

Mobile Device Movement/Accelerometer Sensor

Hands-On Lab

Author: JeanCarl Bisson | jbisson@us.ibm.com | @dothewww



AT&T 4G 2:07 PM 27% phone-sensor-app.mybluemix.net

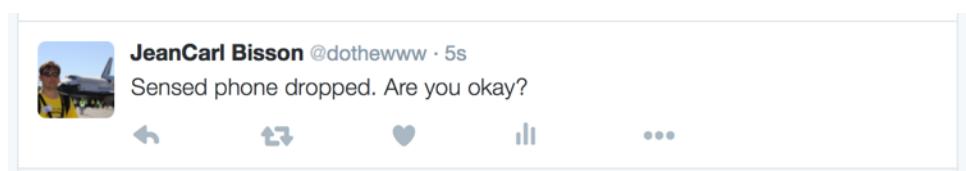
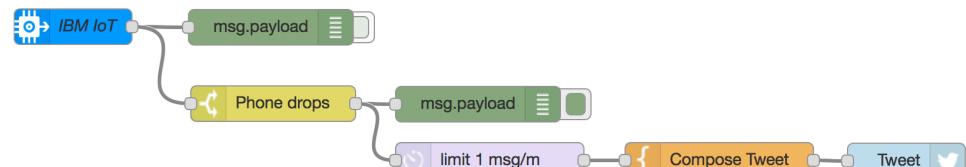
Your device ID: abcd1234
Connected

Published Messages: 43

Movement		
Alpha	Beta	Gamma
-312.91	22.41	-0.17

Acceleration		
X	Y	Z
-0.02	0.11	0.09

Location	
Latitude	Longitude
0	0



A digital copy of this lab/code can be found at:
<http://ibm.biz/lab-mobile-accelerometer-sensor>



Topics covered: Node-RED, Watson Internet of Things Platform, and Twitter.

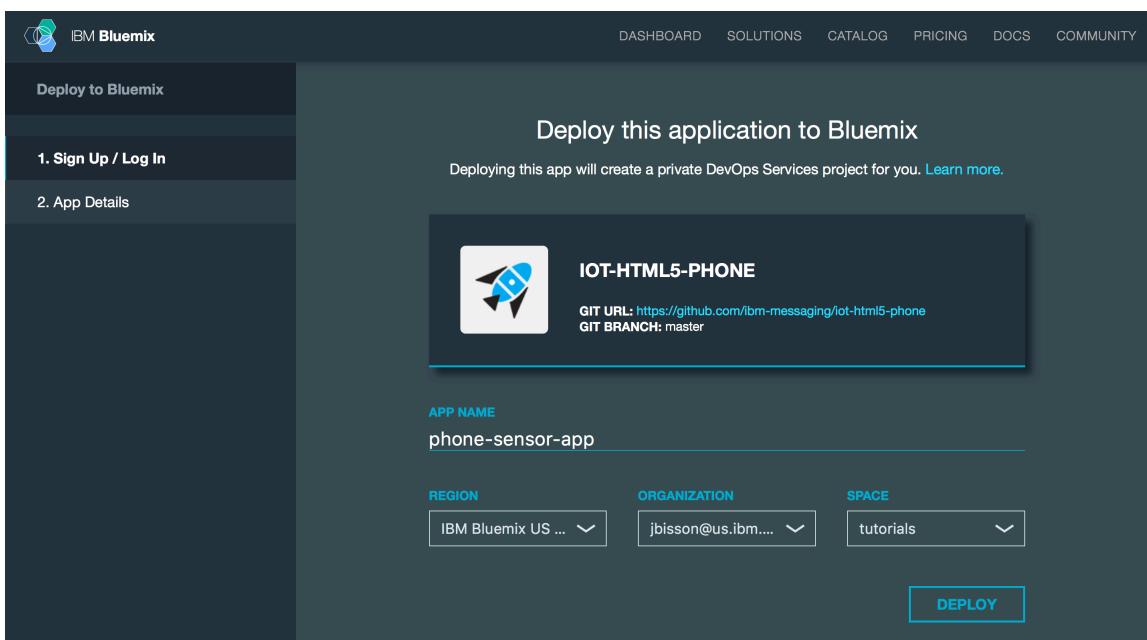
In this lab, we will deploy a web application that measures a mobile device's position, acceleration, and location. The application will send this data to the IBM Watson's Internet of Things Platform service in IBM Bluemix. In the second part of this lab, we will create a simple Node-RED application in Bluemix that listens to these device events and respond by tweeting a message when a sudden acceleration occurs.

