IoT Platform and then using the registration information to configure the device to connect to Watson IoT Platform. The connection of the device with Watson IoT Platform must be done with the MQTT protocol or HTTP REST API. If the device cannot be connected directly, Watson IoT Platform provides the option to connect the device through a gateway, which serves as an access point to Watson IoT Platform for other devices. Connecting a gateway to Watson IoT Platform involves creating a gateway device type and registering the gateway with Watson IoT Platform and then using the registration information to connect the gateway to Watson IoT Platform.

- Members

  Controlling and managing access to Watson IoT Platform organization is done from the Members dashboard. Adding users may be done by adding, inviting, or importing them. Controlling the different level of access for the users may be achieved by assigning roles. The following options are also available:

  - Editing users.
  - Blocking user access.
  - Limiting user access.
  - Removing users.

- Apps

  By using Watson IoT Platform, you can build and manage applications that interact with and use information from connected devices in Watson IoT Platform by using the following supported protocols and standards:

  - C#.
  - Java.
  - MQTT messaging.
  - Node.js.
  - Node-RED.
  - Python.

  Connecting these apps with Watson IoT Platform is achieved by generating API keys and tokens or binding the application directly to Watson IoT Platform in IBM Cloud.

- Access Management

  Roles are sets of permissions that can be used to grant or restrict access to specific operations. Roles can be used to manage permissions for groups of users, applications, and gateways. Here are the types of roles:

  - User roles may be assigned while adding, inviting, or registering new users to the organization.
  - Application roles are assigned to grant or deny applications access to specific operations. All application roles deny access to the following operations:
    - All risk management operations.
    - Configure storage parameters.
    - Configure authentication providers.
    - Create, update, or delete email configuration.
  - Gateways have a limited number of roles that govern the definition of the gateway and the ability to register devices to the Watson IoT Platform.

- Usage

  Usage metrics show how much devices and applications use Watson IoT Platform. They show a summary of the total usage for the current month and the previous month, and details of data that is transferred during different periods. The metrics are refreshed every 2 hours.

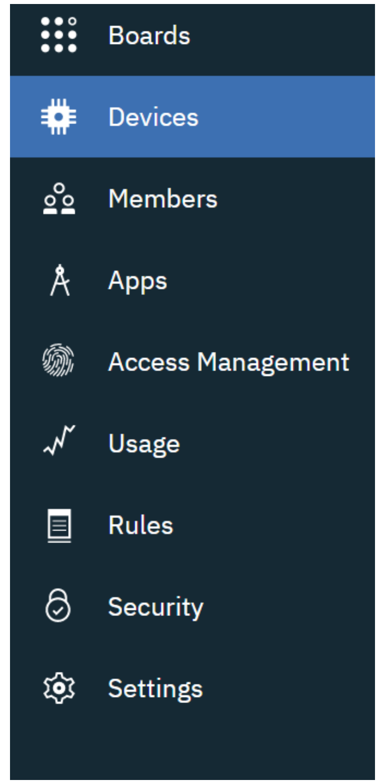
- Rules

  With Watson IoT Platform, you may set up rules that trigger when an event that is received by Watson IoT Platform causes a change to the device state. Embedded rules are condition-based decision points that match real-time device data with predefined threshold values or other property data to trigger the rule if a condition is met. Embedded rules may specify the conditions that trigger a rule and then set up an action in response to the trigger, for example, sending an alert to a user's device and an email to an administrator when the temperature of device spikes.

IBM Training

Watson IoT Platform dashboard: Left navigation pane (cont.)

- Security: Configuring policies to enhance connection security and control access to the server from devices.
- Settings: General settings that allow modifying global organization information and locally enable experimental Watson IoT Platform features.



Introduction to IBM Watson IoT Platform

Figure 3-12. Watson IoT Platform dashboard: Left navigation pane (cont.)

The left navigation pane has the following extra features:

- Security

  Policy configuration is available to enhance connection security and control access to the server from devices. Use a *Connection Security* policy to set the default security level that is applied to all devices. There is also the option to add custom rules for specific devices. Organizations that use advanced security can restrict access to the server from certain devices by using a *blacklist* or can use a *whitelist* to grant server access to specific devices. You may not use a blacklist and a whitelist concurrently; enabling one means disabling the other one.

- Settings

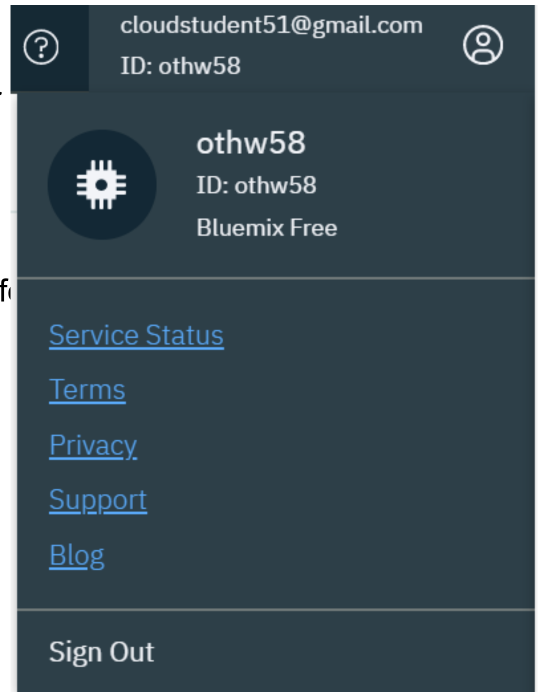
  General settings show and allow modifying global organization information and locally enabling experimental Watson IoT Platform features. The settings have three main sections: platform, data and devices, and security.

  - *Platform* shows the organization data as an organization ID, the date that it was created, the organization type, and geographical location. With it, you can enable the last cache event and the experimental features, such as Watson IoT Platform Edge and the Weather Company visualization.
  - *Data and devices* include custom device management packages that can be uploaded to add device management functions and a device simulator that simulates devices and device data to get running quickly on Watson IoT Platform.
  - *Security* is responsible for managing connection security, CA certificates, and messaging server security.

IBM Training

Watson IoT Platform dashboard: Top bar

- The top bar shows the account settings and the documentation, and it provides quick access to different actions across Watson IoT Platform.
- The *organization ID* is the identifier for the user account across Watson IoT Platform.
- *Service Status* shows the health dashboard where you can view the status of the key components of the service instance.
- *Terms show* the updated terms and conditions f Watson IoT Platform.
- *Privacy* shows the IBM Privacy Statement.
- *Support* includes quick-start tutorials, documentation, and FAQs for both IBM Cloud and Watson IoT Platform.



Introduction to IBM Watson IoT Platform

Figure 3-13. Watson IoT Platform dashboard: Top bar

The top bar shows account settings such as the organization ID and type. It also includes *Service Status*, *Terms*, *Privacy*, and *Support*. The *sign-out* option is also available in this menu. The question mark sign (?) in the top bar opens a menu that gives quick access to different actions across Watson IoT Platform, such as add device, generate API key, add member, and add role. It also includes direct links to the following items:

- Documentation: Shows Watson IoT Platform official documentation.
- Quickstart: A quick and easy tutorial that demonstrates how to connect devices to Watson IoT Platform.
- Blog: A community for Watson IoT Platform that shows announcements about different updates that are related to Watson IoT Platform.

References:

https://status.internetofthings.ibmcloud.com/

https://www-03.ibm.com/software/sla/sladb.nsf/sla/bm-6738-07

https://www.ibm.com/privacy/us/en/

https://cloud.ibm.com/docs

3.3. Device management and security policies

IBM Training

Device management and security policies

Introduction to IBM Watson IoT Platform

Figure 3-14. Device management and security policies

IBM Training

Topics

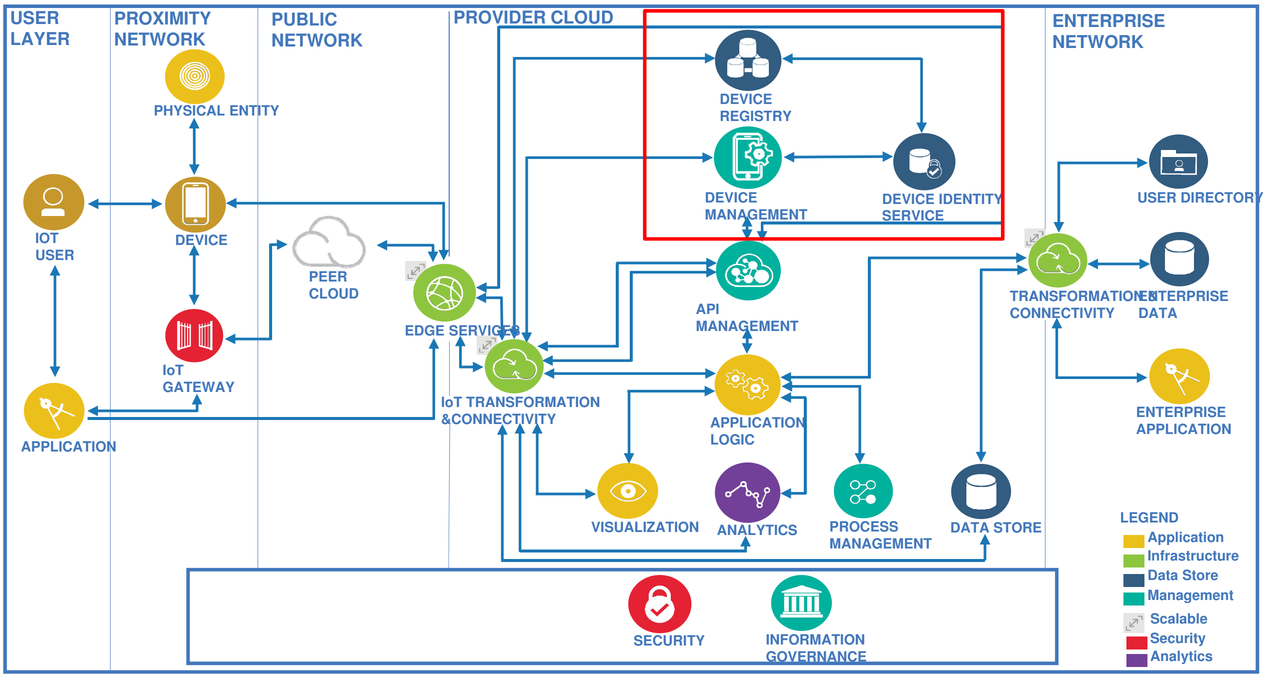
- Watson IoT Platform features overview
- Watson IoT Platform dashboard
- Device management and security policies
- Application management
- Access management
- Advanced features in Watson IoT Platform
- Watson IoT Platform recipes

Introduction to IBM Watson IoT Platform

Figure 3-15. Topics

IBM Training

Device management: Reference architecture



Introduction to IBM Watson IoT Platform

Figure 3-16. Device management: Reference architecture

This figure represents the *IoT reference architecture* and shows how to connect to IoT devices and build quickly scalable apps and visualization dashboards to gain insights from IoT data by using IBM Cloud IoT, data, and AI services.

The highlighted section shows the components that are related to the *device management*:

- *Device registry*: Stores information about devices that the IoT system may read, communicate with, control, provision, or manage. You can enable this component by using Watson IoT Platform.
- *Device management*: Provides an efficient way to manage and connect devices securely and reliably to the cloud platform. You can enable this component by using Watson IoT Platform.
- *Device identity service*: Identifies the device services after the user registers a device. You can enable this component by using Watson IoT Platform or IBM Maximo (which is considered a leading enterprise asset management solution).

Reference:

https://www.ibm.com/cloud/garage/architectures/iotArchitecture/reference-architecture

IBM Training

Device management

- One of Watson IoT Platform main features is its ability to connect, control, and manage devices and gateways.
- Connecting a device or a gateway to Watson IoT requires the following items:
  - HTTP or MQTT protocols support.
  - Messages compatible with Watson IoT Platform.
- The steps to connect a device or a gateway are as follows:
  1. Register the device.
  2. Set up the device for MQTT messaging by using the registration information.
  3. Send device messages to the platform.
- The device management capabilities are provided through a device management API and a device management agent that is installed on devices.
- Device management actions include restart, download and install firmware updates, and reset devices to factory settings.
- The device management capabilities can be extended by using device management extension packages.

Introduction to IBM Watson IoT Platform

Figure 3-17. Device management

Watson IoT Platform securely connects, controls, and manages devices and gateways. Watson IoT Platform recognizes devices and gateways as two classes of device. Watson IoT Platform provides actions that can be initiated for one or more devices. These actions can be initiated by using the dashboard or the REST API. The types of available actions are *device actions* and *firmware actions*.

Managed devices are defined as devices that contain a device management agent. A device management agent is a set of logic that allows the device to interact with the Watson IoT Platform Device Management service by using the Device Management Protocol. Managed devices can perform device management operations, including location updates, firmware downloads and updates, restarts, and factory resets. The Device Management Protocol defines a set of supported operations. A device management agent can support a subset of the operations, but it must make a Manage Device request. A device that supports firmware action operations must also support observation. The Device Management Protocol is built on top of the MQTT messaging protocol.

Unmanaged devices are the devices that do not need to be managed, and Watson IoT Platform no longer sends new device management requests to the device. Unmanaged devices can continue to publish error codes, log messages, and location messages.

By default, the following device management actions are supported by the Watson IoT Platform:

- Device restart
- Factory reset
- Firmware download
- Firmware update

Device management capabilities can be extended by using device management extension packages. A device management extension package is a JSON document that defines at least one device management action. The actions can be initiated for any device that supports the actions by using either the Watson IoT Platform dashboard or the REST API.

References:

https://console.bluemix.net/docs/services/IoT/devices/device_mgmt/index.html

https://console.bluemix.net/docs/services/IoT/devices/device_mgmt/requests.html

https://console.bluemix.net/docs/services/IoT/devices/device_mgmt/custom_actions.html

IBM Training

Device Management: Devices and gateways

- A *device* is anything that has a connection to the internet and has data to send to or receive from the cloud. Devices are used to send event information such as sensor readings to the cloud and to accept commands from applications in the cloud.
- If the devices cannot be directly connected to the internet, a *gateway* device can be built to retrieve and send data to applications in the Watson IoT Platform organization.
- Devices and gateways are registered to Watson IoT Platform by adding the device's information and creating an authentication token for the device to link it with the physical device.
- Each device on Watson IoT Platform should be associated with a device type.
- *Device types* are groups of devices that share characteristics or behaviors.

.

Introduction to IBM Watson IoT Platform

Figure 3-18. Device Management: Devices and gateways

Devices

A *device* is anything that has a connection to the internet and has data to send to or receive from the cloud. You can use devices to send event information such as sensor readings to the cloud and to accept commands from applications in the cloud.

Devices publish data to Watson IoT Platform by using *events*. The device controls the content of the event and assigns a name for each event that is sent. When an event is received by the Watson IoT Platform from a device, the credentials of the connection on which the event was received are used to determine from which device the event was sent. This architecture prevents a device from impersonating another device.

Before the data can be received from IoT devices, devices must be connected to Watson IoT Platform. Connecting a device to Watson IoT Platform involves registering the device with Watson IoT Platform and then using the registration information to configure the device to connect to Watson IoT Platform.

Devices must meet the following requirements for communicating with Watson IoT Platform:

- The device must be able to communicate by using HTTP or MQTT protocols.
- Device messages must conform to the Watson IoT Platform message payload requirements.

Gateways

If the devices cannot be directly connected to the internet, a gateway device can be built to retrieve and send data to applications in the Watson IoT Platform organization. Client libraries, samples, and information are provided to help connect device gateways to the Watson IoT Platform organization and applications. Gateways connect to Watson IoT Platform by using the MQTT or HTTP messaging protocol. Client libraries for developing gateways that can connect to Watson IoT Platform are available in the following languages:

- C++
- C#
- Embedded C
- Java
- Mbed C++
- Node.js
- Node-RED
- Python

The Watson IoT Platform Edge Analytics feature enables Watson IoT Platform Analytics to run on a gateway device. Watson IoT Platform Edge Analytics are used to analyze and respond to data that is collected at the edge of the network. Device data can be also sent to Watson IoT Platform for more analytic processing, dashboard visualization, as input to other analytics, or to be stored in a cloud-based historian repository.

Device types

Each device or gateway on Watson IoT Platform should be associated with a device type. Device types are groups of devices or gateways that share characteristics or behaviors. A device type has a set of attributes. When a device or gateway is added to the Watson IoT Platform, the attributes in its device type are used as a template.

References:

https://console.bluemix.net/docs/services/IoT/devices/device_dev_index.html

https://console.bluemix.net/docs/services/IoT/gateways/gw_dev_index.html

https://console.bluemix.net/docs/services/IoT/iotplatform_task.html

https://console.bluemix.net/docs/services/IoT/gateways/dashboard.html

IBM Training

Device Management: Devices and gateways (cont.)

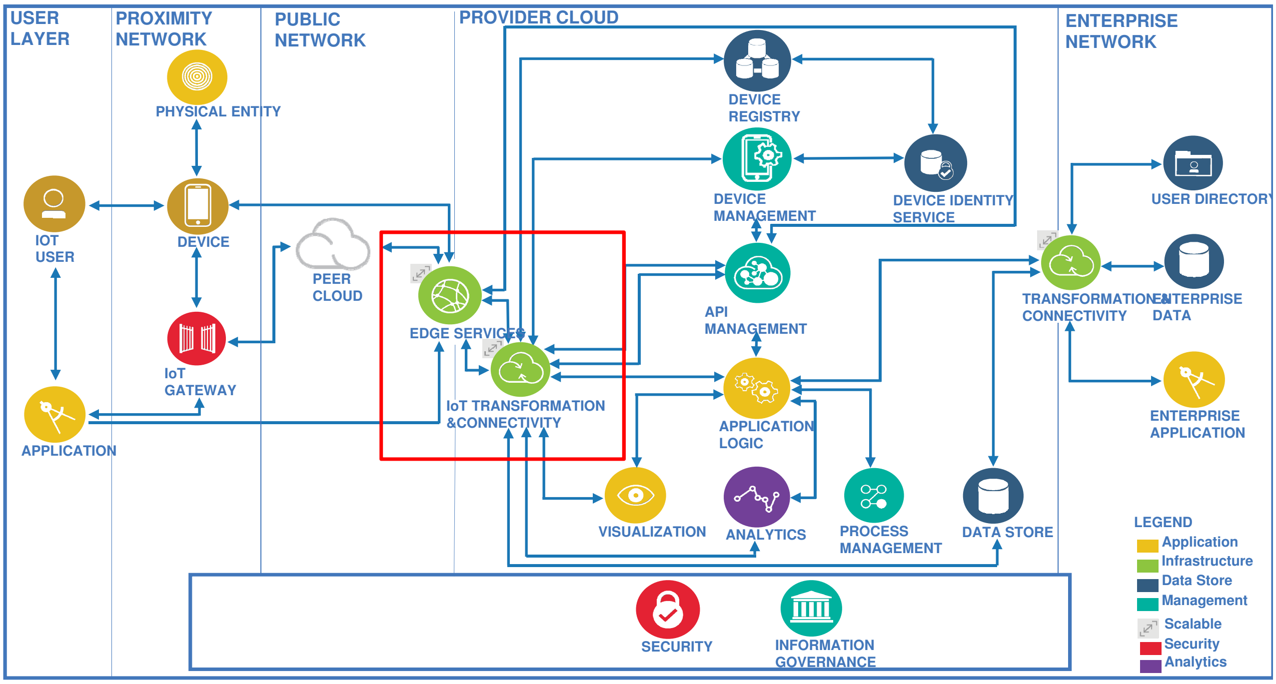
- A device type has a set of attributes. When a device is added to the Watson IoT Platform, the attributes in its device type are used as a template.
- Watson IoT Platform recognizes *devices* and *gateways* as the two classes of device.

Introduction to IBM Watson IoT Platform

Figure 3-19. Device Management: Devices and gateways (cont.)

IBM Training

Device management: Reference architecture (Connectivity)



Introduction to IBM Watson IoT Platform © Copyright IBM Corporation 2019, 2020

Figure 3-20. Device management: Reference architecture (Connectivity)

This figure represents the *IoT reference architecture* and shows how to connect to IoT devices and build quickly scalable apps and visualization dashboards to gain insights from IoT data by using IBM Cloud IoT, data, and AI services.

The highlighted section shows the components that are related to device *connectivity*, which are:

- *IoT transformation and connectivity* enable scalable messaging, transformation, and secure connectivity. You can enable it by using Watson IoT Platform to connect devices with applications to analyze the data and connect the user with these applications and devices.
- *Edge service* enables the capability to deliver content through the internet (DNS, CDN, firewall, and load balancer).

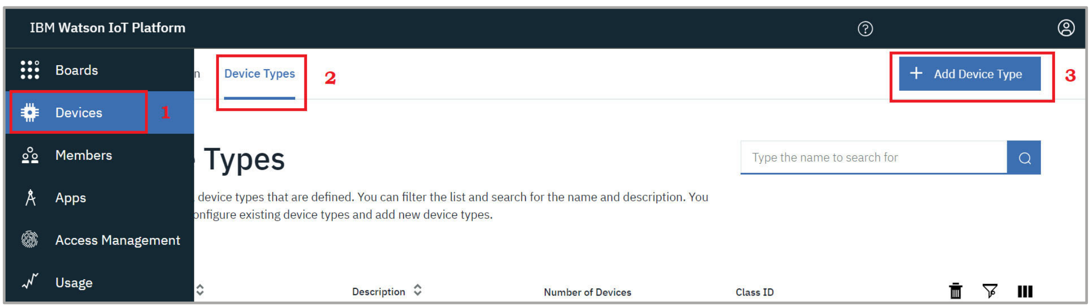
Reference:

https://www.ibm.com/cloud/garage/architectures/iotArchitecture/reference-architecture

IBM Training

Device Management: Device registration

- New devices may be registered to the Watson IoT Platform by clicking **Devices** > **+ Add Device**.
- Create the device type before creating the device and adding it to the platform.
- You can create device types as follows:
  1. From the left navigation pane, click **Devices**.
  2. Select the **Device Types** tab from the top navigation.
  3. Click **+ Add Device Type**.



Introduction to IBM Watson IoT Platform

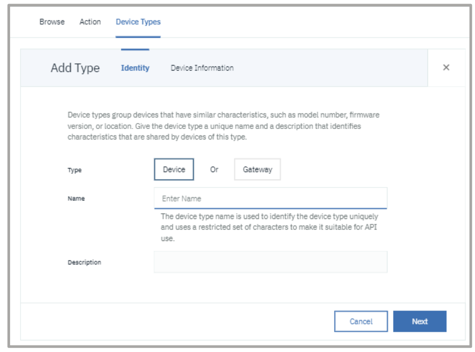
Figure 3-21. Device Management: Device registration

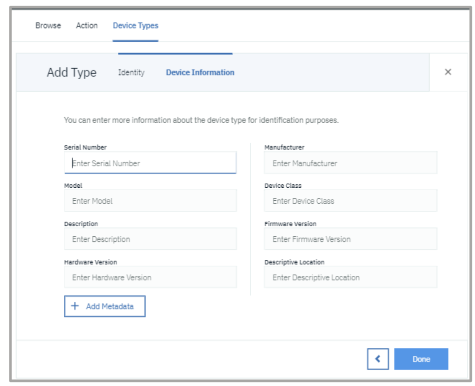
Each device that is connected to the Watson IoT Platform must be associated with a device type. Device types are groups of devices that share common characteristics. Before you create the device, you must add a device type.

IBM Training

Device Management: Device registration (cont.)

4. Choose the device type and provide a unique device name to identify the device, and then click **Next**.
5. Provide the metadata such as Serial Number, Model and Manufacturer, and then click **Done**.





Introduction to IBM Watson IoT Platform

Figure 3-22. Device Management: Device registration (cont.)

The device type name is used to identify uniquely the device type and uses a restricted set of characters to make it suitable for API use.

Description is an optional field that provides a simple description about this device type.

In the device information, there are some optional fields that can be completed, such as the serial number, manufacturer, and model. You also can enter the device metadata in JSON format. The device information can be edited anytime, even after you create the device type.

Reference:

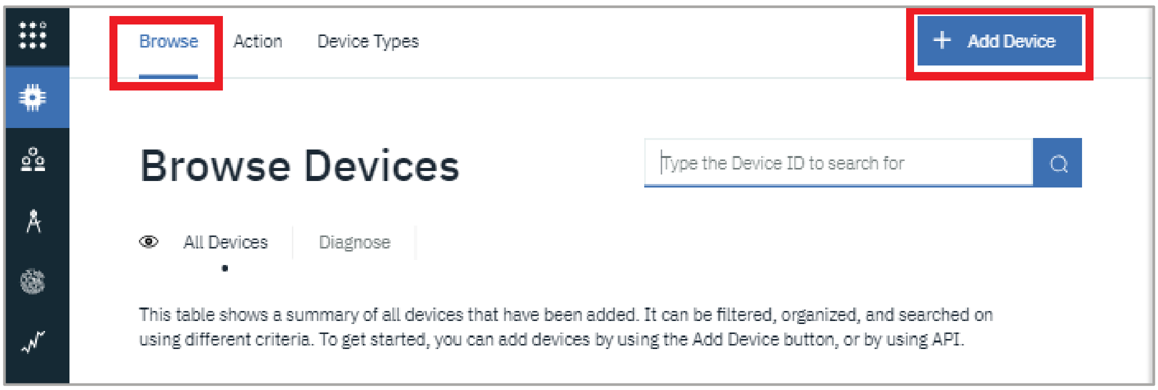
https://console.bluemix.net/docs/services/IoT/iotplatform_task.html

IBM Training

Device Management: Device registration (cont.)

After you create the device type, create the device as follows:

1. From the left navigation pane, click **Devices**.
2. Select **Browse** from the top navigation.
3. Click **+ Add Device**.



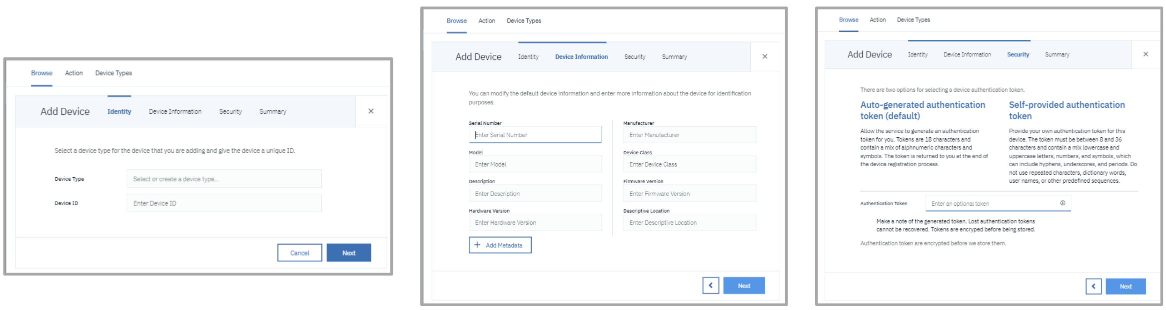
Introduction to IBM Watson IoT Platform

Figure 3-23. Device Management: Device registration (cont.)

IBM Training

Device Management: Device registration (cont.)

4. Select the device type, enter the unique ID for the device, and then click **Next**.
5. Enter the device metadata, such as Serial Number, Model, and Manufacturer, and then click **Next**.
6. Add a secure authentication token or leave the field blank so that the token is generated by Watson IoT Platform, then click **Next**.
7. Review the device data and store the security token in a secure place to retrieve it when it is needed, and then click **Done**.



Introduction to IBM Watson IoT Platform

Figure 3-24. Device Management: Device registration (cont.)

