

Internet of Things Part 3

Máster en Big Data y Data Science

Pedro Luis Pérez Latorre

Internet of Things

Índice

4. Práctica Básica IoT	03	6. Práctica avanzada IoT	47
4.0. Creación cuenta IBM Cloud	04	6.1. Modificación del flujo Node-RED	48
4.1. Instalación Servicio Internet Of Things Platform	05	6.2. Configuración Watson Studio	51
4.2. Lanzamos IoT Platform	07	6.3. Conexión Cloudant con Notebook	57
4.3. Añadir un dispositivo que envíe mensajes MQTT	08		
4.4. Instalación Android App	12	7. Recomendaciones finales	63
4.5. Verificación de las lecturas en IoT Platform	22		
4.6. Monitoriza lecturas con un dashboard	23		
5. Práctica intermedia IoT	29		
5.1. Despliegue App Node-RED	30		
5.2. Conectar con el servicio IoT Platform	32		
5.3. Habilitar Continuous Delivery	33		
5.4. Accedemos a Node-RED	36		
5.5. Instalando extensión IBM en Node-RED	38		
5.6. Procesado de los mensajes en Node-RED	40		
5.6. Persistencia de datos en Cloudant	44		

4. Práctica básica IoT

Una primera toma de contacto con dispositivos y plataforma IoT

- 4.0. Creación cuenta IBM Cloud
- 4.1. Instalación Servicio Internet Of Things Platform
- 4.2. Lanzamos IoT Platform
- 4.3. Añadir un dispositivo que envíe mensajes MQTT
- 4.4. Instalación Android App
- 4.5. Verificación de las lecturas en IoT Platform
- 4.6. Monitoriza lecturas con un dashboard

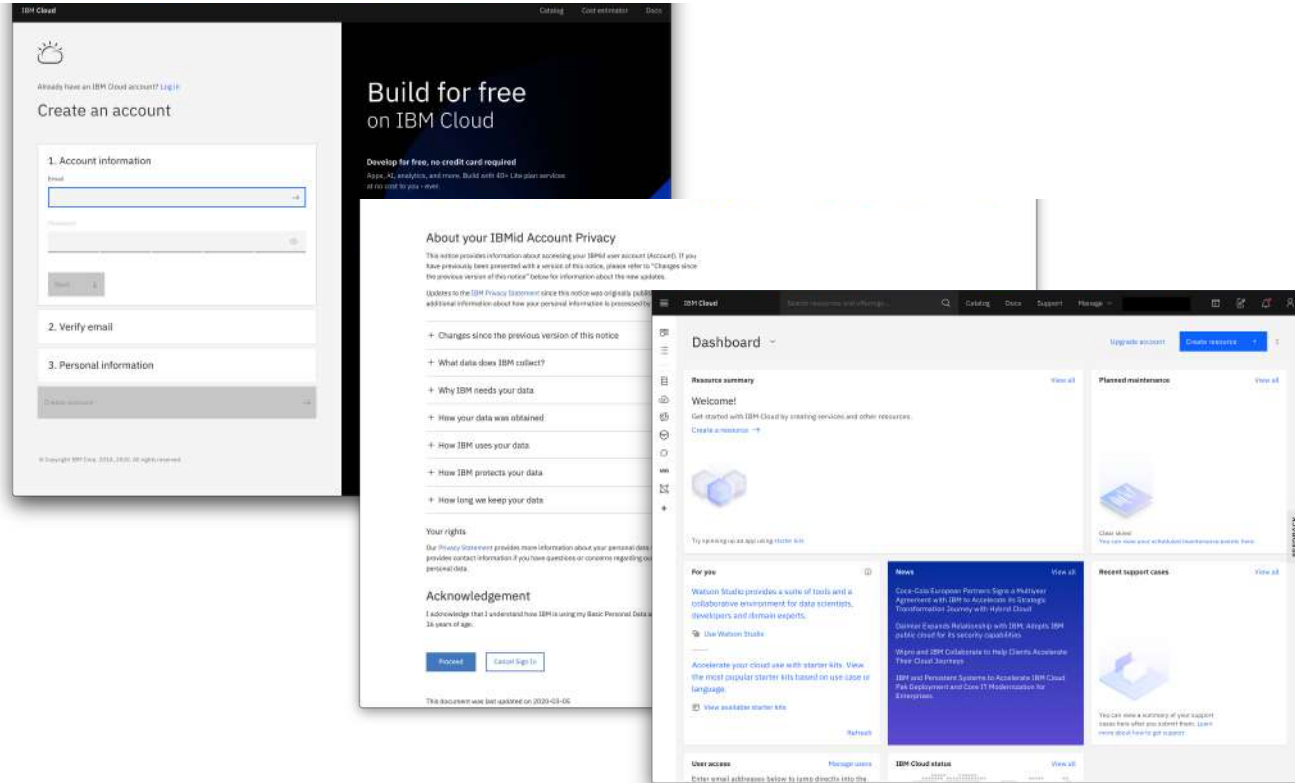
4. Práctica básica IoT

4.0. Creación cuenta IBM Cloud

Si aún no se dispone de una cuenta de IBM Cloud, el primer paso es crear una cuenta:

[Link a creación de cuenta IBM Cloud gratuita](#)

Se trata de un proceso sencillo que no conlleva más de 5 minutos y que no precisa de incluir información bancaria ni de tarjetas de crédito

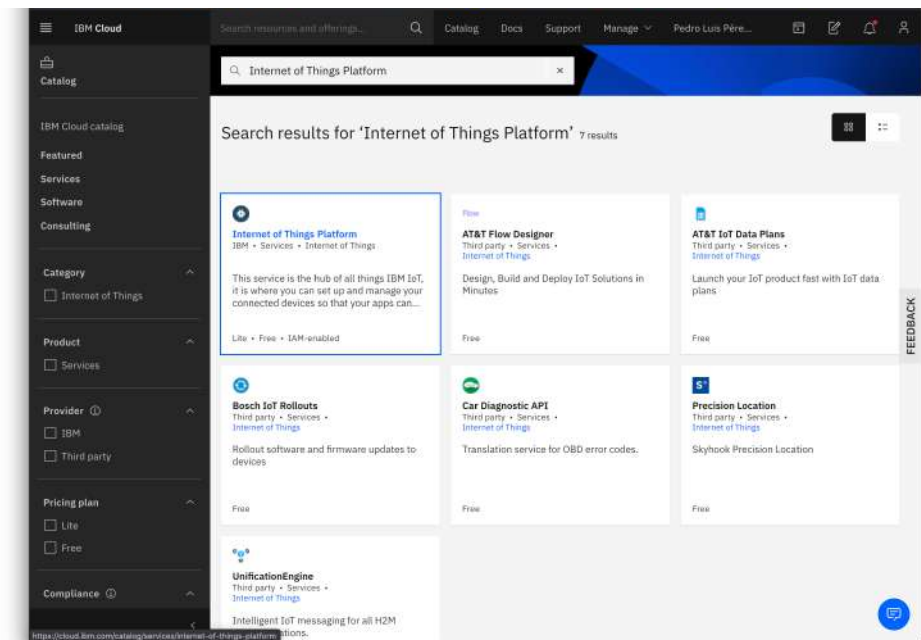


4. Práctica básica IoT

4.1. Instalación Servicio Internet Of Things Platform

Seguimos los siguientes pasos para crear una aplicación Node-RED:

1. Log in en [IBM Cloud](#).
2. Hacemos click en [Catalog](#) y buscamos [Internet of Things Platform](#)
3. Hacemos click sobre la casilla de [Internet of Things Platform](#)



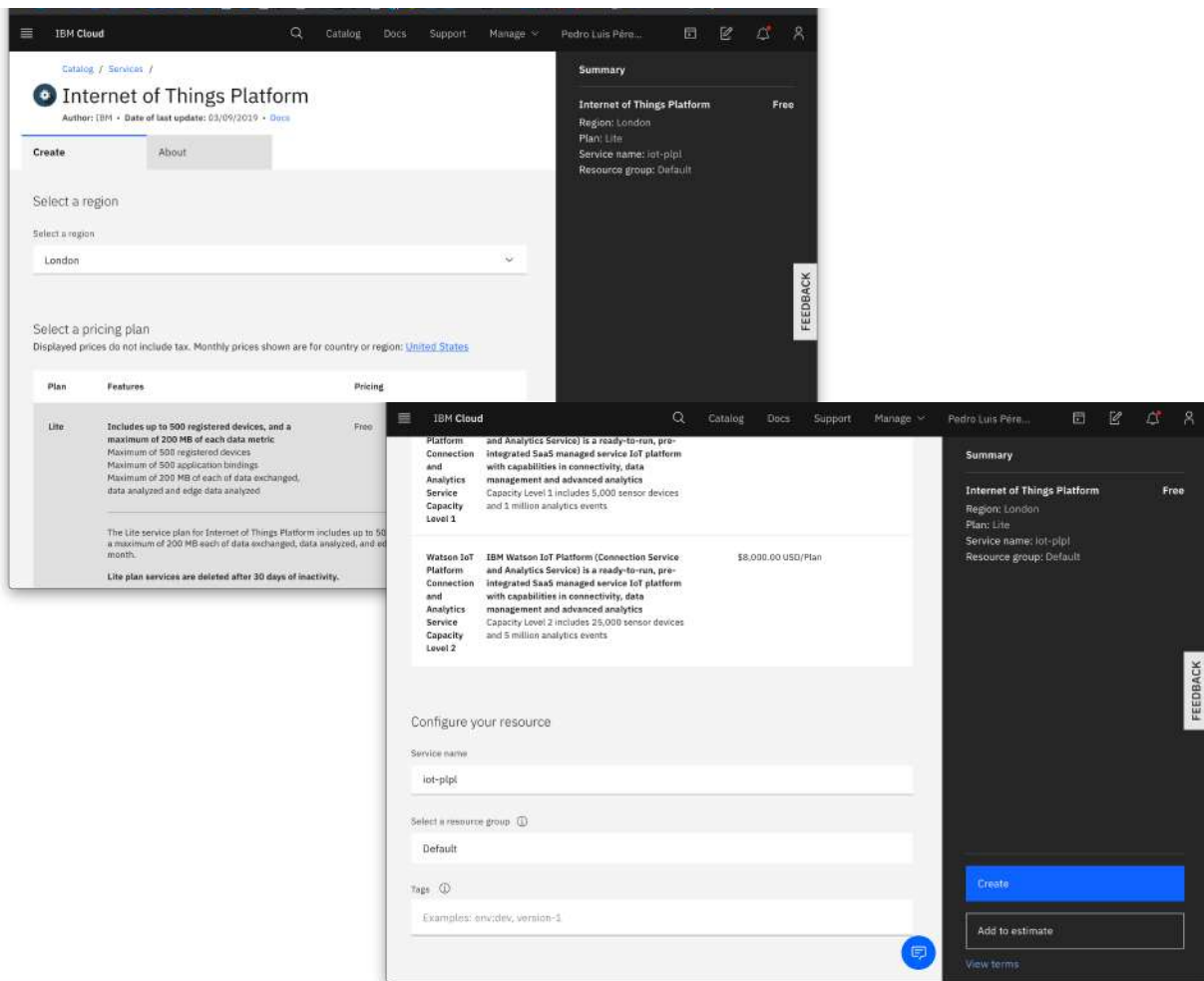
4. Práctica básica IoT

4.1. Instalación Servicio Internet Of Things Platform

4. En la pestaña **Create** y completamos los siguientes campos:

- **Region** – Muy importante! Se debe seleccionar la misma ubicación para todos los servicios
 - **London** – en el caso de este tutorial
- **Pricing plan** – se deja - **Lite**
- **Service name** – Se precisa un nombre único, por lo que una sugerencia es el uso de iniciales y el nombre del servicio
 - **iot-plpl** – en el caso de este tutorial
- **Resource group** – Se deja **Default**
- **Tags** – Se deja **vacío**

5. Una vez completados los campos pulsamos en **Create** y esperamos

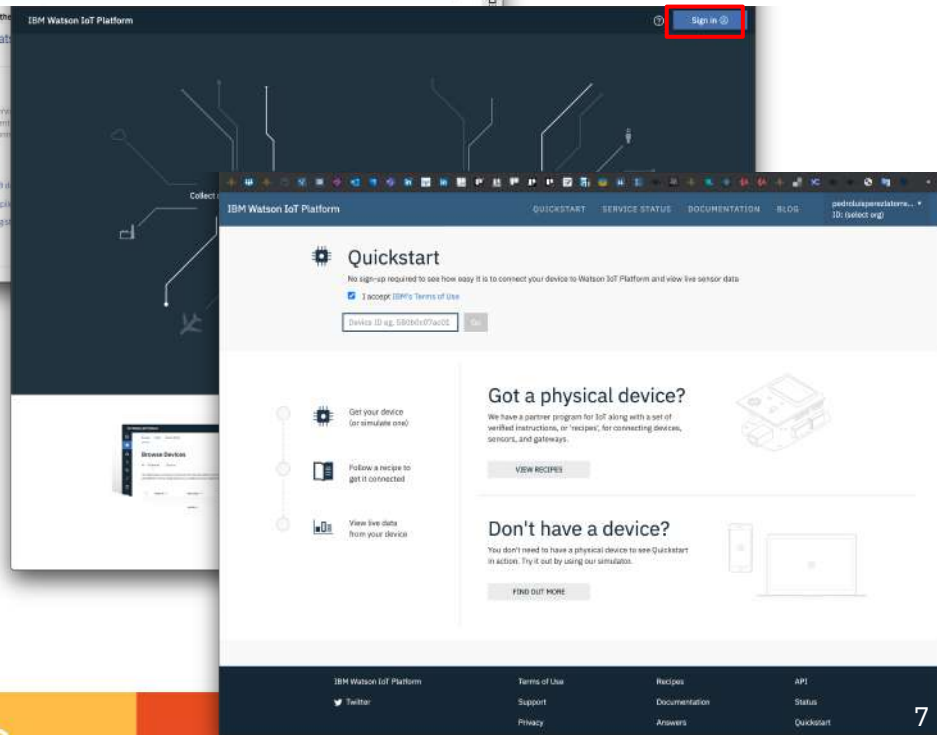
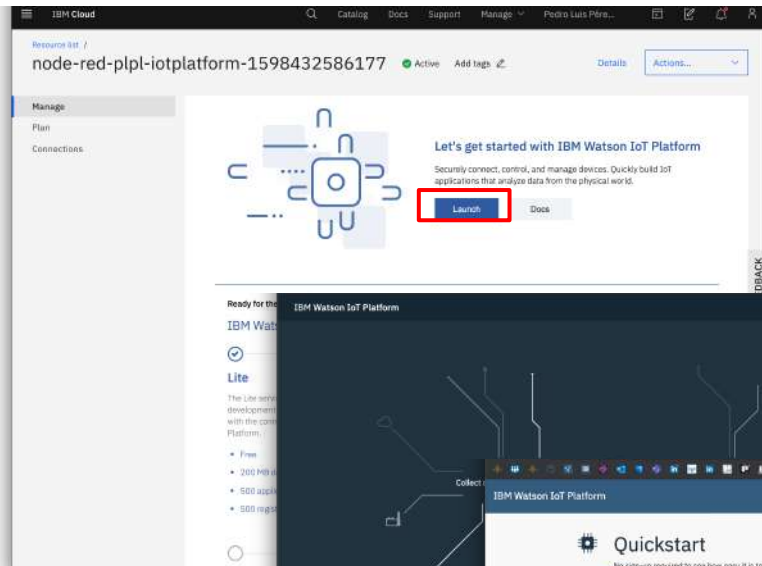


4. Práctica básica IoT

4.2. Lanzamos IoT Platform

Una vez instalado el servicio:

1. Sobre la pantalla pulsamos en [Launch](#)
3. Por último, hacemos [log in](#)
4. Ya estamos dentro de nuestra plataforma IoT

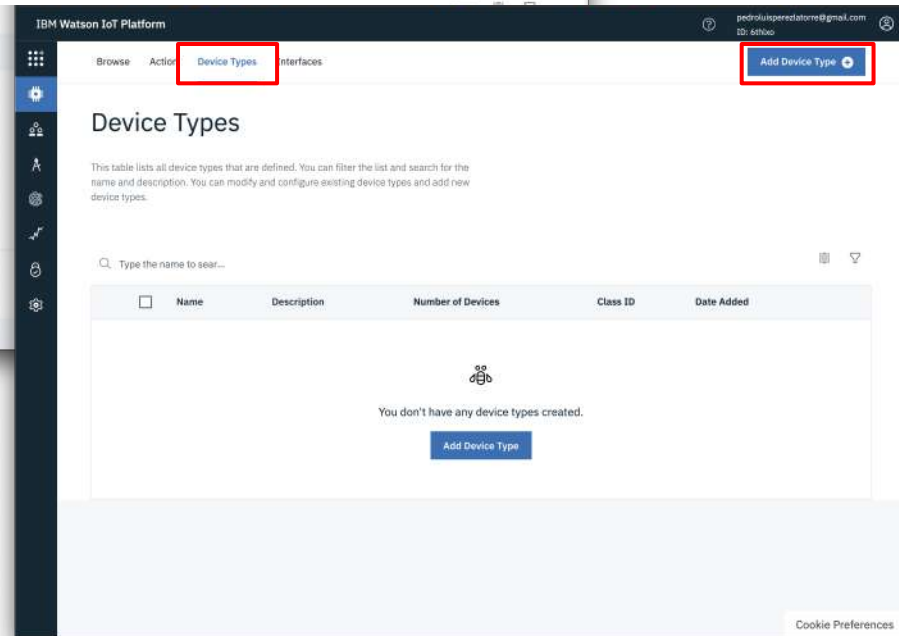
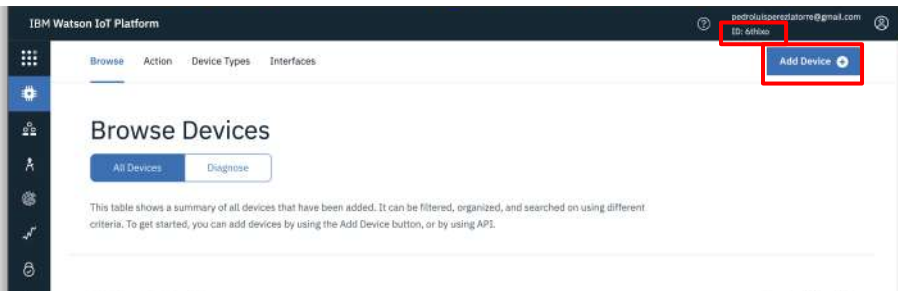


4. Práctica básica IoT

4.3. Añadir un dispositivo que envíe mensajes MQTT

Con la consola de IoT Platform abierta:

1. Primero nos fijamos en el ID, en el caso del autor [d5idpl](#). Este identificador enlazará la plataforma con los dispositivos que conectemos.
2. A continuación, para añadir un dispositivo pulsamos sobre [Add Device](#)
3. En la nueva ventana y en la pestaña [Device Types](#) pulsamos sobre [Add Device Type](#)



4. Práctica básica IoT

4.3. Añadir un dispositivo que envíe mensajes MQTT

En la plataforma, es posible tener varios tipos de dispositivos, y varios dispositivos en cada tipo. Un tipo de dispositivo es un grupo de activos que comparten características; por ejemplo, pueden proporcionar los mismos datos de sensores.

- En nuestro caso, el nombre del tipo de dispositivo debe ser **Android**. Pulsamos **Next**
- En la siguiente pantalla se pueden introducir metadatos sobre el tipo de dispositivo, como un número de serie o un modelo. No es necesario especificar esta información para esta práctica. Simplemente click en **Finish**.

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Type

Identity Device Information

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

Type Or

Name

Description

Cancel Next

Device Types

This table lists all device types that are defined. You can filter the list and search for the name and description. You can modify and configure existing device types and add new device types.

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Type

Identity Device Information

These attributes will be used as a template for new devices that are assigned this device type [Edit Metadata](#)

Serial Number	<input type="text" value="Enter Serial Number"/>	Manufacturer	<input type="text" value="Enter Manufacturer"/>
Model	<input type="text" value="Enter Model"/>	Device Class	<input type="text" value="Enter Device Class"/>
Description	<input type="text" value="Enter Description"/>	Firmware Version	<input type="text" value="Enter Firmware Version"/>
Hardware Version	<input type="text" value="Enter Hardware Version"/>	Descriptive Location	<input type="text" value="Enter Descriptive Location"/>

Back Finish

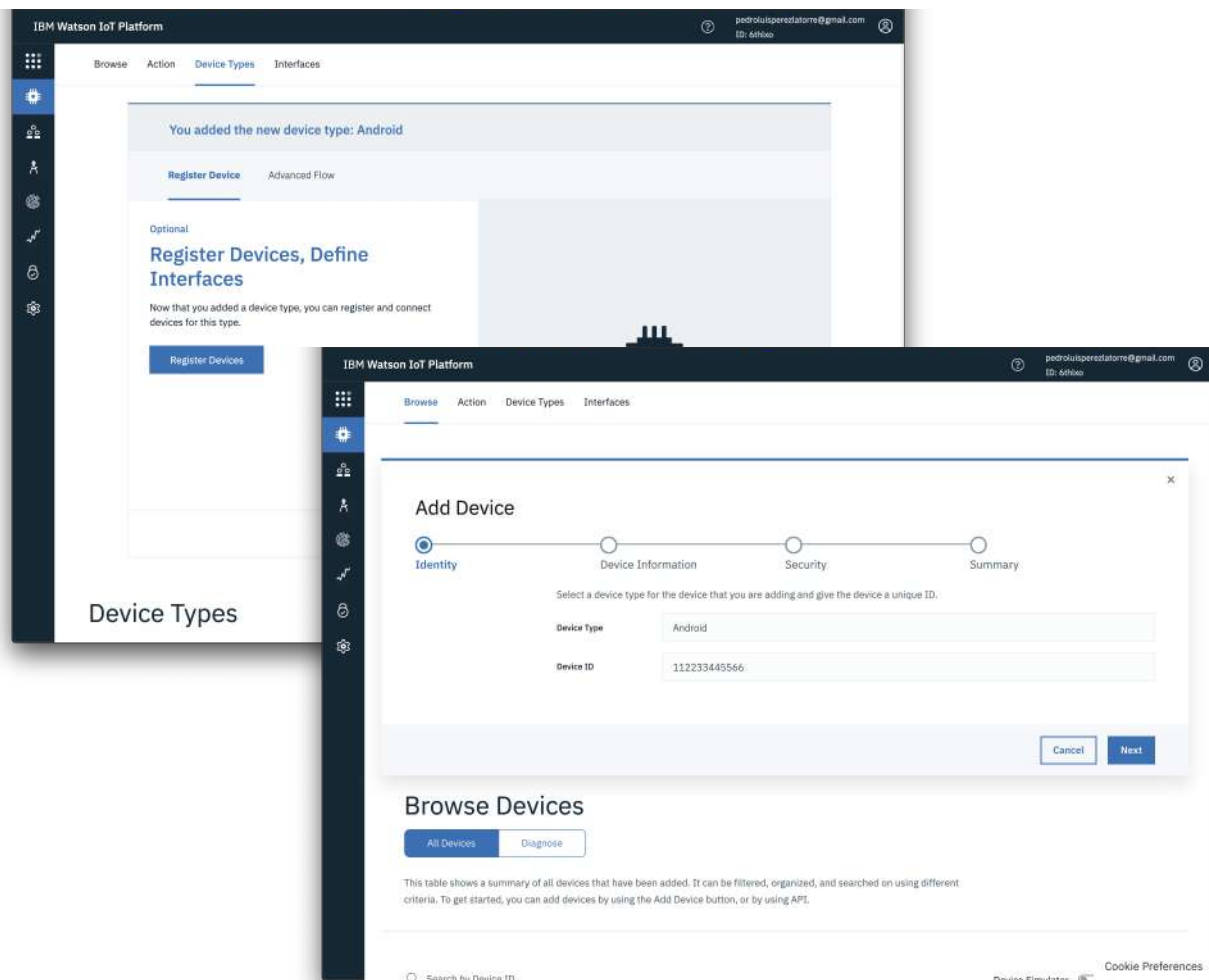
Device Types

This table lists all device types that are defined. You can filter the list and search for the name and description. You can modify and configure existing device types and add new device types.

