Lab Center – Hands-on Lab

Session 4104
Session Title Beam Me Up Watson

Dave Locke, IBM
John Walicki, IBM
Ernesto Manuel Cantone, STMicroelectronics

Table of Contents

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.

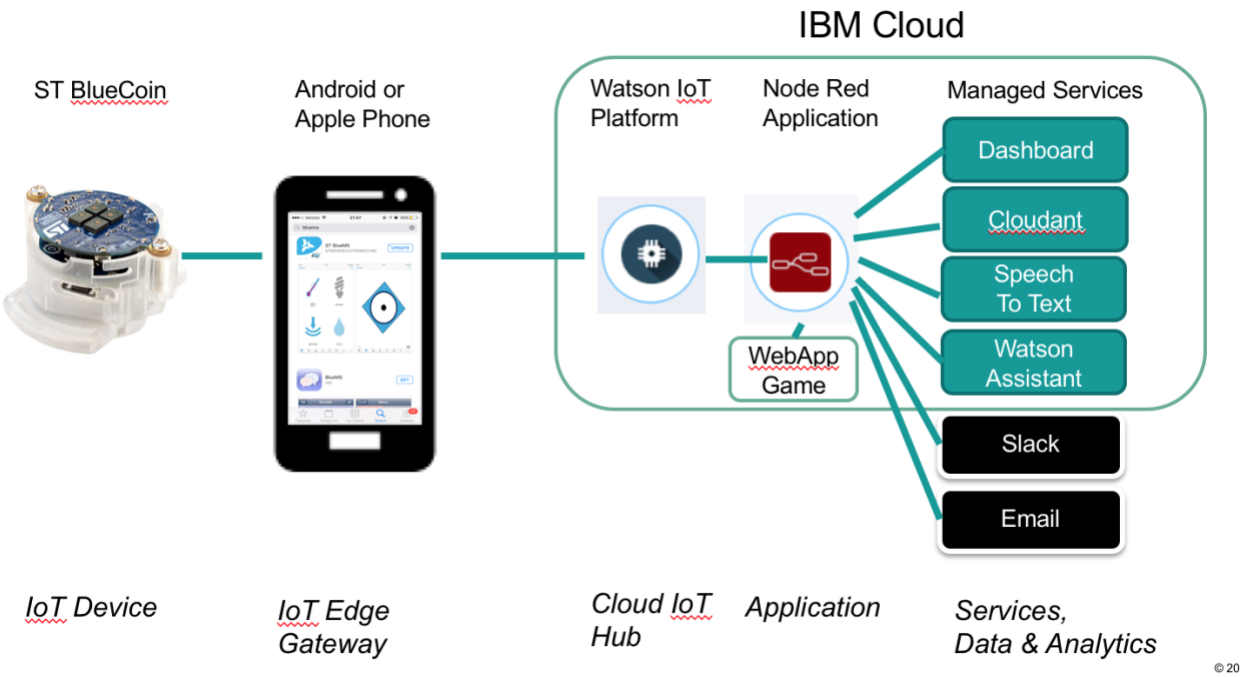
IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current 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 STMicrolectronics, combines a ST BlueCoin IoT starter kit, a mobile app and a set of IBM Cloud services to create a lapel communicator and sensor platform. The lab shows 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, Speech to Text 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 rich 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 an 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
  - 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 including IoT, Data, Analytics, DevOps and Watson.
- Watson IoT Platform
    - A managed service provided in the IBM Cloud
    - Acts as a hub for connecting and managing devices
    - Enables data from devices to be handed to upstream applications, services and analytics
    - Enables data and commands to be sent to devices.
- Node-RED
    - An application development tool and runtime offered in the IBM Cloud
    - Enables applications to be built using visual wiring combined with drag and drop.
    - In some cases, applications can be created without writing a single line of code.
    - Node-RED also runs in many other environments including laptops and Raspberry Pis.
- Dashboard
    - A Web dashboard will be created as part of the lab enabling visualization of device and application data
- Cloudant
    - A document-based data store provided as a managed Service in the IBM Cloud
    - Can be used to store device and application data
    - Is used by Node-RED to store configuration information
- Speech to Text
    - An IBM Cloud service that converts speech to text
    - Used to transcribe the voice data collected by the BlueCoin to Text
- Watson Assistant
    - 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.
    - Provided as a managed service in IBM Cloud
    - Transcribed voice will be used to interact with Watson Assistant
- Slack
    - An external service that the lab will interact
    - Message will be posted to a slack channel and any messages posted to the channel will be received by the app and displayed.
- Web Application
    - A pre-built web application in the form of a game
    - The lab will extend the game to use the BlueCoin to control the game and to publish the high score to slack

Pre-reqs

- IBM Cloud ID
    - To run the lab an IBMid and cloud account 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 IBMid and account or use an enterprise account.
        - A Lite account that provides free access to IBM Cloud and 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 may not run.
- Android or Apple smart phone or tablet
    - 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 or tablet capable of running the ST BlueMS mobile application.
    - The ST BlueMS mobile application is available on both the Google and Apple app stores
        - Android https://play.google.com/store/apps/details?id=com.st.bluems&hl=en
        - iOS https://itunes.apple.com/gb/app/st-bluems/id993670214?mt=8
    - The smart phone or device must have Bluetooth (BLE) support enabled for the BlueCoin to communicate to the phone/tablet.
    - The smart phone / tablet must have internet connectivity either via mobile data or via Wi-Fi.
    - Note: The screenshots in this PDF are taken from an Android device. The features available in the iOS app are the same but the interaction differs.

Routes through the lab

The lab is broken into a number of sections which are presented in a logical order. It is recommended to follow the labs from start to finish. Alternatively, there are sections which can be grouped together and followed in a different sequence or missed if time is short.

Core (must be followed)

- ST ClueCoin
- Sections 1 through 5

Optional group: Slack integration and Game controlled by BlueCoin

- Sections 6 and 7

Optional group: Transcribe Voice and Voice interaction

- Sections 8 and 9

Each individual section is designed to take approximately 10 minutes.

ST BlueCoin

This lab uses hardware in the form of the BlueCoin Starter Kit from STMicroelectronics. The BlueCoin integrated development and prototyping platform for augmented acoustic and motion sensing for IoT applications builds on the listening and balancing capabilities of the human ear.

It lets you explore advanced sensor fusion and signal processing functions with a 4 digital MEMS microphone array, a high-performance 9-axis inertial and environmental sensor unit and time-of-flight ranging sensors.

An Introduction to STMicroelectronics and the BlueCoin Starter Kit are provided in a separate document that covers; -

- About
- BlueCoin device / HW intro
- Device to Phone Connectivity
- Smart Phone App
- Phone to Watson IoT Platform Quickstart (sensor data to cloud)

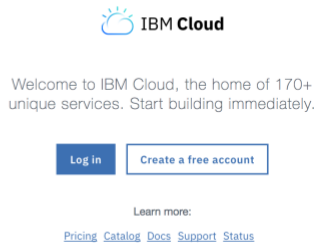
The training material for the BlueCoin can be found here
https://github.com/lockedj/BeamMeUpWatson/blob/master/4104%20BlueCoin%20Hands-on%20Training.pdf
Before progressing, you should be familiar with the BlueCoin and related ST BlueMS mobile application. The BlueCoin must be setup and communicating with the mobile app. In addition connectivity to the IBM Cloud Quickstart service must be working.
The ST SensorTile is another starter kit that has similar capability to the BlueCoin. The material in this lab can also be used with the SensorTile.

Section 1 – Login and prepare your IBM Cloud account

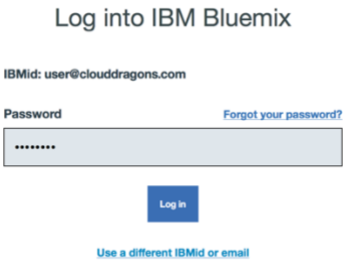
This section covers how to login to IBM Cloud.
Think2018 Attendees will be provided with an IBMid for use during the lab.

Step 1.1 Login

- In a new browser tab, go to http://bluemix.net and click “Log in”
  - If you do not already have an IBM id, 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*

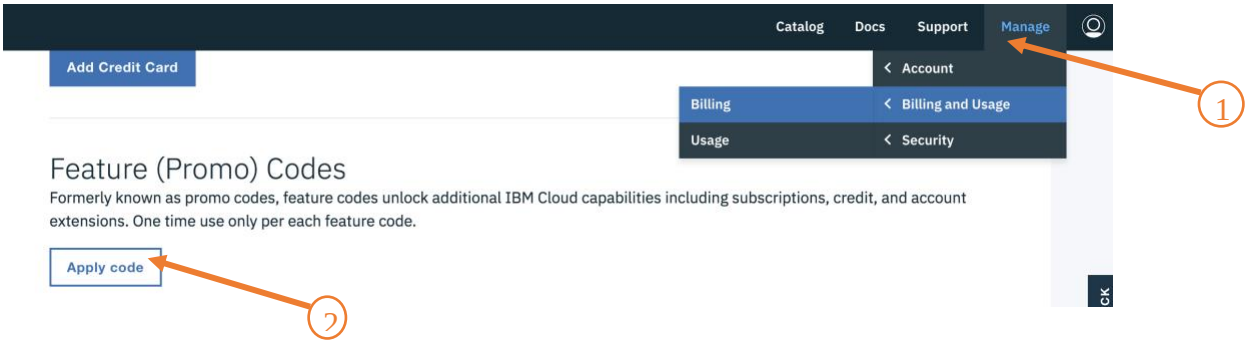


- Log in to IBM Cloud with your IBMid and password



Step 1.2 Think 2018 attendees apply promo code

- For Think 2018 attendees a promo code is supplied that will upgrade from a Lite account to a Trial account with more memory enabling all of the sections of the workshop to be completed. Once logged, apply the promo code



- Select menu **Manage,** then **Billing and Usage,** then **Billing (1)**
- Under Feature (Promo) Codes click **Apply code (2)**
- Enter and apply the code you have been provided with

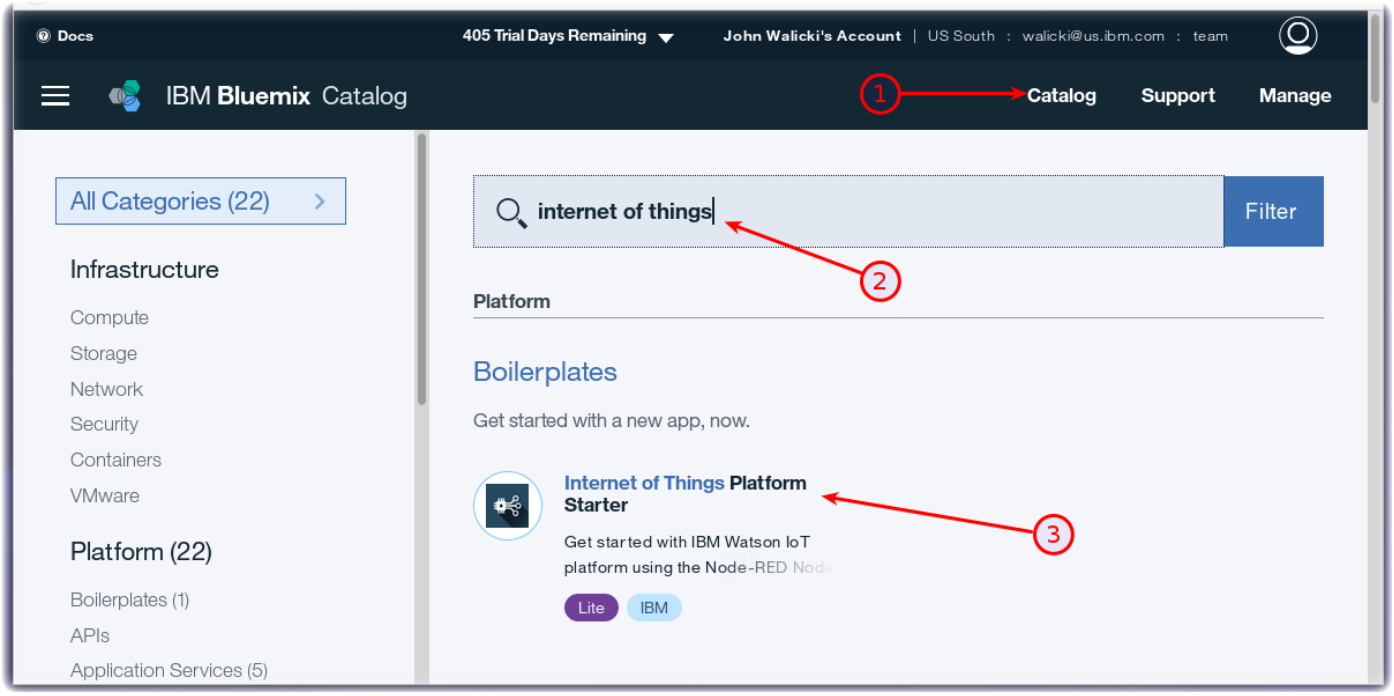
Section 2 – Create an Internet of Things Starter App

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 will be used to develop our application that will be hosted and run in the IBM Cloud.

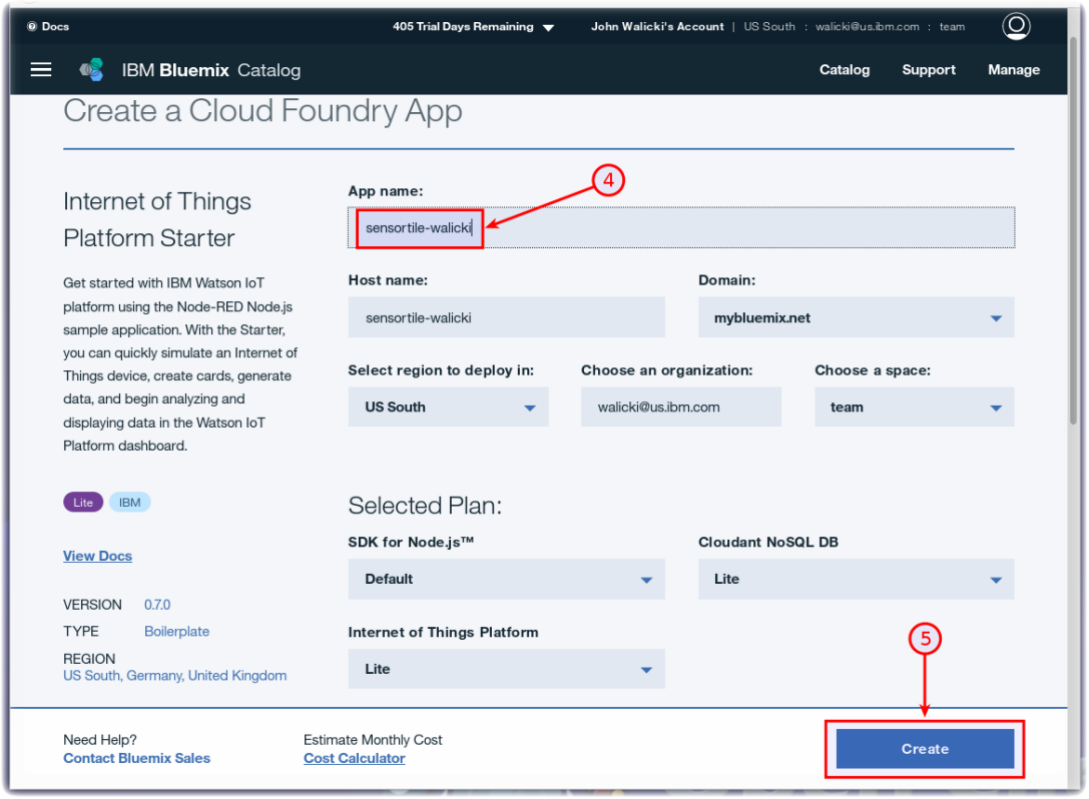
Step 2.1 – Create an IoT Starter Application

To get started quickly a Boilerplate will be used to speed up the process of creating instances of WIoTP and Node-RED. The boilerplate performs multiple actions that if required can also 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



- 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



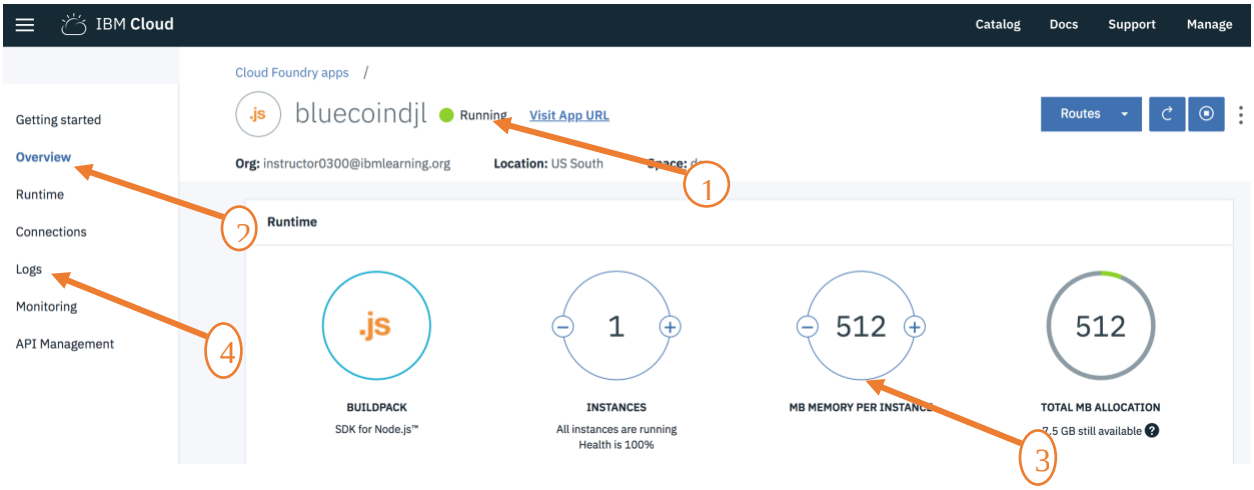
- Press the **Create** button (5).

- 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 (4 in the screenshot below) and see activity logs from the platform and Node.js runtime.
  - It should not take more than 5 mins to stage and start.