The device ID is used to identify the device in the Watson IoT Platform dashboard, and it is also a required parameter for connecting your device to
Watson IoT Platform.

Hint

For network connected devices, the device ID might be, for example, the device MAC address without any separating colons.

In the device Information, there are some optional fields that can be completed, such as serial number, manufacturer, and model. You also can enter the device metadata in JSON format. The device information can be edited anytime, even after you create the device.

An authentication token is the secure token that is considered the primary mean to connect the created device on Watson IoT Platform with the real physical device or a simulated one.

There are two options for selecting a device authentication token:

- Auto-generated authentication token (default)
- Self-provided authentication token

An auto-generated authentication token generates an authentication token that has 18 characters and contains a mix of alphanumeric characters and symbols.

The self-provided authentication token must be 8 - 36 characters and contains a mix of lowercase and uppercase letters, numbers, and symbols. The token must not contain repeated character sequences, dictionary words, usernames, or other predefined sequences.

Tokens are encrypted before they are stored.

Important

The generated token should be store in a safe place. Lost authentication tokens cannot be recovered.

Reference:

https://console.bluemix.net/docs/services/IoT/iotplatform_task.html

IBM Training

Security policies

- With Watson IoT Platform, you may apply an advanced security level for the devices' connection to the platform.
- The *default connection security rule* is the security rule that is used for all devices in the platform.
- You may create *custom connection rules* for specific device types.
- The security levels, which use Transport Layer Security (TLS), that are available to apply to these rules are as follows:
  - TLS Optional
  - TLS with Token Authentication.
  - TLS with Client Certificate Authentication
  - TLS with Token and Client Certificate Authentication
  - TLS with Client Certificate or Token
- You may add CA certificates to the Watson IoT Platform and use them to authenticate the signature of client-side certificates.

Introduction to IBM Watson IoT Platform

Figure 3-25. Security policies

Default Connection Security is the value that is applied to all devices except for devices that have custom connection settings. These policies affect how the devices connect to the server, but they do not change any settings on the actual device or send any messages to the device. The default security rules are as follows:

- TLS Optional
- TLS with Token Authentication
- TLS with Client Certificate Authentication
- TLS with Token and Client Certificate Authentication
- TLS with Client Certificate or Token

CA certificates can be added to the Watson IoT Platform and used to authenticate the signature of client-side certificates.

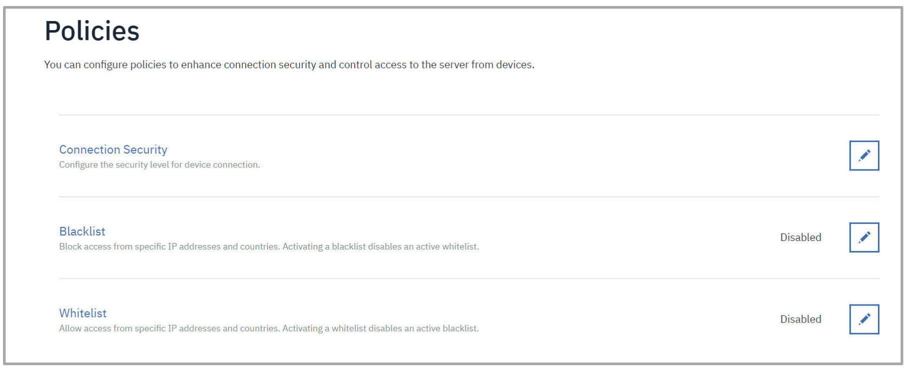
Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/reference/guardian/security_policies.html

IBM Training

Blacklist and whitelist

- A *blacklist* blocks access to the server through certain devices that have specific IP addresses, ranges of IP addresses, or countries.
- A *whitelist* grants access to specific devices that have specific IP addresses, ranges of IP addresses, or countries.
- You may not use both a blacklist and whitelist concurrently; enabling one means disabling the other one.



Introduction to IBM Watson IoT Platform

Figure 3-26. Blacklist and whitelist

Organizations that use advanced security can restrict access to the server through certain devices by using a *blacklist*, or they can use a *whitelist* to grant server access to specific devices. You may not use both a blacklist and whitelist concurrently; enabling one means disabling the other one.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/reference/guardian/security_policies.html

3.4. Application management

IBM Training

Application management

Introduction to IBM Watson IoT Platform

Figure 3-27. Application management

IBM Training

Topics

- Watson IoT Platform features overview
- Watson IoT Platform dashboard
- Device management and security policies
- Application management
- Access management
- Advanced features in Watson IoT Platform
- Watson IoT Platform recipes

Figure 3-28. Topics

IBM Training

Developing applications

- With Watson IoT Platform, you may build and manage applications that interact with and use information from connected devices.
- The following supported protocols, standards, and programming languages are used to develop the applications:
  - Node.js
  - Node-RED
  - MQTT messaging
  - C#
  - Java
  - Python
- Applications can publish device events and commands. Applications are used to process the data that came from registered devices and publish it into Watson IoT Platform.

Introduction to IBM Watson IoT Platform

Figure 3-29. Developing applications

Client libraries, samples, and information are provided to help you build applications for your Watson IoT Platform organization. With Watson IoT Cloud, you may write code to build and manage applications that interact with and use information from connected devices in Watson IoT Platform by using the following supported protocols, standards, and programming languages:

- C#
- Java
- MQTT messaging
- Node.js
- Node-RED
- Python

Here are the application capabilities over Watson IoT Platform:

- An application can publish events as though they came from any registered device.
- An application can publish a command to any registered device.
- An application can subscribe to events from one or more devices.
- An application can subscribe to commands that are being sent to one or more devices.
- An application can subscribe to monitoring messages. Monitoring messages are used to track connection statuses and debug connection issues. Monitoring messages are sent when a device or application connects, disconnects, or fails to connect.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/applications/app_dev_index.html

IBM Training

Developing applications (cont.)

- Applications can subscribe to device events, commands, and status messages to monitor the status of one or more devices.
- Watson IoT Platform sends monitoring messages when a device or application connects, disconnects, or fails to connect. These monitoring messages can be subscribed to through gateways and applications.

Introduction to IBM Watson IoT Platform

Figure 3-30. Developing applications (cont.)

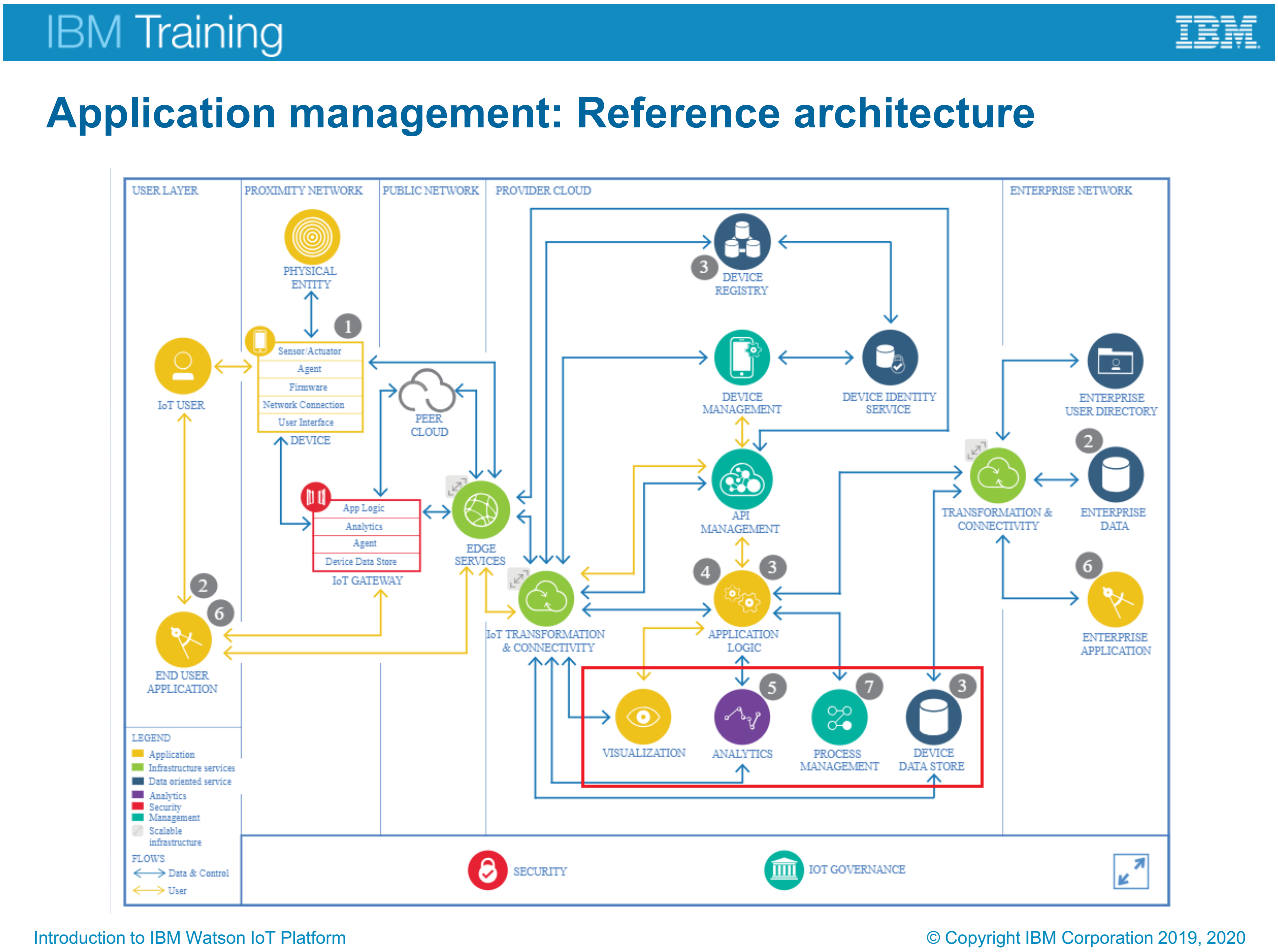


Figure 3-31. Application management: Reference architecture

This figure represents the *IoT reference architecture* and shows how to connect to IoT devices and build quickly scalable apps and visualization dashboards to gain insights from IoT data by using IBM Cloud IoT, data, and AI services.

The highlighted section shows the components that are related to *application management*:

- *Visualization* facilitates the exploration of and interaction with data from the data repositories, actionable insight applications, or enterprise applications. Examples for these applications are IBM Cognos and IBM Statistical Package for the Social Sciences (IBM SPSS).
- *Analytics* guides and automates data analysis, discovery, and visualization. Examples of tools that are used in analytics are IBM BigInsights for Apache Hadoop, IBM Watson Machine Learning, IBM Watson APIs, Apache Spark, and Streaming Analytics.
- *Process management* manages the process workflow. Examples are Watson IoT Platform and IBM Message Hub.
- *Device data store* stores data from IoT devices so that the data can be integrated with processes and applications that are part of the IoT system. Examples of the databases in IBM Cloud are IBM Cloudant NoSQL DB, IBM Db2 Warehouse on Cloud, and IBM Object Storage.

Reference:

https://www.ibm.com/cloud/garage/architectures/iotArchitecture/reference-architecture

IBM Training

Connecting applications

- MQTT is the primary protocol that devices and applications use to communicate with Watson IoT Platform.
- MQTT is a publish and subscribe messaging transport protocol that is designed for the efficient exchange of real-time data between sensor and mobile devices.
- HTTP REST API can be also used to build and customize applications that interact with the organization on Watson IoT Platform.
- Watson IoT Platform HTTP REST API supports the following capabilities and functions for applications:
  - Organization information retrieval
  - Bulk device, device types, and device diagnostic operations
  - Connection problem determination
  - Last event cache
  - Device management request operations
  - Usage management operations
  - Device event publishing
  - Service status querying

Introduction to IBM Watson IoT Platform

Figure 3-32. Connecting applications

MQTT

MQTT is the primary protocol that devices and applications use to communicate with the Watson IoT Platform. MQTT is a publish and subscribe messaging transport protocol that is designed for the efficient exchange of real-time data between sensor and mobile devices. MQTT runs over TCP/IP, and although it is possible to code directly to TCP/IP, the MQTT protocol can be handled by using a library that can handle these details. A wide range of MQTT client libraries are available. IBM contributes to the development and support of several client libraries. Applications and devices work with different MQTT topic spaces. Devices work within a device-scoped topic space, and applications have full access to the topic space for an entire organization.

HTTP REST API

With your HTTP REST API, you can build and customize applications that interact with the organization on Watson IoT Platform. The Watson IoT Platform HTTP REST API supports the following capabilities and functions for applications:

- Organization information retrieval
- Bulk device operations (list, add, and remove)
- Device type operations (list, create, delete, view details, and update)
- Device operations (list, add, remove, view details, update, view location, and view management information)
- Device diagnostic operations (clear logs, retrieve logs, add log information, delete logs, get specific logs, clear error codes, get device error codes, and add error codes)
- Connection problem determination (list device connection log events)
- Last event cache (view the last event for a specific device)
- Device management request operations (list device management requests, initiate requests, clear request status, get details of a request, list all request statuses by device, and get the request status for a specific device)
- Usage management operations (retrieve the total amount of data used)
- Device event publishing (HTTP messaging APIs)
- Service status querying (retrieve service statuses by organization)

In addition to using the MQTT messaging protocol, it is also possible to configure applications to publish events and commands to the Watson IoT Platform over HTTP.

References:

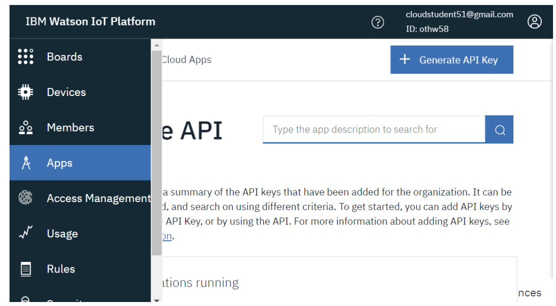
https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/applications/mqtt.html

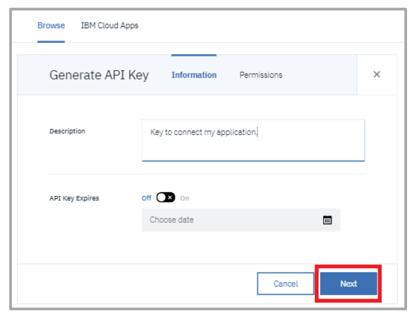
https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/reference/api.html

IBM Training

Connecting applications: API key connection

- Connecting an application to Watson IoT Platform can be achieved either by using *API keys and tokens* or *by binding the application directly from IBM Cloud to the Watson IoT Platform service*.
- To create API keys and an authentication token, complete the following steps:
  1. From the left navigation pane, click **Apps**.
  2. Click **+ Generate API Key**.
  3. (Optional) Insert a description for the API key and set the expiry date for the key.
  4. Click **Next**.





Introduction to IBM Watson IoT Platform

Figure 3-33. Connecting applications: API key connection

To connect an application to Watson IoT Platform, either use API keys and tokens or bind your application directly to Watson IoT Platform in IBM Cloud.

API key connection

API keys are used when connecting applications to a Watson IoT Platform organization. Applications require an API key to connect to an organization, and a unique authentication token must be used with that API key.

IBM Cloud binding connection

Applications may be bound to a Watson IoT Platform organization from IBM Cloud. By binding the application, it can communicate with service instances only in the same space or organization. All the required data for the application to communicate with the service instance may be found in the *VCAP_SERVICES* environment variable. If the application is bound to multiple services, the *VCAP_SERVICES* variable includes the connection information for each service instance. However, a user can use service instances from other spaces or organizations in the same way that an external app does. Instead of creating a binding, use the credentials to configure directly your app instance.

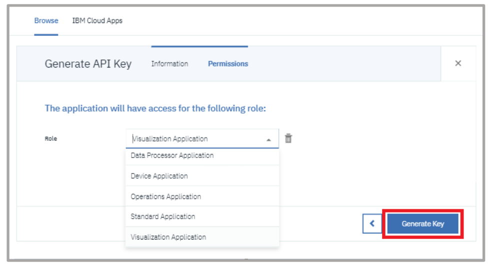
Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/applications/app_dev_index.html#api-key

IBM Training

Connecting applications: API key connection (cont.)

5. Select a role for the application, and then click **Generate Key**.
6. Make a note of the API key and token pair. Authentication tokens are unrecoverable. If this token is lost or is forgotten, the API key must be reregistered to generate a new authentication token.



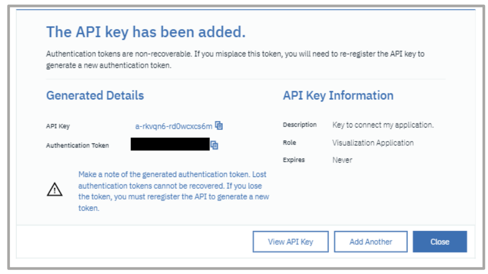


Figure 3-34. Connecting applications: API key connection (cont.)

3.5. Access management

Figure 3-35. Access management

IBM Training

Topics

- Watson IoT Platform features overview
- Watson IoT Platform dashboard
- Device management and security policies
- Application management
- Access management
- Advanced features in Watson IoT Platform
- Watson IoT Platform recipes

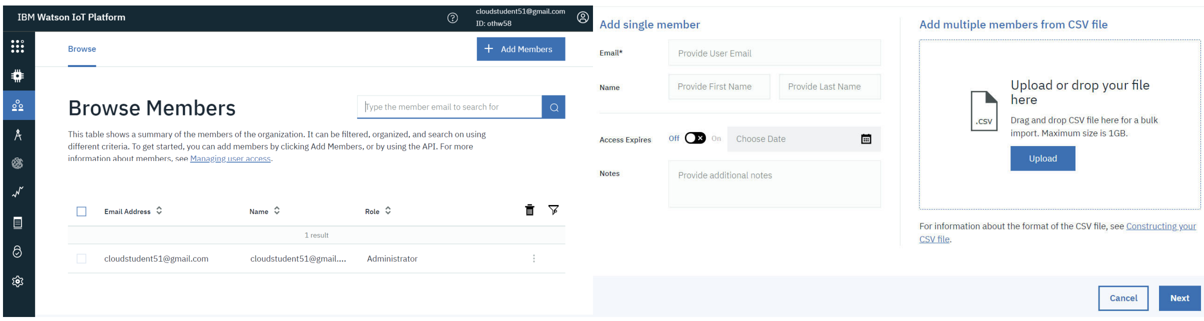
Introduction to IBM Watson IoT Platform

Figure 3-36. Topics

IBM Training

Managing user access

With Watson IoT Platform, you can control and manage access to the organization.



Introduction to IBM Watson IoT Platform

Figure 3-37. Managing user access

With Watson IoT Platform, you a control and manage access to the organization. You can add members by adding, inviting, or importing them. Each member should be assigned to a role to define their access level over the platform. You can add users either by adding individual members by using the Add function or by adding members simultaneously by using the Import function. By default, members accounts do not expire. While adding users to a Watson IoT Platform organization, you optionally can set an expiry date for their account.

To add a single member, complete the following steps:

1. In the Watson IoT Platform dashboard, go to **Members**.
2. Click **Add Members** and select the **Add** tab.
3. Enter the IBMid of the member.
4. Click **Next**
5. Select a role for the member.
6. Optional: Set an expiry date for the member.
7. Click **Add**.

To add multiple members simultaneously, upload a `.csv` file that contains the IBMid, the role, and the optional expiry date of each member by completing the following steps:

1. In the Watson IoT Platform dashboard, go to **Members**.
2. Click **Add Members** and select the **Import** tab.
3. Browse your files or drag the .csv file into the **Upload CSV** window.
4. Select a default role to use if a role that is specified in the .csv file is not recognized.
5. Map the column numbers in your .csv file to the corresponding IBMid, role, and (optional) expiry date entries.
6. Select the appropriate comma or semicolon column separator to match the separator that is used in your .csv file.
7. Click **Import** to import the IBMids and create the members.

Here is a sample .csv file with comma delimitation:

```
user1@sample.com,PD_DEVELOPER_USER,2018-03-13
user2@sample.com,PD_OPERATOR_USER,2018-03-13
user3@sample.com,PD_ADMIN_USER,2018-03-13
```

Users can be edited, blocked, have their access limited to specific devices, or removed entirely from the Watson IoT Platform organization.

Reference:

https://cloud.ibm.com/docs/services/IoT/add_users?topic=iot-platform-managing-user-access

IBM Training

Managing roles

- *Roles* are sets of permissions that can be used to grant or restrict access to specific operations.
- Roles can be used to manage permissions for groups of users, applications, and gateways.
- *User roles (Member roles)* are assigned to the users while adding, inviting, or registering a user to Watson IoT Platform.
- *Application roles (API roles)* are assigned to grant or deny applications access to specific operations.
- By default, there is a set of predefined roles for both user and application roles.
- The predefined roles cannot be updated or removed from the Watson IoT Platform, but it is possible to create custom roles.
- Custom roles may have unique combinations of permissions that are available to the predefined roles.

Introduction to IBM Watson IoT Platform

Figure 3-38. Managing roles

Roles are sets of permissions that can be used to grant or restrict access to specific operations. Roles are used to manage permissions for groups of users, applications, and gateways.

User roles

User roles can be assigned when adding, inviting, or registering a user in Watson IoT Platform. User roles can also be assigned or be changed at any time by using the Watson IoT Platform dashboard. The following standard user roles are available as the predefined user roles:

- Administrator: A *super-user* role that grants access to all user-related APIs. Administrators cannot access operations that are restricted to devices and applications.
- Operator: Intended for front-end organization users. Grants access to most organization operations, access control operations, third-party operations, and risk management operations.
- Developer: Grants unrestricted access to device operations, log operations, cache operations, historical operations, and third-party service operations. This role provides limited access to organization, access control, and risk management operations.
- Reader: The default user role. Grants limited access to operations that are available to all users.
- Analyst: This role has the same access level as the reader role, but it is intended for the analyst who is responsible for viewing the processed data and events on the platform.

Application roles

Application roles can be assigned to grant or deny applications access to specific operations. All application roles deny access to the following operations:

- All risk management operations.
- Configure storage parameters.
- Configure authentication providers.
- Create, update, or delete email configurations.

The following standard application roles are available as the predefined application roles:

- Standard: The default application role. Grants access to most application operations but no user or role operations.
- Operations: Grants access to the broadest range of operations but denies access to subscribe or publish operations.
- Back-end Trusted: Intended for applications that do not require interaction from the systems operator. Denies access to device management, organization, role, or extension operations.
- Data Processor: Intended for applications that perform analytics and data processing. Data Processor applications are granted limited access to organization operations and user operations.
- Visualization: Intended for applications that are responsible for generating visualizations of data. Visualization applications can access live and stored data operations and dashboard operations.
- Device: Intended for applications that take the role of devices, that is, they provide a source of data that is sent to Watson IoT Platform as though it is a device. Device applications are granted only limited access to operations.

The predefined roles cannot be update or removed from the Watson IoT Platform, but it is possible to create custom roles.

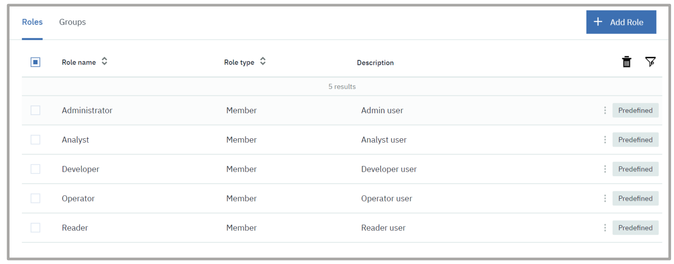
Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/reference/rbac/index.html#user_roles

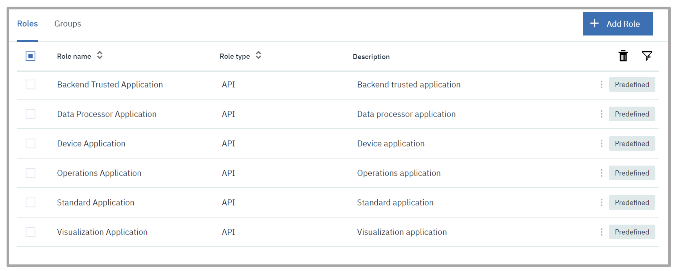
IBM Training

Managing roles (cont.)

- User predefined roles are as follows:
  - Administrator
  - Analyst
  - Developer
  - Operator
  - Reader



- Application predefined roles are as follows:
  - Back-end Trusted
  - Data Processor
  - Device
  - Operations
  - Standard
  - Visualization



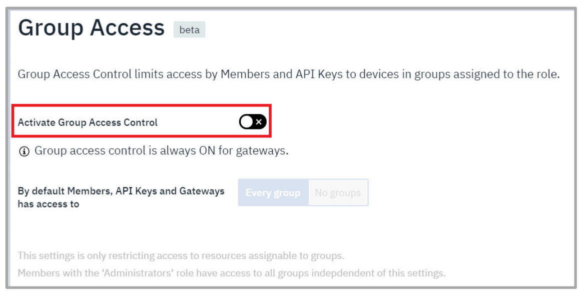
Introduction to IBM Watson IoT Platform

Figure 3-39. Managing roles (cont.)

IBM Training

Managing group access

- *Groups* are used to grant members and API keys access to specific devices.
- Devices are added to a group during its creation, and then members and API keys may be added to the group with assigned roles within the group.
- The combination of roles and groups determines which devices the users and API keys can access and the actions that they can perform on the devices.
- Group access control for Members and API keys is disabled by default, but it may be activated from the Settings section.



Introduction to IBM Watson IoT Platform

Figure 3-40. Managing group access

Users with the administrator role can use Watson IoT Platform groups to grant members and API keys access to specific devices. After the creation of a group and adding devices to it, members and API keys may be added to the group, and roles may be assigned to them within the group. The combination of roles and groups determines which devices the users and API keys can access and the actions that they can perform on the devices.

Groups may be managed by using the Watson IoT Platform dashboard user interface or by using the Watson IoT Platform access control APIs.

Important

The Groups feature in the Watson IoT Platform UI is available only as part of a limited beta program. Future updates might include changes that are incompatible with the current version of this feature

3.6. Advanced features in Watson IoT Platform

IBM Training

Advanced features in Watson IoT Platform

Introduction to IBM Watson IoT Platform

Figure 3-41. Advanced features in Watson IoT Platform

IBM Training

Topics

- Watson IoT Platform features overview
- Watson IoT Platform dashboard
- Device management and security policies
- Application management
- Access management
- Advanced features in Watson IoT Platform
- Watson IoT Platform recipes

Introduction to IBM Watson IoT Platform

Figure 3-42. Topics

IBM Training

Watson IoT Platform Edge

- Watson IoT Platform Edge extends Watson IoT Platform capabilities to edge devices.
- *Edge devices* are on the edge of a network within an organization, such as sensors and industrial controllers at a physical location.
- Watson IoT Platform Edge allows the data to be processed inside of the devices before it is sent to the cloud.
- Watson IoT Platform Edge may be used as follows:
  1. Create edge nodes that consist of a gateway and a device that is at the edge of the network.
  2. Configure those nodes with the services that run inside of them.
  3. Watson IoT Platform Edge deploys those services to the nodes.
  4. Edge nodes can send messages from devices by using the edge gateway. The gateway processes the messages, transforms the data, and sends the data to the cloud, depending on how the interfaces are configured.

Introduction to IBM Watson IoT Platform

Figure 3-43. Watson IoT Platform Edge

Watson IoT Platform Edge extends Watson IoT Platform capabilities to edge devices. *Edge devices* are on the edge of a network within an organization. Examples include sensors and industrial controllers at a physical location, such as a factory.

