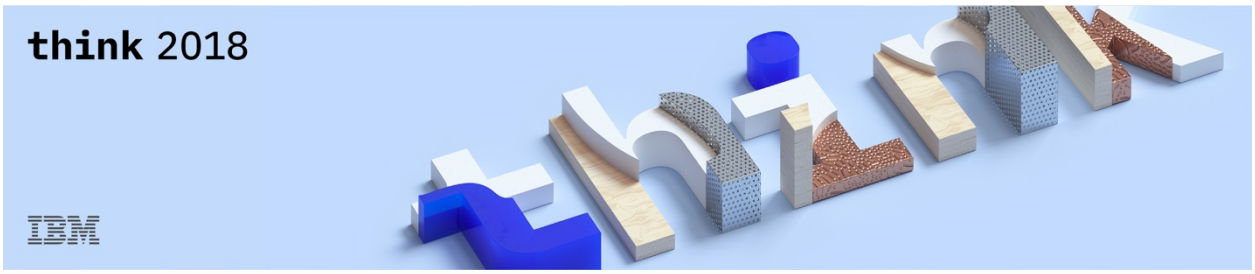


Lab Center – Hands-on Lab

Session 4966

Getting Started with Watson IoT Edge Gateways

Mats Gothe
Senior Design Lead
Watson Content & IoT Platform

mats.gothe@se.ibm.com

1 Table of Contents

Disclaimer

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

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

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

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

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed.
Regardless, our warranty terms apply.”

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

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

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

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor

shall constitute legal or other guidance or advice to any individual participant or their specific situation.

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

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

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

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

1 Introduction to this lab

1.1 Welcome to this lab on Getting started with Watson IoT Edge Gateways

In this hands-on lab you will deepen your understanding about the Watson IoT Platform and the IoT Edge. You will use the Watson IoT Platform to create and configure an IoT Edge Gateway and an Edge Sensor device. You will configure and connect a small Raspberry Pi device to run as a IoT Edge Gateway. And you will develop and deploy IoT Edge workloads, like Data Management data transformation to the Edge Gateway. You will also select, provision and monitor the edge services and analytics workloads to run on the Edge Gateway.

1.2 About Watson IoT Platform

The IBM Watson Internet of Things Platform is a fully managed, cloud-hosted service available in IBM Cloud.

Devices connect and start sending data securely to the Watson IoT Platform service using the light-weight standardized MQTT messaging protocol. From there, applications can access real-time device data, and devices can be managed using secure APIs or the IoT Platform dashboard.

1.3 About Watson IoT Edge

Edge devices generally live on the edge of the network within an enterprise. They are specific hardware sensors, industrial controllers, or other industrial devices residing on private or potentially proprietary networks at a physical location.

Edge devices are creating quite a challenge for systems integrators since most are generally building cloud-based applications which rely on receiving information from IoT devices to accomplish the business goal. There are many challenges that exist in developing a path for this information flow, including: message/network protocols, bandwidth requirements, redundant data reporting and data routing/orchestration

The IBM Watson Internet of Things platform is developing new IoT Edge capabilities to create, configure and manage distribution of workloads to IoT Edge gateways. In this lab we will use IoT Edge Preview 2 beta.

Using the Watson IoT platform and IoT Edge, applications can

- Connect devices at the edge to an IoT Edge Gateway
- Avoid data loss by using the Store and Forward capability on the IoT Edge gateway

- Filter and Transform data from connected devices by distributing Data Management workloads from the Watson IoT Platform service at the IBM Cloud to the IoT Edge gateway
- Create custom edge services using Docker images and automate provisioning these services using the IoT Edge Service Catalog
- Use the MQTT message broker and routing rules on the IoT Edge gateway to publish and subscribe to messages across edge services on the IoT Edge gateway, the IoT Platform and other services on the IBM Cloud.

Watson IoT Platform and IoT Edge provides:

- IoT Edge Agent for Intel 64-bit, ARM 64-bit and ARM 64-bit Linux architectures that is installed and running on the Edge Gateway to manage the IoT edge service
- IoT Edge Service Catalog with edge services that can be configured to run on edge gateways
- Additions to Device Types and Devices in the IoT Platform Dashboard and APIs to create Edge Gateways.
- Additions to Devices in the IoT Platform Dashboard and APIs to monitor the status of edge services running at an edge gateway

In the first section of this lab you will be getting started on your workstation and logging into IBM Cloud and the Watson IoT Platform.

After the first section you will be familiarizing yourself with the Watson IoT Platform and new capabilities for IoT Edge and then start configuring the Watson IoT Platform for IoT Edge gateways and sensor devices.

2 Getting started

2.1 Starting your Workstation

In this lab you will use a Windows 7 workstation. This workstation is used to run Firefox Web browser. All access to IBM Cloud and IBM Watson IoT Platform will be made using the Firefox Web browser. Using the workstation, you will also establish an SSH connection to the Raspberry Pi using the PuTTY program.

At the start of this lab, all workstations should have been started and ready for you to use with automatic login.

Note: If you fail to log into your workstation, ask your lab facilitators for help

2.2 Logging into IBM Cloud and Watson IoT Platform

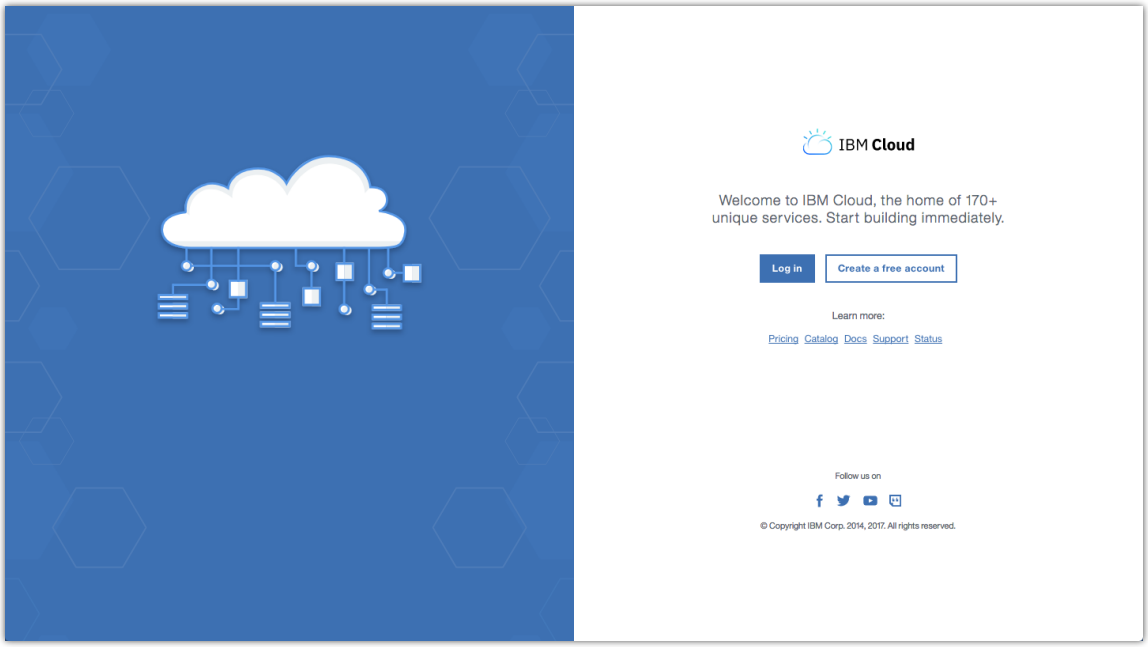
IBM Cloud, previously named Bluemix, is a cloud platform as a service (PaaS) developed by IBM. It supports several programming languages and services as well as integrated DevOps to build, run, deploy and manage applications on the cloud. IBM Cloud is based on Cloud Foundry open technology and runs on SoftLayer infrastructure.

The IBM Watson Internet of Things Platform is a fully managed, cloud-hosted service available in IBM Cloud, that makes it simple to derive value from Internet of Things (IoT) devices.

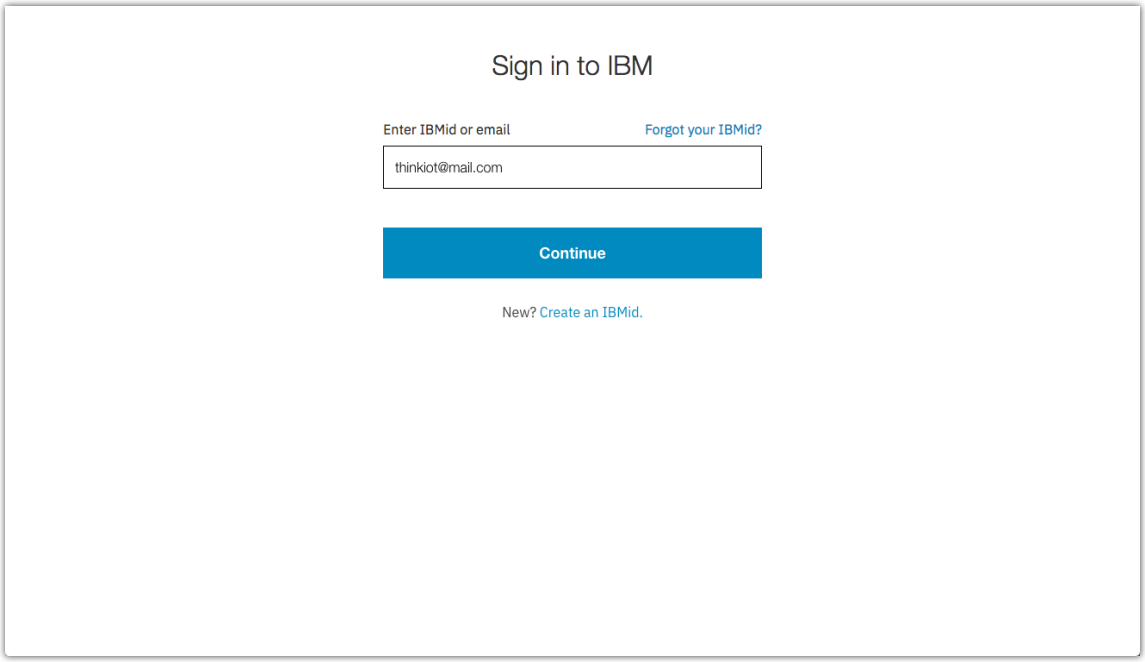
Devices can get connected and start sending data securely to the IBM Watson Internet of Things Platform cloud service using the open, lightweight MQTT messaging protocol. From there, you can setup and manage your devices using your online dashboard or our secure APIs, so that your apps can access live and historical data quickly. With your devices connected to the IoT platform, you are now ready to start creating applications using your device data.

In this lab you will use the IBM Watson Internet of Things Platform service available in IBM Cloud. To get access to the IoT platform you first have to log into IBM Cloud and then browse to the IBM Watson Internet of Things Platform service.

