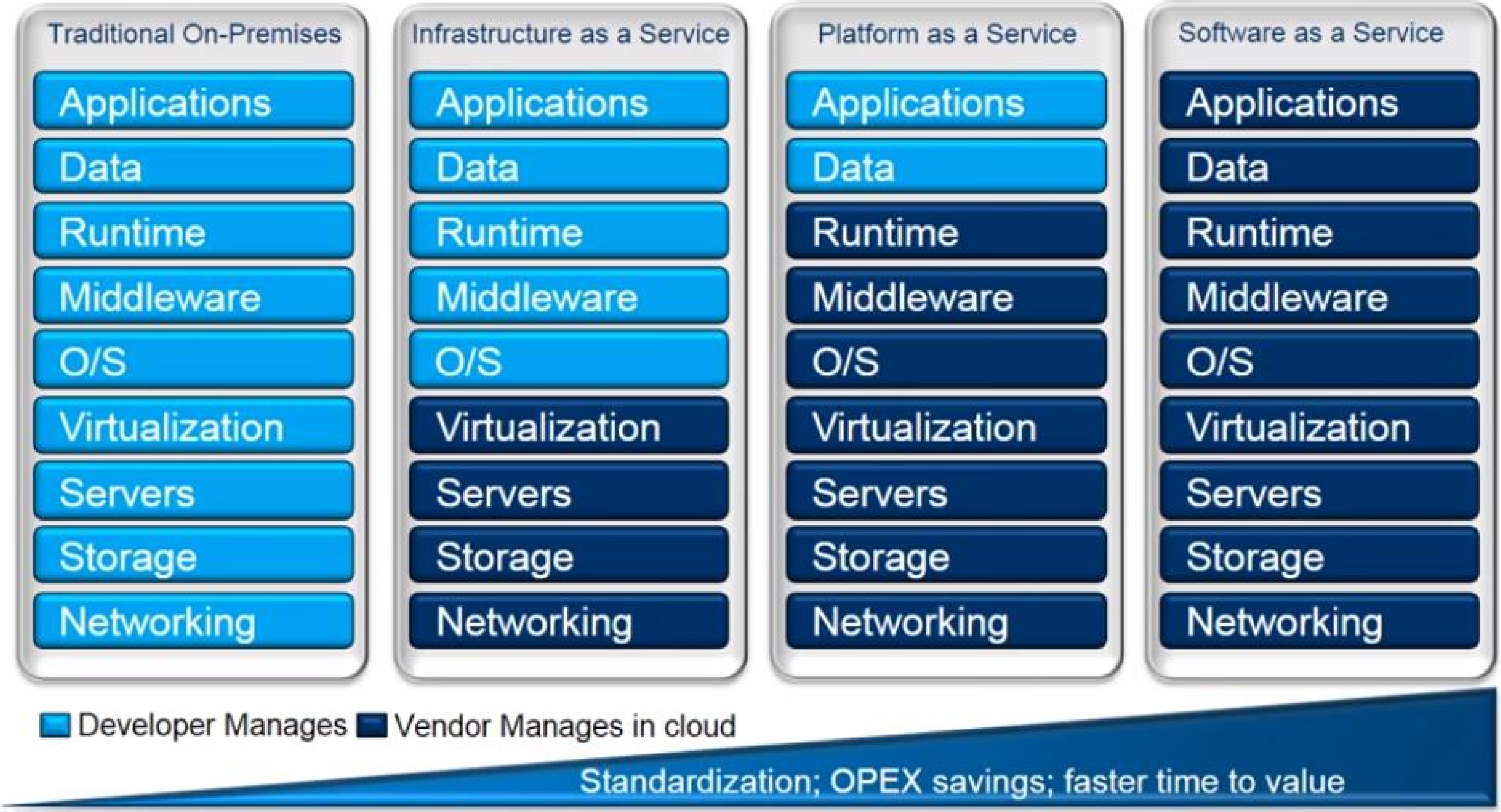
BUILD YOUR FIRST IOT APP

JIGSAW ACADEMY™
THE ONLINE SCHOOL OF ANALYTICS

Cloud Computing: IaaS, PaaS, SaaS



Traditional On-Premises	Infrastructure as a Service	Platform as a Service	Software as a Service
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

🟦 Developer Manages ■ Vendor Manages in cloud

Standardization; OPEX savings; faster time to value

- *Data ingestion*: Ingress of data at scale to the cloud.
- *Device identity*: Manage unique device identities and control device access to the solution.
- *Device management*: Manage device metadata and perform operations such as device reboots and firmware upgrades.
- *Command and control*: To cause the device to take an action, send messages to a device from the cloud.
- *Rules and actions*: To act on specific device-to-cloud data, the solution back end uses rules.
- *Predictive analytics*: The solution back end analyses device-to-cloud data to predict when specific actions should take place. For example, analysing aircraft engine telemetry to determine when engine maintenance is due.

Turn your Smartphone into an IOT Device

You'll use an IOT Starter Mobile app on your smartphone.
(*which provides the list of simple commands that your smartphone can interpret****)***

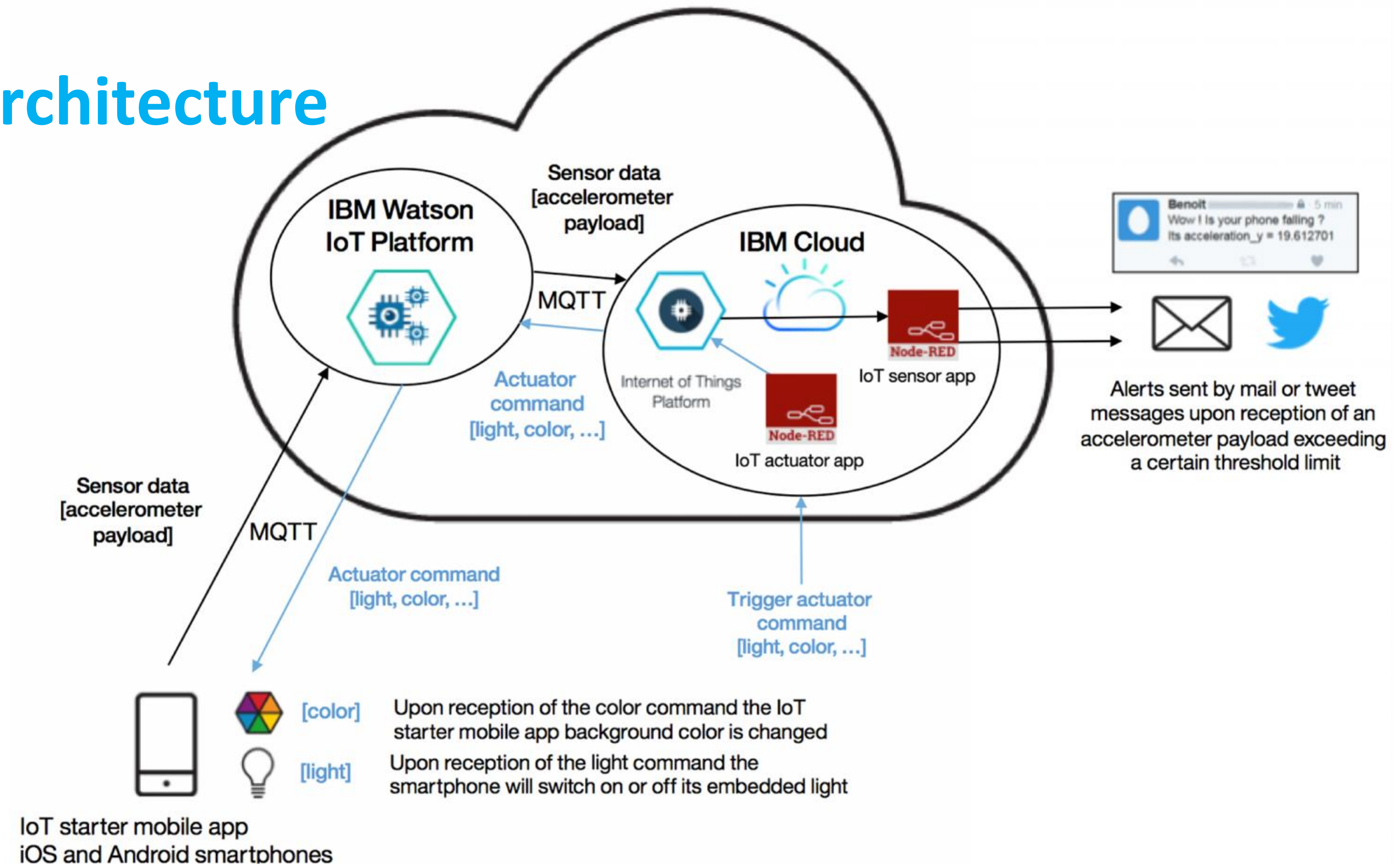
- To transform your smartphone into a **sensor**, we will build an IOT app that
  - Reads the accelerometer values that are sent from your smartphone and
  - Reacts with a specific action whether the values reach a certain threshold
  - E.g: When your smartphone is falling down or moving too fast and its acceleration exceeds a certain threshold limit, you will be alerted

- To transform your smartphone into an **actuator**, we will build an IOT app that
  - Sends specific commands to your smartphone
  - Can react to the commands it receives.
  - E.g.: For instance, based on the position of your smartphone, we will turn on or off the light or change the background color of your smartphone.

