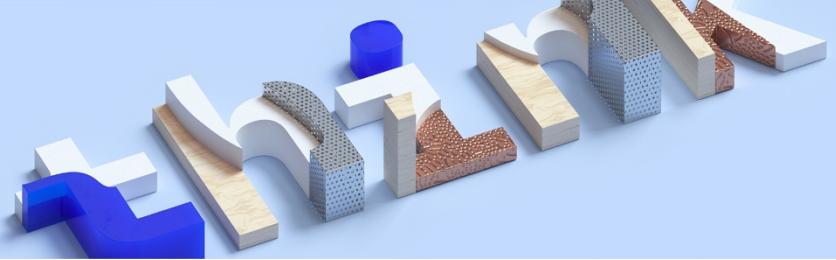


think 2018

IBM



Lab Center – Hands-on Lab

Session 4104

Session Title Beam Me Up Watson

Dave Locke, IBM

John Walicki, IBM

Ernesto Manuel Cantone, ST

Table of Contents

Disclaimer.....	4
Overview of the Lab.....	6
Components used in the Lab :-.....	6
Pre-reqs.....	7
ST	8
Section 1 – Login and prepare your IBM Cloud account.....	8
Section 2 – Create an Internet of Things Starter App.....	11
Step 1 – Create an IoT Starter Application	11
Step 2 - Launch the IoT Starter Application	12
Step 3 – Open the Node-RED visual programming editor	13
Section 3 – Connect the BlueCoin to WIoTP	15
Step 1 – IBM Cloud dashboard	16
Step 2 – Register the BlueCoin device with the Watson IoT Platform	16
Step 2.1 Launch the WIOTP dashboard	16
Step 2.2 Define a Device Type for the BlueCoin	18
Step 2.3 Register an instance of the BlueCoin.....	19
Step 2.4 Note the org id for WIOTP.....	22
Step 3 – Configure the ST BlueMS mobile application to send data to WIoTP	22
Step 4 – View the raw data in WIOTP dashboard	23
Section 4 – Create a Hello World app in IBM Cloud	26
Step 4.1 Node-Red intro	26
Step 4.2 Build the HelloWorld app	26
Step 4.3 Test the HelloWorld app.....	27
Step 4.4 Modify the HelloWorld app.....	27
Step 4.5 Delete the HelloWorld app	28
Completed HelloWorld app.....	28
Section 5 – Create a Dashboard to Visualise BlueCoin Data.....	28
Step 5.1 Receive the compass data from the BlueCoin in Node-Red.....	29
Step 5.2 Install the Node-Red dashboard nodes	32
Step 5.2 Create a simple dashboard	32
Step 5.3 Change the frequency sensor events are sent to the cloud	34
Step 5.3 Add direction logic to the application	35

Completed Compass Dashboard app.....	36
Rich Example Dashboard application	37
Section 6 Control a Game using the BlueCoin.....	37
Section 7 Transcribe Speech from BlueCoin to Text.....	38
Step 7.1 Configure the IBM Cloud Speech to Text Service.....	38
Step 7.2 Configure the BlueMS mobile app to talk to the cloud S2T service.....	40
Step 7.3 Test the Speech to Text	40
Section 8 Using Voice in the Cloud Application	40
Step 8.1 Bind the Speech to Text service to the Cloud Application	40
Step 8.2 Setup the Speech to Text Proxy	41
Step 8.3 Change BlueMS mobile app to point at the Speech to Text Proxy	42
Section 9 Use voice and sensor to post a slack message.....	42
Step 9.1 Install a Slack Node-Red node	43
Section 10 Summary.....	43
We Value Your Feedback!.....	44

Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results like those stated here.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

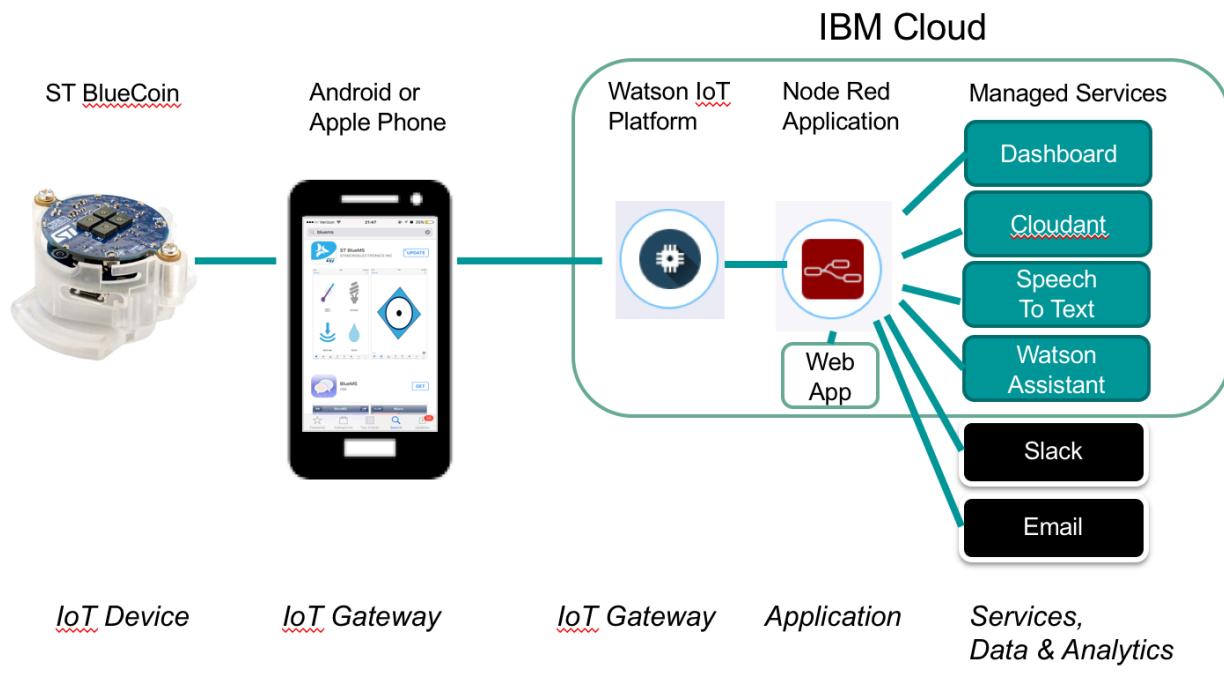
© 2018 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Overview of the Lab

This lab, jointly presented by IBM and STMicroelectronics, combines a ST BlueCoin IoT starter kit, a mobile app and a set of IBM Cloud services to recreate a Star Trek lapel communicator. Following the lab you will learn how to connect the BlueCoin, send sensor data and voice, and build an IBM Cloud app using a number of services including Watson IoT Platform, Node-Red and Watson Assistant to consume and interact with the sensor and voice data.

The diagram below shows how the components of the Lab link together. What is built in the lab is an exemplar of a typical IoT solution. The lab provides a solid grounding in the principles of IoT and once completed will enable richer and more diverse projects to be undertaken.



Components used in the Lab :-

- ST BlueCoin Starter Kit
 - The device that collects sensor data such as motion and voice
 - It communicates with a smart phone via the Bluetooth protocol
- Smart Phone
 - The phone has several roles
 - It acts as a IoT Gateway enabling data from the BlueCoin to be exchanged with the IBM Cloud
 - It runs a mobile application that interacts with the BlueCoin, enabling the data collected by the bluecoin to be visualization and optionally send to the cloud.

- IBM Cloud
 - o At a high level it provides the capability to
 - Host and run applications that can be written in a wide variety of programming languages and hosted in different runtime environments including containers.
 - Provides a catalog of “managed services” that an application can make use of. The services are split into categories includines IoT, Data, Analytics, DevOps and Watson.
- Watson IoT Platform
 - o A managed service provided in the IBM Cloud
 - o Acts as a hub for connecting and managing devices
 - o Enables data from devices to be handed to upstream applications and services and for applications and services to send data and commands to devices.
- Node Red
 - o An application development tool and runtime offered in the IBM Cloud
 - o Enables applications to built using visual wiring combined with drag and drop metaphor. In some cases applications can be created without writing a single line of code.
 - o Node Red also runs in many other environments including laptops and Raspberry Pis.
- Dashboard
 - o A Web dashboard will be created as part of the lab enabling visualization of device and application data
- Cloudant
 - o A document-based data store provided as a managed Service in the IBM Cloud
 - o Can be used to store device and application data
 - o Is used by Node-Red to store configuration information
- Speech to Text
 - o An IBM Cloud service that converts speech to text
 - o Used to transcribe the voice data collected by the BlueCoin to Text
- Watson Assistant
 - o Enables the creation of “assistant” type applications using a set of skills, There are a set of default skills such as weather and time plus the capability to extend with new skills.
 - o Provided as a managed service in IBM Cloud
 - o Transcribed voice will be used to interact with Watson Assistant
- Slack
 - o An external service that the lab will interact with to post messages to a slack channel
- Email
 - o An external service that the lab will interact with to send an email when an event occurs
- Web Application
 - o A pre-built web application in the form of a game
 - o The lab will extend the game to use the BlueCoin to control the game

Pre-reqs

- IBM Cloud ID
 - o To run the lab an IBM Cloud ID is required.

- For the Think 2018 conference hands on labs an ID will be allocated to each attendee.
- Outside of the Think 2018 conference a user can register for their own IBM Cloud ID or use an enterprise account.
- A Lite account that provides free access to IBM Cloud can be created here <https://console.bluemix.net/>
- The majority of the lab will work with a Lite account. There are some parts of the workshop where an account with 512Mb of memory is required, a Lite account is restricted to 256Mb meaning certain parts of the lab cannot be run
-
- Android or Apple smart phone
 - The BlueCoin starter kit connects to a smart phone which in turn connects to the IBM Cloud. The lab requires an Android or Apple smart phone capable of running the ST BlueMS mobile application is available.
 - The ST BlueMS mobile application is available on both the Google and Apple app stores

ST

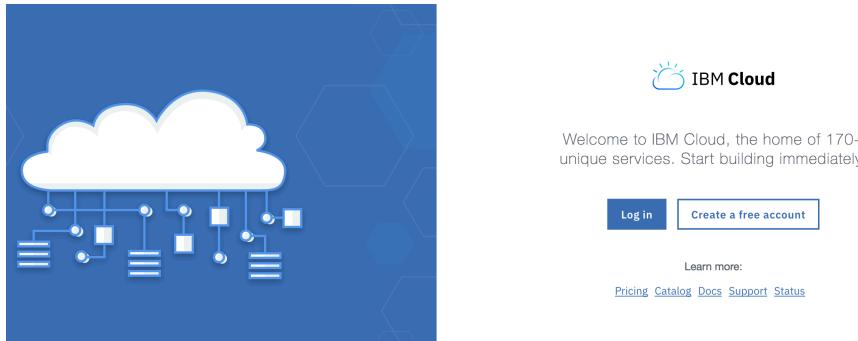
The ST parts of the lab are provided in a separate document that covers;–

- About
- Bluecoin device / HW intro
- Device to Phone Connectivity
- Smart Phone App
- Phone to WLoTP Quickstart (sensor data to cloud)

In this Section, we will create a free trial account on Bluemix.