4. Práctica básica IoT

4.3. Añadir un dispositivo que envíe mensajes MQTT

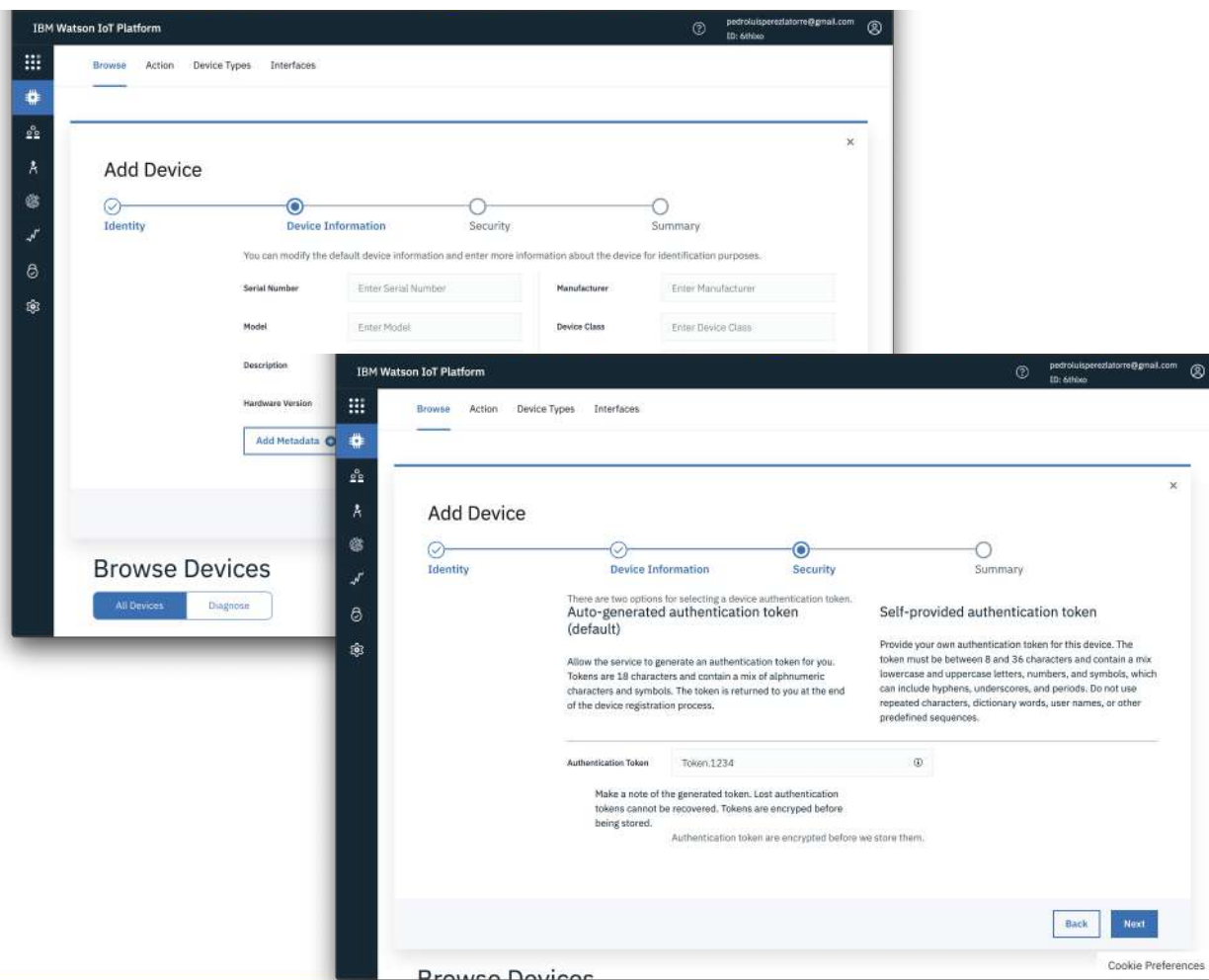
- Una vez creado el tipo de dispositivo, registramos uno nuevo haciendo click en [Register Devices](#).
- La identificación del dispositivo puede ser, por ejemplo, la dirección MAC de un smartphone. Sin embargo, debe ser única dentro de la plataforma. Para la práctica, introducimos [112233445566](#) y después pulsamos [Next](#)



4. Práctica básica IoT

4.3. Añadir un dispositivo que envíe mensajes MQTT

- Se muestra una página en la que se pueden introducir **metadatos** sobre el dispositivo. La dejamos en **blanco** y haga clic en **Next**.
- En la página de seguridad, se debe introducir un valor para el token de autenticación. Se debe recordar este valor para más tarde, en el caso de la práctica **Token.1234**. A continuación, click en **Next**.
- Por último, hacemos click en **Finish**



4. Práctica básica IoT

4.4. Instalación Android App

En esta parte de la práctica se va “crear” un sensor capaz de enviar lecturas a nuestra plataforma IoT.

Para esto, es posible utilizar un dispositivo Android o bien, un emulador.

Si sigues la vía de la emulación, existe libertad de elección en cuanto al software.

En la práctica se va a utilizar

[Android Studio](#)

Android Studio está disponible para **Windows, Mac OSX y Linux**. Sus instalaciones están documentadas en la página oficial.



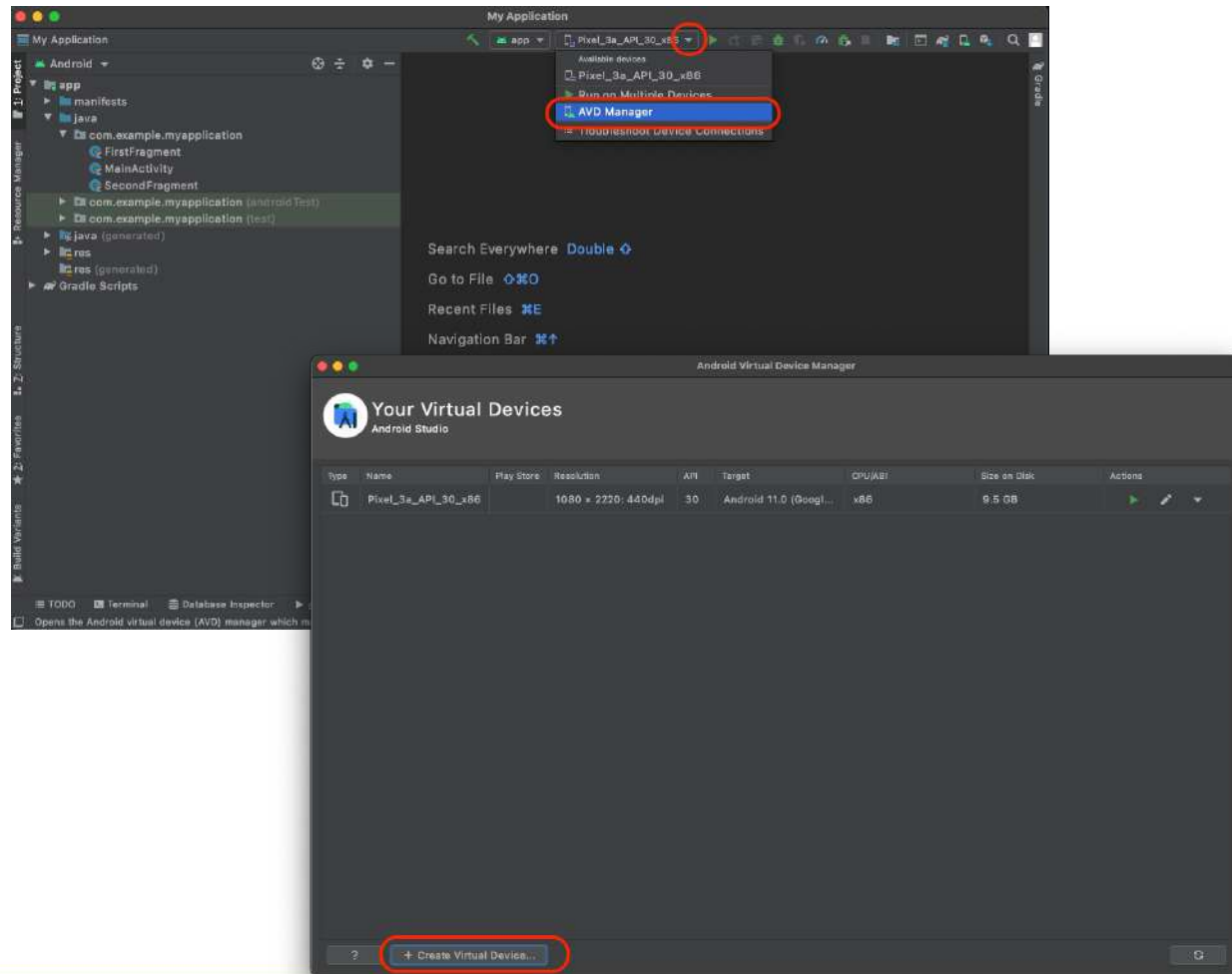
4.4. Instalación Android App

4.4.1. Descargar y lanzar el dispositivo

Una vez esté instalado y corriendo Android Studio, debemos descargar una imagen de Android que incorpore **Play Store**.

Para ello, pulsamos sobre el desplegable del dispositivo y pulsamos sobre **AVD Manager**.

En la nueva ventana, pulsamos sobre **+ Create Virtual Device**



4.4. Instalación Android App

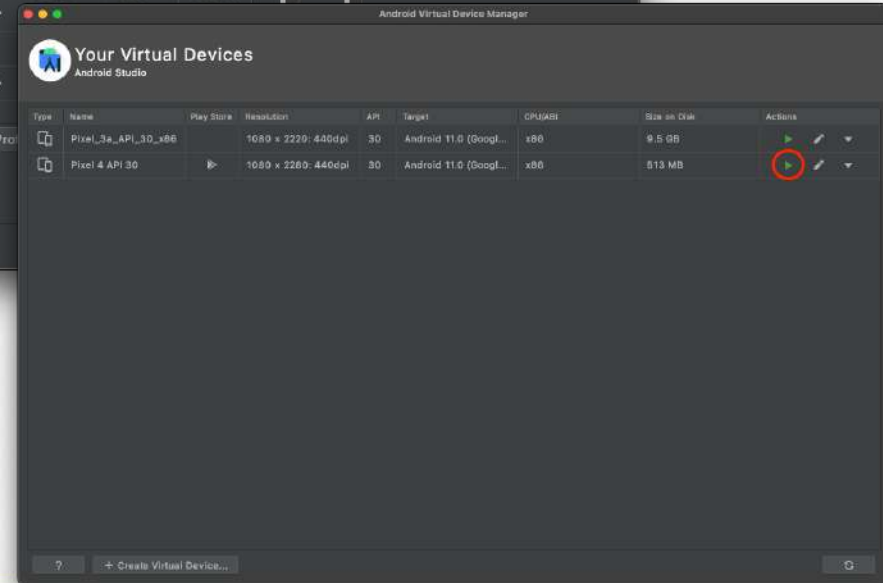
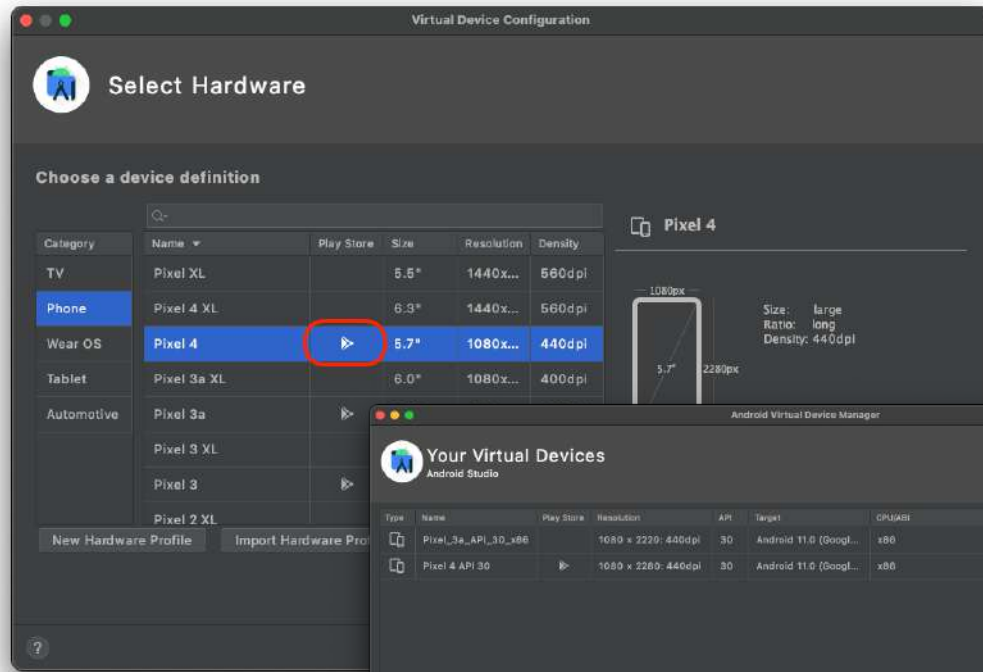
4.4.1. Descargar y lanzar el dispositivo

En la ventana de selección de dispositivos, elegimos uno que disponga del servicio [Play Store](#).

Para el caso de la práctica se ha elegido el [Pixel 4 API 30](#), pero se podría elegir cualquier otro con PlayStore.

A continuación pulsamos en [Next](#) y continuamos con la instalación según los pasos que nos indiquen.

Una vez instalado, podemos pulsar sobre el botón de [Play](#) para iniciar el emulador.

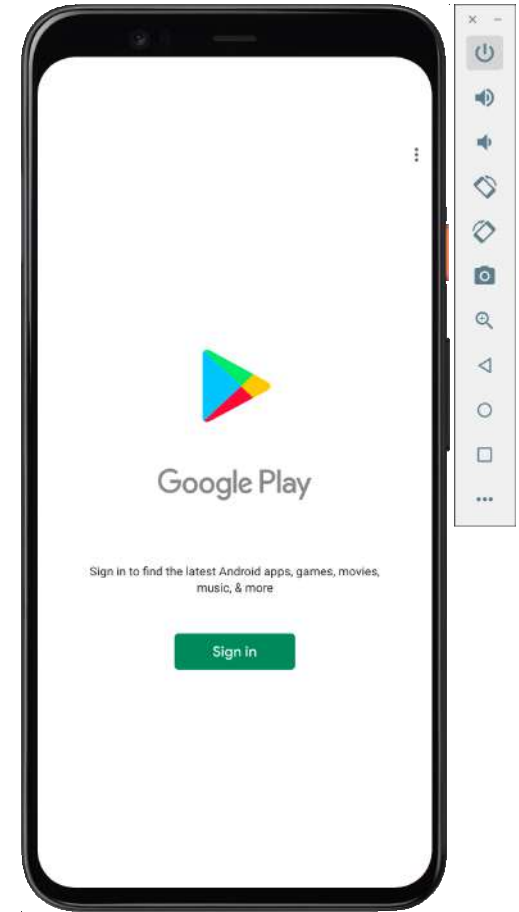
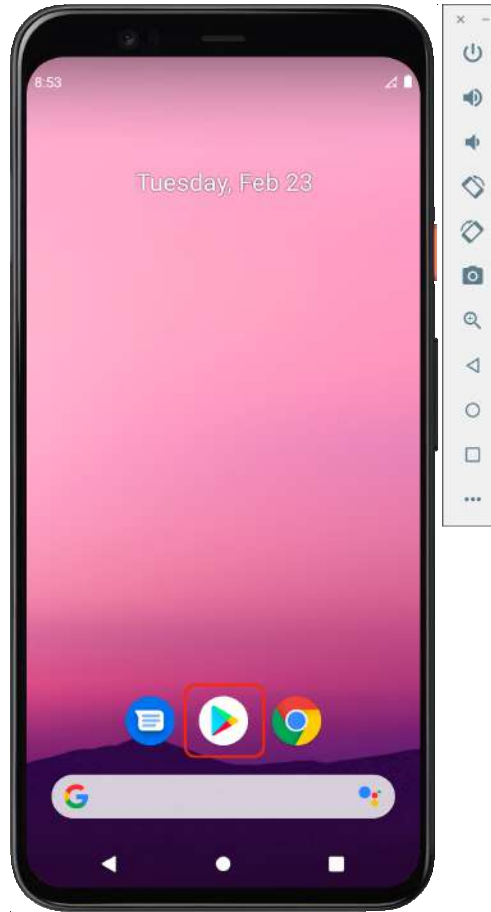


4.4. Instalación Android App

4.4.2. Login en Play Store y descarga de App

Una vez el dispositivo esté funcionando, solo queda tener iniciar [Play Store](#) con una [cuenta de Google](#).

El desarrollo de la práctica no incurre ningún coste económico, por lo que se puede utilizar una cuenta de [Gmail](#) si se dispone de ella, o bien abrirse una nueva.



4.4. Instalación Android App

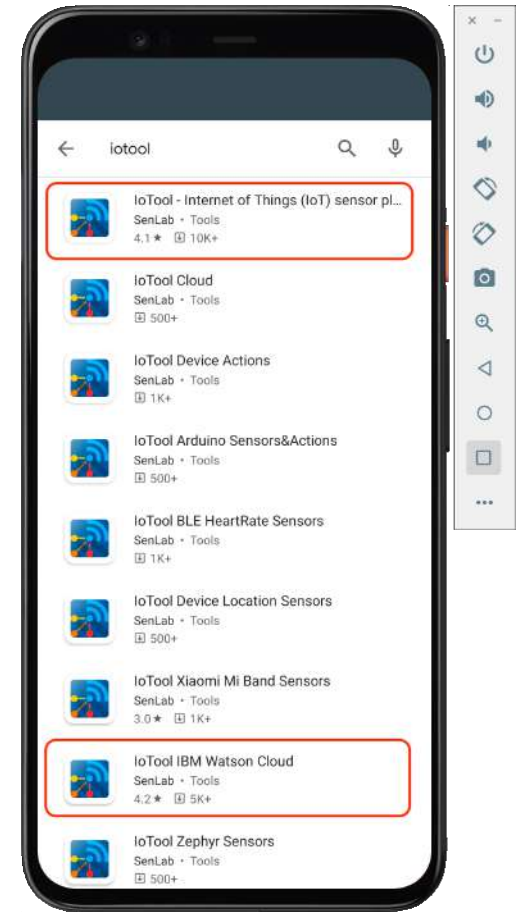
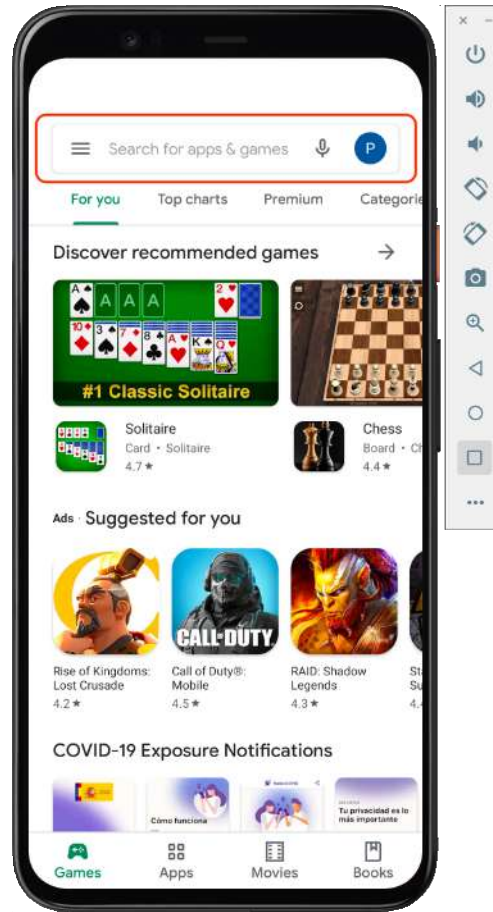
4.4.2. Login en Play Store y descarga de App

Una vez hayamos realizado el login, utilizamos la barra superior para buscar la app **IoTool**

De todas las opciones que aparecen, nos quedamos con la primera:

IoTool – Internet of Things (IoT) sensor platform

IoTool IBM Watson Cloud



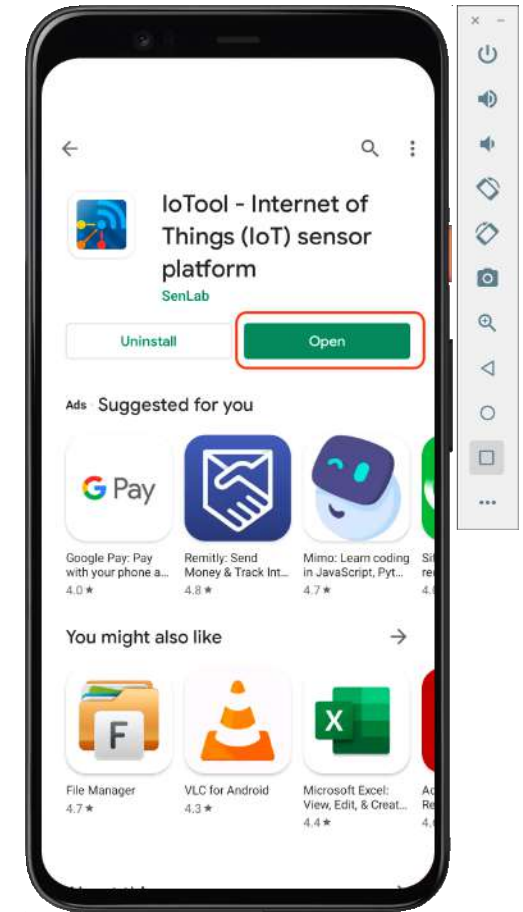
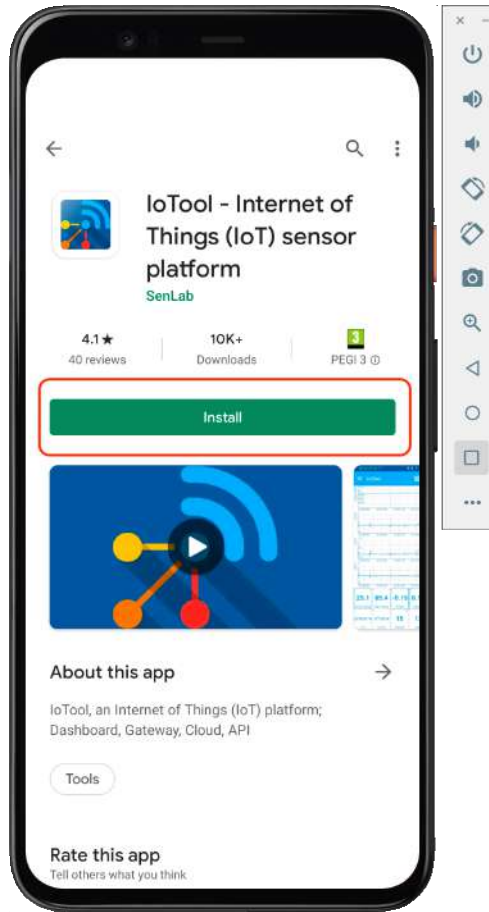
4.4. Instalación Android App

4.4.2. Login en Play Store y descarga de App

Pulsamos sobre [Install](#) y dejamos que tenga lugar el proceso.