What you'll need to build these IoT apps

- An IBM Cloud account.
- A smartphone (an Android or iOS smartphone).
  - For an iOS smartphone, a valid iOS Developer License and Apple Xcode.
- A twitter account to be notified by a tweet (optional).
- An email account with SMTP relay capabilities to be notified by an email message (optional).

Architecture



IBM Watson IoT Platform

Sensor data [accelerometer payload]

MQTT

IBM Cloud

Internet of Things Platform

Node-RED

IoT sensor app

Node-RED

IoT actuator app

Actuator command [light, color, …]

Benoit · 5 min
Wow ! Is your phone falling ?
Its acceleration_y = 19.612701

Alerts sent by mail or tweet messages upon reception of an accelerometer payload exceeding a certain threshold limit

Sensor data [accelerometer payload]

MQTT

Actuator command [light, color, …]

Trigger actuator command [light, color, …]

[color] Upon reception of the color command the IoT starter mobile app background color is changed

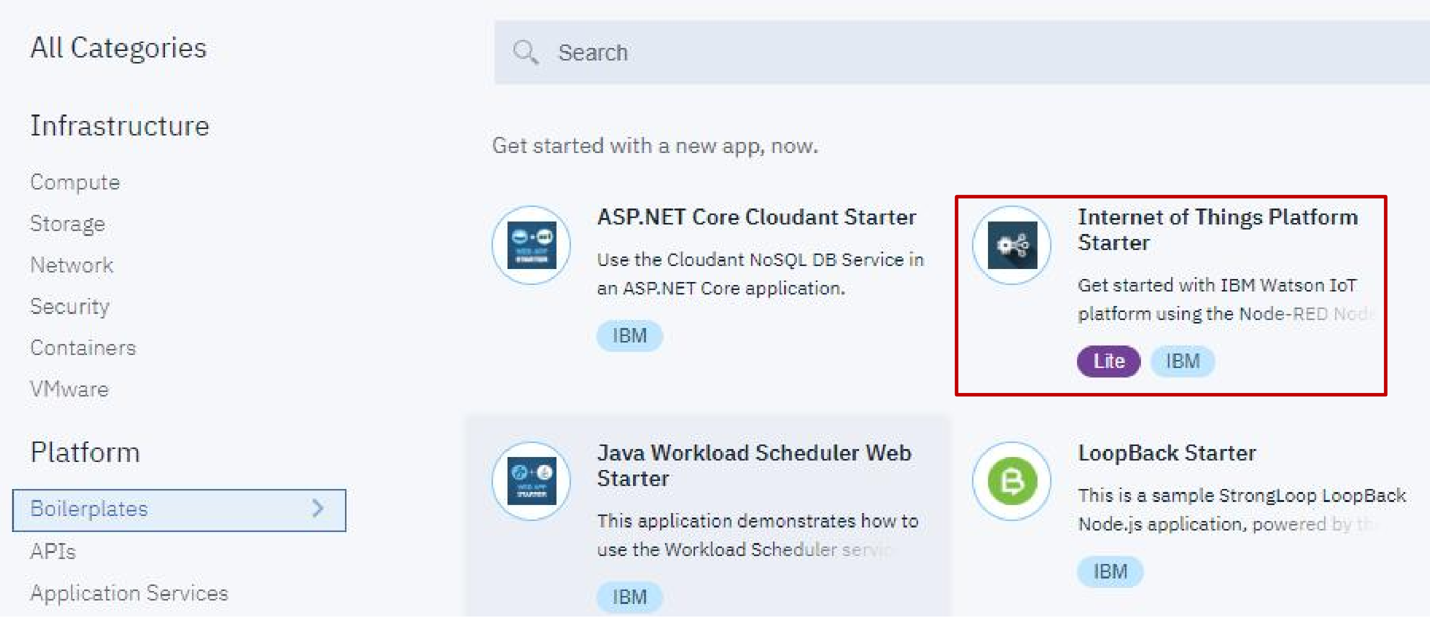
[light] Upon reception of the light command the smartphone will switch on or off its embedded light

IoT starter mobile app
iOS and Android smartphones

STEP 1: Create your main IOT app in the IBM Cloud

Our main IoT app in IBM Cloud will be our backend application in our IOT solution. Our backend application will read, receive, and interpret incoming payload messages that contain sensor data. Additionally, it will define and send IoT commands to actuators based on certain conditions being met.

- Log in to your IBM Cloud account.
- In the IBM Cloud catalog, from the Boilerplates section, select the Internet of Things Platform Starter boilerplate.



All Categories

Search

Infrastructure

Compute
Storage
Network
Security
Containers
VMware

Platform

Boilerplates >
APIs
Application Services

Get started with a new app, now.

ASP.NET Core Cloudant Starter
Use the Cloudant NoSQL DB Service in an ASP.NET Core application.
IBM

Internet of Things Platform Starter
Get started with IBM Watson IoT platform using the Node-RED Node
Lite IBM

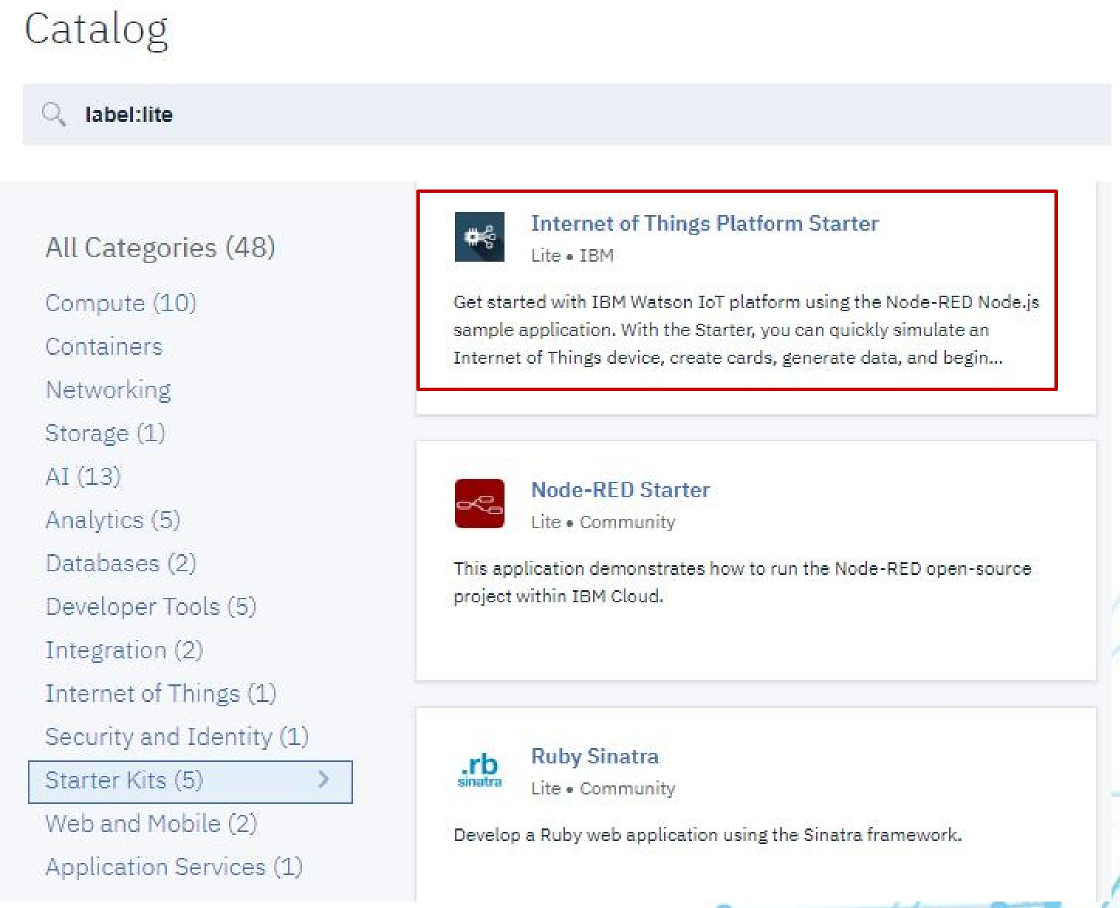
Java Workload Scheduler Web Starter
This application demonstrates how to use the Workload Scheduler servic
IBM

LoopBack Starter
This is a sample StrongLoop LoopBack Node.js application, powered by th
IBM

STEP 1: Create your main IOT app in the IBM Cloud

Our main IoT app in IBM Cloud will be our backend application in our IOT solution. Our backend application will read, receive, and interpret incoming payload messages that contain sensor data. Additionally, it will define and send IoT commands to actuators based on certain conditions being met.

- Log in to your IBM Cloud account.
- In the IBM Cloud catalog, from the Boilerplates section, select the Internet of Things Platform Starter boilerplate.



