

A developer's guide to the Internet of Things (IoT)- Coursera

DJ's notes

2019

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 1.1 | What is IoT? | 3 |
| 1.2 | What is the value of IoT? | 3 |
| 1.3 | Why is IoT so special ? | 3 |
| 1.4 | How does IoT works | 3 |
| 2 | Rapid application development in the cloud | 4 |
| 2.1 | Cloud computing | 4 |
| 2.2 | IBM computing | 4 |
| 2.3 | Rapid application for IoT | 4 |
| 2.4 | Introduction to NodeRED | 6 |
| 2.5 | Additional NodeRED nodes | 10 |
| 3 | Rapid application development on a Raspberry Pi | 10 |
| 3.1 | Rasberry Pi and SenseHAT | 10 |
| 3.2 | NodeRED on Raspberry Pi part 1 | 12 |
| 3.3 | NodeRED on Raspberry Pi part 2 | 13 |
| 3.4 | Introduction to the Watson IoT Platform | 14 |
| 3.5 | Controlling a device | 14 |
| 4 | Lower level programming for the Internet of Things | 14 |
| 4.1 | Watson IoT APIs | 14 |
| 4.2 | MQTT | 15 |
| 4.3 | Deploying Applications to the IBM Cloud | 15 |
| 4.4 | Controlling a device | 15 |
| 5 | Error FAQ | 16 |

1 Introduction

1.1 What is IoT?

- IoT is a network of interconnected things, objects, devices, and systems that connects and transmit data and exchange data among them.
- The data exchanged is collected, analyzed and acted on them.

1.2 What is the value of IoT?

- Collecting data from IoT based devices (such as wearables and smart appliances) enables business to learn more about their operating environment.
- This way, businesses can identify and act on the potential to create new value.
- Value by unlocking your revenue from existing products and services, value by inspiring new working practices and processes, value by changing or creating new business models or strategies.
- The potential of Internet of Things lies on the intelligence.

1.3 Why is IoT so special ?

- Standard connectivity which can send data to the cloud from anywhere.
- Discrete industries come together
- Provide a service

1.4 How does IoT works

- How does IoT work? <https://www.youtube.com/watch?v=QSIPNh0iMoE&feature=youtu.be>
- The Future of IoT at Work. https://www.youtube.com/watch?v=4jjcznMXF8M&index=4&list=PLBF0HYVTEVoDzBoFYC9PJq-pUT_Ykde0g

2 Rapid application development in the cloud

2.1 Cloud computing

1. Infrastructure-as-a-Service (IaaS)
2. Platform-as-a-Service (PaaS)
3. Software-as-a-Service (SaaS)
4. Function-as-a-Service (FaaS)

2.2 IBM computing

1. Get IBM cloud Lite account for 6months to get access to Bluemix
<https://e5.onthehub.com/WebStore/Security/LtiSignIn.ashx>
2. Apply code
<https://console.bluemix.net/account/billing?accountId=e6269f255a404d6cb1a5849dae2f9510>
3. IBM cloud Identity and access management (IAM)
4. Cloud Foundry Orgs, add a region. <https://console.bluemix.net/account/organizations?accountId=e6269f255a404d6cb1a5849dae2f9510>

2.3 Rapid application for IoT

1. Deploying Node-RED on the IBM Cloud
 - a Catalog <https://console.bluemix.net/catalog>
 - b Node-RED editor guide <https://nodered.org/docs/user-guide/editor>
 - c Go to Node Red (different from course video probably due to update) <https://console.bluemix.net/catalog/starters/node-red-starter>
 - d Enable https://console.bluemix.net/apps/23d84272-669c-44a2-91a3-f009a176740e?paneId=overview&env_id=ibm:yp:eu-gb
 - e Create Platform API key <https://console.bluemix.net/iam/#/apikeys>
 - f Go to Git profile and create an access token (or ssh key) https://git.eu-gb.bluemix.net/profile/personal_access_tokens
 - g Copy and save the token
 - h By default, the project is created as a private repository.