A continuación pulsamos sobre [Open](#) para lanzar la app.

Nos hará una petición de permisos que permitiremos con [Allow](#) y a continuación podremos ver un rápido **tutorial**.

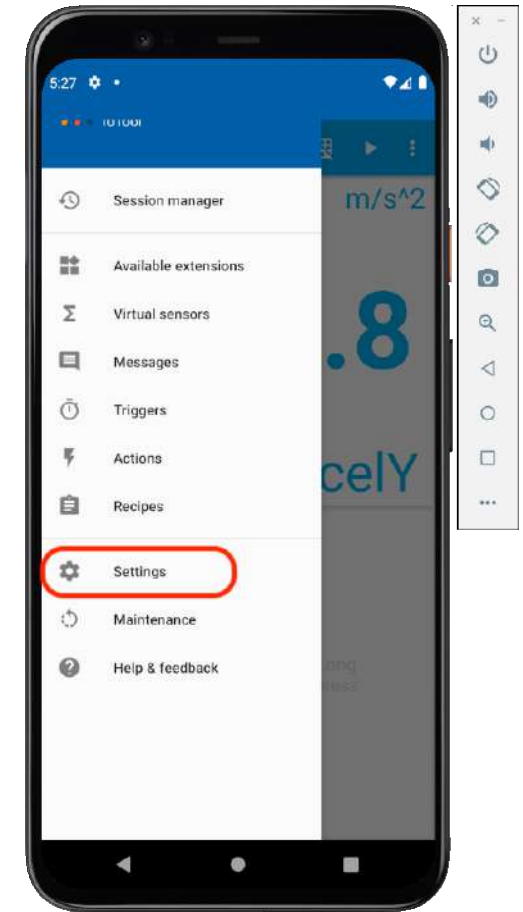
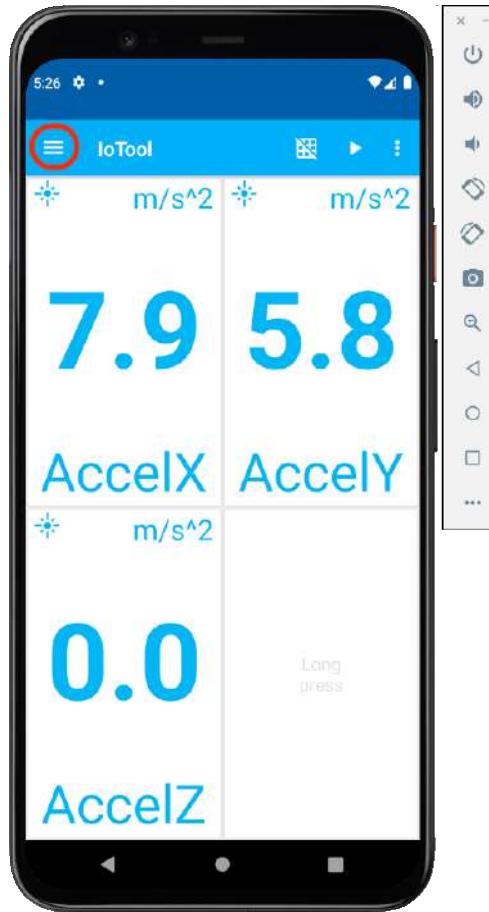


4.4. Instalación Android App

4.4.3. Configuración de la App

Una vez hemos instalado y lanzado la app de **IoTool**, procedemos a su configuración.

En la pantalla de inicio, pulsamos sobre el icono de la “**hamburguesa**” y a continuación en **Settings**



4.4. Instalación Android App

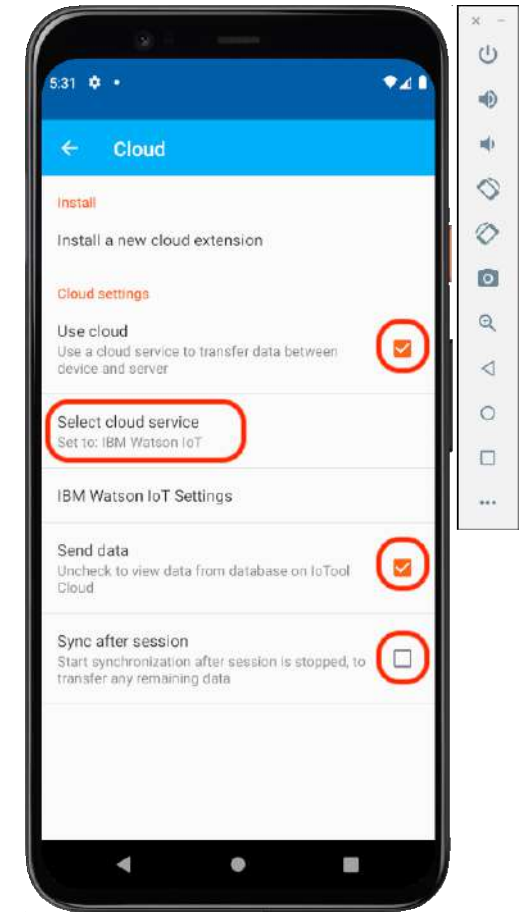
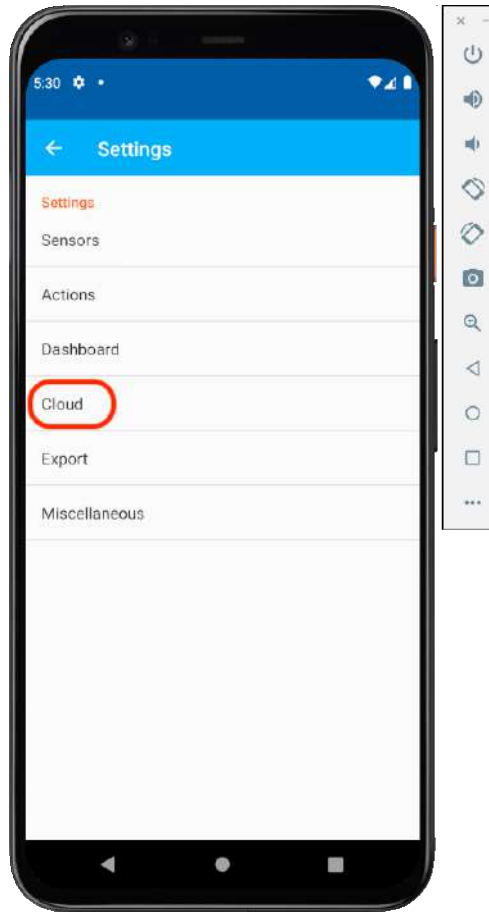
4.4.3. Configuración de la App

En la pantalla de configuración, pulsamos sobre **Cloud**

En la pantalla de configuración de Cloud elegimos como servicio **IBM Watson IoT**

En el resto de opciones dejamos marcado **Use cloud** y **Send data**.

Sync after session lo dejamos sin marcar.



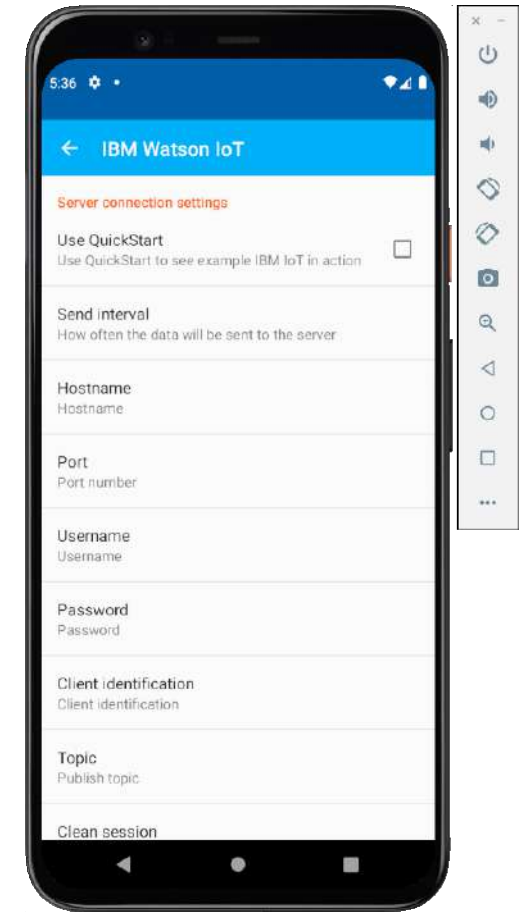
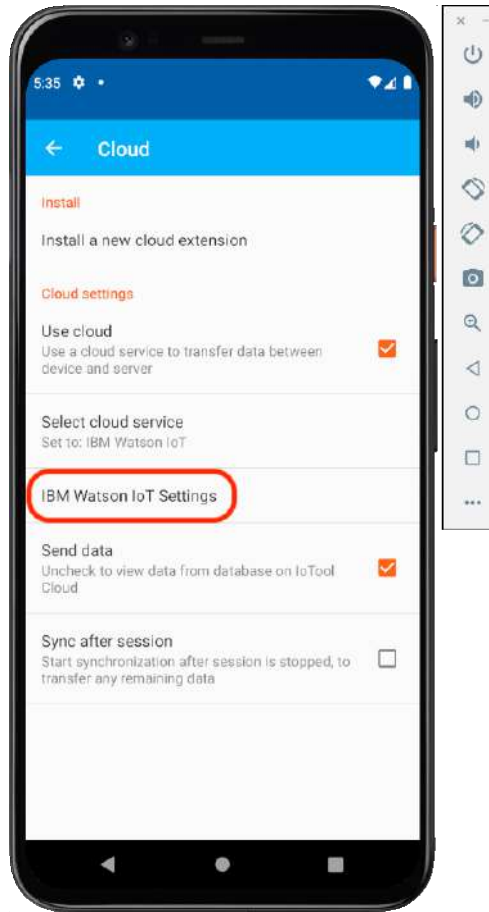
4.4. Instalación Android App

4.4.3. Configuración de la App

Aún en la pantalla de configuración de Cloud ahora pulsamos sobre **IBM Watson IoT Settings**

En esta pantalla debemos configurar los siguientes campos:

- Send interval: 1s
- Hostname:
ssl:// d5idpl.messaging.internetofthings.ibmcloud.com
- Port: 8883
- Username: use-token-auth
- Password: **Token.1234**
- Client identification:
d:d5idpl:Android:112233445566
- Topic: iot-2/evt/accel/fmt/json
- Clean session: ON
- Quality of Service: 2
- Condensed: OFF



4.4. Instalación Android App

4.4.4. Generación de datos

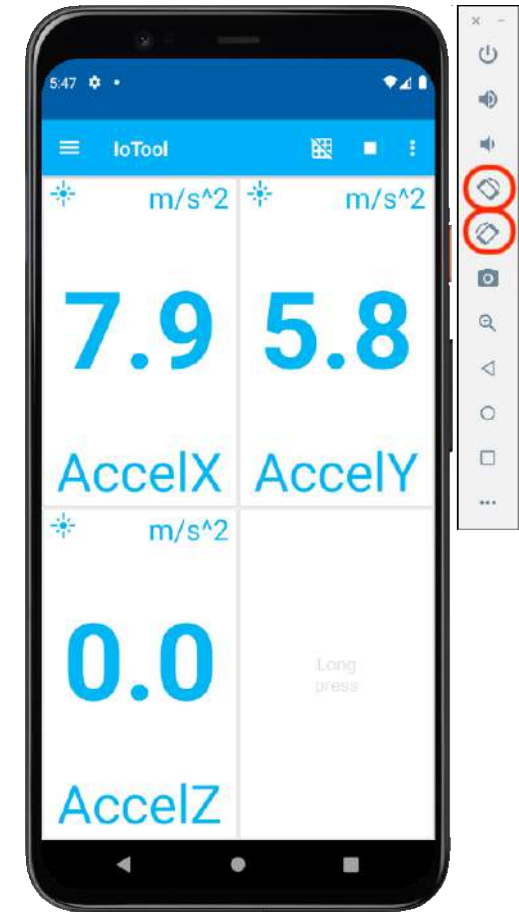
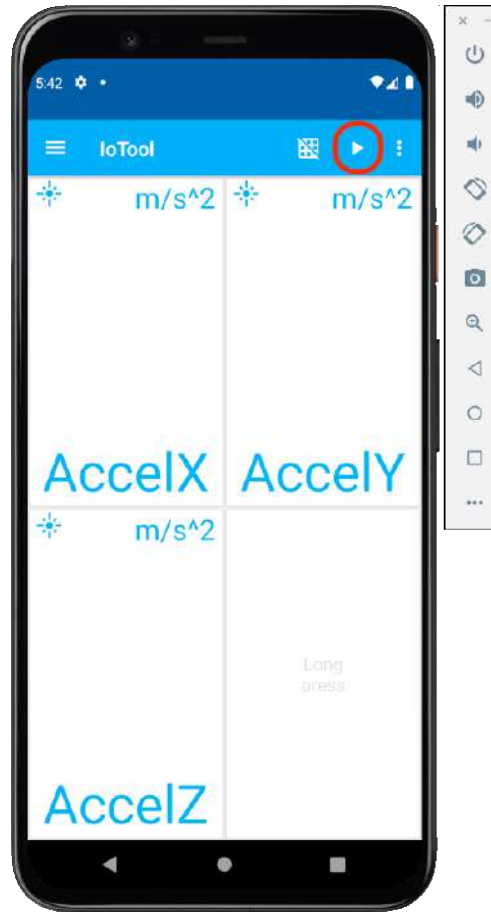
Una vez configurada la app, volvemos a la pantalla principal.

Para iniciar el registro de datos, pulsamos en el botón Play.

Una vez esté la aplicación activa, veremos las lecturas de los acelerómetros de nuestro terminal.

Si se trata de un terminal físico real, podemos agitarlo, girarlo y moverlo para ver como los sensores registran datos.

Si es un dispositivo digital, nos ayudamos de los botones de rotación para generar lecturas.



4. Práctica básica IoT

4.5. Verificación de las lecturas en IoT Platform

De nuevo a la consola de Watson IoT Platform:

1. En la pestaña lateral **Devices** y en la pestaña superior **Browse** deberíamos visualizar nuestro dispositivo.
2. Haciendo click sobre el icono de la **flecha** a la derecha de tu ID de dispositivo accedemos a una página de "Eventos recientes". Deberíamos ver los eventos que vienen de tu smartphone.
3. Haciendo click en alguno de los **eventos** veremos los mensajes que se envían desde el smartphone en formato **JSON**.

The image displays three overlapping screenshots of the IBM Watson IoT Platform interface. The top screenshot shows the 'Browse Devices' page with a table of devices. The second screenshot shows the 'Device Drilldown' page for device ID 112233445566, displaying connection information and recent events. The third screenshot shows the 'Event Payload' for a specific event, displaying a JSON message.

Device Drilldown - 112233445566

Connection Information

Basic connection information about this device.

Property	Value
Device ID	112233445566
Device Type	Android
Date Added	26 Aug 2020 12:58
Added By	pedroluisperelatorre@gmail.com
Connection Status	Connected
Connection Time	26 Aug 2020 15:26
Client Address	37.11.228.33 SecureToken

Recent Events

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":4.903325,"acceleration_y":...	json	a few seconds ago
accel	["d":{"acceleration_x":4.903325,"acceleration_y":...	json	a few seconds ago
accel	["d":{"acceleration_x":4.903325,"acceleration_y":...	json	a few seconds ago
accel	["d":{"acceleration_x":4.903325,"acceleration_y":...	json	a few seconds ago
accel	["d":{"acceleration_x":4.903325,"acceleration_y":...	json	a few seconds ago

Event Payload

Event Name: accel

Time Received: 26 Aug 2020 15:31

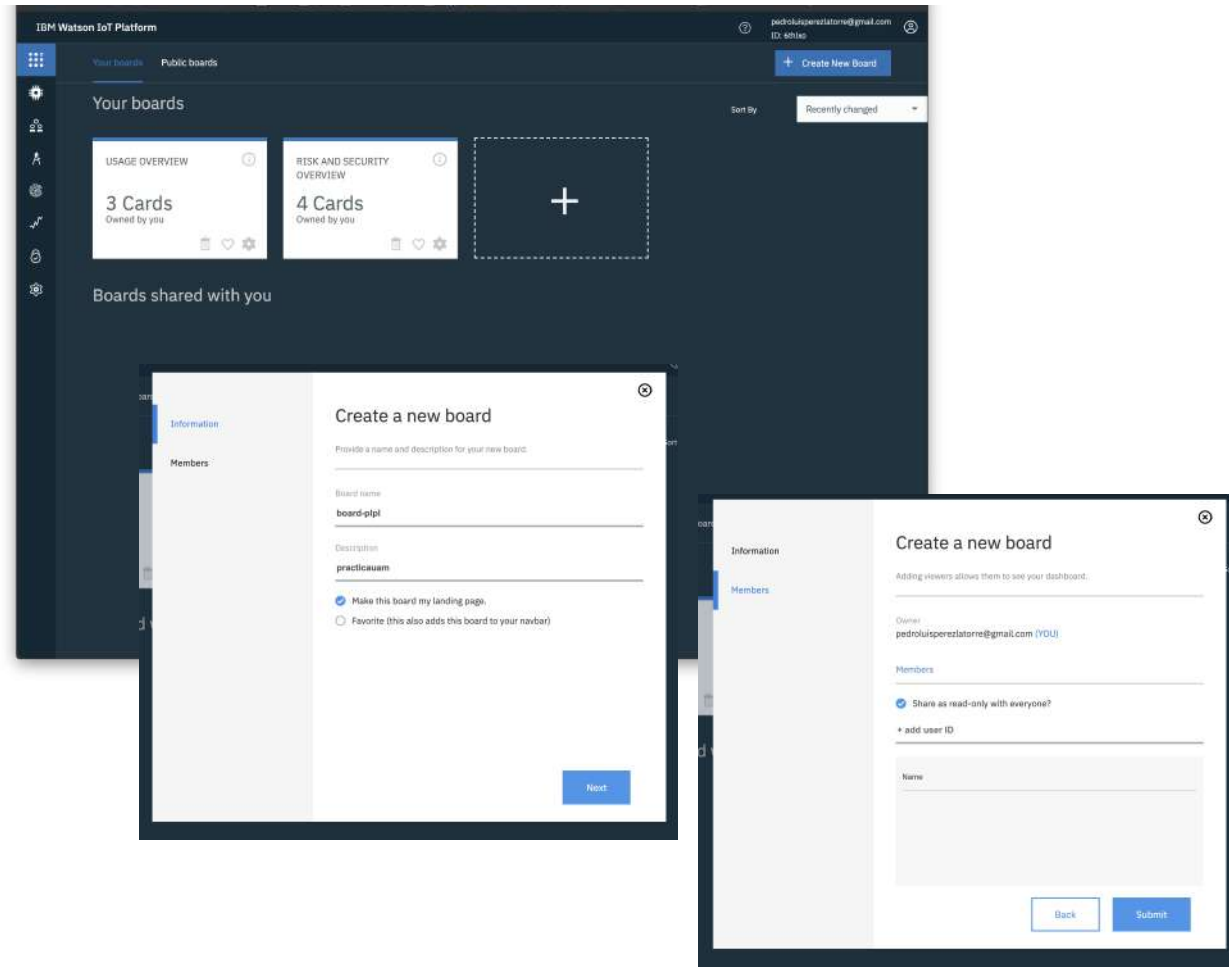
```
{
  "d": {
    "acceleration_x": 4.903325,
    "acceleration_y": -4.903325,
    "acceleration_z": -9.80665,
    "pitch": 0,
    "roll": 0,
    "yaw": 0,
    "longitude": 0,
    "latitude": 0,
    "heading": 0,
    "speed": 0,
    "timestamp": "2020-08-26T15:31:07.531-02:00"
  }
}
```

4. Práctica básica IoT

4.6. Monitoriza lecturas con un dashboard

Una vez hemos comprobado que nuestra plataforma capta las señales producidas por el sensor, pasamos a recogerlas en un dashboard para su monitorización.