Internet of Things Platform Starter
Lite • IBM

Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the Starter, you can quickly simulate an Internet of Things device, create cards, generate data, and begin...

Node-RED Starter
Lite • Community

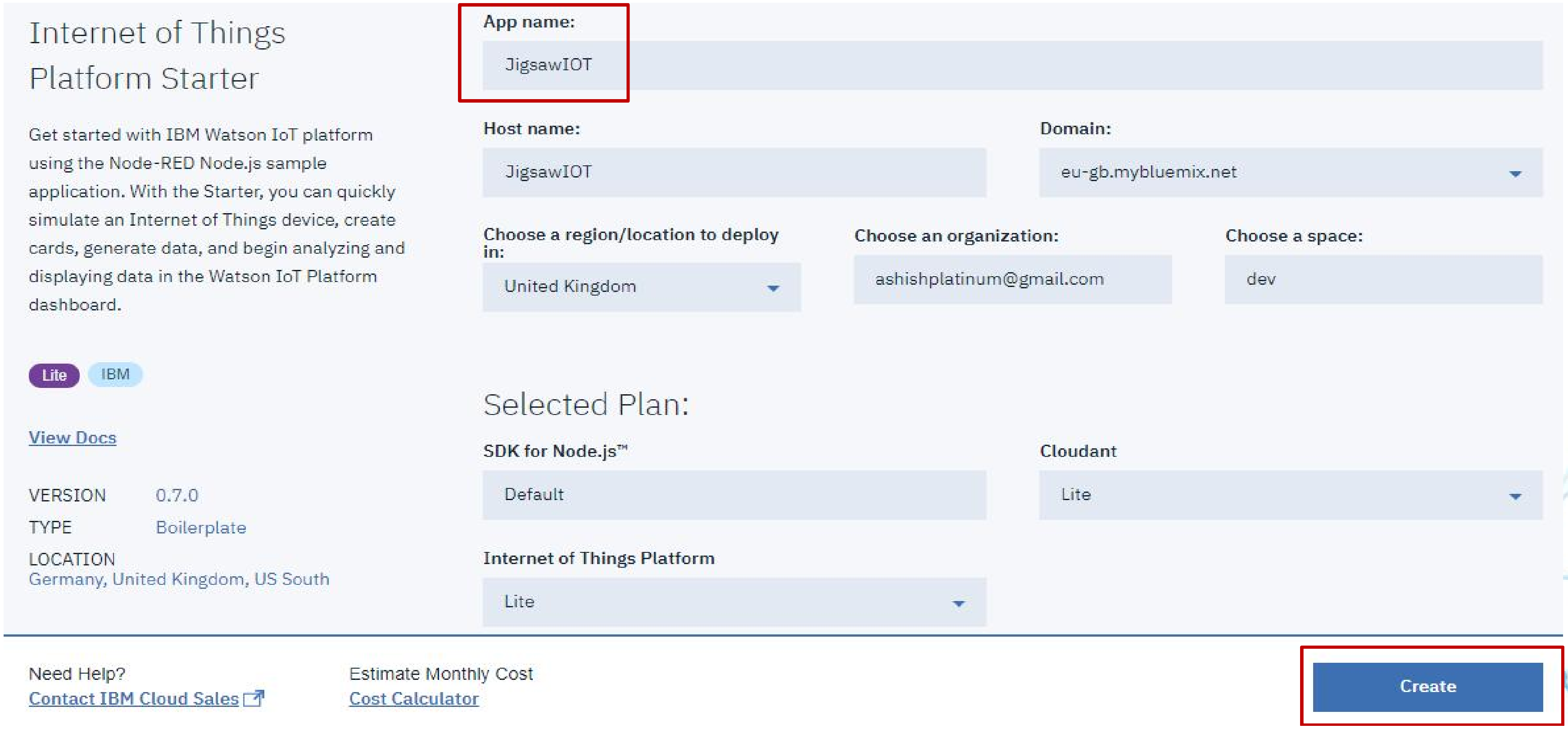
This application demonstrates how to run the Node-RED open-source project within IBM Cloud.

Ruby Sinatra
Lite • Community

Develop a Ruby web application using the Sinatra framework.

STEP 1: Create your main IOT app in the IBM Cloud

- In the right pane, specify a unique name for your app, which is used to create a unique host name for your app. Then, click **Create**.
- The Internet of Things Platform Starter boilerplate automatically adds the Internet of Things Platform service and a Cloudant NoSQL Database service to your app. It also includes a Node-RED runtime environment for your app.



Internet of Things Platform Starter

Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the Starter, you can quickly simulate an Internet of Things device, create cards, generate data, and begin analyzing and displaying data in the Watson IoT Platform dashboard.

VERSION	0.7.0
TYPE	Boilerplate
LOCATION	Germany, United Kingdom, US South

App name: JigsawIOT

Host name: JigsawIOT

Domain: eu-gb.mybluemix.net

Choose a region/location to deploy in: United Kingdom

Choose an organization: ashishplatinum@gmail.com

Choose a space: dev

Selected Plan:

SDK for Node.js™: Default

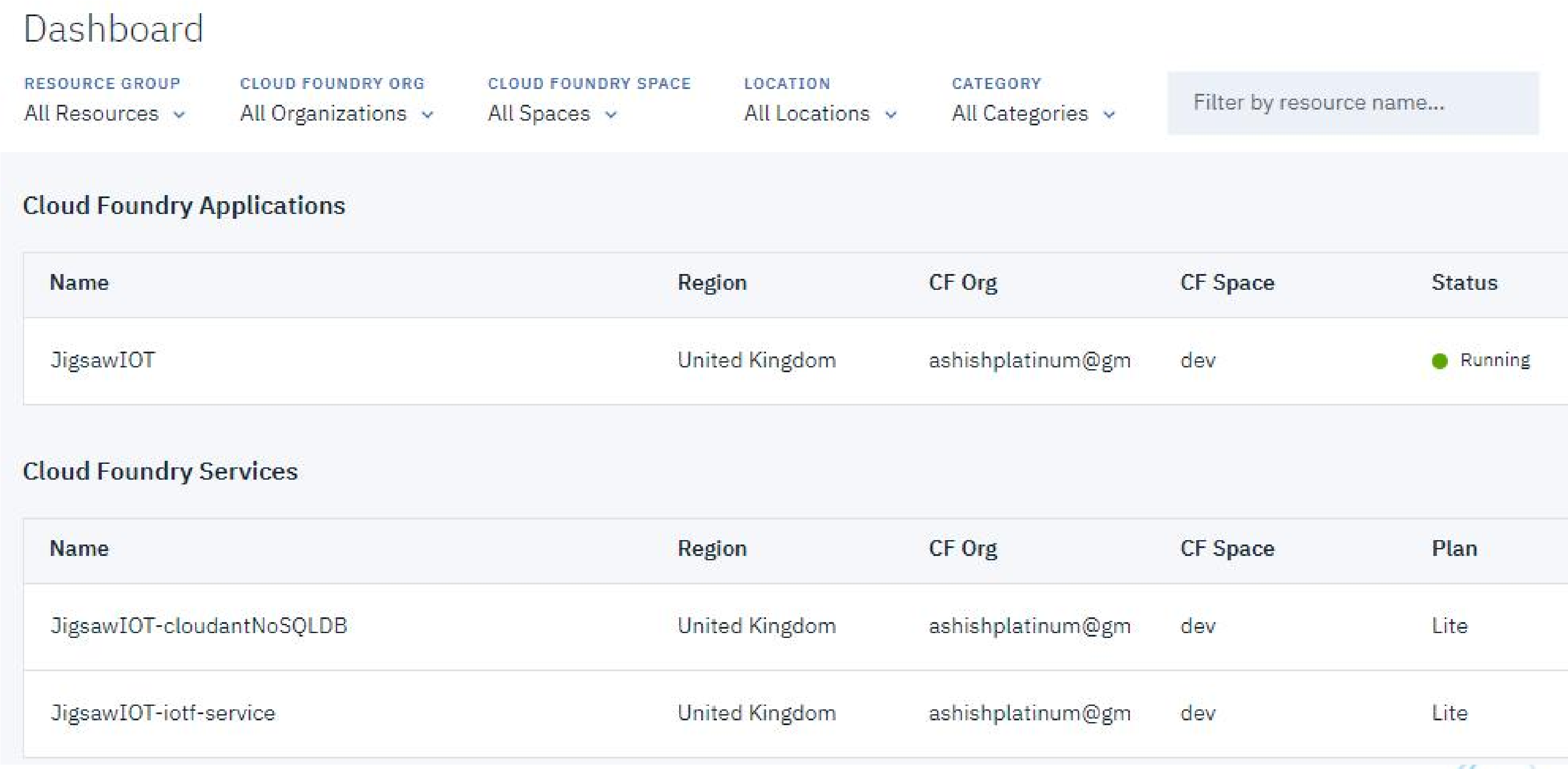
Cloudant: Lite

Internet of Things Platform: Lite

Need Help? Contact IBM Cloud Sales

Estimate Monthly Cost Cost Calculator

STEP 1: Create your main IOT app in the IBM Cloud



Dashboard

RESOURCE GROUP: All Resources

CLOUD FOUNDRY ORG: All Organizations

CLOUD FOUNDRY SPACE: All Spaces

LOCATION: All Locations

CATEGORY: All Categories

Cloud Foundry Applications

Name	Region	CF Org	CF Space	Status
JigsawIOT	United Kingdom	ashishplatinum@gm	dev	● Running