Step 2.2 – Increase the memory available to the application

If the IBM Cloud account has more than 256Mb available, the memory available for the Starter Application should be increased to 512Mb

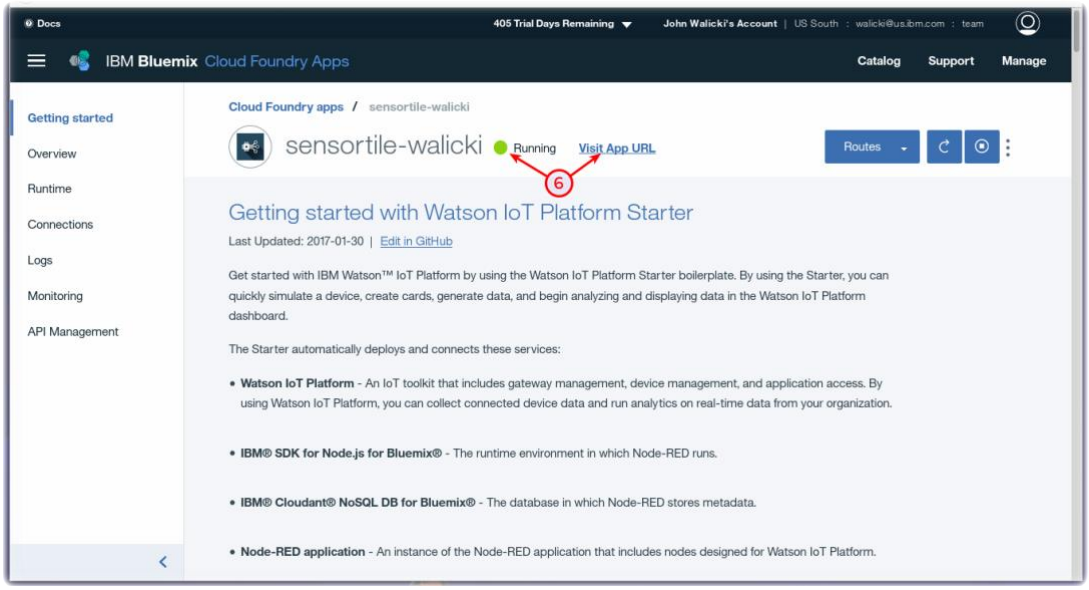
- Once the Green "**Running**" (1) icon appears



- Select **Overview** (2) for the application
- The MB Memory per instance defaults to **256MB**. Increase this to **512MB (3)**
- Select **Save** and the application will be restaged (stopped and restarted). Once restarted the application will have 512MB available.
    - Note: To make the memory allocation change persistent the manifest.yml file needs to be updated. The change is not needed for the lab as the starter app will not be restarted.

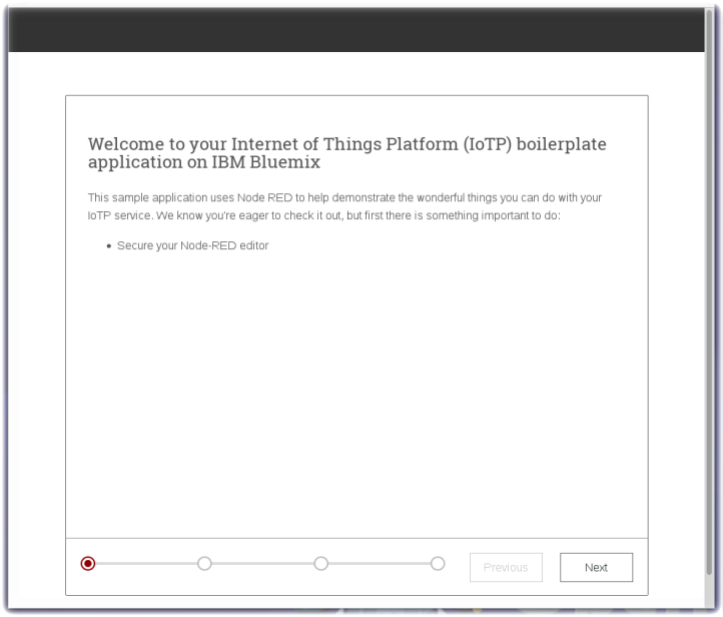
Step 2.3 - Launch the IoT Starter Application

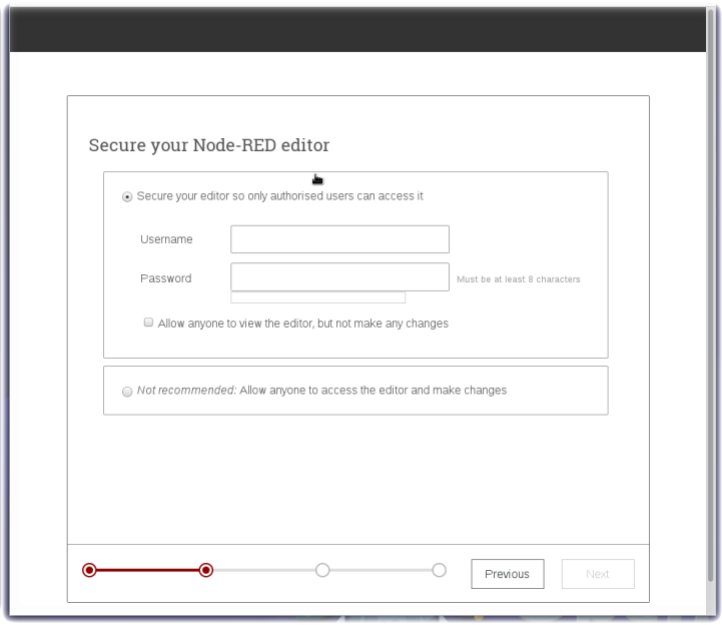
- Once the Green "**Running**" icon appears, Click the **Visit App URL** link (6).

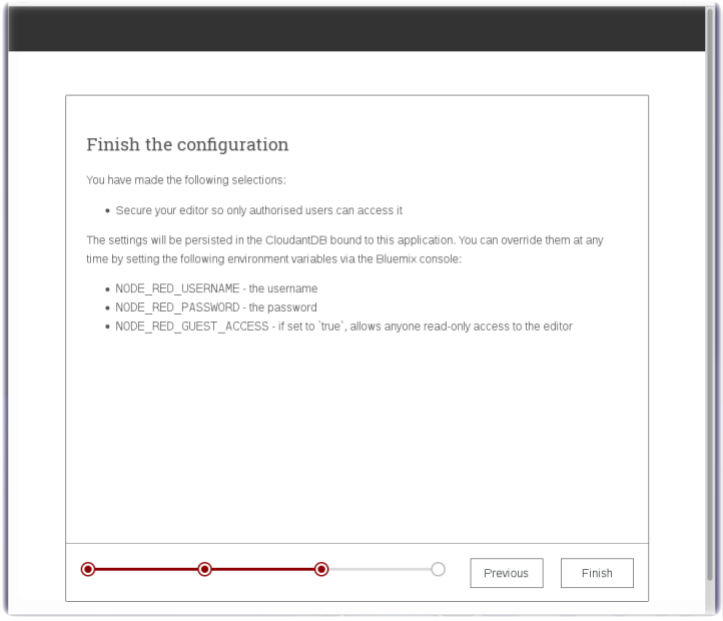


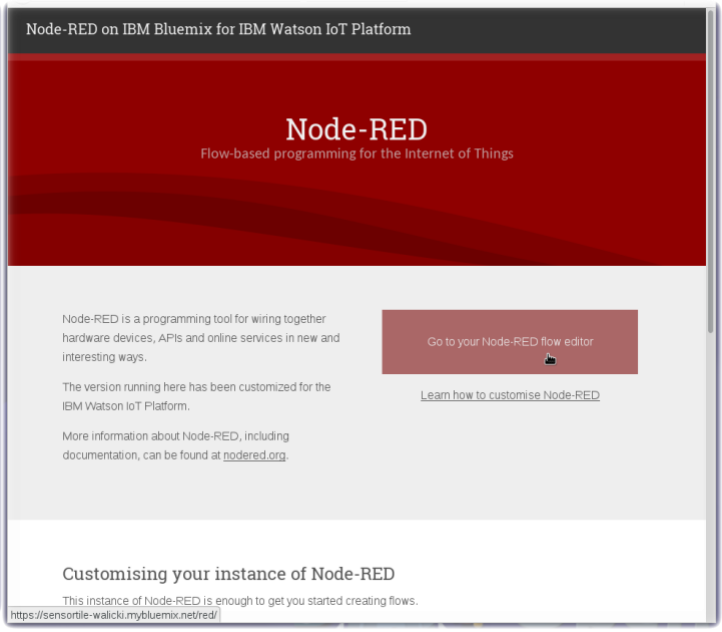
Step 2.4 – Configure the Node-RED visual programming editor

- 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. Click **Next**
- On the next screen select a username / password to access the Node-RED editor. **Remember** your username / password
- Click **Next**
- Click **Finish**
- Click the red button **Go to your Node-RED flow editor** to launch the editor

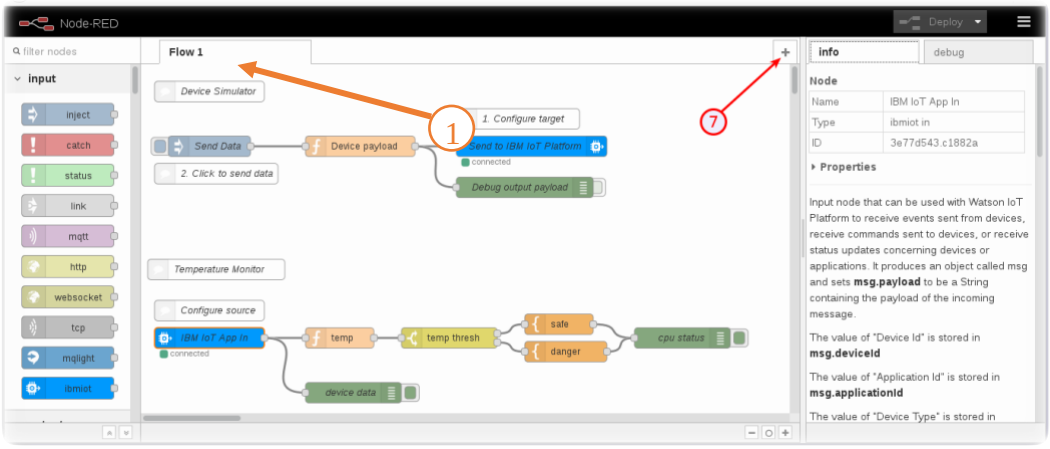

Welcome to your Internet of Things Platform (IoTP) boilerplate application on IBM Bluemix
This sample application uses Node RED to help demonstrate the wonderful things you can do with your IoTP service. We know you're eager to check it out, but first there is something important to do:
• Secure your Node-RED editor
Previous
Next


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
Previous
Next


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 on IBM Bluemix for IBM Watson IoT Platform
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.
- Nodes are wired together to create a program.
- The default "sample" flow (application) 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.
- There is plenty more to come on Node-RED later in the lab.



- The default flow is not used in the lab and should be deleted. Double click Flow 1 tab (1) then select delete.

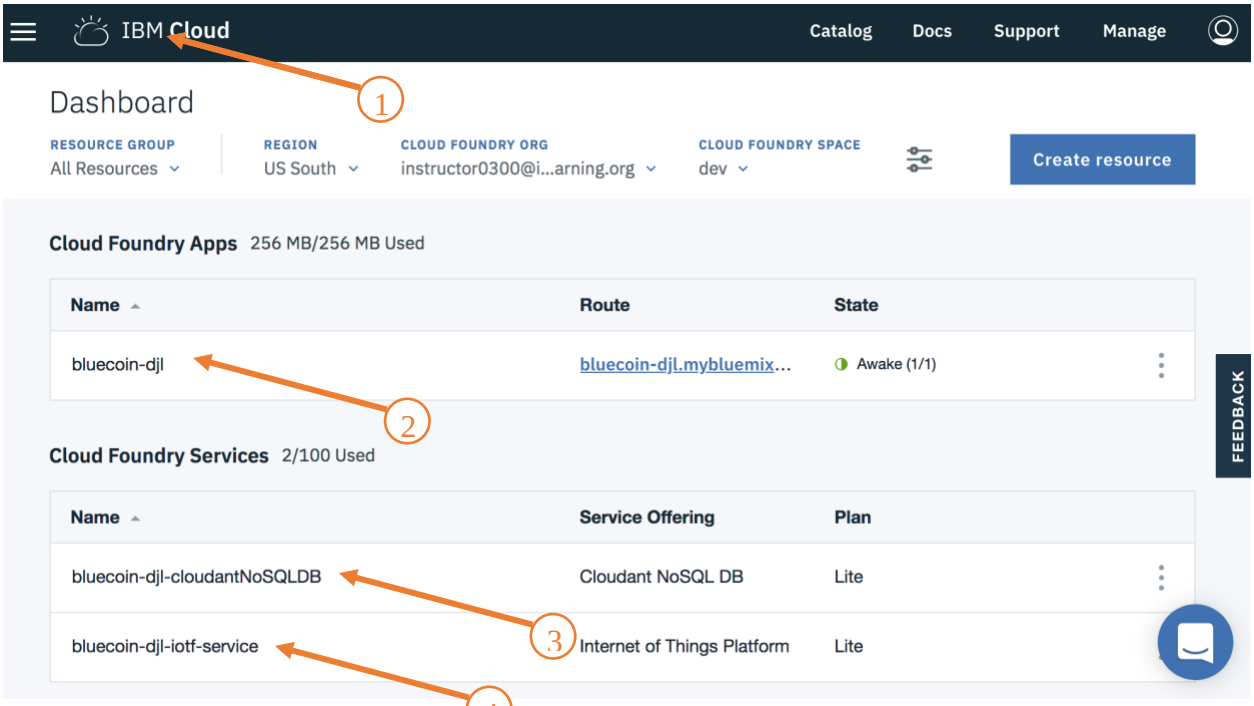
Section 3 – Connect the BlueCoin to WIoTP

Earlier in the lab the BlueCoin was connected to IBM Cloud Quickstart service and the sensor data from the BlueCoin was 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 3.1 – 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



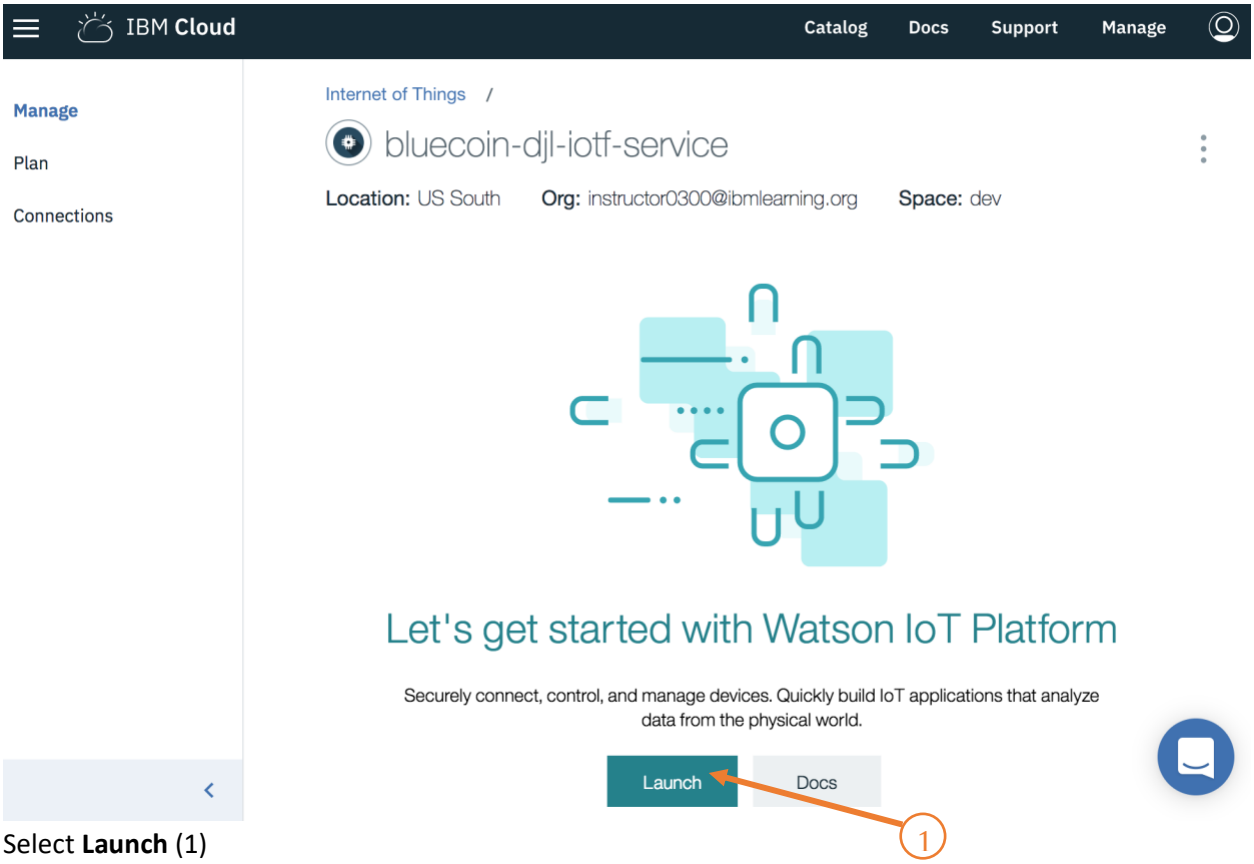
- The dashboard shows the applications and services that were created by the Internet of Things Starter boilerplate: -
    - One Cloud Foundry Application (2)
    - Two services, a cloudant document-based data store (3) that is used by node-red to store its configuration data and can optionally be used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

Step 3.2 – Register the BlueCoin device with Watson IoT Platform

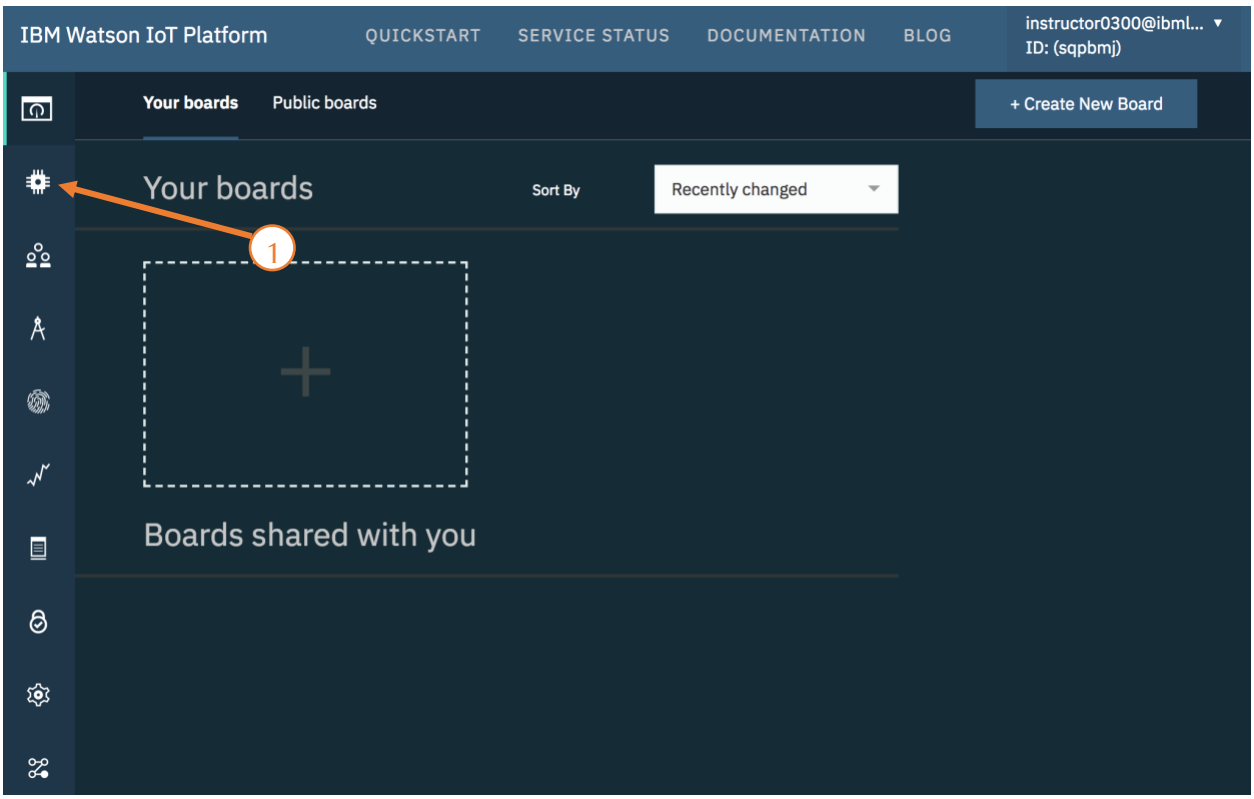
The Watson IoT Platform (WIOTP) dashboard is the user interface for interacting with all resources related to the WIoTP service. The dashboard will be used to register the BlueCoin with the platform.