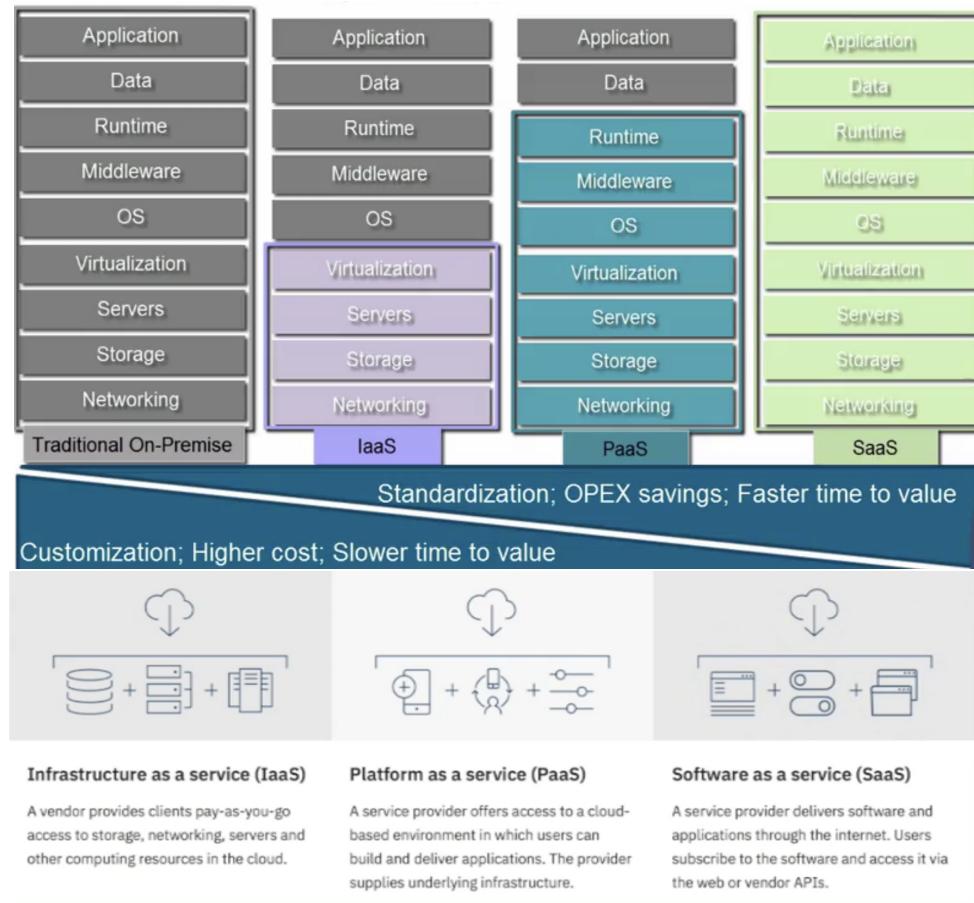


Figure 1: Cloud Computing

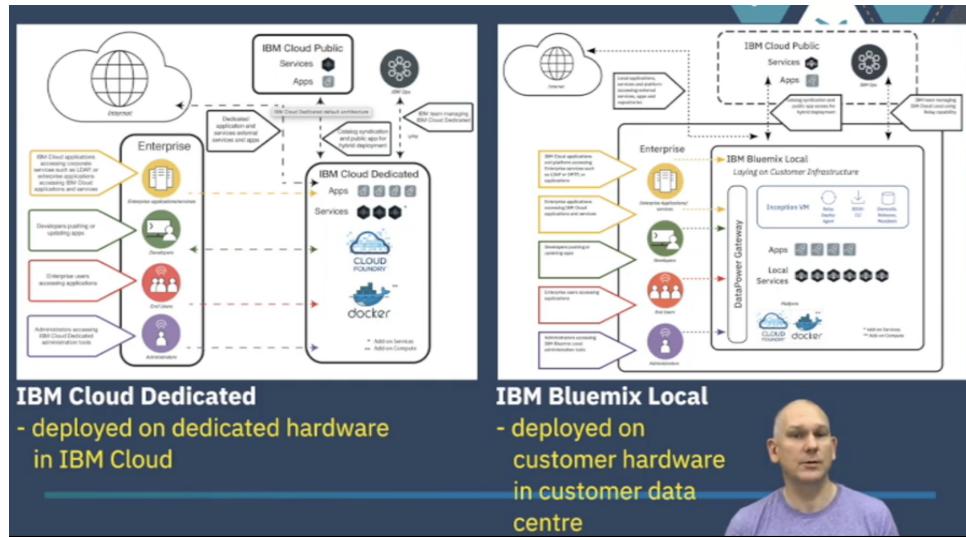


Figure 2: IBMcloud

```
// To access the repository on ibm git repository
git clone https://username@git.eu-gb.bluemix.net/username/jnode.git
// password is the token
```