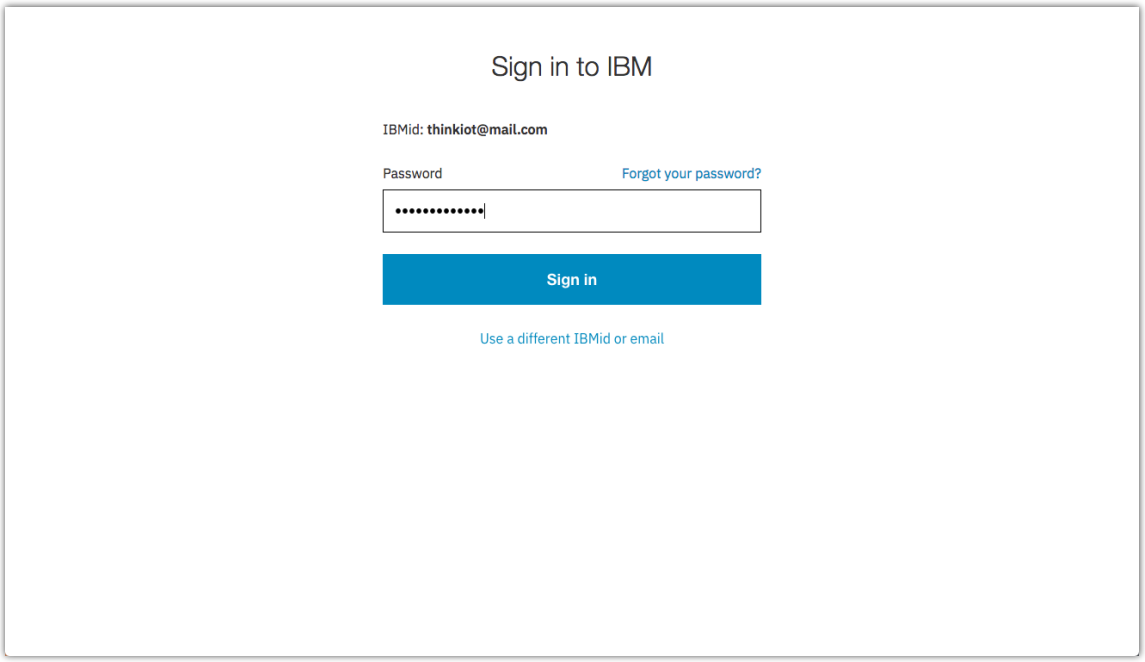
1. Open the Firefox browser on your workstation
2. Enter https://bluemix.net
   The IBM Cloud welcome page opens



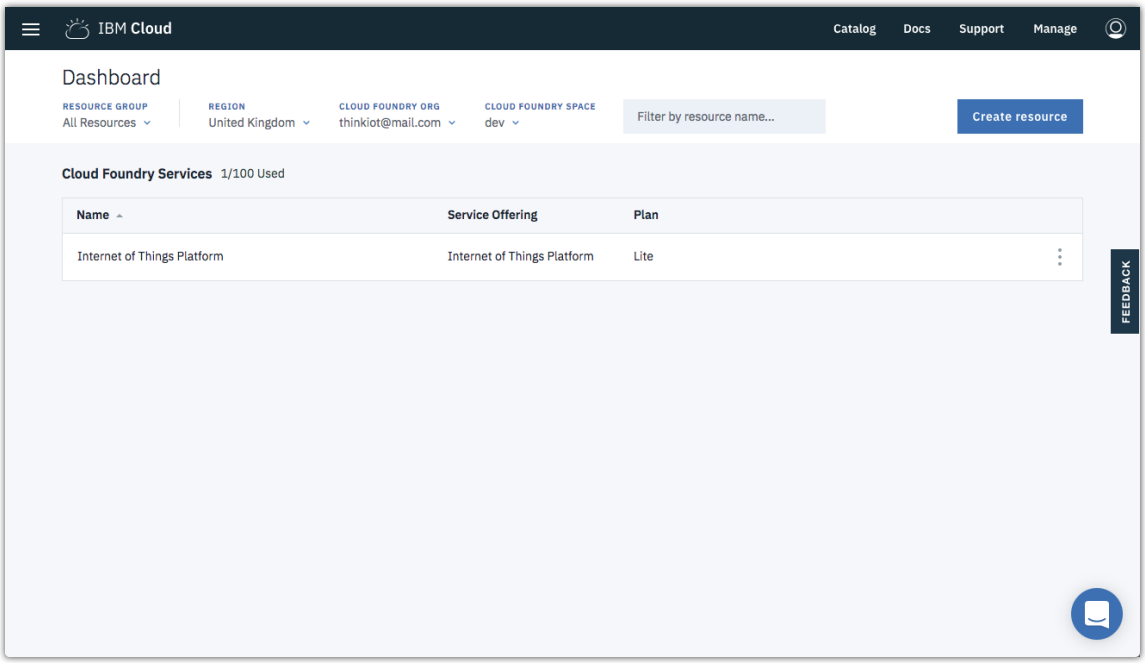
3. Click on **Log in**
4. Enter the IBM ID



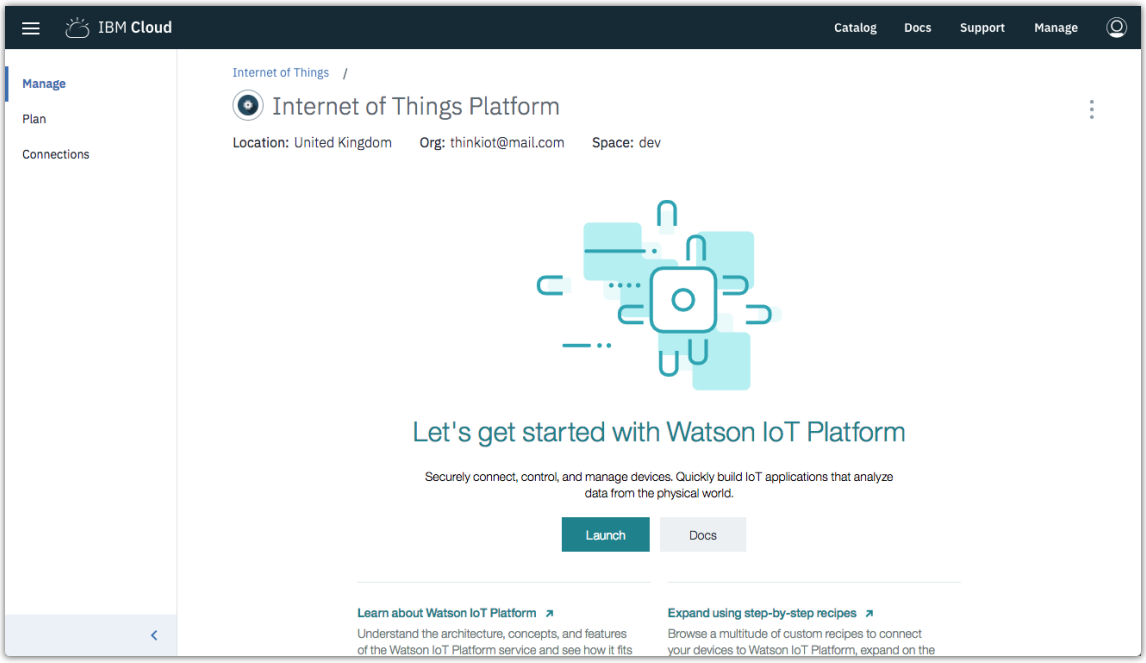
5. Enter the password and click **Sign in**



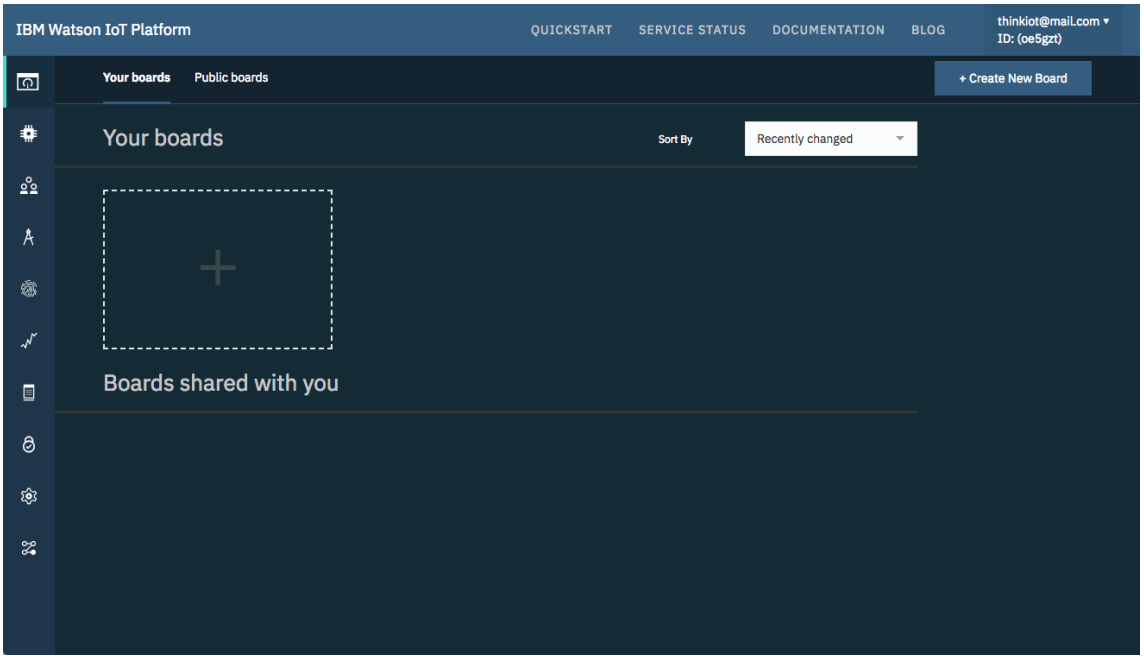
6. The IBM Cloud Dashboard is loaded.



7. In the list of Services, click on the **Watson Internet of Things Platform** service. The Watson Internet of Things Platform service opens.



8. Click **Launch** to open the IoT platform web interface.
   The Watson IoT Platform opens and shows the platform dashboard.



You have now successfully logged into IBM Cloud and launched the IBM Watson IoT Platform service. You are now ready to switch to the IoT platform *organization* that you will use in this lab.

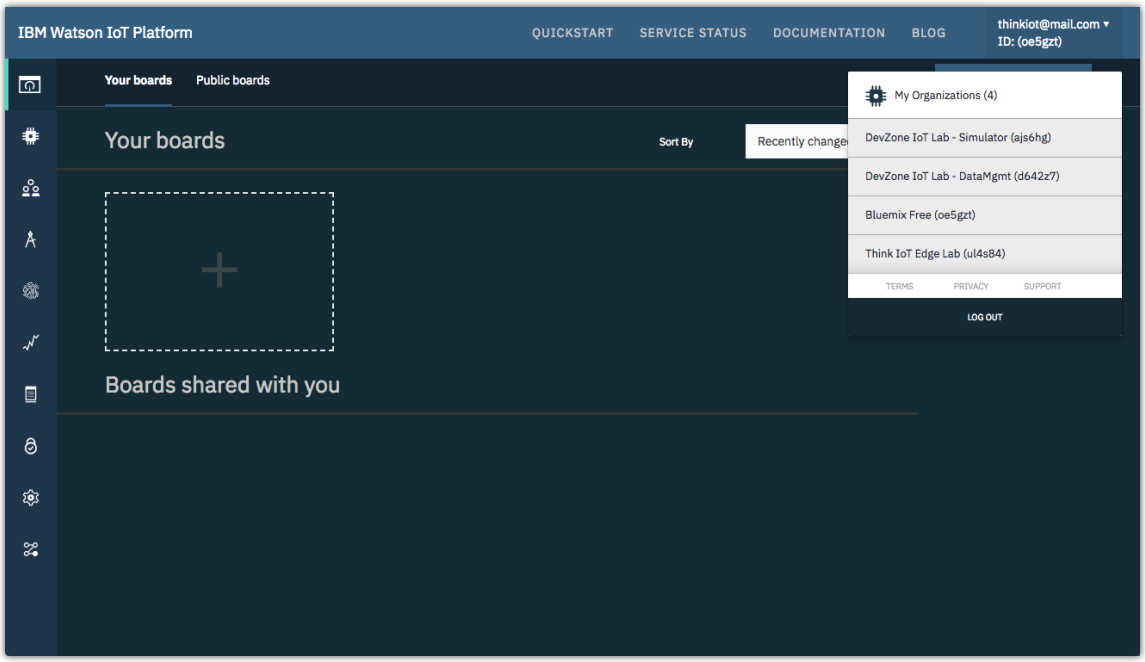
When you register with the Watson IoT Platform, you are given an organization ID. Your organization ID is a unique six-character identifier for your account. Organizations ensure that your data is organized and accessible by your devices and applications. An IoT platform organization is hence a workspace that independently of other organizations

manages users, devices and device data. An organization can be shared, and you can invite other IBM Cloud users to your organization.