Step 3.2.1 Launch the WIOTP dashboard

- In the IBM Cloud dashboard select the WIoTP instance (4) to bring up the launch page for the WIoTP dashboard

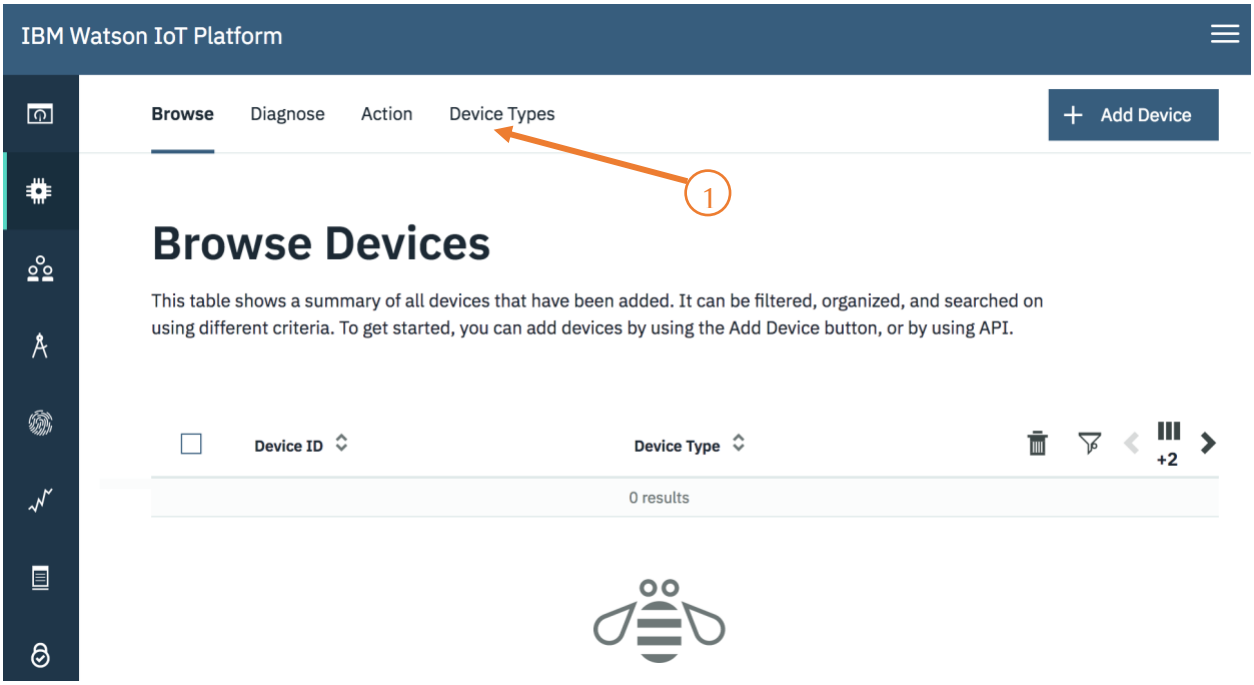


- Select **Launch** (1)
- Now head to the **Devices** page of the dashboard by selecting the **Devices** icon (1)

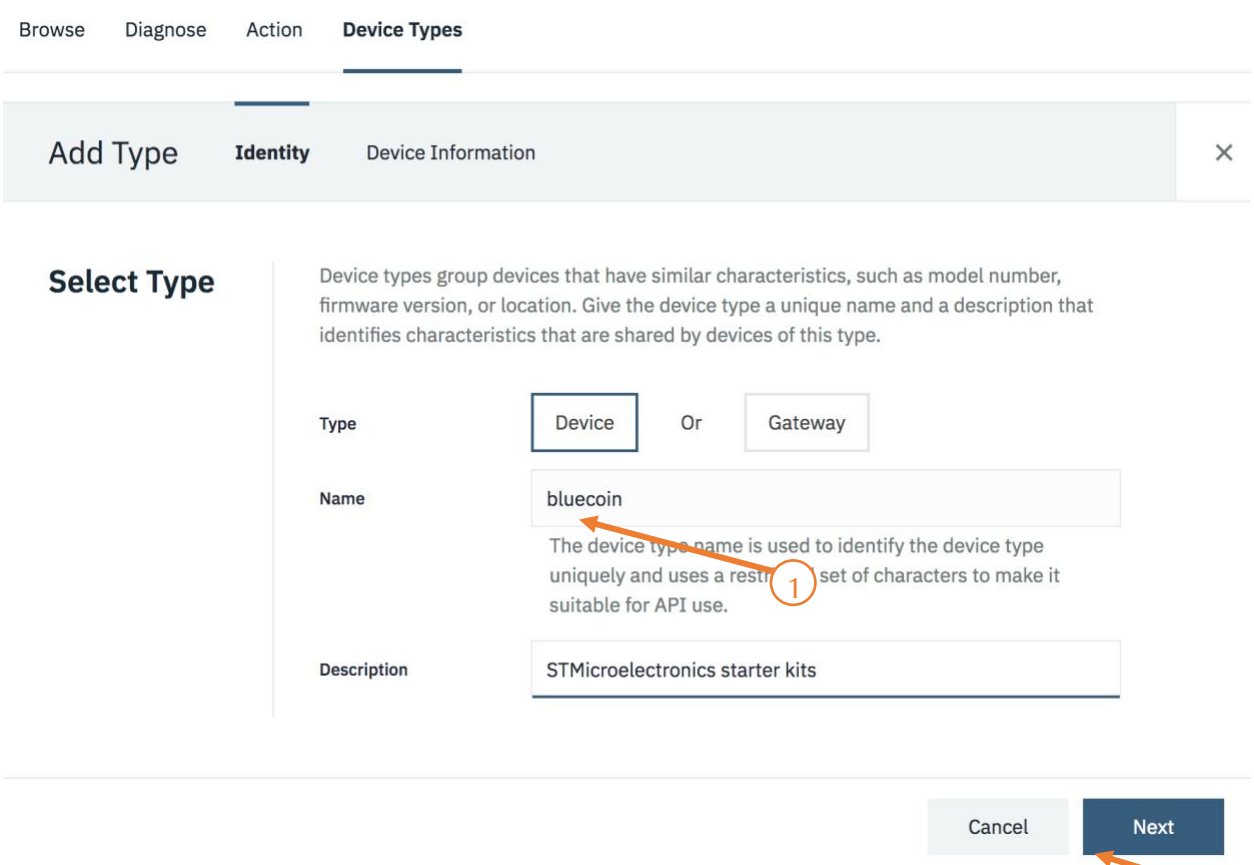


Step 3.2.2 Define a Device Type for the BlueCoin

- 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)



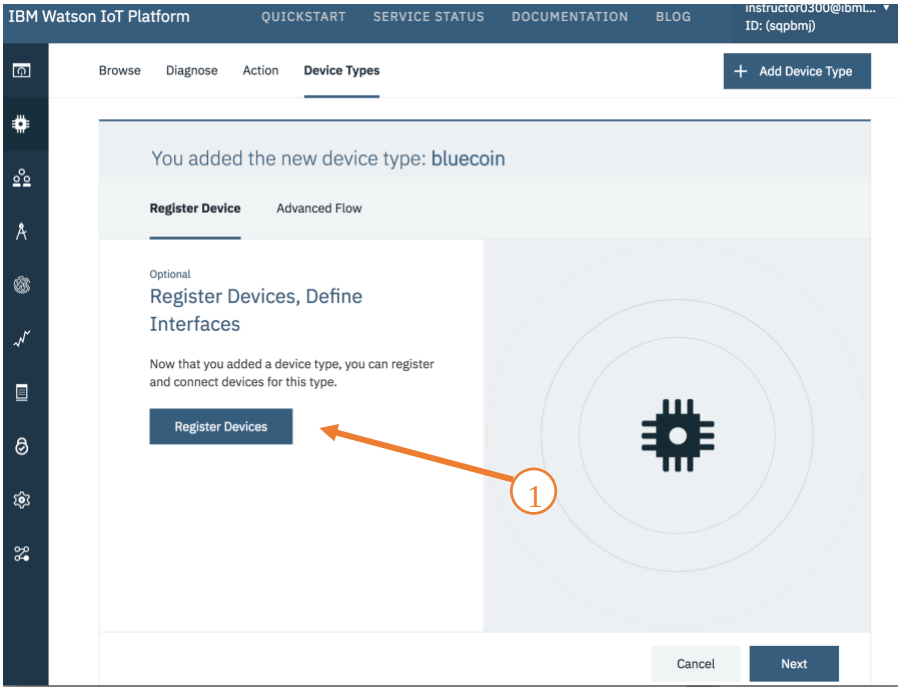
- Select **+ Device Types** in the top right of the dashboard



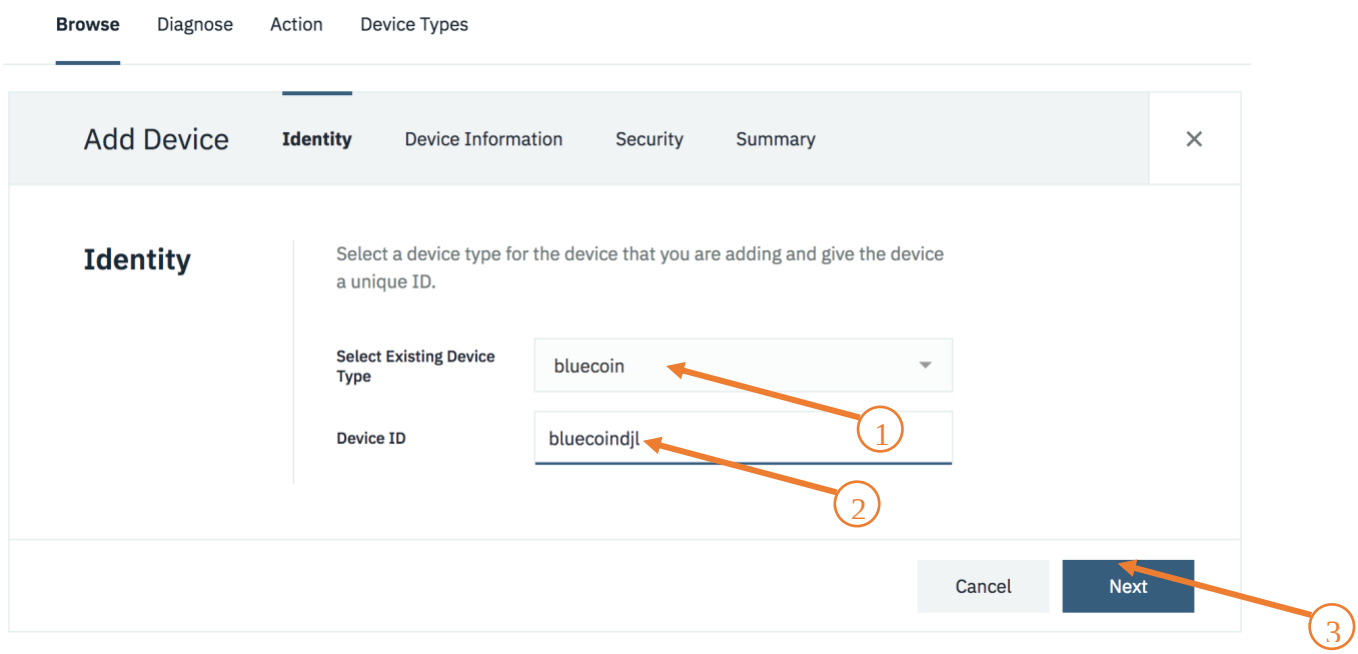
- Give the Device Type a name (1) of *bluecoin* and optionally a description
- Select **Next** (2) and on the next page optionally fill in information related to the BlueCoin and then select **Done**.

Step 3.2.3 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

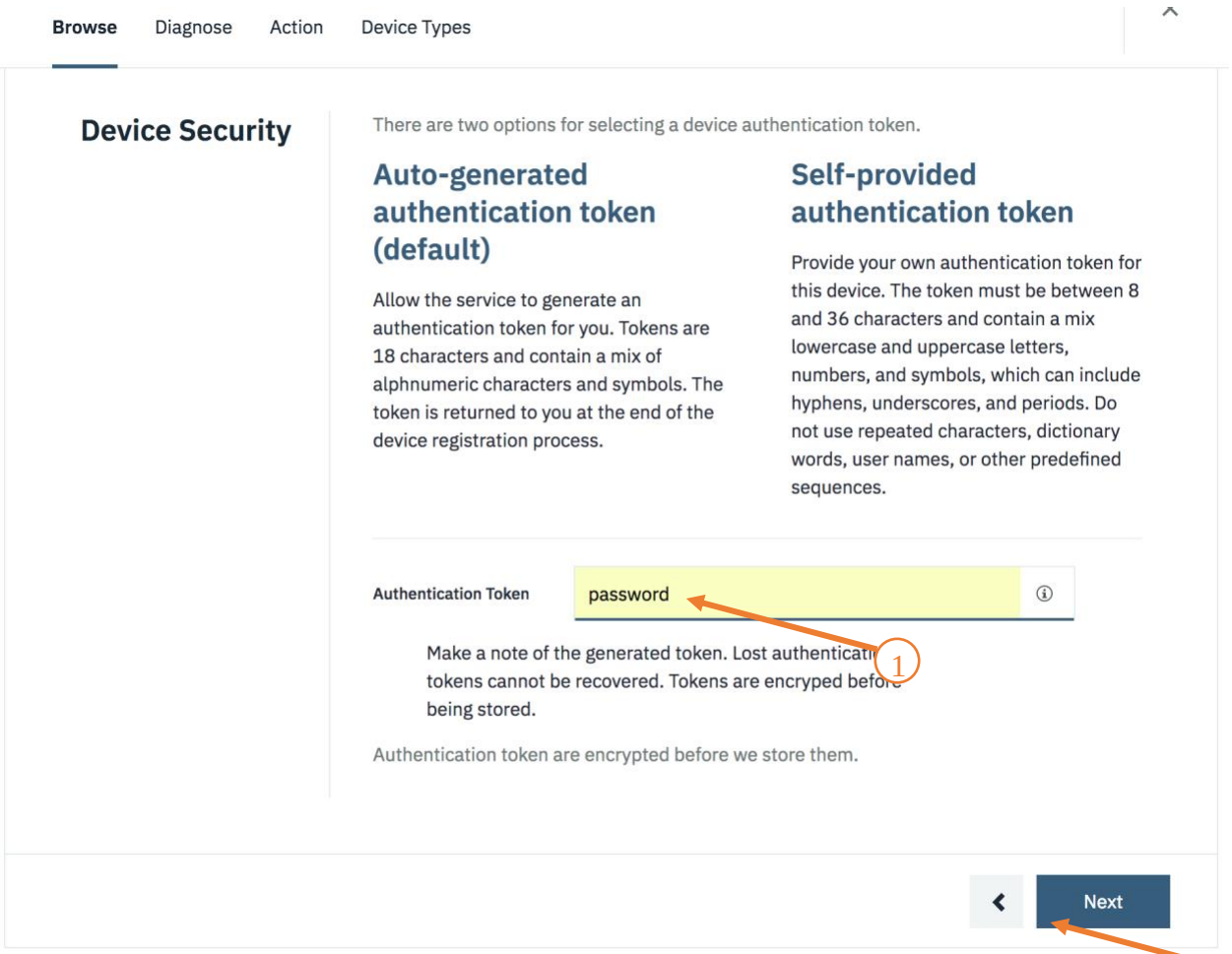


- Select **Register Devices (2)** to open the dialog to register your device

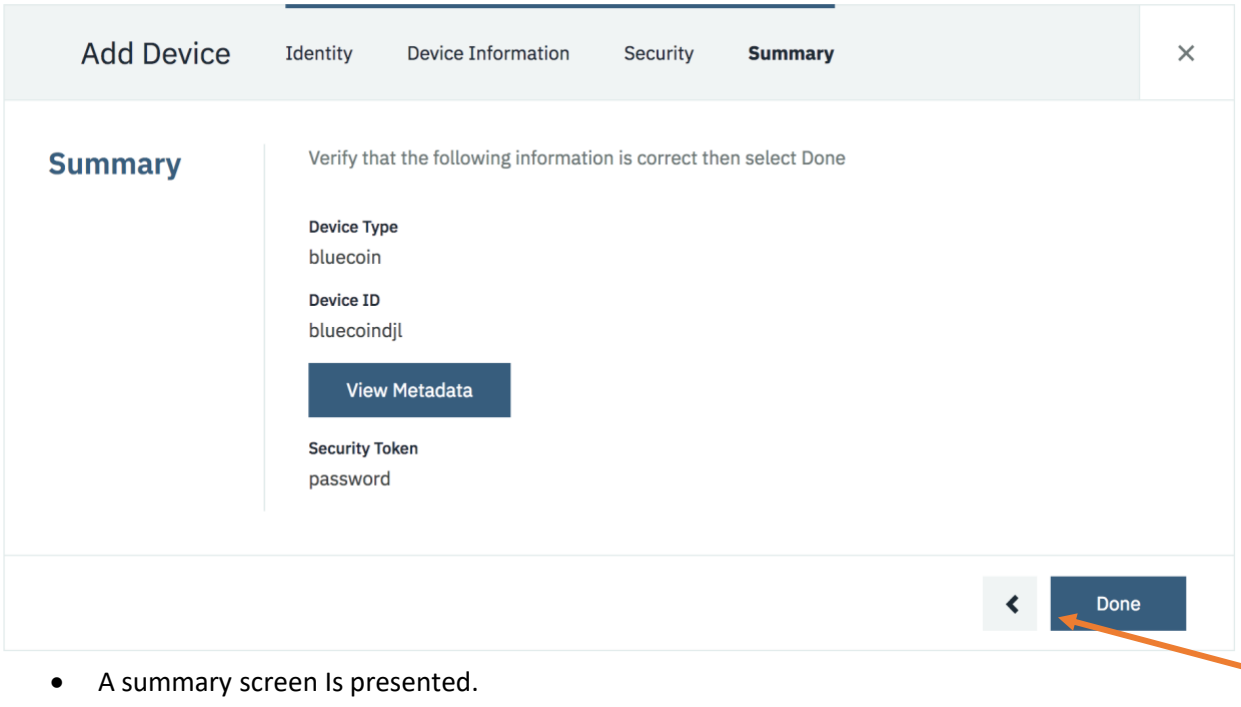


- Select the device type you just created (1)
- Every device registered to an instance of WIoTP must have a unique name. For instance, a serial number or a MAC address make good unique identifiers. Fill in the **Device ID (2)**, for the lab a name like bluecoin<*yourinitials*> will work but in real world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