Create a web-based mobile device sensor.....	3
Create an accelerometer sensor/alert system	8

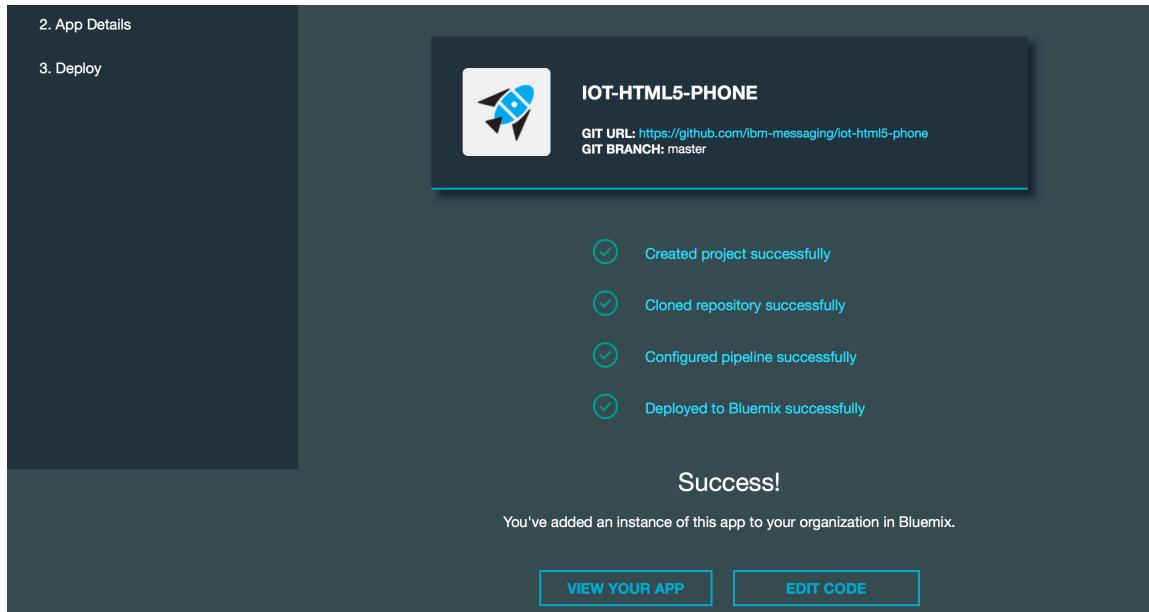
Create a web-based mobile device sensor

In this section, we will deploy a sample Internet of Things application that reads a mobile device's position, accelerometer and location and sends this data to the IBM Watson IoT Platform service in IBM Bluemix.

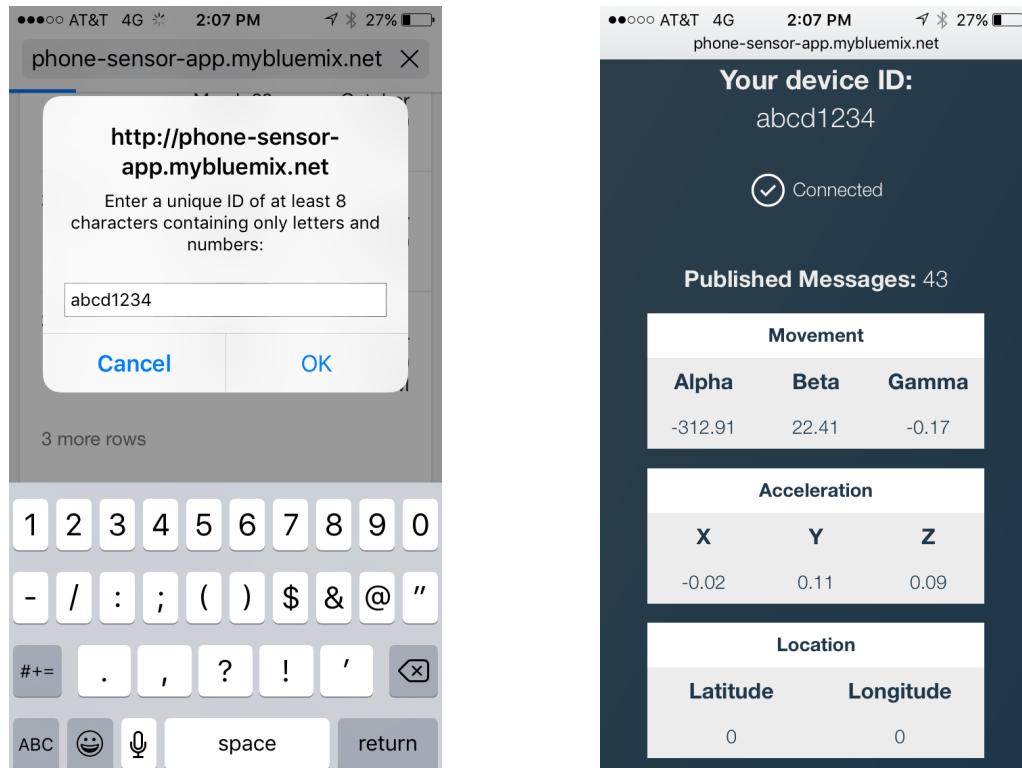
1. Go to GitHub repo <http://ibm.biz/phone-sensor-app>
2. In the Read Me section, click on the  button.
3. Enter a custom app name or use the default given. You can also customize the region, organization, and space the application should be deployed to. Click on **Deploy**. Make note of the app name, it will be part of the URL for the application you need to access in the next step.