Section 1 – Login and prepare your IBM Cloud account

- In a new browser tab, goto <http://bluemix.net> and click “Log in”
 - If you do not already have an account, click on Create a free account and follow the instructions
 - For info, you may spot the term Bluemix in a few places, this is the old name of what is now IBM Cloud



- Log in to IBM Cloud with your IBMId and password

Log into IBM Bluemix

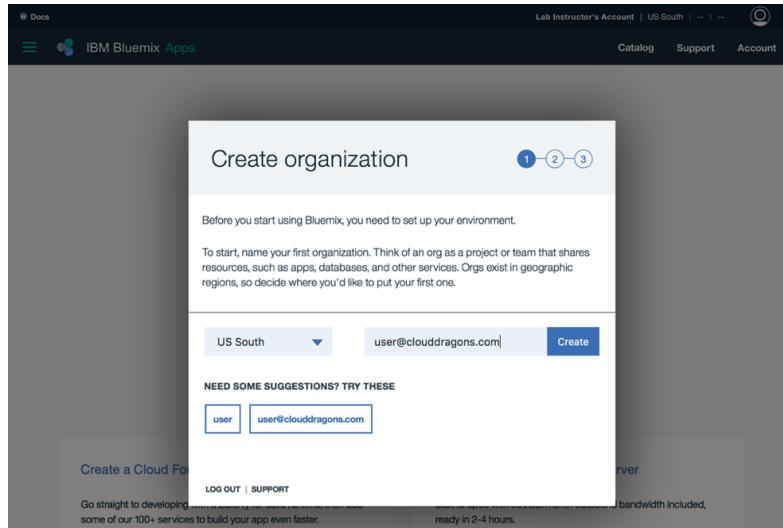
IBMid: user@clouddragons.com

Password	Forgot your password?

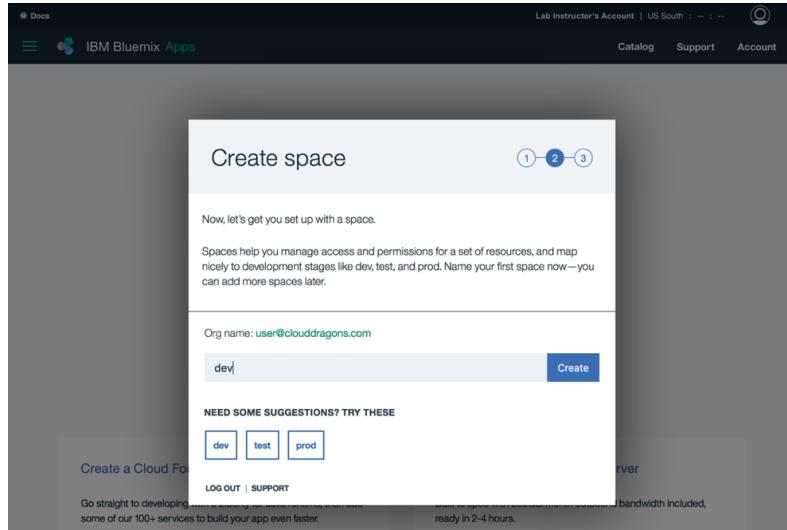
Log in	

[Use a different IBMid or email](#)

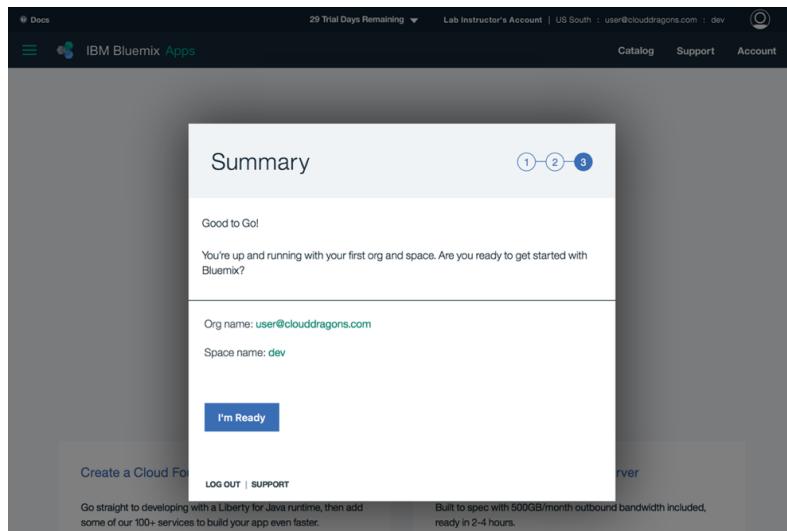
- Create an organization with the same name as your email (IBMid)
- TODO – org and space created by default? Change words with a brief description



- Create a space.
 - A space is typically used to manage and provide separation of development, test and production



- Click "I'm Ready"



Section 2 – Create an Internet of Things Starter App

In this section we will create both an instance of the Watson IoT Platform (WIoTP) service and a Node Red application development and runtime environment. WIoTP is a hub for connecting devices and exchanging data with the devices. Node Red is what we will use to develop our application that will be hosted and run in the IBM Cloud.

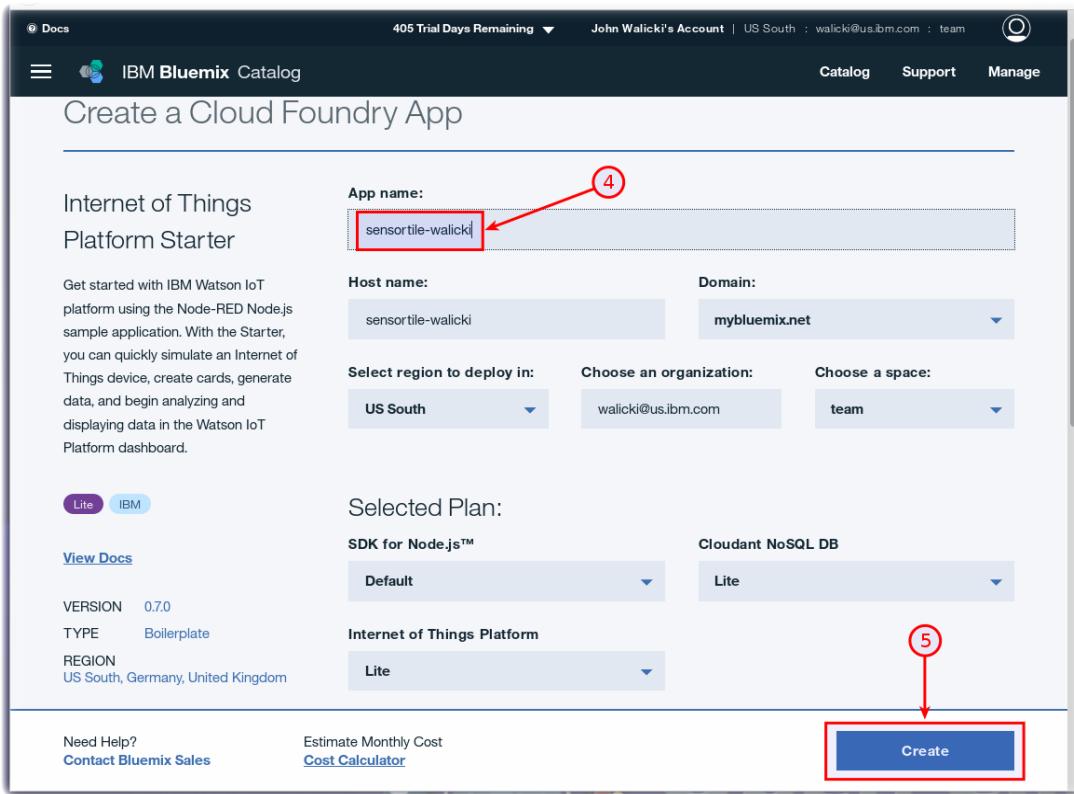
Step 1 – Create an IoT Starter Application

We will use a Boilerplate to speed up the process of creating instances of WIoTP and Node Red. The boilerplate performs multiple actions that can if needed be performed individually.

- Click on the **Catalog** (1) and search for '**internet of things**' (2)
- The **Internet of Things Platform Starter** (3) boilerplate is a pattern with pre-assembled services that work together. The Internet of Things Platform Starter includes a Node-RED Node.js web server, Cloudant database to store the sensor data, and the IoT platform service so you can connect devices. Select this boiler plate

The screenshot shows the IBM Bluemix Catalog interface. At the top, there is a navigation bar with 'Docs', '405 Trial Days Remaining', 'John Walicki's Account | US South : walicki@us.ibm.com : team', and user profile icons. Below the navigation bar, the main header says 'IBM Bluemix Catalog'. On the left, there is a sidebar with categories: 'All Categories (22)', 'Infrastructure' (Compute, Storage, Network, Security, Containers, VMware), and 'Platform (22)' (Boilerplates (1), APIs, Application Services (5)). The main content area has a search bar with the text 'internet of things' (2). Below the search bar, under the 'Platform' category, there is a section for 'Boilerplates' with the heading 'Get started with a new app, now.' and a card for the 'Internet of Things Platform Starter' (3). The card includes a description: 'Get started with IBM Watson IoT platform using the Node-RED Node.', a 'Lite' button, and an 'IBM' button. A red arrow points from the number (1) to the 'Catalog' link in the navigation bar. Another red arrow points from the number (2) to the search bar. A third red arrow points from the number (3) to the 'Internet of Things Platform Starter' card.

- Give your application a name that is unique (4) e.g. bluecoin-yourinitials. If you choose **myapp**, your application will be located at <http://myapp.mybluemix.net>. There can only be one “**myapp**” application and URL registered in IBM Cloud



- Press the **Create** button (5).
- IBM Cloud will create an application in your account based on the services in the boilerplate. This is called staging an application. It can take a few minutes for this process to complete. While you wait, you can click on the **Logs** tab and see activity logs from the platform and Node.js runtime.

Step 2 - Launch the IoT Starter Application

- Once the Green “**Running**” icon appears, Click the **Visit App URL** link (6).

Cloud Foundry apps / sensortile-walicki

sensortile-walicki Running Visit App URL

6

Getting started with Watson IoT Platform Starter

Last Updated: 2017-01-30 | [Edit in GitHub](#)

Get started with IBM Watson™ IoT Platform by using the Watson IoT Platform Starter boilerplate. By using the Starter, you can quickly simulate a device, create cards, generate data, and begin analyzing and displaying data in the Watson IoT Platform dashboard.

The Starter automatically deploys and connects these services:

- Watson IoT Platform - An IoT toolkit that includes gateway management, device management, and application access. By using Watson IoT Platform, you can collect connected device data and run analytics on real-time data from your organization.
- IBM® SDK for Node.js for Bluemix® - The runtime environment in which Node-RED runs.
- IBM® Cloudant® NoSQL DB for Bluemix® - The database in which Node-RED stores metadata.
- Node-RED application - An instance of the Node-RED application that includes nodes designed for Watson IoT Platform.

Step 3 – Open the Node-RED visual programming editor

- A new browser tab will open to the Node-RED start page. Node-RED is an open-source Node.js application that provides a visual programming editor that makes it easy to wire together flows. Select a username / password to access the Node-RED editor. **Remember** your username / password. Click the red button **Go to your Node-RED flow editor** to launch the editor.