- 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 identity it and to trust it. Either a certificate or a Token can be used for this purpose.
- Fill in an **Authentication Token (1)** that is memorable. It is important to remember the token as 1) it will be needed later 2) it is non-recoverable once the registration process is completed.
- Select **Next (2)**



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

Step 3.2.4 Note the org id for WIoTP

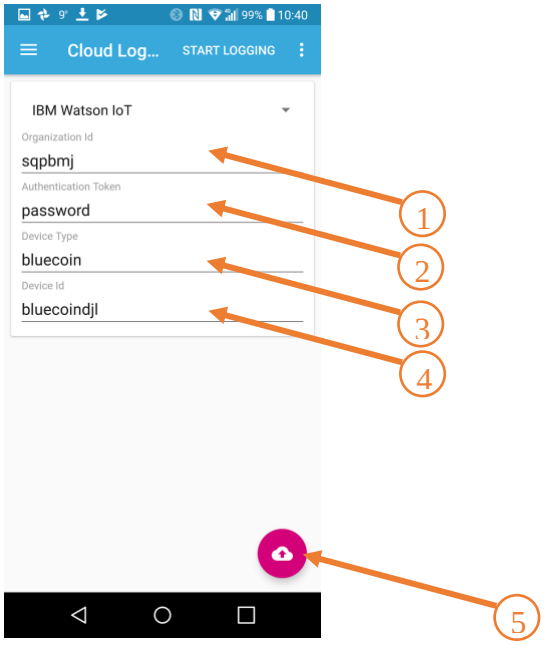
Every instance of WIoTP is uniquely identified by an organization identifier or org id for short. When connecting a device, the org id will be configured on the device.

- One place to find the org id is to look at the url for the WIoTP dashboard. It will look like :- 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.

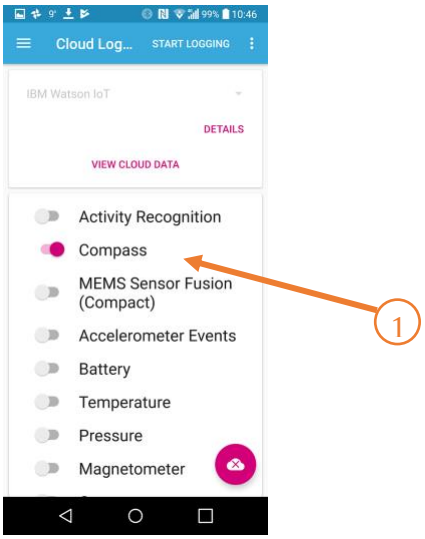
Step 3.3 – Configure the ST BlueMS mobile application to send data to WIoTP

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

- Open the ST BlueMS Mobile Application, discover and connect to your BlueCoin, then 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 sent 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 (1)** Once selected the app starts sending the data to WIoTP

Step 3.4 – View the raw data in WIoTP dashboard

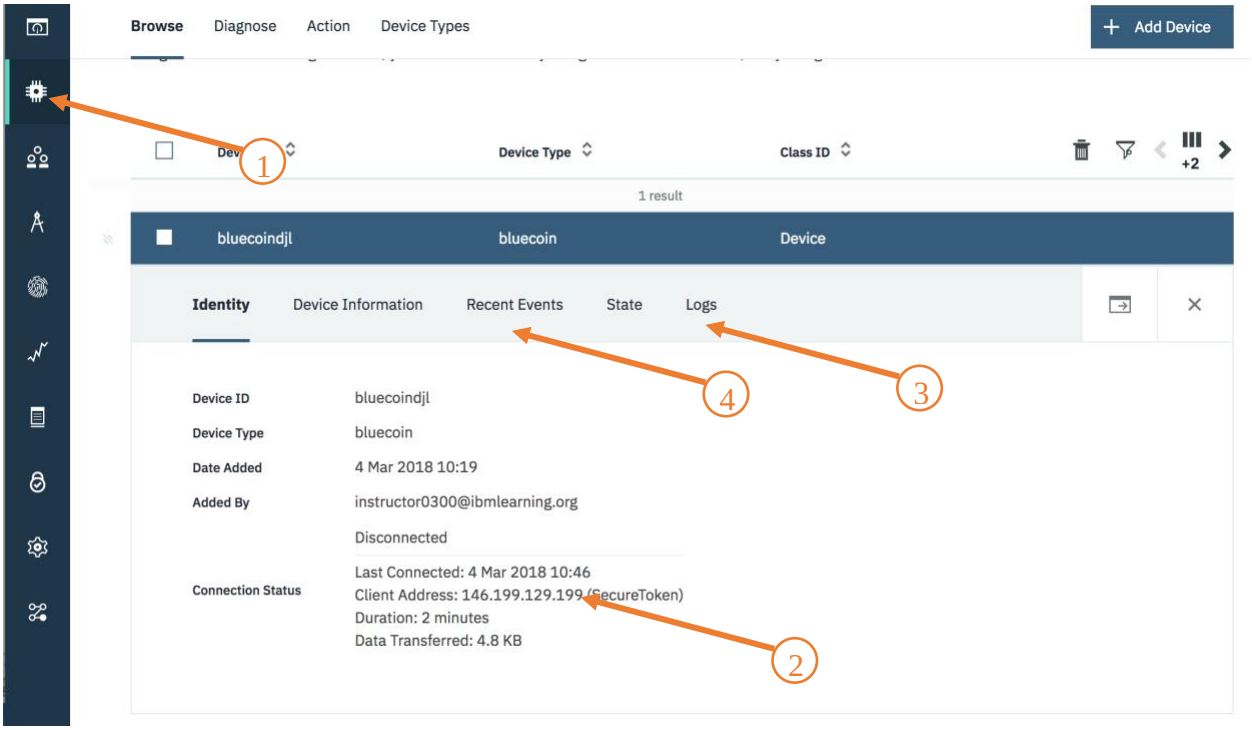
Once data is being sent to WIoTP it can be viewed in a few ways including: -

1. Raw data in the WIoTP dashboard
2. Create an administration dashboard with UI widgets inside of WIOTP

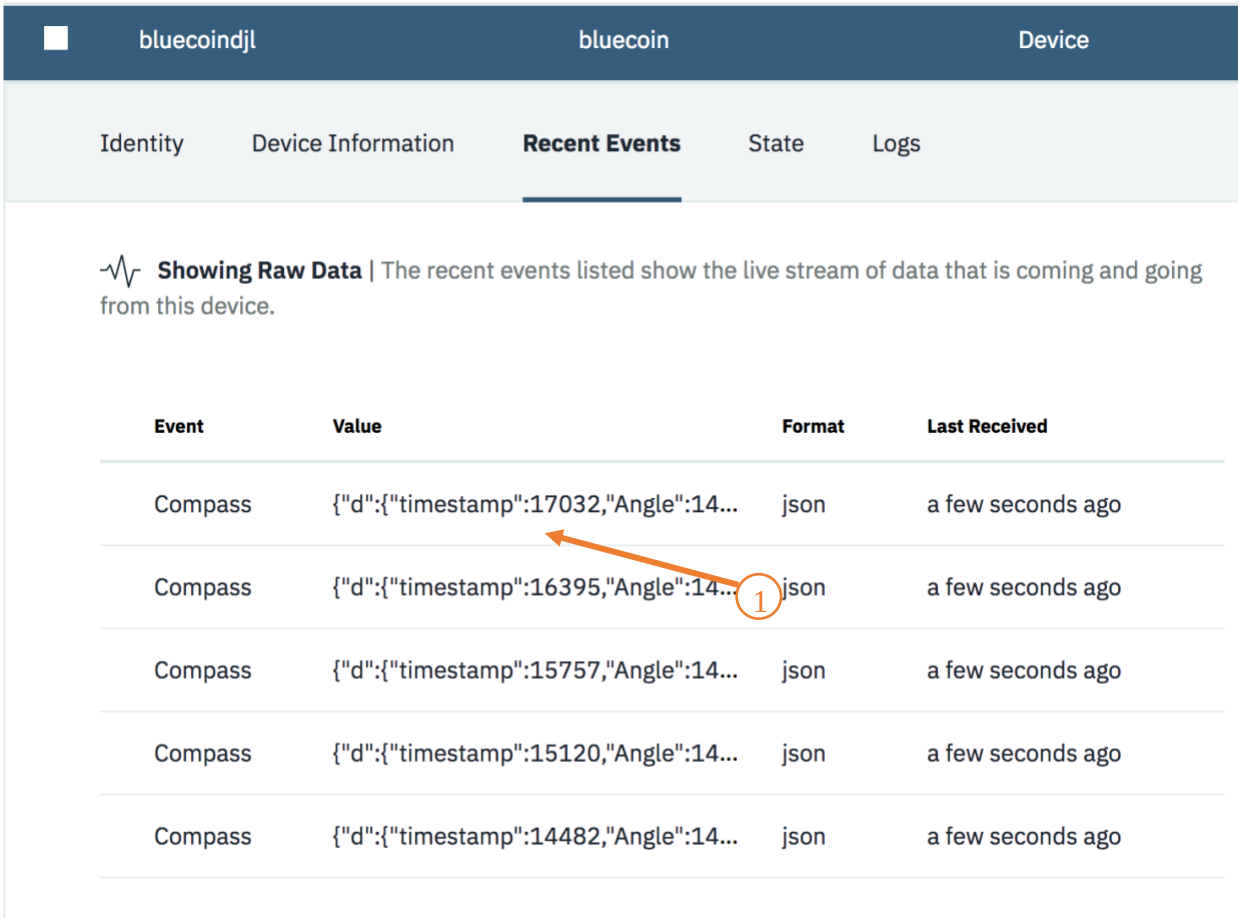
3. Build an application to view the data outside of WIOTP

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.

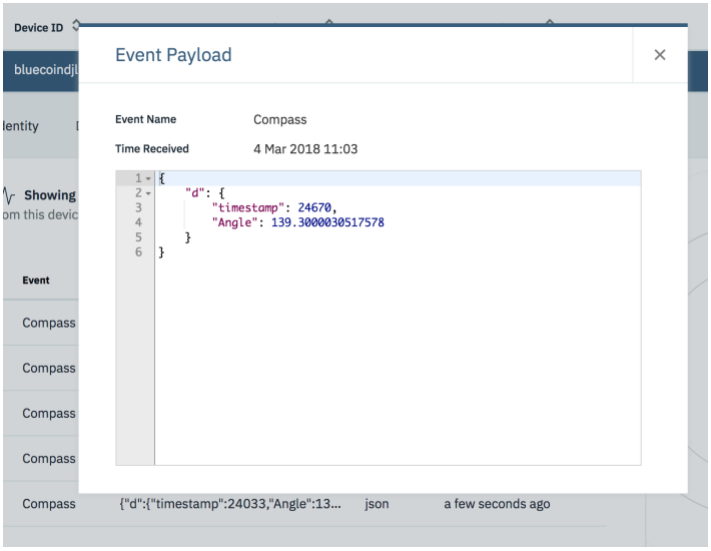
- Head back to the WIoTP dashboard and select the devices view (1)
- Then select your BlueCoin device



- 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.



- As an event arrives it is displayed as the first item in the list of events. Even though it is called recent events the view only displays new events as they arrive from a connected device.
- Select one of the events (1) to see the payload of the event. For a Compass event there are two properties, the time the compass reading was taken, and the Angle. The angle will be used later in the lab

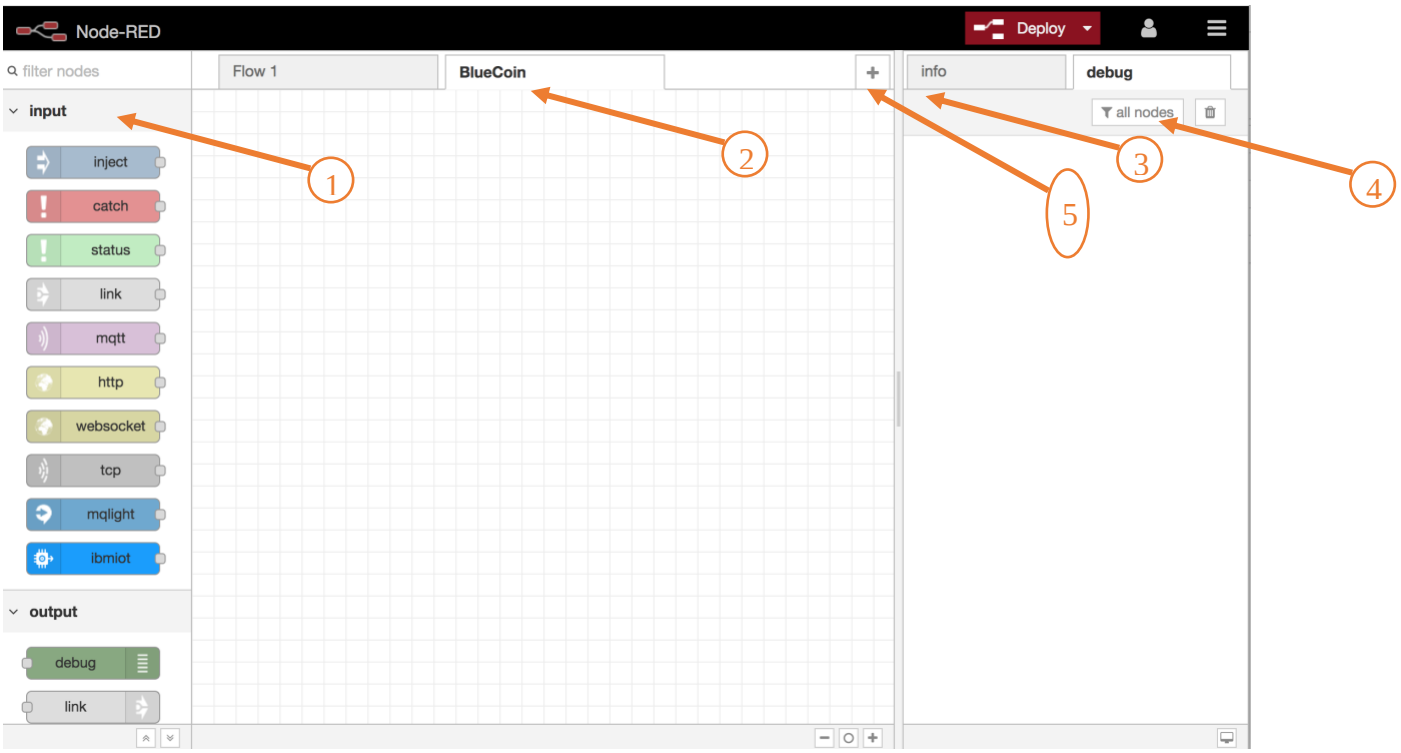


Section 4 – Create a Hello World app in IBM Cloud

Before building an IoT app let's get use to the Node-RED development environment and build a simple hello world application.

Step 4.1 Node-Red intro

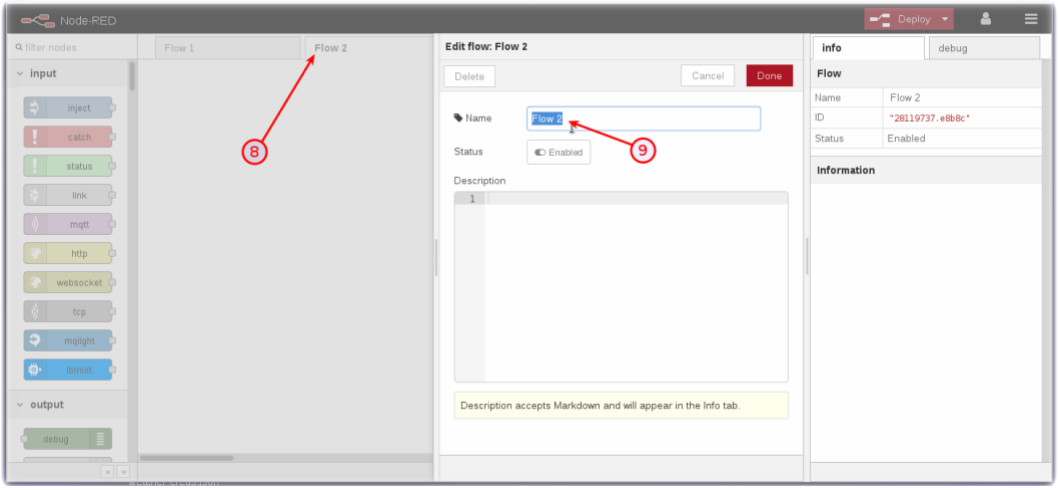
- Head back to you Node-Red development environment from Section 2, Step 3



- 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 but with the ability to install other node of which that are thousands that that have been contributed by IBM and the community. A little later in the lab we will install some new nodes.
- We will build our HelloWorld app in its own **Flow** (2)
- 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.