With Watson IoT Platform Edge, data can be processed inside of the devices before it is sent to the cloud. To use Watson IoT Platform Edge, edge nodes are created and configured with the services that are to be run inside of them. Then, Watson IoT Platform Edge deploys those services to the nodes.

With the Watson IoT Platform Edge services running in them, edge nodes can send messages from devices with the edge gateway. The gateway processes the messages, transforms the data, and sends the data to the cloud, depending on how the interfaces are configured.

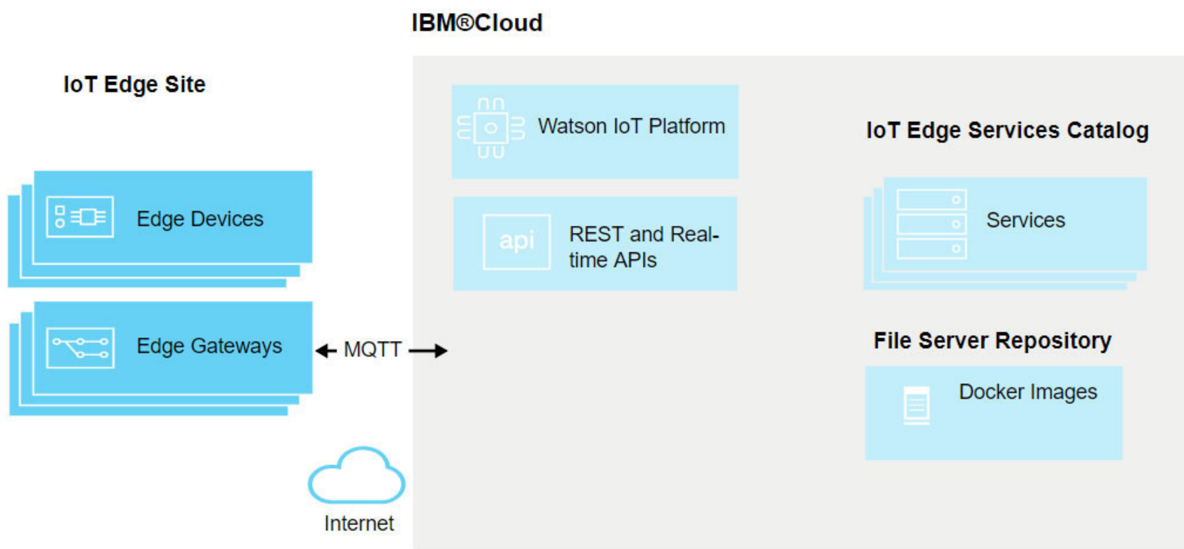
The Watson IoT Platform Edge preview provides the Core IoT default service, along with the ability to create custom services, such as a service to predict failures in a manufacturing floor robot and send failure notifications. An example of an edge device is a sensor that monitors CPU usage and sends an alert when usage exceeds a certain percentage.

Important

The Watson IoT Platform Edge feature is available only as part of a limited preview program. Future updates might include changes that are incompatible with the current version of this feature.

IBM Training

Watson IoT Platform Edge (cont.)



Introduction to IBM Watson IoT Platform

Figure 3-44. Watson IoT Platform Edge (cont.)

Here are the Watson IoT Platform Edge components:

- Edge node: An edge node consists of a gateway and a device that is at the edge of the network.
- Edge device: A device, such as a sensor, that is at the edge of a network and runs services that process data before it is sent to the cloud. An example of an edge device is a sensor that monitors CPU usage and sends an alert when usage exceeds a certain percentage.
- Edge gateway: Gateways are specialized devices that have the combined capabilities of an application and a device, which allows them to serve as access points for other devices. When Watson IoT Platform Edge is enabled, edge gateways enable devices that cannot connect directly to the internet to access the Watson IoT Platform service by first connecting to the gateway device at the edge.
- Edge gateway type: A gateway type groups gateway devices that share attributes, such as model number or location. When Watson IoT Platform Edge capabilities are enabled and an edge gateway device is added to Watson IoT Platform, the attributes from the edge gateway type are used as a template for the new edge gateway device.
- Edge services: Services are capabilities that are added to a Watson IoT Platform Edge gateway and perform processes such as data manipulation. You build applications that use the services.

- Watson IoT Platform Edge Services Catalog: The default Core IoT Service is included in the catalog and added to all edge nodes. You can add custom services based on your business needs. The Watson IoT Platform Edge Services Catalog holds all the available custom services.
- File Server Repository: The File Server Repository holds Docker containers and images. All the Watson IoT Platform Edge processing capabilities or services work inside Docker containers. You select the Docker images that are run inside of the devices.

IBM Training

Simulating device data

- Watson IoT Platform allows the creation of simulated events for devices.
- The simulated events data can be used to learn about, test, and demonstrate fully functioning Watson IoT Platform features.
- Event details may be configured for each device, or a default configuration may be applied to all devices.
- Simulated event configuration may be exported so that it can be reused or shared to set up other simulations.
- This feature may be enabled from the settings in the left navigation pane.

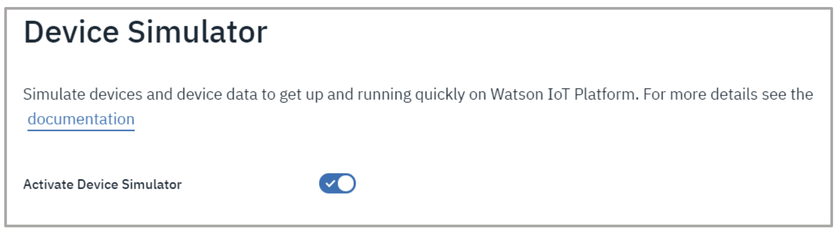


Figure 3-45. Simulating device data

By using the Watson IoT Platform device simulator, simulated events may be configured for devices. Simulated event data may be used to learn about, test, and demonstrate fully functioning Watson IoT Platform features. Existing device types, devices, and the simulator may be used to generate new devices for existing types. Event details can be configured for each device, or you may set a default configuration that is applied to all devices. A simulated event configuration may be exported so that it can be reused or shared to set up other simulations.

To simulate device data, complete the following steps:

1. From the left navigation pane, select **Settings**.
2. In the Data and Devices section, activate the device simulator.
3. From the main navigation pane, select **Devices**. A message in the lower right of the window indicates that no simulations are running.
4. Click the "0 Simulations running" message.
5. In the Simulations window, click **Create Simulation**.
6. Select an existing device type that you want to simulate data for or create a device type by typing the device type name.
7. Click **Import/Export Simulation** to use an existing configuration, or manually configure the simulation details for the device type:
   a. Click **+ New Event Type** to add an event type to the device type.
   b. Enter a name for the event type.
   c. Under Schedule, set the frequency of the event.
   d. Edit the event type details in JSON format and save the updated event type.
8. Select a registered device for the simulation.
9. To create more simulations, click **+ New Simulation** and repeat the configuration steps for each device to which you want to apply custom settings. The message in the lower right of the window shows the number of active simulations.
10. Optional: After configuring simulations for devices, export the simulation details so that they can be reused or shared:
    i. In the **Simulations** window, click **Import/Export**.
    ii. Select the **Export** tab.
    iii. Copy the simulation details to the clipboard or download them in a JSON file.
11. On the Devices page, browse to one of the simulated devices and select the **Recent Events** tab to verify that the simulated events are working.

Reference:

https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/platform/reference/dashboard/device_sim.html

3.7. Watson IoT Platform recipes

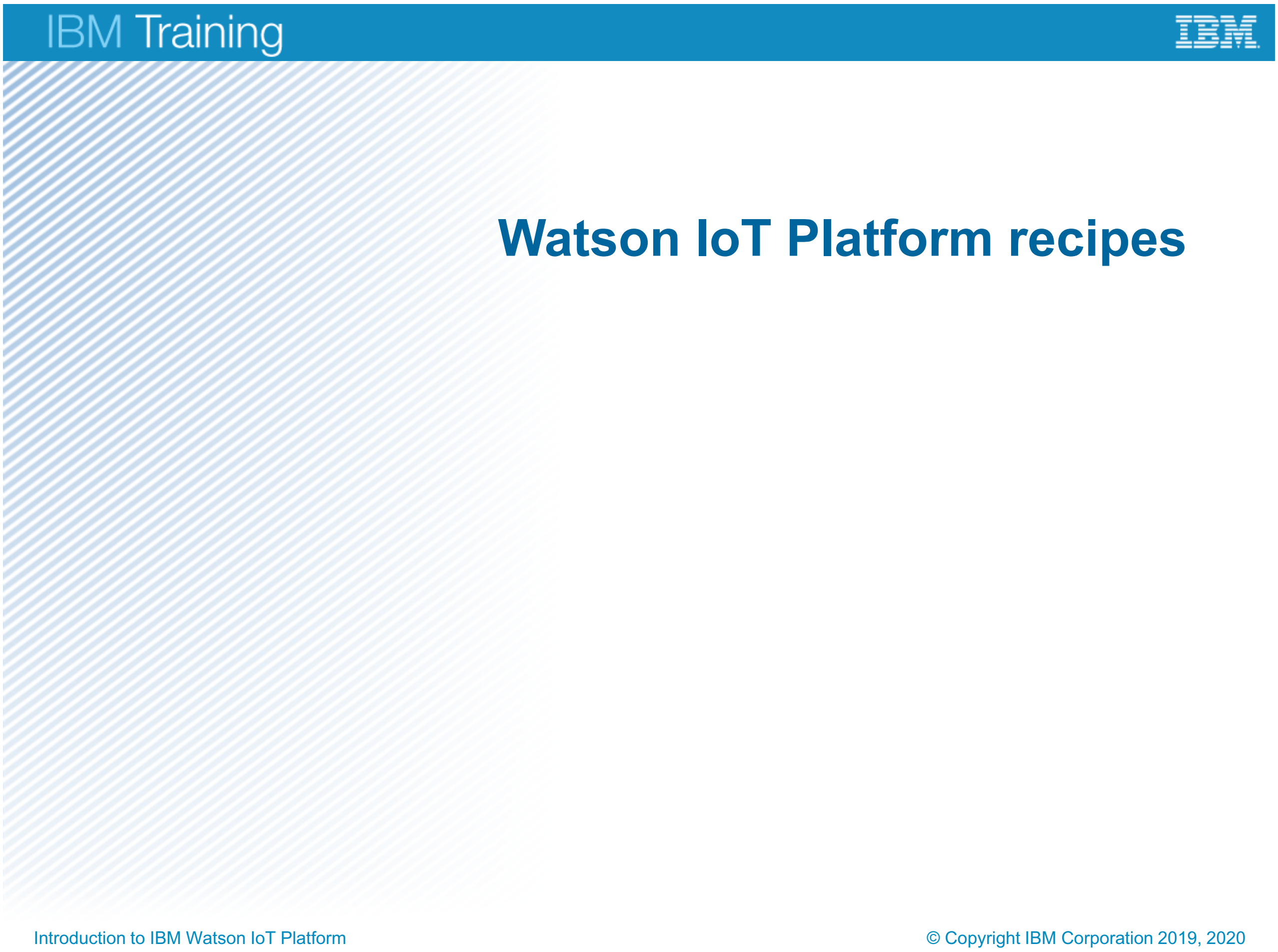


Figure 3-46. Watson IoT Platform recipes

IBM Training

Topics

- Watson IoT Platform features overview
- Watson IoT Platform dashboard
- Device management and security policies
- Application management
- Access management
- Advanced features in Watson IoT Platform
- Watson IoT Platform recipes

Introduction to IBM Watson IoT Platform

Figure 3-47. Topics

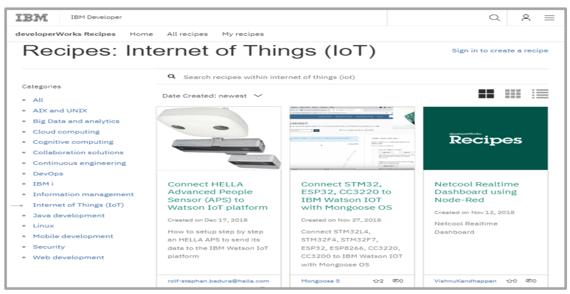
IBM Training

Watson IoT Platform recipes

- *IBM Developer* is the main IBM developers' community for Watson IoT Platform recipes. It is a complete, web-based community that has many recipes that are developed by other developers. You can find these recipes at the following website:

  https://developer.ibm.com/recipes/tutorials/category/internet-of-things-iot/

- Here are some example showcases that use Watson IoT Platform:
  - Connect an Arduino Uno device to the IBM Watson IoT Platform.
  - Connect a Raspberry Pi to IBM Watson IoT Platform.



Introduction to IBM Watson IoT Platform

Figure 3-48. Watson IoT Platform recipes

References:

- IBM IoT Recipes Home: https://developer.ibm.com/recipes/tutorials/category/internet-of-things-iot/
- Connect a Raspberry Pi to Watson IoT Platform: https://developer.ibm.com/recipes/tutorials/raspberry-pi-4/
- Connect an Arduino Uno device to the Watson IoT Platform: https://developer.ibm.com/recipes/tutorials/connect-an-arduino-uno-device-to-the-ibm-internet-of-things-foundation/

IBM Training

Using The Force: Move a BB-8 with your mind

Watch this video: https://youtu.be/K-4mp_e5v4A

Introduction to IBM Watson IoT Platform

Figure 3-49. Using The Force: Move a BB-8 with your mind

Here is another example for IoT recipes that demonstrate how you can use your mind to move BB-8 droids (the BB-8 droid is a robot that is in the Star Wars film series).

References:

- https://youtu.be/K-4mp_e5v4A
- https://www.ibm.com/blogs/bluemix/2016/04/taking-1st-steps-using-the-force/
- https://developer.ibm.com/recipes/tutorials/control-your-spherobb8-using-ibm-watson-iot-platform-and-nodered/
- https://www.ibm.com/blogs/bluemix/2015/12/the-force-bb8-emotiv-insight-bluemix/

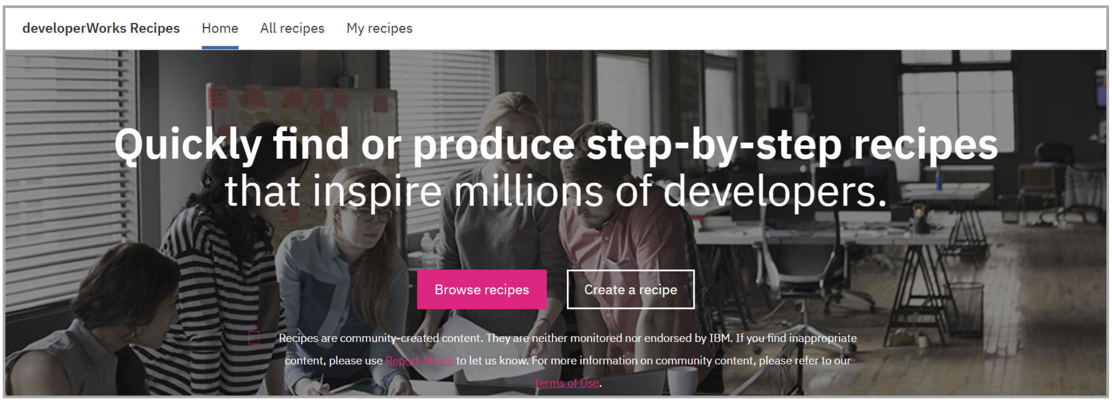
Third-party hardware

- Emotiv Insights: https://emotiv.com/insight.php
- Sphero BB-8 Droid: http://www.sphero.com/starwars

IBM Training

Creating your own IoT recipes

- In addition to having access to other developer recipes, you can create your own IoT recipes and publish them on the IBM Developer recipes communities.
- To create a recipe, go to https://developer.ibm.com/recipes/, click **Create a recipe**, and then follow the steps to create your recipe.



Introduction to IBM Watson IoT Platform

Figure 3-50. Creating your own IoT recipes

Reference:

https://developer.ibm.com/recipes/tutorials/recipe_guidelines/

IBM Training

Unit summary

- List the features of Watson IoT Platform.
- Use the Watson IoT Platform dashboard.
- Manage devices and set up security policies on Watson IoT Platform.
- Manage user access on Watson IoT Platform.
- Manage applications on Watson IoT Platform.
- Describe the advanced features of Watson IoT Platform.
- Use recipes to help jump-start application development.

Introduction to IBM Watson IoT Platform

Figure 3-51. Unit summary

IBM Training

Review questions

1. True or False: Your Watson IoT Platform organization may be created without registering with IBM Cloud.

2. True or False: Devices that do not support either the MQTT protocol or REST API may be connected to Watson IoT Platform.

3. ____ are groups of devices that share characteristics or behaviors.
   A. Device models
   B. Device types
   C. Device groups
   D. Gateways

4. True or False: Watson IoT Platform predefined roles cannot be modified.

Introduction to IBM Watson IoT Platform

Figure 3-52. Review questions

IBM Training

Review questions (cont.)

5. Connecting an application to Watson IoT Platform requires _____.
   A. Binding the application directly from IBM Cloud
   B. API key
   C. Transport Layer Security (TLS)
   D. API key and authentication token
   E. A and B
   F. A and D

Introduction to IBM Watson IoT Platform

Figure 3-53. Review questions (cont.)

IBM Training

Review answers

1. True or False: Your Watson IoT Platform organization may be created without registering with IBM Cloud.

2. True or False: Devices that do not support either the MQTT protocol or REST API may be connected to Watson IoT Platform.

3. ____ are groups of devices that share characteristics or behaviors.
   A. Device models
   B. Device types
   C. Device groups
   D. Gateways

4. True or False: Watson IoT Platform predefined roles cannot be modified.

Introduction to IBM Watson IoT Platform

Figure 3-54. Review answers

1. False. Watson IoT Platform is a cloud-hosted service, so it requires an IBM Cloud account to use it.
2. True. Devices that do not support either the MQTT protocol or REST API still can be connected to the Watson IoT Platform through a gateway device that supports the MQTT protocol or REST API. The gateway is connected to Watson IoT Platform.
3. B. Device types.
4. True.

IBM Training

Review answers (cont.)

5. Connecting an application to Watson IoT Platform requires _____.
   A. Binding the application directly from IBM Cloud
   B. API key
   C. Transport Layer Security (TLS)
   D. API key and authentication token
   E. A and B
   F. A and D

Figure 3-55. Review answers (cont.)

5. F. A and D. Connecting an application to Watson IoT Platform can be achieved either by using an API key and an authentication token or by binding the application directly from IBM Cloud.

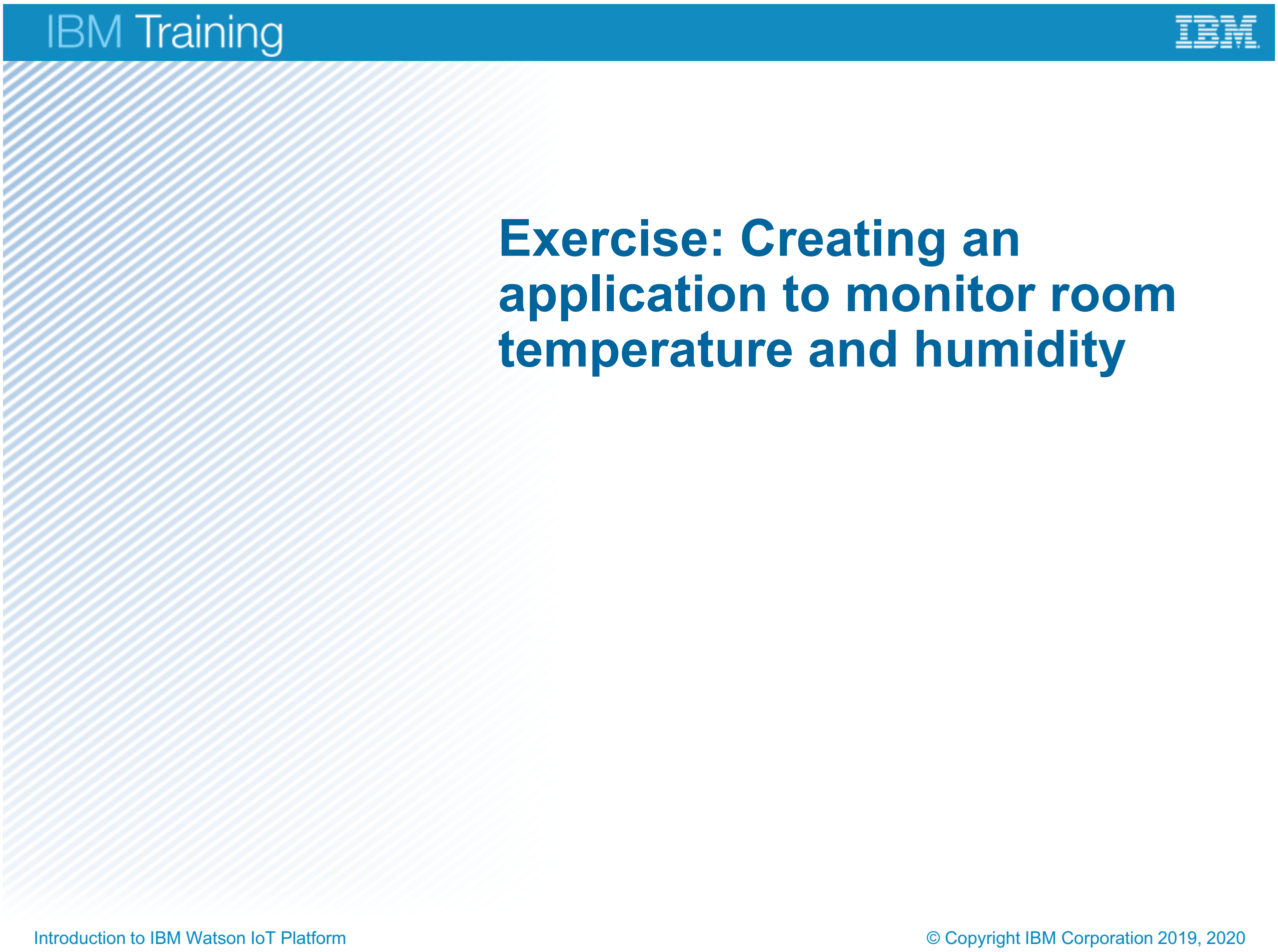


Figure 3-56. Exercise: Creating an application to monitor room temperature and humidity

IBM Training

Exercise objectives

- This exercise helps you understand how to create a simple IoT system by using Watson IoT Platform to create a sample application to monitor the temperature and humidity of a facility and optimize the use of air conditioning. You use the Watson IoT Platform device simulator to simulate the devices that send temperature and humidity values through the internet to Watson IoT Platform, and then you visualize the data that is sent by the devices in real time.
- After completing this exercise, you should be able to:
  - Create a Watson IoT Platform service on IBM Cloud.
  - Define a device on Watson IoT Platform and retrieve the credentials.
  - Use the Watson IoT Platform simulator to monitor temperature and humidity.
  - Chart the data on Watson IoT Platform.

Introduction to IBM Watson IoT Platform

Figure 3-57. Exercise objectives

After completing this exercise, you should be able to:

- Create a Watson IoT Platform service on IBM Cloud.
- Define a device on Watson IoT Platform and retrieve the credentials.
- Use the Watson IoT Platform simulator to monitor temperature and humidity.
- Chart the data on Watson IoT Platform.

Unit 4. Introduction to Node-RED

Estimated time

01:00

Overview

This unit introduces Node-RED, a flow-based programming tool for wiring together hardware devices, APIs, and online services. This unit describes the Node-RED flow editor and its components.

IBM Training

Unit objectives

- Explain the concept of flow-based programming.
- Explain what Node-RED is.
- Describe the Node-RED workspace and its components.
- Describe the Node-RED palette and the Palette Manager and explain how to use them to build flows.

Introduction to Node-RED

Figure 4-1. Unit objectives

4.1. Introduction to flow-based programming

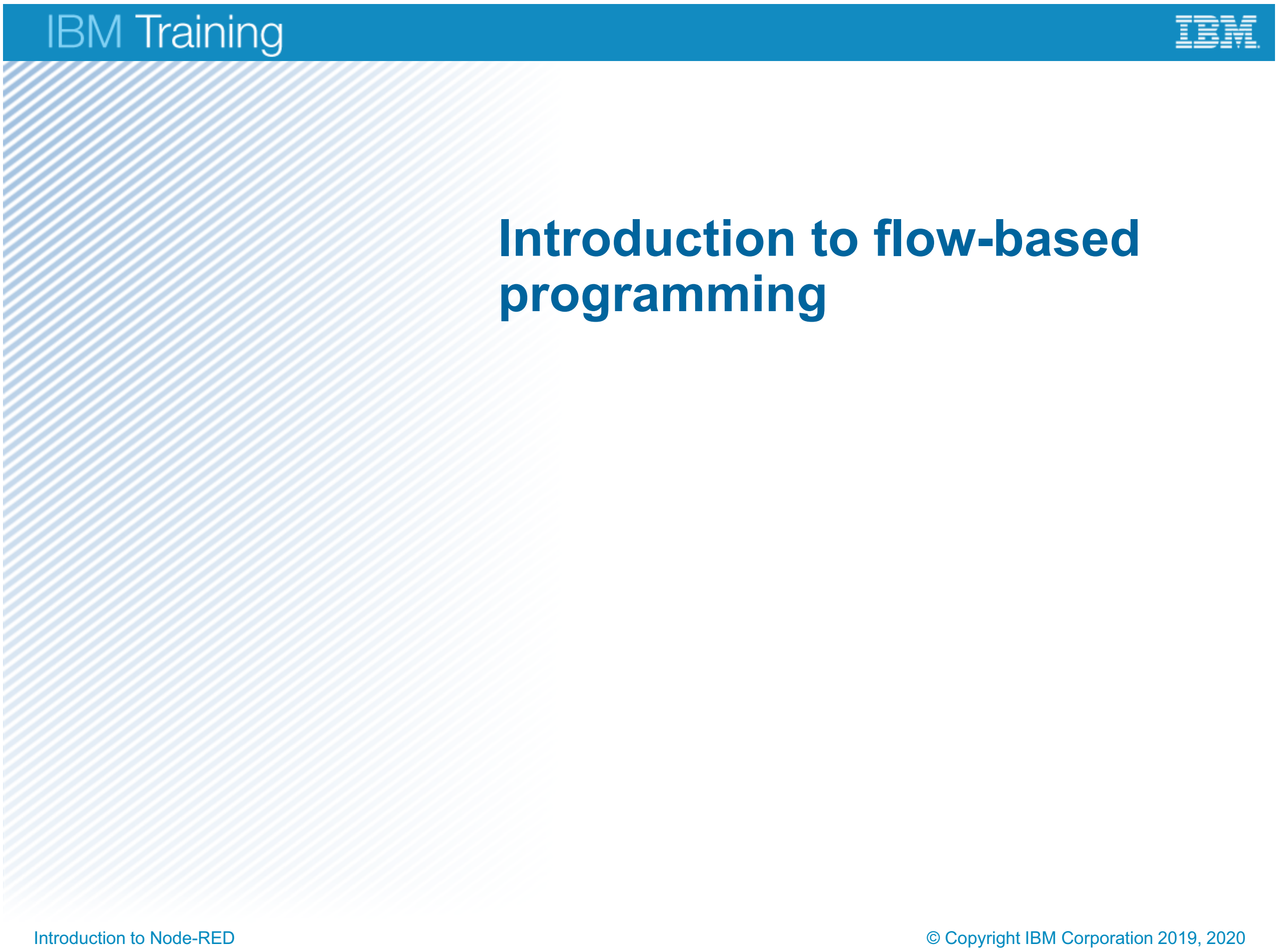


Figure 4-2. Introduction to flow-based programming

IBM Training

Topics

- Introduction to flow-based programming
- Node-RED overview
- Node-RED flow editor
- Node-RED Palette Manager

Introduction to Node-RED

Figure 4-3. Topics

IBM Training

Introduction to flow-based programming

- Flow-based programming (FBP) is a programming model that was invented by J. Paul Rodker Morrison in the late 1960s.
- The model defines applications as a set of “black box” processes that communicate by using streams of data chunks that flow through a network of asynchronous processes. This data exchange can be done through message passing.
  - A *black box* in computer science is the system or object that can be viewed in terms of its inputs and outputs without knowing its internal structure.
  - *Message passing* is a technique that is used in software development as a way for the application objects to interact with each other.