In this lab we will use a shared IoT platform organization across all lab workstations. Your lab facilitators will inform you about the organization id and organization name.

To switch to the organization

1. Click on the organization menu in the upper right-hand corner of the IoT Platform screen.
2. In the list, choose the **Think IoT Edge Lab (ul4s84)** organization name to be used in this lab.
3. The Watson IoT Platform will reload the page and open the selected organization

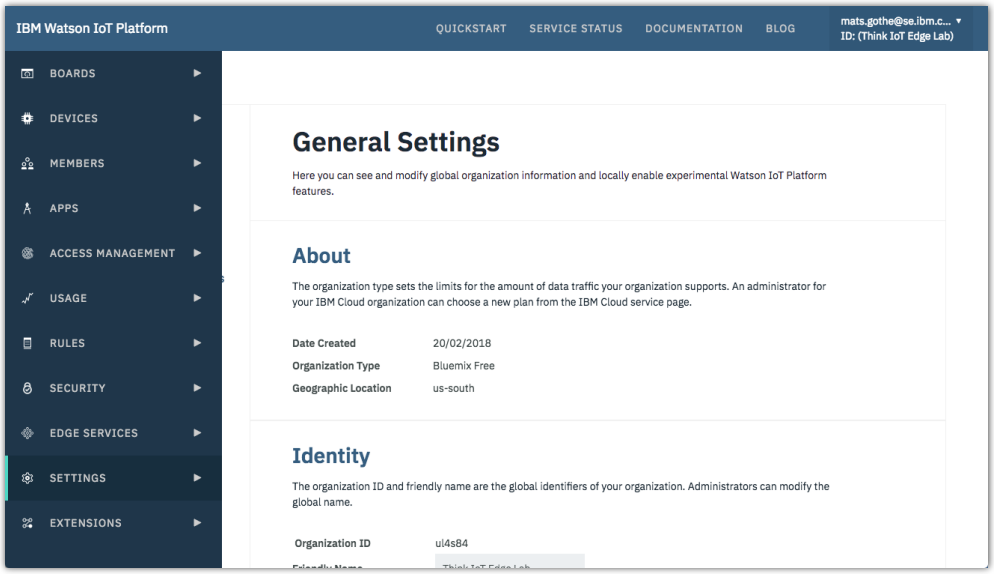


You have now successfully switched to the Watson IoT Platform organization that we will use in this lab.

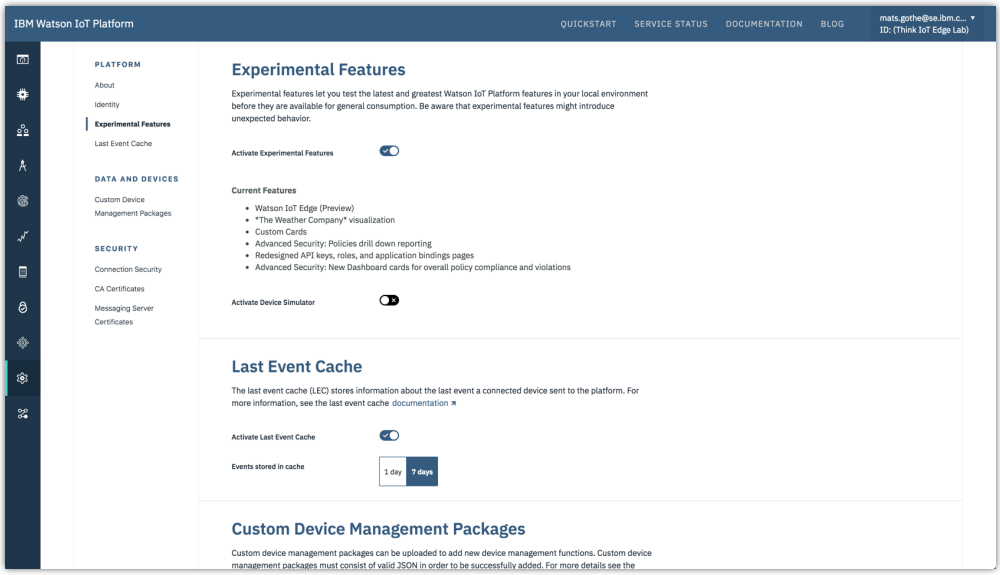
2.3 Activating IoT Edge Preview 2 Beta

In this lab we will be using the Watson IoT Platform and IoT Edge Preview 2 Beta.
To activate the IoT Edge Preview 2 Beta

1. From the navigation bar, select **Settings** to open the settings page.



2. Scroll down to the Experimental Features section,
   or click on **Experimental Features** in the left-hand section outline



3. Experimental Features section, turn on the switch to **Activate Experimental Features**.

You have now activated Watson IoT Edge Preview 2 beta.

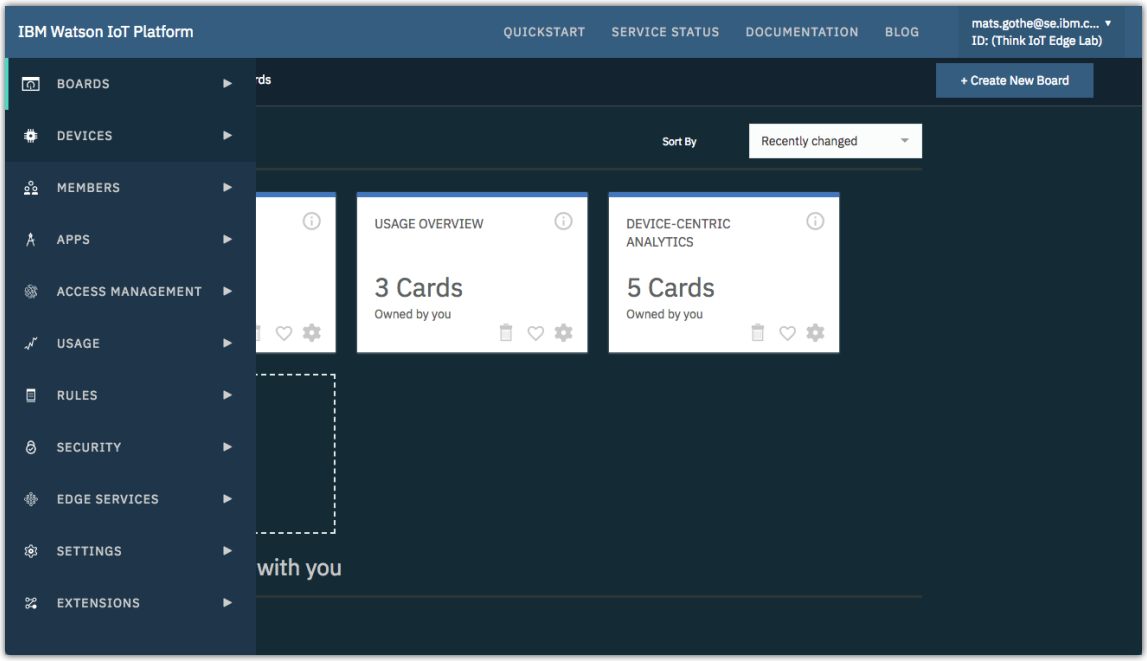
In the next section you will get an overview of the capabilities in Watson IoT Platform.
We will also introduce the topic of IoT Edge.

3 Overview of the Watson IoT Platform

IBM Watson IoT Platform provides a powerful web application for administrators, operators and developers with secure access to IoT devices and device data. In this section of the lab you will familiarize yourself with the IBM Watson IoT Platform user interface.

The navigation bar on the left-hand side provides access to the capabilities of the platform

- Boards – Opens the dashboard and shows the boards and cards
- Devices – Opens a browser for registered devices and their device types
- Members – User management
- Apps – API Key management
- Access Management – Roles and Permissions
- Usage – Metrics of usage
- Rules – Analytics rules and actions
- Security – Risk and Security Policies
- Edge Services – Catalog of Services configurable to run on edge gateways
- Settings – Administration settings. For example, client and server certificates.
- Extensions – Additional capabilities, optionally enabled

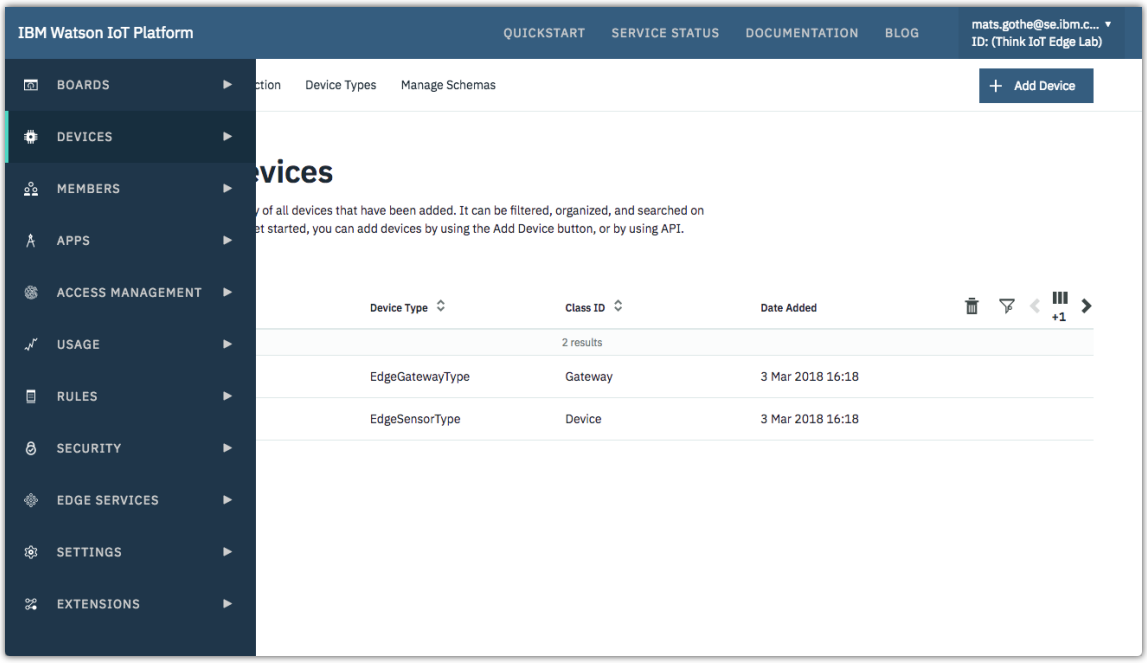


3.1 Exploring Devices and Device Types

Devices are things, like smart physical sensors and actuators, that connect to the IoT platform over internet. The Watson IoT Platform helps you define and manage your devices.