Welcome to your Internet of Things Platform (IoTP) boilerplate application on IBM Bluemix

This sample application uses Node RED to help demonstrate the wonderful things you can do with your IoTP service. We know you're eager to check it out, but first there is something important to do:

- Secure your Node-RED editor

Secure your Node-RED editor

Secure your editor so only authorised users can access it

Username

Password Must be at least 8 characters

Allow anyone to view the editor, but not make any changes

Not recommended: Allow anyone to access the editor and make changes

Finish the configuration

You have made the following selections:

- Secure your editor so only authorised users can access it

The settings will be persisted in the CloudantDB bound to this application. You can override them at any time by setting the following environment variables via the Bluemix console:

- NODE_RED_USERNAME - the username
- NODE_RED_PASSWORD - the password
- NODE_RED_GUEST_ACCESS - if set to 'true', allows anyone read-only access to the editor

Previous Finish

Node-RED
Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

The version running here has been customized for the IBM Watson IoT Platform.

More information about Node-RED, including documentation, can be found at nodered.org.

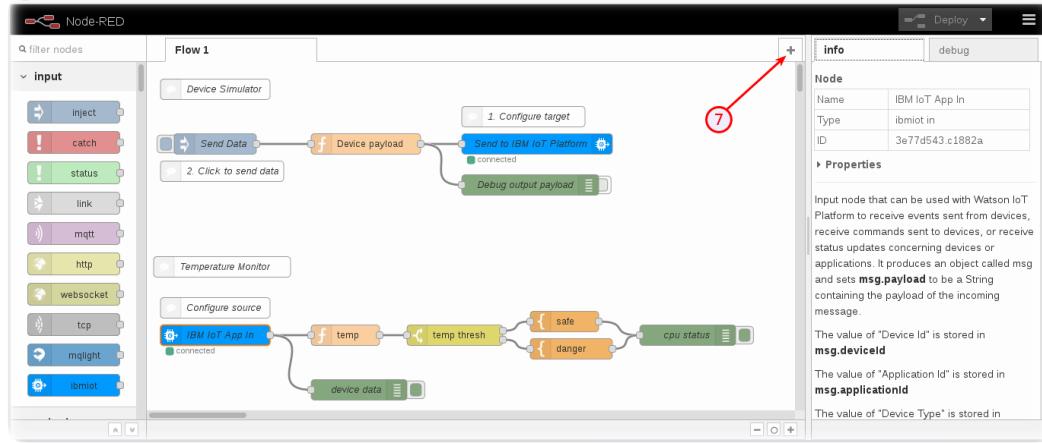
[Go to your Node-RED flow editor](#)

[Learn how to customise Node-RED](#)

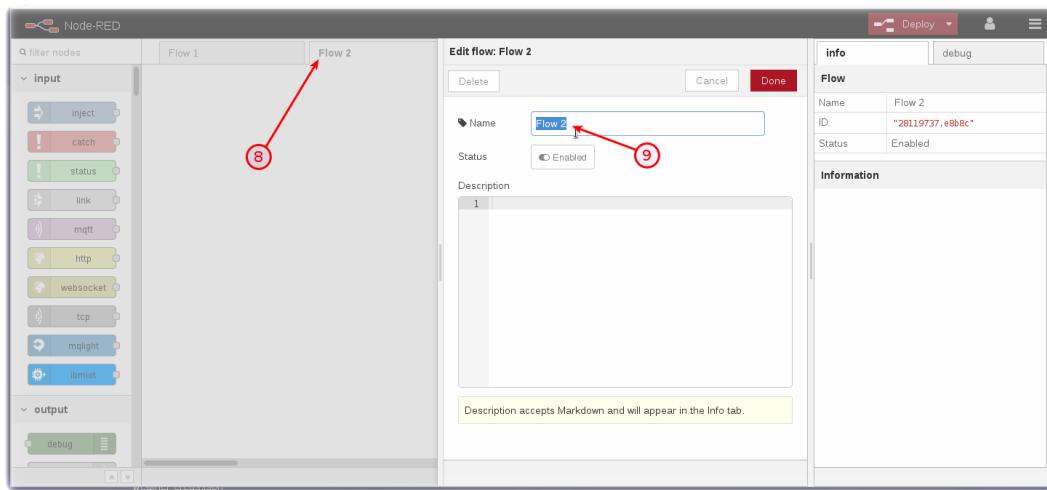
Customising your instance of Node-RED

This instance of Node-RED is enough to get you started creating flows.
<https://sensortile-walicki.mybluemix.net/red/>

- The Node-RED Visual Programming Editor will open with a default flow.
- On the left side is a palette of nodes that you can drag onto the flow.
- You can wire nodes together to create a program.
- There is a sample IoT Starter flow (application) that receives data from a device via WIoTP, extracts the temperature from the message payload, and based on the temperature decides if it is a safe or dangerous temperature. This flow is not applicable to this workshop and can optionally be deleted



- A Node-Red application can be split into multiple parts where each part is constructed in its own **Tab**. Click the **+** icon (7) to add a new tab. Click on the **Flow 2** tab header (8).
- Rename this tab from **Flow 2** to **BlueCoin** (9)



- The BlueCoin tab will be used later in the Lab to build an application

Section 3 – Connect the BlueCoin to WIoTP

Earlier in the lab the BlueCoin was connected to IBM Quickstart and the sensor data from the BlueCoin visualized via a web page. IBM Quickstart is a special instance of WIoTP. In this section the BlueCoin

will be connected to the instance of WIoTP that was created in the previous section and the data visualized in the WIoTP dashboard.

Step 1 – IBM Cloud dashboard

The IBM Cloud dashboard is the user interface through which all of resources can be reached.

- Click on IBM Cloud (1) in the top left of the dashboard of the IBM Cloud page to access the dashboard

The screenshot shows the IBM Cloud dashboard. At the top, there's a navigation bar with 'IBM Cloud' (1), 'Catalog', 'Docs', 'Support', 'Manage', and a profile icon. Below the navigation bar, the dashboard header includes 'Dashboard', 'REGION US South', 'CLOUD FOUNDRY ORG instructor0300@i...arning.org', 'CLOUD FOUNDRY SPACE dev', and a 'Create resource' button. The main content area has two sections: 'Cloud Foundry Apps' and 'Cloud Foundry Services'. The 'Cloud Foundry Apps' section lists one application: 'bluecoin-djl' with a route 'bluecoin-djl.mybluemix...'. The 'Cloud Foundry Services' section lists two services: 'bluecoin-djl-cloudantNoSQLDB' (3) using 'Cloudant NoSQL DB' and 'bluecoin-djl-iotf-service' (4) using 'Internet of Things Platform'. Arrows from the numbered circles point to the corresponding items in the list.

- The dashboard shows the applications and services that were created by the Internet of Things Starter boilerplate. There will be :
 - One Cloud Foundry Application (2)
 - Two services, a cloudant document based data store (3) that is used by node-red to store its configuration data and can optionally be used by the application to store data such as the sensor data from the Bluecoin. An instance of the Internet of Things Platform (4)

Step 2 – Register the BlueCoin device with the Watson IoT Platform

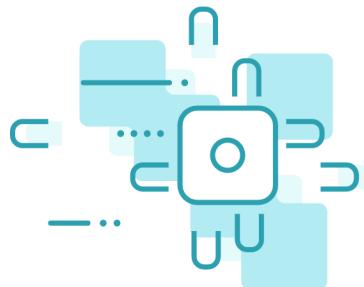
The Watson IoT Platform (WIOTP) dashboard is the user interface for interacting with all resources related to the WIoTP service. The dashboard will be used to register the BlueCoin with the platform.

Step 2.1 Launch the WIOTP dashboard

- Head to the WIOTP dashboard by selecting the WIOTP instance from the IBM Cloud dashboard (4). This brings up the launch page for the WIOTP dashboard

Manage
[Plan](#)
[Connections](#)
[Internet of Things](#) /

 bluecoin-djl-iotf-service

[Location: US South](#)
[Org: instructor0300@ibmlearning.org](#)
[Space: dev](#)


Let's get started with Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

[Launch](#)
[Docs](#)


- Select **Launch** (1)
- Now head to the **Devices** page of the dashboard by selecting the **Devices** icon (1)

IBM Watson IoT Platform [QUICKSTART](#) [SERVICE STATUS](#) [DOCUMENTATION](#) [BLOG](#)

[instructor0300@ibm... ▾
ID: \(sqpbmj\)](#)

Your boards Public boards + Create New Board

Your boards Sort By: Recently changed

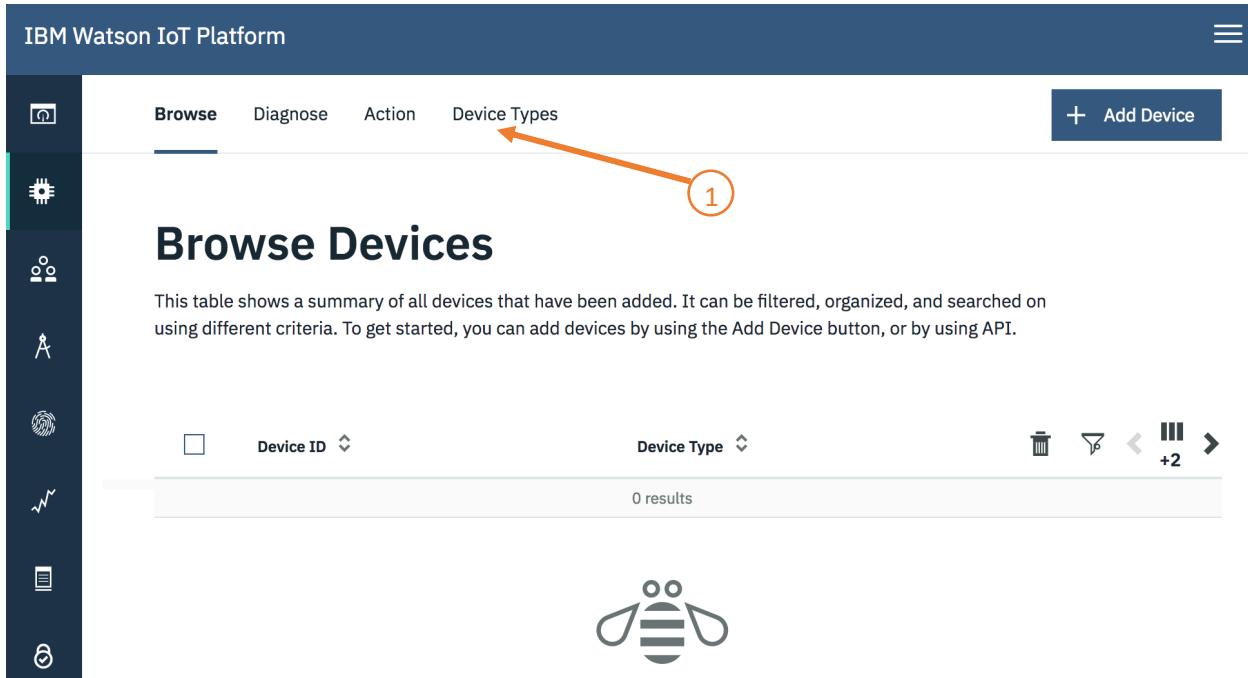
 1

Boards shared with you

 1

Step 2.2 Define a Device Type for the BlueCoin

- There are no devices currently registered. A device must be classified by type before instances of the device can be registered. Select **Device Types** (1)



The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'IBM Watson IoT Platform', 'Browse' (which is selected), 'Diagnose', 'Action', 'Device Types' (with an orange arrow pointing to it and a red circle containing the number '1'), and '+ Add Device'. On the left, there's a vertical sidebar with icons for Home, Devices, Diagnose, Actions, Device Types, Metrics, and Events. The main content area is titled 'Browse Devices' and contains a summary message: 'This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.' Below this is a table header with columns for 'Device ID' and 'Device Type', and a row of controls including a checkbox, a trash icon, a search icon, and navigation arrows. The table body displays '0 results'. At the bottom center is a small icon of a bee.