Introduction to Node-RED

Figure 4-4. Introduction to flow-based programming

Flow-based programming (FBP) views an application as a network of asynchronous processes that communicate by using streams of structured data chunks that are called *information packets*. In this view, the focus is on the application data and the transformations that are applied to it to produce the wanted outputs. The network is defined externally to the processes as a list of connections that is interpreted by a piece of software that is called the "scheduler".

FBP is an effective way to produce reliable, maintainable, and large business applications. There is increasing interest in it worldwide.

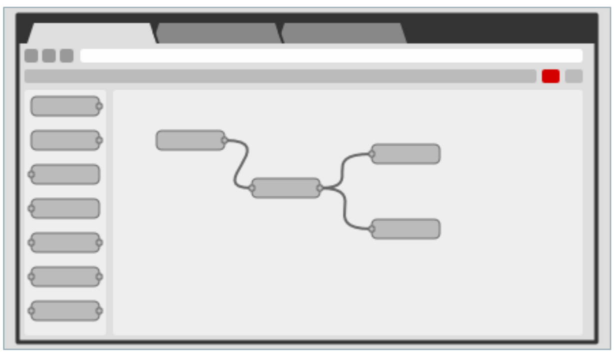
References:

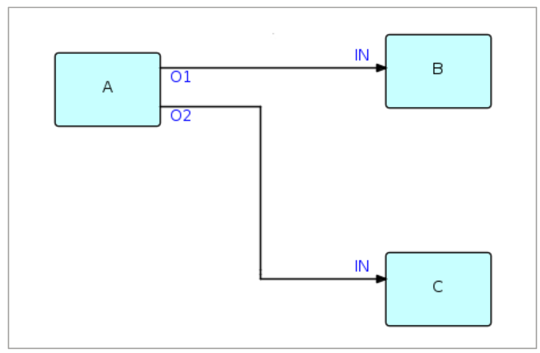
- http://www.jpaulmorrison.com/fbp/
- https://en.wikipedia.org/wiki/

IBM Training

Flow-based programming characteristics

- FBP is visual and component-oriented, which means that the processes inside an application can be reconnected endlessly to form different applications without having to be changed internally.
- FBP is a “coordination language”, not a programming language.
- FBP makes it simple to reconfigure the application’s components.
- FBP is language-independent.





Introduction to Node-RED

Figure 4-5. Flow-based programming characteristics

FBP characteristics:

- FBP is component-oriented, which means that the processes inside an application can be reconnected endlessly to form different applications without having to be changed internally.
- FBP makes it simple to reconfigure the application’s components, which means you do not have to change or open coding files to modify the application. You may modify the nodes, switch them, or add new or delete nodes.

4.2. Node-RED overview

IBM Training

Node-RED overview

Introduction to Node-RED

Figure 4-6. Node-RED overview

IBM Training

Topics

- Introduction to flow-based programming
- Node-RED overview
- Node-RED flow editor
- Node-RED Palette Manager

Introduction to Node-RED

Figure 4-7. Topics

IBM Training

What is Node-RED

- Node-RED is a programming tool for wiring together hardware devices, APIs, and online services.
- Node-RED was developed in early 2013 as a side-project by Nick O'Leary and Dave Conway-Jones of the IBM Emerging Technology Services group.
- What began as a proof-of-concept for visualizing and manipulating mappings between Message Queuing Telemetry Transport (MQTT) topics became a general and extensible tool.

Introduction to Node-RED

Figure 4-8. What is Node-RED

Node-RED is a Node.js-based run time at which you point a web browser to access the flow editor. Within the browser, you create your application by dragging nodes from your palette into a workspace and wire them together. With a single click, the application is deployed back to the run time and ran.

It is a way to build a fully functioning back end by using a drag style interface and minimum coding experience.

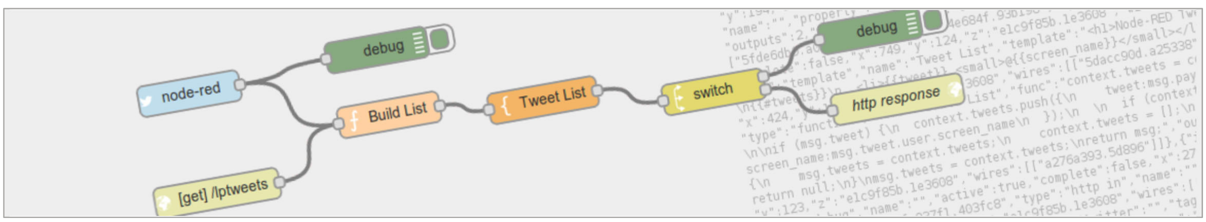
The palette of nodes can be extended by installing new nodes that are created by the Node-RED community. The flows that you create can be easily shared as JSON files.

Reference:
https://nodered.org/about/

IBM Training

Node-RED features

- Browser-based flow editing:
  - Node-RED provides a browser-based editor that makes it easy to wire together flows by using the wide range of nodes in the palette.
  - JavaScript functions can be created within the editor by using a rich text editor.
  - A built-in library saves useful functions, templates, or flows for reuse.
- Built on Node.js:
  - Taking full advantage of its event-driven and non-blocking model.
  - It is easy to extend the range of palette nodes to add new capabilities.
- Social development:
  - The flows that are created in Node-RED are stored by using JSON files, which can be easily imported and exported for sharing with others.



Introduction to Node-RED

Figure 4-9. Node-RED features

Building Node-RED on Node.js makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi and in the cloud.

Node-RED can be deployed on local environments on devices like Raspberry Pi and on the cloud like IBM Cloud.

4.3. Node-RED flow editor

IBM Training

Node-RED flow editor

Introduction to Node-RED

Figure 4-10. Node-RED flow editor

IBM Training

Topics

- Introduction to flow-based programming
- Node-RED overview
- Node-RED flow editor
- Node-RED Palette Manager

Introduction to Node-RED

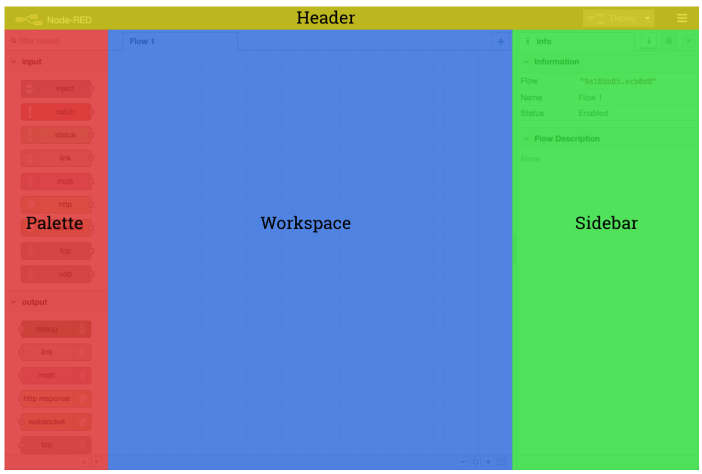
Figure 4-11. Topics

IBM Training

Node-RED flow editor window components

The editor window consists of four components:

- The *header* at the top, which contains the deployment button, main menu, and, if user authentication is enabled, the user menu.
- The *palette* on the left, which contains the nodes that are available for use.
- The *main workspace* in the middle, where flows are created.
- The *sidebar* on the right.



Introduction to Node-RED

Figure 4-12. Node-RED flow editor window components

The editor window consists of the following components:

- The *header*, which contains the deployment button to save changes, the main menu, and if user authentication is enabled, the user menu.
- The *palette*, which contains nodes that are used to make a flow. The palette has three types of nodes: inputs, outputs, and functions.
- The *main workspace*, which contains the flow that is used.
- The *sidebar*, which contains the information tab to view information about the nodes and to access the help menu, and the debug tab to view messages from the nodes.

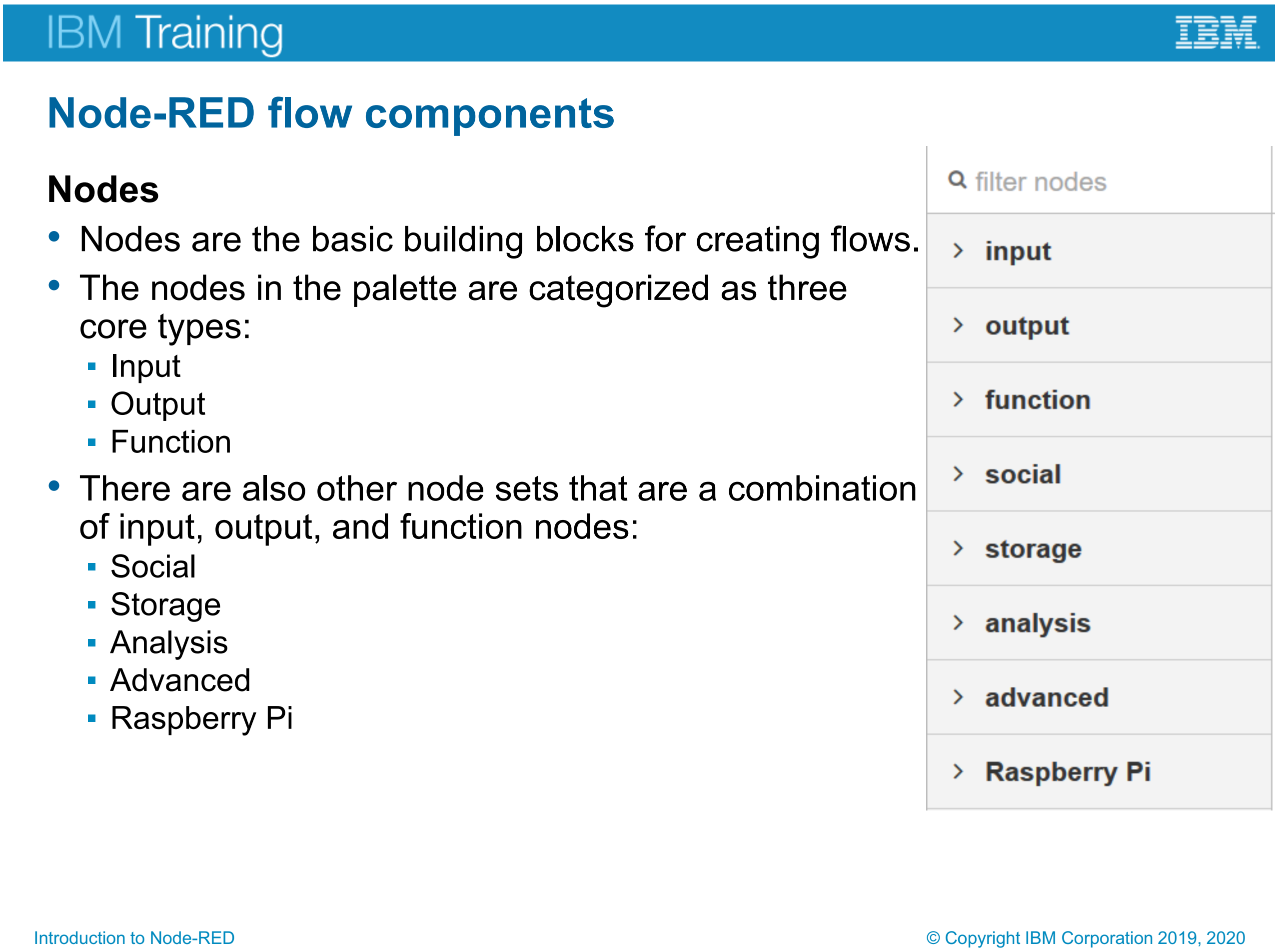


Figure 4-13. Node-RED flow components

The nodes in Node-RED can be added to the workspace by using any of the following methods:

- Dragging them from the palette.
- Using the quick-add dialog.
- Import them from the library or clipboard.

The nodes in the palette are categorized according to their type:

- An *input node* is an inject node that you can use to trigger manually a flow by clicking the node's button within the editor or trigger automatically a flow at regular intervals.
- An *output node* is a debug node that you can use to display messages in the Debug sidebar within the editor.
- A *function* is a node that you can use to run JavaScript code against the messages that are passed through it.

There are other node sets that are a combination of input, output, and function nodes:

- *Social nodes* can be used to support interaction with email and with Twitter. They enable flows to send or receive email or tweets.
- *Storage nodes* are the default node set because it is targeted at devices such as the Raspberry Pi, are limited, and focus on file-based storage.
- *Analysis nodes* can be used as segmentation nodes to try to determine the sentiment of an incoming message based on the words that are used in the message, for example, an email or tweet.
- *Advanced nodes* can be used as executable nodes to run a system command and return its output.
- Raspberry Pi.

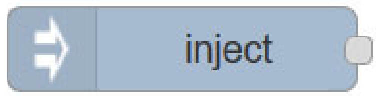
Reference:

http://noderedguide.com/node-red-lecture-4-a-tour-of-the-core-nodes/

IBM Training

Input nodes

- Input nodes are nodes that are used to start the flow.
- There are seven basic input nodes that are installed by default.
- You must include at least one input node in your message flow.
- Examples:
  - The *inject node* injects a timestamp or user-configured text into a message. It can be configured to inject manually, at a set interval, or at specific times.



  - The *mqtt node* subscribes to an MQTT broker and listens on a topic. It returns any data that is published on the topic as a new message.

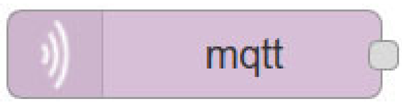


Figure 4-14. Input nodes

Input nodes cover the basic communications mechanisms that IoT applications are likely to use. These apps range from lower-level internet protocols such as UDP and TCP through to the higher-level HTTP and the publish and subscribe MQTT.

Inject nodes are input nodes that are used to trigger a flow, which you can do by clicking the node's button within the editor or setting up a configuration to trigger automatically flows at regular intervals. These nodes include a button on their left edge so that you can have some interaction with the node from within the editor.

Node-Red provides both an *MQTT* subscribe (input) and publish (output) node. The configuration for these nodes is almost identical as the main part of the configuration concerns the actual client connection.

References:

- http://noderedguide.com/
- http://www.steves-internet-guide.com/configuring-the-mqtt-publish-node/

IBM Training

Output nodes

- The output nodes are essentially the mirror images of the basic set of input nodes and provide a way to send data on the same set of protocols.
- You must include at least one output node to propagate messages to subsequent nodes that you connect to the subflow.
- For example, a *debug node* provides a simple way to view messages, which are displayed in the debug pane.

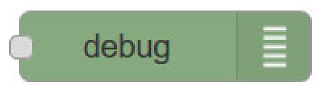


Figure 4-15. Output nodes

Output nodes are the endpoints of the flow. They are used to send data to an external service or pin on a device, and they may generate response messages.

Reference:
http://noderedguide.com/

IBM Training

Function nodes

- The function category contains various nodes that perform specific processing functions.
- The functions range from the simple delay and switch nodes to the programmable function node that can be adapted to almost any programming need.
- Examples:
  - The *function node* allows JavaScript code to be run against the messages that are passed through it.

  - The *switch node* routes messages based on their properties.

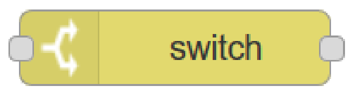


Figure 4-16. Function nodes

Function nodes are used when there is no existing node that is dedicated to your task.

A function node exposes a single JavaScript function. Using the function node, you can write your own JavaScript code that runs against the messages that are passed in, which returns zero or more messages to downstream nodes for processing.

Reference:
http://noderedguide.com/

IBM Training

Node-RED flow components (cont.)

Wires

- *Wires* define the connections between node input and output endpoints in a flow.
- Nodes are wired together by pressing the left mouse button on a node's port, dragging it to the destination node, and releasing the mouse button.

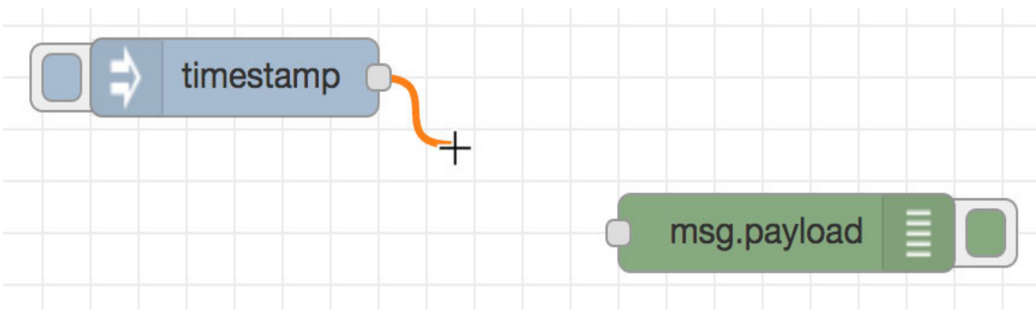


Figure 4-17. Node-RED flow components (cont.)

A Node-RED flow works by passing messages between nodes. This function is done by wiring nodes together.

Wires connect the output endpoints of nodes to inputs nodes, which indicate that messages that are generated by one node should be processed by the connected node next.

Messages usually have a *payload* property, which is the default property that most nodes work with to view data.

Node-RED also adds a property that is called *_msgid*, which is an identifier for the message that can be used to trace its progress through a flow.

Reference:
http://noderedguide.com/

IBM Training

How Node-RED helps in IoT applications

- IBM provides Node-RED nodes to help you connect your devices, gateways, and applications to Watson IoT Platform and create IoT solutions quickly.
- Example nodes:
  - Watson IoT Node

  - IBM IoT App Node

Introduction to Node-RED

Figure 4-18. How Node-RED helps in IoT applications

Watson IoT Node is a pair of nodes for connecting your devices or gateways to the Watson IoT Platform. Devices or gateways can use these nodes to send events and to receive commands from the application.

- The input node receives device commands from the IBM Watson Internet of Things Platform (Watson IoT Platform).
- The output node sends device events to the Watson IoT Platform.

IBM IoT App Node is a pair of nodes for connecting your applications to Watson IoT Platform. Applications can use the nodes to receive device events and send commands back to the device.

- The input node receives device events.
- The output node sends device commands and device events on the behalf of a device.

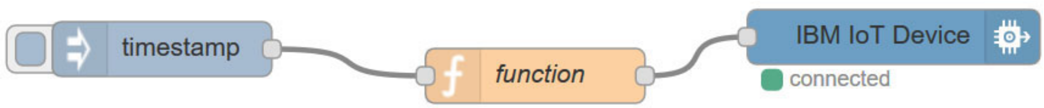
References:

- https://console.bluemix.net/docs/services/IoT/applications/dev_nodered.html#dev_nodered
- https://www.npmjs.com/package/node-red-contrib-ibm-watson-iot
- https://flows.nodered.org/node/node-red-contrib-scx-ibmiotapp

IBM Training

Node-RED flow example

- Device connecting part



- Watson IoT Platform connecting part

Introduction to Node-RED

Figure 4-19. Node-RED flow example

This example simulates a temperature device by using the Watson IoT Node, which sends the temperature readings to Watson IoT Platform every second. An application subscribes to these temperature events and triggers a shutdown command when the temperature crosses 30 degrees Celsius.

The device connecting part includes the following nodes:

- The inject node from the input, which is configured with an interval time of 1 second.
- The function node generates random numbers (as a temperature simulator; in a production environment, the nodes read from sensors).
- The Watson IoT Platform output node sends the temperature to Watson IoT Platform.

The Watson IoT Platform connecting part includes the following nodes:

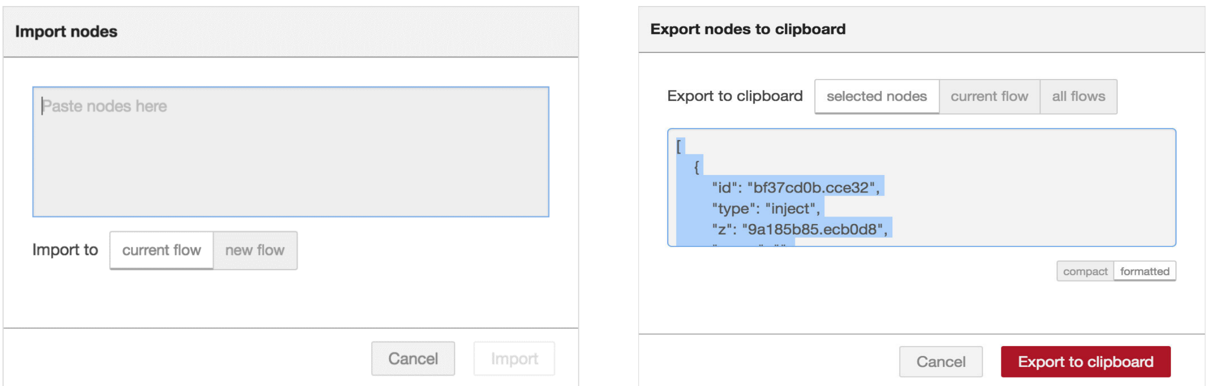
- The ibmiot input node gets the reading.
- The function node extracts the temperature value from the payload.
- The debug node prints the extracted temperature value in the debug pane.

Reference:
https://developer.ibm.com/recipes/tutorials/getting-started-with-watson-iot-platform-using-node-red/

IBM Training

Importing and exporting flows

Flows can be imported and exported from the editor by using the JSON format.



Introduction to Node-RED

Figure 4-20. Importing and exporting flows

Importing flows:

- To import a flow, open the Import dialog, paste the JSON flow, and click **Import**.
- The Import button is active only if a valid JSON flow is pasted into the dialog.
- You may import the nodes into the current flow or create a flow for them.

Exporting flows:

- Use the Export dialog to copy the JSON flow from the editor.
- You can export either the selected nodes, the current flow (including its tab node), or the complete flow configuration.
- You can export compact or formatted JSON. The compact option generates a single line of JSON with no white space. The formatted JSON option is formatted over multiple lines with full indentation, which can be easier to read.

4.4. Node-RED Palette Manager

IBM Training

Node-RED Palette Manager

Introduction to Node-RED

Figure 4-21. Node-RED Palette Manager

IBM Training

Topics

- Introduction to flow-based programming
- Node-RED overview
- Node-RED flow editor
- Node-RED Palette Manager

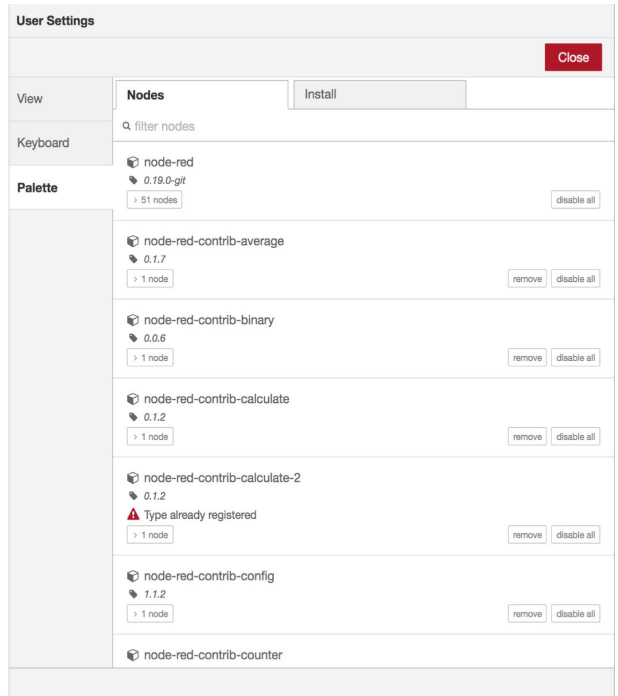
Introduction to Node-RED

Figure 4-22. Topics

IBM Training

Palette Manager

- Because not all nodes can be found in the palette, you can use the Palette Manager to install new nodes into the palette.
- You can access it under the Palette tab of the User Settings dialog.
- The Palette Manager has two tabs:
  - Nodes
  - Install



Introduction to Node-RED

Figure 4-23. Palette Manager

The Palette Manager has two tabs:

- The Nodes tab shows the modules that are installed in the run time:
  - Each entry in the Nodes tab shows the name and version of the module along with a list of the individual node types that the module provides.
  - There are options to remove, disable, or upgrade each module. If a node is in use within the flow, it cannot be removed or disabled.
- The Install tab shows the available modules that can be installed:
  - You can use the Install tab to search for available modules and install them.
  - To search for a module, enter its name in the search bar. The search results show the details of the modules, including when it was last updated and a link to its documentation. It can be installed by clicking **Install**.

IBM Training

Unit summary

- Explain the concept of flow-based programming.
- Explain what Node-RED is.
- Describe the Node-RED workspace and its components.
- Describe the Node-RED palette and the Palette Manager and explain how to use them to build flows.

Introduction to Node-RED

Figure 4-24. Unit summary

IBM Training

Review questions

1. True or False: Node-RED can be deployed to cloud environments only.

2. With the Node-RED flow editor, you can (select all that applies):
   a. Share your code with others as JSON files.
   b. Build applications faster.
   c. Encrypt your source code.
   d. All the above.

3. What are the three core types of the nodes that are found in the palette?
   a. Input, Social, and Output.
   b. Input, Watson, and Output.
   c. Input, Output, and Storage.
   d. Input, Output, and Function.

Introduction to Node-RED

Figure 4-25. Review questions

IBM Training

Review questions (cont.)

4. True or False: If a node is correctly configured, it displays a red triangle.

5. With the Import option, you can:
   - A. Export the current flow.
   - B. Export compact or formatted JSON.
   - C. Import only valid JSON that is pasted into the dialog.
   - D. None of the above.

Introduction to Node-RED

Figure 4-26. Review questions (cont.)

IBM Training

Review answers

1. True or False: Node-RED can be deployed to cloud environments only.

2. With the Node-RED flow editor, you can (select all that applies):
   A. Share your code with others as JSON files.
   B. Build applications faster.
   C. Encrypt your source code.
   D. All the above.

3. What are the three core types of the nodes that are found in the palette?
   A. Input, Social, and Output.
   B. Input, Watson, and Output.
   C. Input, Output, and Storage.
   D. Input, Output, and Function.

Introduction to Node-RED

Figure 4-27. Review answers

IBM Training

Review answers (cont.)

4. True or False: If a node is correctly configured, it displays a red triangle.

5. With the Import option, you can:
   A. Export the current flow.
   B. Export compact or formatted JSON.
   C. Import only valid JSON that is pasted into the dialog.
   D. None of the above.

Figure 4-28. Review answers (cont.)