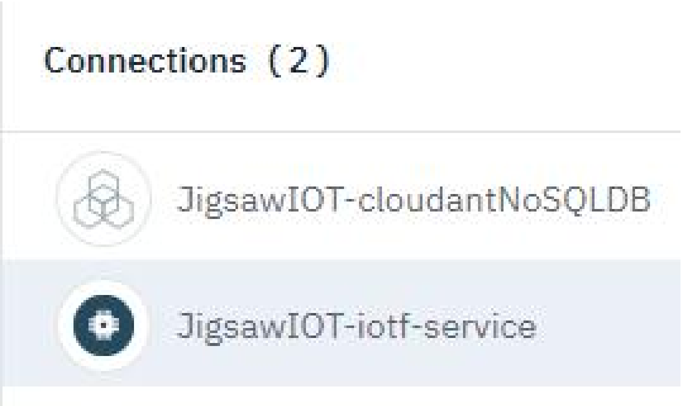
Cloud Foundry Services

Name	Region	CF Org	CF Space	Plan
JigsawIOT-cloudantNoSQLDB	United Kingdom	ashishplatinum@gm	dev	Lite
JigsawIOT-iotf-service	United Kingdom	ashishplatinum@gm	dev	Lite

STEP 2: Register your smartphone

You need to register your smartphone in the Watson IOT Platform before you can connect it to the IoT. The Watson IOT Platform service automatically allocates an IOT organization to you. An IOT organization is a space that is used for connecting and managing devices to the Watson IoT Platform so that your applications can access their live and historical data.

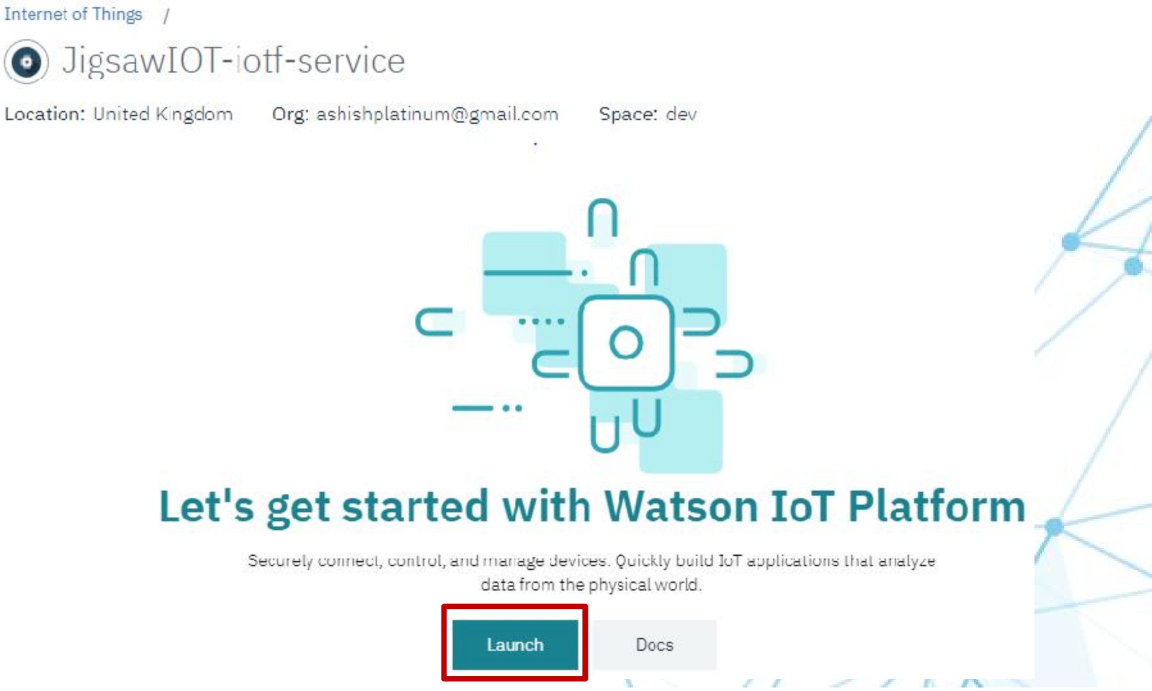
- In the Overview view of your app, under Connections, click the Internet of Things Platform service, named something like iot<your name>-iotf-service. Then click launch.



Connections (2)

JigsawIOT-cloudantNoSQLDB

JigsawIOT-iotf-service



Internet of Things /

JigsawIOT-iotf-service

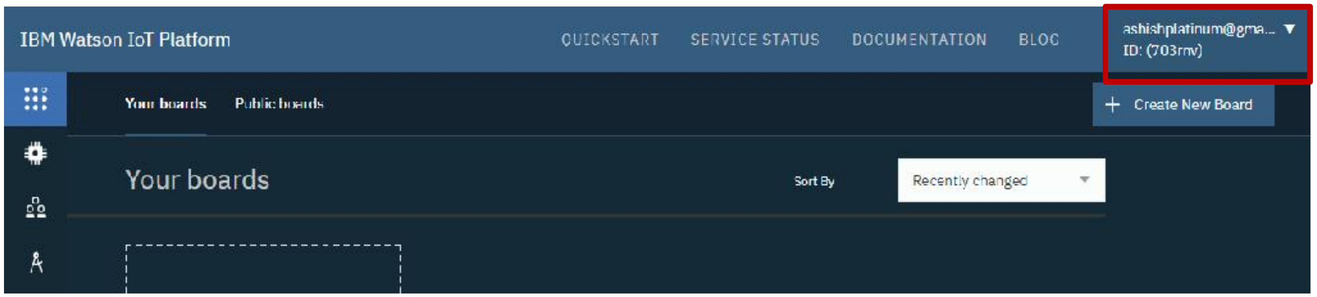
Location: United Kingdom Org: ashishplatinum@gmail.com Space: dev

Let's get started with Watson IoT Platform

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

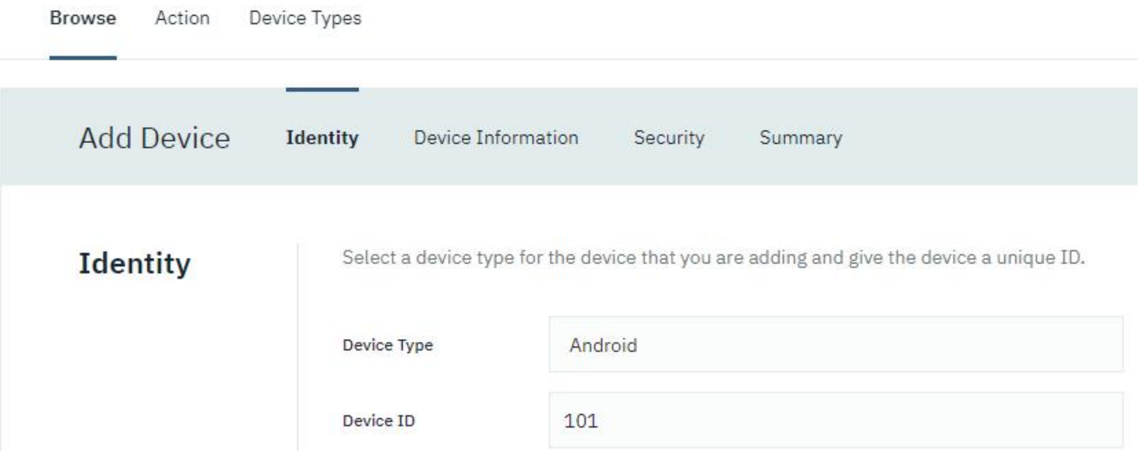
STEP 2: Register your smartphone

- The dashboard for IBM Watson IoT Platform opens in a new browser tab. The organization ID is assigned to your app and is displayed in the upper right. You will need this ID later for configuring the mobile app. Below, the organization ID is (703rnv)



ID: (703rnv)

- On the left menu, which pops out when you hover over it, click Devices. Then, click Add a device type. In your organization, you can have multiple device types each with multiple devices. A device type is a group of devices that share characteristics; In our case, the device type name must be "Android" (this device type name is required by the app that you will use later).