4. IBM Bluemix will create a new project, clone the repo, and deploy the application to the region and space you specified. When the application has been deployed successfully, you should see two buttons appear at the bottom of the page. Open your application in a browser on your mobile device and visit <http://<app-name>.mybluemix.net> (change <app-name> to whatever app name you chose).



5. Enter a unique ID of at least 8 characters and tap **OK**. You should see the status change to Connected, and start publishing messages. These messages are sent to the Watson IoT Platform service in Bluemix and contain the movement of the phone and any acceleration the phone may experience.



6. Using a computer, open the application overview in your Bluemix dashboard. Click on the Internet of Things Platform link in the left sidebar. Click on the green **Launch** button.

The screenshot shows the IBM Bluemix Internet of Things dashboard. At the top, there's a navigation bar with icons for three, the IBM logo, and 'IBM Bluemix Internet of Things'. On the right are links for 'Catalog', 'Support', and 'Account'. Below the navigation is a breadcrumb trail: 'Phone-sensor-app < Internet of Things Platform'. Underneath are tabs for 'Manage' (which is selected), 'Plan', and 'Connections'. A large central icon depicts a central node connected to various sensors and actuators. Below the icon, the text 'Welcome to Watson IoT Platform' is displayed. A subtitle reads: 'Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.' At the bottom of the main content area are two buttons: a teal 'Launch' button and a grey 'Docs' button. To the left of the main content, there's a 'Learn about Watson IoT Platform' section with a link to developerWorks recipes. To the right, there's a 'Connect and expand by using developerWorks recipes' section with a link to developerWorks.

7. This is the IBM Watson IoT Platform dashboard. You can see the connected devices that have been registered through the application you just deployed and ran. Hover over the icons in the left sidebar and click on Devices.

The screenshot shows the IBM Watson IoT Platform dashboard. At the top, there's a header with the title 'IBM Watson IoT Platform' and links for 'QUICKSTART', 'SERVICE STATUS', 'DOCUMENTATION', and 'BLOG'. On the far right, it shows the user's email 'jbisson@us.ibm.com' and ID 'ID: (r17jhg)'. Below the header, the main area is titled 'All Boards' with a blue 'Create New Board' button. To the left is a vertical sidebar with icons for Home, Settings, Analytics, and Device Management. The main content area is titled 'Your boards' and shows four cards: 'USAGE OVERVIEW' (3 Cards, Owned by you), 'RISK AND SECURITY OVERVIEW' (4 Cards, Owned by you), 'RULE-CENTRIC ANALYTICS' (6 Cards, Owned by you), and 'DEVICE-CENTRIC ANALYTICS' (5 Cards, Owned by you). Each card has a trash, heart, and gear icon at the bottom.

The screenshot shows the IBM Watson IoT Platform dashboard. On the left, a sidebar menu lists various sections: BOARDS, DEVICES, MEMBERS, APPS, USAGE, RULES, SECURITY, SETTINGS, and EXTENSIONS. The main area displays three cards under the heading 'RISK AND SECURITY OVERVIEW': '4 Cards' (Owned by you), '6 Cards' (Owned by you), and 'DEVICE-CENTRIC ANALYTICS' (5 Cards). A blue button '+ Create New Board' is located in the top right corner.

8. The next screen displays all the devices that have been registered to your instance of the IBM Watson IoT Platform service. You should see the Device ID that you entered on your mobile device. Click on it.

The screenshot shows the 'Devices' page in the IBM Watson IoT Platform. The left sidebar has icons for Boards, Devices, Members, Apps, Usage, Rules, Security, Settings, and Extensions. The main content area is titled 'Devices' and includes buttons for 'Browse', 'Diagnose', 'Action', 'Device Types', 'Manage Schemas', 'Refresh', and '+ Add Device'. A table lists one device: 'abcd1234' (Device Type: 'iotphone', Class ID: 'Device', Date Added: 'Feb 3, 2017 7:29:13 PM'). The table has columns for Device ID, Device Type, Class ID, Date Added, and Location. There are also icons for refresh, search, and delete.

9. A dialog displays the recent events that have been sent from that device. Scroll down to the **Sensor Information** section.