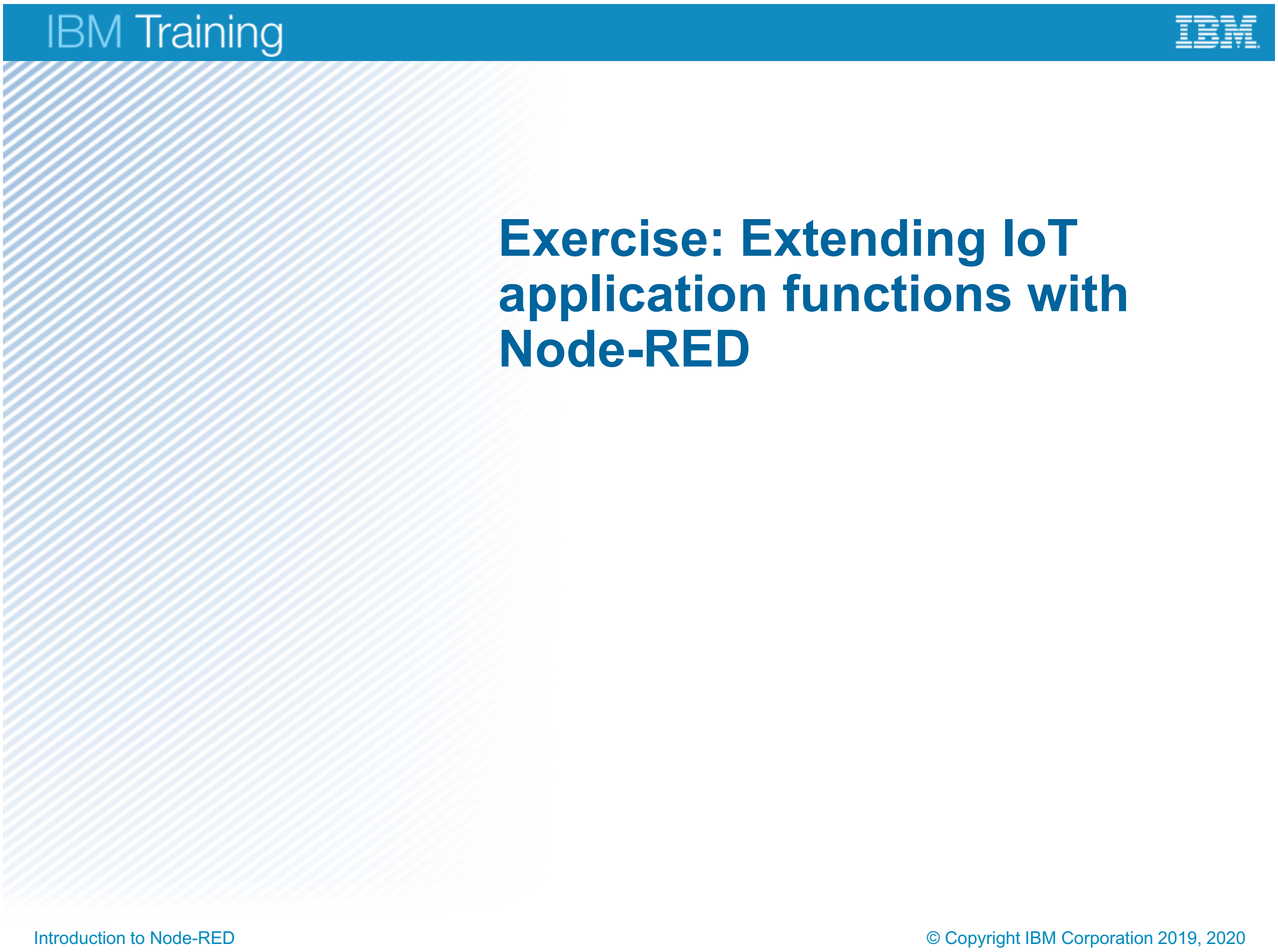


Figure 4-29. Exercise: Extending IoT application functions with Node-RED

IBM Training

Exercise objectives

- In this exercise, you set up a process to use the Node-RED application and the Watson Assistant service.
- After completing this exercise, you should be able to:
  - Connect your Watson Assistant service to your Node-Red application
  - Configure Watson Assistant nodes in your Node-Red flow.
  - Deploy the Node-RED flow and send user input to the assistant through the Node-RED flow.

Introduction to Node-RED

Figure 4-30. Exercise objectives

Unit 5. IBM Watson IoT Platform network protocols

Estimated time

01:00

Overview

This unit describes the common protocols that can be used to connect devices to Watson IoT Platform: MQTT and HTTP. This unit explains the difference between the MQTT publish and subscribe model and the Request/Response model in HTTP.

IBM Training

Unit objectives

- Describe the Hypertext Transfer Protocol (HTTP) protocol.
- Explain the concept of the publish and subscribe model.
- Describe the Message Queuing Telemetry Transport (MQTT) protocol.
- Compare the MQTT and HTTP protocols.

Figure 5-1. Unit objectives

5.1. HTTP protocol overview

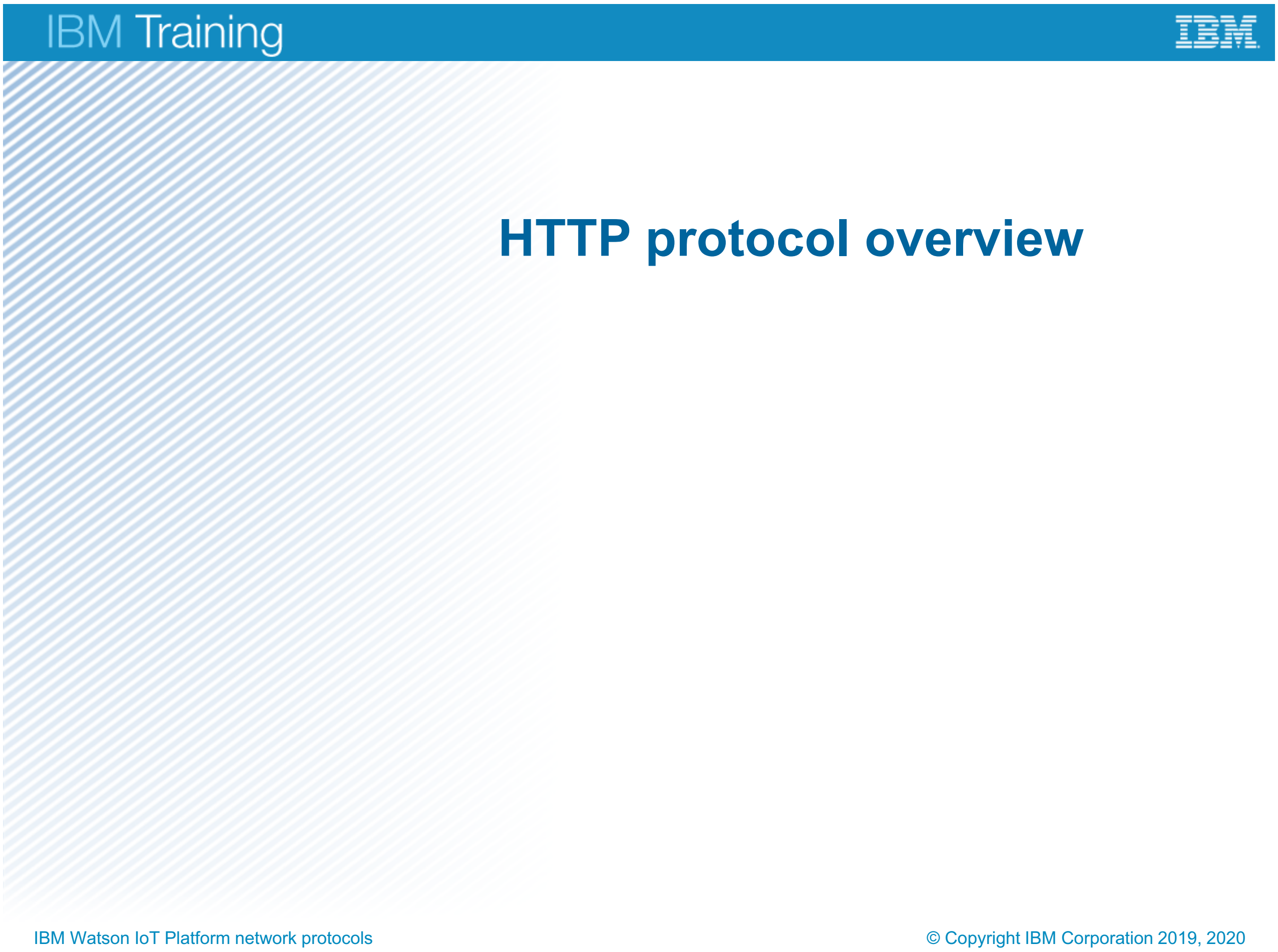


Figure 5-2. HTTP protocol overview

IBM Training

Topics

- HTTP protocol overview
- MQTT protocol overview
- HTTP and MQTT protocol comparison

IBM Watson IoT Platform network protocols

Figure 5-3. Topics

IBM Training

HTTP protocol overview

- HTTP definition
- Client/server model
- HTTP Request and Response

IBM Watson IoT Platform network protocols

Figure 5-4. HTTP protocol overview

IBM Training

HTTP definition

- The Hypertext Transfer Protocol (HTTP) is an *application-level* protocol for distributed and collaborative hypermedia information systems.
- It is a generic and *stateless* protocol that can be used for many tasks beyond hypertext, such as name servers and distributed object management systems by extending its *request methods*, *error codes*, and *headers*.

IBM Watson IoT Platform network protocols

Figure 5-5. HTTP definition

The HTTP protocol has the following components:

- Methods (verbs): **GET**, **POST**, **PUT**, **DELETE**, and **PATCH**
- Error codes (status codes):
  - 1XX: Information
  - 2XX: Success
  - 3XX: Redirection
  - 4XX: Client Error
  - 5XX: Server Error

- Headers

  Headers are the components for the header sections for both requests and responses. A header includes a name and values.

  Here are two example headers:

  - Content-Type: Describes the media type of the body of the **POST** and **PUT** methods.
  - Content-Length: Describes the length of the response body in octets.

  Developers can define their own headers.

Reference:

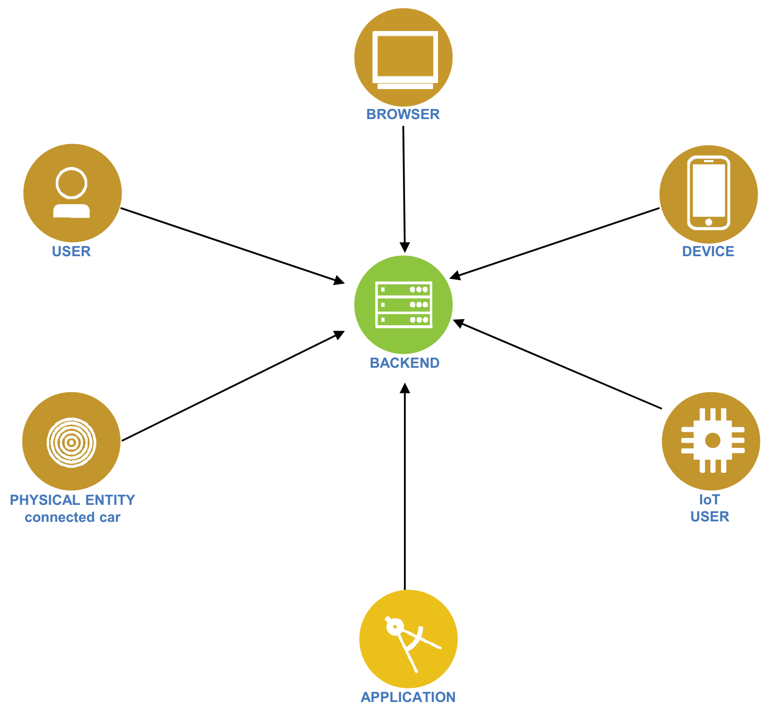
https://tools.ietf.org/html/rfc2616

IBM Training

Client/server model

The term *client/server* describes a type of distributed processing in which an application is divided into two parts.

Each part might be on separate operating systems, but they work together to provide a service to the user.



IBM Watson IoT Platform network protocols

Figure 5-6. Client/server model

In general, a back-end service can be databases or applications that are developed with any language (such as C, Java, Python, C++, or JavaScript). The client can be a user, physical entity (connected sensor), mobile device, or another application that uses a common programming language.

Microservices, also known as the microservice architecture, is an architectural style that structures an application as a collection of services.

An application that calls another application is the basic principle of *microservices*, which is an architecture pattern that you can use to build a distributed loosely coupled network of services that communicate with each other by using the HTTP protocol.

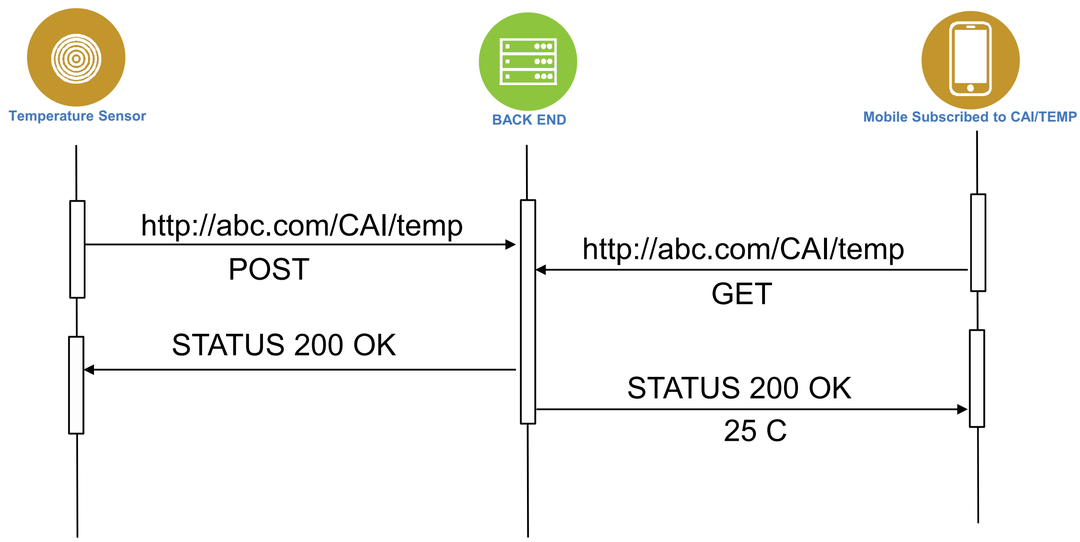
Reference:

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.ieak500/ieak511.htm

IBM Training

HTTP Request and Response

- HTTP Request and Response provide a simplified view of the TCP/IP protocols.
- Here is an example of how the seven layers model can simplify and hide the complexity of the underlying layers.



IBM Watson IoT Platform network protocols

Figure 5-7. HTTP Request and Response

Here are some important highlights regarding HTTP/1.1:

- The client always initiates the connection. If the client wants to retrieve or post something from the backend, the client still must initiate the connection.
- After the server responds with any status code, the connection is terminated.

Questions to be discussed:

- What if the server wants to send the user a message, such as “Turn off the device”?
- What if you want to keep the connection open for a long period to stream data, such as car trips?
- What if the user or device wants to communication with another user or device?

5.2. MQTT protocol overview

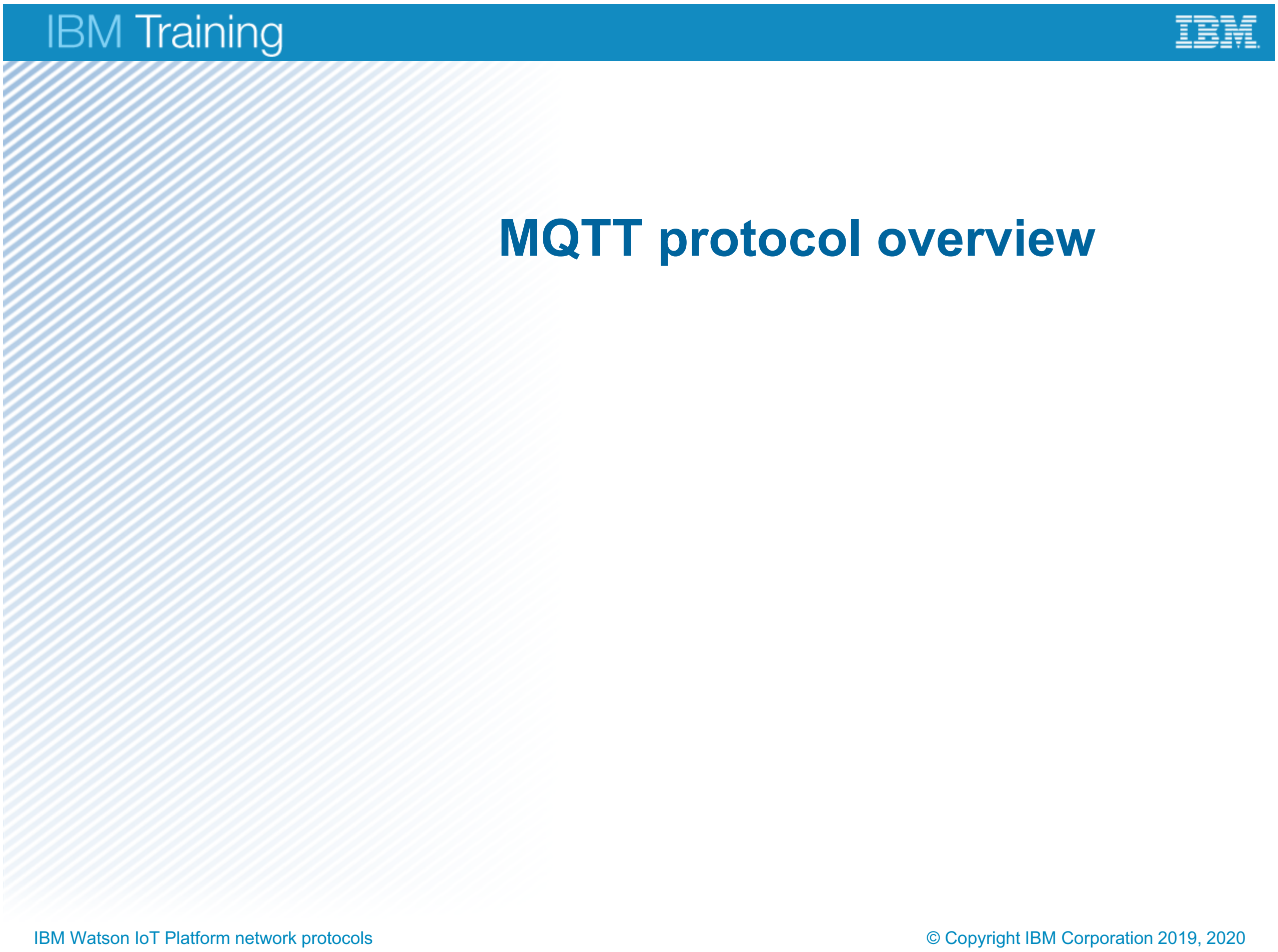


Figure 5-8. MQTT protocol overview

IBM Training

Topics

- HTTP protocol overview
- MQTT protocol overview
- HTTP and MQTT protocol comparison

IBM Watson IoT Platform network protocols

Figure 5-9. Topics

IBM Training

MQTT protocol overview

- MQTT definition
- MQTT components
- Publish and subscribe pattern
- MQTT quality of service (QoS)

IBM Watson IoT Platform network protocols

Figure 5-10. MQTT protocol overview

IBM Training

MQTT definition

- MQTT is a machine-to-machine (M2M) IoT connectivity protocol.
- It is a lightweight publish and subscribe messaging transport protocol.
- It is useful for connections to remote locations where a small code footprint is required, or network bandwidth is at a premium.

IBM Watson IoT Platform network protocols

Figure 5-11. MQTT definition

According to https://www.mqtt.org, Andy Stanford-Clark of IBM and Arlen Nipper of Cirrus Link authored the first version of the MQTT protocol in 1999. In 2013, IBM submitted MQTT V3.1 to OASIS.

MQTT V3.1 is the most used version with the TCP/IP protocol. Version 3.1.1, also know as MQTT-SN, supports networks that do not use TCP/IP, such as Zigbee. The core difference between MQTT V3.1 and MQTT-SN V3.1.1 is that MQTT V3.1 depends on TCP and MQTT-SN depends on UDP or Bluetooth.

In this course, the focus is on MQTT V3.1.

IBM Training

MQTT components

- *Topic* is the place to or from which a device wants to put or retrieve a message.

- *Broker* is the server that handles the data transmission between the clients.

IBM Watson IoT Platform network protocols

Figure 5-12. MQTT components

MQTT components:

- In MQTT, the word *topic* refers to an UTF-8 string that the broker uses to filter messages for each connected client. The topic consists of one or more topic levels. Each topic level is separated by a forward slash (topic level separator).

  The client can subscribe to the exact topic or use a wildcard. For example, subscription to house/+/temperature results in all message being sent to the topic house/living-room/temperature and any topic with an arbitrary value in the place of living room, for example, house/kitchen/temperature.

  The plus sign is a single-level wildcard and allows arbitrary values for only one hierarchy. If you need to subscribe to more than one level, for example, to the entire subtree, there is also a multilevel wildcard (#). You can use it to subscribe to all underlying hierarchy levels. For example, house/# subscribes to all topics beginning with house.

- An MQTT *broker* is a server that receives all messages from the clients and then routes the messages to the appropriate destination clients.

  There are many brokers that implement the MQTT protocol. One of the most popular and commonly used is the Mosquito broker.

IBM Training

MQTT components (cont.)

- *Message* is the data that a device receives “when subscribing” from a topic or sends “when publishing” to a topic.
- *Publish* is the process that a device uses to send its message to the broker.
- *Subscribe* is the process that a device uses to retrieve a message from the broker.

IBM Watson IoT Platform network protocols

Figure 5-13. MQTT components (cont.)

In a *publish and subscribe system*, a device can publish a *message* on a topic, or it can be subscribed to a topic to receive messages.

For example, if Device1 publishes on a topic and Device 2 is subscribed to the same topic as device 1 is publishing in, Device 2 receives the message.

Reference:

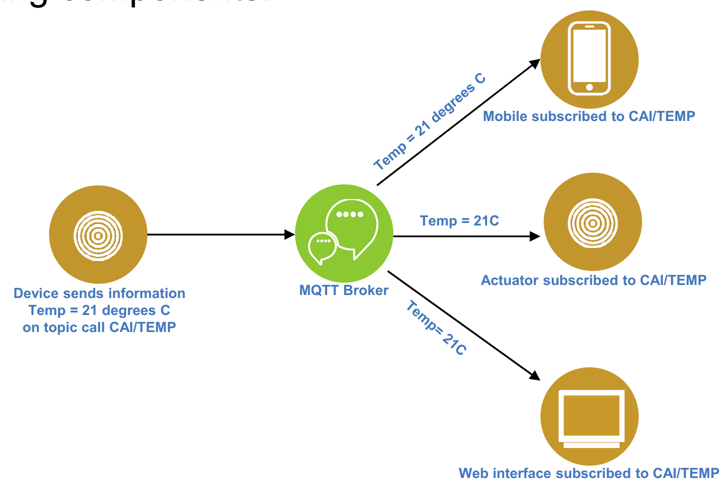
https://1sheeld.com/mqtt-protocol/

IBM Training

Publish and subscribe pattern

In general, the publish and subscribe pattern is composed of the following components:

- Publisher
- Subscriber
- Broker



IBM Watson IoT Platform network protocols

Figure 5-14. Publish and subscribe pattern

- The publish and subscribe pattern (also known as pub/sub) provides an alternative to traditional client/server architecture. In the client-sever model, a client communicates directly with an endpoint.
- The pub/sub model decouples the client that sends a message (the publisher) from the client or clients that receive the messages (the subscribers). The publishers and subscribers never contact each other directly. In fact, they are not even aware that the other exists. The connection between them is handled by a third component (the broker).
- The job of the broker is to filter all incoming messages and distribute them correctly to subscribers. So, let’s dive a little deeper into some of the general aspects of pub/sub.

This slide shows an example of the publish and subscribe pattern. The publisher sends the message to the broker, and all subscribers are connected to the same broker.

- The publisher (in this case, a temperature sensor) sends temperature information to the MQTT broker.

- There are three subscribers:
  - Mobile, which is controlled by an operation engineer.
  - An actuator that controls another machine.
  - A web dashboard that represents the control loop.
- When the sensor sends its information to a predefined topic (in this case, CAI/TEMP), all subscribers that are subscribed to this topic receive a copy of this message.

Note

Any device, depending on its privileges, can be a publisher or subscriber.

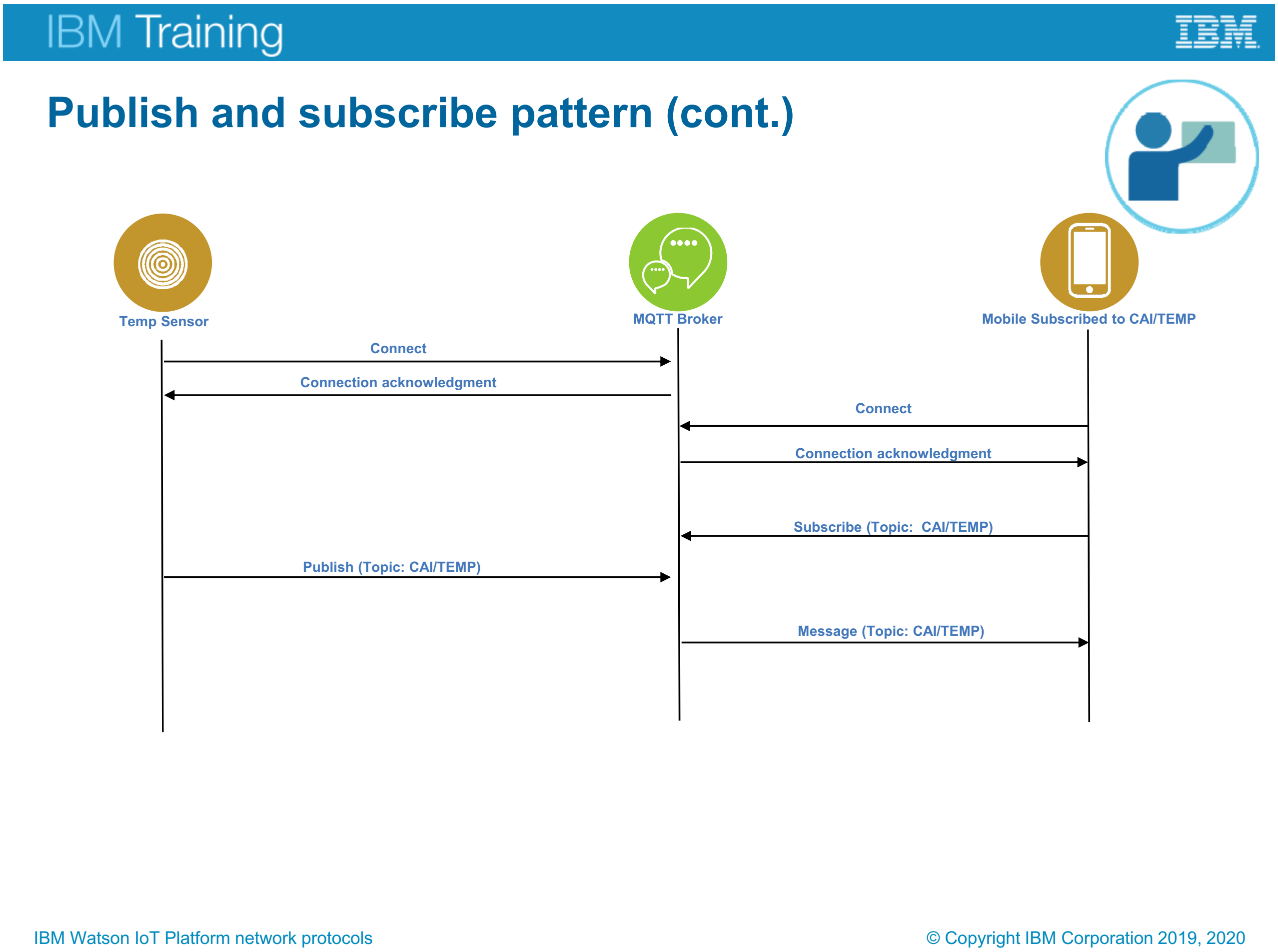


Figure 5-15. Publish and subscribe pattern (cont.)