- i Open the cloned project in Atom, make changes in the editor, add commit messages and commit to the master branch. (gitplus package needed)

2.4 Introduction to NodeRED

1. Introduction to NodeRED 1

- a Access NodeRED <https://jnode.eu-gb.mybluemix.net>
- b Provide a username and password
- c Follow the guide and enter <https://jnode.eu-gb.mybluemix.net/red/#flow/e7bfdaf.a4b2b28>

2. Introduction to NodeRED 2

- a Nodes works by sending JavaScript messages.
 - i. Messages flow along the connectors between nodes
 - ii. When javascript arrives, it is made available as the msg variable.
 - iii. The most important content in a message is found in the payload property
- b To change settings of a node, double click the node.
- c Where are the nodes stored?
 - i. Cloud: NoSQL
 - ii. Raspberrypi: Local
- d How to store the nodes on the NodeRED flow editor user interface?
 - i. Up right: Menu/Export and save as a library
 - ii. Local board: click on sheet then dragging over required nodes and save as text files
- e How to import the nodes on the NodeRED user interface?
 - i. Up Right: Menu/Import text files
 - ii. **Passwords** are not saved when the import and export flows. Need to reinput any passwords stored in the flow.
- f How to install nodes?
 - i. Menu/Manage palette
 - ii. npm install <https://github.com/node-red/node-red>

3. Introduction to NodeRED 3

a Import a function node example

```
// 0. Copy code in references/week2_NodeRED function node_p1.pptx and  
    paste on the node editor.  
  
// A. You must return a javascript object.  
    return {'hello j'}; // return string in javascript object. Correct!  
    return 'hello j', // return string object. An error message will show  
        up on the Sidebar/Debug ("Function tried to send a message of type  
        string")  
  
// B. How does the functional node handle a Javascript array object?  
  
var msg1 = { payload:"first message" };  
var msg2 = { payload:"second message" };  
var msg3 = { payload:"third message" };  
var msg4 = { payload:"fourth message" };  
return [msg1, msg2, msg3, msg4] // An array of four objects.  
  
// If one output is set, only the first message is sent. See  
    "Array".  
  
// If three outputs are set, three messages. See "Multiple returns".  
return [msg1, [msg2, msg3], msg4] // Combine an array within an array  
  
// Set three outputs, msg1 to output1, msg2+3 to output2, msg4 to  
    output3  
  
// Why is this functionality useful? It allows splitting a large  
    set of data into separate messages.  
return [[msg1, msg2, msg3, msg4]] // Enclose an array within an array,  
    the function node will separate the items in the inner array, and  
    send all of them as separate four messages.  
  
// C. How does the JSON object send data down to node?  
  
//C1. Create a json object  
msg.payload ={"a": {"b": "test"}},  
return msg;  
  
//C2. Create a json string  
msg.payload ='{"a": {"b": "test"}}',  
return msg;  
  
//C3. Create a JSON.parse method on the JSON.string  
msg.payload = msg.payload.a.b;  
return msg; //When you interact with third party sites using  
    APIs, libraries, and packages, it's quite common to get the  
    JSON data back as a string.  
  
//C4. Status and logging functions  
//Many APIs in JavaScript use function callbacks. Therefore, we
```

```

        can all an asynchronous function, and when that function
        completes, it will then carry on the flow.

    node.status({text:"started"});

    setTimeout(function() {
        node.status({fill:"red",shape:"ring",text:"stage 1"});

        setTimeout(function() {
            node.status({fill:"green",shape:"dot",text:"stage 2"});
            node.error("This is an error", {payload : 'Caught
                error'}); //catchnode: Catch for a specific or set of
                nodes on this sheet.

            setTimeout(function() {
                node.error("Second error"); //fire an error message,
                go to debugger and system console.
                node.status({});
                node.send({payload : 'Completed'});
            }, 1000); //Set a timer in (1000=1s)

        }, 1000);

    }, 1000);

    node.warn("Exiting node");
    return;