- Select **+ Device Types** in the top right of the dashboard

Device types group devices that have similar characteristics, such as model number, firmware version, or location. Give the device type a unique name and a description that identifies characteristics that are shared by devices of this type.

Type	Device	Or	Gateway
Name	bluecoin	The device type name is used to identify the device type uniquely and uses a restricted set of characters to make it suitable for API use. 1	
Description	STMicroelectronics starter kits		

- Give the Device Type a name (1) such as *bluecoin* and optionally a description
- Select **Next** (2) and on the next page optionally fill in information related to the BlueCoin and then select **Done**.

Step 2.3 Register an instance of the BlueCoin

- Now the Device Type has been created one or more BlueCoins can be registered to WIoTP. Registering a device to WIoTP enables the device to be securely connected and for additional meta information and interfaces to be defined for use in managing the device and for upstream applications to make use of

The screenshot shows the IBM Watson IoT Platform interface. At the top, there are links for QUICKSTART, SERVICE STATUS, DOCUMENTATION, and BLOG. On the right, the user's email (instructor0300@ibm...) and ID (sqpbmj) are displayed. Below this, a navigation bar includes options like Browse, Diagnose, Action, and Device Types, with 'Device Types' being the active tab. A blue button labeled '+ Add Device Type' is visible. The main content area displays a message: 'You added the new device type: bluecoin'. It includes two tabs: 'Register Device' (selected) and 'Advanced Flow'. Below these tabs, there is optional information: 'Register Devices, Define Interfaces'. A note states: 'Now that you added a device type, you can register and connect devices for this type.' A large central icon depicts a microchip or gear. At the bottom of the page are 'Cancel' and 'Next' buttons.

- Select **Register Devices (2)** to open the dialog to register your device

The screenshot shows the 'Add Device' dialog box. The tabs at the top are 'Identity' (selected), 'Device Information', 'Security', and 'Summary'. The 'Identity' tab contains instructions: 'Select a device type for the device that you are adding and give the device a unique ID.' It has two input fields: 'Select Existing Device Type' (set to 'bluecoin') and 'Device ID' (set to 'bluecoindjl'). A 'Next' button is located at the bottom right. Orange arrows and circles highlight the following steps: (1) points to the 'bluecoin' selection in the dropdown; (2) points to the 'bluecoindjl' text in the 'Device ID' input field; and (3) points to the 'Next' button.

- Select the device type you just created (1)
- Every device registered to an instance of WIoTP must have a unique name. For instance a serial number or a MAC address make good unique identifiers. Fill in the **Device ID (2)**, for the lab a name like bluecoin<yourinitials> will work but in real world solutions ensuring a device has a unique identity is important.

- Select **Next (3)**
- On the **Device Information** window optionally fill in information about the BlueCoin then select **Next** to move to the Device Security window

Browse Diagnose Action Device Types

Device Security

There are two options for selecting a device authentication token.

Auto-generated authentication token (default)

Allow the service to generate an authentication token for you. Tokens are 18 characters and contain a mix of alphanumeric characters and symbols. The token is returned to you at the end of the device registration process.

Self-provided authentication token

Provide your own authentication token for this device. The token must be between 8 and 36 characters and contain a mix lowercase and uppercase letters, numbers, and symbols, which can include hyphens, underscores, and periods. Do not use repeated characters, dictionary words, user names, or other predefined sequences.

Authentication Token ⓘ

Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored. 1

Authentication token are encrypted before we store them.

◀ Next 2

- Ensuring a device connects securely is important. It is recommended that devices connect using TLS to provide secure network connection between the device and WIoTP. The device needs to provide some credentials to WIoTP to enable WIoTP to identify it and to trust it. A Certificate can be used for this purpose or alternatively a Token can be used.
- If the **Authentication Token (1)** field is left blank a token will be generated. For the lab fill in a token that is memorable. In either case ensure you remember the token as 1) it will be needed later 2) it is non-recoverable once the registration process is completed.
- Select **Next (2)**

Add Device Identity Device Information Security **Summary** X

Summary

Verify that the following information is correct then select Done

Device Type
bluecoin

Device ID
bluecoindjl

View Metadata

Security Token
password

Done

- A summary screen is presented.
- Select **Done (1)** to complete registration of the BlueCoin

Step 2.4 Note the org id for WIoTP

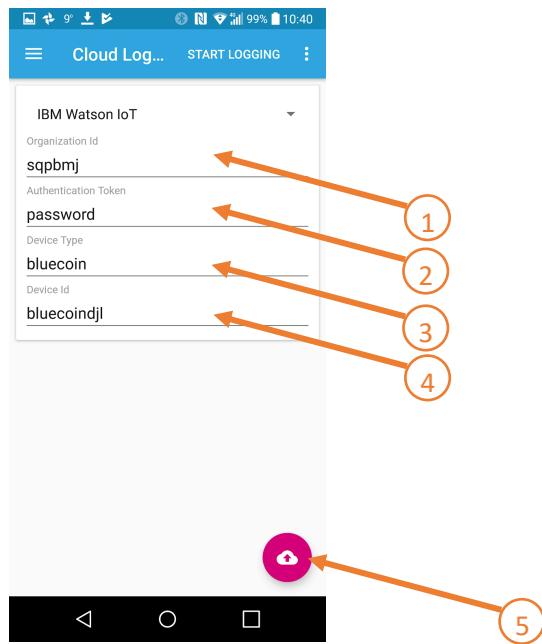
Every instance of WIoTP has a unique identity known as the organization or org id. When connecting a device, the org id will be required.

- One place to find the org id is to look at the url for the WIoTO dashboard . It will look similar to :-
<https://sqpbmj.internetofthings.ibmcloud.com/dashboard/#/devices/browse-v2?add=bluecoin>
- The first part or the URL in this case **sqpbmj** is the org id. Take a note of your orgid for use later in the lab.

Step 3 – Configure the ST BlueMS mobile application to send data to WIoTP

The ST BlueMS mobile application has built in capability to connect and interact with WIoTP

- Open the ST BlueMS Mobile Application, connect to your BlueCoin and turn to the **Cloud Logging** menu. Choose the **IBM Watson IoT** option from the dropdown.



- Enter the **Organization Id** (1) from step 2.4
- Enter the **Authentication token** (2) from step 2.3
- Enter the **Device Type** (3) from step 2.2
- Enter the **Device ID** (4) from step 2.3
- The mobile application is now ready to receive data from the BlueCoin and send it to WIoTP. To start sending data select the cloud icon (5)
- This brings up a screen providing a choice of which data is to be sent to the cloud.

-
- Let's start by selecting **Compass** (2) Once selected the app starts sending the data to WIoTP

Step 4 – View the raw data in WIoTP dashboard

Once data is being sent to WIoTP it can be viewed in a few ways including: -

1. Raw data in the WIoTP dashboard
2. Create an administration dashboard with UI widgets inside of WIOTP
3. Build an application to view the data outside of WIOTP

In the lab we will start with 1. and move on to 3. Creating an administrative dashboard is quick and fun and left as an exercise for after the lab is completed.

- Head back to the WIoTP dashboard and select the devices view (1)
- Then select your BlueCoin device

Identity	Device Information	Recent Events	State	Logs
Device ID	bluecoindjl			
Device Type	bluecoin			
Date Added	4 Mar 2018 10:19			
Added By	instructor0300@ibmlearning.org			
Connection Status	Disconnected	Last Connected: 4 Mar 2018 10:46 Client Address: 146.199.129.199 (secureToken) Duration: 2 minutes Data Transferred: 4.8 KB		

- A great deal of information can be seen in the deices view. The front panel (2) shows the current connection status and when it last connected and if it connected securely
- The **Logs** view (3) shows a list of log messages showing when the device connected along with errors.
- Select the **Recent Events (4)** view. Here we can see the raw data as it arrives from the device.

bluecoindjl	bluecoin	Device		
Identity	Device Information	Recent Events	State	Logs
Showing Raw Data The recent events listed show the live stream of data that is coming and going from this device.				
Event	Value	Format	Last Received	
Compass	{"d":{"timestamp":17032,"Angle":14...}	json	a few seconds ago	
Compass	{"d":{"timestamp":16395,"Angle":14...}	json	a few seconds ago	①
Compass	{"d":{"timestamp":15757,"Angle":14...}	json	a few seconds ago	
Compass	{"d":{"timestamp":15120,"Angle":14...}	json	a few seconds ago	
Compass	{"d":{"timestamp":14482,"Angle":14...}	json	a few seconds ago	

- As an event arrives it is displayed as the first item in the list of events. Note :- this view only displays events when a device is connected and is actively sending in data.
- Select one of the events (1) to see the payload of the event. For a Compass event the time the reading was taken and the Angle can be seen.

Device ID: bluecoindjl

Showing Raw Data | The recent events listed show the live stream of data that is coming and going from this device.

Event Name: Compass

Time Received: 4 Mar 2018 11:03

```

1 <pre>{"d": {
2   "timestamp": 24670,
3   "Angle": 139.3000030517578
4 }
5 </pre>
6

```

Event: Compass

Event: Compass

Event: Compass

Event: Compass

Event: Compass

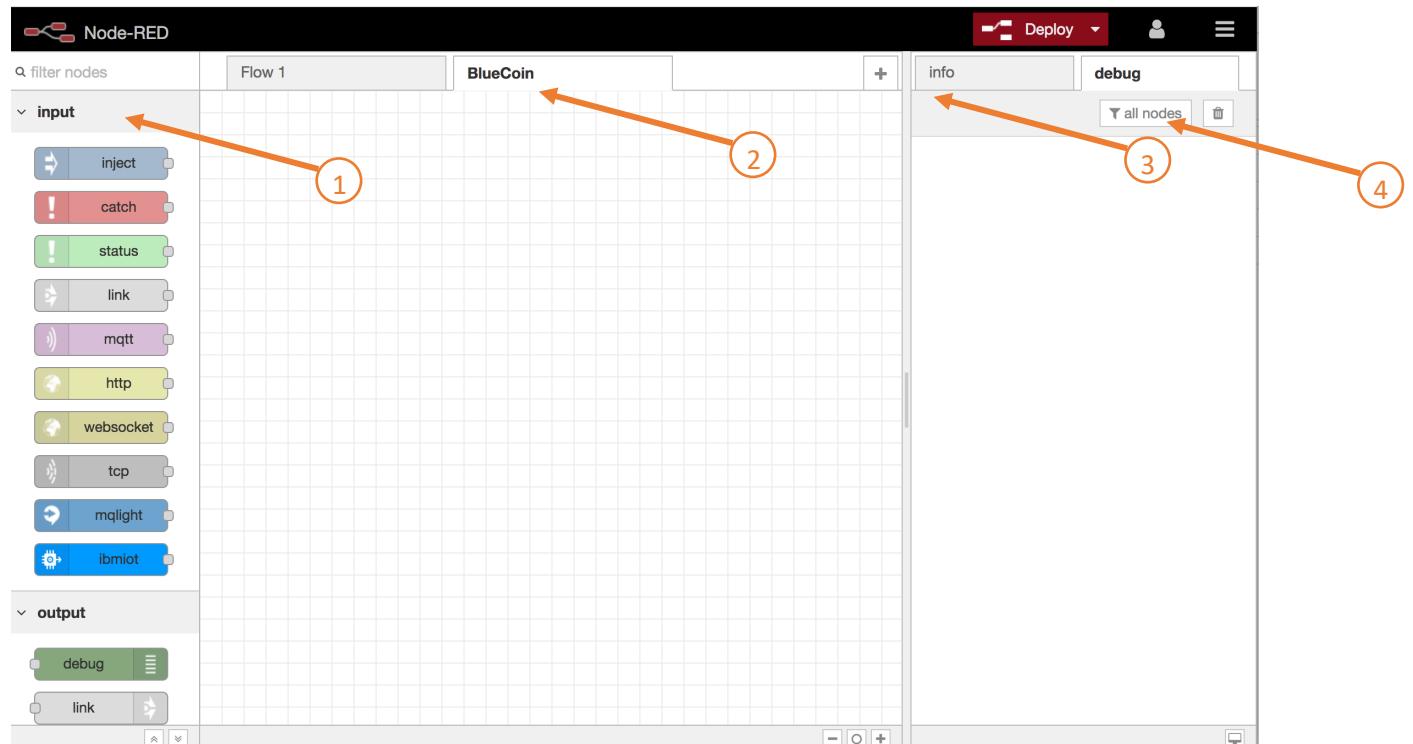
Event: Compass

Section 4 – Create a Hello World app in IBM Cloud

Before building an IoT app we will get use to the Node-Red development environment and build a simple hello world application.