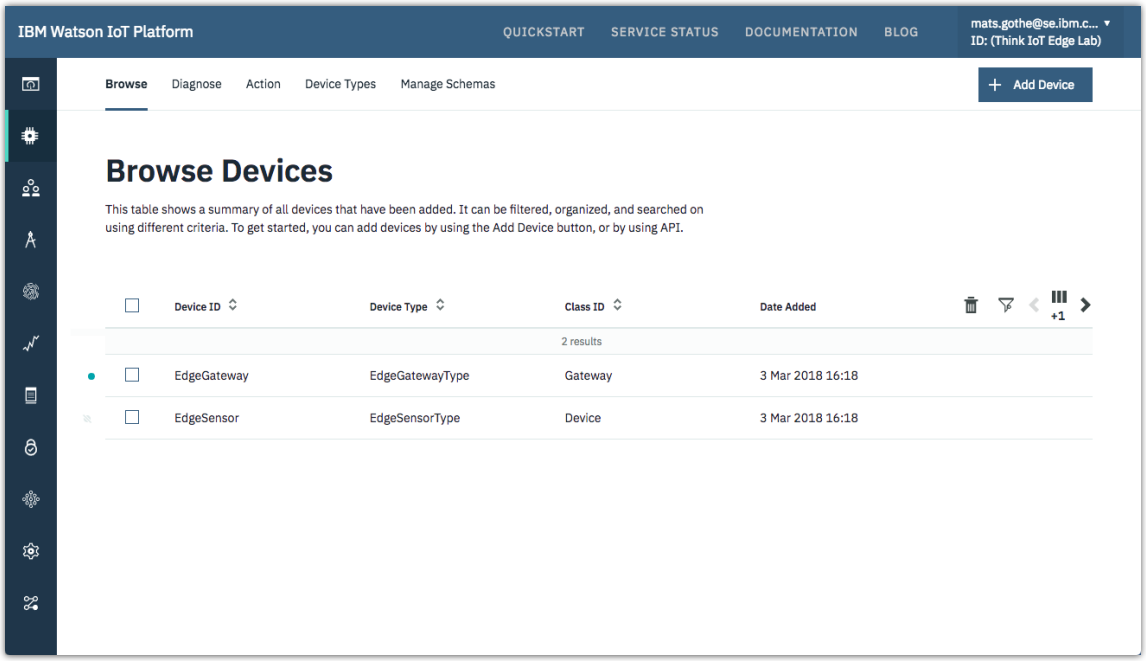
To view devices registered in the IoT platform organization

1. Move your mouse pointer to the left side navigation bar. The navigation bar slides out and shows the IoT platform capability sections.



2. In the navigation bar, choose **Devices.**
   The Devices page opens. This view shows all devices registered in this organization.

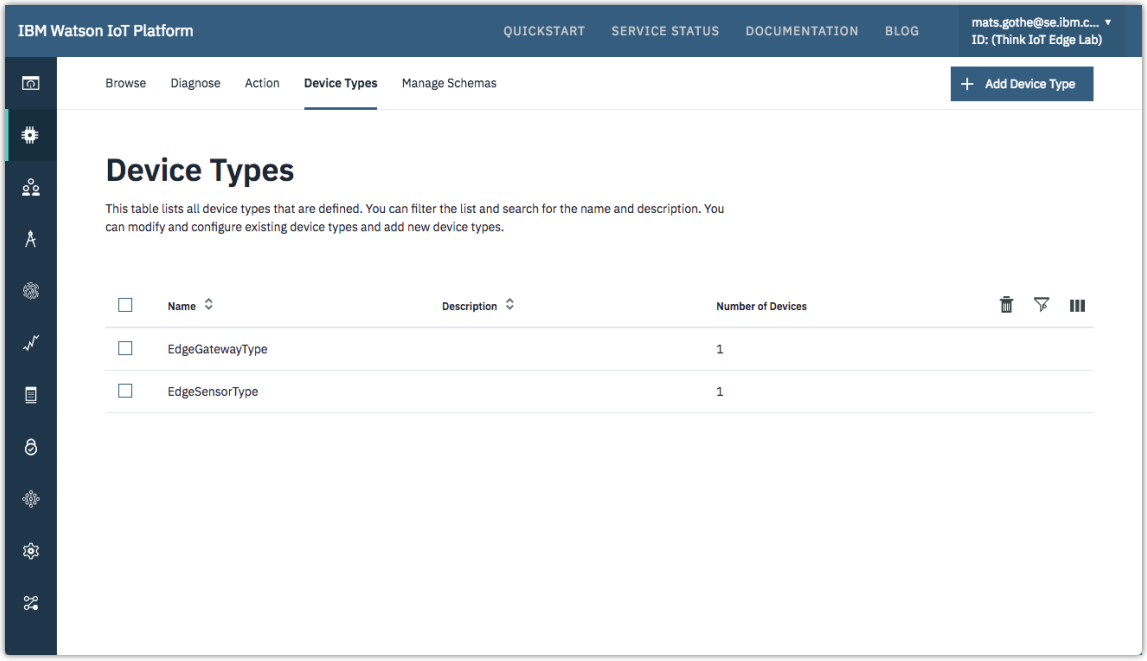
   Note: At the start of the lab there will only be a few devices created in the IoT platform organization.



Devices of the same type, brand, make or model are of a *Device Type*. The devices view above also indicate the type for each registered device.

To view all Device Types in the IoT platform organization

1. From the section tabs at the top of the page, choose **Device Types**.
2. The Device Types page opens. This page shows all device types registered in this organization.



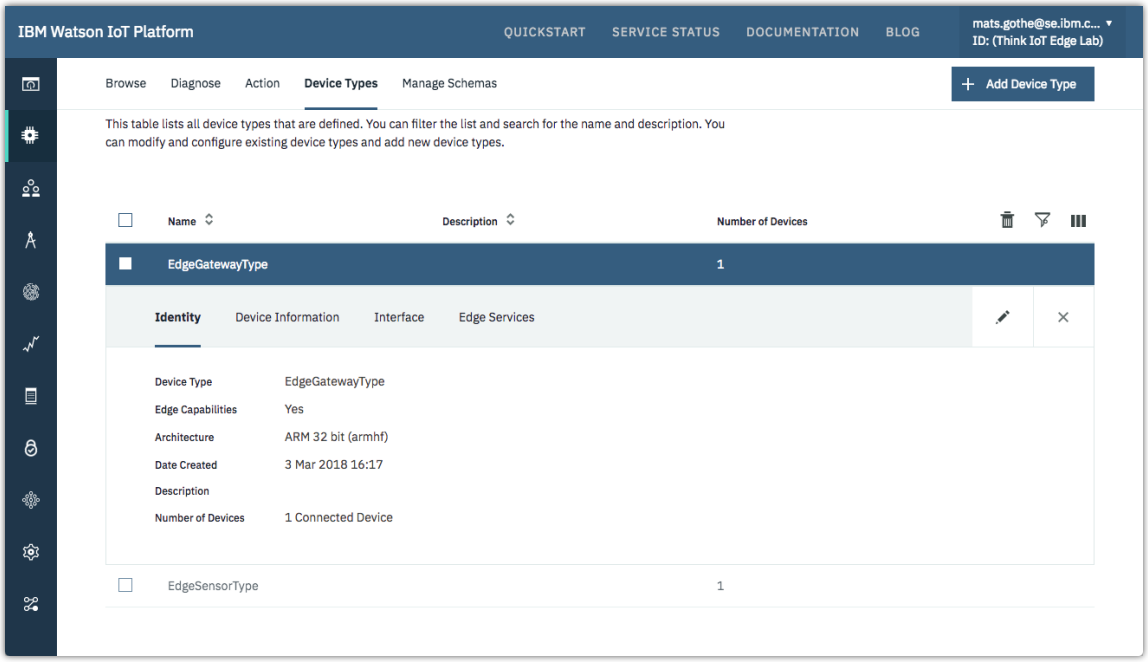
Note: The Device and Device Type pages provides a list of existing resources in the IoT Platform. Using the **+ Add Device Type** and **+ Add Device** buttons, new devices and types can be added to the IoT platform

The IoT platform also categorizes devices and device types in different kinds. Each kind has a special behavior and purpose.

- *Devices* are sensors and actuators that connect directly to the IoT platform using MQTT or HTTPS protocols. Devices publish messages with sensor readings
- *Gateways* are network devices that connect directly to the IoT platform and acts on behalf of devices that are connected to the gateway. Gateways pass events and commands from and to their connected devices.
- *Edge Gateways* are a special kind of gateways that can run IoT Edge workloads like data management transformation and custom developed edge services.

To view details on a device or a device type

1. Click on a device or a device type row in the list view. For example, the device type **EdgeGatewayType**
2. The list view row expands and shows detailed information of the selected device or a device type.



Note: The Device and Device Type pages provides a list of existing resources in the IoT Platform. Using the **+ Add Device Type** and **+ Add Device** buttons, new devices and types can be added to the IoT platform

You have now explored Devices and their Device Types in your IoT platform organization. We will now proceed and explore IoT Edge Services.

3.2 Exploring Edge Services

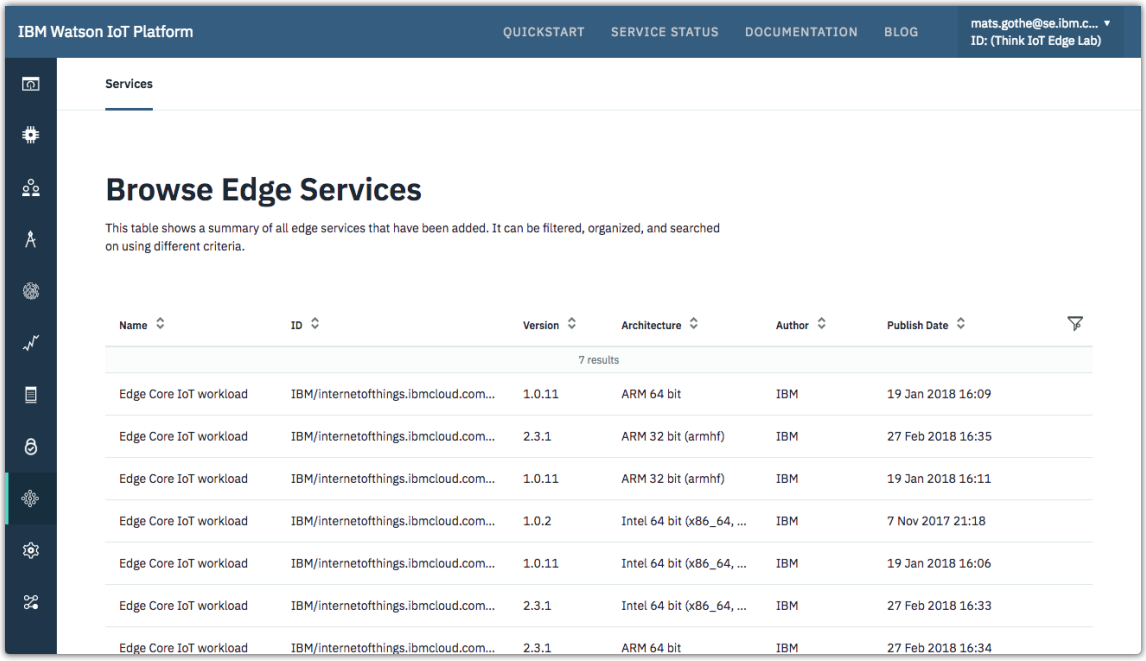
The Watson IoT Platform provides an IoT Edge Service Catalog with all edge services available to edge gateways. The catalog contains IoT edge services that are provided by IBM authors or by the authors in the currently selected IoT Platform organization.