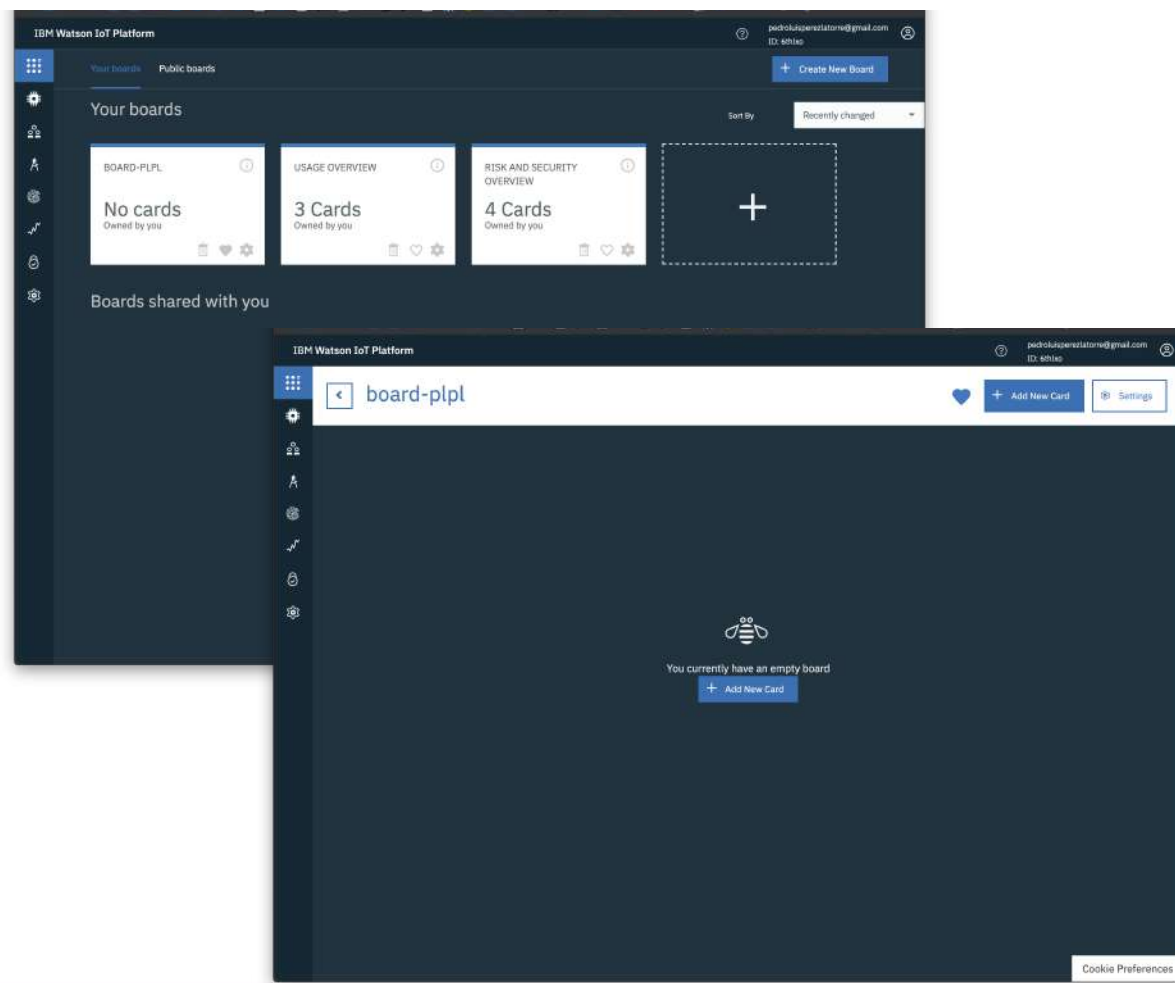
1. Hacemos click en la pestaña lateral **Boards** donde visualizamos los dashboards disponibles
2. Haciendo click sobre el icono de **+** creamos un nuevo board.
3. Rellenamos el **Board name** y la **Description** (opcional) y pulsamos **Next**
4. En la siguiente pantalla configuramos la privacidad y pulsamos en **submit**



4. Práctica básica IoT

4.6. Monitoriza lecturas con un dashboard

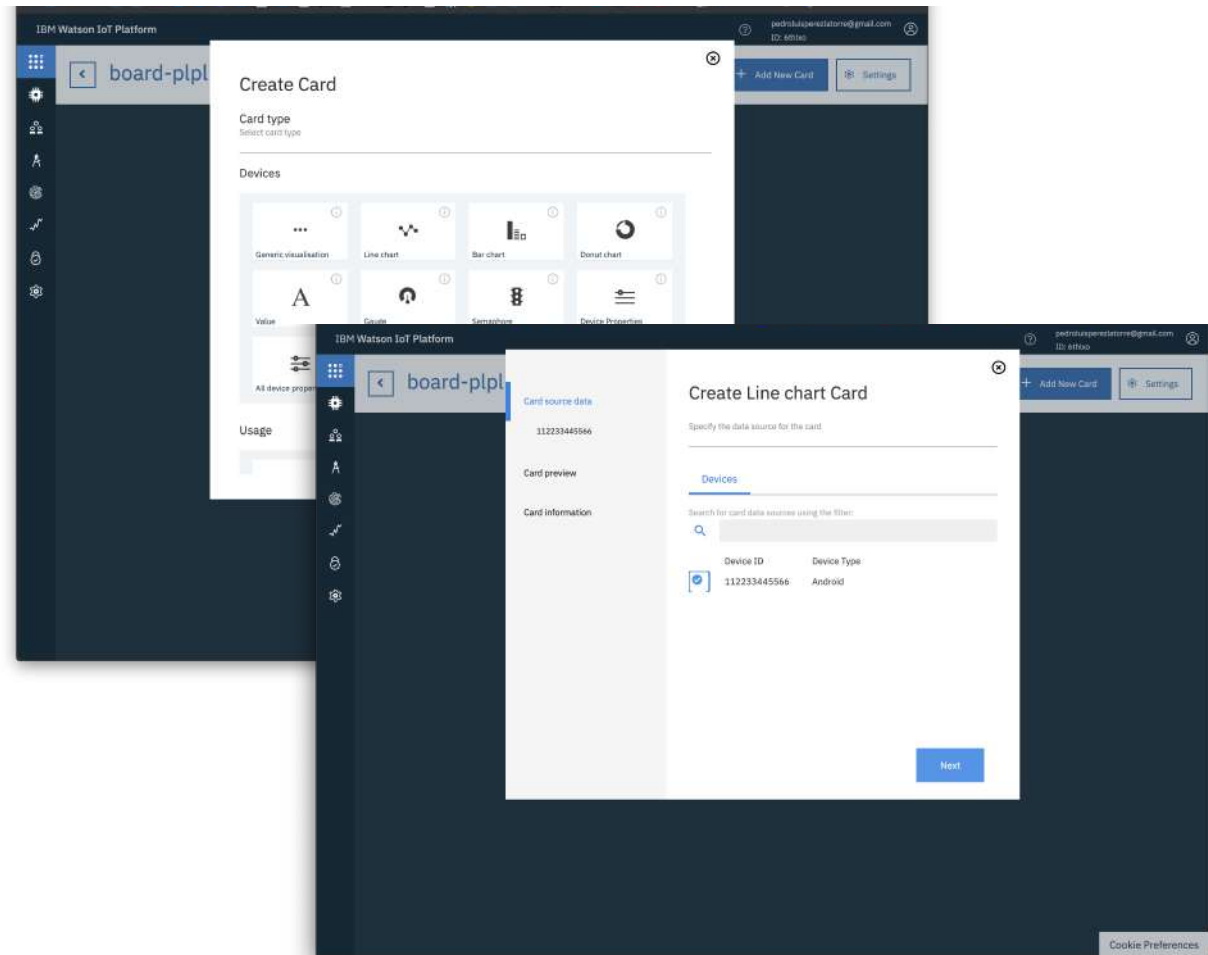
5. Volvemos a la pantalla de boards y comprobamos que vemos el recién creado, que figurará vacío, sin cards.
6. Hacemos click sobre este y en la nueva pantalla hacemos click sobre [+ Add New Card](#)



4. Práctica básica IoT

4.6. Monitoriza lecturas con un dashboard

- Se abrirá un menú en el que podemos elegir el tipo de visualización que deseamos incluir en el dashboard. Elegimos [Line chart](#)
- A continuación, deberíamos visualizar nuestro dispositivo, por lo que lo seleccionamos y le damos a [Next](#)



4. Práctica básica IoT

4.6. Monitoriza lecturas con un dashboard

9. A continuación, debemos hacer click sobre [Connect new data set](#)

10. En el menú que se despliega debemos completar:

Event – [accel](#)

Property – [AccelerometerX@StarterSensor](#)

Name – [accelx](#)

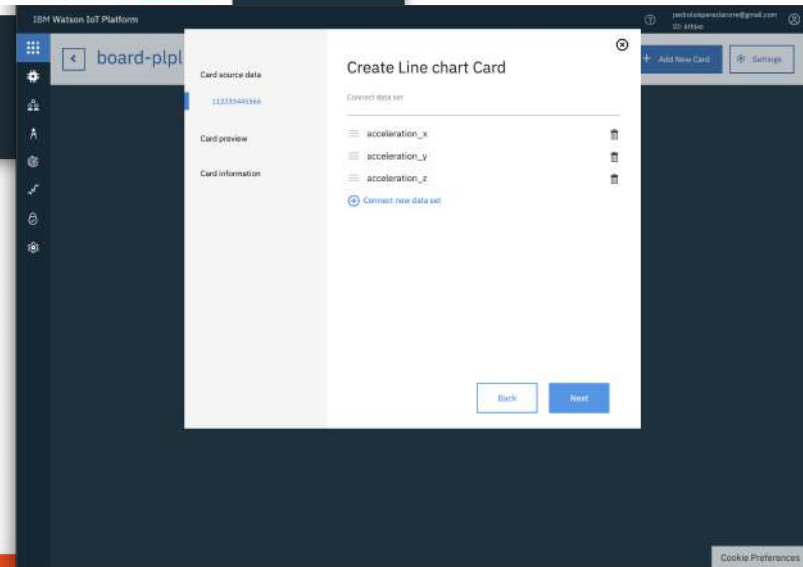
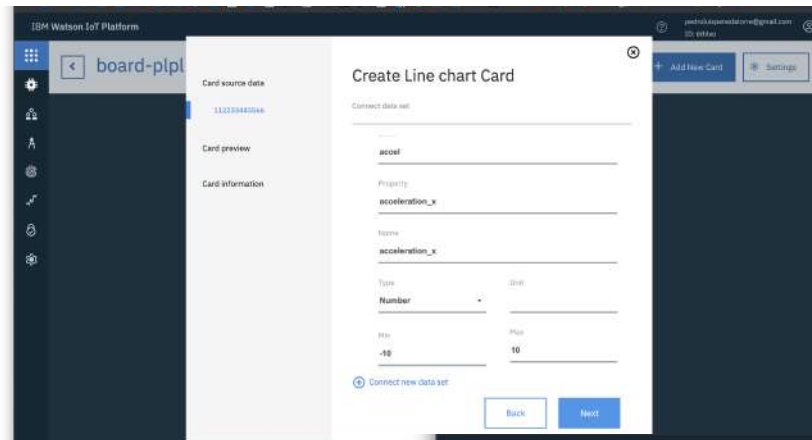
Type – [Float](#)

Min – [-10](#)

Max – [10](#)

11. Antes de pulsar Next, hacemos scroll hacia abajo y hacemos click sobre [+ Connect new data set](#)

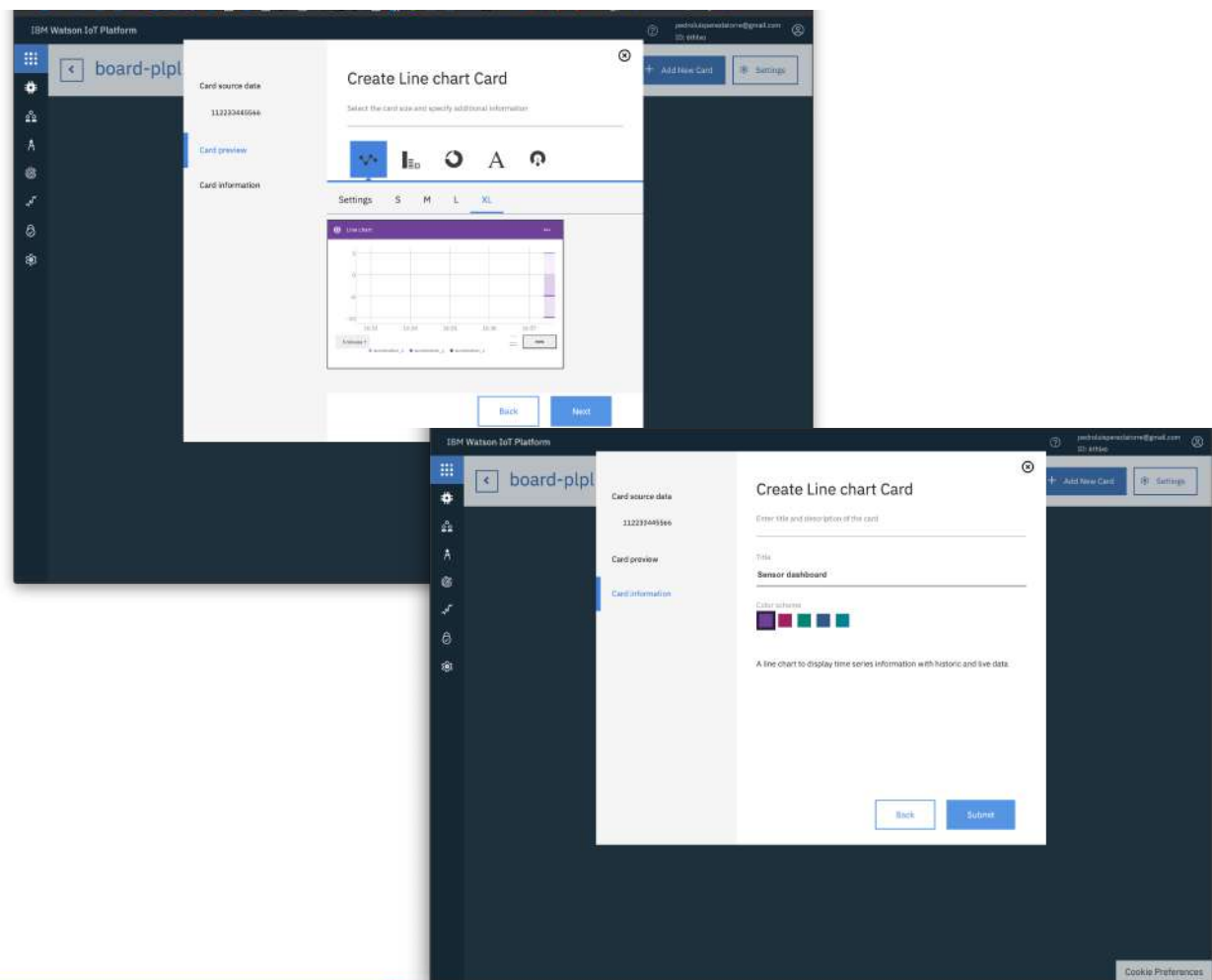
12. Repetimos el proceso para el acelerómetro del “eje y” y el del “eje z” y pulsamos sobre [Next](#)



4. Práctica básica IoT

4.6. Monitoriza lecturas con un dashboard

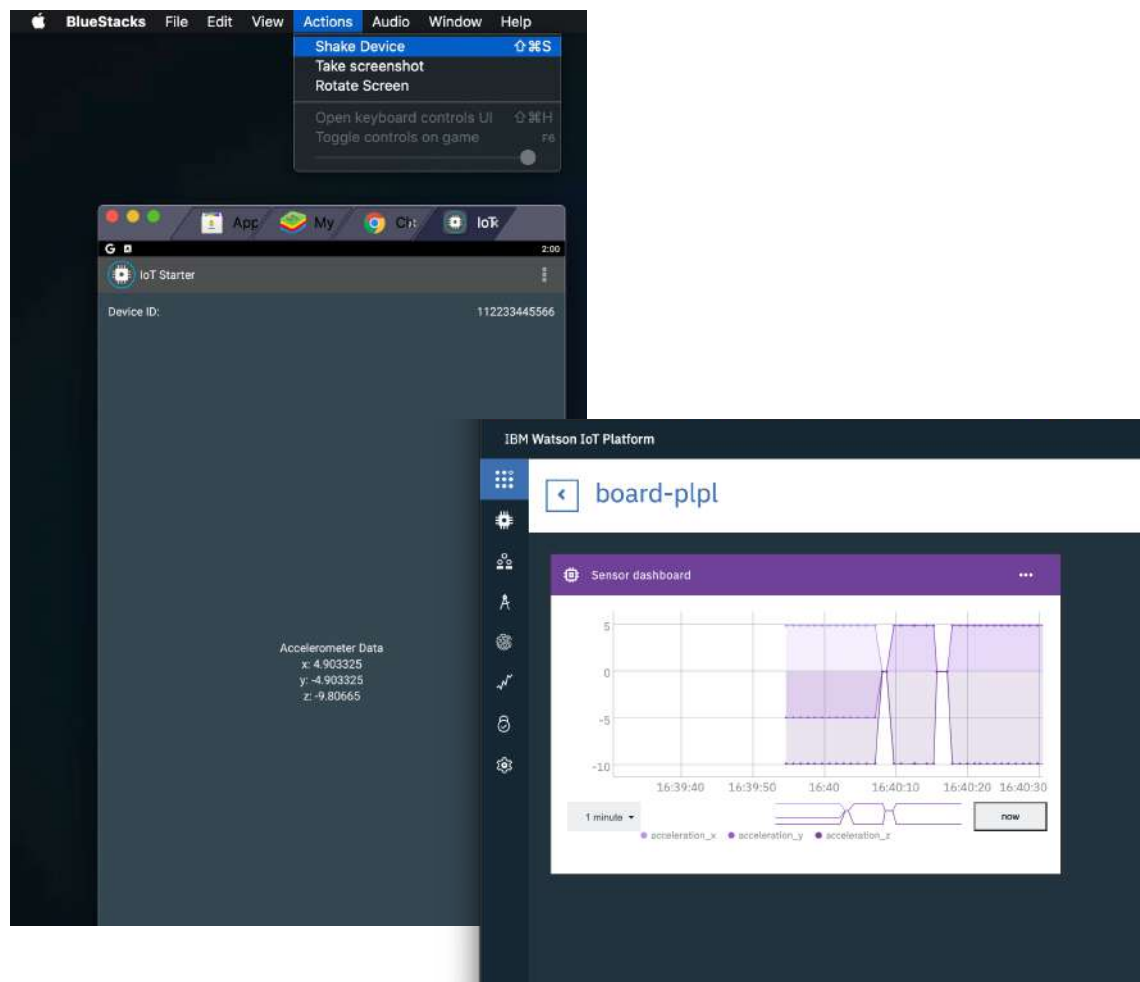
13. A continuación seleccionamos el tipo de gráfico, elegimos **línea** y tamaño **XL**. Pulsamos en **Next**.
14. Elegimos el esquema de colores y pulsamos sobre **Submit**



4. Práctica básica IoT

4.6. Monitoriza lecturas con un dashboard

15. Por último, si volvemos a agitar el dispositivo (ya sea en simulador o en emulación) seremos capaces de [monitorizar en tiempo real](#) las lecturas del sensor de movimiento del dispositivo.



5. Práctica intermedia IoT

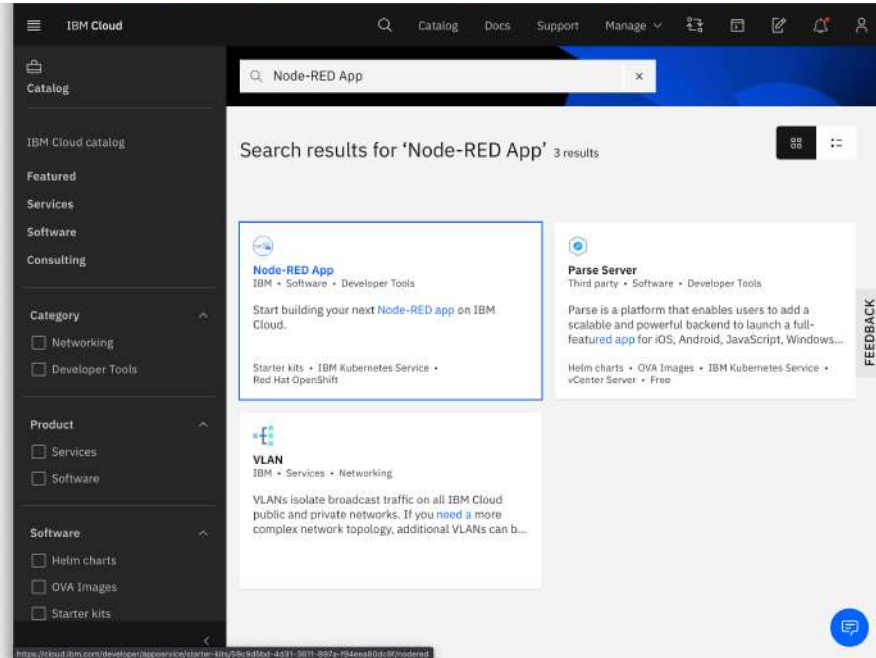
Un paso más en hacia las aplicaciones IoT

- 5.1. Despliegue App Node-RED
- 5.2. Conectar con el servicio IoT Platform
- 5.3. Habilitar Continuous Delivery
- 5.4. Accedemos a Node-RED
- 5.5. Instalando extensión IBM en Node-RED
- 5.6. Procesado de los mensajes en Node-RED
- 5.7. Persistencia de datos en Cloudant

5.1. Despliegue app Node-RED

Seguimos los siguientes pasos para crear una aplicación Node-RED:

1. Log in en [IBM Cloud](#).
2. Hacemos click en [Catalog](#) y buscamos [Node-RED App](#)
3. Hacemos click sobre la casilla de [Node-RED App](#)



5. Práctica intermedia IoT

5.1. Despliegue app Node-RED

4. Pulsamos sobre la pestaña [Create](#) y completamos los siguientes campos:

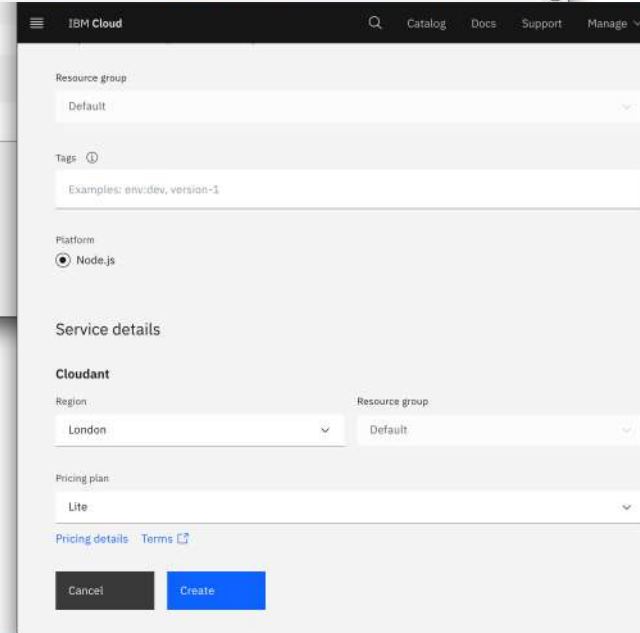
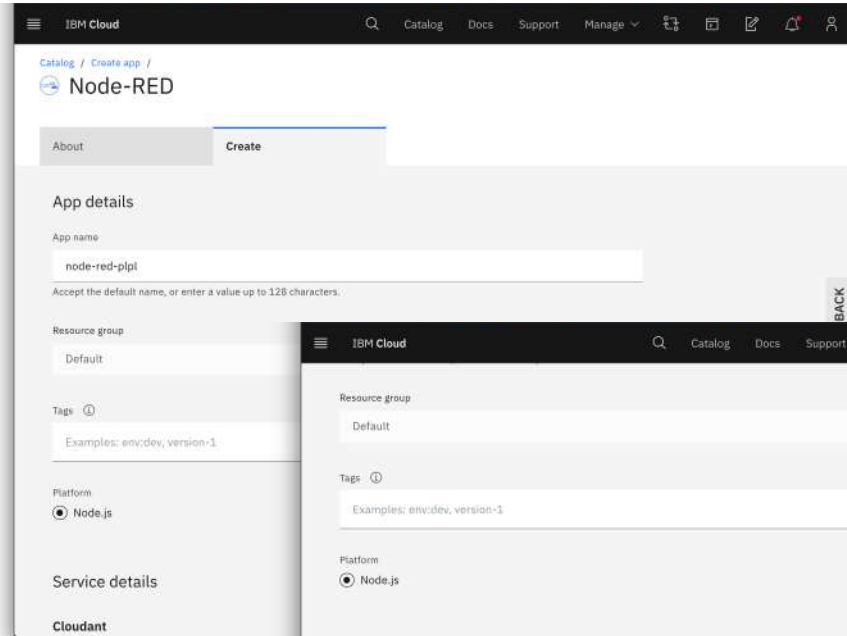
App Details

- **App name** – Se precisa un nombre único, por lo que una sugerencia es el uso de iniciales y el nombre del servicio
 - [node-red-plpl](#) – en el caso del tutorial
- **Resource group** – Se deja [Default](#)
- **Tags** – Se deja [vacío](#)

Service Details

- **Region** – Muy importante! Se debe seleccionar la misma ubicación para todos los servicios
 - [London](#) – en el caso del tutorial
- **Resource group** – Se deja [Default](#)
- **Pricing plan** – Se deja [Lite](#)