```

b Store and share nodes

i. Guide

<https://nodered.org/docs/writing-functions#storing-data>

ii. Store data using context object

- A. Context guide <https://nodered.org/docs/user-guide/context>
- B. Data is saved in memory and will not survive across restarts of the system
- C. Context scopes

- context.: between connected nodes
- flow.: on the sheet
- global.: all the nodes and sheets

D. Method

- .set: set a value

- .get: get a value

c Run new packages

- i. Node.js: I can easily bring a new package and use it in my application by using the require keyword.
- ii. In NodeRED: your function code runs in a sandbox which is not allowed to use the required keyword. So how to add a package your function node wants to use to the global system?
 - In Atom, check jnode/bluemix-settings.js
 - In Atom, check jnode/packages.js
 - Push to git
 - On the Note-RED sheet, add a function, edit function node

```
//Add an inject (timestamp)
// Add a function node

var crc = global.get('crc'); // package crc
msg.payload = crc.crc32('hello').toString(16);
return msg;

// Connect to "entire message"
// Get access to the package
```

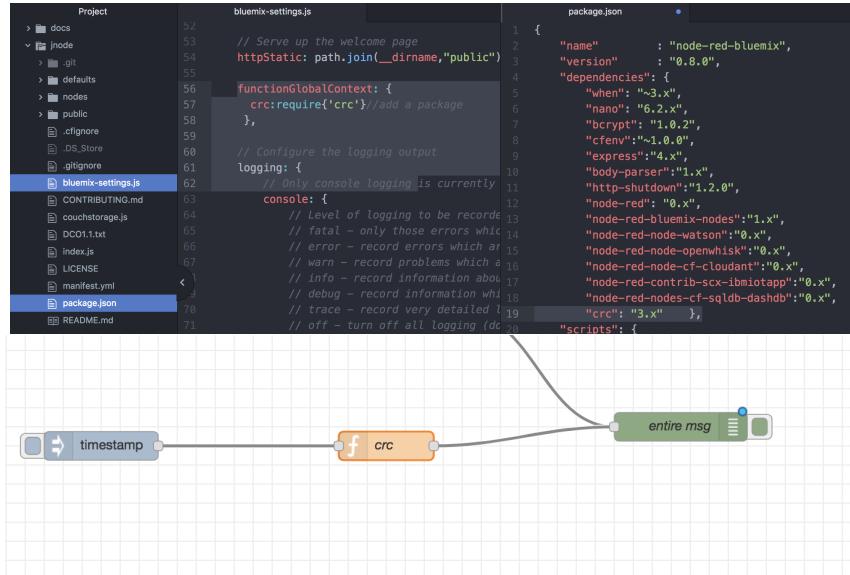


Figure 3: Add a package

2.5 Additional NodeRED nodes

3 Rapid application development on a Raspberry Pi

3.1 Rasberry Pi and SenseHAT

1. A quick look at devices and sensor options
 - a GPIO: general purpose input output
 - b Grovekit
 - c Message Queue Telemetry Transport (MQTT)
 - d Raspberry Pi <https://www.raspberrypi.org>
 - e Sense HAT: the board has sensors for gyroscope, accelerometer, magnetometer, temperature, barometric pressure and humidity.
 - f What do you need?
 - power supply
 - HDMI cable for the monitor
 - keyboard
 - mouse
 - SD card
2. Setting up a Raspberry Pi
 - a Overwrite format microSD card using SD Card formatter (Quick format in the future)
 - b Format to MS_FAT (Mac Disk Utility)
 - c Download Rasbian Stretch 4.14 (with desk top and recommended software)
 - d Install Rasbian .zip file using Etcher <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

```
// Enter RPI shell, CTRL+ALT
// Copy IP address
ifconfig //or
hostname -I
// Enable ssh server
sudo raspi-config
// Select "5 Internet Options"
```

```

Enable ssh, VNC (allow connect to the RPi UI remotely). SPI (sensors), GPIO
(40 pins), 1-Wire interface (low speed data) and camera

// Connect to ssh

ssh pi@ ip address

// Select"Host name" to change a name

// Reboot RPI

sudo reboot -n

sudo rpi-update

sudo reboot -n

// Update Raspbian OS

sudo apt-get update //download

sudo apt-get -y upgrade //install

sudo apt-get -y autoremove //remove installation packages

// Update Node.js and nodered

update-nodejs-and-nodered

// Allow node applications to run bluetooth scans

sudo setcap cap_net_raw+eip $(eval readlink -f `which node`) //Use back

quotes here, not single quotes

// Install a text editor

sudo apt-get install -y geany

//Reboot

sudo reboot -n