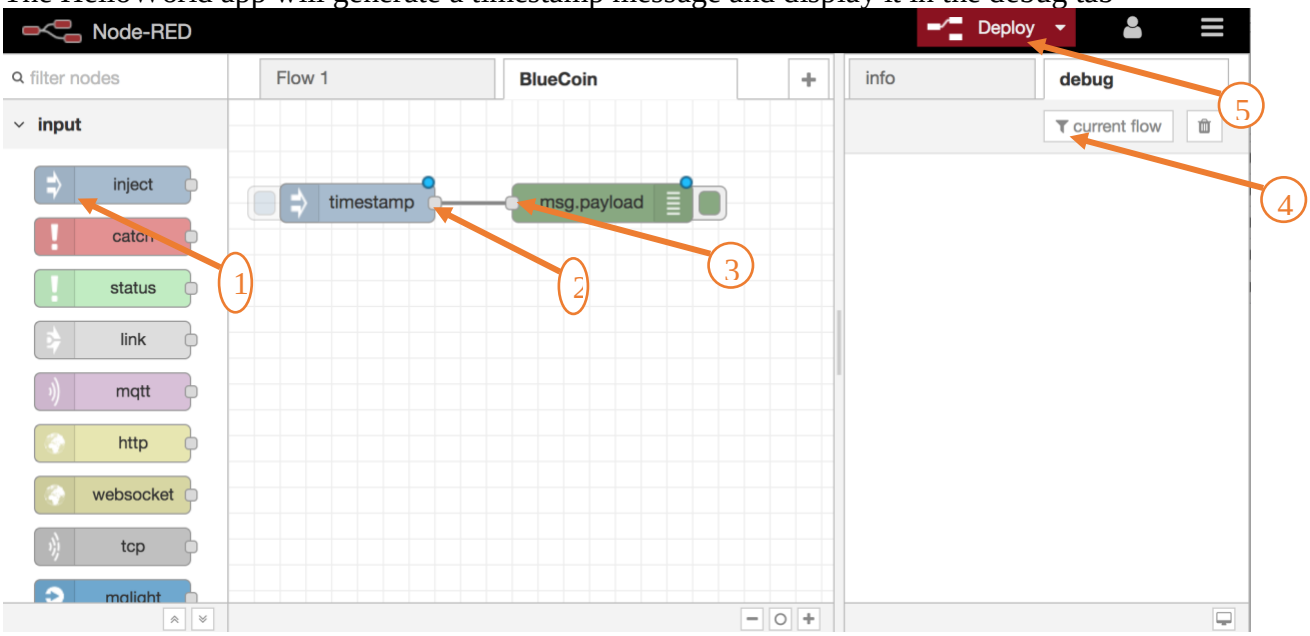
Step 4.2 Create a new Flow

- A Node-RED application can be split into multiple parts where each part is constructed in its own **Flow**. Click the **+** icon (5) to add a new tab.
- Double Click on the **Flow 2** tab header (8).
- Rename this tab from **Flow** 2 to **BlueCoin** (9)



Step 4.3 Build a HelloWorld app

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

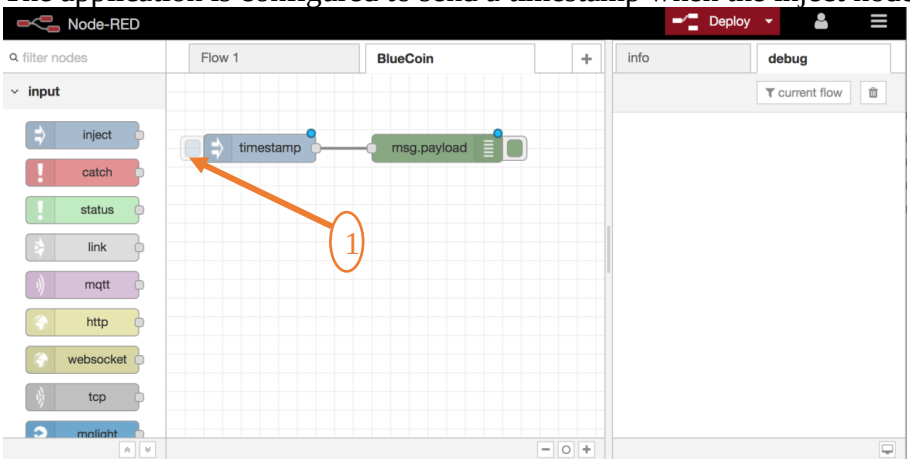


- Drag an **Inject (1)** node from the Input palette onto the BlueCoin flow. When the node is dropped onto the flow its name may change. Node-RED will choose a name based on the purpose of the node, but the name can be manually changed by editing the properties of the node.
- Drag a **Debug** node from the Output palette onto the flow.
- Wire the output of the Inject node (2) to the input of the Debug node (3). To do this select the output of the inject node (2) and drag the cursor to the input of the debug node. This will create a wire through which message will flow.

- The application is now ready to deploy. Select **Deploy (5)** and the HelloWorld application is made live

Step 4.4 Test the HelloWorld app

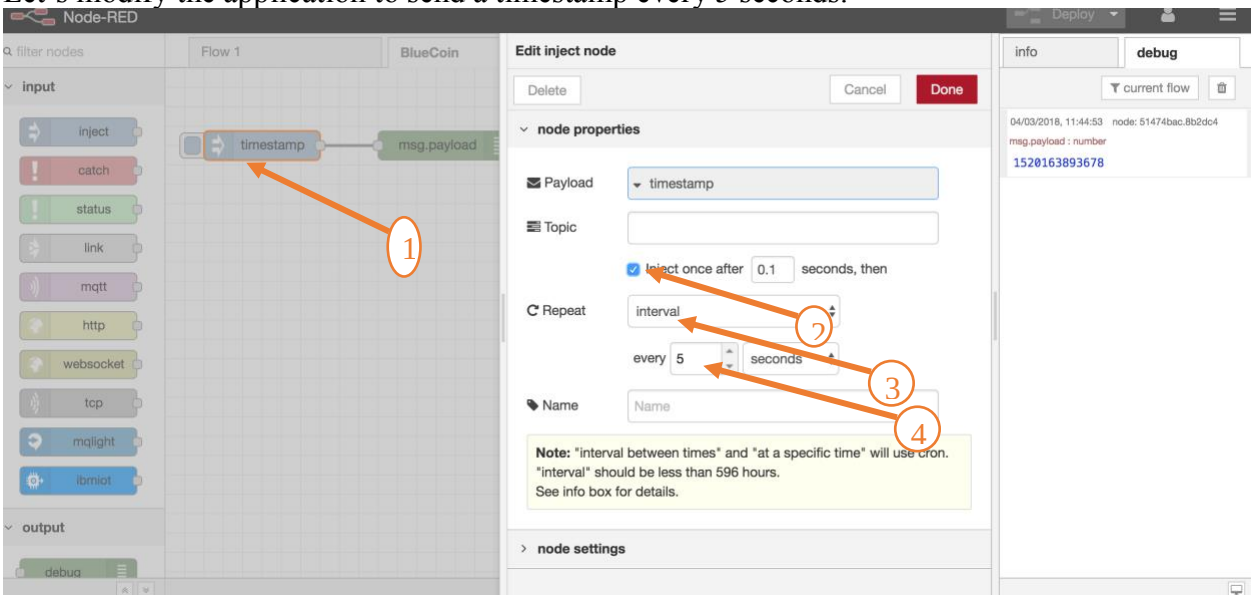
The application is configured to send a timestamp when the inject node is triggered



- Select the icon on the left-hand side of the inject node (1). This will trigger the inject node to send a timestamp. The timestamp will be sent via the output node along the wire to the input of the debug node. Ensure the debug tab is selected, on this tab the timestamp output by the debug node will be displayed

Step 4.5 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 4.6 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,

- Double click the "BlueCoin" title / label of the flow.
- Select **Delete**

Completed HelloWorld app

- For info 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. In Node-RED open the menu (top right, three horizontal bars), select **Import** then **Clipboard**. Paste the clipboard into the dialog and then select **Import**. Drop the resulting nodes and wires onto the flow. Then **Deploy**

Section 5 – Create a User Interface to Visualize Data from the BlueCoin

One of the first stages when developing an IoT application is to find a way to visualize data from the device. In this section we will build a dashboard to visualize the data from the BlueCoin. There are a set of dashboard nodes for Node-RED that will be used to build the user interface.

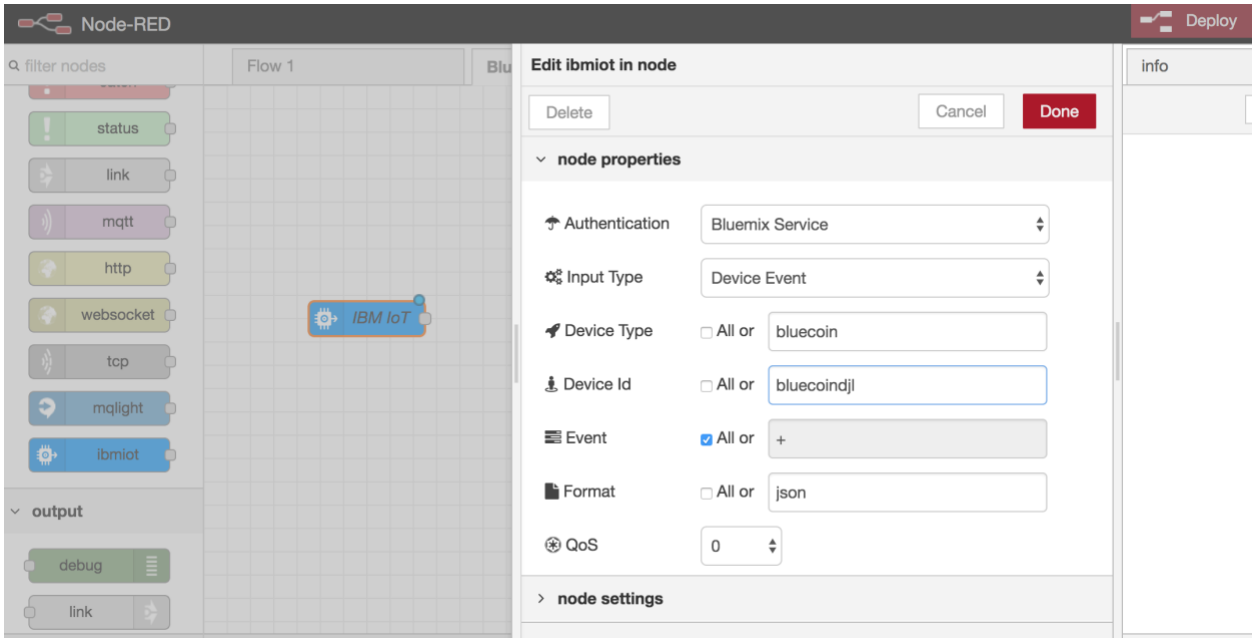
Step 5.1 Create a new Flow to develop user interface in

- Follow step 4.2 and create a new flow called BlueCoin

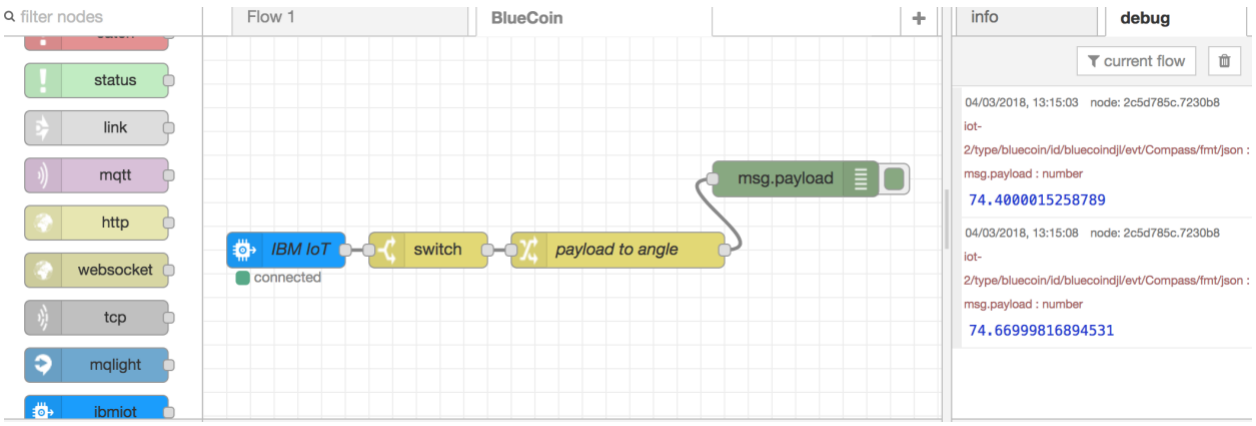
Step 5.2 Receive the compass data from the BlueCoin in Node-Red

Before building the user interface, lets setup Node-RED to receive data from WIoTP

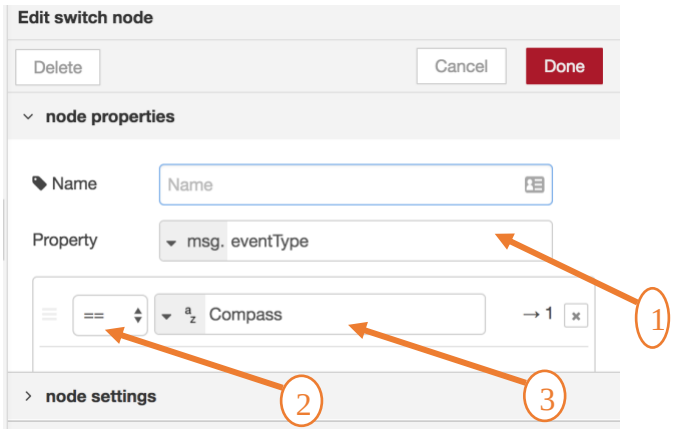
- In the **input** category of your Node-RED palette, find the **ibmiot** node and drag it onto your flow. The ibmiot input node is used to receive data from WIoTP and hence the device.



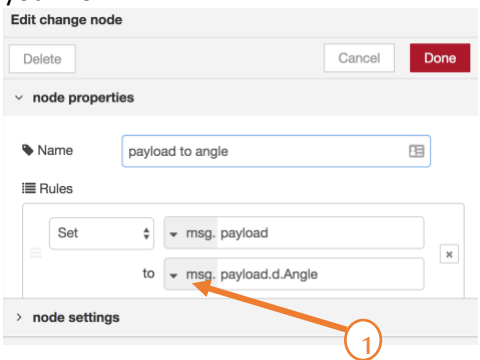
- Double click on the **IBM IoT** in node to configure it
    - Change **Authentication** from QuickStart to Bluemix service. This configures the IoT node to receive data from the WIoTP instance created with the boilerplate.
    - Change **Device Type** to the device type created earlier (bluecoin). Alternatively, as there is only one device used set the **All** checkbox
    - Change **Device Id** to your chosen device id. Alternatively, as there is only one device used set the **All** checkbox
- Click on the **Done** button



- In the **function** category of your Node-RED palette, find the **switch node** and drag it onto your flow. The switch node will be used to select just messages / events that originate from the compass and ignore messages from the other sensors.
- Double click the **switch** node

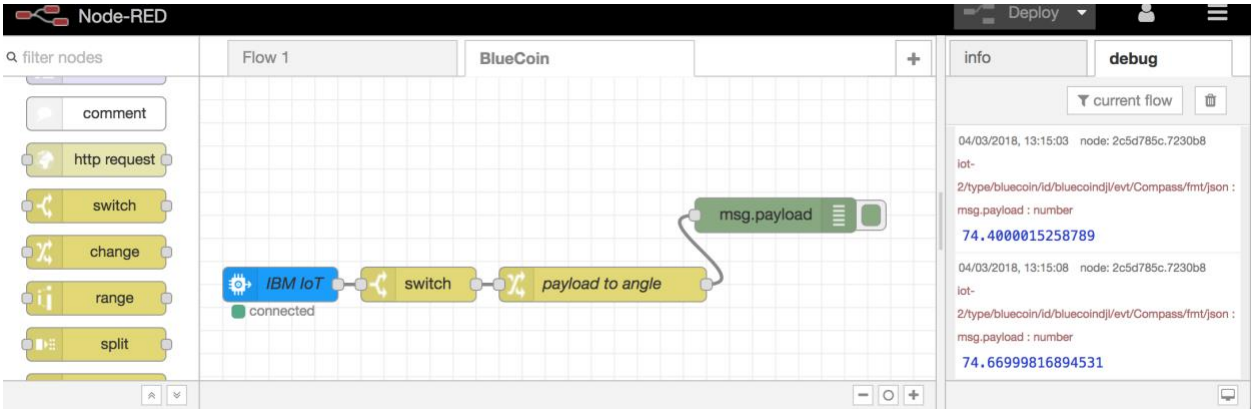


- Configure the switch node to pick out events of type Compass
  - Change Property to **msg.eventType** (1)
  - Each message received from a device is classified by type. For the BlueCoin each type of sensor has a specific event type. For the compass it is *Compass*. Make a selector using **==** comparison (2) and string of **Compass** (3) in order to only send compass events to output 1.
    - Other event types include; - Pressure Accelerometer, Accelerometer_Events and Temperature
  - Click **Done**
  - This will create a selector node with one output that will select and output only events of type Compass
- The payload of an event/message often contains more data than is needed, also some nodes require a single property for their input, this is the case for the dashboard nodes. A change node can be used to manipulate properties, in this case it will be used to extract just the Angle property from the message. In the **function** category, find the **change** node and drag it onto your flow.



- Double click on the **change** node to open the configuration dialog
  - Configure the **Rules** by clicking on the "**a/z**" dropdown of **to** (1) and set it to msg**. payload.d.Angle.** (msg. is the in the dropdown and payload.d.Angle needs to be entered in the text box)
  - Optionally give the node a name "set payload to Angle"
  - Click on the **Done** button.
- In the **output** category, find the **debug** node and drag it onto your flow.
- Wire the four nodes together. IBM IoT node to switch, switch to change and change to debug

- Click the Deploy  button on the top of menu bar to deploy the Node-RED flow.
- Select the **debug tab** on the right sidebar of your Node-RED flow.