The screenshot shows the IBM Watson IoT Platform interface. On the left, a sidebar lists various sections: Devices, Connection Information, Recent Events (selected), Device ID, Sensor Information, Metadata, Device Information, Diagnostic Logs, Error Codes, Connection Log, and Actions. The main content area is titled "Device abcd1234". It contains two tabs: "Connection Information" and "Recent Events". The "Connection Information" tab shows details like Device ID (abcd1234), Device Type (iotphone), Date Added (Friday, February 3, 2017), Added By (jbisson@us.ibm.com), and Connection State (Connected on Friday, February 3, 2017 at 7:29:57 PM from 50.143.173.191 with an insecure connection). The "Recent Events" tab lists ten sensorData events received on Feb 3, 2017, at 7:30:55 PM, each in json format.

Event	Format	Time Received
sensorData	json	Feb 3, 2017 7:30:55 PM
sensorData	json	Feb 3, 2017 7:30:55 PM
sensorData	json	Feb 3, 2017 7:30:56 PM
sensorData	json	Feb 3, 2017 7:30:56 PM
sensorData	json	Feb 3, 2017 7:30:56 PM
sensorData	json	Feb 3, 2017 7:30:56 PM
sensorData	json	Feb 3, 2017 7:30:56 PM
sensorData	json	Feb 3, 2017 7:30:56 PM
sensorData	json	Feb 3, 2017 7:30:56 PM
sensorData	json	Feb 3, 2017 7:30:56 PM

10. The Sensor Information section displays the last data that is submitted to the IoT Platform by the device. As you move the device, the values change.

The screenshot shows the IBM Watson IoT Platform interface. The sidebar and main content area are identical to the previous screenshot, but the "Sensor Information" section is now selected. This section displays a table of sensor data points with their corresponding datatypes, values, and times received. The values change as the device moves.

Event	Datapoint	Value	Time Received
sensorData	d.id	abcd1234	Feb 3, 2017 7:31:06 PM
sensorData	d.ts	1486179067465	Feb 3, 2017 7:31:06 PM
sensorData	d.lat	0	Feb 3, 2017 7:31:06 PM
sensorData	d.lng	0	Feb 3, 2017 7:31:06 PM
sensorData	d.ax	-0.02	Feb 3, 2017 7:31:06 PM
sensorData	d.ay	0	Feb 3, 2017 7:31:06 PM
sensorData	d.az	0.02	Feb 3, 2017 7:31:06 PM
sensorData	d.oa	21.25	Feb 3, 2017 7:31:06 PM
sensorData	d.ob	-4.52	Feb 3, 2017 7:31:06 PM
sensorData	d.og	-18.16	Feb 3, 2017 7:31:06 PM

Create an accelerometer sensor/alert system

In this section, we will create a simple application that listens to the data the device sends to the Watson IoT platform and respond to a sudden acceleration. For example, this could simulate a sudden fall.

1. In the Bluemix dashboard, click on the **Catalog** link. Create a new Node-RED application by clicking on the **Node-RED Starter** tile under the **Boilerplates** section.

The screenshot shows the IBM Bluemix Catalog interface. On the left, there's a sidebar with categories like All Categories, Infrastructure, Apps (with Boilerplates selected), and Services. Under Apps, Boilerplates is expanded, showing options like Cloud Foundry Apps, Containers, OpenWhisk, Mobile, and a sub-section for Boilerplates. The sub-section for Boilerplates contains several tiles, one of which is the "Node-RED Starter" tile. This tile has a blue background, a white icon, and the text "Node-RED Starter" and "This application demonstrates how to run the Node-RED open-source project within IBM Bluemix." Below the catalog, the URL "https://console.ng.bluemix.net/catalog/starters/node-red-starter/" is visible.

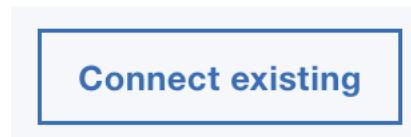
2. Enter a name for the application. Choose a host name that is unique. Make note of this host name, it will be used in step #7. Click on **Create**.

The screenshot shows the "Create a Cloud Foundry App" page for the Node-RED Starter. At the top, there's a "View all" link and a search bar. Below that, the title "Create a Cloud Foundry App" is displayed. The main form has fields for "App name:" (set to "phone-app-application"), "Host name:" (set to "phone-app-application"), and "Domain:" (set to "mybluemix.net"). To the right of these fields, there are dropdown menus for "Selected Plan:" (set to "SDK for Node.js™ Default") and "Cloudant NoSQL DB" (set to "Lite"). Below the form, there are two circular icons: one for "SDK for Node.js™" and another for "Cloudant NoSQL DB". At the bottom of the page, there are links for "Need Help?", "Contact Bluemix Sales", "Estimate Monthly Cost", and "Cost Calculator", along with a large blue "Create" button.