Identity

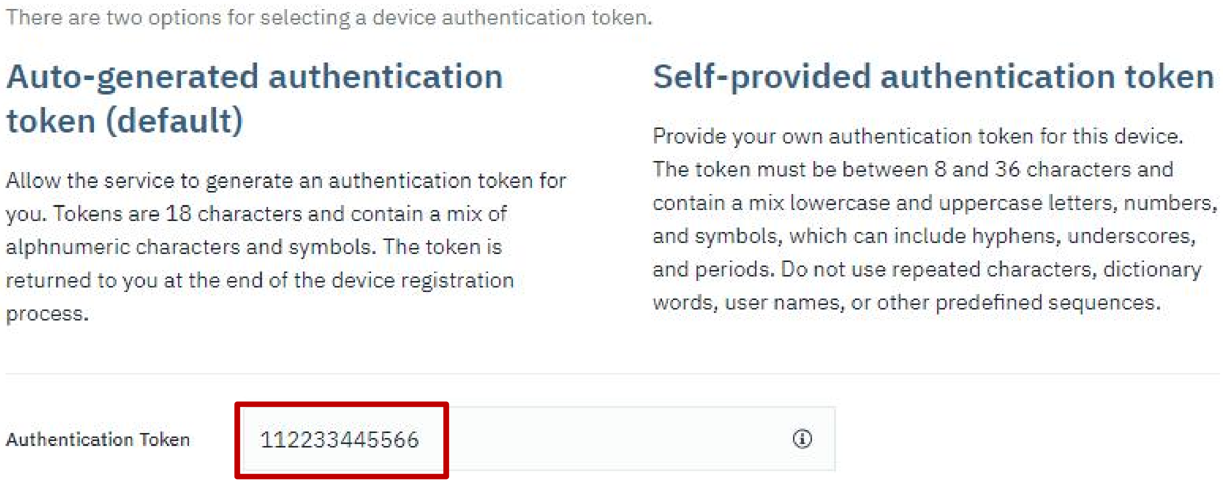
Select a device type for the device that you are adding and give the device a unique ID.

Device Type: Android

Device ID: 101

STEP 2: Register your smartphone

- Click Next. A page is displayed where you can enter metadata about the device type, such as a serial number or model. You don't need to specify this information for this demo. Just click Done.
- Provide a value for the authentication token. Remember this value for later. Then, click **Next**.
- Click **Done**.



There are two options for selecting a device authentication token.

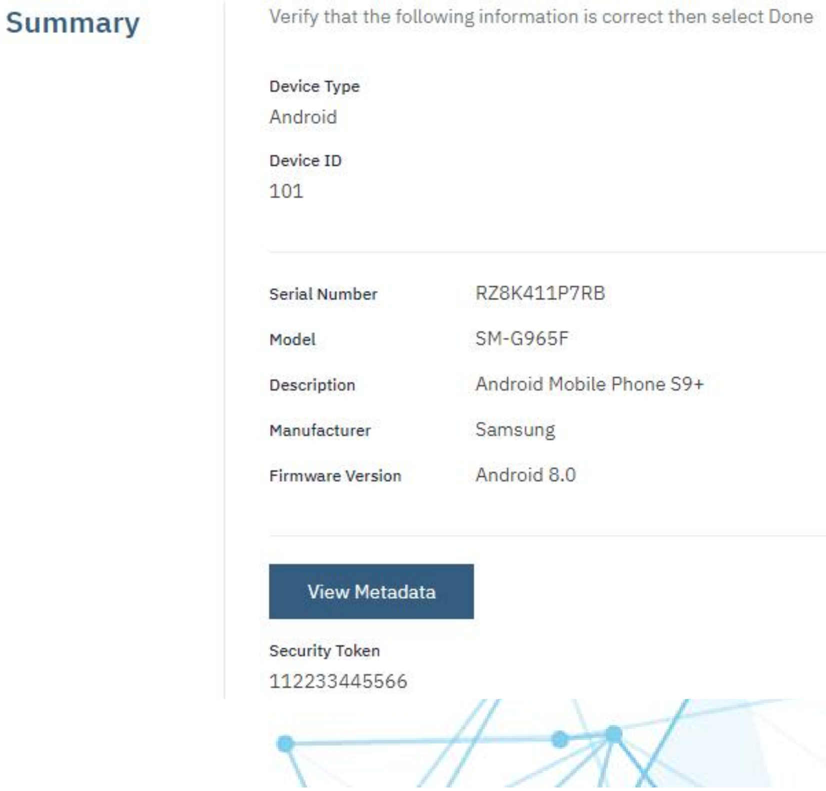
Auto-generated authentication token (default)

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

Self-provided authentication token

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

Authentication Token: 112233445566



Summary

Verify that the following information is correct then select Done

Device Type
Android

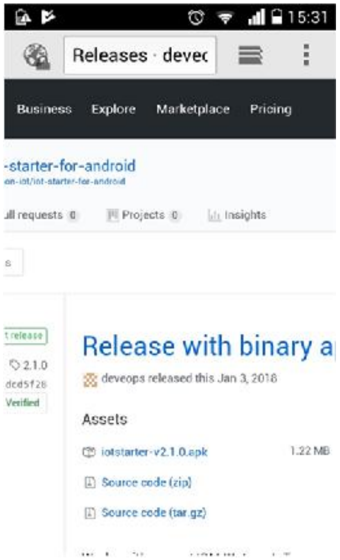
Device ID
101

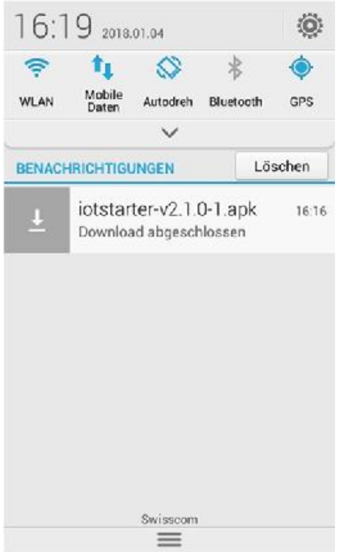
Serial Number	RZ8K411P7RB
Model	SM-G965F
Description	Android Mobile Phone S9+
Manufacturer	Samsung
Firmware Version	Android 8.0

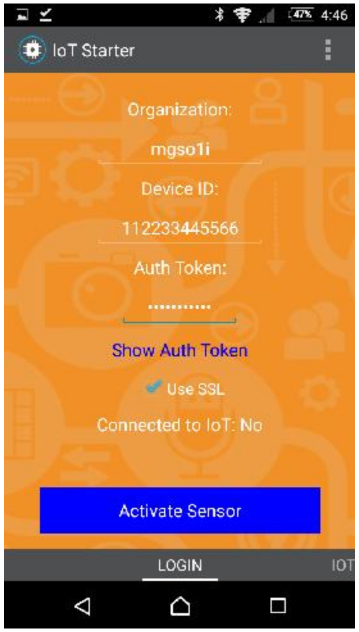
Security Token
112233445566

STEP 3: Prepare the Android Phone

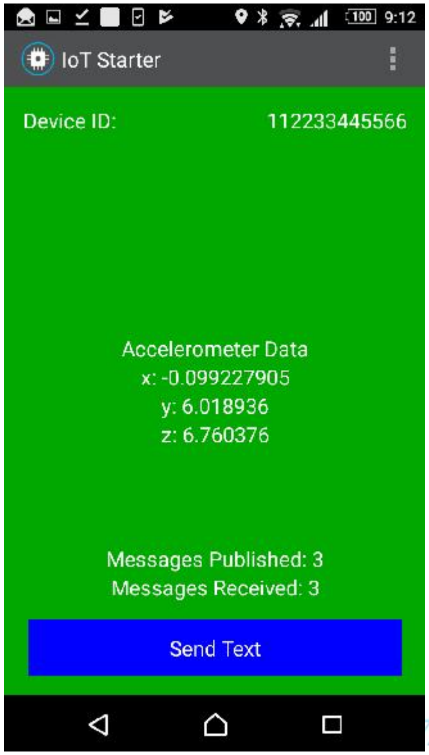
1. On your phone, go to **Settings** > **Security**. Under Device Administration, enable **Unknown sources**. Now you can install .apk files from outside of Google Play.
2. Open the browser on your phone, and enter this URL:
   `https://github.com/deveops/iot-starter-for-android/releases`
3. Search for the **iotstarter-v2.1.0.apk** link, and click the link to download the .apk file.
4. Click the downloaded file, and confirm that you want to install the app.
5. The IoT Starter app is now installed on your Android device.