```

Check Error FAQ

- e Remote access to RPI from mac (make sure RPI ssh and VNC are enabled and RPI is rebooted) <https://www.raspberrypi.org/documentation/remote-access>
-

```

// In Mac, download VNC viewer, log in to my account

// Install VNC server and password

// In Pi

// Install the VNC server

sudo apt-get install tightvncserver //tightserver //password length

of 8 ideally, otherwise truncated, but typing the same

// Start the VNC server from the terminal

vncserver :1 -geometry 1920x1080 -depth 24 // full HD resolution

// Restart a server at next boot, and every subsequent boot

sudo systemctl enable vncserver-x11-service.service

// Stop VNC Server

```

```

        sudo systemctl stop vncserver-x11-serviced.service

        // Download VNC viewer log in to my account
        // Start vnc server
        vncserver //New desktop is raspberrypi:1 (ipaddress:1)
        // Enter password

        // Remote connect to Mac VNC server

```

3.2 NodeRED on Raspberry Pi part 1

```

// Open NodeRED
node-red-start
// Web brower
127.0.0.1:1880
// Open terminal
cd /home/j/.node-red
vim settings.js //install vim first
// Add crc package
functionGlobalContext: {
    crc:require("crc")
}
// Install package
    sudo npm install crc
// For changes to take effect
    node-red-stop
    node-red-start
// Add a string CRC calculation function in Node-RED sheet
    var crc = global.get('crc');
    msg.payload = crc.crc32('good day').toString(16);
    return msg;
// Add an inject input node, payload ==string
// Add debug output node
// Close node-red window
    CTRL + C //does not stop Node-RED running
node-red-stop // stop Node-RED running

```

Check Error FAQ

```
// In sending binary data across the net, it's often converted to a text format
cd /home/j/.node-red
npm install node-red-node-base64 cd /home/j/.node-red
node-red-stop
node-red-start

//auto start Node-Red at every boot
sudo systemctl enable nodered.service
```

3.3 NodeRED on Raspberry Pi part 2

```
// How to run a system command in the node?

Add a exec functional node

netstat // Command (for network statistics)
-a //Append, here you can add another command or a parameter "-a"
//Use stdout to link to debug node
```

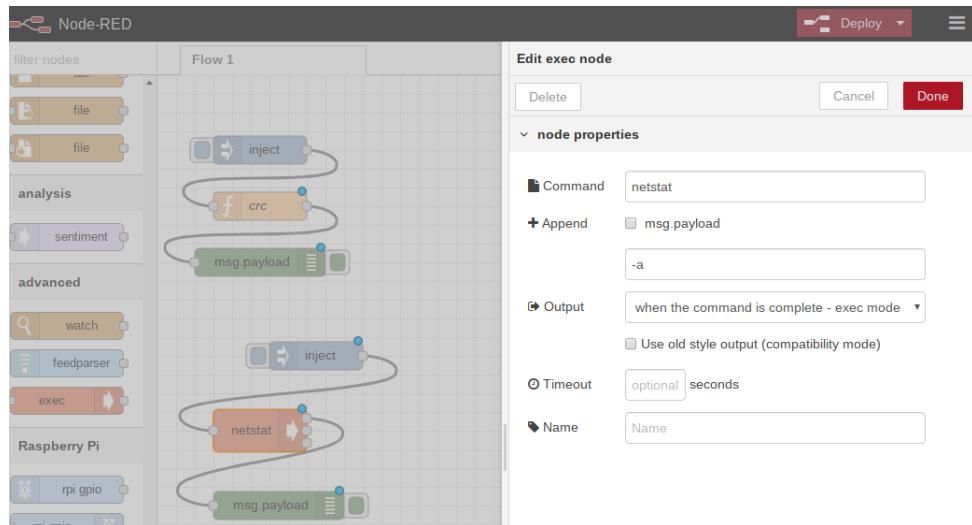


Figure 4: Execute a system command using a node

IBM IoT quick start service

1. Sandbox
 - a No security or pre-registration of device
 - b Allow publishing data on the application and data visualization
2. How to use Output/IBM IoT Device node?