The IoT Edge Service Catalog maintains detailed information on each IoT Edge service

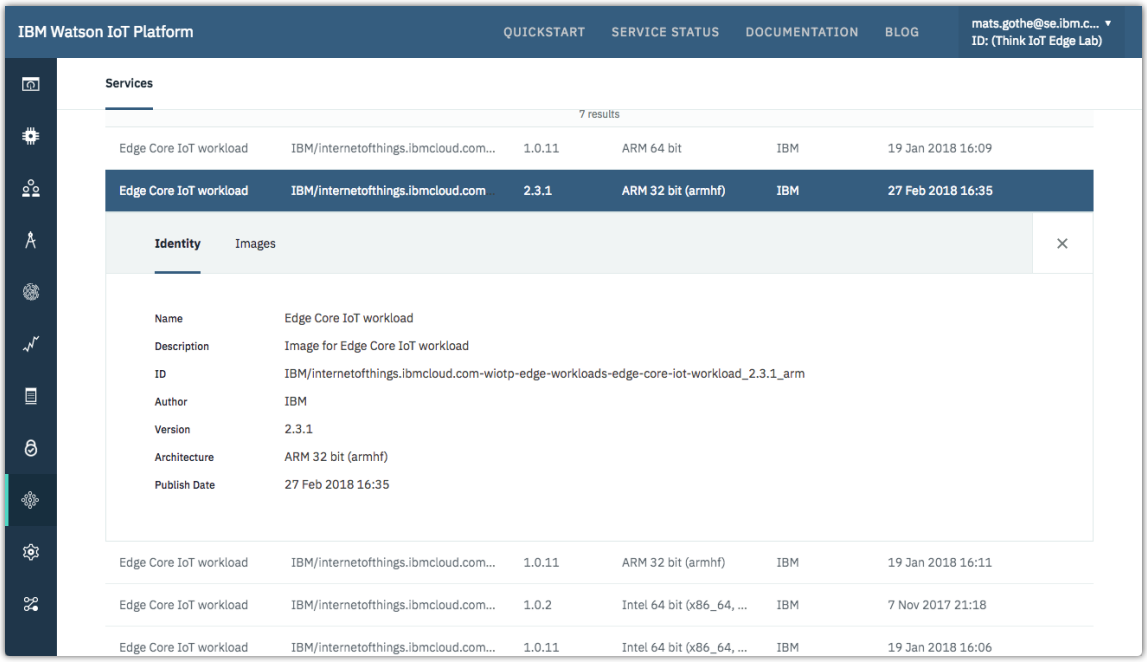
- The name and version of the service
- The gateway architecture of the service, for example AMD64, ARM32 or ARM64
- The author and the creation date
- The images composing the service

To view services in the IoT Edge Service Catalog

1. From the navigation bar, select **Edge Services** to navigate to the services section.
2. The IoT Edge Service Catalog opens with the **Services** tab selected.

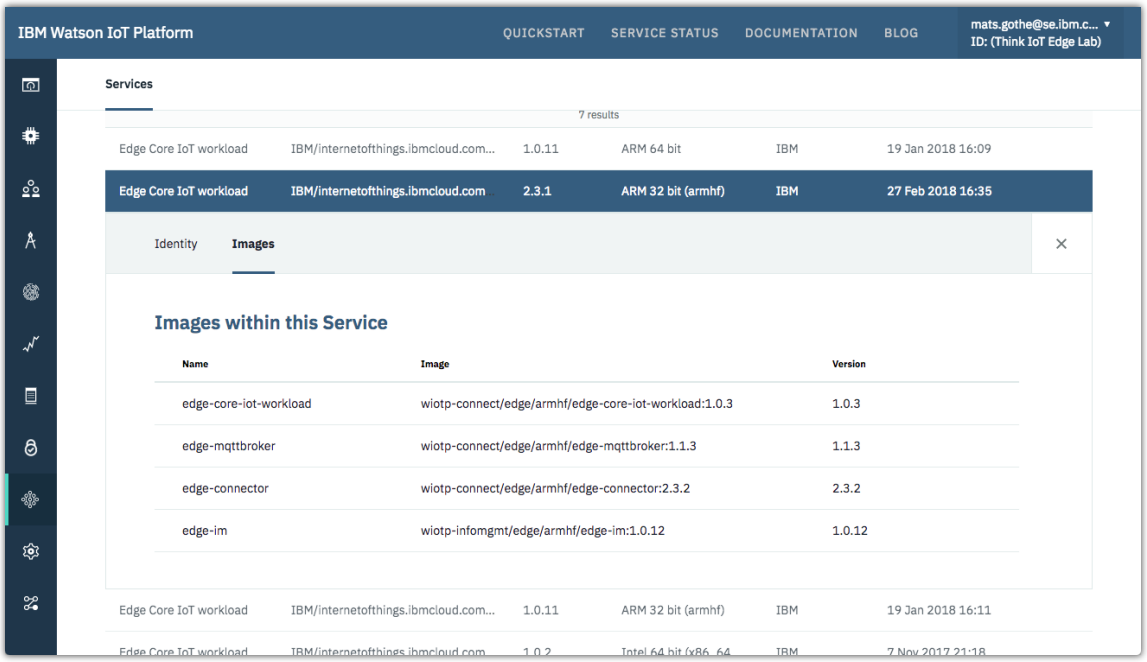


3. Click and expand on of the edge services, for example the **Edge Core IoT workload** for **ARM 32 bit.**
   The row expands and shows the detailed information on the selected edge service.



4. Click on the **Images** tab.

   Note: The images tab shows the list of Docker images that has been configured for this service. These images will be downloaded and run as Docker containers on edge gateway that are configured to run this edge service.



You have now a basic understanding of device types, devices and gateways, and edge services

We will now create a new Gateway Type and select services from the IoT Edge Service catalog.

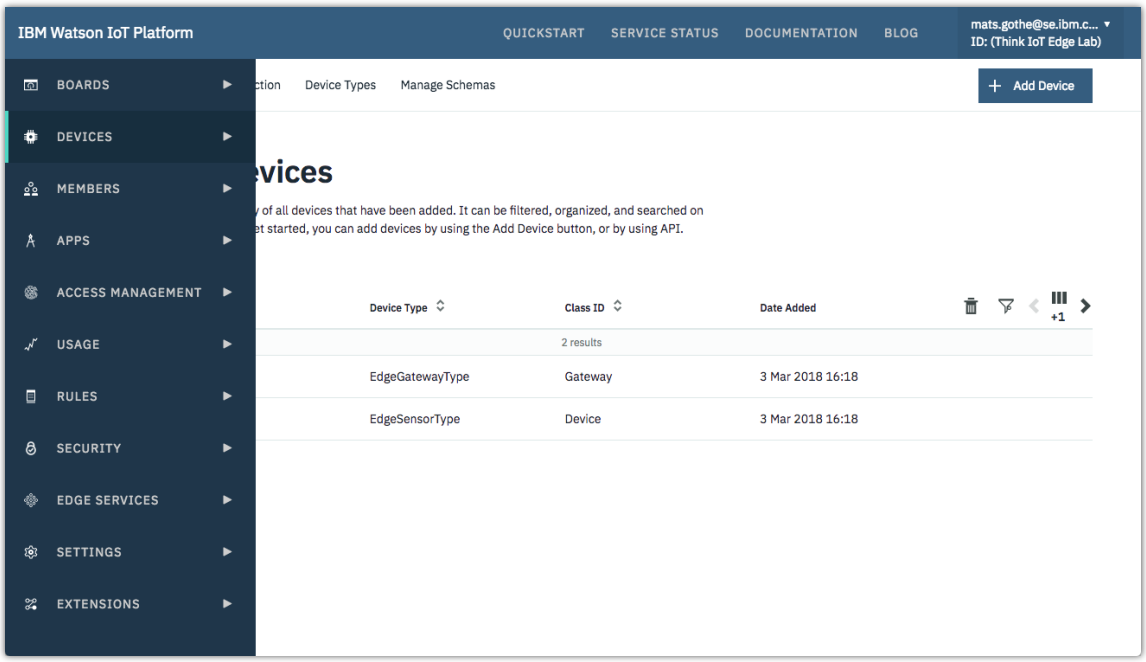
4 Configuring Watson IoT Edge

4.1 Creating a new Edge Gateway Type

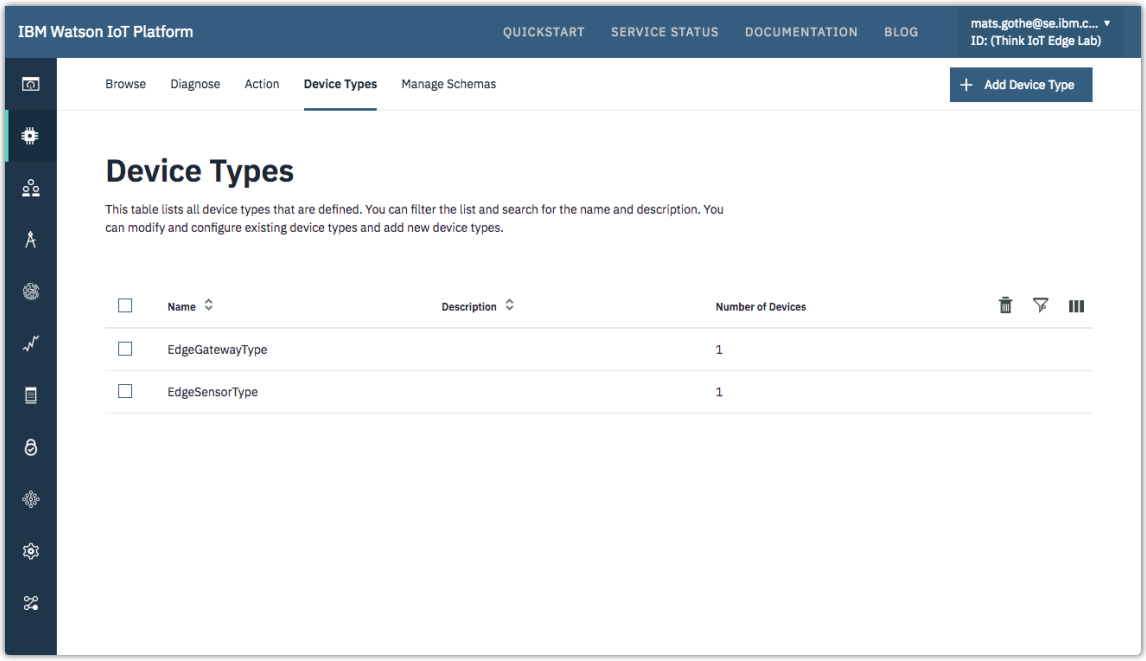
You will now create a new device type representing your type of edge gateways.

To create a new Device Type

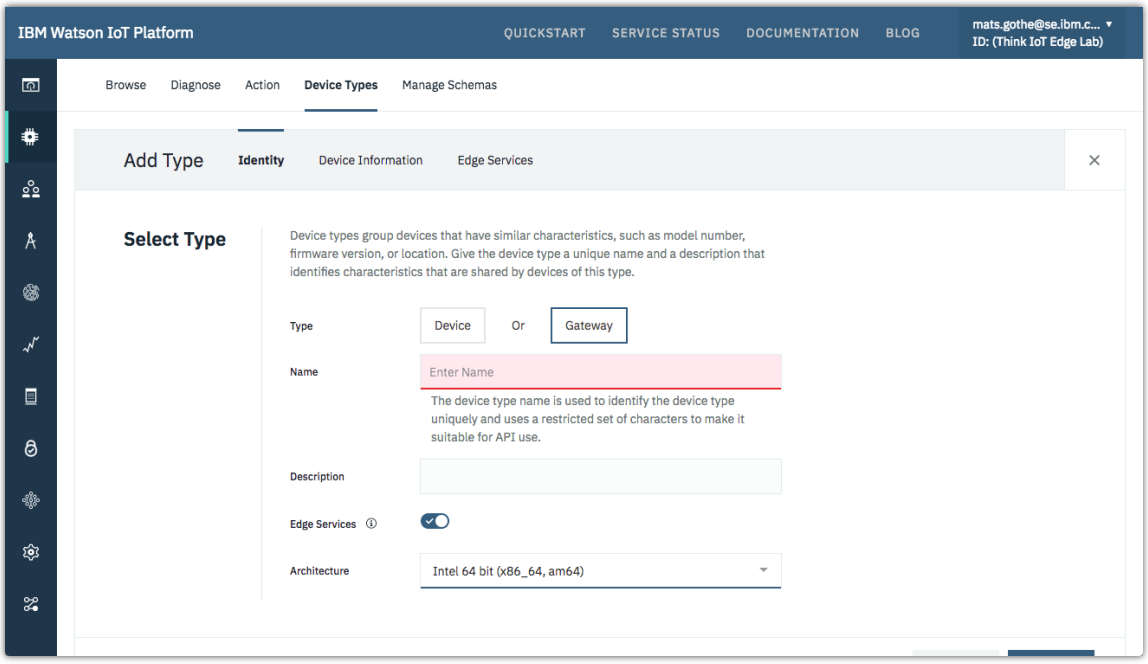
1. On the navigation bar, select **Devices** to navigate to the devices section.