5. Una vez completados los campos pulsamos en [Create](#) y esperamos

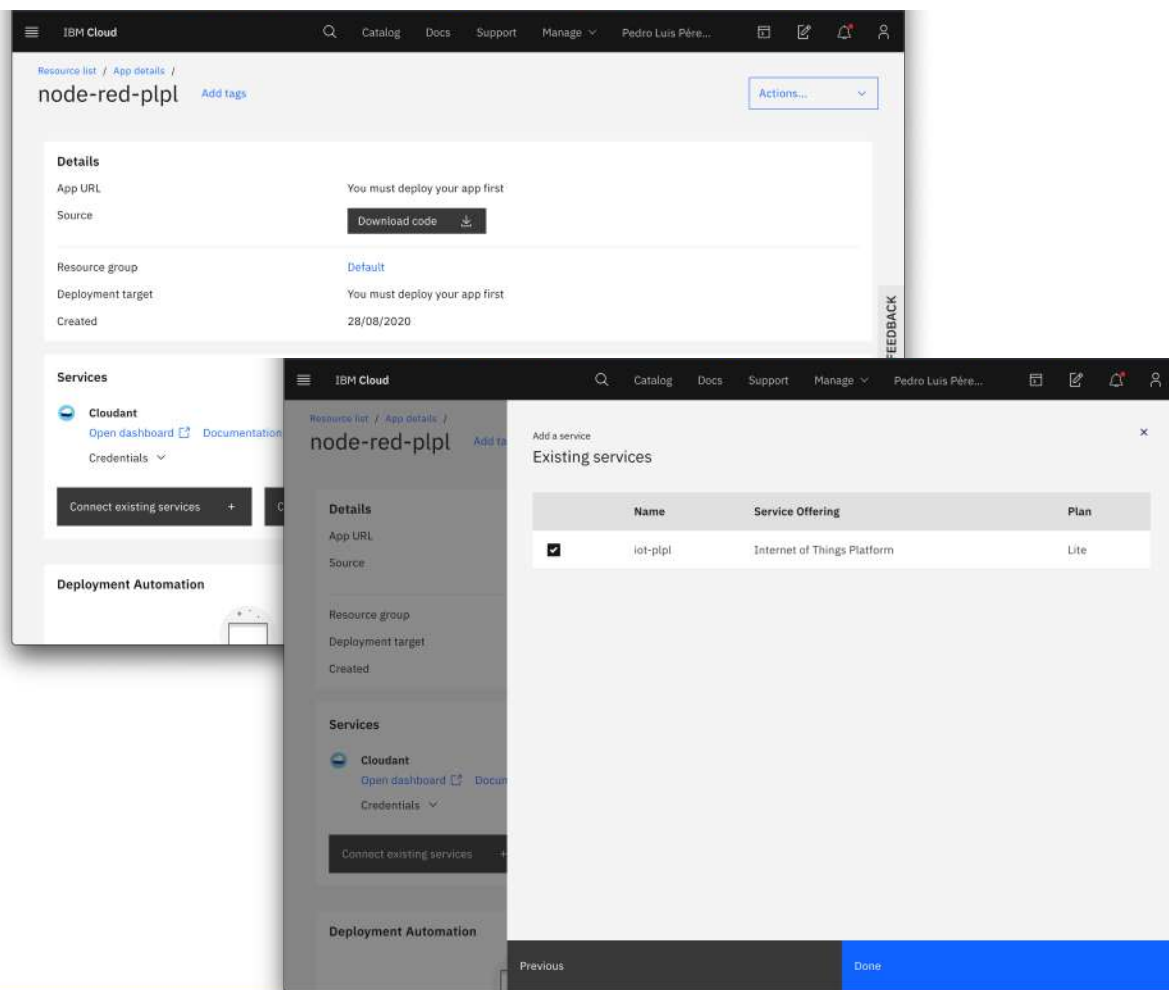


5. Práctica intermedia IoT

5.2. Conectar con el servicio IoT Platform

Una vez desplegado el servicio anterior seguimos los siguientes pasos:

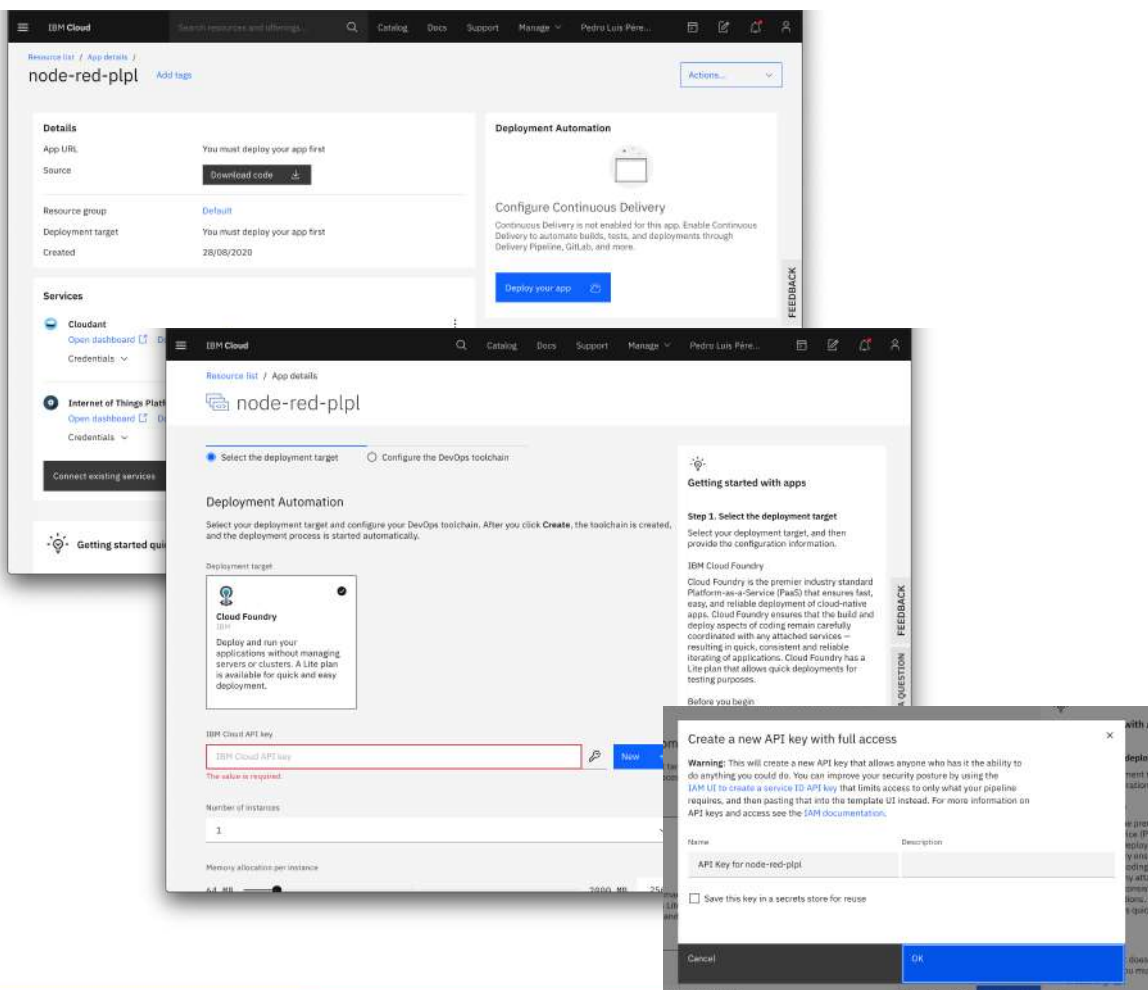
1. Pulsamos sobre [Connect existing Service +](#).
2. Seleccionamos el servicio previamente creado de [Internet of Things Platform](#) y pulsamos sobre [Done](#)



5. Práctica intermedia IoT

5.3. Habilitar Continuous Delivery

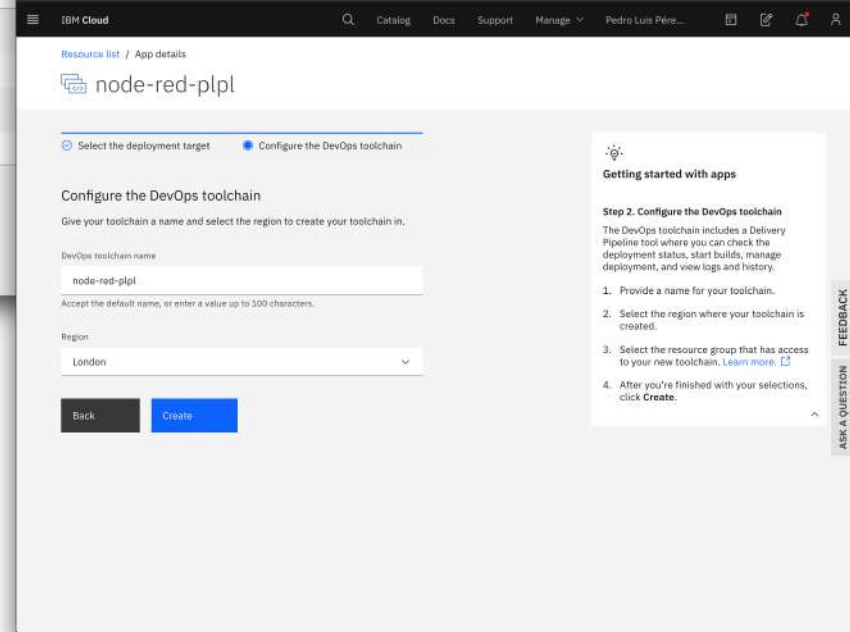
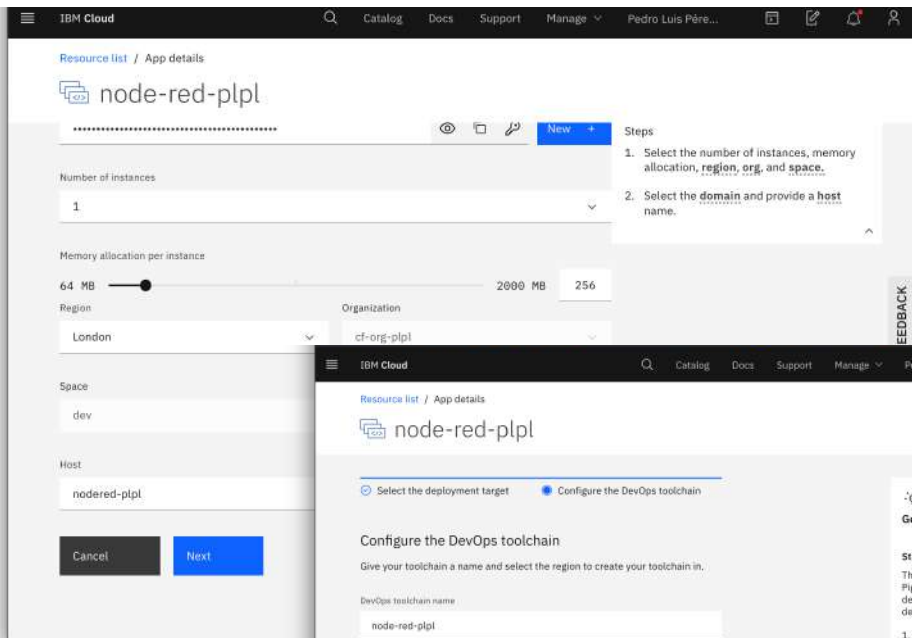
1. Primero, pulsamos sobre **Deploy your app**.
2. A continuación, crear una **IBM Cloud Api Key** pulsando sobre **New** y posteriormente en **Ok**



5. Práctica intermedia IoT

5.3. Habilitar Continuous Delivery

3. Haciendo scroll hacia abajo, seleccionamos la **Region London**
4. Los campos **Organization, Space** y **Host** deberían completarse automáticamente.
5. Una vez todo completado, pulsamos en **Next**
6. En la nueva ventana, dejamos el **DevOps toolchain name** el que nos viene por defecto
7. Y de nuevo en la **Region**, seleccionamos **London**
8. Una vez todo completado, pulsamos en **Create**



5. Práctica intermedia IoT

5.3. Habilitar Continuous Delivery

- Una vez ha comenzado el **despliegue**, este **tomará unos minutos** para completarse
- Las etapas que se suceden son:
 - No stages detected
 - In progress
 - Success
- Durante el despliegue, podemos ver el log pulsando sobre **Status**

The image displays three overlapping screenshots of the IBM Cloud interface, illustrating the deployment process for an application named 'node-red-plpl'.

The top screenshot shows the 'App details' page for 'node-red-plpl'. It includes fields for 'App URL', 'Source' (a GitHub repository), 'Resource group' (Default), 'Deployment target', and 'Created' date (26/08/2020). Below this, the 'Services' section lists 'Cloudant' and 'Internet of Things Platform'. The 'Deployment Automation' section shows the deployment is in progress, with 'No stages detected'.

The middle screenshot shows the 'Deployment Automation' page for 'node-red-plpl'. It displays the 'Name', 'Location' (London), and 'Tool integrations'. The 'Delivery Pipelines' section shows the pipeline is 'In progress' with 'Last input' from 'Last commit by IBM Cloud'.

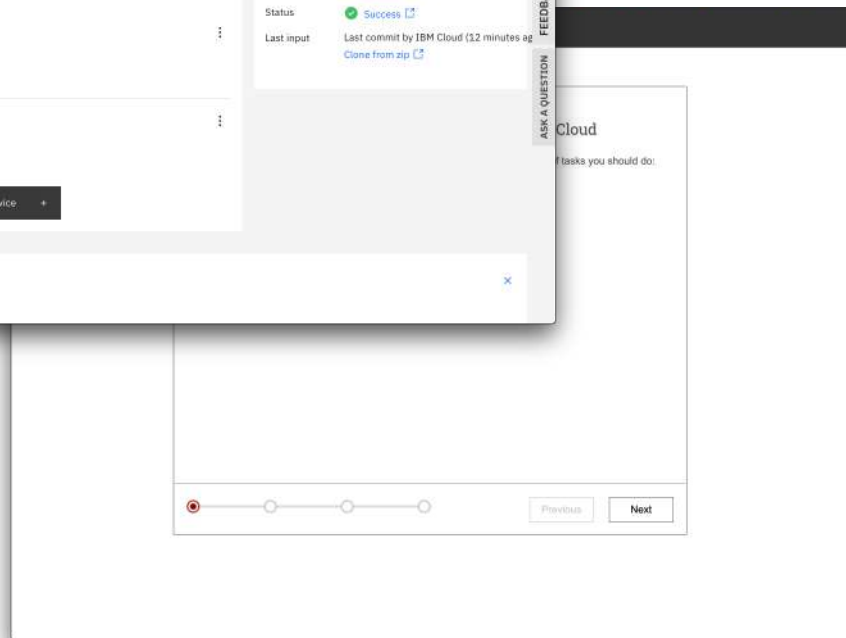
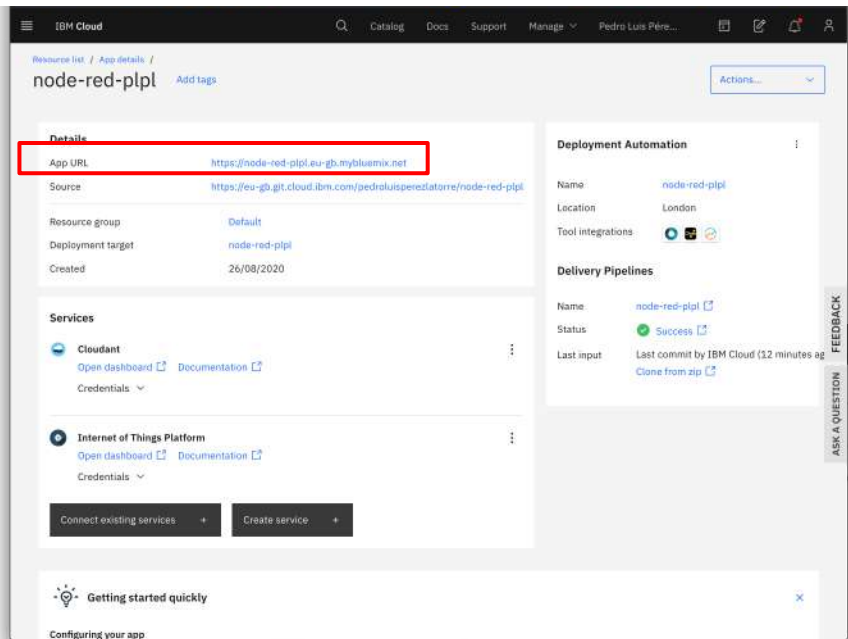
The bottom screenshot shows the 'node-red-plpl | Delivery Pipeline' page. It displays two stages: 'BUILD' (Stage Passed) and 'DEPLOY' (Stage Running). The 'BUILD' stage shows 'Last input' from 'Last commit by IBM Cloud' and 'Jobs' including 'Build' (Passed 4m ago). The 'DEPLOY' stage shows 'Last input' from 'Stage: BUILD / Job: Build' and 'Jobs' including 'Rolling Deploy' (Running).

5. Práctica intermedia IoT

5.4. Accedemos a Node-RED

Una vez finalizado el despliegue de Node-RED ya estaría listo para utilizar:

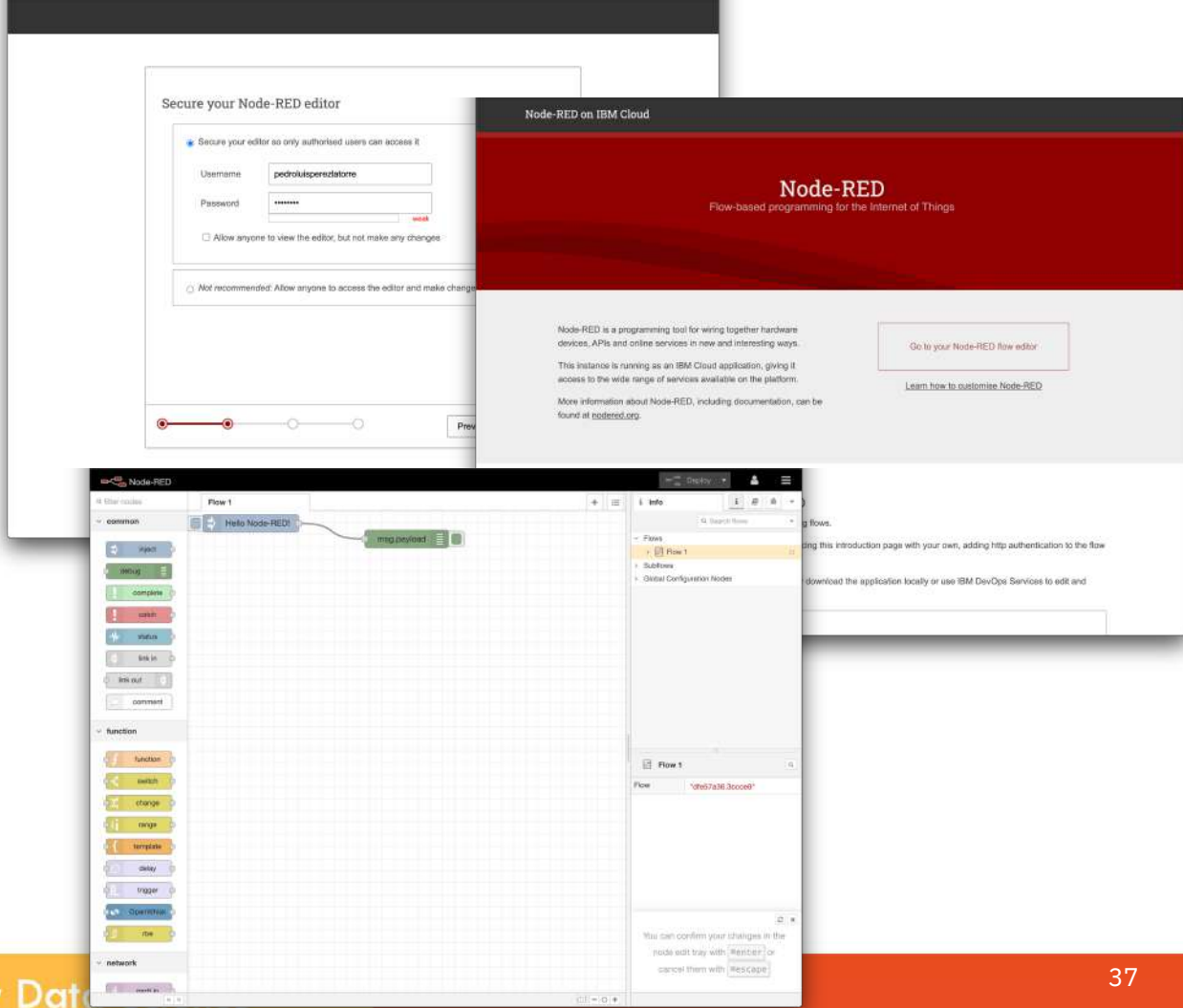
1. Hacemos click sobre [App URL](#).
2. Sobre la ventana emergente pulsamos en [Next](#)



5. Práctica intermedia IoT


5.4. Accedemos a Node-RED

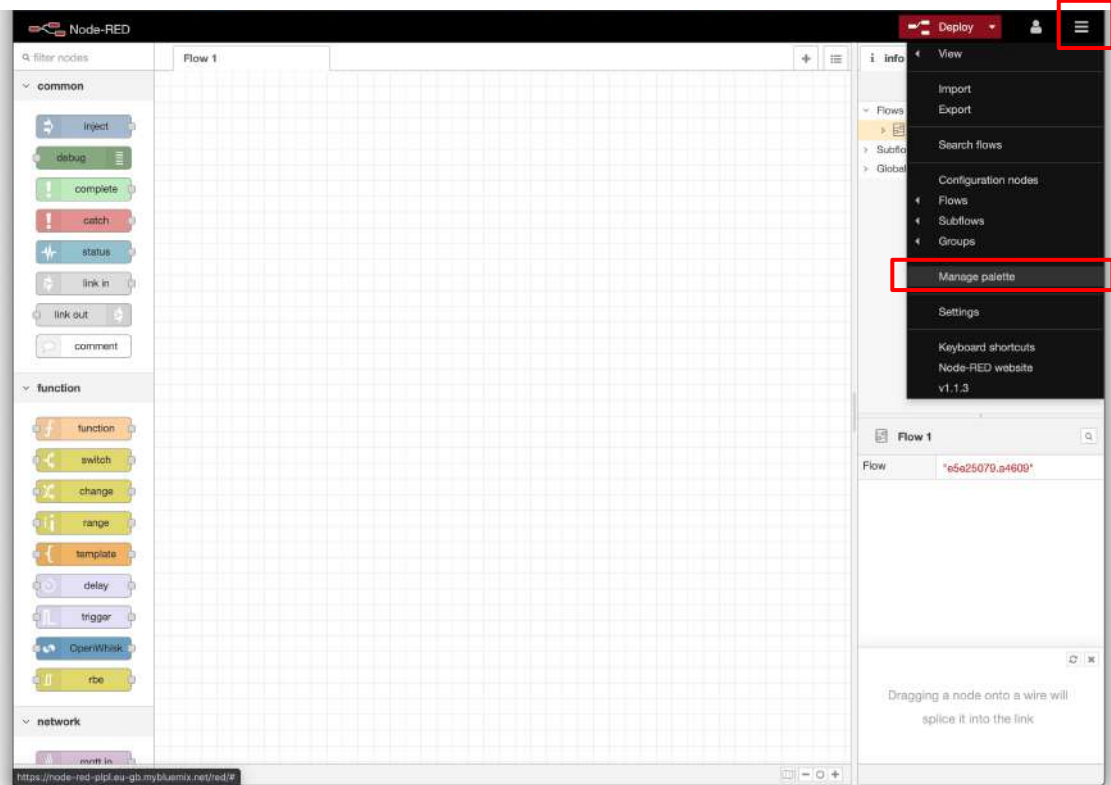
3. Aseguramos nuestro editor introduciendo **Username** y **Password**.
4. Pulsamos sobre **Next** en 2 pasos y por último en **Finish**
5. Pulsamos sobre **Go to your Node-RED flow editor**
6. Hacemos **log in** y ya estaríamos en el editor.