3. Wait for the application to finish staging. Click on the **Overview** link in the left sidebar.

The screenshot shows the IBM Bluemix Cloud Foundry Apps interface. At the top, there's a navigation bar with 'Catalog', 'Support', and 'Account' links. Below the navigation is a sidebar with 'Dashboard', 'Getting started', 'Overview', 'Runtime', 'Connections', 'Logs', and 'Monitoring' options. The main content area displays an application named 'phone-app-application' with a status of 'Starting'. A 'View app' button is in the top right. The 'Getting started' section contains two numbered steps: 1) After your application has started, click on the Routes URL or enter the following URL in a browser: <http://<yourhost>.mybluemix.net>. Step 2) Click Go to your Node-RED flow editor. Below these steps is a section titled 'Customizing your Node-RED instance' with a 'Download CF Command Line Interface' button.

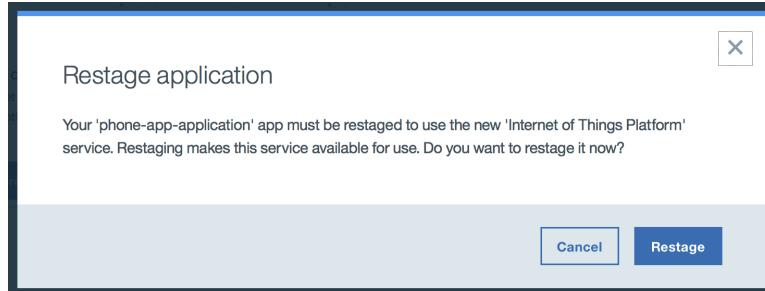
- We need to connect our application to the Watson IoT Platform service that was created in the other application. To do this, we “connect [an] existing” service to our new application. Both applications will then share access to this service concurrently. The first application sends the data to the service; the second application subscribes to the data from the service. Click on **Connect existing** in the top-right corner.



- Select the IBM Watson IoT Platform service and click **Connect**.

The screenshot shows a 'Connect existing service' dialog box. At the top left is the text 'Connect existing service'. On the right is a blue hexagonal close button. Below this is a 'Services' tab. Under the 'Services' tab, there is a list with one item: 'Internet of Things Platform-3m IBM' with its icon. At the bottom of the dialog is a note 'Select compatible service to connect to jcsampleapp' and a blue 'Connect' button.

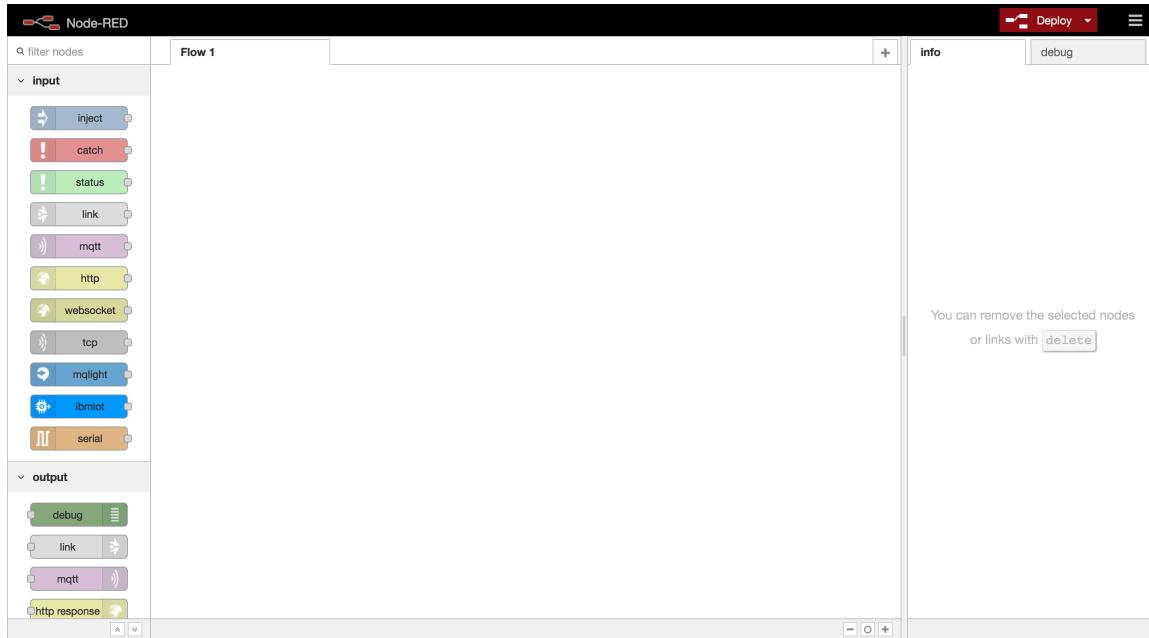
- When prompted that the application must be restaged, click on **Restage**.