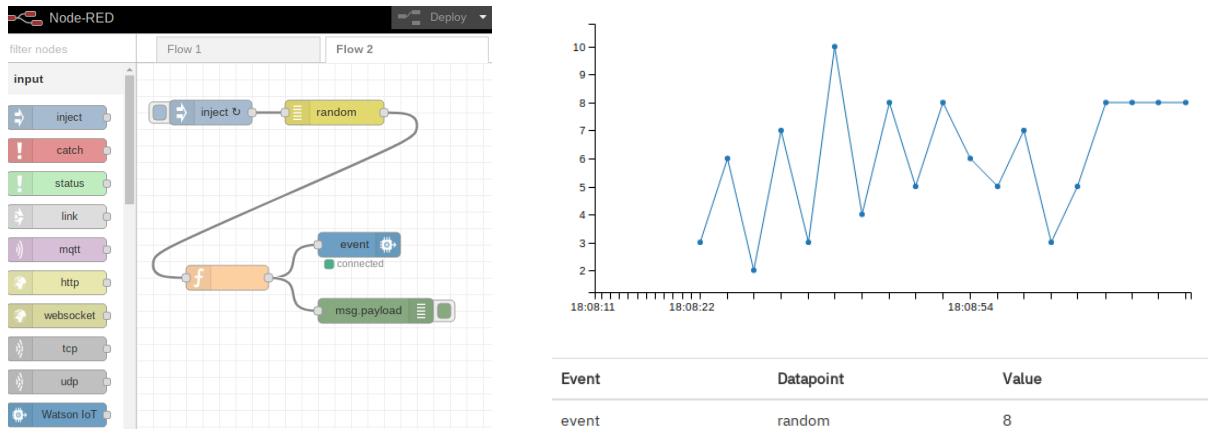


Figure 5: IBM IoT quick start

IoT quick start service

```
// Double click and configure quick startup
// Click quickstart id link //Visualization page using the generated device ID
// Add an Input/inject node
    Payload: string; Repeat: interval, every 3 seconds // Configure to send an empty
        string every 3 seconds
// Add a Function/random node to send the random data to the quick start application

// Add a function node to format the data (for the platform to understand before
    sending the data to
    Quickstart to visualize the data)
    msg.payload = {"d" : {"random" : msg.payload}};
    return msg;
// Add an Output/debug node
```

3.4 Introduction to the Watson IoT Platform

3.5 Controlling a device

4 Lower level programming for the Internet of Things

4.1 Watson IoT APIs

1.

2.

4.2 MQTT

4.3 Deploying Applications to the IBM Cloud

4.4 Controlling a device

5 Error FAQ

Installation

```
sudo rpi-update

//Error: Failed to download update for rpi-update! Make sure you have
      ca-certificates installed and that the time is set correctly
      sudo CURL_CA_BUNDLE=/etc/ssl/certs/ca-certificates.crt rpi-update
      Ref: https://github.com/Hexxeh/rpi-update/issues/206

ssh pi@ipaddress

//Error: can not access RPI from mac using ssh, VNC
// In RPI
Enable ssh, VNC in RPI
// In Mac // not working
cd /Users/j.ssh/
rm known_hosts //https://www.raspberrypi.org/forums/viewtopic.php?t=172423
//Not solved!!

//In RPI: Can not fetch archives
sudo apt-get clean
sudo apt-get update
//In RPI: dpkg interupted
sudo dpkg --configure -a
```

Run Node-RED

```
// In RPI: Click run Node-RED
// Failed to execute child process "xterm" (No such file or directory)
sudo apt install xterm

npm install crc
//npm notice created a lockfile as package-lock.json and audited 4 packages
// use sudo npm install crc
```