5. Práctica intermedia IoT

5.5. Instalando extensión IBM en Node-RED

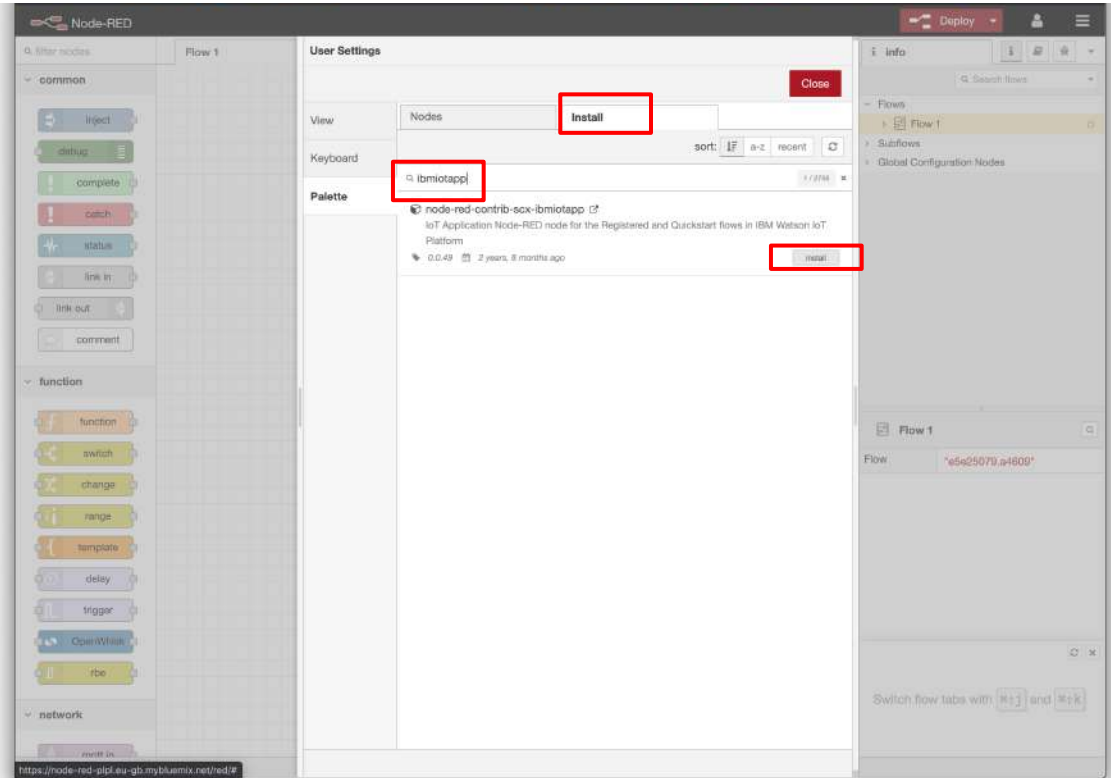
1. Nada más abrir Node-RED encontraremos 2 nodos que componen una instrucción “Hello world”. Podemos probar a ejecutarla o simplemente los borramos.
2. Antes de empezar a trabajar, vamos a descargar unos componentes necesarios. Pulsamos sobre el icono superior derecho  , y en el desplegable pulsamos sobre **Manage palletes**



5. Práctica intermedia IoT

5.5. Instalando extensión IBM en Node-RED

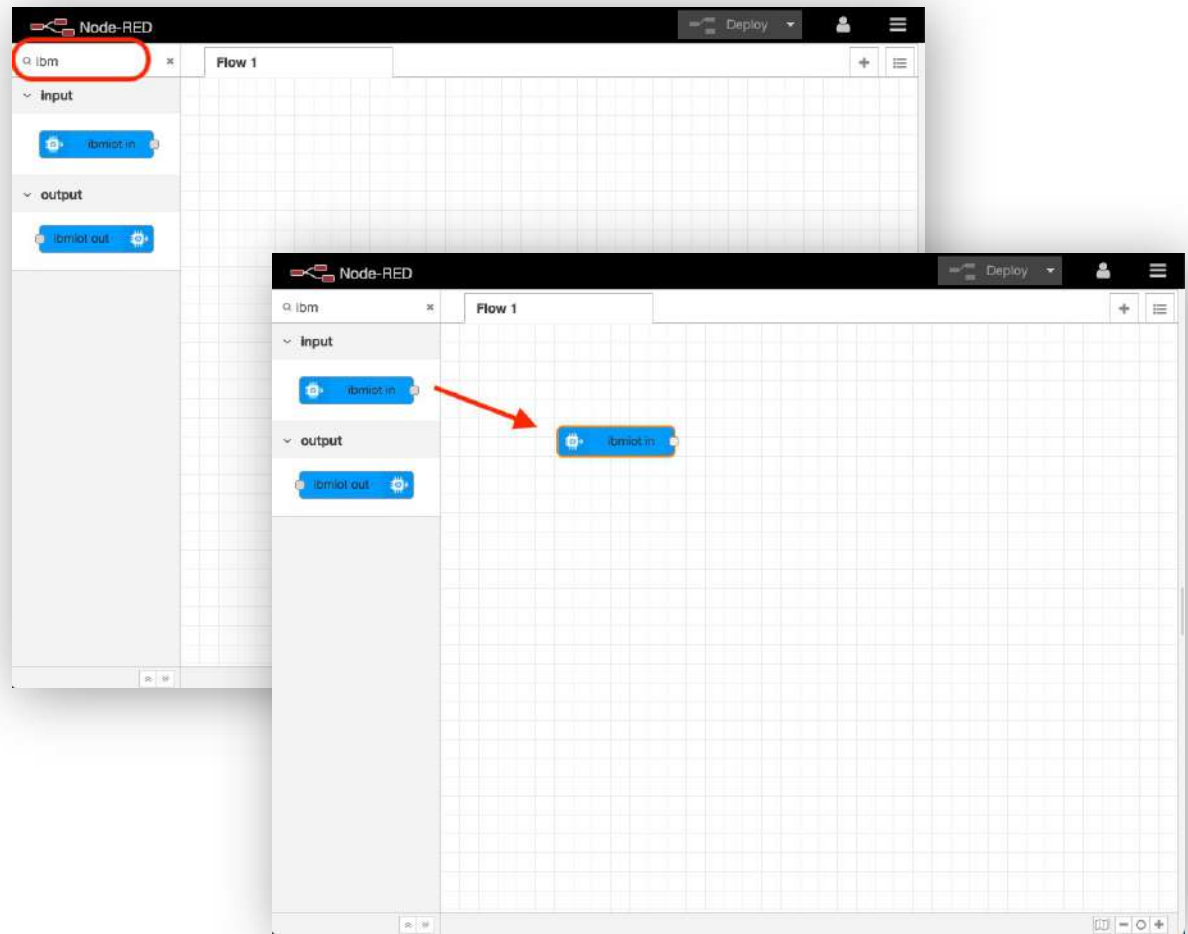
3. En el menú que se despliega, nos posicionamos en la pestaña **Install** y escribimos en el campo de búsqueda **ibmiotapp**
4. Pulsamos sobre **Install**



5. Práctica intermedia IoT

5.6. Procesado de los mensajes en Node-RED

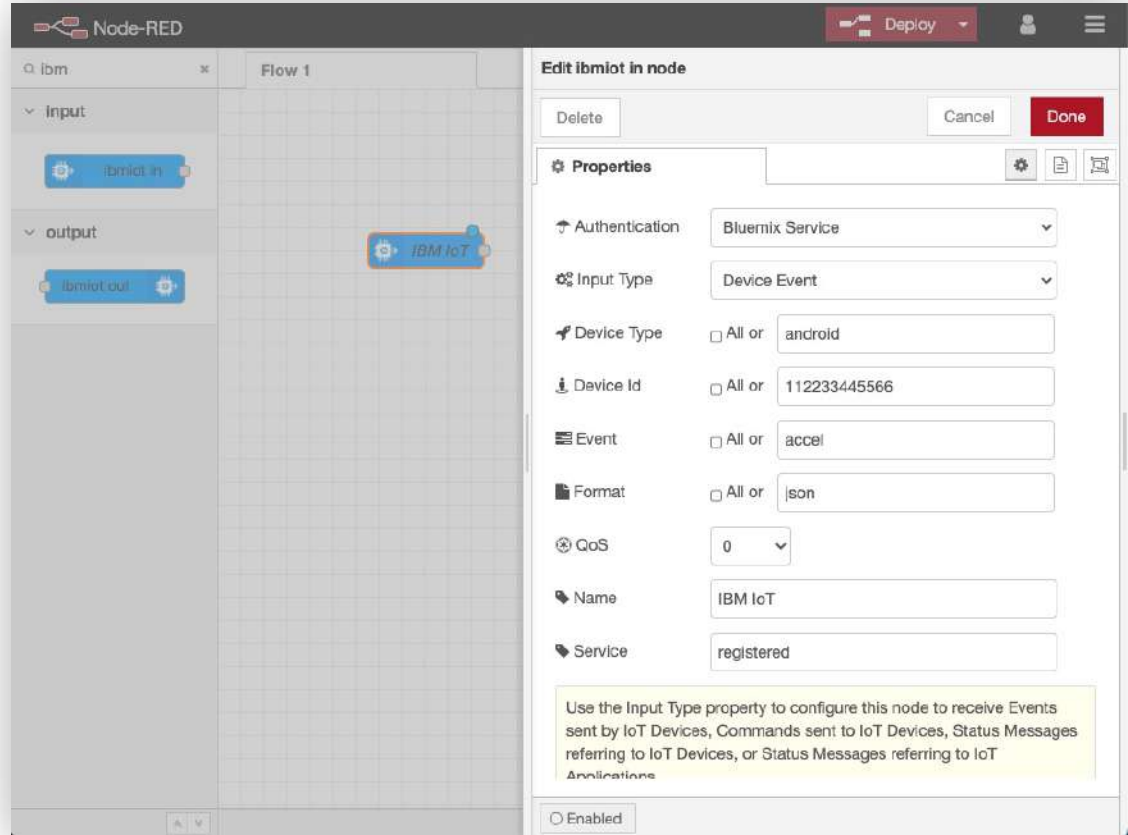
1. Ya con la extensión de IBM instalada, buscamos en la barra de filtrado de nodos y buscamos **ibmiot**.
2. Arrastramos el nodo **ibmiot in** al espacio de trabajo.



5. Práctica intermedia IoT

5.6. Procesado de los mensajes en Node-RED

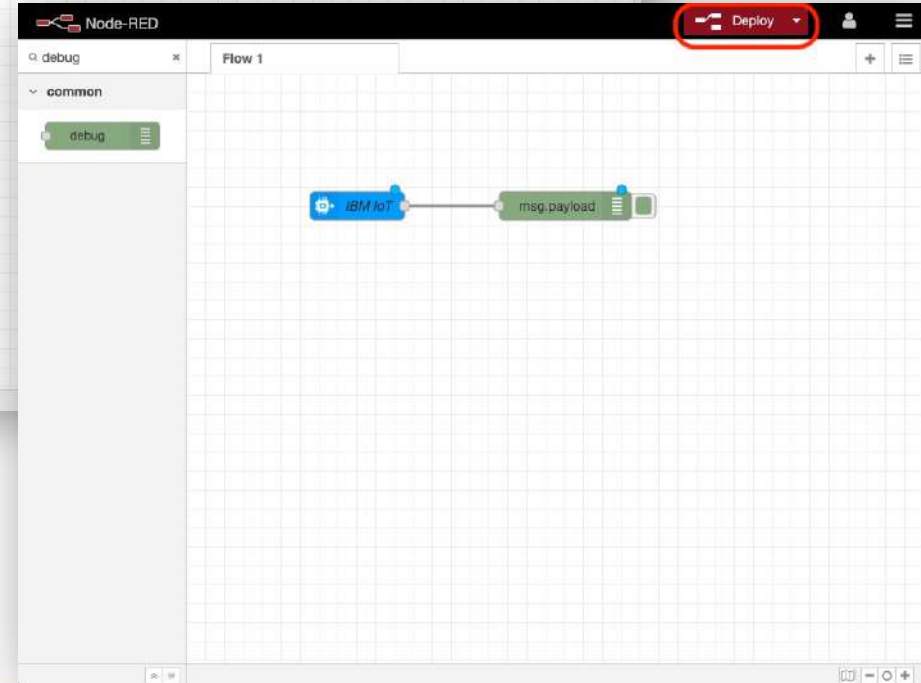
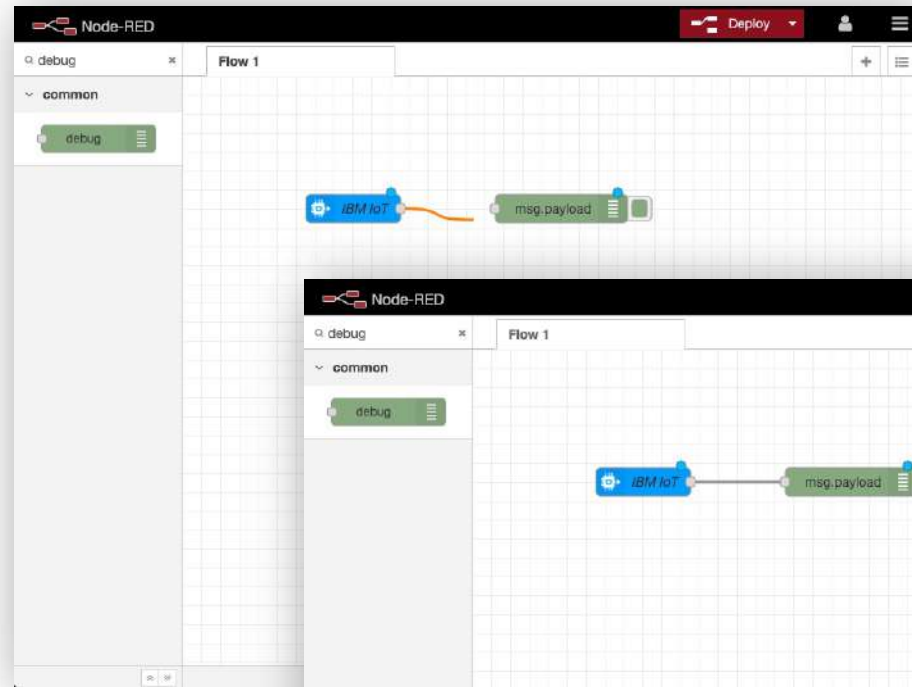
- Una vez colocado el nodo en el espacio de trabajo, hacemos doble click para abrir las propiedades.
- Configuramos el nodo según los siguientes valores:
 - Authentication – Bluemix Service
 - Input Type – Device Event
 - Device Type – android
 - Device Id – 112233445566
 - Event – accel
 - Format – json



5. Práctica intermedia IoT

5.6. Procesado de los mensajes en Node-RED

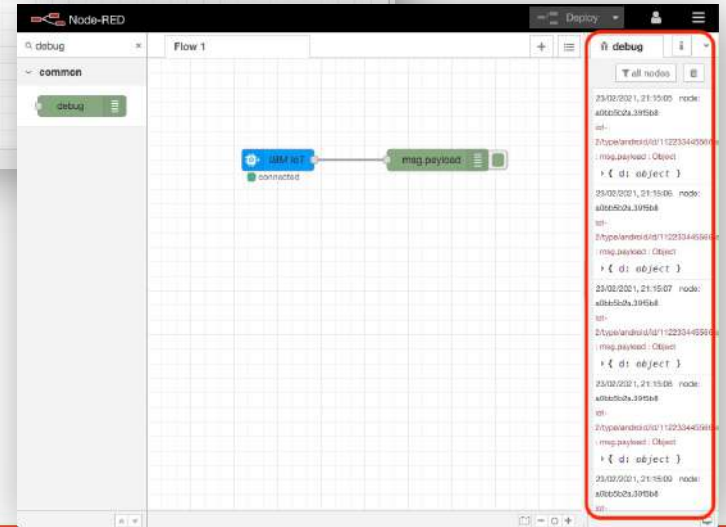
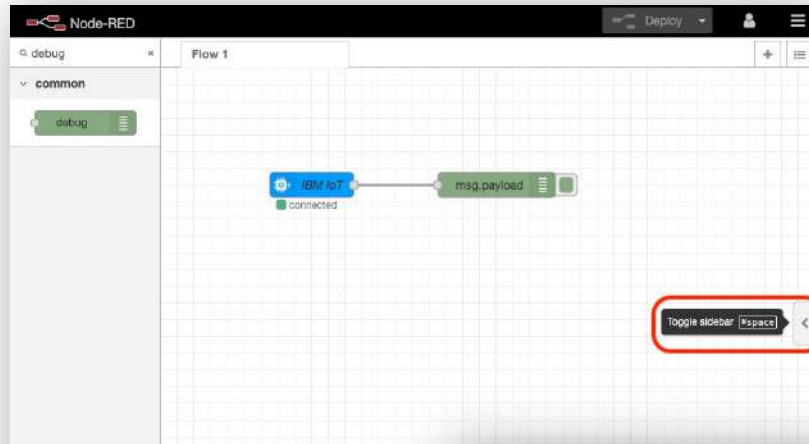
5. A continuación en el buscador buscamos el nodo **debug** y lo arrastramos hasta el espacio de trabajo.
6. Pinchando sobre el conector del nodo **IBM IoT** arrastramos hasta el de **msg.payload**.
7. Pulsamos en Deploy



5. Práctica intermedia IoT

5.6. Procesado de los mensajes en Node-RED

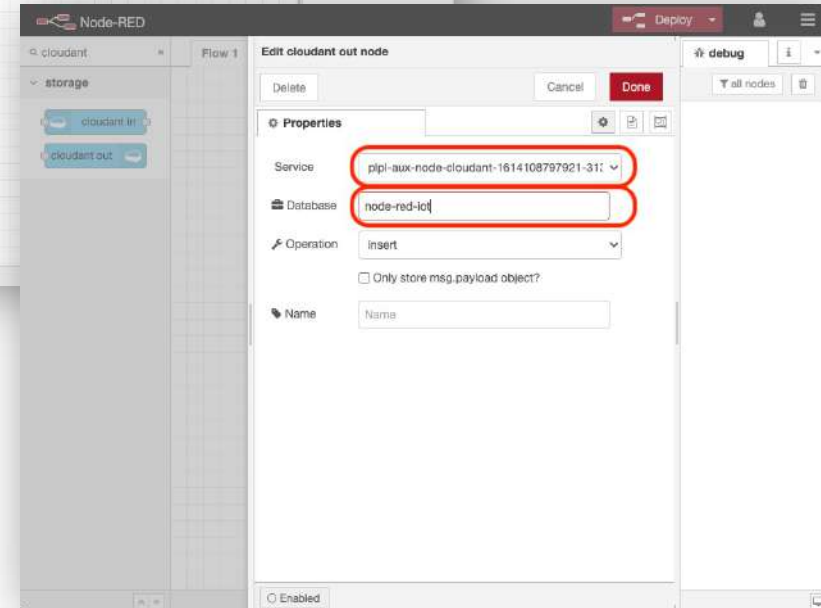
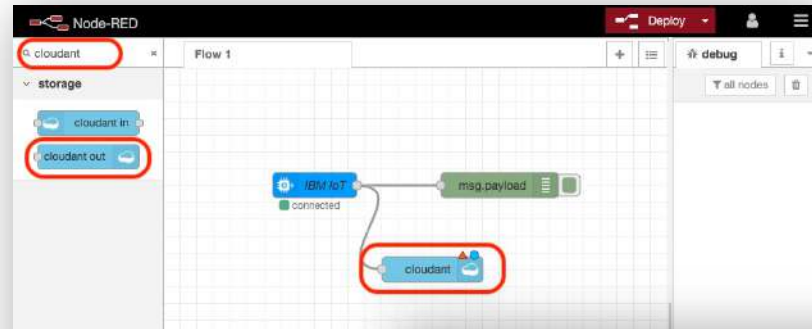
8. Si no está visible la barra lateral de **debug**, la desplegamos con la flecha.
9. Si está la app generando datos, seremos capaces de verlos en la **barra de debug**.



5. Práctica intermedia IoT

5.7. Persistencia de datos en Cloudant

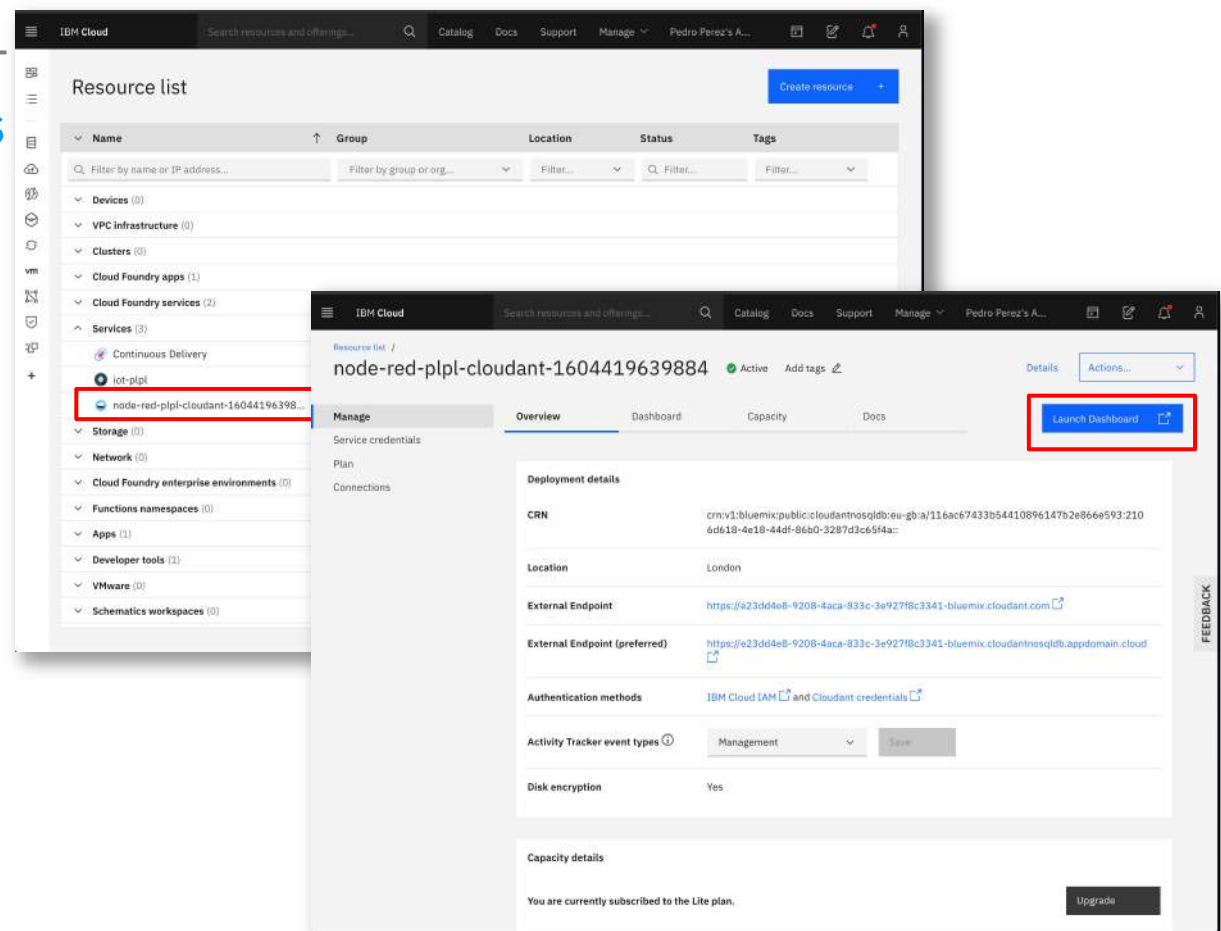
1. Ahora que somos capaces de recibir y visualizar los datos, procederíamos a persistirlos en una base de datos en la nube.
2. En el buscador escribimos **Cloudant** y arrastramos el nodo de salida y lo conectamos con **IBM IoT**.
3. Hacemos doble click sobre el nodo de Cloudant para abrir la ventana de opciones. Tan solo hay que **seleccionar el servicio de Cloudant ya desplegado** en IBM Cloud enlazado con las instancias de Node e IoT Platform, y le damos un **nombre a la Database**.