Organization: mgso1i
Device ID: 112233445566
Auth Token: ••••••••••••
✓ Use SSL
Connected to IoT: No

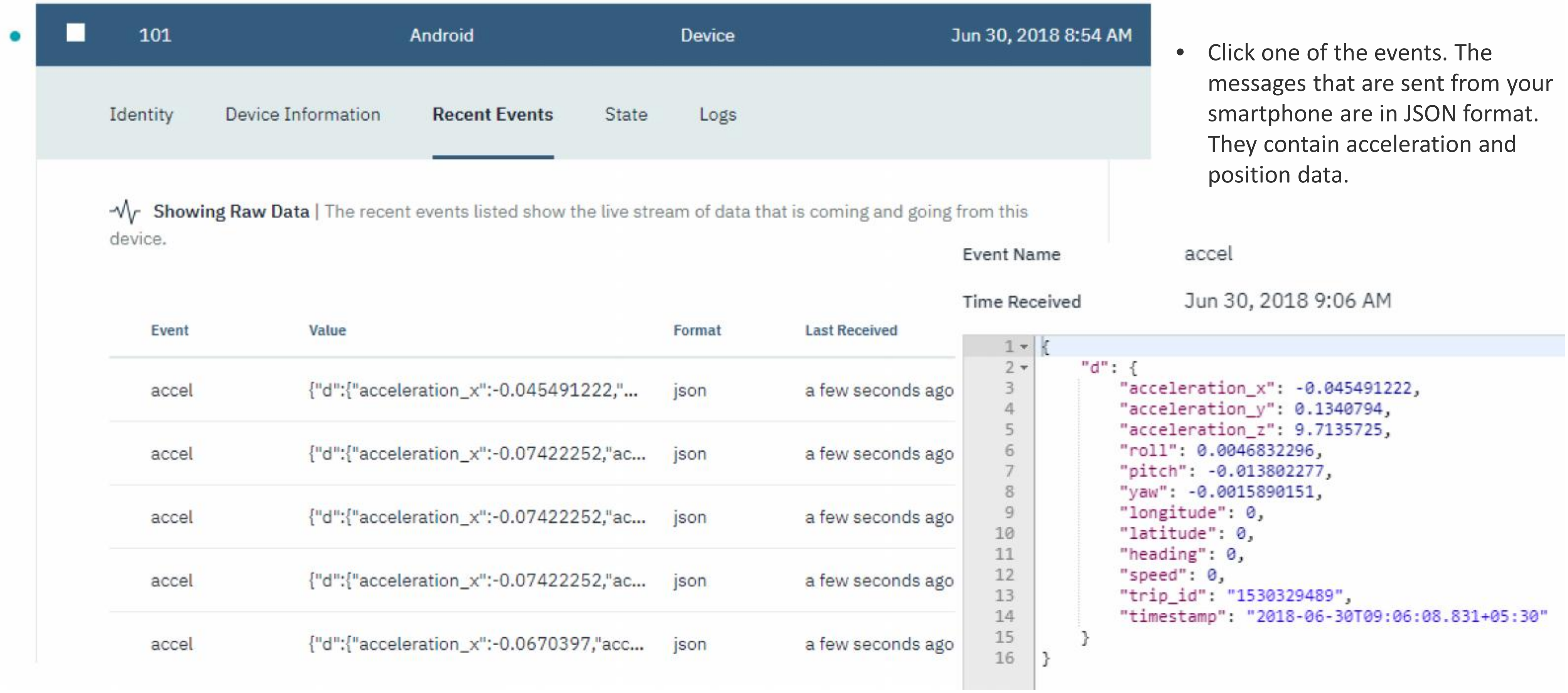


Device ID: 112233445566

Accelerometer Data
x: -0.099227905
y: 6.018936
z: 6.760376

Messages Published: 3
Messages Received: 3

STEP 4: Verify messages are getting delivered



101 | Android | Device | Jun 30, 2018 8:54 AM

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

Event	Value	Format	Last Received
accel	{"d":{"acceleration_x":-0.045491222,"...	json	a few seconds ago
accel	{"d":{"acceleration_x":-0.07422252,"ac...	json	a few seconds ago
accel	{"d":{"acceleration_x":-0.07422252,"ac...	json	a few seconds ago
accel	{"d":{"acceleration_x":-0.07422252,"ac...	json	a few seconds ago
accel	{"d":{"acceleration_x":-0.0670397,"acc...	json	a few seconds ago

Event Name accel

Time Received Jun 30, 2018 9:06 AM

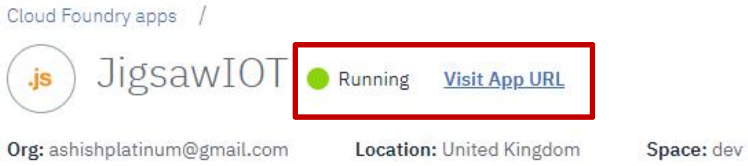
```
{
    "d": {
        "acceleration_x": -0.045491222,
        "acceleration_y": 0.1340794,
        "acceleration_z": 9.7135725,
        "roll": 0.0046832296,
        "pitch": -0.013802277,
        "yaw": -0.0015890151,
        "longitude": 0,
        "latitude": 0,
        "heading": 0,
        "speed": 0,
        "trip_id": "1530329489",
        "timestamp": "2018-06-30T09:06:08.831+05:30"
    }
}
```

- Click one of the events. The messages that are sent from your smartphone are in JSON format. They contain acceleration and position data.

STEP 5: Process messages in a Node-RED flow

Node-RED is a visual tool that makes it easy to wire and process Internet of Things messages.

- Now, we will enhance our IBM Cloud IOT app by using a Node-RED flow to process messages from your smartphone, and then send messages back to your smartphone. The phone will react on these messages by changing the background color in the app.
- Launch your app - https://jigsawiot.eu-gb.mybluemix.net/



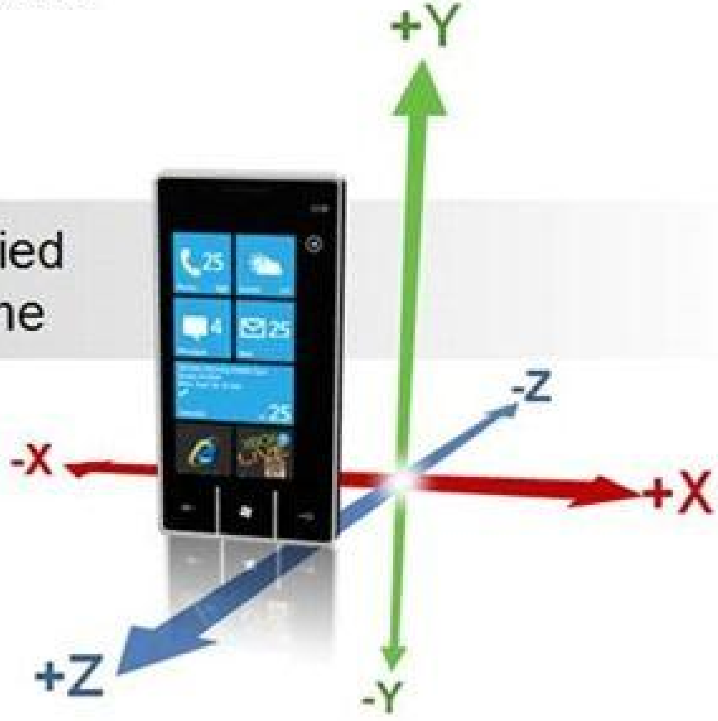
Org: ashishplatinum@gmail.com **Location:** United Kingdom **Space:** dev

- Open Node-RED flow editor - https://jigsawiot.eu-gb.mybluemix.net/red/
- The editor opens, containing a sample flow.

- Using the drag-and-drop features of this editor, you can plug together a flow of messages.
- Although you can create your own flow here, we will import the code below. But first, select all existing nodes, and delete them by pressing the Delete key.