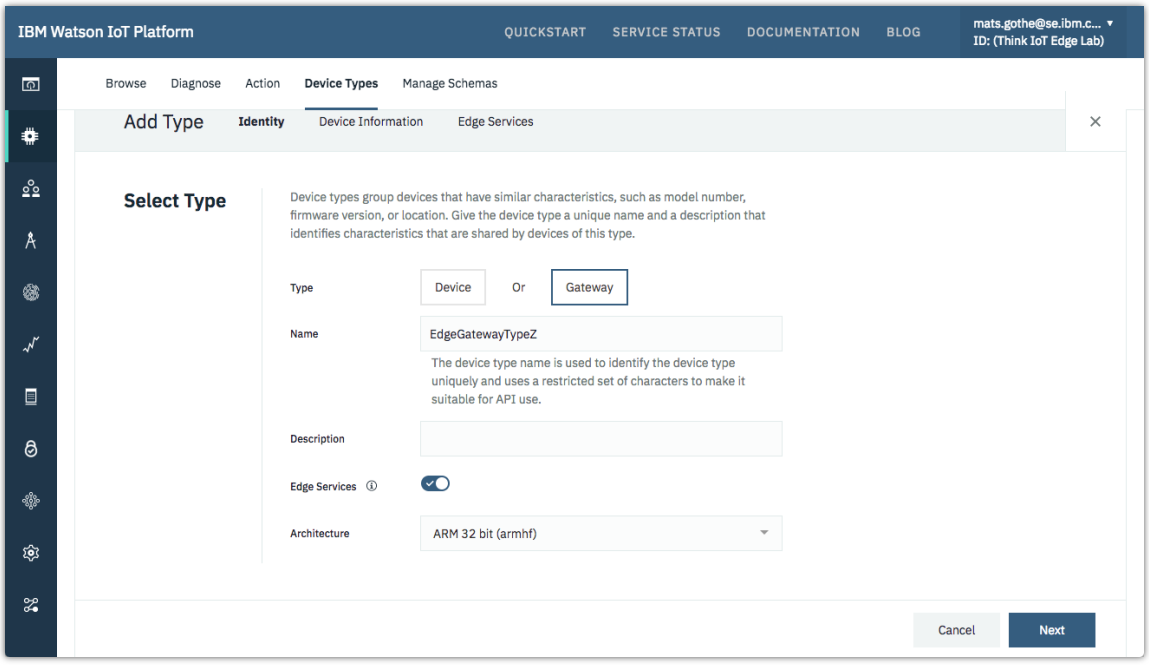
2. Choose the **Device Types** tab.



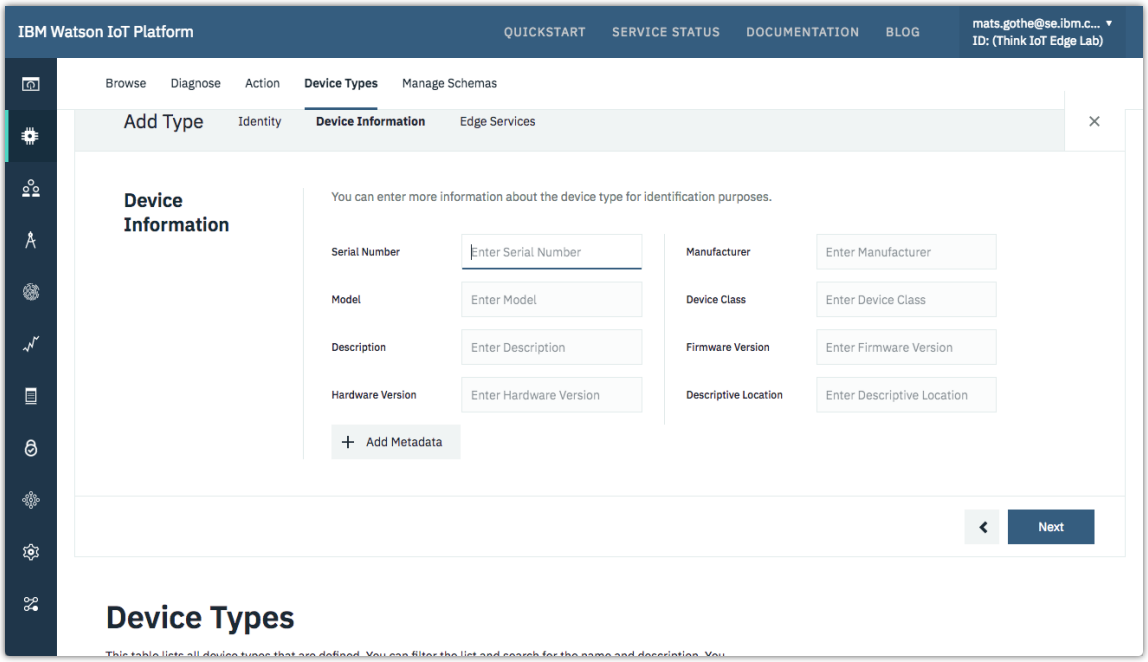
3. Click the **+ Add Device Type** button to add a new device type.
   The Add Type page opens.



4. On the Add Type page
   a. Select the **Gateway** type
   b. Enter a unique **Name**. Use the gateway type name given for your lab station
   c. Turn on the **Edge Services** to make this type an Edge Gateway Type
   d. In the Architecture drop-down, choose the **ARM 32 bit (armhf)** option

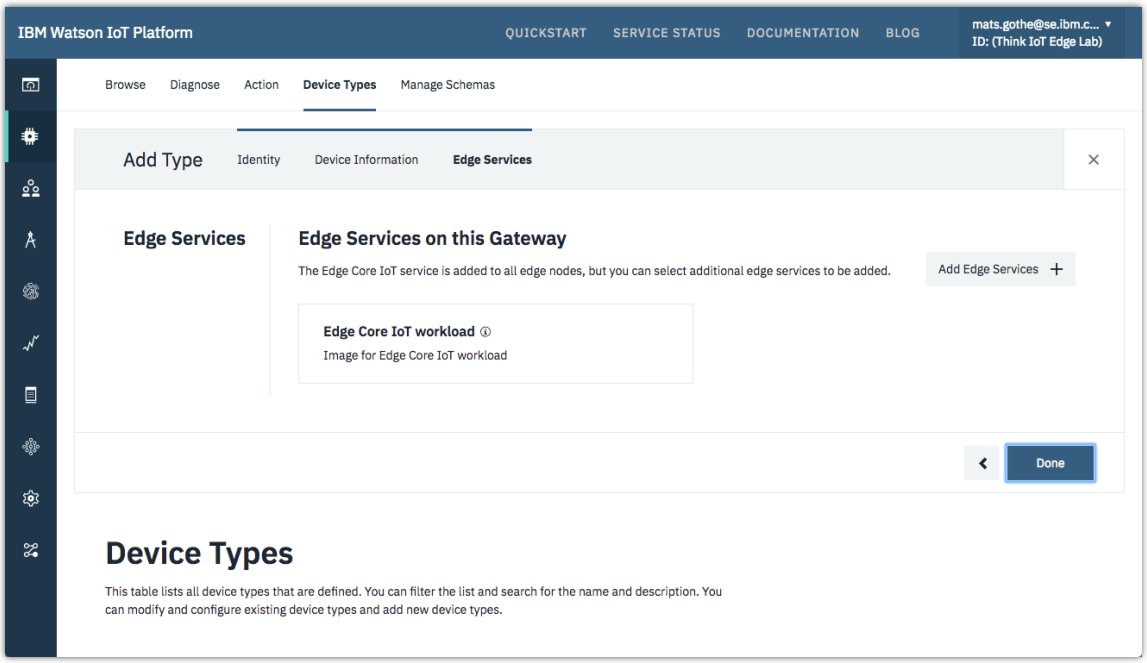


5. Click **Next**. The Device Type Information page opens.



Note: A device type may optionally specify a default template for device information metadata. In this lab we will not use this device information.

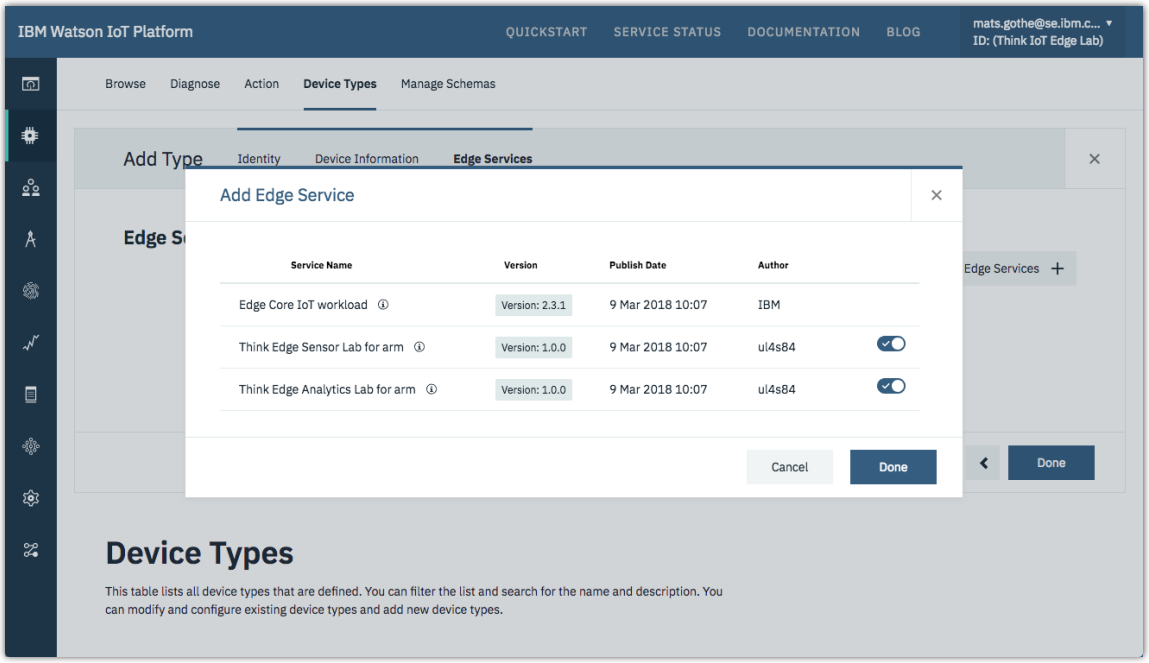
6. Click **Next**. The Edge Services page opens. In this page you can select and configure the edge services to run on edge gateways of this gateway type.