Step 4.1 Node-Red intro

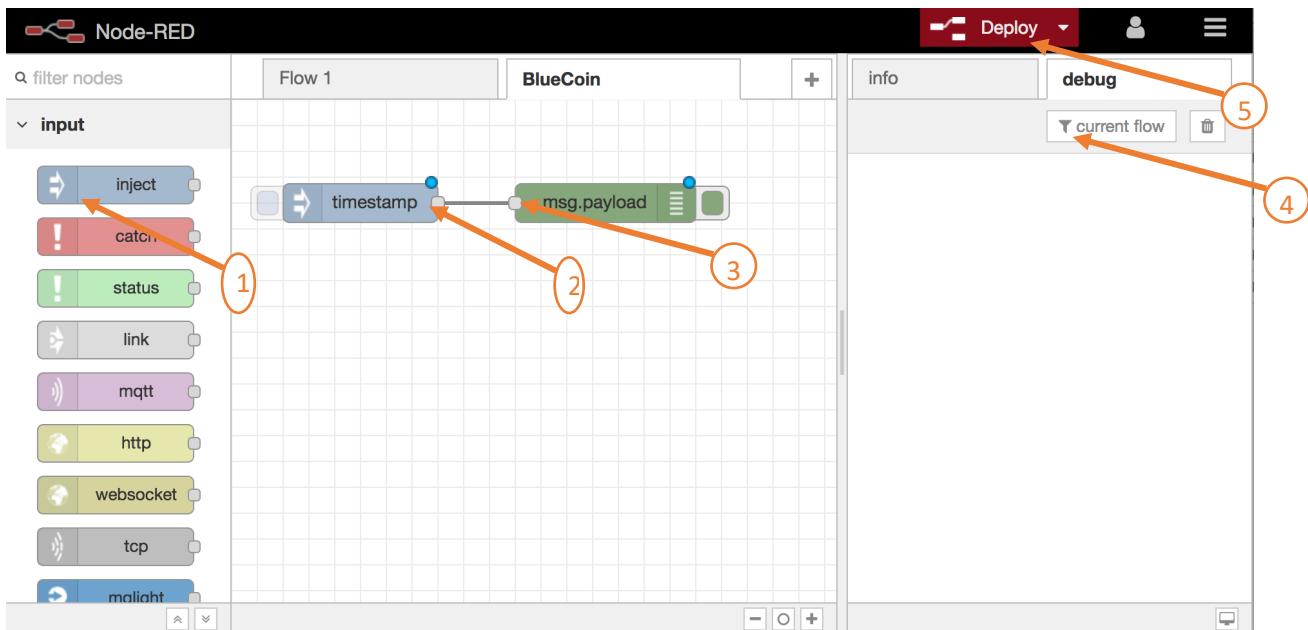
- Head back to your Node-Red development environment from Section 2, Step 3



- On the left-hand side (1) are a palette of Nodes that can be used to build an application. The nodes are categorized into groups such as input, output, function and storage. Node-red and the nodes are open source. Node-red comes with a default set of nodes. There are many additional nodes that have been contributed by IBM and the community that can be installed. A little later in the lab we will install some new nodes.
- We will build our HelloWorld app in the BlueCoin tab (flow) created earlier in the lab.
- When a Node is selected information about the node can be seen in the info tab (3)
- When an application has been deployed, the debug tab (4) will show any debug output.

Step 4.2 Build the HelloWorld app

The HelloWorld app will generate a timestamp message and display it in the debug tab



- Drag an **Inject** (1) node from the Input palette onto the BlueCoin flow.
- Drag an **Debug** node from the Output palette onto the BlueCoin flow.
- Wire the output node of the Inject node (2) to the input node of the Debug node (3)
- Select the Debug tab and change All nodes to **current flow** (4). This restricts the debug view to only showing output from the current flow, in this case BlueCoin
- The application is not ready to deploy. Select **Deploy** (5) and the HelloWorld application is made live

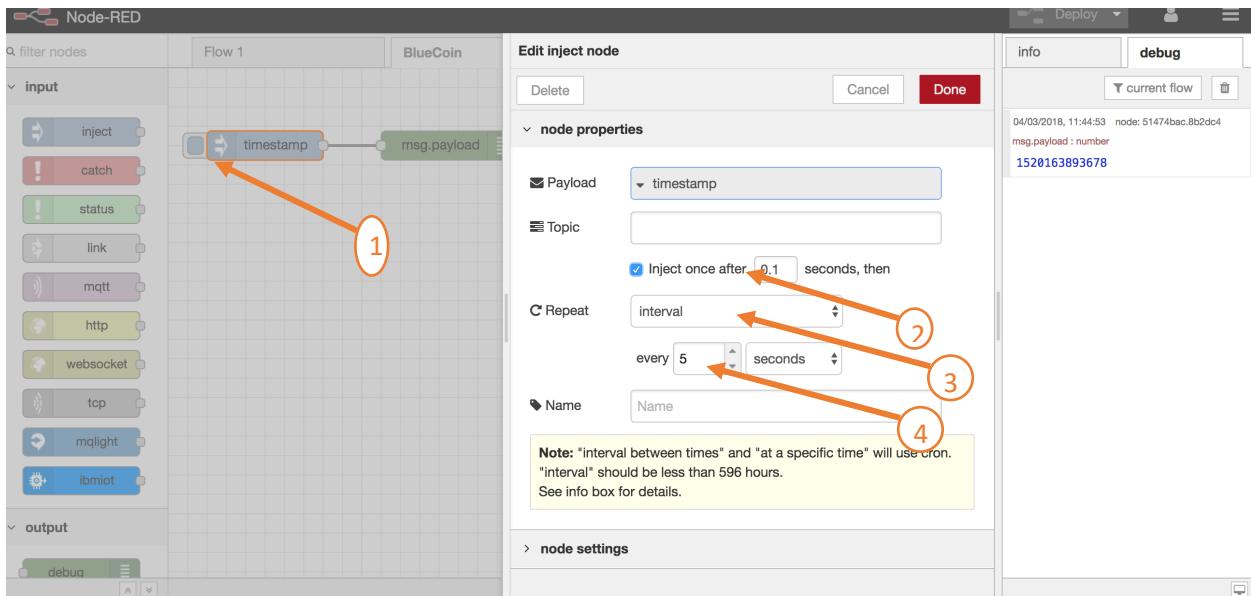
Step 4.3 Test the HelloWorld app

The application is configured to send a timestamp when the inject node is triggered

- Select the icon on the left-hand side of the inject node (2). This will trigger the inject node to inject a timestamp. The timestamp will be sent via the output node along the wire to the input of the debug node. The debug node will then display it in the debug tab.

Step 4.4 Modify the HelloWorld app

Let's modify the application to send a timestamp every 5 seconds.



- Double click on the inject node (1) in the bluecoin flow to open the configuration panel for the node
- Select the **Inject once after checkbox** (2) to trigger the flow to run as soon as the application is deployed
- Change repeat to **Interval (3)**
- Change the interval to every 5 seconds
- **Deploy** the application. Every 5 seconds a new timestamp will be seen in the debug panel.

Step 4.5 Delete the HelloWorld app

The hello world app has shown the basics of Node-Red but has nothing to do with IoT, lets delete it and move on to something a little more interesting,

- Select the inject node and debug node and delete them. The simplest way to delete all nodes in a flow is to use **Ctrl A** and then the **delete** key.
- Once the flow is empty hit **Deploy**

Completed HelloWorld app

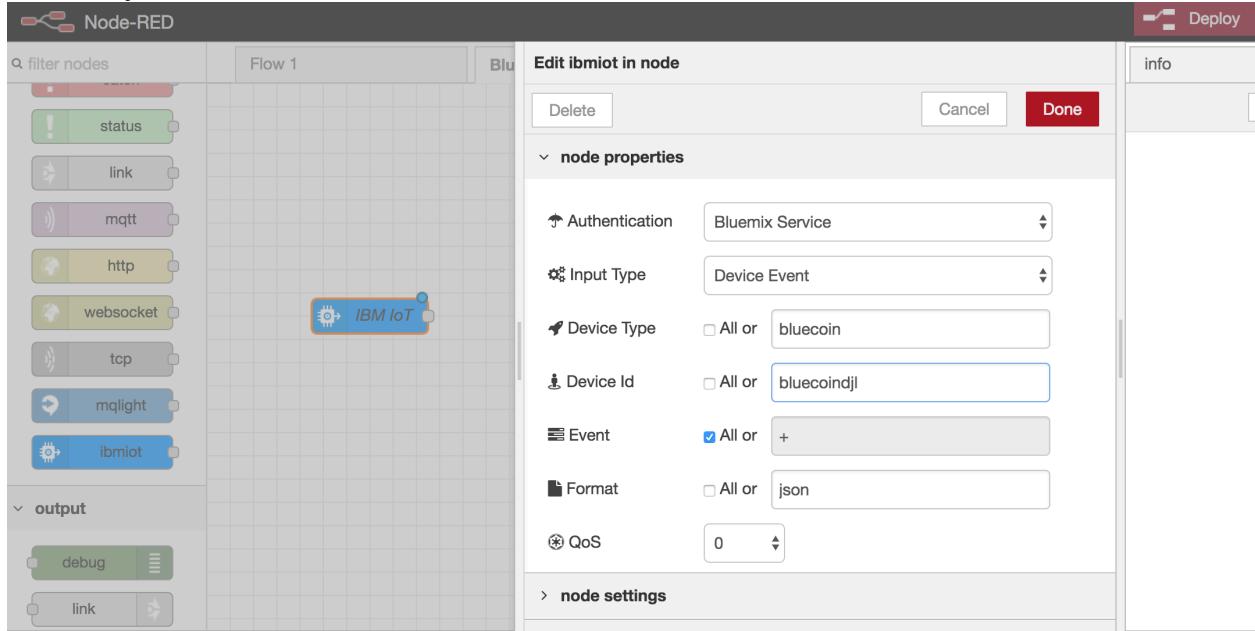
- A complete version of the HelloWorld app can be found at <https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows> as helloworld.flow
- This can be imported into Node-RED by copying the contents of the file to the clipboard and then use the Import clipboard feature of Node-RED