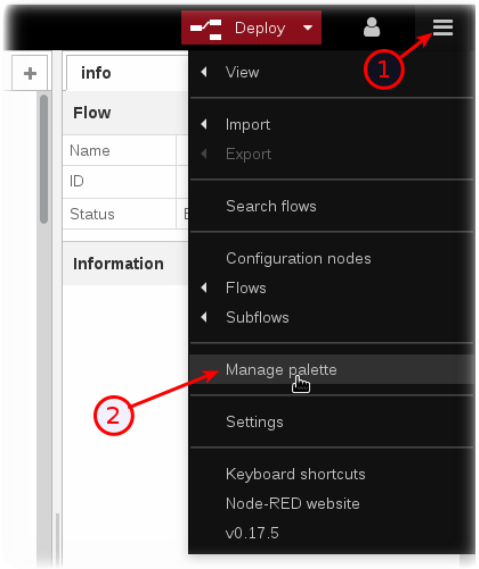


- Ensure the BlueCoin is connected and the ST BlueMS app, CloudLogging feature is set to send Compass events. The debug tab will display the compass angle as detected by the BlueCoin. By default, a new angle will be seen every 5 seconds.

Step 5.3 Install the Node-RED dashboard nodes

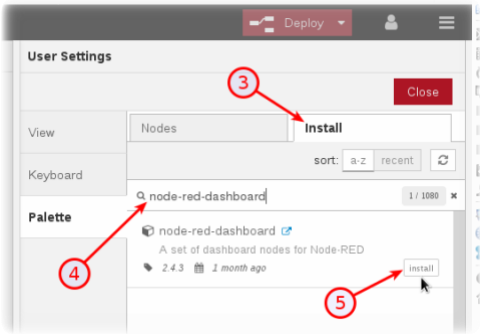
The IoT Starter Application deployed into IBM Cloud includes a small subset of Node-RED nodes. The Node-RED palette can be extended with over one thousand additional nodes for different devices and functionality. These NPM nodes can be browsed at http://flows.nodered.org

In this Step, you will add the Node-RE**D 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)
- Select the **Install** tab (3), type **node-red-dashboard** (4) in the search box
- Press the **Install** button (5) for the node-red-dashboard nodes.



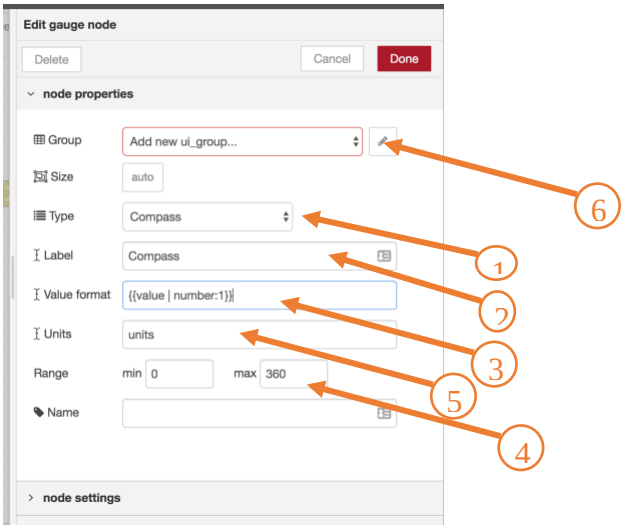
- Press the **Install** button in the next dialog.
- A set of **dashboard** nodes are now available in the palette to help create a graphical user interface. You will need to scroll to the bottom of the palette to see them.

Step 5.4 Create a simple dashboard

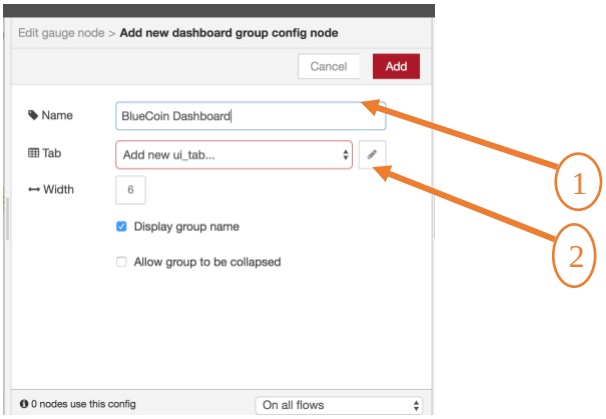
The dashboard nodes will be used to create a simple user interface that graphically displays the compass data

- Drop a **gauge** node onto your flow
- Wire the output of the **change** node to the input of the **gauge** node

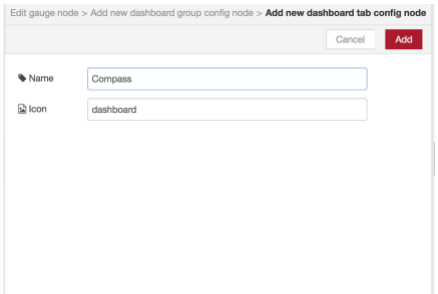
- Double click on the gauge node to open the panel to configure the node



- 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) The :1 limits the display of 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** (1)
- Select the edit icon next to Add new ui tab (2)

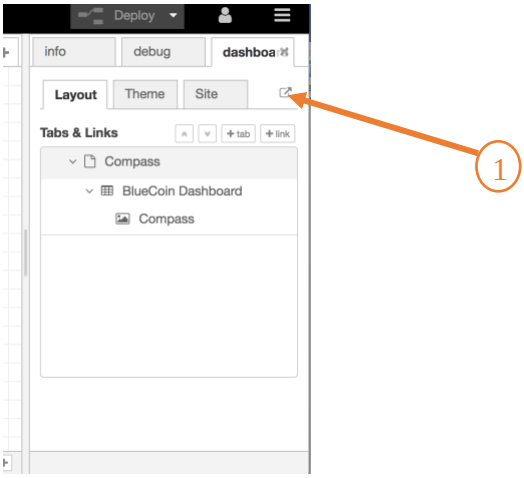


- Give it a name of **Compass**
- Select **Add** to finish the creation of the Compass tab

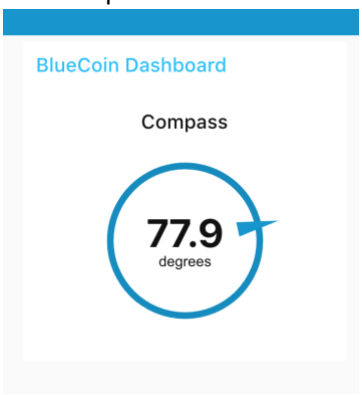
- Select **Add** to finish the creation of the BlueCoin group
- Select **Done** to complete the creation of the Compass gauge
- **Deploy** the application
- The “simple” dashboard is now deployed.

Step 5.5 Launch the dashboard

- There are several ways to launch the dashboard. The simplest is from the **Dashboard** tab that has appeared next to the debug Tab



- Select the **launch icon (1)**
  - Alternatively, in a new browser tab, enter a url of https://*bluecoin-djl*.mybluemix.net/ui/ where bluecoin-djl is the name of you IBM Cloud application
- The compass dashboard will display and look like this: -



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

Step 5.6 Change the frequency sensor events are sent to the cloud

The Blue MS application sends data to the cloud at a default rate of 1 event every 5 seconds. This limits the responsiveness of the user interface. The frequency can be changed in the mobile app

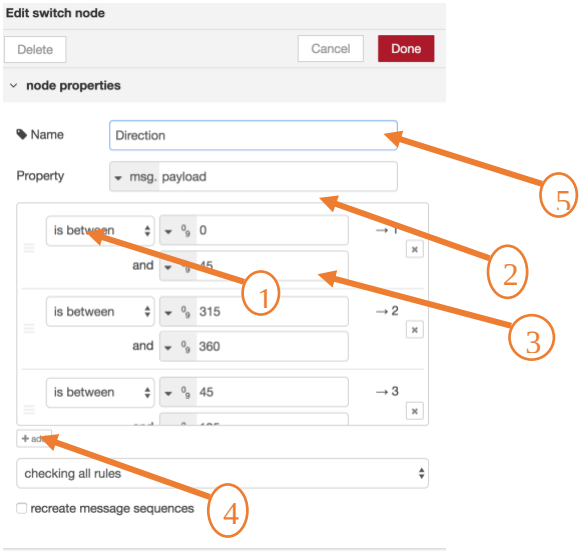
- Select the **Cloud Logging** page of the mobile app. Make sure the the app is not currently sending data to the cloud.
- Select the menu (three dots in top right corner on Android)
- Select **update interval** and select 0.5 seconds to send 2 events every second

- Start sending the data to cloud by pressing the cloud icon
- The compass user interface will now update pretty much in real time.

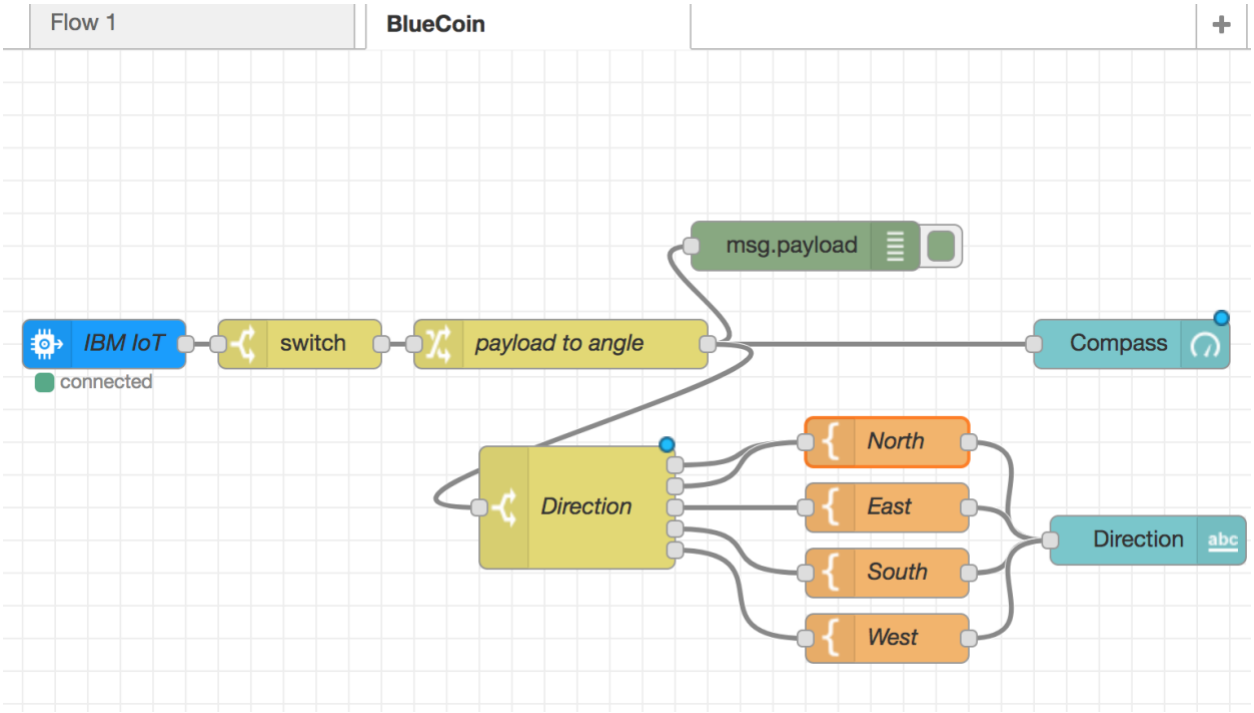
Step 5.7 Add direction logic to the application

In this next step some simple logic will be added to the application to determine if the compass is baring North, East, South or West based on the raw Angle reading from the sensor

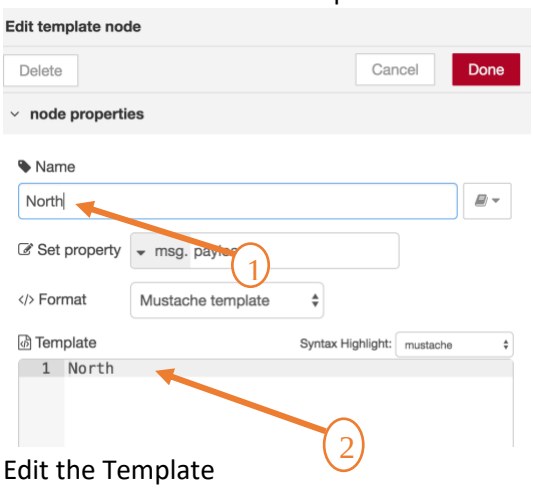
- Drag a **switch** node from the function palette to the flow
- Wire the output of the change node that outputs the angle to the input of the switch node
- Double click on the switch node to open the configuration panel for the node



- Configure the selector to determine if Angle equates to North:
  - Change the **name** to **Direction**
  - Change the selector type from **==** to **is between** (1)
  - Set the from value to **0** (2)
  - Set and (to value) to **45** (3)
  - Select the **+ add** button to add the next selector
- Configure a set of selectors following the previous step with the following ranges
  - **315** to **360** equates to North
  - **45** to **135** equates to East
  - **135** to **225** equates to South
  - **225** to **315** equates to West
  - Click **Done**
- Drop a **Template** node from the function palette onto the flow



- Wire the **Direction** output 1 to the input of the template
- Double click to edit the template



- Edit the Template
- Set the Name to **North**
- Set the template content to **North**
- Create a Template for **East**, **South** and **West** and configure as for North but with appropriate content.
- Wire up the remaining outputs from the Direction node the relevant template. Output 2 to North, 3 to East, 4 to South and 5 to West
- Drag a **text** widget (not text input) from the dashboard palette and drop on the flow
- Wire the output of each of the 4 direction Templates to the input of the Text widget.
- Edit the **text** widget and change the label to **Direction**
- **Deploy** the application
- Launch the dashboard is not already opened.

- Rotate the BlueCoin, as well as displaying a compass with the value in degrees it will also display the compass baring of North, West, East or South.

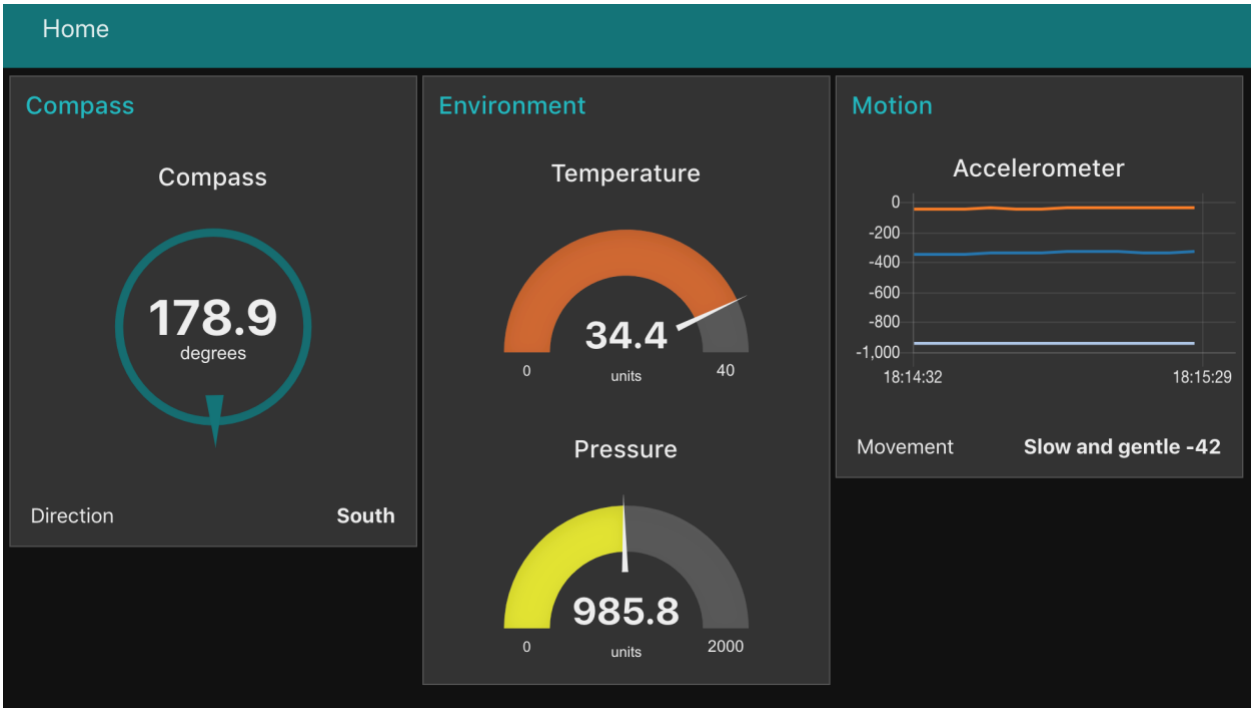
Completed Compass Dashboard app

- For info, 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. The clipboard can then be imported into a Node-RED flow using the Import Clipboard menu
- After importing the flow edit the ibmiot input node to use your device type and id

Rich Example Dashboard application

Optional, a more sophisticated dashboard application is provided as an example

- The ST BlueMS Cloud Logging page must be configured to send the following sensor data
  - Compass
  - Accelerometer
  - Temperature
  - Barometric Pressure
- WIoTP has a max inbound message rate. To ensure this is not exceeded, set the frequency / interval that the ST BlueMS application sends data to 5 secs
- Import the dashboard from https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows as richdashboard.flow
- **Deploy** the flow.
- The dashboard should now look like this: -



Section 6 Invoke an External Service – Slack

In this section we will enhance the application to talk to a service that is external to the IBM Cloud. There are a massive variety of services to choose from, for this lab Slack has been chosen. The application will be modified to both receive and post messages to Slack. External services can be invoked from 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 be 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.