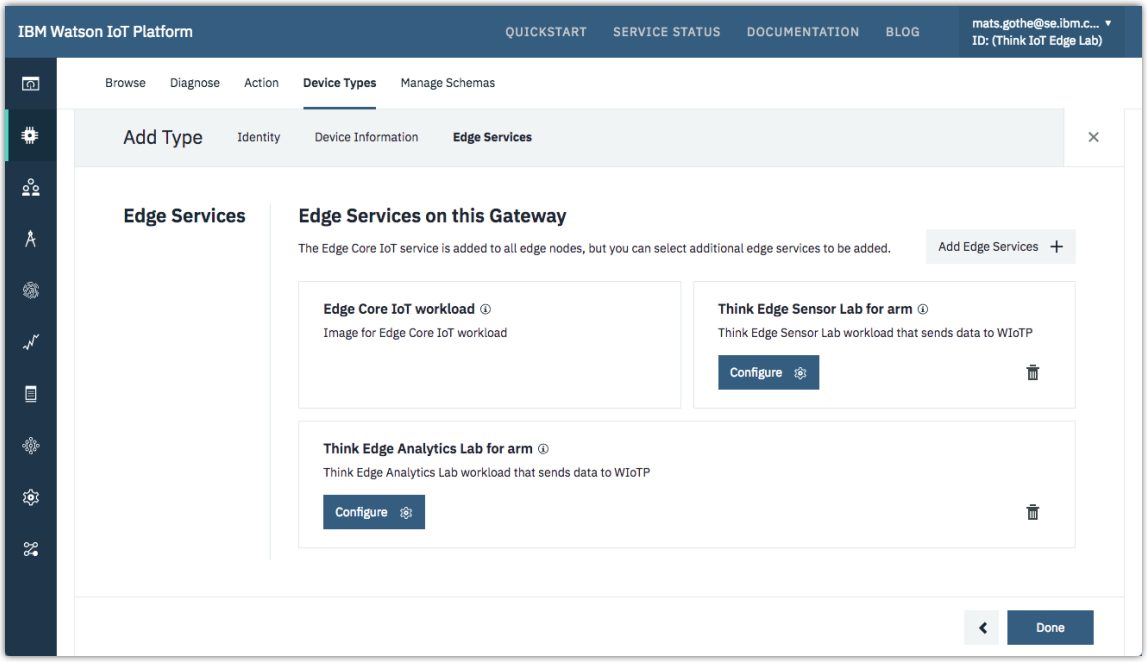
Note: The Edge Core IoT Workload is a mandatory edge service on edge gateways.

7. Click on **Add Edge Service +** to add additional edge services for your edge gateway in this lab.

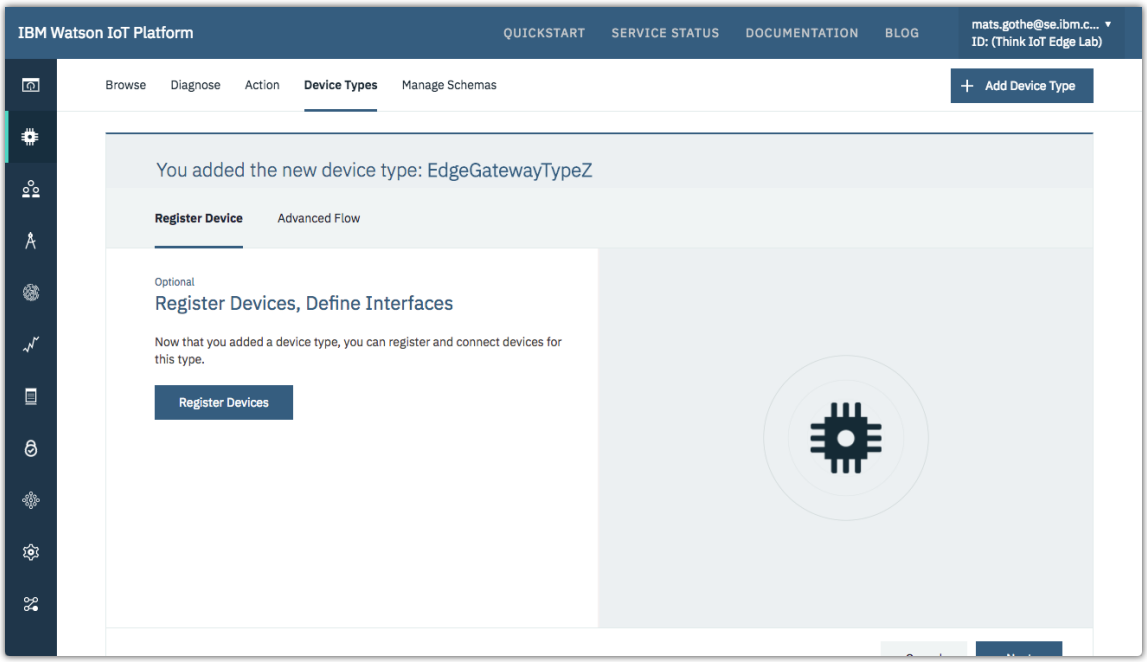
8. In the Add Edge Services dialog, select the **Think Edge Sensor Lab for arm** and **Think Edge Analytics Lab for arm** services.



9. Click **Done**



10. Click **Done** to complete the adding of the gateway type



You have now created the Edge Gateway Type that will be used when creating an Edge Gateway instance in the next section.

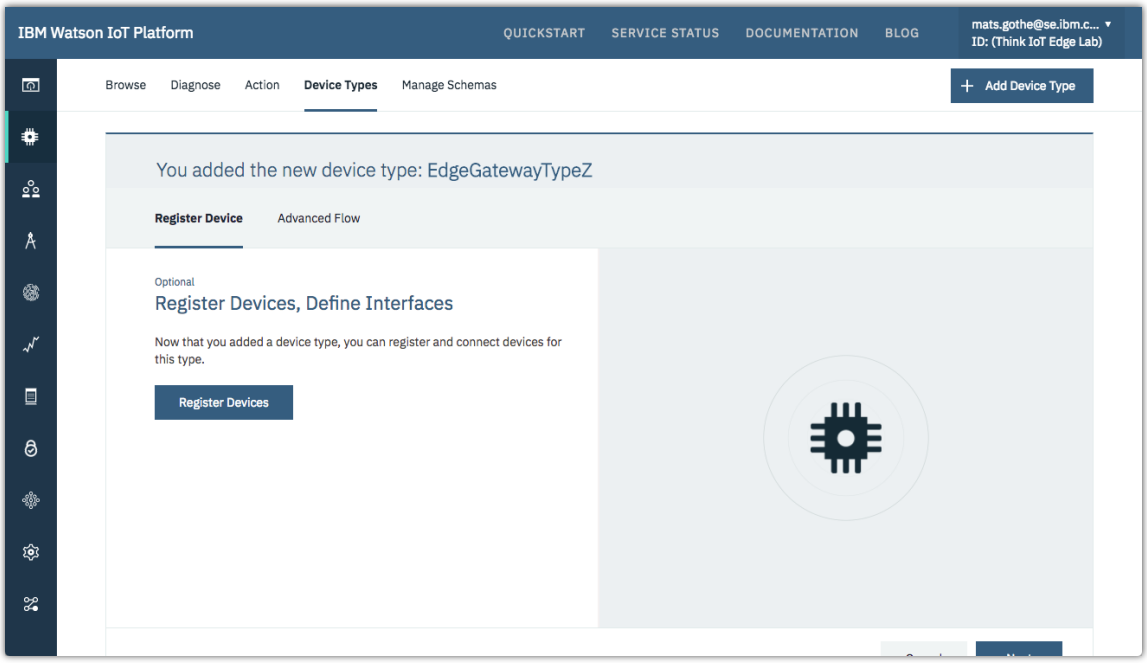
When proceeding to the next step we will use the Register Devices button on the confirmation screen above. Keep this page open for next step.

4.2 Creating a new Edge Gateway

All devices has to be registered prior to connecting to the Watson IoT Platform. Registering a device will declare its identity. A security token will be generated to be used by the device when authenticating to the IoT platform.

To create a new Edge Gateway

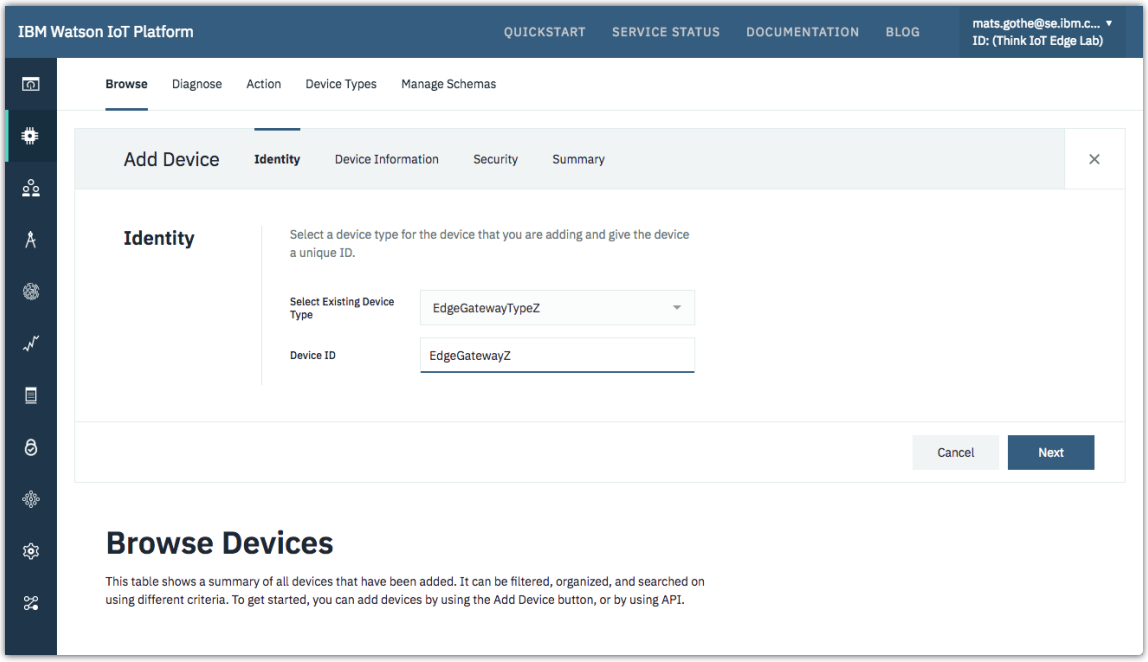
1. On the New Device Type information page below, click **Register Devices** button to add a new gateway.



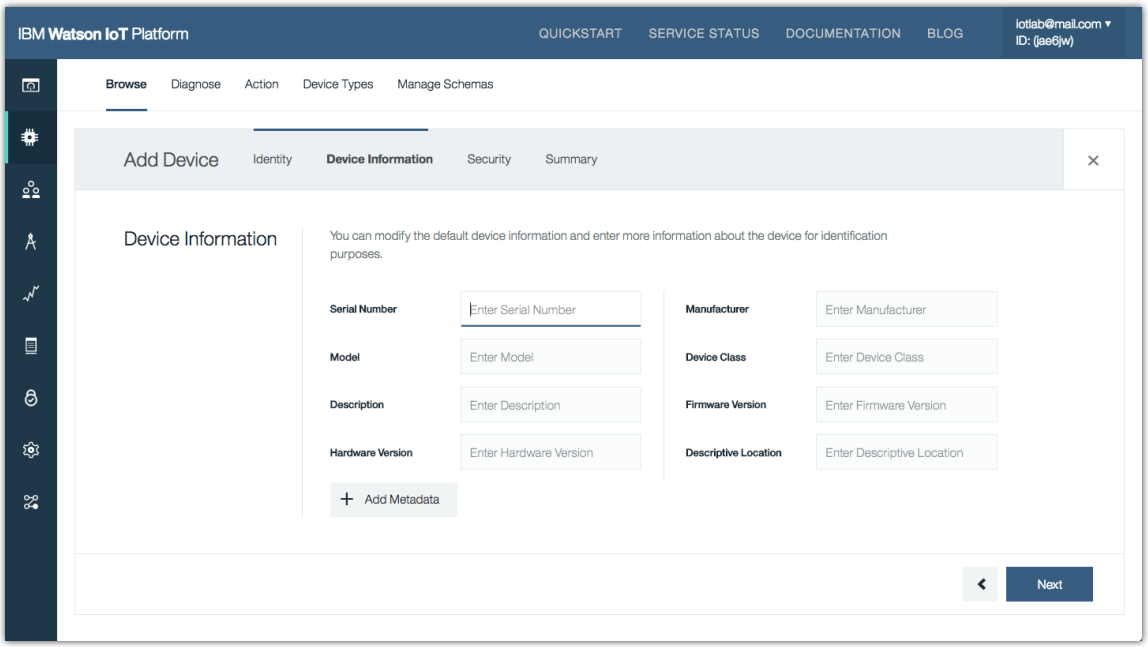
Alternatively: In case the page above was closed, do the following.