This slide shows the following pattern:

1. The Temp Sensor connects to the MQTT broker.
2. The mobile device connects to the same MQTT broker.
3. The mobile device is subscribed to the topic CAI/TEMP, so when the Temp Sensor publishes any data on topic CAI/TEMP, the broker send the message to the mobile device.

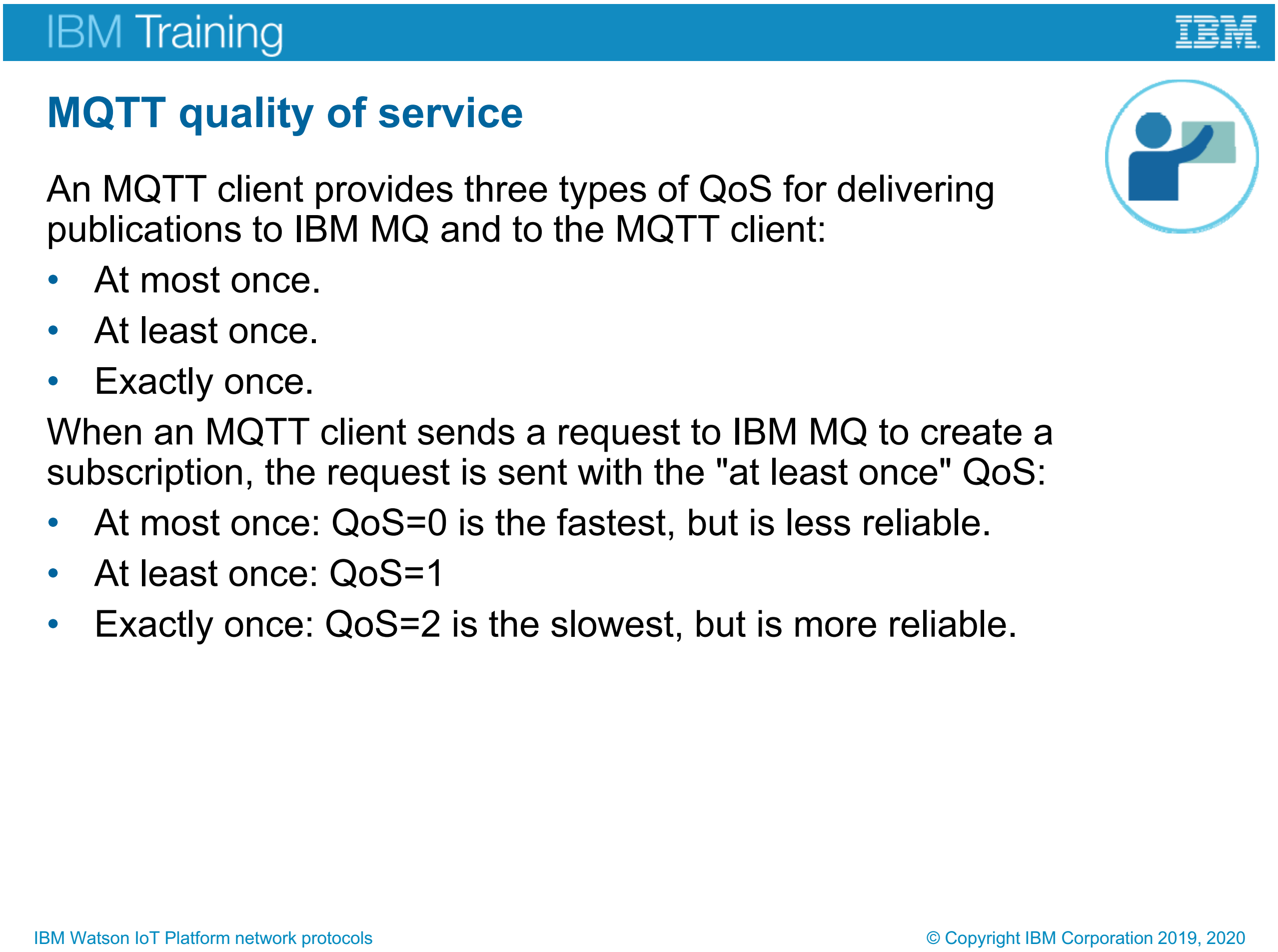


Figure 5-16. MQTT quality of service

QoS is a balance between speed and reliability. Here are some examples of QoS in action:

- A sensor must send a continuous stream of temperature readings every second. In this case, QoS=0 is likely the best option because in a worst-case scenario where you might lose a few seconds, this lapse can be accommodated.
- An actuator receives a command to turn off a car engine if it is stolen. In this case, QoS=1 is likely the best option because you want to ensure that the car engine is turned off, and if you turn off the car engine twice it should not cause an issue.
- You have a robot that works on hazardous environments, and you are sending tuning commands to control its arm. For such a mission-critical situation, QoS=2 is likely the best option because sending a command twice might cause an error.

Reference:

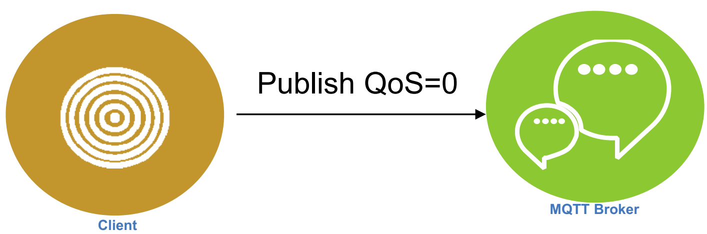
https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.0.0/com.ibm.mq.dev.doc/q029090_.htm

IBM Training

The three types of QoS

QoS=0

- This service level guarantees a best-effort delivery. There is *no* guarantee of delivery.
- The recipient does not acknowledge receipt of the message and the message is not stored and retransmitted by the sender.
- QoS0 is often called “fire and forget” and provides the same guarantee as the underlying TCP protocol.



IBM Watson IoT Platform network protocols

Figure 5-17. The three types of QoS

At most once (QoS=0):

- The message is delivered at most once or it is not delivered at all. Its delivery across the network is not acknowledged.
- The message is not stored. The message might be lost if the client is disconnected or the server fails.
- QoS=0 is the fastest mode of transfer. It is sometimes called "fire and forget".
- The MQTT protocol does not require servers to forward publications at QoS=0 to a client. If the client is disconnected when the server receives the publication, the publication might be discarded depending on the server. The telemetry (MQXR) service does not discard messages that are sent

Reference:

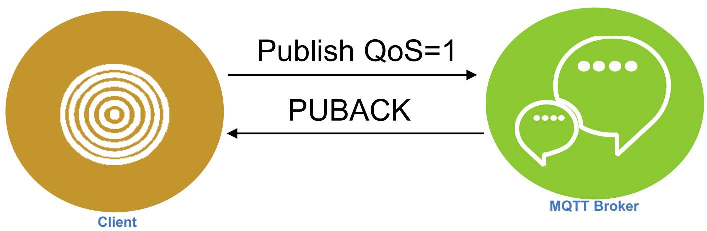
- https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.0.0/com.ibm.mq.dev.doc/q029090_.htm
- https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/

IBM Training

The three types of QoS (cont.)

QoS=1

- QoS=1 guarantees that a message is delivered at least one time to the receiver.
- The sender stores the message until it gets a PUBACK packet from the receiver that acknowledges receipt of the message.
- It is possible for a message to be sent or delivered multiple times.



IBM Watson IoT Platform network protocols

Figure 5-18. The three types of QoS (cont.)

At least once (QoS=1):

- QoS=1 is the default mode of transfer.
- The message is always delivered at least once. If the sender does not receive an acknowledgment, the message is sent again with the DUP flag set until an acknowledgment is received. As a result, the receiver can be sent the same message multiple times and might process it multiple times.
- The message must be stored locally at the sender and the receiver until it is processed.
- The message is deleted from the receiver after it processes the message. If the receiver is a broker, the message is published to its subscribers. If the receiver is a client, the message is delivered to the subscriber application. After the message is deleted, the receiver sends an acknowledgment to the sender.
- The message is deleted from the sender after it receives an acknowledgment from the receiver.

References:

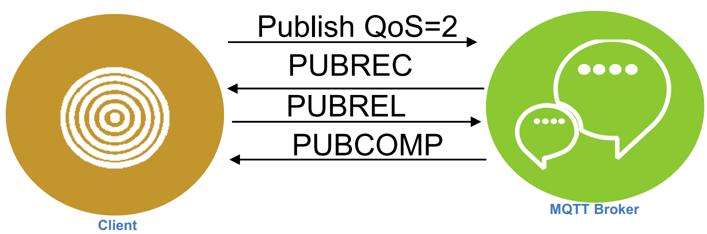
- https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.0.0/com.ibm.mq.dev.doc/q029090_.htm
- https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/

IBM Training

The three types of QoS (cont.)

QoS=2

- QoS=2 is the highest level of service in MQTT. This level guarantees that each message is received only once by the intended recipients.
- QoS 2 is the safest and slowest quality of service level. The guarantee is provided by at least two request/response flows (a four-part handshake) between the sender and the receiver.
- The sender and receiver use the packet identifier of the original PUBLISH message to coordinate delivery of the message.



IBM Watson IoT Platform network protocols

Figure 5-19. The three types of QoS (cont.)

Exactly once (QoS=2):

- The message is always delivered exactly once.
- The message must be stored locally at the sender and the receiver until it is processed.
- QoS=2 is the safest but slowest mode of transfer. It takes at least two pairs of transmissions between the sender and receiver before the message is deleted from the sender. The message can be processed at the receiver after the first transmission.
- In the first pair of transmissions, the sender transmits the message and gets acknowledgment from the receiver that it stored the message. If the sender does not receive an acknowledgment, the message is sent again with the DUP flag set until an acknowledgment is received.

- In the second pair of transmissions, the sender tells the receiver that it can complete processing the PUBREL message. If the sender does not receive an acknowledgment of the PUBREL message, the PUBREL message is sent again until an acknowledgment is received. The sender deletes the message that it saved when it receives the acknowledgment to the PUBREL message.
- The receiver can process the message in the first or second phases if it does not reprocess the message. If the receiver is a broker, it publishes the message to subscribers. If the receiver is a client, it delivers the message to the subscriber application. The receiver sends a completion message back to the sender that it finished processing the message.

References:

- https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_9.0.0/com.ibm.mq.dev.doc/q029090_.htm
- https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/

5.3. HTTP and MQTT protocol comparison

IBM Training

HTTP and MQTT protocol comparison

IBM Watson IoT Platform network protocols

Figure 5-20. HTTP and MQTT protocol comparison

IBM Training

Topics

- HTTP protocol overview
- MQTT protocol overview
- HTTP and MQTT protocol comparison

IBM Watson IoT Platform network protocols

Figure 5-21. Topics

IBM Training

HTTP and MQTT protocol comparison

Most protocols fall into either of the following two categories:

- Client/server, which is like HTTP.
- Publish and subscribe, which is like MQTT.

When you design an IoT system, you must consider the protocol that you will use:

- HTTP/1.1 is the most widespread protocol that is used for the web. It is limited for use with an IoT solution because it goes in one direction only, which is client to server. Also, the communication terminates when the request is fulfilled.
- MQTT 3.1 is designed for IoT and its unreliable communication. Also, the device always initiates the outbound connection, which is inherently secure.

IBM Watson IoT Platform network protocols

Figure 5-22. HTTP and MQTT protocol comparison

These protocols do not replace each other and neither one is better than the other. The protocol that you use depends on your situation and device support.

MQTT is broadly supported by many libraries and providers. Many open source and commercial brokers of MQTT are available, such as HiveMQ, Eclipse Mosquitto, IBM WebSphere MQ, and RabbitMQ. There is also a customized version of MQTT that is provided by Watson IoT.

Reference:

https://www.eclipse.org/paho/

IBM Training

Unit summary

- Describe the Hypertext Transfer Protocol (HTTP) protocol.
- Explain the concept of the publish and subscribe model.
- Describe the Message Queuing Telemetry Transport (MQTT) protocol.
- Compare the MQTT and HTTP protocols.

IBM Watson IoT Platform network protocols

Figure 5-23. Unit summary

IBM Training IBM

Review questions

1. True or false: Watson IoT Platform supports only MQTT.
2. Which of the following statements is correct about QoS=1:
   a. QoS=1 guarantees that a message is delivered at least one time to the receiver.
   b. QoS=1 is called “fire and forget”.
   c. QoS=1 is the highest level of service in MQTT.
   d. None of the above statements is correct.
3. True or false: HTTP is a stateful protocol.
4. True or false: MQTT has four levels of QoS.

IBM Watson IoT Platform network protocols

Figure 5-24. Review questions

IBM Training

Review questions (cont.)

5. What does “house/living-room/temperature” indicates:
   a. Broker name.
   b. Topic.
   c. Message header.
   d. Publisher name.

Figure 5-25. Review questions (cont.)

IBM Training

Review answers

1. True or False: Watson IoT Platform supports only MQTT.
   The answer is False.

2. Which of the following statements is correct about QoS=1:
   A. QoS=1 guarantees that a message is delivered at least one time to the receiver.
   B. QoS=1 is called “fire and forget”.
   C. QoS=1 is the highest level of service in MQTT.
   D. None of the above statements is correct.
   The answer is A.

3. True or False: HTTP is a stateful protocol.
   The answer is False.

4. True or False: MQTT has four levels of QoS.
   The answer is False. There are only three levels.

IBM Watson IoT Platform network protocols

Figure 5-26. Review answers

1. False. Watson IoT Platform supports both MQTT and HTTP.
2. A. QoS=1 guarantees that a message is delivered at least one time to the receiver.
3. False. HTTP is a stateless protocol that can be used for many tasks beyond hypertext.
4. False. There are only three levels: QoS=0, QoS=1, and QoS=2.

IBM Training

Review answers (cont.)

5. What does “house/living-room/temperature” indicate:
   a. Broker Name.
   b. Topic.
   c. Message header.
   d. Publisher name.

   The answer is B.

IBM Watson IoT Platform network protocols

Figure 5-27. Review answers (cont.)

5. B.

IBM Training

Exercise: Monitoring server CPU usage and reporting statistics

IBM Watson IoT Platform network protocols

Figure 5-28. Exercise: Monitoring server CPU usage and reporting statistics

IBM Training

Exercise objectives

- In this exercise, you implement an IoT use case about monitoring server CPU usage in a server room. You use your local workstation as a server by installing Node-RED on your PC or notebook. Your system reports CPU usage to a cloud application through the Watson IoT Platform.
- After completing this exercise, you should be able to:
  - Set up Node-RED on a local workstation.
  - Configure devices on Watson IoT Platform.
  - Connect a Node-RED application that is running locally to a cloud application (in this exercise, you use Node-RED on IBM Cloud to represent the cloud application).

IBM Watson IoT Platform network protocols

Figure 5-29. Exercise objectives

Unit 6. Enhancing Internet of Things solutions with REST APIs

Estimated time

02:30

Overview

This unit provides an overview of application programming interfaces (APIs), their value, and usage. This unit explains the representational state transfer (REST) API. It describes how to use Watson IoT Platform APIs. This unit also describes how to access Watson AI and Cloudant services on IBM Cloud by using REST APIs.

IBM Training

Unit objectives

- Explain the concept of the application programming interface (API) and describe the types of APIs.
- Explain the characteristics of REST APIs.
- List the Watson artificial intelligence (AI) services.
- Describe the IBM Cloudant database service on IBM Cloud.
- Describe how to access Watson and Cloudant services on IBM Cloud with REST APIs.

Enhancing Internet of Things solutions with REST APIs

Figure 6-1. Unit objectives

6.1. Application programming interface overview

IBM Training

Application programming interface overview

Enhancing Internet of Things solutions with REST APIs

Figure 6-2. Application programming interface overview

IBM Training

Topics

- Application programming interface overview
- Watson IoT Platform HTTP REST API
- Accessing Watson and Cloudant services with REST APIs
- Enhancing an IoT solution with Watson AI

Enhancing Internet of Things solutions with REST APIs

Figure 6-3. Topics

IBM Training

API definition

- An *application programming interface* (API) is a software intermediary that allows two applications to communicate. In other words, an API is the messenger that delivers your request to the provider that you are requesting it from and then delivers the response back to you.
- APIs are pervasive both in the mobile and web arenas. They are integral to almost every action in software. With just a few taps or clicks, they enable ordering a pizza, booking a hotel, rating a song, or downloading software. APIs work quietly in the background to make the interactivity that you expect and rely upon possible.

Enhancing Internet of Things solutions with REST APIs

Figure 6-4. API definition

Here is an example of how an API might work in everyday life:

You are searching for a hotel room from an online travel booking site. Using the site's online form, you select the city that you want to stay in, the check-in and check-out dates, number of guests, and number of rooms. Then, you click **Search**.

The travel site aggregates information from many different hotels. When you click **Search**, the site then interacts with each hotel's API, which delivers results for available rooms that meet your criteria. This process can all happen within seconds because of an API, which acts like a messenger that runs back and forth between applications, databases, and devices.

IBM Training

Web services APIs

- Representational State Transfer (REST)
- Simple Object Access Protocol (SOAP)

Enhancing Internet of Things solutions with REST APIs

Figure 6-5. Web services APIs

There are different web services APIs:

- *SOAP* is a protocol that defines the communication method and the structure of the messages. The data transfer format is XML.
- *REST* is a set of architectural principles. The most used data format is JSON or XML. Often, the service offers a choice, which means that the client can request one or the other data format by including "`json`" or "xml" in the URL path or in a URL parameter.

IBM Training

REST API definition

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

Enhancing Internet of Things solutions with REST APIs

Figure 6-6. REST API definition

REST is an architecture style for building resources on the web. Examples of resources for a website include 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 URI to describe the network location of the resource.

REST provides the following HTTP methods:

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

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

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

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

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

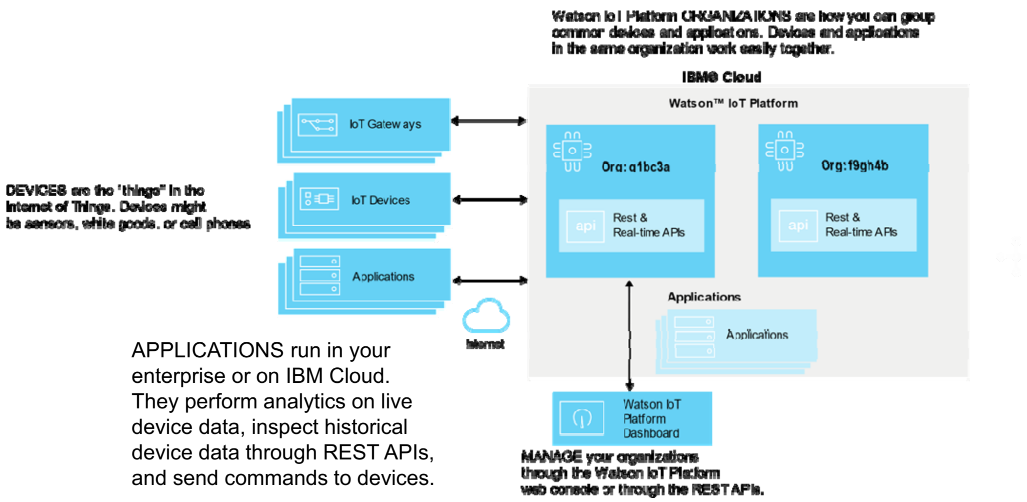
The **POST** method is used to post data to the server. In this case, the parameters are posted in the body of the request, not in the URI.

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

IBM Training

IoT solutions and APIs

- Normally, you use Watson IoT Platform for device registration. However, this process is not effective for a production IoT solution, such as automating factory devices, where the number of devices that must be registered is large.
- APIs make this process easier and faster.



Enhancing Internet of Things solutions with REST APIs

Figure 6-7. IoT solutions and APIs

Some devices do not support MQTT but support HTTP, which is one way of using HTTP APIs to connect devices on an IoT platform and perform automated device registration and data collection.

IBM Training

Popular API examples

- Google Maps API
- YouTube APIs
- Twitter APIs
- Amazon Product Advertising API

Figure 6-8. Popular API examples

Here are some popular API examples:

- Google Maps API: With Google Maps APIs, developers can embed Google Maps into web pages by using a JavaScript or Flash interface. The Google Maps API works on mobile devices and desktop browsers.
- YouTube APIs: With YouTube APIs, developers can integrate YouTube videos and functions into websites or applications. YouTube APIs include the YouTube Analytics API, YouTube Data API, YouTube Live Streaming API, YouTube Player APIs, and others.
- Twitter APIs: Twitter offers two APIs:
  - With the REST API, developers can access core Twitter data.
  - With the Search API, developers can interact with Twitter Search and trends data.
- Amazon Product Advertising API: With the Amazon Product Advertising API, developers can access Amazon's product selection and discovery functions to advertise Amazon products to monetize a website.

6.2. Watson IoT Platform HTTP REST API

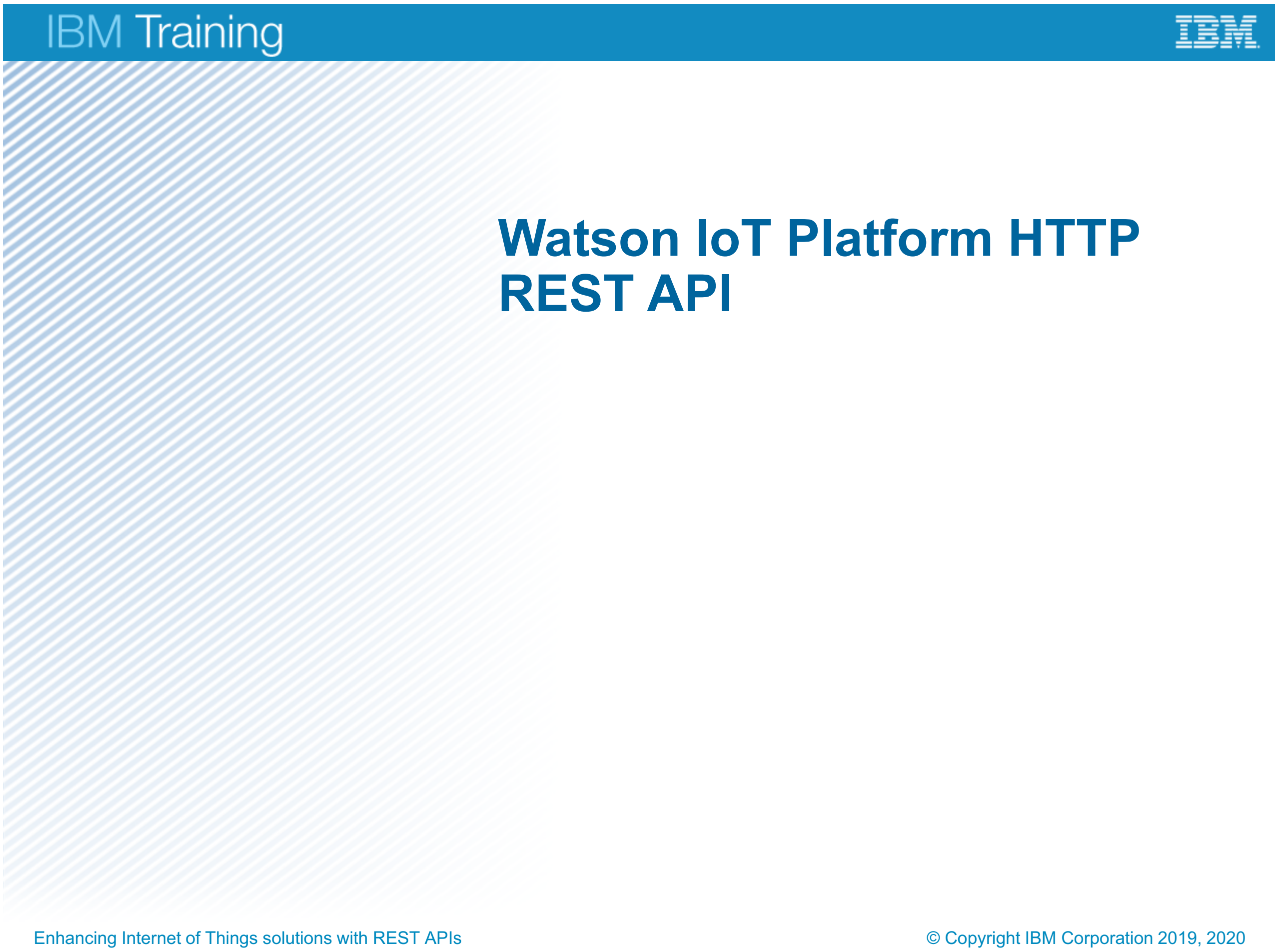


Figure 6-9. Watson IoT Platform HTTP REST API

IBM Training

Topics

- Application programming interface overview
- Watson IoT Platform HTTP REST API
- Accessing Watson and Cloudant services with REST APIs
- Enhancing an IoT solution with Watson AI

Enhancing Internet of Things solutions with REST APIs

Figure 6-10. Topics

IBM Training

IBM Watson IoT Platform REST APIs

- The IBM Watson IoT Platform REST API simplifies access to IoT devices and data that allows developers to build analytic applications, dashboards, and IoT apps for mobile platforms.
- It makes it easier for developers to manipulate different operations for device management when accessing and storing device data in addition to connecting gateway devices.

Enhancing Internet of Things solutions with REST APIs

Figure 6-11. IBM Watson IoT Platform REST APIs

Watson IoT Platform provides APIs for developing code for devices, gateways, and applications that connect to Watson IoT Platform. The APIs allow the publication of events and commands by applications and devices indistinguishable from events and commands that are published by using MQTT.

IBM Training

IBM Watson IoT Platform REST APIs (cont.)

Title	Description	
Organization Administration	The Organization Administration APIs are used to configure an organization and include creating and deleting devices, checking usage, service status, and diagnosing device connection problems.	View APIs
Device Management	The Device Management APIs are used for interacting with managed devices by using the device management protocol.	View APIs
Information Management	The Information Management APIs are used to access device event data and to get and update device location and obtain weather information for that location.	View APIs
State Management	The State Management APIs are used to help you organize and integrate data coming in to and going out of the IBM® Watson™ IoT Platform.	View APIs
Action Manager	The Action Manager APIs are used to configure actions that can run business logic when they are triggered by an event that occurred within the Platform.	View APIs
Historian Connector	The Historian Connector APIs are used to find and configure compatible services that are used to store your IoT device data.	View APIs
Risk Management	The Risk Management APIs are used for managing policies in order to enhance connection security and control access to the server from devices.	View APIs
Security	The Security APIs are used to manage the authentication and authorization of users, API keys, and devices.	View APIs
HTTP Messaging	The HTTP Messaging APIs are used for the publication of events and commands by applications and devices.	View APIs
Client Connection State	The Client Connection State APIs are used to allow users to retrieve and query the client connection state for clients that have connected to the IBM® Watson™ IoT Platform.	View APIs

Enhancing Internet of Things solutions with REST APIs

Figure 6-12. IBM Watson IoT Platform REST APIs (cont.)

This slide shows the landing page of the available APIs.

The HTTP APIs are protected by HTTP basic authentication. When you generate an API key by using the dashboard, you are given a key and an authentication token.