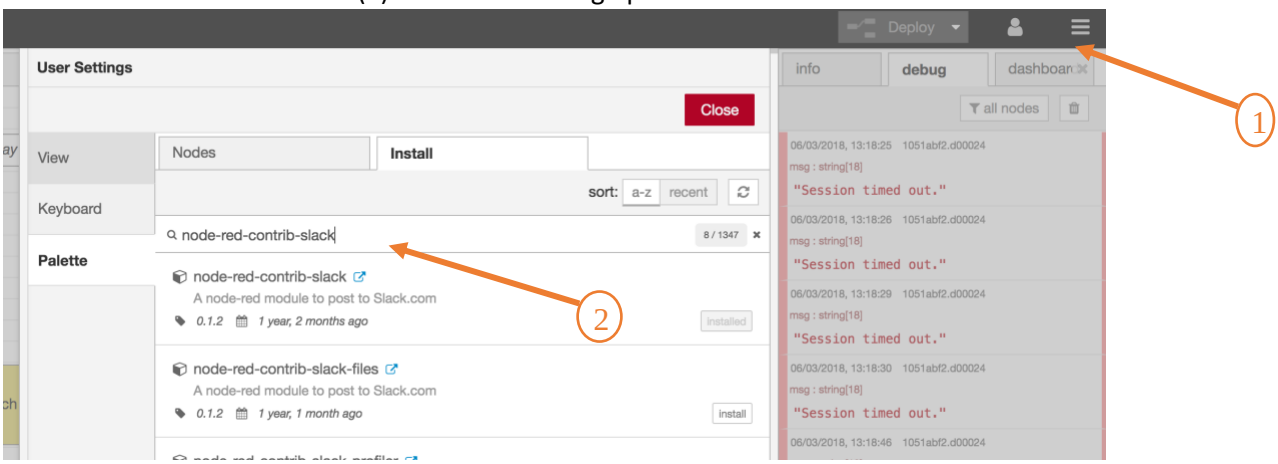
For Think 2018 a slack channel called nodered_test has been setup. It has been configured to accept posts from a program (Node-RED) via a webhook and for a bot to receive messages. Once the Think conference has completed access to this slack channel will be disabled. To setup slack integration with your own channel follow the slack instructions for setting up a WebHook and a Bot.

- *Slack Bots: https://api.slack.com/bot-users*
- *Slack Webhooks: https://api.slack.com/incoming-webhooks*

Note: Testing has shown that the 256Mb available in an IBM Lite account runs out of steam around this point. Installation of the Slack nodes may tip the memory use over the 256Mb limited. If possible use an account / org with more memory.

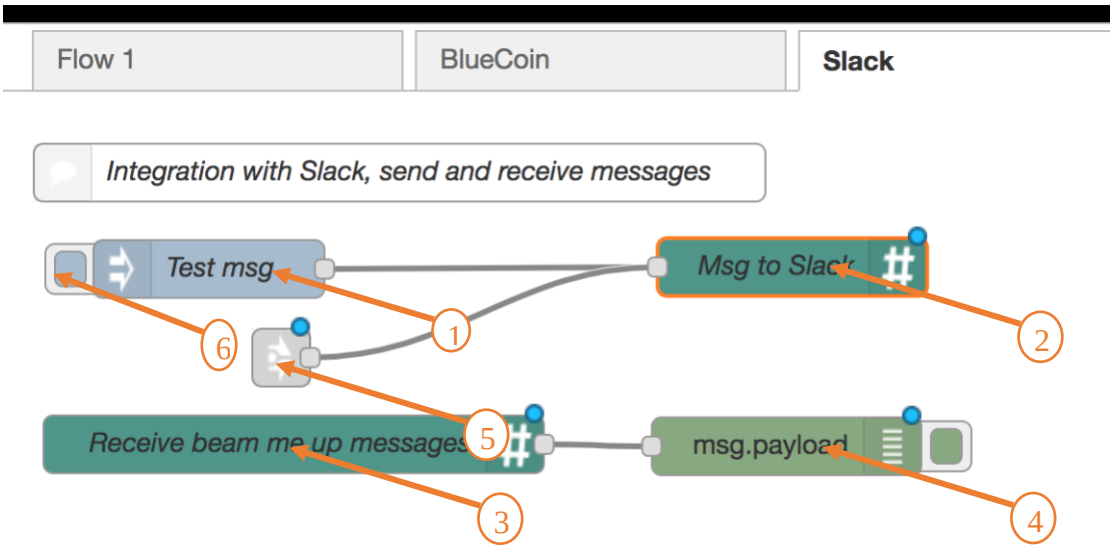
Step 6.1 Install Slack Node-RED nodes

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



- Select the **install** tab
- 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

Step 6.2 Configure a flow to send and receive Slack messages



- Create a new flow called Slack
- Drop an **inject** node (1) onto the Flow and configure it to send a String, make the message in the String something that you will recognize as the message is going to a slack channel that is shared
- Drop a **slack** (2) node onto the flow and wire the output of the inject node to the input of the slack node.
- Double click the slack node and change the **WebHook URL** to https://hooks.slack.com/services/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 Put your name in the Posting UserName
- Drag a **link (5)** node from the input category and drop onto the Slack flow
- Edit the Link node and give it a name of **SendToSlack**

- Wire the output of the Link node to the input of the Slack output node.
- Drop a **slack bot in (3**) node from the social group of nodes onto the flow
- Double click the slack bot in node and configure
    - **Bot API Token** to
        1. xoxb-32
        2. 7821318
        3. 007-0FRO
        4. 7zMdDUhsZ
        5. cDkdfXYp1cY
    - where the 5 parts of the string need to be concatenated together into a single string starting with X and ending with p
    - **Channel** to **nodered_test**
    - **Name** to Receive beam me up messages
- Drop a **debug** (4) node onto the flow
- Wire the output of the slackbot node to the input of the debug node.
- The flow is now set to receive all messages that are published to the nodered_test channel and to publish a test message to the channel

Step 6.3 Send a test message

- To publish a test message, press the button on the left-hand side of the **inject node** (6)
- Your test message will be displayed in the debug tab.

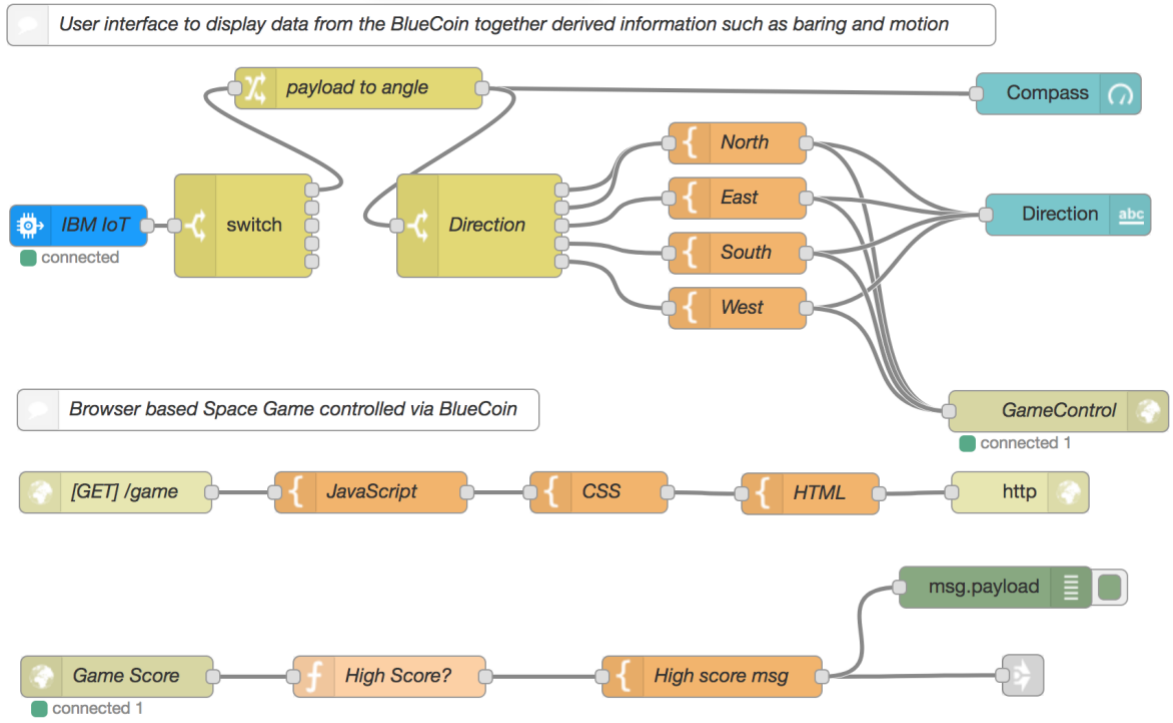
Completed Slack Flow

- For info, a completed version of the slack flow can be found at https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows as Slack.flow
- This can be imported into Node-RED by copying the contents of the file to the clipboard. The clipboard can then be imported into a Node-RED flow using the Import Clipboard menu
- The api token contained in the imported flow may not be valid. If this is during the Think 2018 conference, the token in the section above can be copy and pasted into the SlackBot input node.

Section 7 Control a Game using the BlueCoin

In this section a space themed game will be loaded. The game is written as a web application. Node-RED will act as the web server, enabling the game to be loaded into a web browser. The BlueCoin will be used as the game controller to control the space ship flying around the screen.

After completing this section, the BlueCoin flow will look like this



Section 7.1. Import the game into Node-RED

- Select the BlueCoin flow that contains the nodes receiving data from the BlueCoin, that extracts the compass data and displays it on the dashboard.
- Copy the contents of file Game.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** followed by **Clipboard**
- Paste the contents of the clipboard into the **paste nodes here text** input field and then select **Import**
- Drop the nodes onto the **BlueCoin** flow.

Section 7.2. Connect the compass directions to the game

The Compass will be used to control a space ship flying around the screen. The North, South, East, West directions derived from the compass need to be wired to the game.

- Wire the output of the North template node to the Input of the node labelled GameControl. This node uses WebSockets to connect the cloud application to the game that runs inside of the browser
- Repeat the previous step and wire the East, West and South nodes to the input of the node labelled GameControl.

Section 7.3. Send the high score to Slack

If a high score is reached, publish the score to slack. The nodes to send messages to Slack are in a different flow. Nodes in one flow can be wired to nodes in another flow using **link** nodes.

- Select and edit the **link** output node (after the high score msg template). Select the **SendToSlack** checkbox. This will link the high score output nodes in the BlueCoin flow to the send a message part of the Slack flow.
- Select and edit the **High score msg** template node. Change <your name> to your name or a game/screen name
- **Deploy** the application

Section 7.4 Play the game

The game is now ready to play. The goal of the game is to fly a space ship around the screen and hit the "Watson" badge as many times as possible. Each time the badge is hit the score is incremented. The game stops when the space ship flies into one of the boundary walls that enclose the game.

- Open a new tab in your web browser
- Enter an address of https://<yourappname>.mybluemix.net/game replacing <yourappname> with the name of your application. The appname is the same as that used in the address of the Node-RED web page and the dashboard web page.
- Ensure the BlueCoin is connected and sending data at an interval rate of 0.5 seconds. This can be validated by opening and viewing the dashboard page from section 5 and watching the compass move as the BueCoin is rotated.
- Start the game by moving the mouse into the game area and clicking the mouse. The space ship will now start to move around the screen. Hold the blue coin horizontally and rotate it to change direction. As it is rotated the compass directions of North, South, West and East are converted to up, down, left or right controls.
- Fly the space ship around the play area and try to hit the Watson badge as many times as possible.
- If a boundary of the play zone is hit the game ends.
- When the game ends the score is sent to the application in the cloud (via a websocket)
- If your highest score is reached it is published to Slack and also displayed in the debug tab.
- What happens when you hit the Watson badge 3 times?

Completed Game Flow

- For info, a completed version of the slack flow can be found at https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows as CompassAndGame.flow
- This flow includes both the compass nodes from Section 5 and the game nodes. Before importing to the BlueCoin flow, delete the contents of the BlueCoin flow leaving an empty palette to paste the imported flow onto.
- This can be imported into Node-RED by copying the contents of the file to the clipboard. The clipboard can then be imported into a Node-RED flow using the Import Clipboard menu

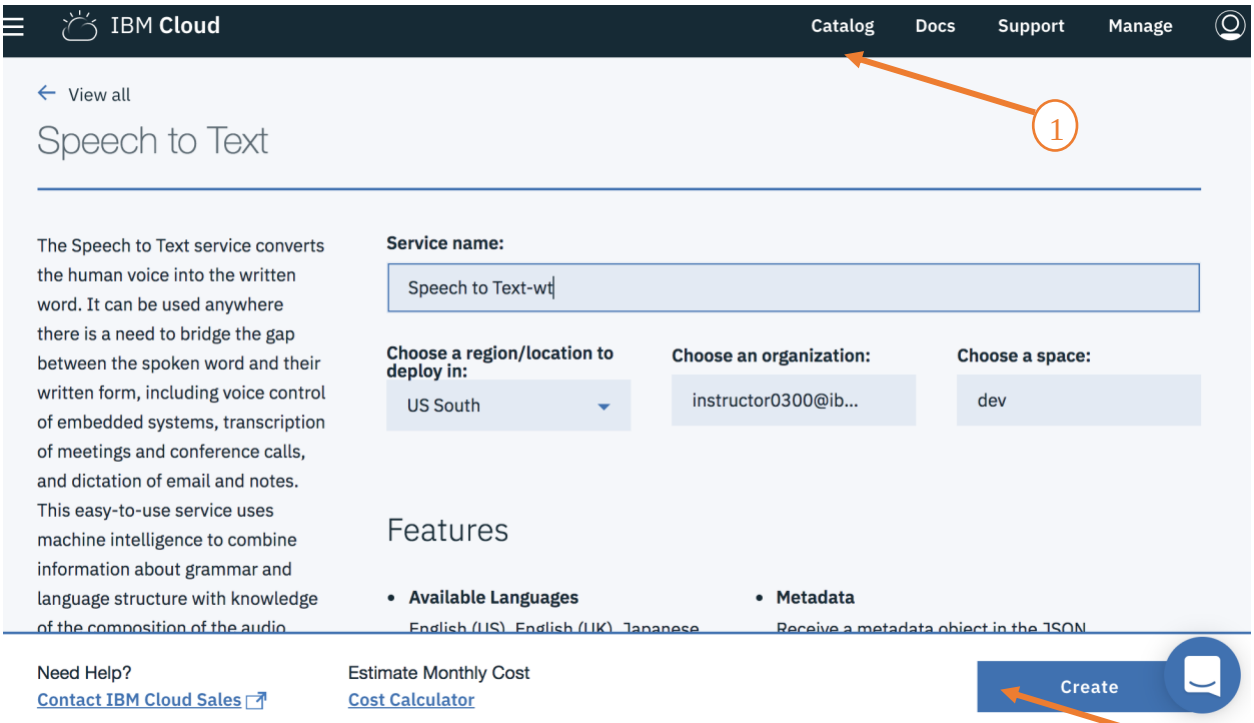
Section 8 Transcribe Speech from the BlueCoin to Text

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

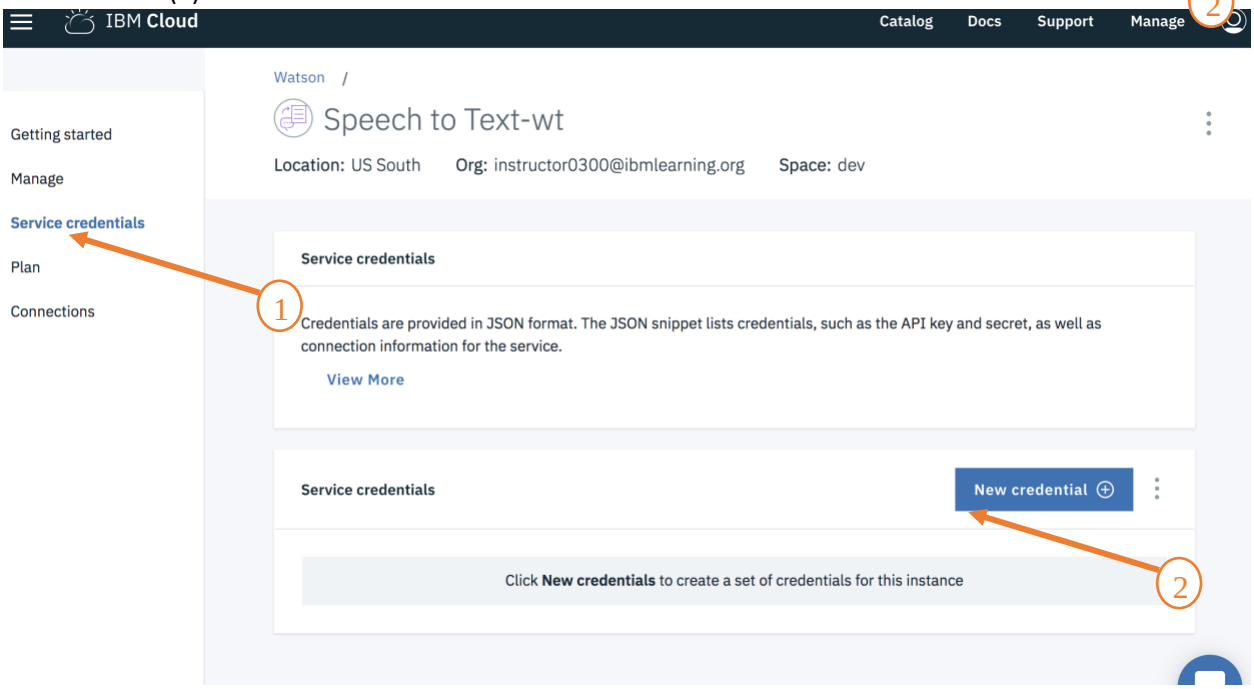
on the BlueCoin will be connected to the Speech to Text service in the IBM Cloud. The transcribed text will be delivered back to the mobile application to display.

Step 8.1 Configure the IBM Cloud Speech to Text Service

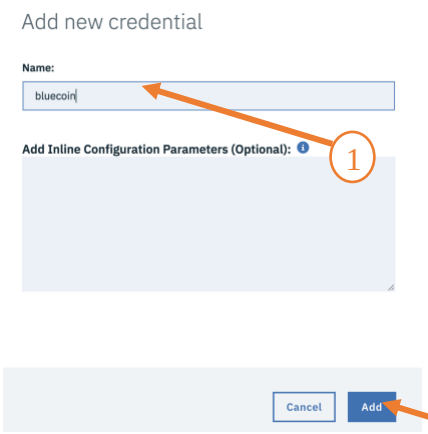
- Head to the IBM Cloud **catalog** (1) and search for speech
- Select the **Speech to Text** service



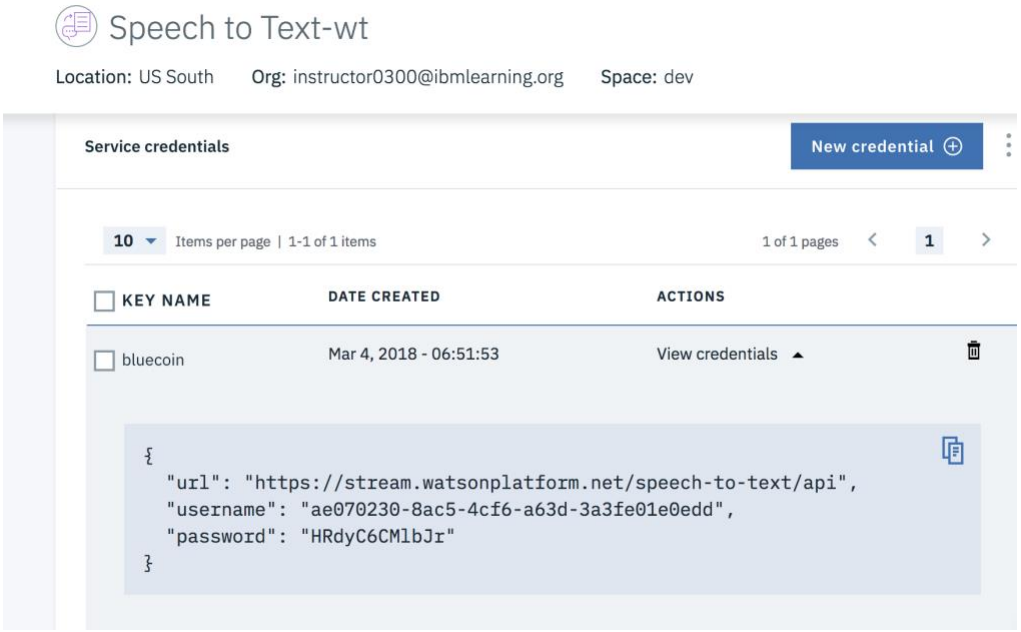
- Select **Create** (2)



- Select **Service Credentials** (1)
- Select **New Credential** (2)



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



- Note the username and password as they are needed in the BlueMS mobile application. The credentials are used to identify and secure the mobile app to the speech to text service.