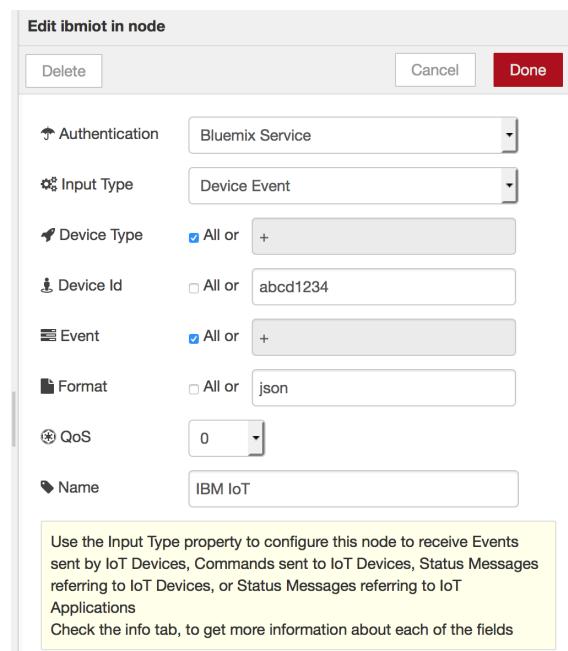
- When the application has restarted, open a browser tab and visit your application's URL (`http://<host-name-from-step-#2>.mybluemix.net`). This is Node-RED, a graphical programming interface where you can drag and drop nodes (predefined functions) and connect them together to construct a flow (a program of functionality). Click on the red button labeled **Go to your Node-RED flow editor**.

A screenshot of the Node-RED in BlueMix interface. The top navigation bar says 'Node-RED in BlueMix'. The main title is 'Node-RED in BlueMix' with the subtitle 'A visual tool for wiring the Internet of Things'. Below the title is a diagram showing a flow of nodes connected by wires. A red button at the bottom right says 'Go to your Node-RED flow editor'. Below the button are three links: 'Learn how to password-protect your instance', 'Learn how to customise Node-RED', and 'More information about Node-RED, including documentation, can be found at nodered.org'.

8. There are four things to note about the Node-RED interface. The palette in the left-sidebar contains nodes that perform pre-defined functionality. Click on each node in the palette and read the details in the info tab in the right sidebar. Drag nodes into the middle canvas and connect the grey nobs together to compose what is called a flow, executed from left to right. In the right sidebar, the info tab explains what the nodes do and their input and outputs; and the debug tab displays data from the debug node. Finally, the red **Deploy** button in the top-right corner deploys any unsaved nodes and flows.



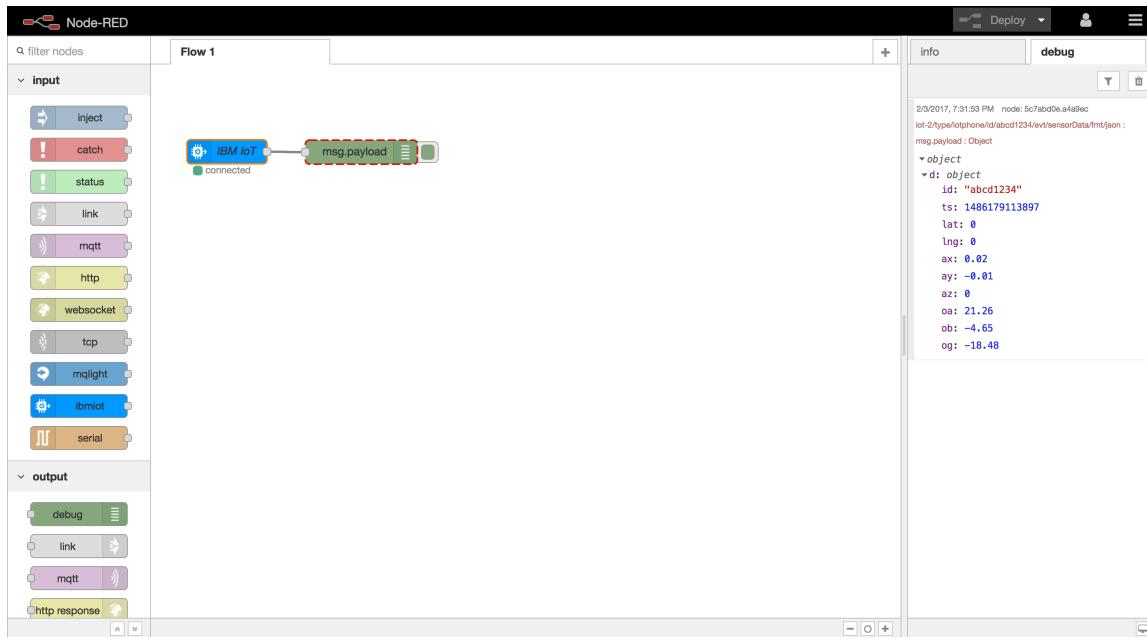
9. Let's create a simple flow that listens to the events from our mobile device and displays the contents. Drag a node and a node to the canvas. Double click on the node and customize the settings as shown below. Use the same Device ID as you had on the mobile device.