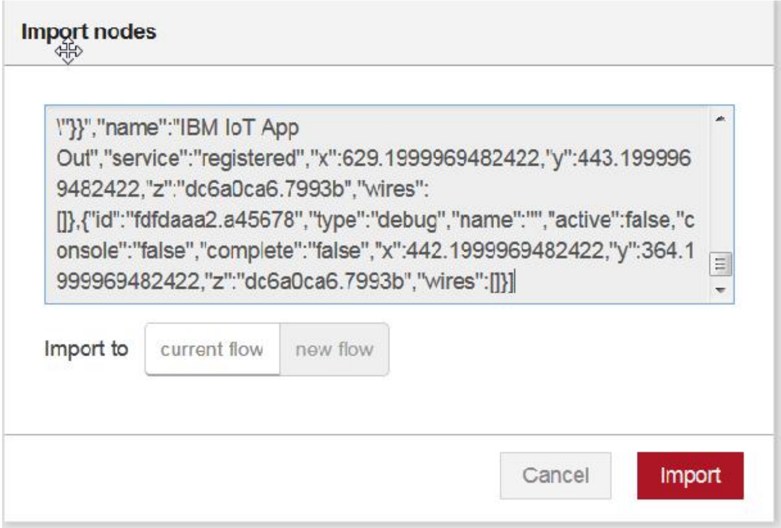
Accelerometer

Measures force applied on each axis over time



STEP 5a: Simple flow

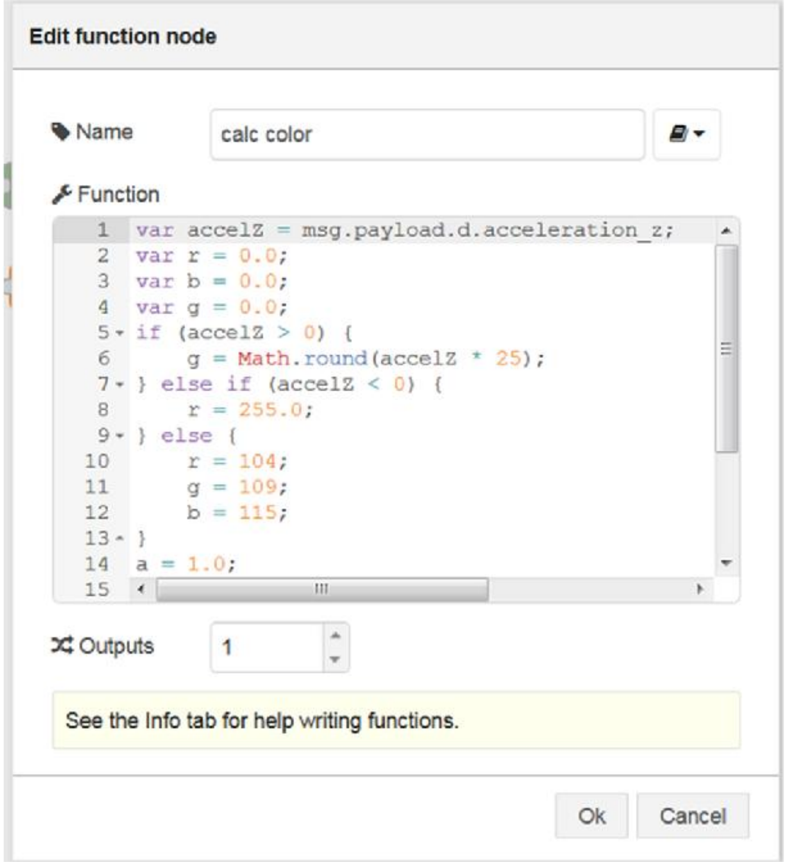
- Download the code (as a long single line of code) as a text file (nodeRedCode.txt) from GitHub.
- In the Node-RED editor, press **Ctrl-I** to open the Import Nodes dialog. Paste the code, and click **OK**.



Import nodes

```
\"}}","name":"IBM IoT App Out","service":"registered","x":629.1999969482422,"y":443.1999969482422,"z":"dc6a0ca6.7993b","wires":[]},{"id":"fdfdaaa2.a45678","type":"debug","name":"","active":false,"console":"false","complete":"false","x":442.1999969482422,"y":364.1999969482422,"z":"dc6a0ca6.7993b","wires":[]}]
```

Import to: current flow | new flow



Edit function node

Name: calc color

Function

```
var accelZ = msg.payload.d.acceleration_z;
var r = 0.0;
var b = 0.0;
var g = 0.0;
if (accelZ > 0) {
    g = Math.round(accelZ * 25);
} else if (accelZ < 0) {
    r = 255.0;
} else {
    r = 104;
    g = 109;
    b = 115;
}
a = 1.0;
```

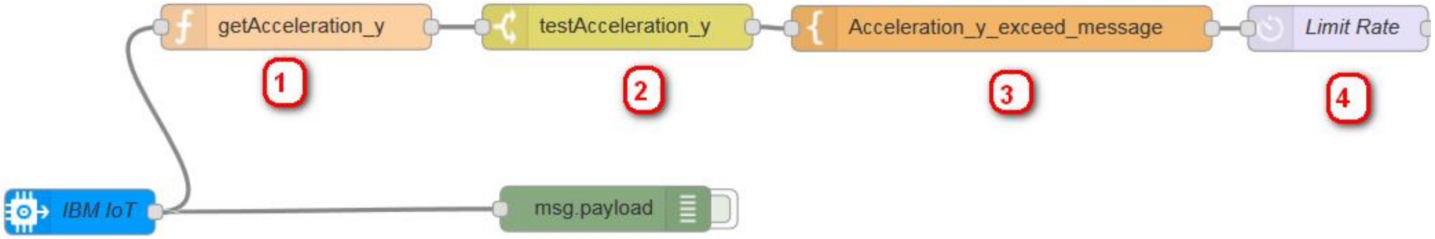
Outputs: 1

See the Info tab for help writing functions.

- Now you need to adapt the flow to your specific parameters. The only relevant parameter is the Device ID. Double-click the node **IBM IoT App out**. In the pop-up window, enter the Device ID that you used earlier (for example, 101), and click Import.
- Inspect the flow. Double-click the calc color node. It calculates the red, green, and blue values based on the incoming z-acceleration value, and passes them on as JSON data.

STEP 5b: Send alert when phone falls

- Extract the "acceleration_y" value from the JSON payload message
- Trigger a value against an acceleration value (such as 7 for Android)
- Debug the notification message
- Optionally, either post a tweet or send an email



- To extract the absolute acceleration_y value of the smartphone from the JSON payload message, we'll add and configure a **function node**. From the function section of the palette, drag a function node to your workspace. Double-click the function node to configure it with these values:
  - Specify `getAcceleration_y` as the name for this node.
  - In this function node, copy and paste the following code:
    - `return {payload: Math.abs(msg.payload.d.acceleration_y)};`
  - Click Done.

STEP 5b: Send alert when phone falls

- To create a trigger on the acceleration_y value, we'll add and configure a switch node. From the function section of the palette, drag a **switch node** to your workspace. Wire the function node to the switch node. Double-click the switch node to configure it with these values:
  - Specify `testAcceleration_y` as the name for this node.
  - Specify `msg.payload` as the property.
  - In the condition value drop-down list, select ">=?", and specify 0.5 for iOS or 7 for Android.
  - In the last drop-down list, select `stopping after first match`.
  - Click Done.

- To define the notification message that you want to send when your phone is falling, we'll add and configure a **template node**. From the function section of the palette, drag a template node to your workspace. Wire the switch node to the template node. Double-click the template node to configure it with these values:
  - Specify `Acceleration_y_exceed_message` as the name for this node.
  - Specify `msg.payload` as the property.
  - Copy and paste the following code to the node editor:
    - `Wow! Is your phone falling?`
    - `Its acceleration y = {{payload}}!`
  - Click Done.

STEP 5b: Send alert when phone falls

- To limit the number of messages for the notification and to avoid duplicate notification messages, we'll add and configure a **delay node**. From the function section of the palette, drag a delay node to your workspace. Wire the template node to the delay node. Double-click the delay node to configure it with these values:
  - Specify `Limit Rate` as the name for this node.
  - From the Action drop-down list, specify `Limit rate to`. Then, set the rate to 1 and select Minute.
  - Check the Drop intermediate message check box.
  - Click Done.

- To define the notification message that you want to send when your phone is falling, we'll add and configure a **template node**. From the function section of the palette, drag a template node to your workspace. Wire the switch node to the template node. Double-click the template node to configure it with these values:
  - Specify `Acceleration_y_exceed_message` as the name for this node.
  - Specify `msg.payload` as the property.
  - Copy and paste the following code to the node editor:
    - `Wow! Is your phone falling?`
    - `Its acceleration y = {{payload}}!`
  - Click Done.
- Double-click the **twitter** out node to configure it with these values:
  - In the twitter out dialog box, select the pencil icon next to the Twitter ID drop-down list.
  - In the next dialog box, click the button to authenticate with Twitter. In the window that is displayed, specify your Twitter credentials and authorize your IBM Cloud app to use your Twitter account.
  - Click Add to add that Twitter account.
  - In the twitter out dialog box, make sure that your ID is selected.