Section 5 – Create a Dashboard to Visualise BlueCoin Data

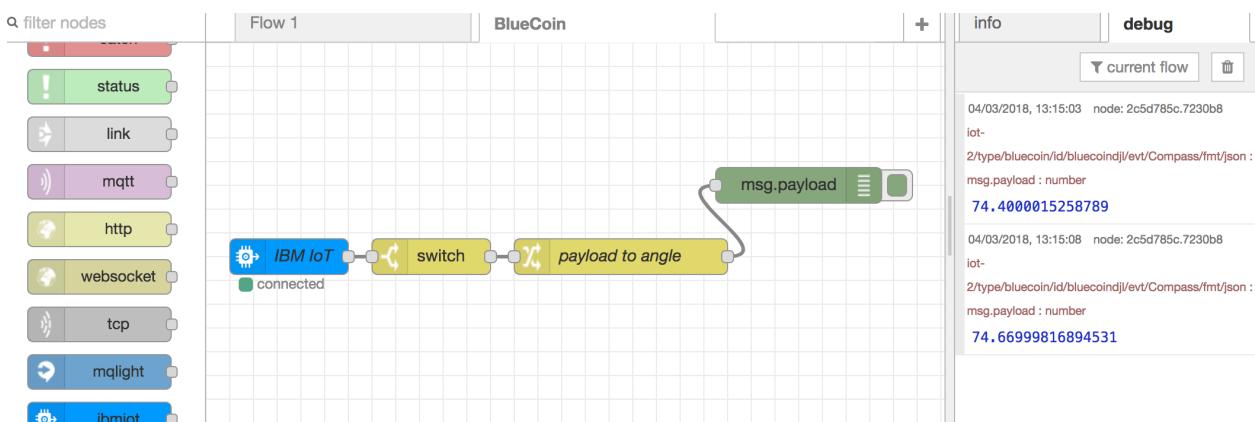
One of the first things stages in an IoT application is build a dashboard to visualize the data from the IoT device. In this section Node-Red will be used to build a Dashboard to display data from the BlueCoin.

Step 5.1 Receive the compass data from the BlueCoin in Node-RED

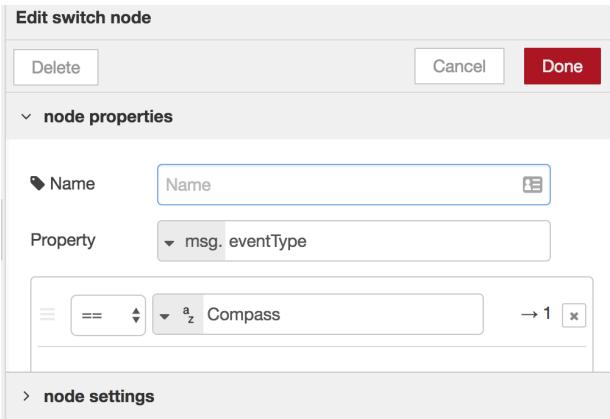
- In the **input** category of your Node-RED palette, find the **ibmiot node** and drag it onto your flow



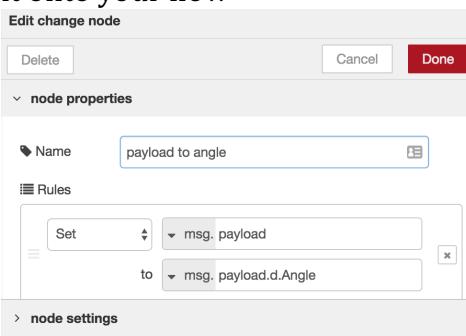
- Double click on the **IBM IoT in** node and configure the node
 - Change QuickStart to Bluemix service. This configures the IoT node to receive data from the WIOTP instance created with the boilerplate.
 - Change Device Type to the device type created earlier
 - Change Device Id to your chosen device id.
- Click on the **Done** button



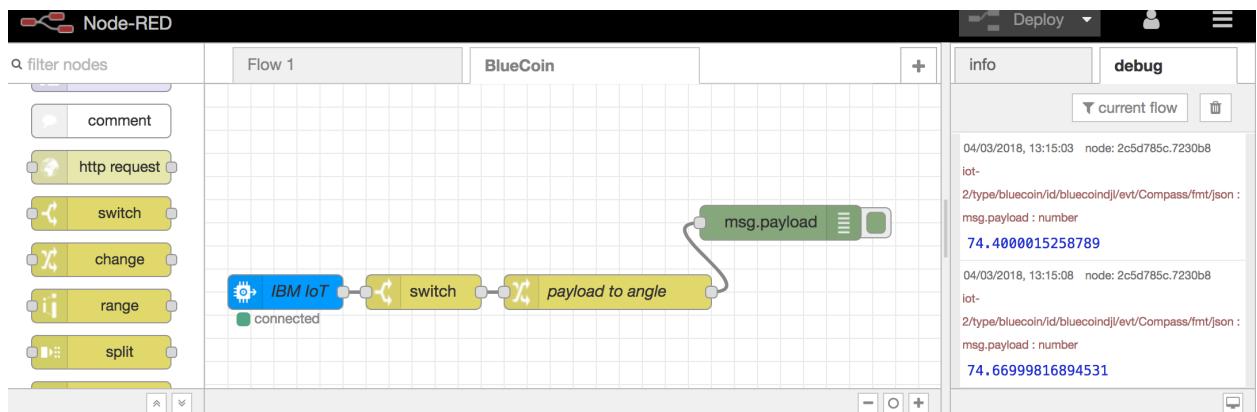
- In the **function** category of your Node-RED palette, find the **switch node** and drag it onto your flow.
- Double click he **switch node**



- Configure the switch node to pick out events of type Compass
 - Change Property to **msg.eventType**
 - Make a selector using **==** comparison and string of **Compass**
 - Click Done
 - This will create a selector node with one output that will output events of type Compass
- In the **function** category of your Node-RED palette, find the **change node** and drag it onto your flow



- Double click on the **change node**
 - Configure the Rules by clicking on the "a/z" dropdown of **to** and set to **msg.payload.d.Angle**
 - Optionally give the node a name "set payload to Angle"
 - Click on the **Done** button.
- In the **output** category of your Node-RED palette, find the **debug** node and drag it onto your flow.
- Wire the four nodes together.
- Click the Deploy button on the top of menu bar to deploy the Node-RED flow.
- Turn to the **debug tab** on the right sidebar of your Node-RED flow.

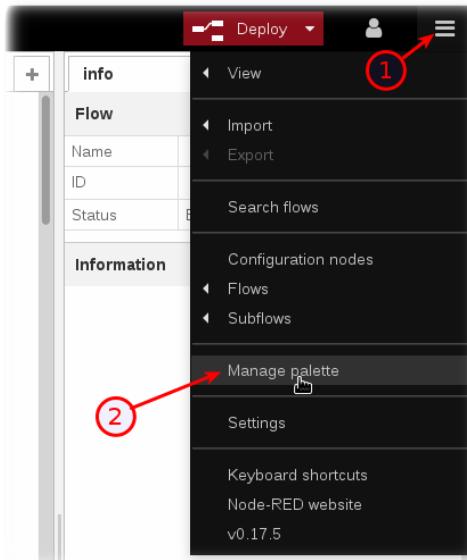


- You should observe the compass angle arriving from the BlueCoin and the ST BlueMS application into your IBM Cloud application every 5 seconds.

Step 5.2 Install the Node-Red dashboard nodes

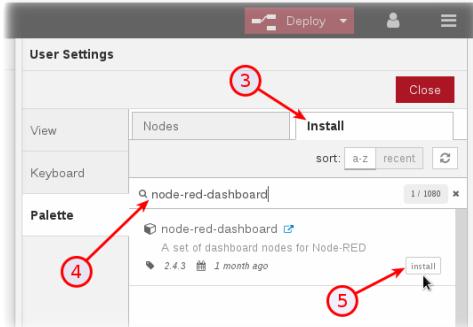
The IoT Starter Application deployed into IBM Cloud includes just a small subset of Node-RED nodes. The Node-RED palette can be extended with over one thousand additional nodes for different devices and functionality. These NPM nodes can be browsed at <http://flows.nodered.org>

- In this Step, you will add the **Node-RED Dashboard** nodes to your Internet of



Things Starter Application.

- Click on the Node-RED **Menu** (1) in the upper right corner, then **Manage palette** (2)
- Turn to the **Install** tab (3), type **node-red-dashboard** (4) and press the **Install** button (5).

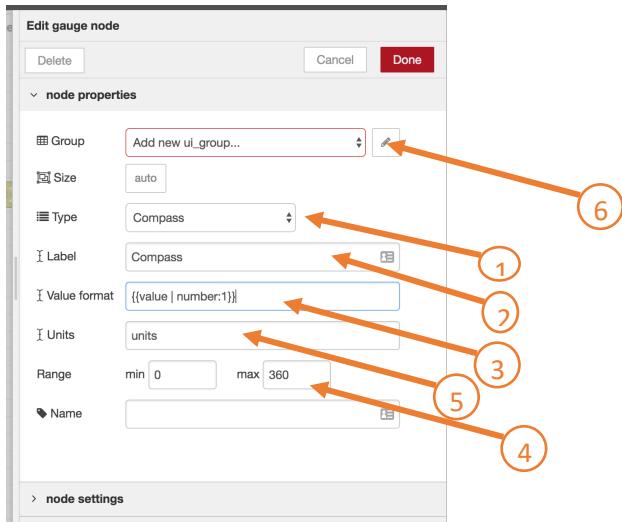


- Press the **Install** button in the next dialog.
- A set of **dashboard** nodes are now available in the palette to help create a graphical user interface

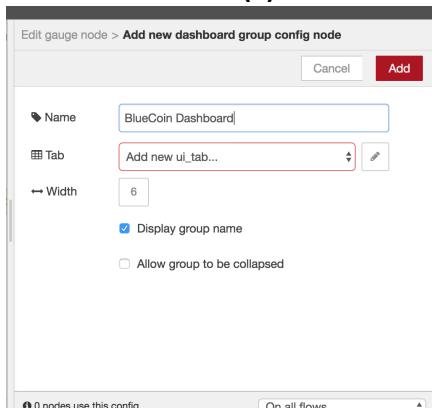
Step 5.2 Create a simple dashboard

The dashboard nodes will be used to create a simple user interface that graphically displays the compass data

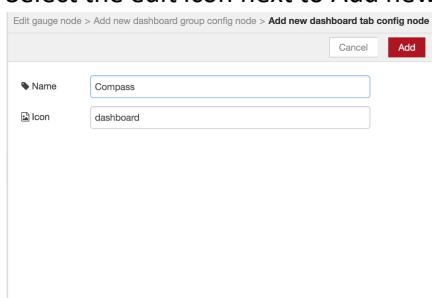
- Drop a **gauge** node onto your flow
- Wire the output of the **change** node to the input of the **gauge** node
- Double click on the gauge node to open the panel to configure the node



- Change the Type of Gauge to **Compass** (1) or Donut if you prefer
- Change Label to **Compass** (2)
- Change Value format to **{{value | number:1}}** (3) this will display the value of the angle to 1 decimal place.
- Change the Range max to **360** (4)
- Change Units to **degrees** (5)
- Select the **edit icon** (6) next to Add new ui-group

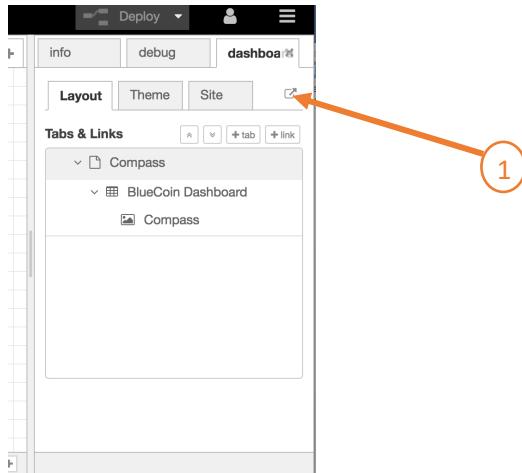


- Change the name of the dashboard to **BlueCoin Dashboard**
- Select the edit icon next to Add new ui tab...