To create an API key and authentication token pair, complete the following steps:

1. In the Platform Service dashboard, select **Apps > API Keys**.
2. Click **Generate API Key**. Important: Note the API key and token pair. Authentication tokens are unrecoverable. If you lose or forget this token, you must reregister the API key to generate a new authentication token.
   - An example of an API key is a-organization_id-a84ps90Ajs.
   - An example of a token is MP$08VKz!8rXwnR-Q*.
3. Add a comment to identify the API key in the dashboard, for example: "Key to connect my application".
4. Click **Finish**.

Reference:

https://cloud.ibm.com/docs/services/IoT/reference?topic=iot-platform-api_overview#api_http

IBM Training

Watson IoT Platform HTTP REST API capabilities

The Watson IoT Platform HTTP REST API supports the following capabilities and functions for applications:

- Organization information retrieval
- Bulk device operations (list, add, and remove)
- Usage management operations (retrieve the total amount of data that is used)
- Device event publishing (HTTP messaging APIs)
- Service status querying (retrieve service statuses by organization)

Figure 6-13. Watson IoT Platform HTTP REST API capabilities

This slide describes the capabilities of Watson IoT REST APIs that you can use to connect devices and communicate with the whole system.

There are two types of HTTP APIs that can be used with Watson IoT Platform:

- You can use HTTP REST APIs to create, update, list, and delete devices.
- You can use HTTP Messaging APIs to send event information from devices to the cloud and accept commands from applications in the cloud.

Reference:

https://cloud.ibm.com/docs/services/IoT/devices?topic=iot-platform-api_overview#api

IBM Training

Watson IoT Platform HTTP REST API capabilities (cont.)

- Device type operations (list, create, delete, view details, and update)
- Device operations (list, add, remove, view details, update, view location, and view management information)
- Device diagnostic operations (clear logs, retrieve logs, add log information, delete logs, get specific logs, clear error codes, get device error codes, and add error codes)
- Connection problem determination (list device connection log events)
- Last event cache (view the last event for a specific device)
- Device management request operations (list device management requests, initiate requests, clear request status, get details of a request, list all request statuses by device, and get the request status for a specific device)

Enhancing Internet of Things solutions with REST APIs

Figure 6-14. Watson IoT Platform HTTP REST API capabilities (cont.)

This slide describes further capabilities of Watson IoT REST APIs that you can use to connect devices and communicate with the whole system.

6.3. Accessing Watson and Cloudant services with REST APIs

IBM Training

Accessing Watson and Cloudant services with REST APIs

Enhancing Internet of Things solutions with REST APIs

Figure 6-15. Accessing Watson and Cloudant services with REST APIs

IBM Training

Topics

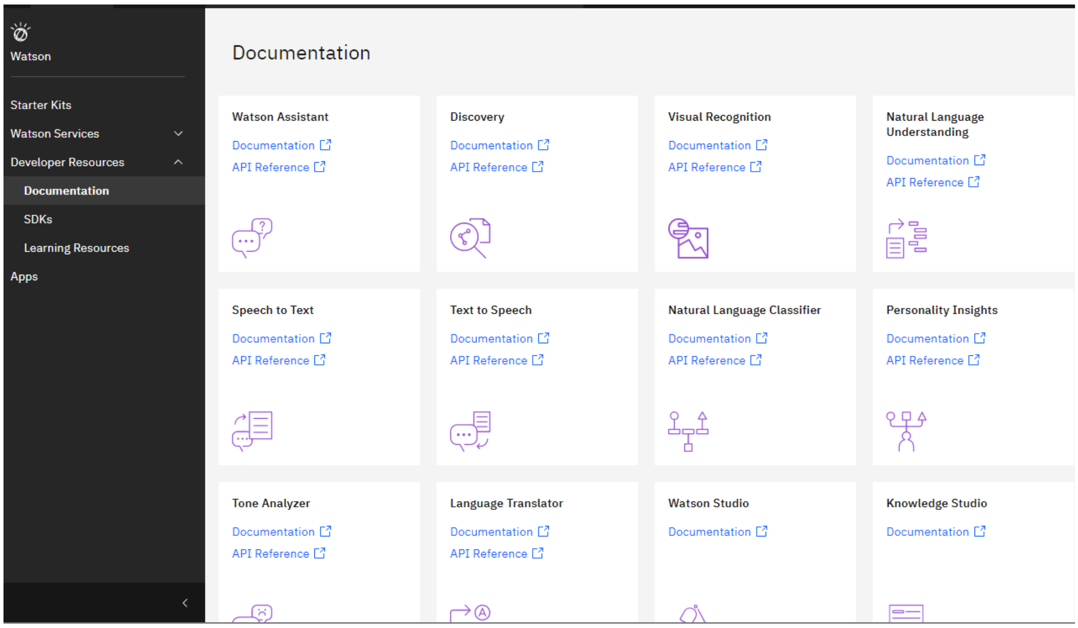
- Application programming interface overview
- Watson IoT Platform HTTP REST API
- Accessing Watson and Cloudant services with REST APIs
- Enhancing an IoT solution with Watson AI

Figure 6-16. Topics

IBM Training

Watson Documentation

Watson Documentation documents services and API references that are needed to integrate with Watson AI services.



Enhancing Internet of Things solutions with REST APIs

Figure 6-17. Watson Documentation

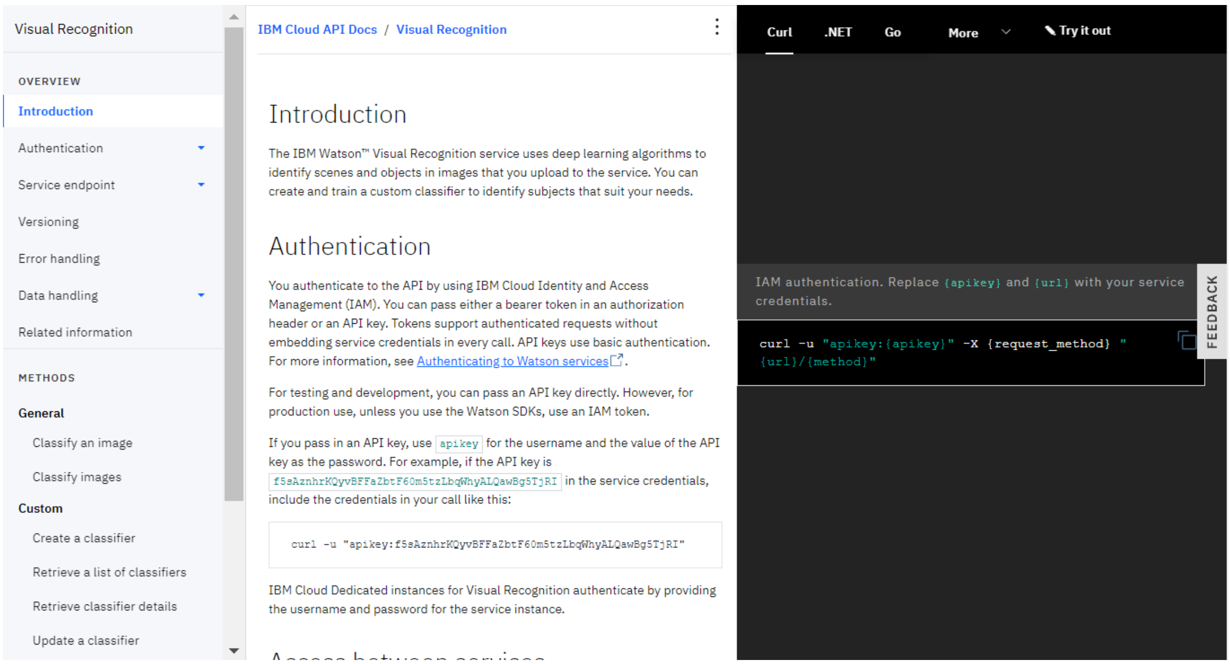
Watson documentation provides API references to each service that can be integrated through APIs or a software development kit (SDK).

Reference:

https://cloud.ibm.com/developer/watson/documentation

IBM Training

Watson Visual Recognition APIs



Enhancing Internet of Things solutions with REST APIs

© Copyright IBM Corporation 2019, 2020

Figure 6-18. Watson Visual Recognition APIs

This slide shows the landing page of the available APIs in the Visual Recognition service.

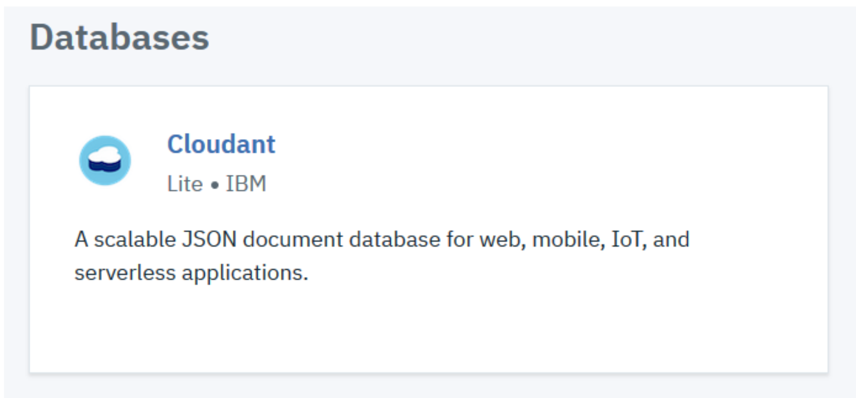
Here are the available API options:

- .NET
- Curl
- Go
- Ruby
- Python
- Node
- Java
- Swift
- Unity

IBM Training

The Cloudant service on IBM Cloud

- *Cloudant* is an IBM software product, which is primarily delivered as a cloud-based service.
- It is also a non-relational and distributed *database* service of the same name.
- It is based on the Apache-backed CouchDB project and the open source BigCouch project.
- It is a fully managed JSON document database.



Enhancing Internet of Things solutions with REST APIs

Figure 6-19. The Cloudant service on IBM Cloud

Another service that has APIs to connect with Watson IoT is the Cloudant database.

Cloudant is a distributed database that is optimized for handling heavy workloads that are typical of large and fast-growing web and mobile apps. Available as a service-level agreement (SLA) -backed and fully managed IBM Cloud service, Cloudant elastically scales throughput and storage independently. The IBM Cloud service provides a fully managed and distributed JSON document database. You can instantly deploy an instance, create databases, and independently scale throughput capacity and data storage to meet your application requirements.

IBM Training

Accessing the Cloudant database by using REST APIs

- **GET**
- **POST**
- **PUT**
- **DELETE**

Enhancing Internet of Things solutions with REST APIs

Figure 6-20. Accessing the Cloudant database by using REST APIs

Cloudant uses a REST API to provide simple and web-based access to data in the Cloudant data store.

The REST API is a programmatic way of accessing the data from your applications. It provides several REST access methods for data read, add, update, and delete functions.

You can use the following HTTP Request methods:

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

IBM Training

Documents in Cloudant

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

Enhancing Internet of Things solutions with REST APIs

Figure 6-21. Documents in Cloudant

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

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

IBM Training

Sample database in Cloudant

The Cloudant sample database that is created by the starter kit is named `my_sample_db`, and it can be found in the following directory:

`https://$USERNAME.cloudant.com/my_sample_db/`


```
update_seq:            "0-g1AAAAP3eJzLYWBgEMhgTm…ID3ZAfCB4gBoJSQBQDEokPV"
db_name:               "my_sample_db"
purge_seq:             0
sizes:                 {…}
props:                 {}
other:                 {…}
doc_del_count:         0
doc_count:             0
disk_size:             133940
disk_format_version:   8
data_size:             0
compact_running:       false
cluster:               {…}
instance_start_time:   "0"
```


Enhancing Internet of Things solutions with REST APIs

Figure 6-22. Sample database in Cloudant

The sample database that is created by the Node.js Cloudant DB Web Starter Boilerplate is named my_sample_db. You can get more information about the database by calling the following REST API:

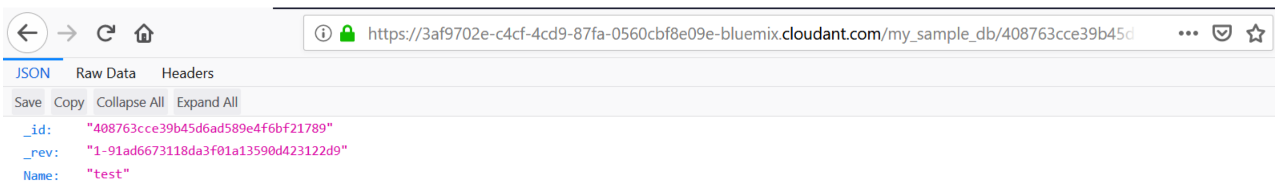
`https://$USERNAME.cloudant.com/my_sample_db/`

$USERNAME is the username of the Cloudant DB that you saw in the IBM Cloud VCAP_SERVICES environment variables.

IBM Training

Reading a document in Cloudant

To access a document with the Cloudant API, run a **GET** request against `https://$USERNAME.cloudant.com/$DATABASE/$DOCUMENT_ID`.



Enhancing Internet of Things solutions with REST APIs

Figure 6-23. Reading a document in Cloudant

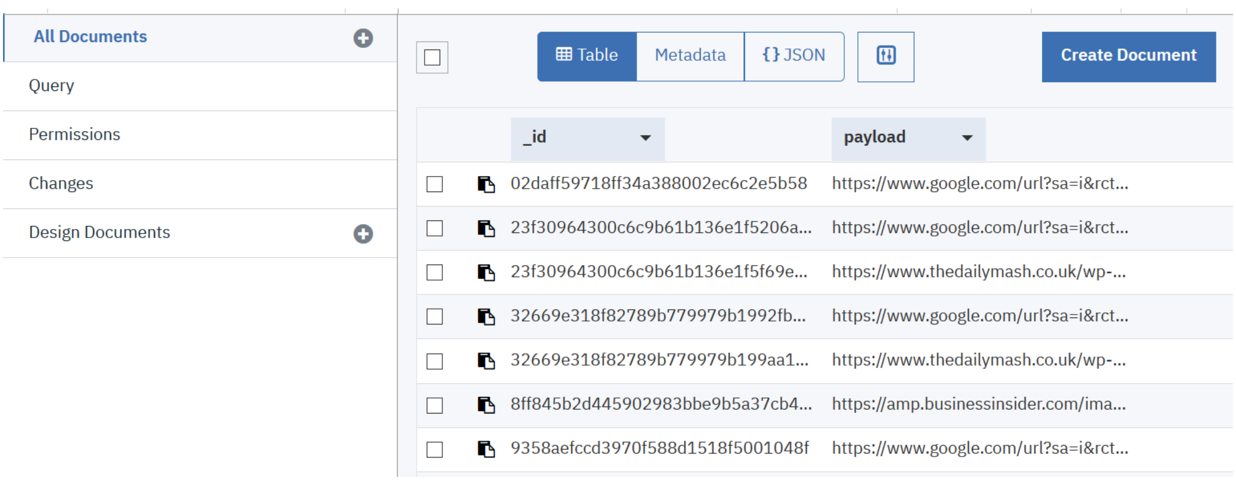
The figure in this slide shows accessing a document from the browser. From your application, you can access the document by using REST APIs.

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

IBM Training

Viewing all documents

To view all the documents that are in a database, run a **GET** request against https://`$USERNAME.cloudant.com/$DATABASE/_all_docs?includ e _doc=true`.



Enhancing Internet of Things solutions with REST APIs

Figure 6-24. Viewing all documents

Cloudant includes an index that is named _all_docs that you can use to build a URL to list all the documents in the database. You can pass to the index an optional parameter that is named include_docs to return the contents of the documents in addition to the _id and _rev. Combining a **POST** request with a view by using the *include_docs=true* query argument enables you to retrieve multiple documents from a database.

6.4. Enhancing an IoT solution with Watson AI

IBM Training

Enhancing an IoT solution with Watson AI

Enhancing Internet of Things solutions with REST APIs

Figure 6-25. Enhancing an IoT solution with Watson AI

IBM Training

Topics

- Application programming interface overview
- Watson IoT Platform HTTP REST API
- Accessing Watson and Cloudant services with REST APIs
- Enhancing an IoT solution with Watson AI

Enhancing Internet of Things solutions with REST APIs

Figure 6-26. Topics

IBM Training

The face detection security system

- One of the IoT solutions with Watson Visual Recognition is to detect the people who enter a facility building by capturing their faces and classifying them according to age group and gender.
- By using applications based on these services, facility building management or security can track the number of visitors and get insights about the age groups of the visitors and their gender.
- This solution can be implemented in any building, shop, or museum that is interested in increasing their marketing efforts or detecting their market segments by analyzing visitor's data.

Enhancing Internet of Things solutions with REST APIs

Figure 6-27. The face detection security system

The *face detection security system* is a system that can detect intruders to restricted or high-security areas and help minimize human error. This system is composed of two parts: hardware and software. The hardware consists of a camera, and the software consists of face-detection and face-recognition algorithm software. When a person enters the zone in question, a series of snapshots is taken by the camera and sent to the software to be analyzed and compared with an existing database of trusted people. An alarm goes off if the user is not recognized.

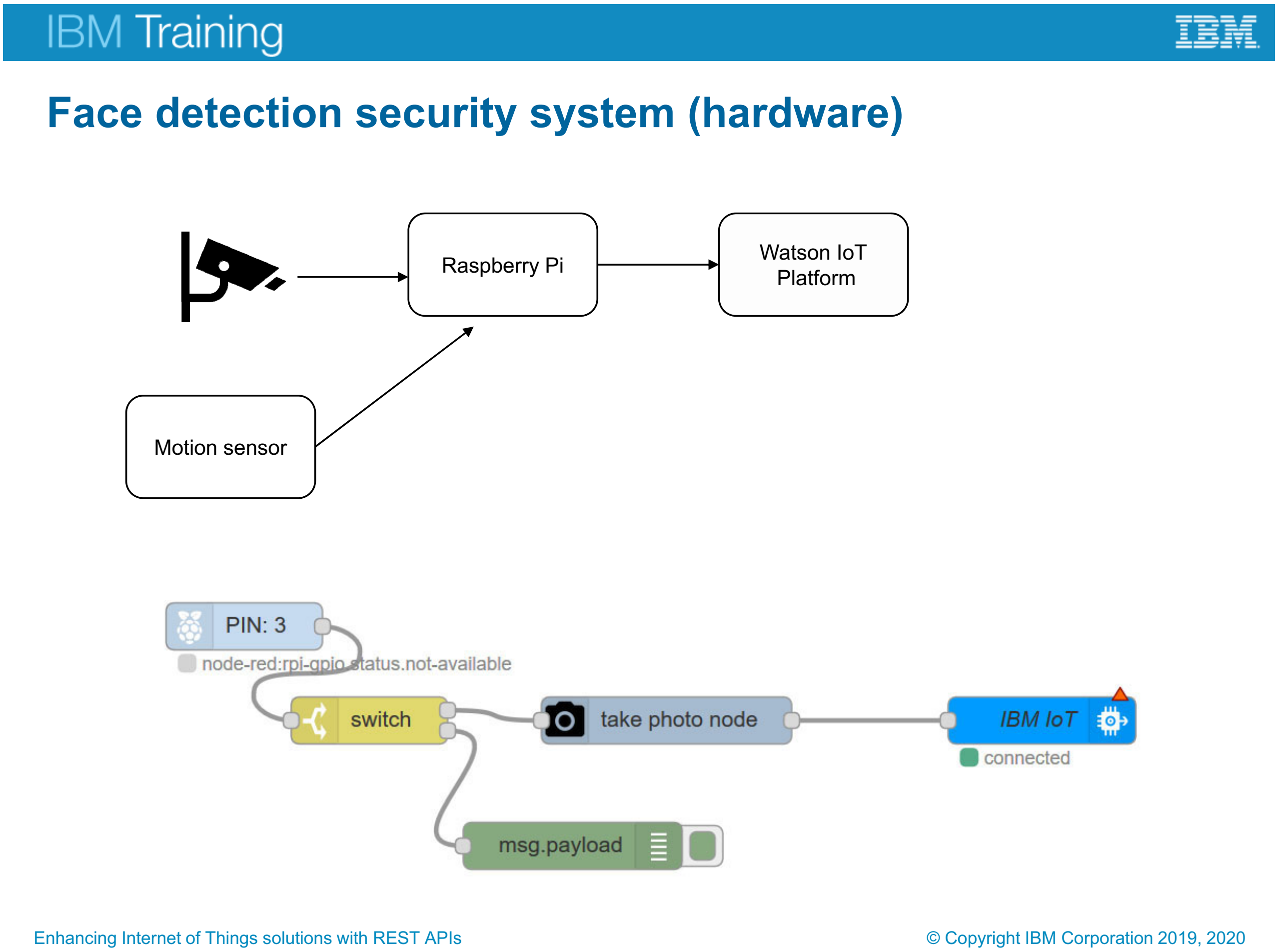


Figure 6-28. Face detection security system (hardware)

1. The motion sensor detects a human moving.
2. A camera takes their picture.
3. A picture is sent to Watson IoT Platform.

IBM Training

Face detection security system (software)

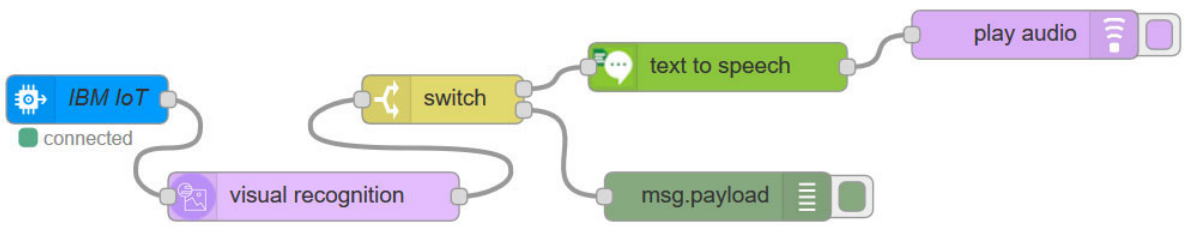


Figure 6-29. Face detection security system (software)

1. Watson IoT Platform receives the image.
2. It sends the image to Watson Visual Recognition.
3. If this person is not recognized, text to speech takes the output result and plays it in Audio sound.

Note

Both Watson Visual Recognition and text to speech Watson services were used with the Watson IoT Platform to enhance the solution.

IBM Training

Unit summary

- Explain the concept of the application programming interface (API) and describe the types of APIs.
- Explain the characteristics of REST APIs.
- List the Watson artificial intelligence (AI) services.
- Describe the IBM Cloudant database service on IBM Cloud.
- Describe how to access Watson and Cloudant services on IBM Cloud with REST APIs.

Enhancing Internet of Things solutions with REST APIs

Figure 6-30. Unit summary

IBM Training

Review questions

1. True or False: APIs are pervasive in mobile only.

2. What is the available API in Watson services?
   A. Curl
   B. C#
   C. PHP
   D. C++

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

Enhancing Internet of Things solutions with REST APIs

Figure 6-31. Review questions

IBM Training

Review questions (cont.)

4. True or False: Watson IoT Platform HTTP REST API supports device diagnostic operations.

5. True or False: To access a document by using the Cloudant API, issue a **POST** request.

Enhancing Internet of Things solutions with REST APIs

Figure 6-32. Review questions (cont.)

IBM Training

Review answers

1. True or False: APIs are pervasive in mobile only
   False. APIs are pervasive both in the mobile and web arenas.

2. What is the available API in Watson services?
   A. Curl
   B. C#
   C. PHP
   D. C++
   The answer is A.

3. True or False. Cloudant databases and documents are accessed by using REST APIs.
   The answer is True.

Enhancing Internet of Things solutions with REST APIs

Figure 6-33. Review answers

IBM Training

Review answers (cont.)

4. True or False: Watson IoT Platform HTTP REST API supports device diagnostic operations.
   The answer is True.
5. True or False: To access a document by using the Cloudant API, issue a **POST** request.
   The answer is False.

Figure 6-34. Review answers (cont.)

IBM Training

Exercise: Integrating IoT applications by using IBM Watson Visual Recognition

Enhancing Internet of Things solutions with REST APIs

Figure 6-35. Exercise: Integrating IoT applications by using IBM Watson Visual Recognition

IBM Training

Exercise objectives

- In this exercise, you learn about the Watson Visual Recognition service and how to integrate it with IoT applications by using Watson IoT Platform Service and Node-RED.
- After completing this exercise, you should be able to:
  - Create a Watson service and integrate it with Node-RED.
  - Create an instance of the Watson Visual Recognition service and classify data.
  - Store and extract data from CloudantDB by using Node-RED.
  - Create a Watson Studio project and test the Watson Visual Recognition classifier.

Enhancing Internet of Things solutions with REST APIs

Figure 6-36. Exercise objectives

Unit 7. Introduction to analytics services on IBM Cloud

Estimated time

01:30

Overview

This unit provides an overview of the analytics services available on IBM Cloud. The topics in this unit belong to the data science domain, but IoT systems usually require combining data science and big data capabilities.

IBM Training

Unit objectives

- Describe the features of IBM Analytics Engine and IBM Streaming Analytics.
- Describe IBM Geospatial Analytics as a use case of IBM Streaming Analytics.
- Explain the geospatial capabilities in IBM Cloudant DB.

Figure 7-1. Unit objectives

7.1. Analytics services on IBM Cloud overview

IBM Training

Analytics services on IBM Cloud overview

Introduction to analytics services on IBM Cloud

Figure 7-2. Analytics services on IBM Cloud overview

IBM Training

Topics

- Analytics services on IBM Cloud overview
- IBM Analytics Engine overview
- IBM Streaming Analytics overview
- Geospatial capabilities in IBM Cloudant

Introduction to analytics services on IBM Cloud

Figure 7-3. Topics

IBM Training

What is the meaning of IOT analytics

- One of the applications of the data analytics is *IoT analytics.*
  - The data that is generated from Internet of Things (IoT) devices is valuable only if it is analyzed.
- IoT analytics services can run many processes (filter, transform, analyze, and enrich) on the data that is collected by external devices on IoT systems that usually require combining data science and big data capabilities.

Introduction to analytics services on IBM Cloud

Figure 7-4. What is the meaning of IOT analytics

Data analytics is defined as a process, which is used to examine large and small data sets with varying data properties to extract meaningful conclusions and actionable insights.

IoT analytics is an application for data analytics. IoT analytics means running analytics algorithms on large volumes of IoT data without managing hardware or an infrastructure. With IoT analytics, data scientists can make better and more accurate decisions for IoT applications and machine learning use cases.

IoT data is highly unstructured, which makes it difficult to analyze with traditional analytics and business intelligence (BI) tools that are designed to process structured data.

IoT analytics services can run many processes (filter, transform, analyze, and enrich) on the data that is collected by external devices on IoT systems that usually require combining data science and big data capabilities.

IBM Training

Types of data analytics provided by IBM Cloud

- Streaming Analytics
- Spatial Analytics
- Time Series Analytics

Introduction to analytics services on IBM Cloud

Figure 7-5. Types of data analytics provided by IBM Cloud

There are different types of DA that you can use and apply to IoT data. Some of these types are as follows:

- Streaming Analytics: An advanced analytic platform that you can use to ingest, analyze, and correlate information as it arrives from different types of data sources in real time.
- Spatial Analytics: The DA method that generates and analyzes data about geographic features, and stores and manages the data on which that information is based.
- Time Series Analytics: A sequence of data values that are measured at successive, though not necessarily regular, points. You can use special SQL functions to process time series data, for example, to identify trends and predict future values based on these trends.