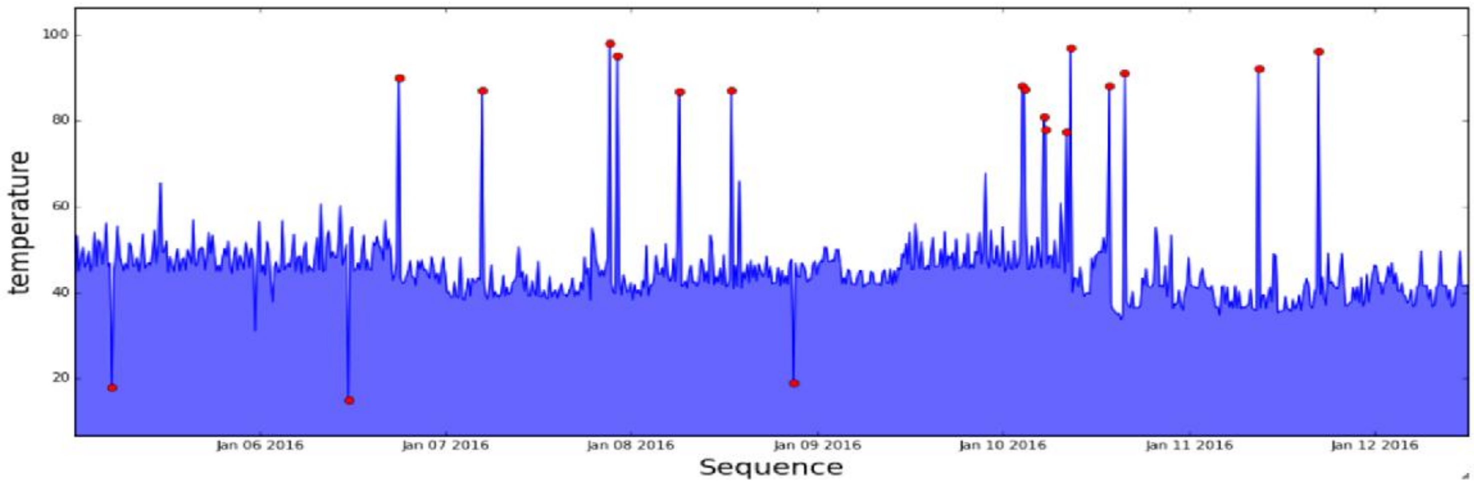
STEP 5b: Send alert when phone falls

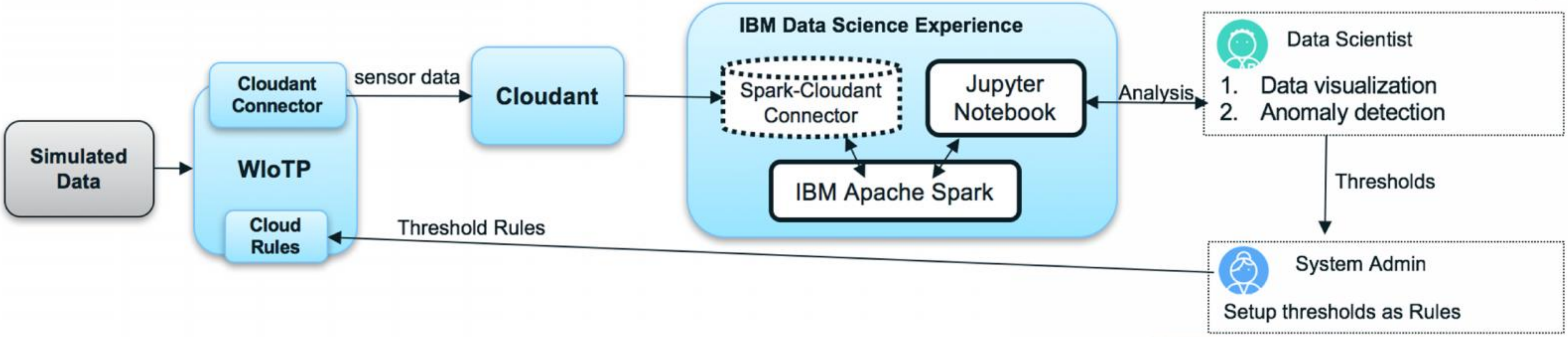
If you have an SMTP email account, you can have the app send an email with the notification that your smartphone is falling. You might need to configure security settings for your email to allow your app to send email.

For example, if you are using a Gmail account, you must enable the Allow less secure apps option in the Sign-in & security settings.

- Double-click the e-mail out node to configure it with these values:
  - In the To field, specify your email address.
  - In the Server field, specify your SMTP server address, such as smtp.gmail.com.
  - In the Port field, specify the SMTP TCP port that sends email, such as 465.
  - In the Userid field, specify the user name to use to authenticate with the SMTP server. This user name might be your email address.
  - In the Password field, specify the password that is associated with your user name.

Next Steps – Anomaly Detection





Simulated Data

WIoTP

Cloudant Connector

Cloud Rules

sensor data

Cloudant

IBM Data Science Experience

Spark-Cloudant Connector

Jupyter Notebook

IBM Apache Spark

Analysis

Data Scientist

1. Data visualization
2. Anomaly detection

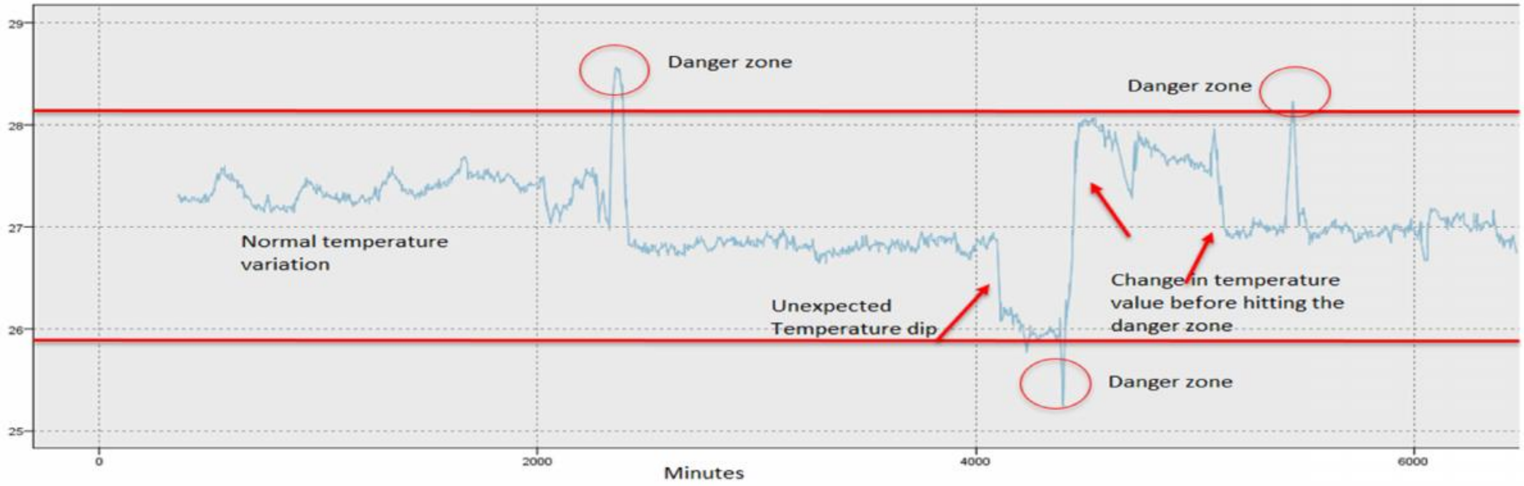
Thresholds

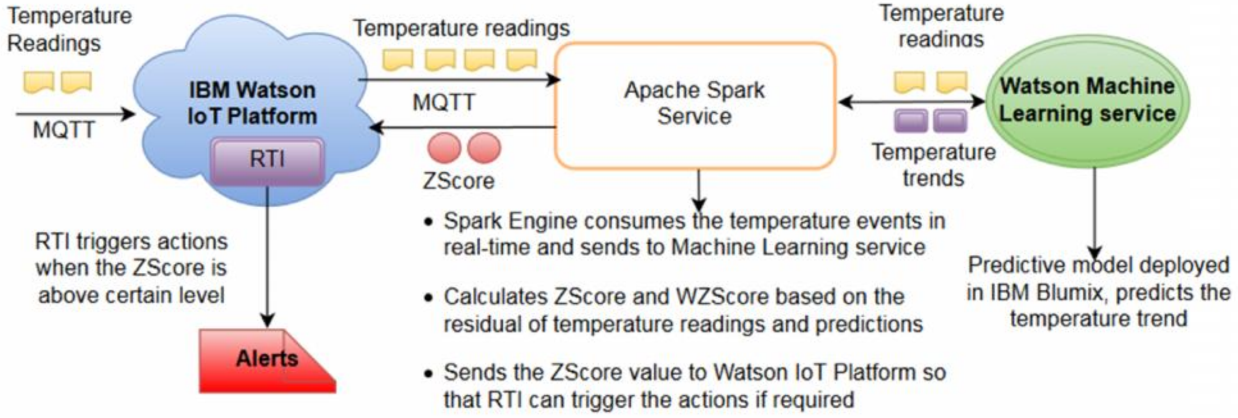
System Admin

Setup thresholds as Rules

Threshold Rules

Machine Learning for detecting anomalous behaviours of things




Temperature Readings
MQTT
IBM Watson IoT Platform
RTI
Temperature readings
MQTT
ZScore
Apache Spark Service
Temperature readings
Temperature trends
Watson Machine Learning service
RTI triggers actions when the ZScore is above certain level
Alerts
Predictive model deployed in IBM Blumix, predicts the temperature trend

- Spark Engine consumes the temperature events in real-time and sends to Machine Learning service
- Calculates ZScore and WZScore based on the residual of temperature readings and predictions
- Sends the ZScore value to Watson IoT Platform so that RTI can trigger the actions if required

ZScore - How abnormal the reading is comparing to all the values in history?

WZScore - How abnormal the reading is comparing to the neighboring values in time?

https://github.com/deveops/iot-starter-for-android/releases/download/2.1.0/nodeRedCode.txt

Backup

https://github.com/deveops/iot-starter-for-android/releases

https://github.com/deveops/iot-starter-for-android/releases/download/2.1.0/nodeRedCode.txt

STEP 6: Plotting Acceleration

- Download the code (as a long single line of code) as a text file (nodeRedCode.txt) from GitHub.
- Rest of the code is here : https://www.ibm.com/developerworks/library/iot-mobile-phone-iot-device-bluemix-apps-trs/index.html

https://www.dropbox.com/sh/qgwmt3q1swmqeye/AACMK_RyYsHE4NIlVmAjDRP4a

Jigsaw@123$

Analysis using Watson Analytics Studio

- Create account: https://www.ibm.com/cloud/watson-studio

- https://github.com/ashish-jigsaw/workshop

- New project for analysis
- New notebook with URL -