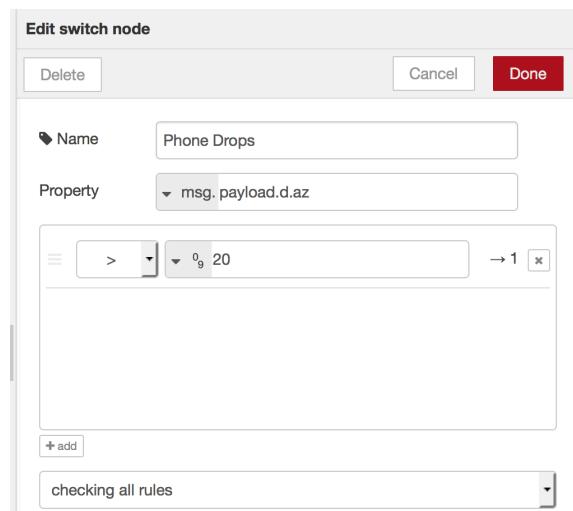
10. The completed flow should look like the following. Click on the  Deploy button in the top-right corner to deploy and save your changes.



11. Click on the debug tab in the right sidebar. The sensor data from the phone should be displayed.

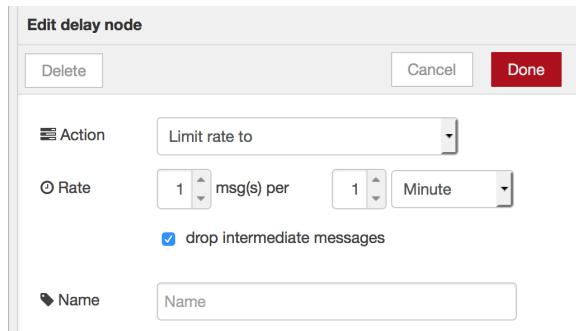


12. Click on the green tab to the right of the debug node in the canvas. This toggles off/on the messages shown in the debug tab. Next, add a  switch statement as shown below.



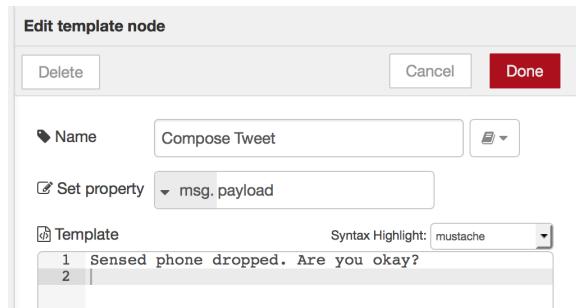
This will filter out any messages where the az value doesn't exceed the value 20. In other words, we only want to proceed with the flow when the value of az exceeds the value 20 (arbitrary value, you can change it if the device is less sensitive).

13. Add a  node as shown below.



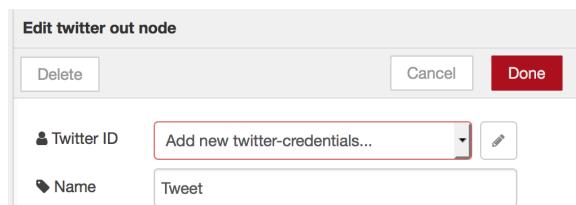
This node limits how often the rest of flow is run. Since the acceleration could exceed this value more than one, without this node, a tweet could be sent several times. With this node, we can limit this flow to send a tweet once per minute.

14. Add a  node as shown below.



This composes a message that we will tweet.

15. Add a  node. Click on the pencil button and authenticate with Twitter. The account you sign in with will be used to send tweets.





Authorize Node RED to use your account?

[Authorize app](#)

[Cancel](#)



Node RED

nodered.org

Node-RED Twitter node

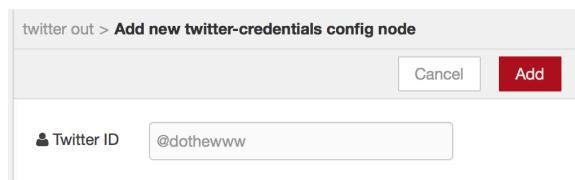
This application will be able to:

- Read Tweets from your timeline.
- See who you follow, and follow new people.
- Update your profile.
- Post Tweets for you.
- Access your direct messages.