Step 8.2 Configure the BlueMS mobile app to talk to the cloud S2T service

- Open the ST BlueMS application on your phone and connect to the BlueCoin
- Select the **SpeechToText** page
- Choose the IBM Watson speech to text engine
    - On Android hit **SELECT** and choose **IBM Watson**
- Choose **English US** or **UK** depending on you accent
- Enter the credentials / key generated in step 8.2
    - On Android, select **SET KEY** then enter the username and password from the cloud speech to text view credentials page, ensuring they are entered exactly as displayed.

Step 8.3 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 and puts it in listening mode. 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. Results will vary based on background noise and other factors like network quality.

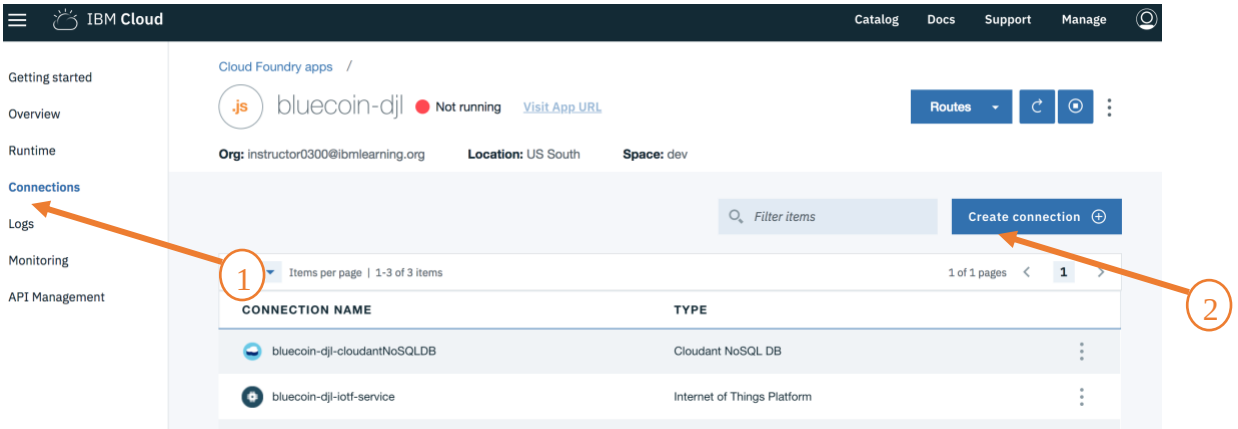
Section 9 Using Voice interaction in the Cloud Application

In section 8, 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 make use of the transcribed text.
This section will use the transcribed text to interact with the IBM Watson Assistant (WA) service. WA enables the creation of digital assistants for personalized and engaging experiences for consumers. It is designed to be easily customizable and integrated into devices, mobile apps, cars, rooms and other connected objects. It knows who it is talking to and understands context from history and real-world sources to enable a delightful conversation and truly helpful recommendations.
In this section our application will be updated to interact with the out of the box version of WA. Only the basic capability of WA is demonstrated in this lab, the power of WA to enable meaningful conversations is a topic on its own.
WA is currently in beta, more info can be found here :- https://developer.ibm.com/iot/watson-assistant/ Access has been provided for use during the Think 2018 conference. Once the conference finishes to access WA follow the link to request access to the WA service.

Step 9.1 Bind the Speech to Text service to the Cloud Application

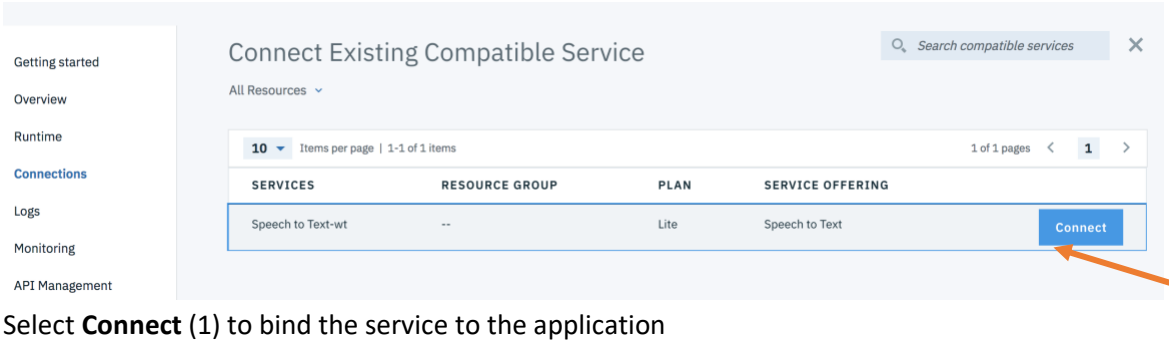
In the IBM Cloud a service can be bound / connected to an application, enabling the application to securely interact with the service. The speech to text service needs to be bound to our application.

- In the IBM Cloud Dashboard select the application



- Select **Connections (1)**
- Select **Create Connection (2)**

- Select the speech to text service created In Section 8

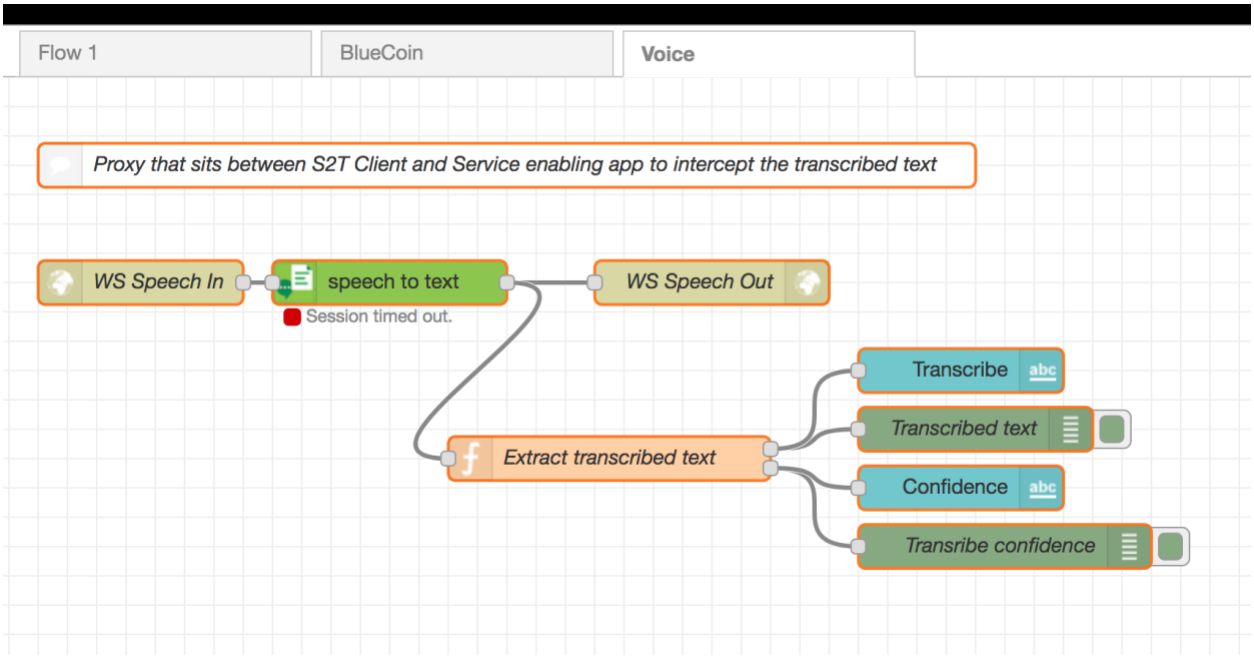


- Select **Connect** (1) to bind the service to the application
- A dialog will pop up, select **Restage.** It will take a few mins to restage the application

Step 9.2 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** followed by **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 9.3 Change BlueMS mobile app to point at the Speech to Text Proxy

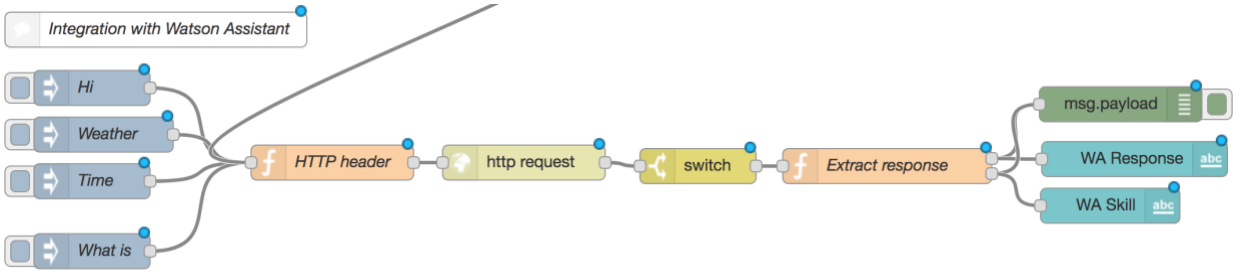
- On the smart phone open the BlueMS app and select the **SpeechToText** page

- 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 8.3. The transcribed text will appear in the Node-RED debug tab and also in the Dashboard. Results will vary based on background noise and other factors like network quality.

Step 9.4 Get started with Watson Assistant

At the time the lab was created there was no Node-RED node specific to Watson Assistant (WA). WA has an HTTP based programming interface. The standard Node-RED http request node can be used to invoke the WA service.

- Drop an **inject** node onto the Voice flow



- Configure the **payload** parameter of the node to send a string of **Hi**
- Drop a **function** node onto the flow. A function node enables snippets of Javasript to be injected into the Node-RED application. This function node will be used to set some properties for use in the HTTP request.
- Configure the function node with the following code:

```
msg.headers={"accept": "application/json"};
msg.payload = { "text": msg.payload, "language": "en-US", "userID": "application-14c", "deviceType": "phone",
"additionalInformation": { "context": {} } };
return msg;
```

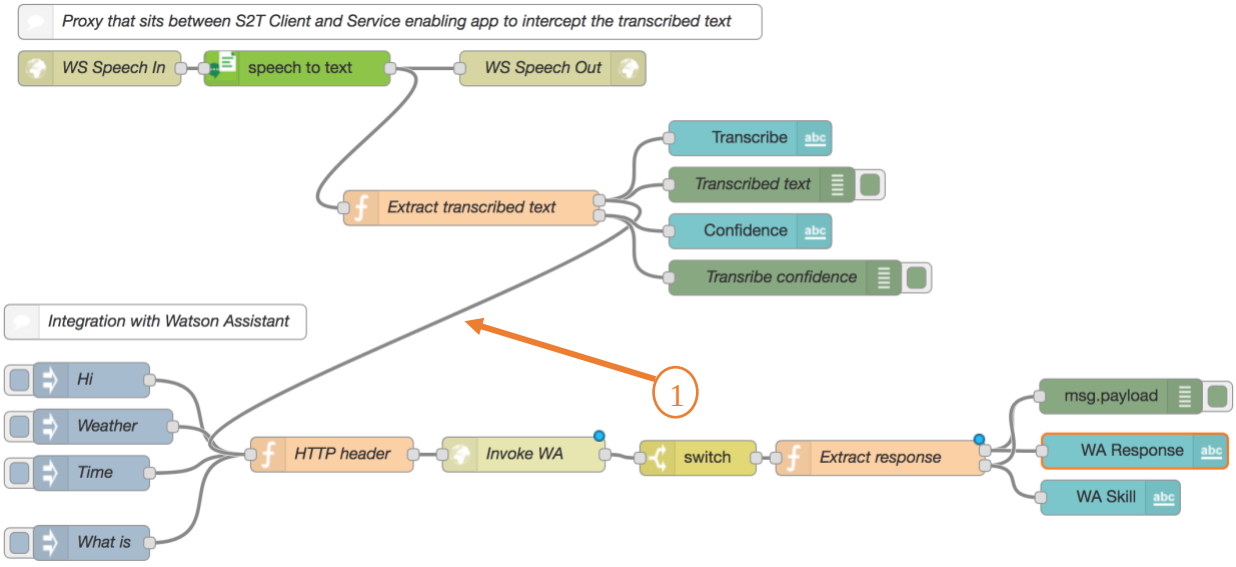
- Wire the output of the inject node to the input of the function node.
- Drop an **http request** node from the function category onto the flow. This is used to invoke WA
- Configure the node :-
  - **Method** to **Post**
  - **URL** to https://watson-personal-assistant-toolkit.mybluemix.net/v2/api/converse?api_key=d8f9e35f-24fe-4641-8bee-fb7cb5f36456
  - **Return** to **a parsed JSON object**
- Wire the output of the function node to the input of the http request node
- Drop a switch node and configure it to select if **msg.payload.reject is false** The node will only send a response through its output if WA has successfully responded to a request.
- Wire the output of the http request node to the input of the switch node
- Drop another **function** node and set it to

```
return [{payload:msg.payload.speech.text},{payload:msg.payload.skill.name}];
```

    - This will return the response from WA and the skill that was used to determine the response
- Wire the output of the switch node to the input of the function node
- Add nodes so that response from WA is displayed in both the Debug Tab and the Dashboard / User Interface.
    - Drop a **debug** node and wire the 2 outputs of the function node to the input of the debug node
    - Drop a **text** dashboard node and configure it to group voice (home). Set the **name** to WA Response
    - Wire the top output of the function node to the input of the WA Response node which will display the text response from WA
    - Drop a **text** dashboard node and configure it to group voice (home). Set the **name** to WA Skill
    - Wire the bottom output of the function node to the input of the WA Skill node which will display the skill used by WA
- **Deploy** the flow. WA can now be tested
- Select the left-hand side of the inject node to send the string Hi through the flow. The debug tab and dashboard will show the result of "How can I help" (or similar)
- Optionally create other inject nodes with other strings to send to WA such as
    - Tell me the time
    - Tell me the weather in las vegas
    - What is a car
    - What is IBM

Step 9.5 Use voice from the BlueCoin to interact with WA

The voice flow has two parts. The first enables the transcribed text to be intercepted and used in the cloud app. The second invokes WA and displays the result of the request. Currently WA is invoked using hard coded text in the inject nodes. The next steps joins the two parts together to enable the voice sent from the BlueCoin to be transcribed and passed to WA.



- Wire output 1 (transcribed text) of the Extract transcribed text function node in the proxy part of the flow to the input of the first function node in the WA part of the flow (1).
- The flow is now complete and can be tested
    - Ensure the Blue MS application is open and configured to send voice data to the proxy
    - Wake up / put the BlueCoin into listening mode (double tap)
    - Talk to the BlueCoin and watch the results. Results will vary based on background noise and other factors such as quality of the network.

Completed Voice Flow

- For info, a completed version of the slack flow can be found at https://github.com/lockedj/BeamMeUpWatson/tree/master/Flows as Voice.flow
- This can be imported into Node-RED by copying the contents of the file to the clipboard. The clipboard can then be imported into a Node-RED flow using the Import Clipboard menu
- The api token for WA contained in the imported flow may not be valid. The token is valid during the Think 2018 conference, once the conference finishes a token can be requested from the WA page https://developer.ibm.com/iot/watson-assistant/

Summary

In completing the lab, you have built an end to end application using the core building blocks of all IoT solutions

- Sensors.
    - The BlueCoin with its sensors enabled both structured data such as motion and environmental data and unstructured data in the form of voice to be collected
- IoT Gateway
    - Your smart device (phone or tablet) enabled the sensor data to be processed locally and for pertinent data to be sent to the cloud.
- Cloud

    - The IBM Cloud provided an environment to host the cloud half of the application and a set of managed services to enable power applications to be quickly composed.
- Managed Services
    - The IBM Cloud provided a set of managed services for the solution / application developer to integrate into their application. Services used in the lab include Cloudant, Watson IoT Platform, Speech to Text and Watson Assistant
- Application Development
    - IBM Cloud supports a wide variety of programming languages to develop applications. Rather than writing lines of code Node-RED was used to visually compose the application.

The knowledge garnered in this lab provides a solid grounding in the principles of IoT, enabling a rich and more diverse set projects to be undertaken. The set of links below relate to the topics in the lab and provide ideas for IoT projects.

To move forward, if you have not already done so register for an IBM Cloud account here :- https://ibm.biz/BdZqYs and then have fun trying some of the links below

Useful links and Ideas for more fun

- IBM
    - IBM Cloud https://bluemix.net
    - Watson Assistant https://developer.ibm.com/iot/watson-assistant/
    - IBM IoT https://www.ibm.com/internet-of-things
    - WIoTP Getting Started Guides https://console.bluemix.net/docs/services/IoT/index.html#gettingstartedtemplate
    - Recipes/How to guides https://developer.ibm.com/recipes/
    - IBM Code, IoT patterns and projects https://developer.ibm.com/code/technologies/iot/
    - Node-RED https://nodered.org/
    - Node-RED additional nodes and example flows https://flows.nodered.org/
- ST
    - BlueCoin http://www.st.com/bluecoin
    - BlueCoin Getting Started Guide https://www.st.com/resource/en/user_manual/dm00405843.pdf
    - Home page http://www.st.com/content/st_com/en.html
    - Discovery Kits and Boards http://www.st.com/en/evaluation-tools/stm32-mcu-eval-tools.html?querycriteria=productId=SS1532

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.