- Give it a name of **Compass**
- Select **Add** to finish the creation of the Compass tab
- Select **Add** to finish the creation of the BlueCoin group

- Select **Done** to complete the creation of the Compass gauge
- **Deploy** the application
- The “simple” dashboard is now deployed. There are several ways to launch the dashboard. The simplest is select the **Dashboard** tab that has appeared next to the debug Tab



- Select the **launch icon (1)**
- Alternatively, launch the dashboard using a url of <https://bluecoin-djl.mybluemix.net/ui/> where bluecoin-djl is the name of your IBM Cloud application
- The compass dashboard will look like this: -



- Every 5 seconds the direction of the compass will be updated

Step 5.3 Change the frequency sensor events are sent to the cloud

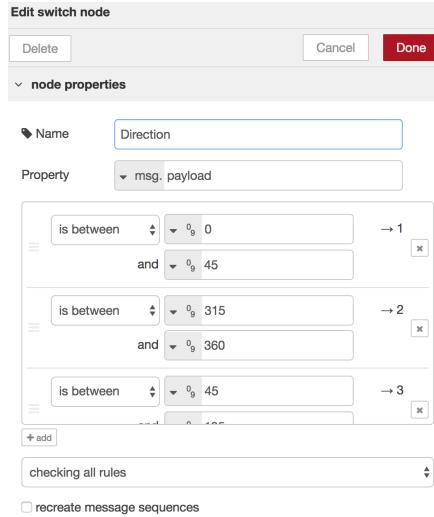
With Smart BMS application sends data to the cloud by default at a 5 second frequency. This limits the responsiveness of the user interface. The frequency can be changed in the mobile app

- Select the **Cloud Logging** page of the mobile app
- Select the menu, three dots in top right corner
- Select **update interval** and select 0.5 seconds.
- The update only takes effect when a new connection is made to the cloud. Select the Cloud icon to stop the existing connection and then select again to make a new connection
- The compass user interface will now update pretty much in real time.

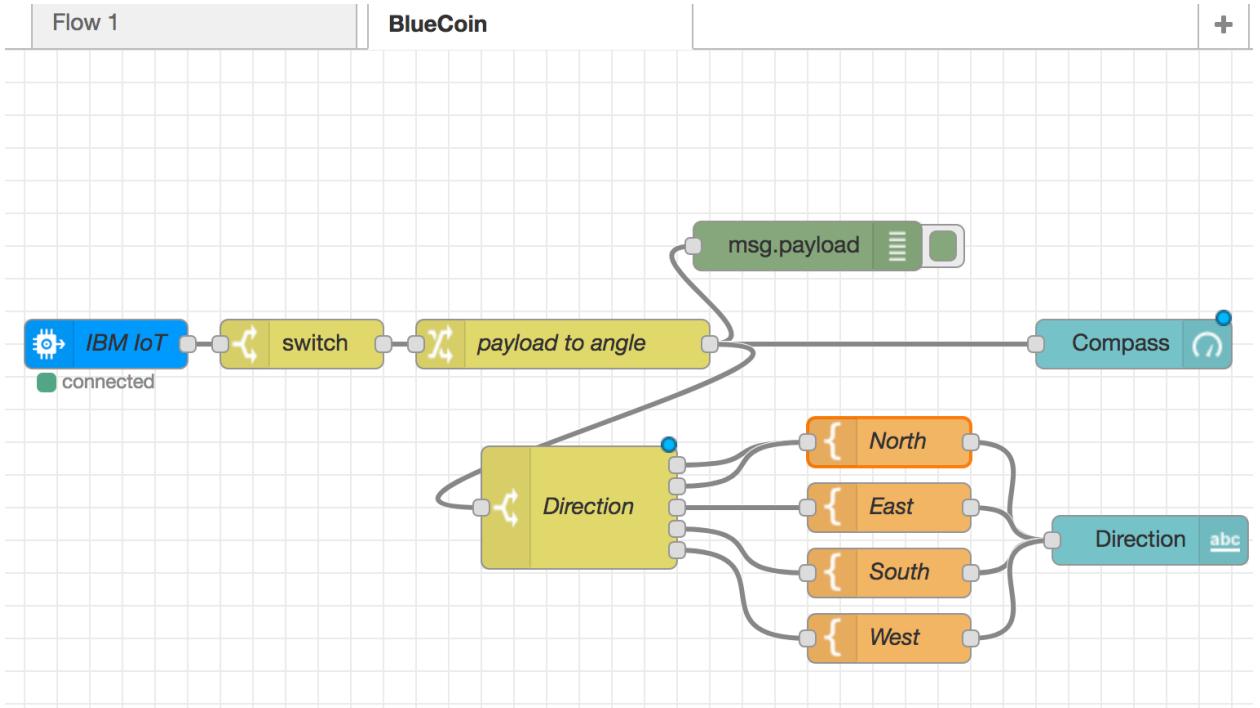
Step 5.3 Add direction logic to the application

In this next step some simple logic will be added the application to determine if the compass is baring North, East, South or West.

- Drag a **switch** node from the function palette to the flow
- Wire the output of the change node that outputs the angle to the input of the switch node
- Double click on the switch node to open the configuration panel for the node



- Configure a set of is between ranges to determine the direction
 - 0 to 45 (N)
 - 315 to 360 (N)
 - 45 to 135 (E)
 - 135 to 225 (S)
 - 225 to 315 (W)
- Drop a **Template** node from the function palette onto the flow



- Wire the direction output 1 to the input of the template
- Edit the Template and set the template content to **North**
- Create a Template for **East**, **South** and **West**
- Wire up the select node outputs to the relevant template
- Drag a **Text** output widget from the dashboard palette
- Wire the 4 direction Templates to the input of the Text widget.
- Edit the **Text** widget and change the label to **Direction**
- **Deploy** the application
- As the bluecoin is moved as well as displaying a compass with the value in degrees it will also display if the baring is North, West, East or South.



Completed Compass Dashboard app

- A completed version of the compass dashboard app can be found at <https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows> as compassdashboard.flow

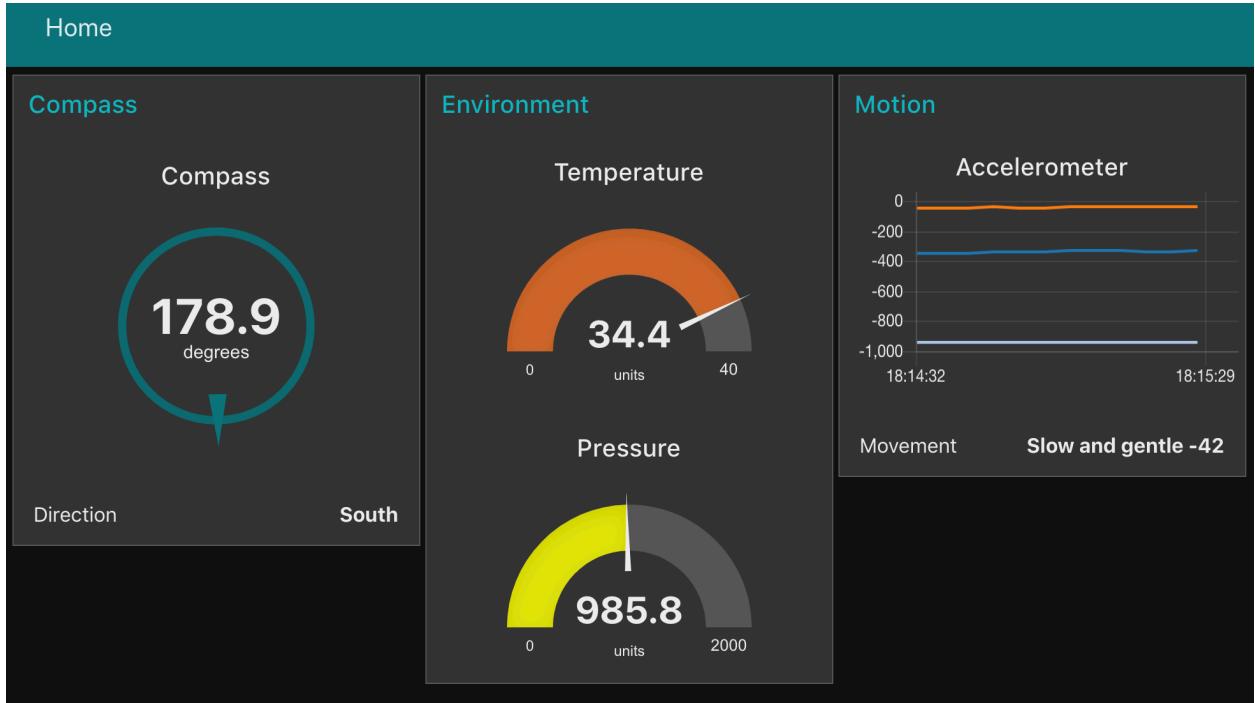
- This can be imported into Node-RED by copying the contents of the file to the clipboard and then use the Import clipboard feature of Node-RED
- After importing the flow edit the ibmiot input node to use your device type and id

Rich Example Dashboard application

A more sophisticated dashboard application is provided as an example.

Optional

- Import the dashboard from <https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows> as richdashboard.flow
- For ST BlueMS Cloud Logging page must be configured to send the following sensor data
 - Compass
 - Accelerometer
 - Temperature
 - Barometric Pressure
- Set the frequency / interval that the ST BlueMS application sends data to every 5 secs.
- After importing the flow edit the ibmiot input node to use your device type and id.



Section 6 Control a Game using the BlueCoin

In this section a space themed game will be loaded. The BlueCoin will be used as the game controller.

<TODO> add in instructions to load the game and control via compass</TODO>

Section 7 Transcribe Speech from BlueCoin to Text

So far in the workshop all of the data that has been used is structured data. In this section unstructured data in the form of voice will be used. The voice data collected by the microphone on the BlueCoin will be connected to the Speech to Text service in the IBM Cloud. The transcribed text will be delivered back to the mobile application to display.

Step 7.1 Configure the IBM Cloud Speech to Text Service

- Head to the IBM Cloud catalog and search for speech
- Select the Speech to Text service

The screenshot shows the IBM Cloud catalog interface. At the top, there is a navigation bar with icons for Catalog, Docs, Support, Manage, and a user profile. Below the navigation bar, the title "Speech to Text" is displayed. To the left, there is a descriptive text block about the service's purpose and capabilities. On the right, there are configuration fields: "Service name" (set to "Speech to Text-wt"), "Choose a region/location to deploy in" (set to "US South"), "Choose an organization" (set to "instructor0300@ib..."), and "Choose a space" (set to "dev"). Below these fields, the "Features" section is visible, listing "Available Languages" (English (US), English (UK), Japanese) and "Metadata". At the bottom right, there is a large blue "Create" button with a white speech bubble icon. An orange arrow points from the number "1" to the "Create" button.