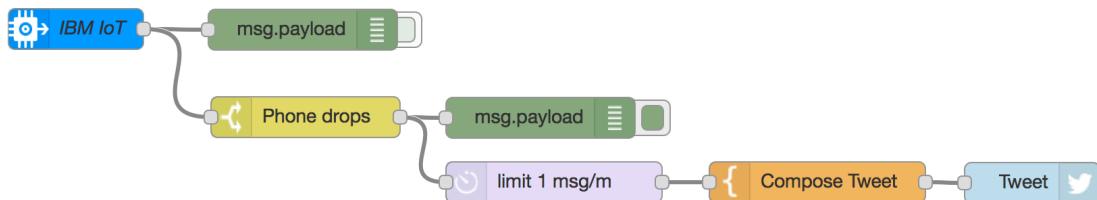
Will not be able to:

- See your Twitter password.

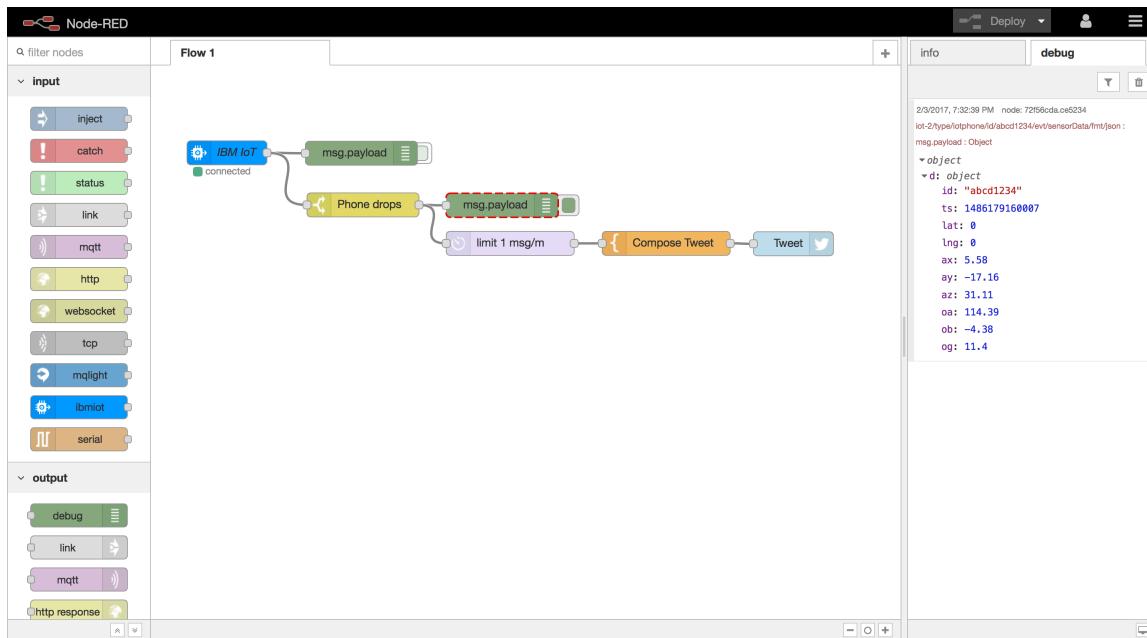
Return back to the node configuration. The settings for the Twitter node should have your username set. Click **Add**.



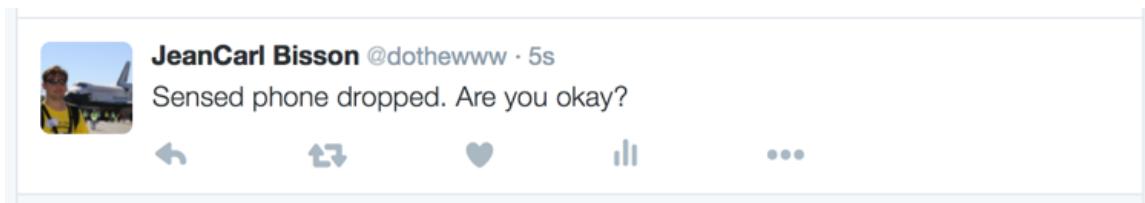
16. Connect the nodes together as shown below.



17. Click  Deploy to deploy and save the changes. Give the phone a quick jolt. The debug tab should display the sensor data.



18. Check the Twitter timeline for the tweet.



This lab demonstrates a simple Node-RED application that tweets a message when the mobile device receives a sudden jolt. However, this proof of concept can be extended.

You can change the message to send a Direct Message to a specific Twitter user, say a friend or family member (Hint: read the documentation of the  node to find out how to send a Direct Message).

You can also listen for a response from the user (Hint: a  node might help) and if the person has fallen and doesn't respond in a timely fashion, alert someone nearby to check on the owner of the mobile device.

What else you could measure based on the orientation, the movement, or the acceleration of the mobile device?