5. Práctica intermedia IoT

5.7. Persistencia de datos en Cloudant

- De vuelta a la consola de IBM Cloud, seleccionaríamos el servicio de **Cloudant**
- Pulsamos sobre **Launch Dashboard** para abrir la consola



5. Práctica intermedia IoT

5.7. Persistencia de datos en Cloudant

- En la consola de Cloudant deberíamos ver la base de datos generada. Si pulsamos sobre esta se abrirá una vista de tabla con los mensajes recibidos.
- En esta nueva vista, podemos filtrar y visualizar los mensajes
- Si pulsamos sobre cualquiera de los mensajes, podemos ver el formato Json con la información del dispositivo, de la plataforma,

The first screenshot shows the 'Databases' page in the Cloudant console. It lists two databases: 'nodered' (26.3 KB, 4 docs) and 'nodered-data' (1.4 MB, 2827 docs). The 'nodered-data' database is highlighted with a red box.

The second screenshot shows the 'nodered-data' database view in table format. It displays a list of documents with columns for 'id', 'key', and 'value'. The first document has an ID of '003fecf00c72e8b8eabf9136ef3f878f' and a key of '003fecf00c72e8b8eabf9136ef3f878f'.

The third screenshot shows the details of the selected document. It displays the JSON payload, which includes fields like 'id', 'acceleration', 'roll', 'pitch', 'yaw', 'longitude', 'latitude', 'heading', 'speed', 'crisp_id', 'timestamp', 'deviceId', 'deviceType', 'eventType', and 'format'.

6. Práctica avanzada IoT

Integración de aplicaciones IoT con herramientas de analítica

6.1. Modificación del flujo Node-RED

6.2. Configuración Watson Studio

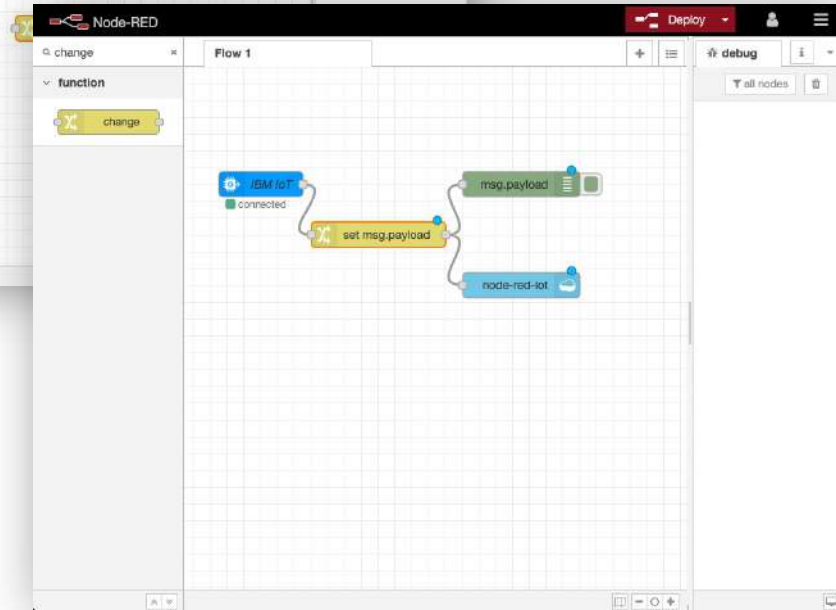
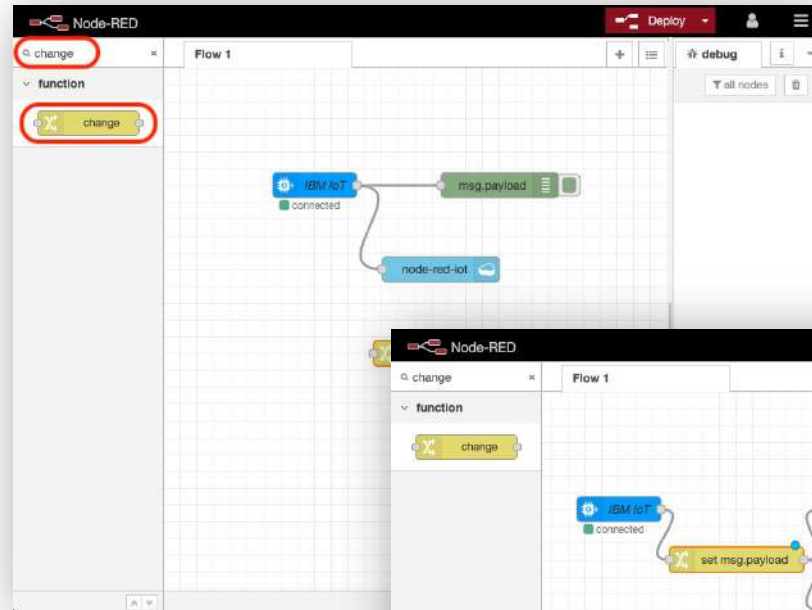
6.3. Conexión Cloudfoundry con Notebook

6. Práctica avanzada IoT

6.1. Modificación flujo Node-RED

Seguimos los siguientes pasos para conectar la aplicación Node-RED con un entorno de modelado analítico:

1. Volvemos al entorno de **Node-Red**
2. En el buscador, escribimos **change** y lo arrastramos al tablero.
3. Borramos la conexión entre los nodos de **IBM IoT** y **Node-red-data** y entre **msg.payload**
4. Reconectamos los nodos de **IBM IoT** con **change** y este con **Node-red-data** y **msg.payload**



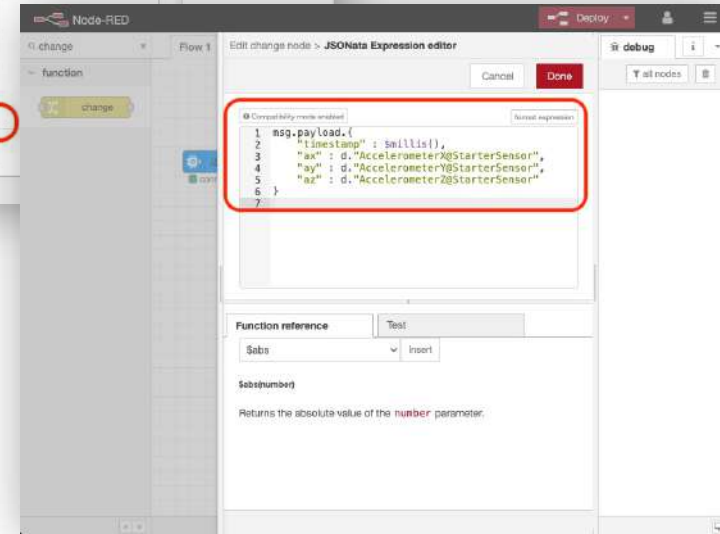
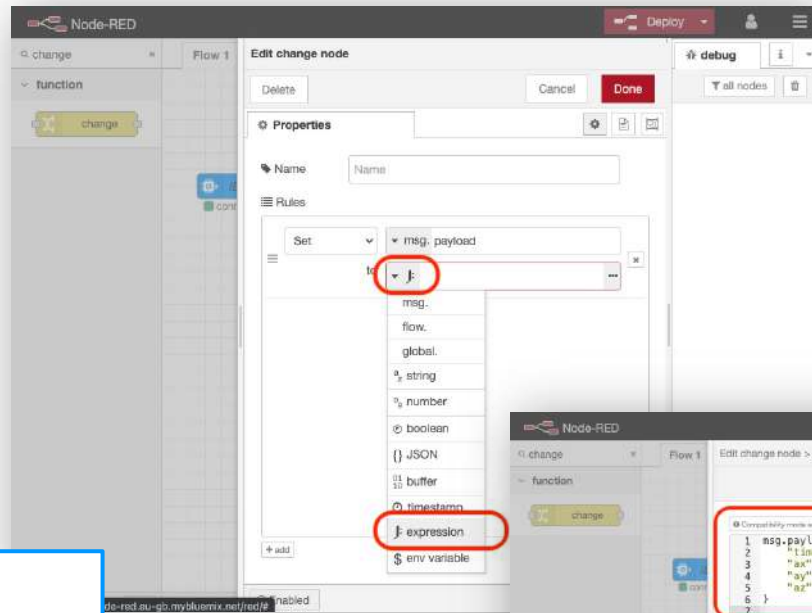
6. Práctica avanzada IoT

6.1. Modificación flujo Node-RED

- Hacemos doble click sobre el nodo de `set.msg.payload` para abrir la configuración
- Dejamos el primer campo como `payload` y cambiamos el de más abajo a `expression`
- Introducimos el siguiente código:

```
msg.payload.{  
  "timestamp" : $millis(),  
  "ax" : d."AccelerometerX@StarterSensor",  
  "ay" : d."AccelerometerY@StarterSensor",  
  "az" : d."AccelerometerZ@StarterSensor"  
}
```

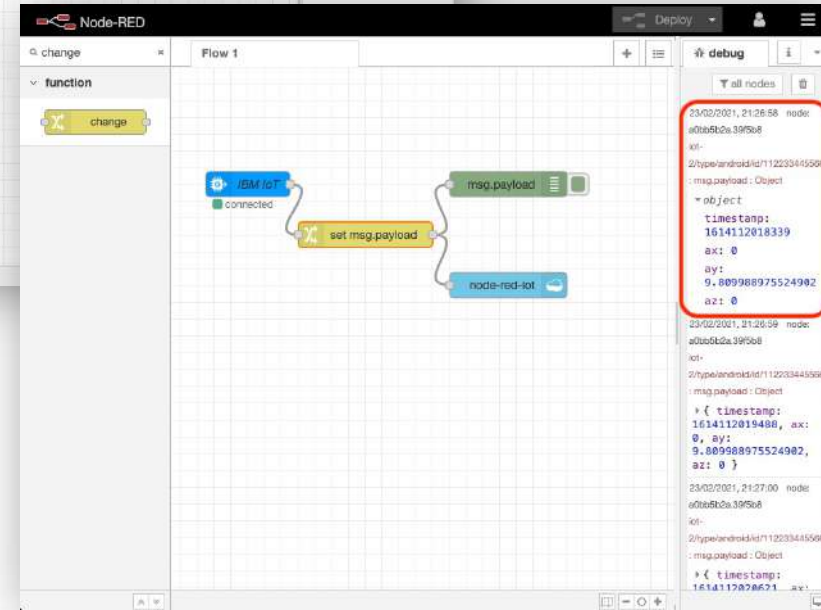
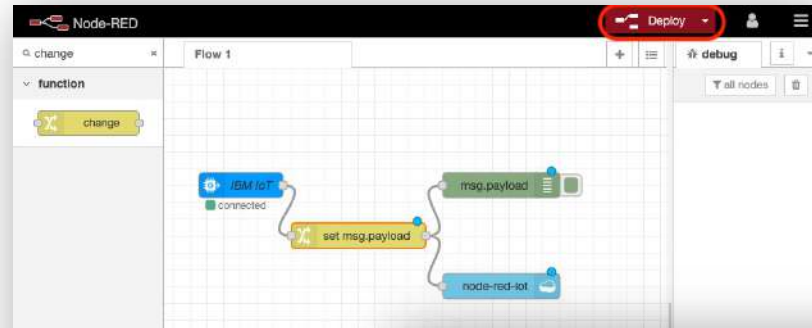
- Tras esto, pulsamos en `done`, y nuevamente en `done`



6. Práctica avanzada IoT

6.1. Modificación flujo Node-RED

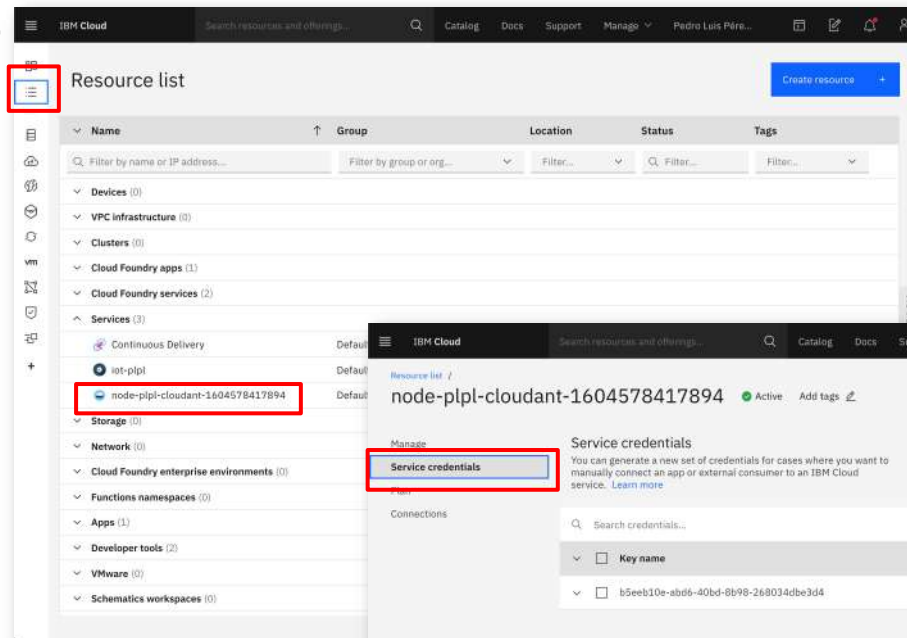
8. Pulsamos sobre **Deploy** y si el sensor está detenido, lo reactivamos.
9. Si hemos seguido los pasos correctamente, en **debug** deberán aparecer los nuevos mensajes ya filtrados para su posterior carga y análisis.



6. Práctica avanzada IoT

6.2. Configuración Watson Studio

1. Volvemos a la consola de [IBM Cloud](#)
2. Dentro de Servicios buscamos la instancia de [Cloudant](#) y pulsamos sobre su nombre.
3. En la nueva ventana pulsamos sobre [Service credentials](#)
4. Pulsamos sobre [New credential](#)



6. Práctica avanzada IoT

6.2. Configuración Watson Studio

5. Nombramos la credencial con un nombre único y pulsamos sobre **Add**
6. Una vez creada la credencial, desplegamos con el botón de la flecha para ver su contenido y lo copiamos para su posterior consulta.

The image displays two screenshots from the IBM Cloud console, illustrating the process of creating and viewing a service credential.

Top Screenshot: Create credential dialog

The "Create credential" dialog is shown over the "Service credentials" page. The "Name" field contains "ws-credential-plpl". The "Role" is set to "Manager". The "Add" button is highlighted in blue.

Bottom Screenshot: Service credentials list and details

The "Service credentials" page shows a list of credentials. The credential "ws-credential-plpl" is selected, and its details are displayed in a red-bordered box:

```
{
  "apikey": "c_vZ21nPFIv80QW8rLGX6bAholx_Omc4YRIK2716C",
  "host": "adf3c11b-e7aa-4b91-a966-467bf85379e1-bluemix.cloudantnosqldb.appdomain.cloud",
  "ian_apikey_description": "Auto-generated for key 88b55389-00f2-41f7-8ecb-22167b163c53",
  "ian_apikey_name": "ws-credential-plpl",
  "ian_role_crm": "crn:v1:bluemix:public:iam:::serviceRole:Manager",
  "ian_serviceid_crm": "crn:v1:bluemix:public:iam-identity::a/c93caad03634c31a12c6249384f89d6::serviceid:ServiceId-ecf7a1df-3685-4796-a6b8-8adf18977851",
  "password": "cc337ad623d1e703e66ae268fc43089ca781f99421b24d8ad66a3377a58de64c",
  "port": 443,
  "url": "https://adf3c11b-e7aa-4b91-a966-467bf85379e1-bluemix.cc337ad623d1e703e66ae268fc43089ca781f99421b24d8ad66a3377a58de64c@adf3c11b-e7aa-4b91-a966-467bf85379e1-bluemix.cloudantnosqldb.appdomain.cloud",
  "username": "adf3c11b-e7aa-4b91-a966-467bf85379e1-bluemix"
}
```

6. Práctica avanzada IoT

6.2. Configuración Watson Studio

- Volvemos al catálogo y buscamos **Watson Studio**, pulsamos sobre el.
- Procedemos a **crear el servicio**. Como puntos a tener en cuenta:
 - La región que debe ser la misma que el resto (**London** en el caso de la práctica)
 - Se recomienda poner un nombre único al servicio.

The image displays a sequence of screenshots from the IBM Cloud console illustrating the steps to create a Watson Studio service:

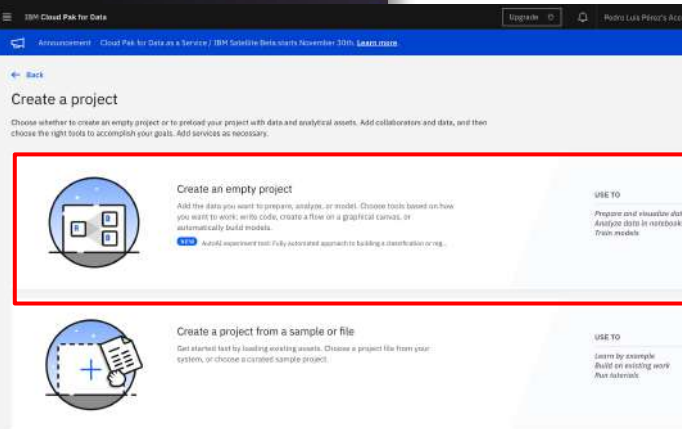
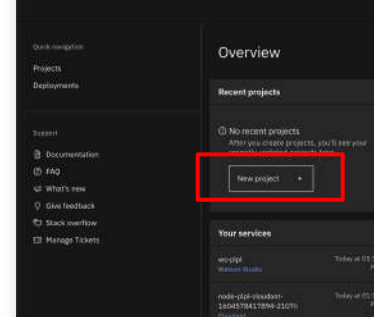
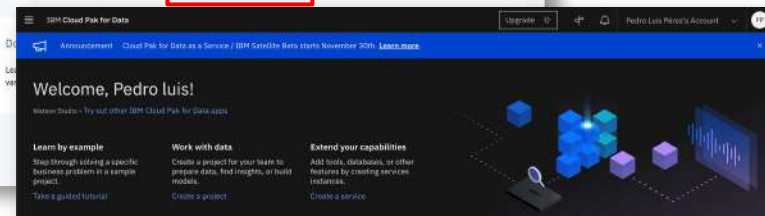
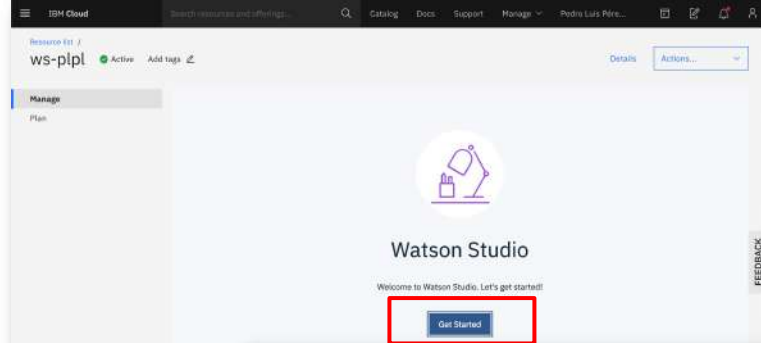
- Search Results:** The first screenshot shows the IBM Cloud Catalog search results for "Watson Studio". The "Watson Studio" entry is highlighted with a red box.
- Select Region:** The second screenshot shows the "Select a region" dropdown menu, with "London" selected and highlighted by a red box.
- Select Pricing Plan:** The third screenshot shows the "Select a pricing plan" section. The "Lite" plan is selected, and its details are visible, including features like "1 authorized user" and "30 capacity unit-hours monthly limit".
- Configure Resource:** The fourth screenshot shows the "Configure your resource" section. The "Service name" field is populated with "wstest" and highlighted by a red box.
- Create Service:** The final screenshot shows the "Create" button highlighted with a red box, ready to be clicked.

Additional details visible in the screenshots include the "Summary" section on the right, which lists the service name as "Watson Studio", the region as "London", and the plan as "Lite". The "Create" button is located at the bottom right of the configuration page.