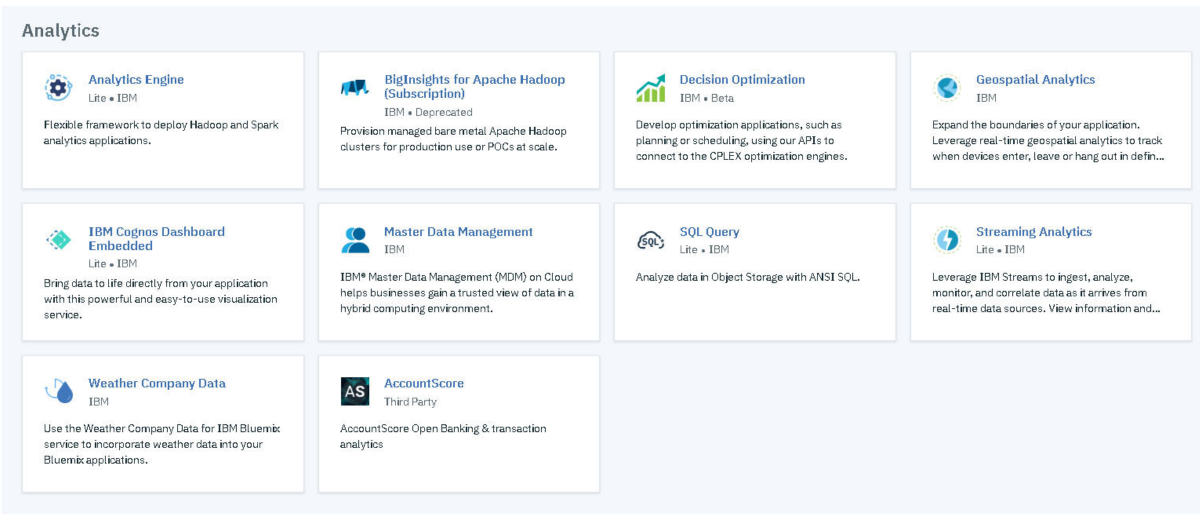
References:

- https://cloud.ibm.com/docs/services/StreamingAnalytics?topic=StreamingAnalytics-gettingstarted
- https://www.ibm.com/support/knowledgecenter/SS6NHC/com.ibm.swg.im.dashdb.analytics.doc/doc/geo_main.html
- https://cloud.ibm.com/docs/services/sql-query?topic=sql-query-ts_intro

IBM Training

Analytics services on IBM Cloud overview

If you want to extract value from data that is collected from IoT sensors, you must use storage and analytics to convert the raw data to analyzed data that has real business value.



Introduction to analytics services on IBM Cloud

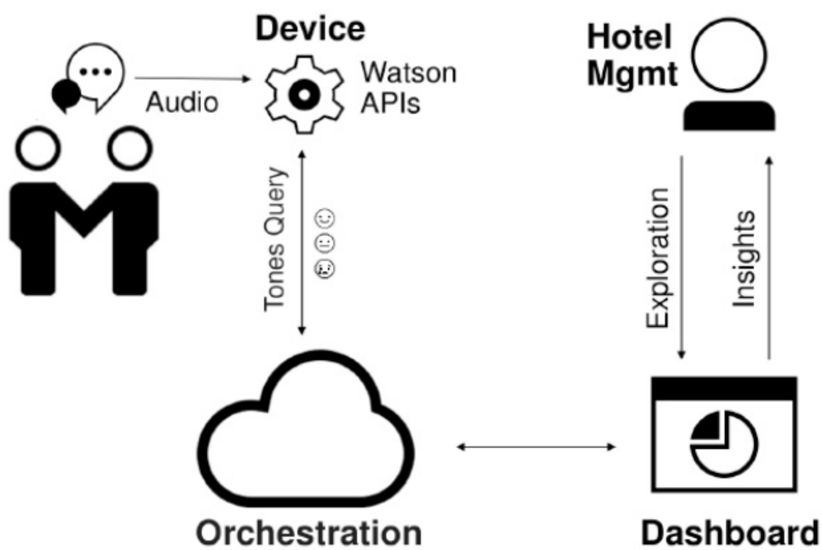
Figure 7-6. Analytics services on IBM Cloud overview

If you want to extract value from data that is collected from IoT sensors, you must use storage and analytics to convert the raw data to analyzed data that has real business value.

This unit provides a brief overview of some IBM Analytics services.

IBM Training

How analytics can support IoT (IBM Edge IoT Analytics)



Introduction to analytics services on IBM Cloud

Figure 7-7. How analytics can support IoT (IBM Edge IoT Analytics)

An *edge computing* application uses the processing power of IoT devices to filter, pre-process, aggregate, or score IoT data. It uses the power and flexibility of cloud services to run complex analytics on data, and in a feedback loop supports decisions and actions about the physical world.

This example demonstrates the workflow of IBM Edge IoT Analytics:

A large global hotel chain wants to improve the process that is used to collect customer satisfaction. They realize that spontaneous information about that date is exchanged daily at their hotels premises but is systematically lost. Guests interact continuously with hotel employees at the reception desk, and all these interactions implicitly or explicitly convey a mood or a tone that directly links to their satisfaction.

What if this information could be somehow captured and analyzed to produce better customer insights? What if real-time customer data from all the hotels in the globe could be easily queried and analyzed by the hotel management board through a simple visual dashboard and query interface?

To address these challenges, you can use *IBM Edge IoT Analytics*. You can use Watson Speech-to-Text and Watson Tone Analyzer as Edge Gateway modules, and a distributed IoT query engine runs as a Cloud module.

You can use inexpensive devices such as a Raspberry Pi that are equipped with a microphone and placed at the reception desks of hotels that act as *edge gateways*.

The edge application modules running on the gateways capture conversations and run speech-to-text and tone analysis directly on the device so that sensitive data is not sent to the cloud. The edge gateways store only the results of tone analysis, which are numerical scores for a set of fixed tone attributes like happiness, sadness, or anger, on a local database.

So that hotel management access this global information, IBM Edge IoT Analytics has an extra module running as an IBM Cloud service that implements a visual dashboard with a query engine to explore IoT information. From the map on the dashboard, it is possible to visualize the devices on which the application is running. The IoT query engine, which is known as EdgeSQL, implements a subset of standard SQL with extensions that makes it suitable to query data that is stored on edge gateways.

Using EdgeSQL, the hotel management can build and run queries like “What is the hotel in the chain that had the most angry interactions with guests?”, “What is the average happiness of customer interactions in Rome?”, or “What was the saddest conversation in Athens within the last three months?”.

All the communication between the cloud and edge devices uses the services that are provided by the Watson IoT Platform.

Reference:

https://www.ibm.com/blogs/internet-of-things/edge-iot-analytics/

7.2. IBM Analytics Engine overview

IBM Training

IBM Analytics Engine overview

Introduction to analytics services on IBM Cloud

Figure 7-8. IBM Analytics Engine overview

IBM Training

Topics

- Analytics services on IBM Cloud overview
- IBM Analytics Engine overview
- IBM Streaming Analytics overview
- Geospatial capabilities in IBM Cloudant

Introduction to analytics services on IBM Cloud

Figure 7-9. Topics

IBM Training

IBM Analytics Engine overview

- Provides a single environment capable of handling workloads of data scientists, data engineers, and data developers
- Combines open source Apache Spark and Apache Hadoop so that developers can quickly deploy analytics applications.
- Separates compute and storage so that there is no data loss if there is a cluster failure.

Figure 7-10. IBM Analytics Engine overview

With IBM Analytics Engine, you can quickly create Apache Spark and Apache Hadoop clusters and customize these clusters by using scripts. You can work with data in IBM Cloud Object Storage and integrate other IBM Watson services like IBM Watson Studio and IBM Machine Learning. You can define clusters that are based on your application's requirements by choosing the appropriate software pack, version, and size of the clusters.

A *cluster* consists of a management instance and one or more compute instances. The management instance consists of three management nodes, which run in the management instance. Each of the compute nodes run in a separate compute instance.

IBM Training

Use cases for IBM Analytics Engine

- Simplifying disaster recovery
  Disaster recovery (DR) in traditional Hadoop deployments can be difficult to implement. IBM Analytics Engine can bypass this difficulty because its storage architecture is different from a traditional Hadoop cluster because it uses IBM Cloud Object Storage.
- Dealing with different workloads
  With IBM Analytics Engine, you have a single permanent cluster that is shared by multiple teams, so each user can start their own personal cluster whenever they need it.

Introduction to analytics services on IBM Cloud

Figure 7-11. Use cases for IBM Analytics Engine

Here are some examples of IBM Analytics Engine use cases:

Simplifying disaster recovery

Disaster recovery (DR) is one of the challenges of traditional Hadoop deployments. With a permanent cluster that is constantly in use, it is difficult to find an appropriate time to run backups. Finding an appropriate backup target is also a problem because few companies want to invest in an entire second cluster to handle backups.

IBM Analytics Engine can bypass these problems because its storage architecture is different from a traditional Hadoop cluster. Instead of each node having its own local storage, the whole cluster is connected to a separate object storage repository.

Object storage solutions such as IBM Cloud Object Storage can automatically distribute multiple replicas of data across a cluster of storage systems in different data centers or even different regions. The data remains highly available (HA) even when one or more storage nodes fail.

With IBM Analytics Engine, DR is standard, so there is no longer any need for Hadoop users or administrators to worry about how, where or when to back up their data.

Dealing with different workloads

Even if your company's big data capabilities are established and your data science workflow is seamlessly efficient, you might face challenges as user numbers rise and demand increases.

It can be difficult to build a single, permanent Hadoop cluster that can serve the needs of multiple different groups of users and types of workload. Segregating the cluster into secure zones for each group can be an administrative headache, and it might not be possible to find a configuration that is optimal for both daytime interactive workloads and overnight batch processes.

With IBM Analytics Engine, you can sidestep these problems. Instead of a single permanent cluster that is shared by multiple teams, each user can start their own personal cluster whenever they need it. As a result, you do not need to worry about managing security within the cluster: You can use the same identity and access management framework that you use for all your other cloud services.

Similarly, there is no need for a "one-size-fits-all" approach to different workloads. At the end of each day, you can start a cluster that is configured to optimize performance for your overnight batch processes. In the morning, you can instantiate a separate cluster with the tools that are required for *ad hoc* interactive queries. If you must scale up or down as your workload profile changes, this task is easily accomplished.

References:

- https://www.ibm.com/cloud/blog/putting-engine-work-ibm-analytics-engine-can-help-harness-hadoop-spark-business-benefit
- https://www.ibm.com/downloads/cas/KDPB1REE

IBM Training

Key features

- Customize the environment and integrate with IBM Watson services.
- Use the power of open source by using technology that is based on Apache Spark and Apache Hadoop.
- Start and scale clusters on demand.

Analytics Engine
IBM • Analytics

Flexible framework to deploy Hadoop and Spark analytics applications.

Lite • Free • IAM-enabled • Service Endpoint Supported

Introduction to analytics services on IBM Cloud

Figure 7-12. Key features

Here are the key features of IBM Analytics Engine:

- Customize the environment: Customize clusters by using third-party analytics libraries and packages, and deploy workloads from IBM Watson Studio and IBM Watson Machine Learning.
- Use the power of open source: Build on an Open Data Platform initiative (ODPi)-compliant Apache Spark and Apache Hadoop stack to expand open source investments. You can integrate analytics tools by using open source APIs and libraries.
- Spin up and scale clusters on demand: Define clusters that are based on your application's requirements by choosing the appropriate software package, version, and size of the cluster. Use the cluster while it is required and delete it when all the jobs are finished.

Reference:

https://cloud.ibm.com/catalog/services/analytics-engine?customCreatePageKey=catalog-custom-key-emb81qj1

7.3. IBM Streaming Analytics overview

IBM Training

IBM Streaming Analytics overview

Introduction to analytics services on IBM Cloud

Figure 7-13. IBM Streaming Analytics overview

IBM Training

Topics

- Analytics services on IBM Cloud overview
- IBM Analytics Engine overview
- IBM Streaming Analytics overview
- Geospatial capabilities in IBM Cloudant

Introduction to analytics services on IBM Cloud

Figure 7-14. Topics

IBM Training

IBM Streaming Analytics overview

IBM Streaming Analytics for IBM Cloud evaluates a broad range of streaming data, unstructured text, video, audio, geospatial data, and sensors, which help organizations spot opportunities and risks and make decisions in real time.

You can use IBM Streams to ingest, analyze, monitor, and correlate data as it arrives from real-time data sources. You also can view information and events as they unfold.

Introduction to analytics services on IBM Cloud

Figure 7-15. IBM Streaming Analytics overview

You can use IBM Streams to ingest, analyze, monitor, and correlate data as it arrives from real-time data sources. You also can view information and events as they unfold.

Here are its key features:

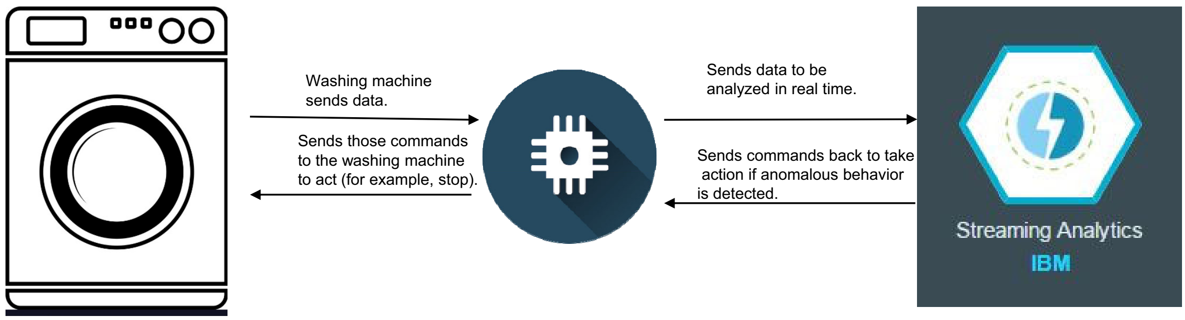
- Analyze data in motion: Perform real-time analysis on data-in-motion by using your IBM Cloud application. The IBM Streaming Analytics service is powered by IBM Streams, which can analyze millions of events per second, which enables submillisecond response times and instant decision-making.
- Deploy your IBM Streams applications to IBM Cloud: Deploy your IBM Streams applications to your IBM Streaming Analytics instance that is running in IBM Cloud. IBM Streams can handle high data rates and perform its analysis with predictable low latency so that your application can operate at the speed of data.

Reference:

https://cloud.ibm.com/catalog/services/streaming-analytics

IBM Training

IBM Streaming Analytics use case



Introduction to analytics services on IBM Cloud

Figure 7-16. IBM Streaming Analytics use case

One of the use cases that uses stream analytics is analyzing data that comes from IBM IoT Platform.

For example, suppose that a washing machine sends its data IBM Streaming Analytics (fluid events, voltage events, and mechanical events). The flow is as follows:

1. The washing machine sends data to IBM IoT Platform.
2. The IoT Platform sends that data to the IBM Streaming Analytics service, which imports these events in real time and performs the following operations:
    a. View Events: Displays the events in the IBM Streaming Analytics dashboard.
    b. Threshold-Based Anomaly detection: Detects the threshold breach and sends a command back to the device to take corrective actions.
    c. Anomaly detection: Detects the anomaly by comparing the current set of values with the previous set and reports to the device by sending a command.
    d. Statistics calculation over a period: Aggregates the events for a specified time, computes the statistics, and sends a command back to the device.

3. The IBM Streaming Analytics service analyzes the data, detects an anomaly, and sends a command to take corrective actions.
4. The IoT Platform sends this command to inform the Washing Machine to act (for example, stop).

7.4. Geospatial capabilities in IBM Cloudant

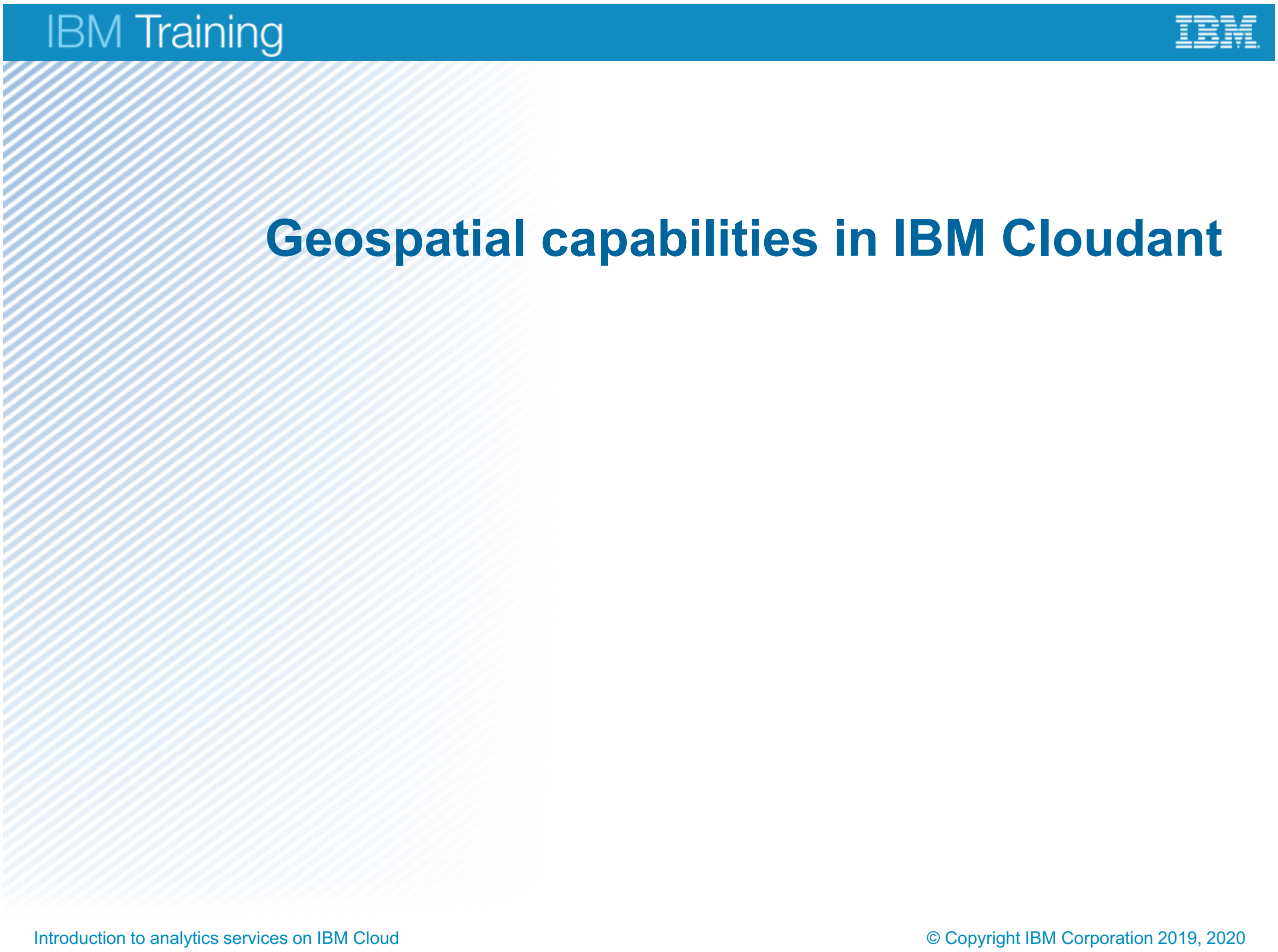


Figure 7-17. Geospatial capabilities in IBM Cloudant

IBM Training

Topics

- Analytics services on IBM Cloud overview
- IBM Analytics Engine overview
- IBM Streaming Analytics overview
- Geospatial capabilities in IBM Cloudant

Introduction to analytics services on IBM Cloud

Figure 7-18. Topics

IBM Training

Key features

IBM Cloudant for IBM Cloud Geospatial (IBM Cloudant Geo) combines the advanced geospatial queries of a Geographic Information System (GIS) with the flexibility and scalability of the IBM Cloudant database-as-a-service (DBaaS) capabilities.

Here are the key features:

- Enables web and mobile developers to enhance their applications by using geospatial operations that go beyond simple bounding boxes.
- Integrates with existing GIS applications so that they can scale to accommodate different data sizes, concurrent users, and multiple locations.
- Provides a NoSQL capability for GIS applications so that large streams of data can be acquired from devices, sensors, and satellites. This data can then be stored, processed, and syndicated across other web applications.

Introduction to analytics services on IBM Cloud

Figure 7-19. Key features

You can use IBM Cloudant Geo to structure your data by using the GeoJSON format. Design documents index the data. As with other IBM Cloudant documents, an initial scan examines all the documents in the database and outputs the first index. Subsequent updates to the documents result in incremental updates to the index.

The key advantage to using IBM Cloudant Geo is that you can identify, specify, or search for documents that are based on a spatial relationship. In effect,

IBM Cloudant uses geometry to provide an extra way of expressing the relationship between and within documents.

An example is specifying a document that is “contained” if it has a geospatial characteristic that fits within a geospatial polygon, which is defined by a series of points.

Reference:

https://cloud.ibm.com/docs/services/Cloudant/api?topic=cloudant-cloudant-nosql-db-geospatial#cloudant-nosql-db-geospatial

IBM Training

GeoJSON format

In 2015, the Internet Engineering Task Force (IETF) and the original specification authors formed a GeoJSON workgroup to standardize GeoJSON. RFC 7946 was published in August 2016 and is the new standard specification of the GeoJSON format. It replaces the 2008 GeoJSON specification.

For more information about this topic, see the following website:
http://geojson.org/

Introduction to analytics services on IBM Cloud

Figure 7-20. GeoJSON format

Here is an example of a relationship that uses a geospatial polygon:

relation=contains&g=POLYGON ((-71.0537124 42.3681995,-71.054399 42.3675178,-71.0522962 42.3667409,-71.051631 42.3659324,-71.051631 42.3621431,-71.0502148 42.3618577,-71.0505152 42.3660275,-71.0511589 42.3670263,-71.0537124 42.3681995))

The GeoJSON format is used to express the following various geographic data structures:

- Point
- LineString
- Polygon
- MultiPoint
- MultiLineString
- MultiPolygon
- GeometryCollection

IBM Training

Mapbox integration with IBM Cloudant Dashboard

IBM Cloudant Geo provides a map visualization by using Mapbox, which introduces an intuitive experience for Geo-Query.

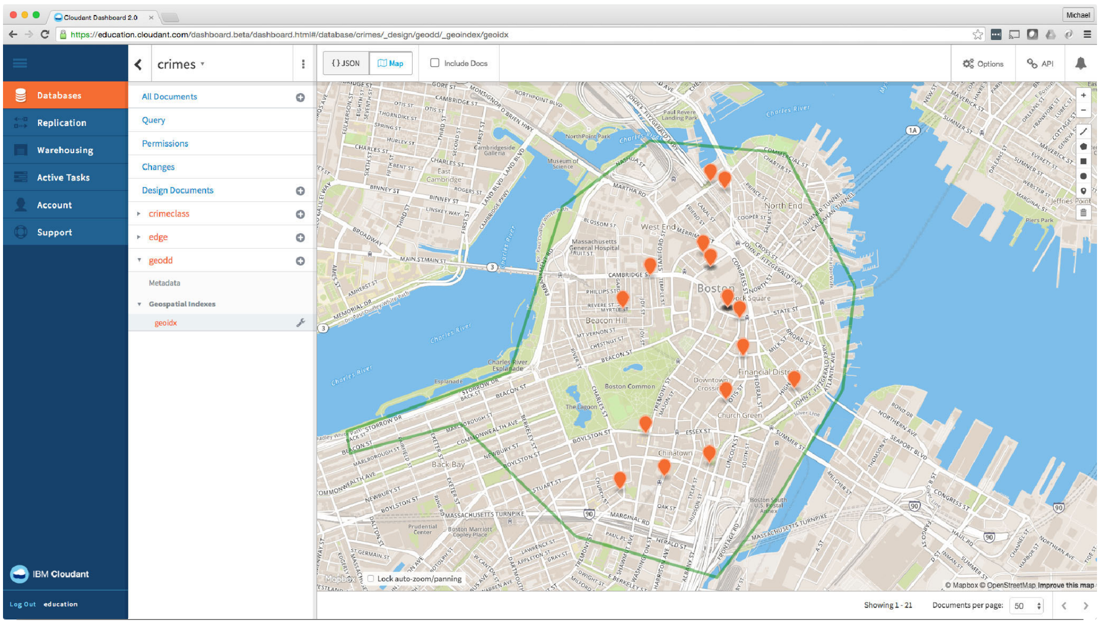


Figure 7-21. Mapbox integration with IBM Cloudant Dashboard

Mapbox is an open source mapping platform for customized maps. The IBM Cloudant APIs and SDKs are the building blocks to integrate locations into any mobile or web app.

References:

- https://www.mapbox.com/
- https://developer.ibm.com/clouddataservices/2016/04/11/geojson-database-cloudant-mapbox/

IBM Training

Unit summary

- Describe the features of IBM Analytics Engine and IBM Streaming Analytics.
- Describe IBM Geospatial Analytics as a use case of IBM Streaming Analytics.
- Explain the geospatial capabilities in IBM Cloudant DB.

Figure 7-22. Unit summary

IBM Training

Review questions

1. True or False: IBM Cloud provides open source products for analytics.
2. True or False: IBM Streaming Analytics does not perform real-time analysis on data.
3. What is the platform that is provided by IBM Cloudant Geo for visualization?
    A. Google Maps.
    B. IBM provides its own mapping.
    C. Mapbox.
    D. None of the above.

Introduction to analytics services on IBM Cloud

Figure 7-23. Review questions

1. True
2. False
3. C

IBM Training

Review questions (cont.)

4. IBM Analytics Engine uses _____ to save data.
   A. Cloudant.
   B. SQL DB.
   C. Object Storage DB.
   D. Mongo DB.
5. What types of DA are provided by IBM Cloud?
   A. Fuzzy logic.
   B. Streaming Analytics.
   C. Time series.
   D. Neural networks.

Figure 7-24. Review questions (cont.)

1. True
2. False
3. C
4. C
5. B and C

IBM Training

Review answers

1. True or False: True or False: IBM Cloud provides open source products for analytics.
   The answer is True. IBM Analytics provide Apache Spark and Hadoop.
2. True or False: IBM Streaming Analytics does not perform real-time analysis on data.
   The answer is False. IBM Streaming Analytics performs real-time analytics on data.
3. What is the platform that is provided by IBM Cloudant Geo for visualization?
   A. Google Maps.
   B. IBM provides its own mapping.
   C. Mapbox.
   D. None of the above.
   The answer is C.

Introduction to analytics services on IBM Cloud

Figure 7-25. Review answers

IBM Training

Review answers (cont.)

4. IBM Analytics Engine uses ______ to save data.
   - A. Cloudant.
   - B. SQL DB.
   - C. Object Storage DB.
   - D. Mongo DB.

   The answer is C.
5. What types of DA are provided by IBM Cloud?
   - A. Fuzzy logic.
   - B. Streaming Analytics.
   - C. Time series.
   - D. Neural networks.

   The answers are B and C.

Introduction to analytics services on IBM Cloud

Figure 7-26. Review answers (cont.)

Figure 7-27. Exercise: Monitoring parking usage by using IBM Cloudant Geospatial

IBM Training

Exercise objectives

- In this exercise, you learn about geofencing, which is one of the common use cases of IoT. This exercise demonstrates how to configure geofencing in IBM Cloudant, and how to build a Node-RED flow that detects when a car reaches certain points.
- After completing this exercise, you should be able to:
  - Demonstrate how you can use Node-RED to build a car simulator.
  - Demonstrate how to use raw data and analyze it in real time to fulfill a business case.
  - Understand how to configure IBM Cloudant to do geospatial analytics.

Introduction to analytics services on IBM Cloud

Figure 7-28. Exercise objectives

IBM Training