The Speech to Text service converts the human voice into the written word. It can be used anywhere there is a need to bridge the gap between the spoken word and their written form, including voice control of embedded systems, transcription of meetings and conference calls, and dictation of email and notes. This easy-to-use service uses machine intelligence to combine information about grammar and language structure with knowledge of the composition of the audio.

Service name:
Speech to Text-wt

Choose a region/location to deploy in:
US South

Choose an organization:
instructor0300@ib...

Choose a space:
dev

Features

- Available Languages
English (US) English (UK) Japanese
- Metadata
Receive a metadata object in the JSON

Need Help?
[Contact IBM Cloud Sales](#)

Estimate Monthly Cost
[Cost Calculator](#)

Create

- Select **Create (1)**
- Select **Service Credentials (1)**

Watson /

Speech to Text-wt

Location: US South Org: instructor0300@ibmlearning.org Space: dev

Service credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

[View More](#)

New credential +

Click **New credentials** to create a set of credentials for this instance

- Select **New Credentials (2)**

Add new credential

Name: bluecoin

Add Inline Configuration Parameters (Optional): [?](#)

[Cancel](#) [Add](#)

- Give the credential a **name (1)** like **bluecoin**
- Select **Add (2)**
- On the resulting screen select **View Credentials**

The screenshot shows the 'Service credentials' section of the IBM Cloud interface. At the top, it displays the service name 'Speech to Text-wt' and user information: Location: US South, Org: instructor0300@ibmlearning.org, Space: dev. Below this is a table titled 'Service credentials' with one item listed:

KEY NAME	DATE CREATED	ACTIONS
bluecoin	Mar 4, 2018 - 06:51:53	View credentials

The 'View credentials' link is expanded, showing the following JSON content:

```
{
  "url": "https://stream.watsonplatform.net/speech-to-text/api",
  "username": "ae070230-8ac5-4cf6-a63d-3a3fe01e0edd",
  "password": "HRdyC6CM1bJr"
}
```

- Note the username and password as they will be entered into the ST BlueMS mobile application

Step 7.2 Configure the BlueMS mobile app to talk to the cloud S2T service

- Open the ST BlueMS application on you phone and connect to the BlueCoin
- Select the **SpeechToText** page
- Hit **SELECT** and choose **IBM Watson**
- The choose **English US** or **UK** depending on you accent
- Select **SET KEY**
- Enter the username and password from the cloud speech to text view credentials page ensuring they are entered exactly as displayed.

Step 7.3 Test the Speech to Text

- To enable the BlueCoin to start listening for voice and the voice to be sent to the speech to text service double tap the BlueCoin. The double click wakes up the BlueCoin, once awake 4 red LEDs will flash alternatively if beamforming is on only one LED will flash.
- Once awake speak to the BlueCoin and shortly after the transcribed text will be displayed in the mobile app.

Section 8 Using Voice in the Cloud Application

In section 7 the voice was sent from the BlueCoin to the Speech to Text Service in IBM Cloud and the transcribed text was sent back and displayed on the mobile app. In this section the Node-RED cloud application will be updated to use the transcribed text.

Step 8.1 Bind the Speech to Text service to the Cloud Application

In IBM Cloud a service can be bound / connected to an application. This enables the application to securely interact with the service.

- In IBM Cloud Dashboard select the application

IBM Cloud Dashboard - Cloud Foundry apps / bluecoin-dj! Not running Visit App URL

Org: instructor0300@ibmlearning.org Location: US South Space: dev

Connections

Logs

Monitoring

API Management

1

Items per page | 1-3 of 3 items

1 of 1 pages < 1 >

Filter items

Create connection +

CONNECTION NAME	TYPE
bluecoin-dj!-cloudantNoSQLDB	Cloudant NoSQL DB
bluecoin-dj!-iotf-service	Internet of Things Platform

2

- Select **Connections (1)**
- Select **Create Connection (2)**
- Select the speech to text service created In Section 7

Connect Existing Compatible Service

All Resources

10 Items per page | 1-1 of 1 items

1 of 1 pages < 1 >

Search compatible services

SERVICES	RESOURCE GROUP	PLAN	SERVICE OFFERING
Speech to Text-wt	--	Lite	Speech to Text

Connect

1

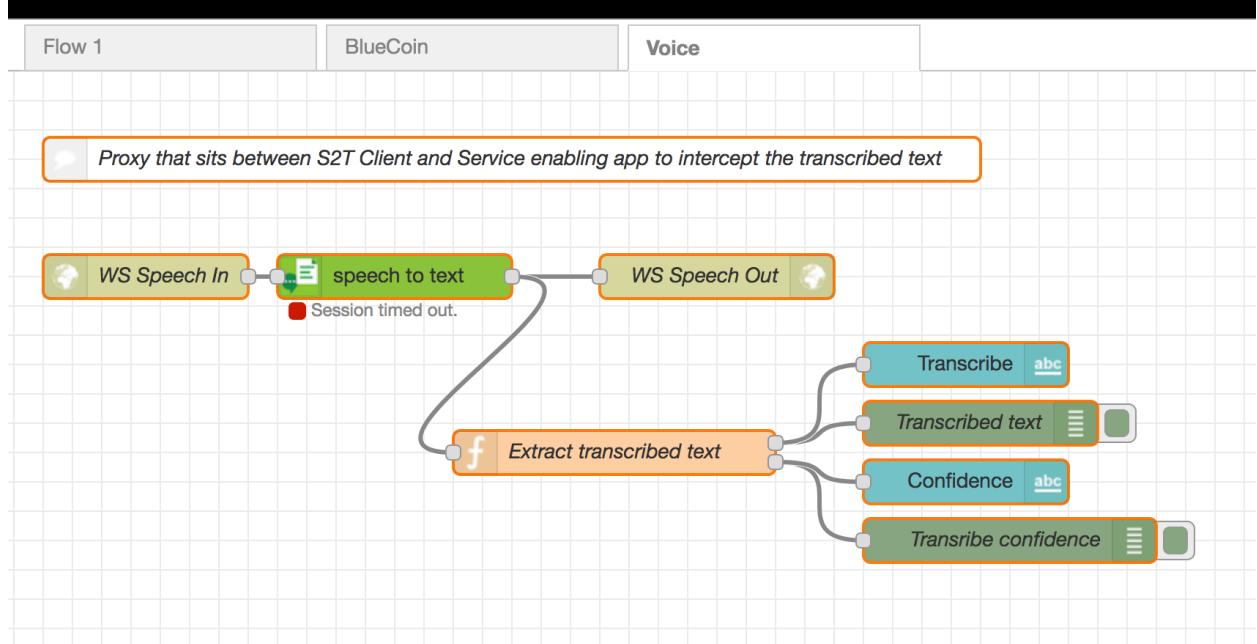
- Select **Connect (1)** to bind the service to the application
- A dialog will pop up, select **Restage**. It will take a few mins to restage the application

Step 8.2 Setup the Speech to Text Proxy

Until now the transcribed text has been sent to the mobile application, this does not allow the cloud application to use it. A speech to text proxy will be introduced into the Node-RED application that enables the transcribed text to be used in Node-RED in addition to being delivered to the mobile application.

- In Node-RED create a new flow called **Voice** and select the flow
- Copy the contents of file **Speech2TextProxy.flow** at <https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows> to the clipboard
- In Node-RED open the menu (3 horizontal bars) and select Import -> Clipboard
- Paste the contents of the clipboard into the paste nodes here text input field and then select **Import**

- Drop the nodes onto the **Voice** flow. It should look like



- Deploy** the flows

Step 8.3 Change BlueMS mobile app to point at the Speech to Text Proxy

- On the smart phone open the BlueMS app and select the **SpeechToText** page
- Select **SET KEY**
- Change the Api Endpoint to <https://bluecoin-djl.mybluemix.net/ws/stt> where *bluecoin-djl* is the name of your application
- Test the speech to text as in Step 7.3. The transcribed text will appear in the Node-RED debug tab and also in the Dashboard

<TODO>

Add in Watson Assistant - Waiting on API key for access to the WA service

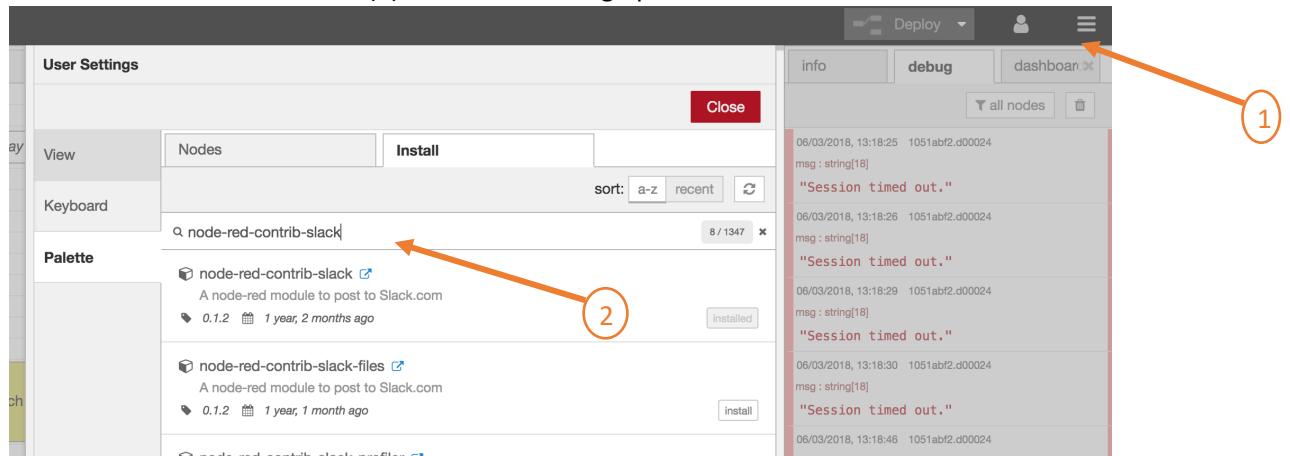
Section 9 Use voice and sensor to post a slack message

In this section we will enhance the application to talk to a service that is external to the IBM Cloud. This lab uses Slack as the external service which the application will post messages to. External services can be integrated into a Node-RED in several ways. If the external service has a HTTP / RESTful API then the HTTP nodes can be used to directly call the external service. In many cases there will a Node-RED node that corresponds to the external service that makes it easier to integrate with the external service, this is the case for Slack.

<TODO>Confirm that 512Mb is needed to add and use slack node. If not use http ? </TODO?

Step 9.1 Install a Slack Node-Red node

- Head to the **Node-Red Menu (1)** and select manage palette



- Enter “**node-red-contrib-slack**” into the search field (2)
- Install** the resulting node-red-contrib-slack node
- Three slack nodes will now be visible in the Social category in the palette

Section 10 Summary

Completed an end to end application using core building blocks from HW device, phone and cloud – attendees in good shape to go and “make”

<TODO> Add summary and follow on...

We Value Your Feedback!

- Don't forget to submit your Think 2018 session and speaker feedback! Your feedback is very important to us – we use it to continually improve the conference.
- Access the Think 2018 agenda tool to quickly submit your surveys from your smartphone, laptop or conference kiosk.