6. Práctica avanzada IoT

6.2. Configuración Watson Studio

9. En la ventana siguiente pulsamos sobre [Get Started](#)
10. En la nueva ventana pulsamos sobre [New Project](#)
11. Entre las dos opciones, elegimos la de [Create an empty project](#)



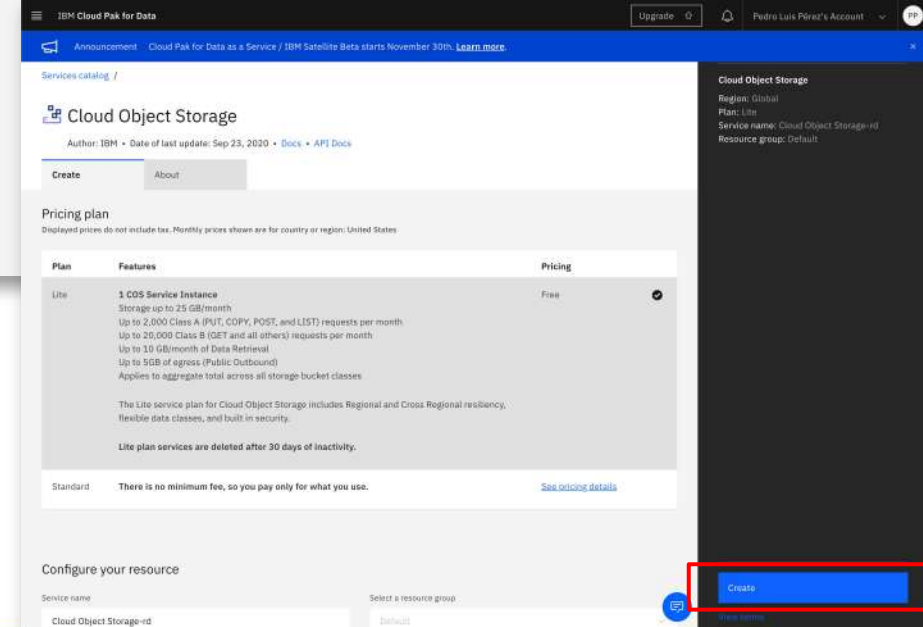
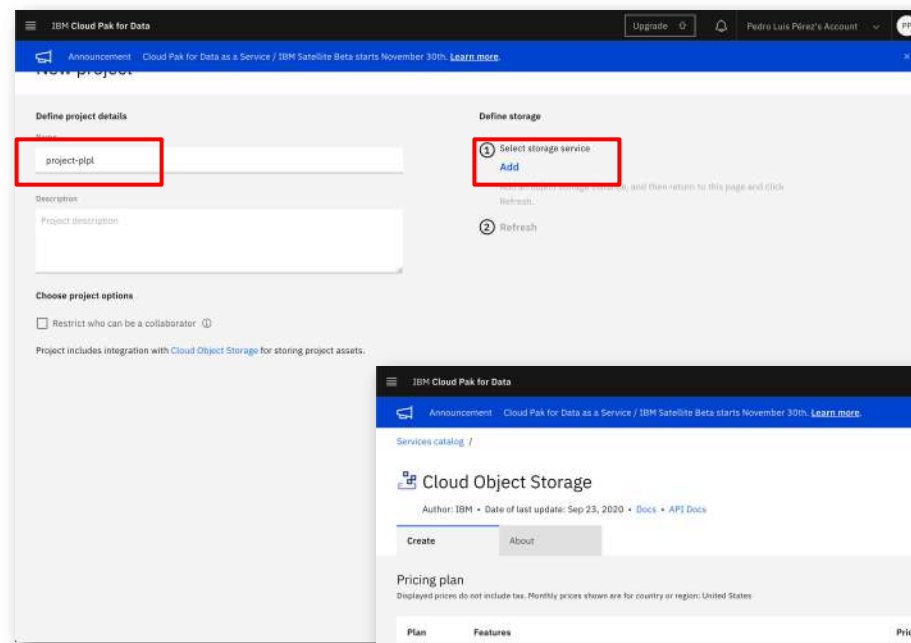
6. Práctica avanzada IoT

6.2. Configuración Watson Studio

12. Damos un nombre al proyecto

13. Y pulsamos sobre [Add](#)

14. En la ventana emergente, pulsamos en [Create](#)

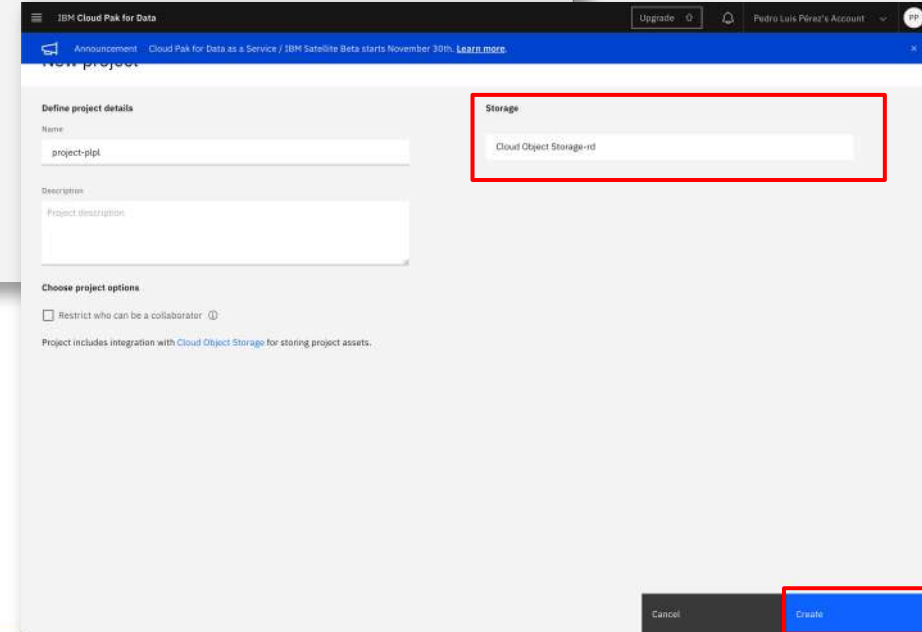
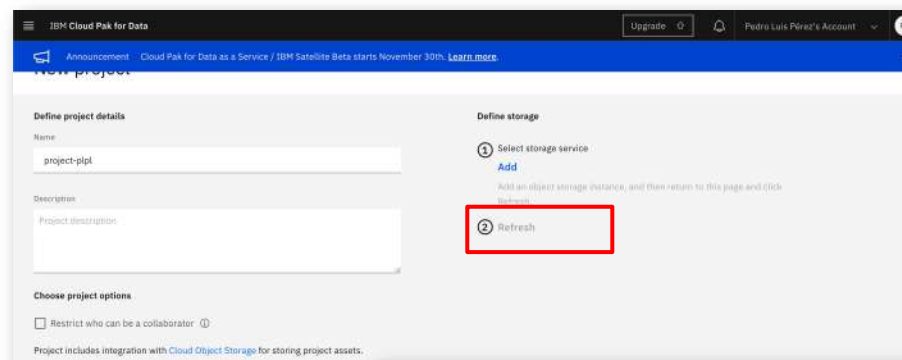


6. Práctica avanzada IoT

6.2. Configuración Watson Studio

15. Pulsamos sobre [Refresh](#)

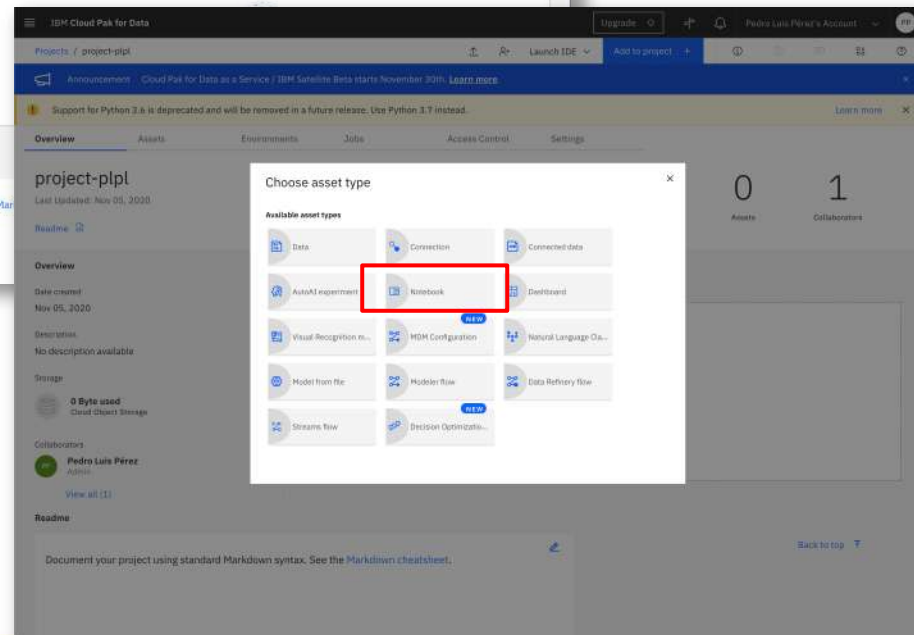
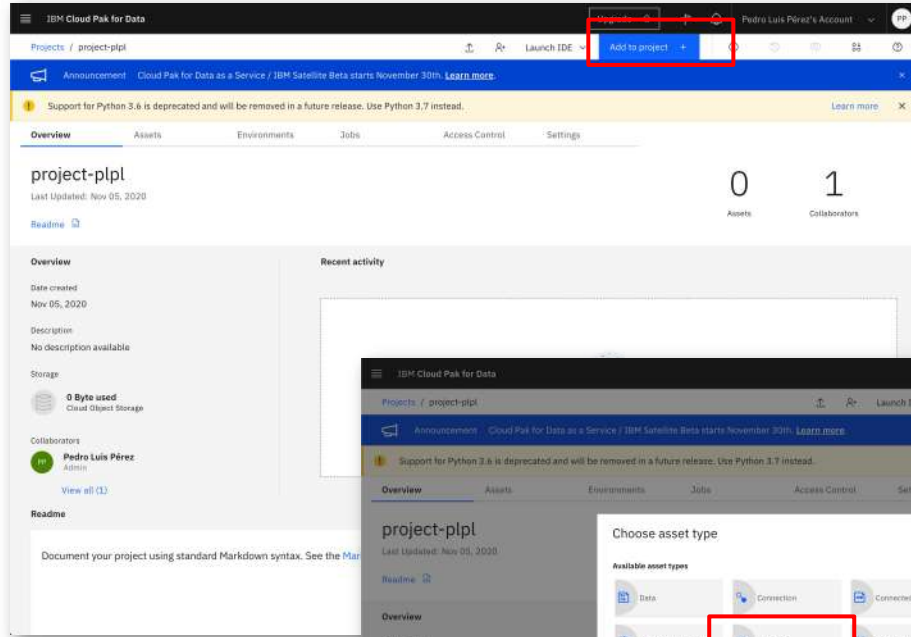
16. Y una vez que aparezca el servicio que acabamos de crear podemos pulsar sobre [Create](#)



6. Práctica avanzada IoT

6.3. Conexión Cloudant con Notebook

1. En la página del Proyecto, pulsamos sobre **Add to Project +**
2. En el desplegable pulsamos sobre **Notebook**



6. Práctica avanzada IoT

6.3. Conexión Cloudant con Notebook

3. Le damos nombre al notebook
4. Como runtime seleccionamos **Default Spark Python 3.6**
5. Pulsamos en **Create**

IBM Cloud Pak for Data

Upgrade ⓘ

Pedro Luis Pérez's Account

New notebook

Blank From file From URL

Name

notebook-plpl

Description (optional)

Type your description here

Select runtime

Default Spark 2.4 & Python 3.6 (Driver: 1 vCPU 4 GB RAM, 2 Executors)

The selected runtime uses 1 driver with 1 vCPU and 4 GB RAM, and 2 executors each with 1 vCPU and 4 GB RAM. It consumes 1.5 capacity units per hour. [Learn more](#) about capacity unit hours and Watson Studio pricing plans.

Language

Python 3.6

Spark version

2.4

Cancel Create

6. Práctica avanzada IoT

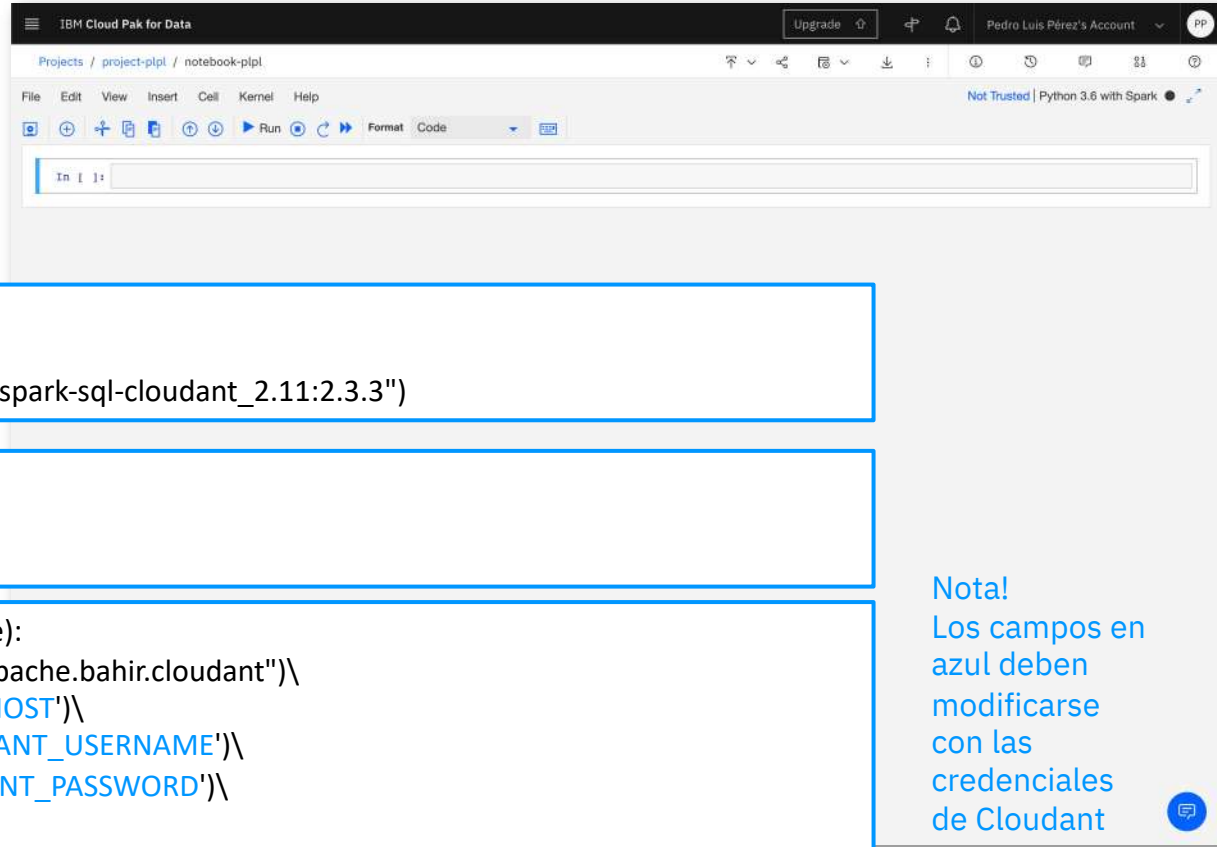
6.3. Conexión Cloudant con Notebook

6. Una vez activo el notebook, introducimos los siguientes comandos:

```
!pip install --upgrade pixiedust
import pixiedust
pixiedust.installPackage("org.apache.bahir:spark-sql-cloudant_2.11:2.3.3")
```

```
import pixiedust
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

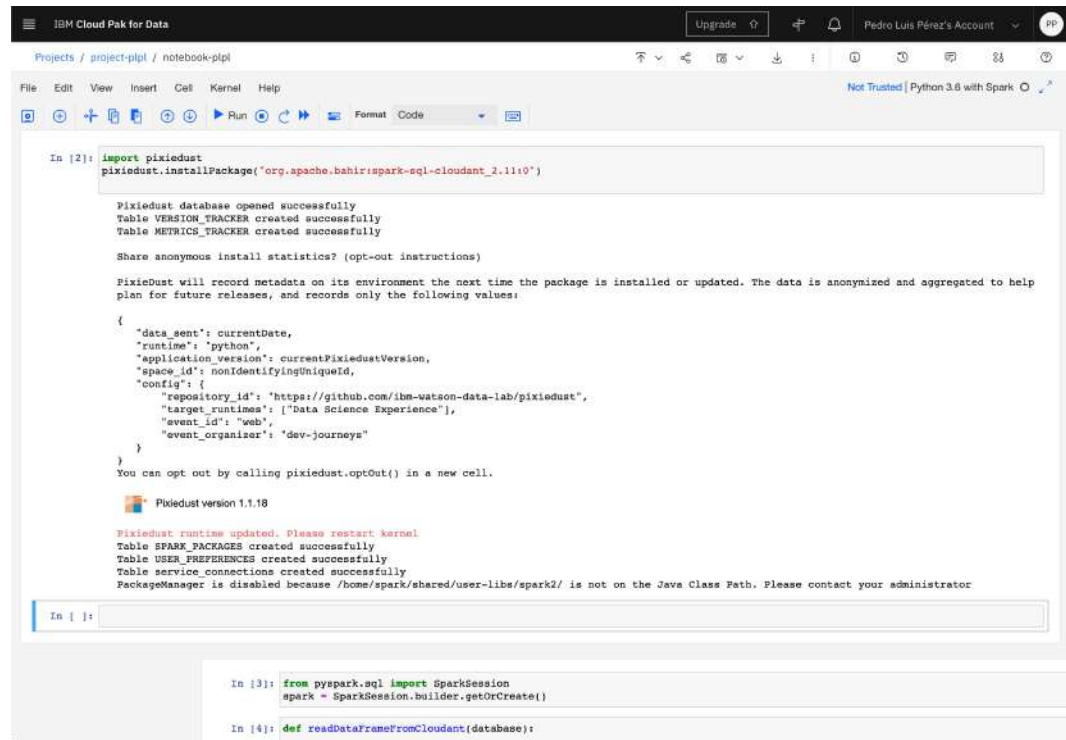
```
def readDataFrameFromCloudant(database):
    cloudantdata = spark.read.format("org.apache.bahir.cloudant")\
        .option("cloudant.host", 'CLOUDANT_HOST')\
        .option("cloudant.username", 'CLOUDANT_USERNAME')\
        .option("cloudant.password", 'CLOUDANT_PASSWORD')\
        .load(database)
    return cloudantdata
```



Nota!
Los campos en azul deben modificarse con las credenciales de Cloudant

6. Práctica avanzada IoT

6.3. Conexión Cloudant con Notebook



The screenshot shows a Jupyter Notebook interface within the IBM Cloud Pak for Data environment. The notebook is titled 'project-plp1 / notebook-plp1'. The code in the first cell installs the 'pixiedust' package from the 'org.apache.bahir:spark-sql-cloudant_2.11.10' repository. The output shows that the database was opened successfully, tables were created, and the package version is 1.1.18. A warning message indicates that the runtime needs to be updated and the kernel restarted.

```
In [2]: !pip install pixiedust
pixiedust.installPackage('org.apache.bahir:spark-sql-cloudant_2.11.10')

Pixiedust database opened successfully
Table VERSION_TRACKER created successfully
Table METRICS_TRACKER created successfully

Share anonymous install statistics? (opt-out instructions)

Pixiedust will record metadata on its environment the next time the package is installed or updated. The data is anonymized and aggregated to help
plan for future releases, and records only the following values:

{
  "data_sent": currentDate,
  "runtime": "python",
  "application_version": currentPixiedustVersion,
  "space_id": nonIdentifyingUniqueId,
  "config": {
    "repository_id": "https://github.com/ibm-watson-data-lab/pixiedust",
    "target_runtimes": ["Data Science Experience"],
    "event_id": "web",
    "event_organizer": "dev-journeys"
  }
}

You can opt out by calling pixiedust.optOut() in a new cell.

Pixiedust version 1.1.18

Pixiedust runtime updated. Please restart kernel
Table SPARK_PACKAGES created successfully
Table USER_PREFERENCES created successfully
Table service_connections created successfully
PackageManager is disabled because /home/spark/shared/user-lib/spark2/ is not on the Java Class Path. Please contact your administrator
```

```
In [3]: from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()

In [4]: def readDataFrameFromCloudant(database):
    cloudantdata = spark.read.format('org.apache.bahir.cloudant')\
        .option('cloudant.host', 'adf3c11b-a7aa-4b91-a966-467bf85379e1-bluemix.cloudantnosqldb.appdomain.cloud')\
        .option('cloudant.username', 'adf3c11b-a7aa-4b91-a966-467bf85379e1-bluemix')\
        .option('cloudant.password', 'cc337ad623die783e6e6a269fc43b89ca781f99421b24d8ad66a3377a58de64c')\
        .load(database)

    return cloudantdata

In [ ]:
```

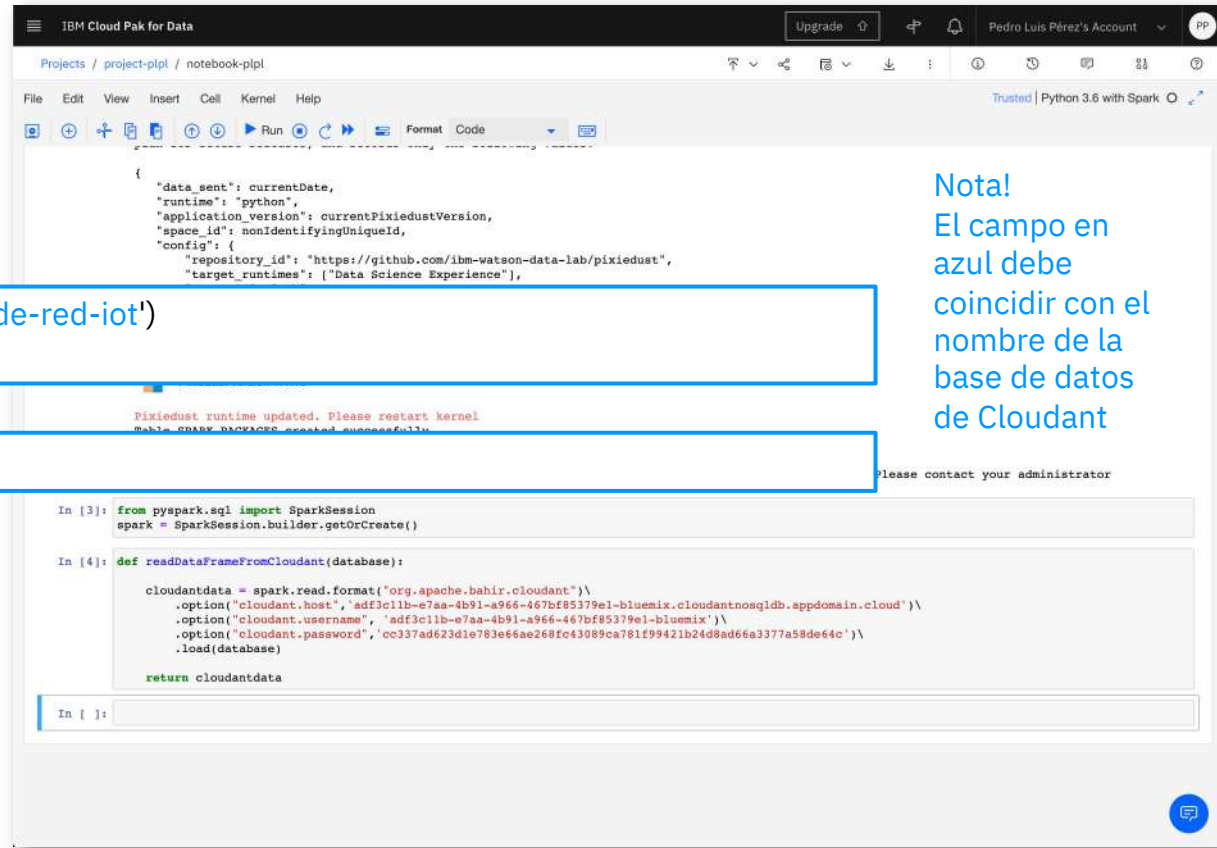
6. Práctica avanzada IoT

6.3. Conexión Cloudant con Notebook

7. Si no hemos recibido ningún mensaje de error, procedemos con los siguientes:

```
df = readDataFrameFromCloudant('node-red-iot')  
df.createOrReplaceTempView('df')
```

```
display(df)
```



6. Práctica avanzada IoT

6.3. Conexión Cloudant con Notebook

- Una vez hecho correr esas dos instrucciones, habremos importado los datos a una tabla.
- A partir de aquí es posible tratar los datos tanto en batch como en streaming.

The screenshot shows the IBM Cloud Pak for Data notebook interface. The top bar indicates the user is logged in as Pedro Luis Pérez's Account. The notebook is titled 'project-plpl / notebook-plpl'. The code cell shows the following commands:

```
In [7]: df = readDataFrameFromCloudant('node-red-data')
df.createOrReplaceTempView('df')
```

The Spark Job Progress section shows a job with 0 progress, 1 stage, and a duration of 4.07 sec.

The code cell shows the following command:

```
In [8]: display(df)
```

The table view displays the following data:

_id	_rev	deviceId	deviceType	eventType	format	topic
064b6ab44a8cbe57d81c5751acc19da	1-5fa217f03dc46716d508ccaf27e51592	112233445566	Android	accel	json	iot-2/type/Android
15e018d9b1e20da1a7adb2734f9415f0	1-0d2b5c69a94d4c4b1a6177c9a3903ab	112233445566	Android	accel	json	iot-2/type/Android
2bhd30db7d46d13a5953c4631904695	1-59b75935910075ee1b3d9f524d7b373	112233445566	Android	accel	json	iot-2/type/Android
4498d007e7d4171142039e811340b8f	1-5c9e9db6d80094d8a799ac13445d8cce	112233445566	Android	accel	json	iot-2/type/Android
4be567cc81c79a586d09178f067b49f	1-5ac171f06cd03988e6aa6a1f52b7e14	112233445566	Android	accel	json	iot-2/type/Android
5215bb57dc5410a4808bdaa60095e723	1-35db4ef7259278ba9c2b16cc472246e0d	112233445566	Android	accel	json	iot-2/type/Android

7. Recomendaciones finales

Una vez terminada la práctica se recomienda

- Desactivar el sensor de la app para que deje de realizar envíos / borrar la app.
- En IBM Cloud, aunque la cuenta es Lite y no debería tener ningún servicio de pago asociado, eliminar las instancias desplegadas para dejar la cuenta limpia.

Muchas gracias!

Pedro Luis Pérez Latorre

—

pedro.luis.perez@ibm.com

ibm.com

© Copyright IBM Corporation 2020. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., 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 at [Copyright and trademark information](#).