On the navigation bar, select **Devices** to open the devices section.
Then click the **+ Add Device** button.

2. On the Add Device page, select the Edge Gateway Type you created in the previous section and enter a unique **Device ID** as the gateway name.
Note: Use the gateway name given for your lab station.

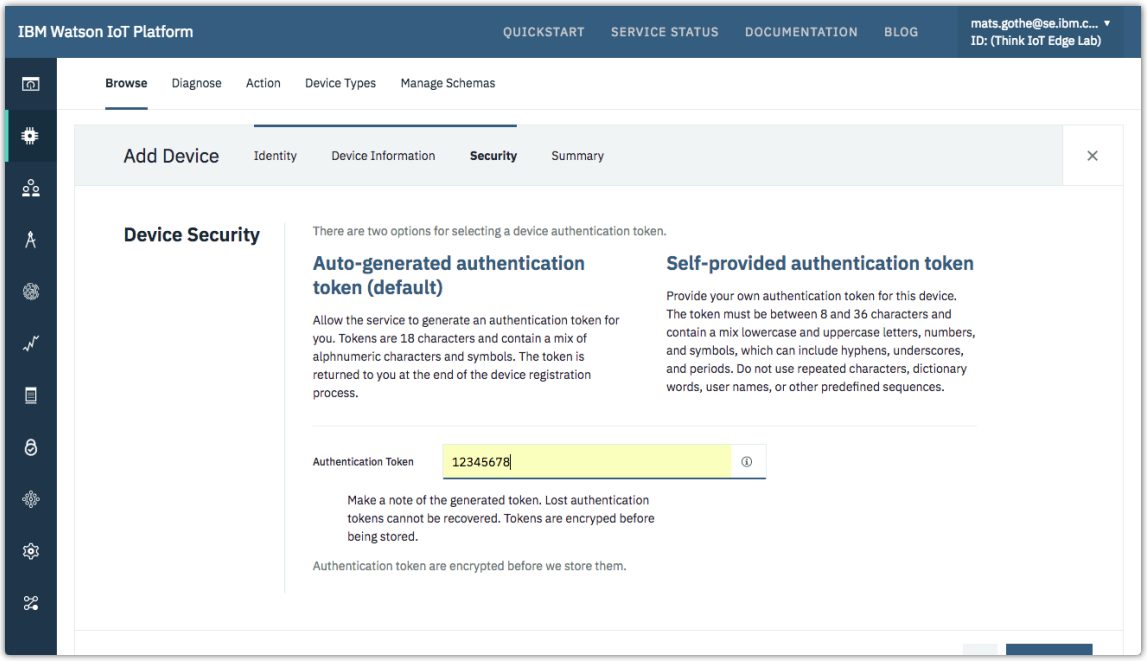


3. Click **Next**. In this lab we will not use this device information.

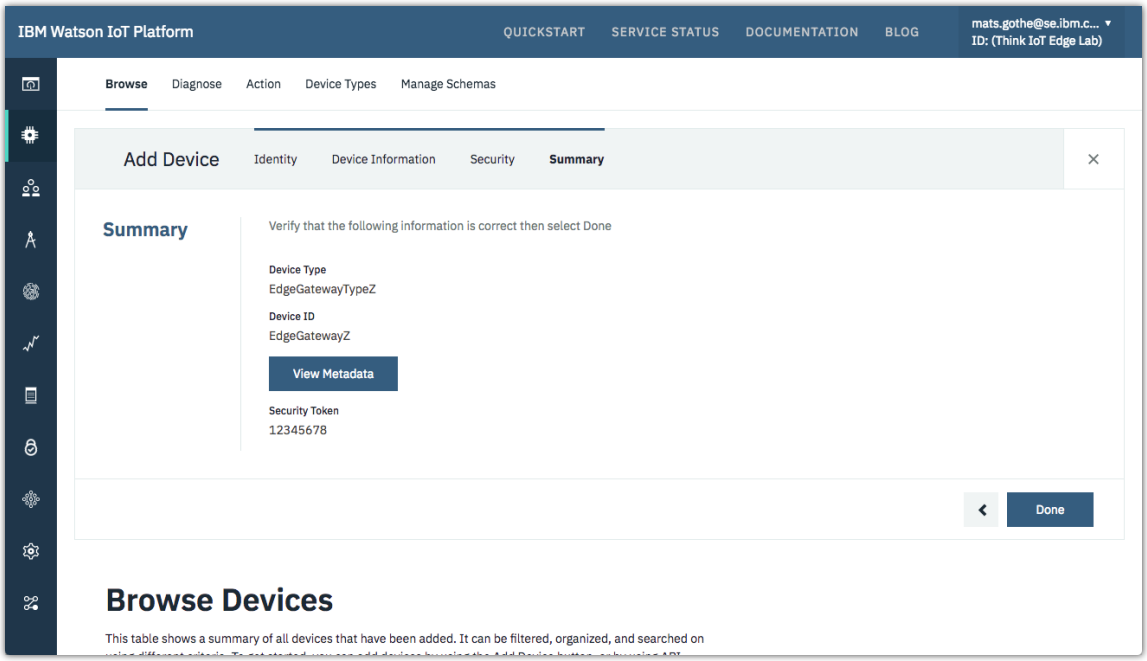


4. Click **Next**. On the Device Security page, enter an 8 digit Authorization Token for this device.

   Note: This token will be used in later steps. Use the token **12345678**. Alternatively, make sure you write down the token entered.

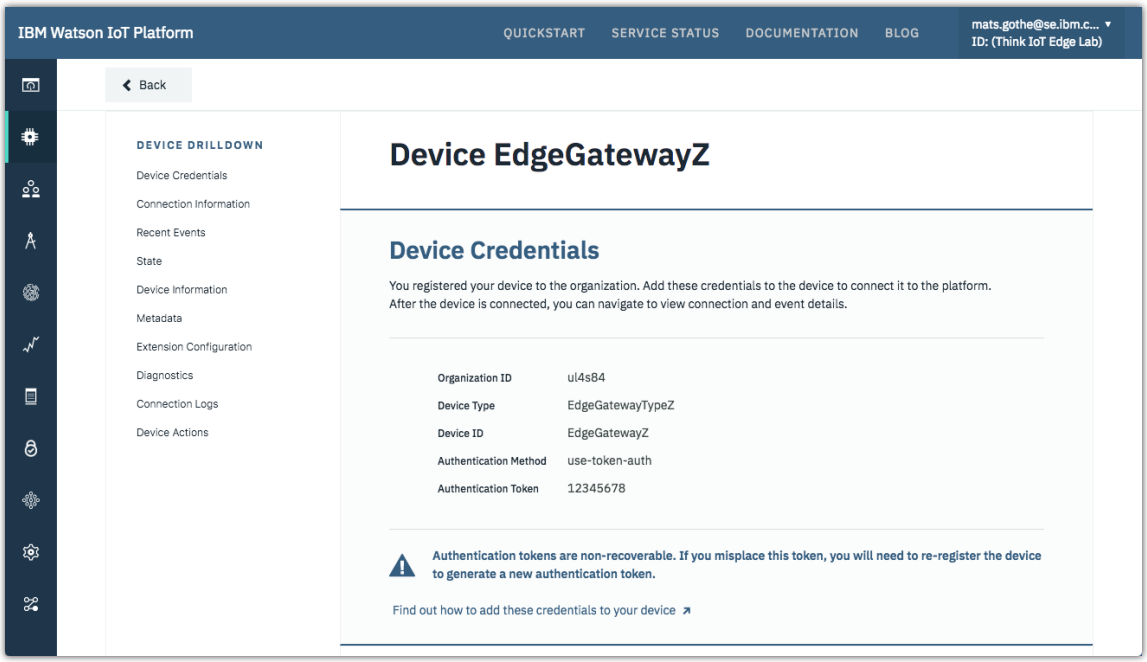


5. Click **Next**. Review the summary information about the gateway you just registered.



6. Click **Done**. The Device drill-down page is opened.

   Note: This page shows device configuration, state and logs. It is a useful page to view device state and diagnose device connection issues.



You have now created the Edge Gateway that will be used to connect sensors and run edge services.

We will now create an Edge Sensor device type and an Edge Sensor device.

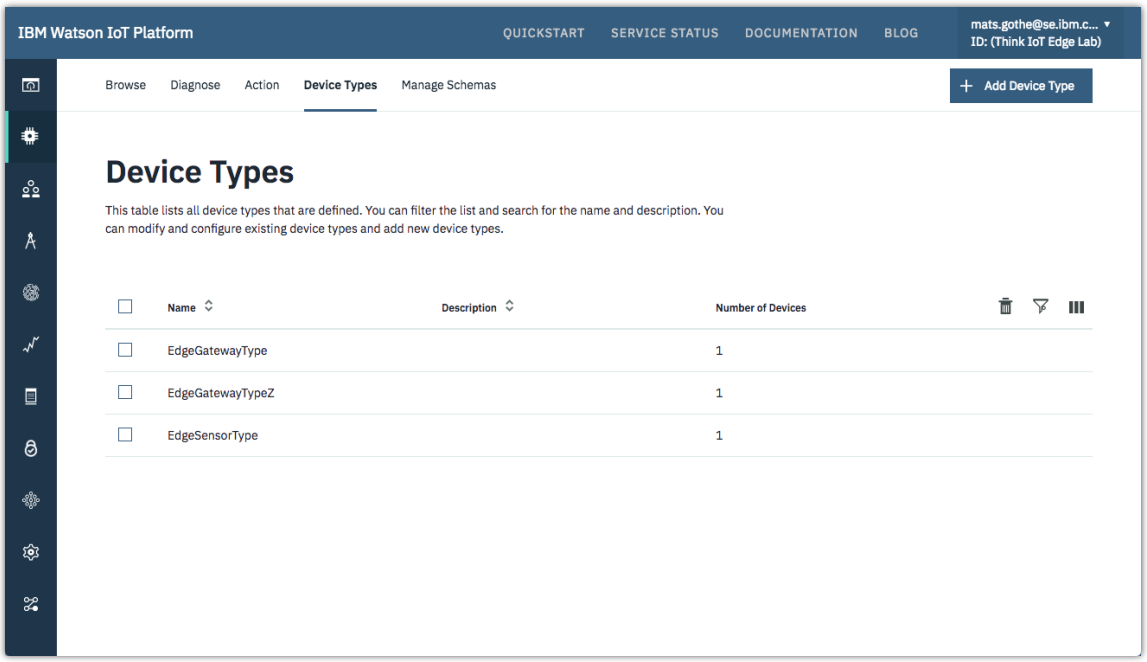
4.3 Creating an Edge Sensor Device

We will now create an edge sensor device type and an edge sensor that will be used to generate the edge sensor data in this lab.

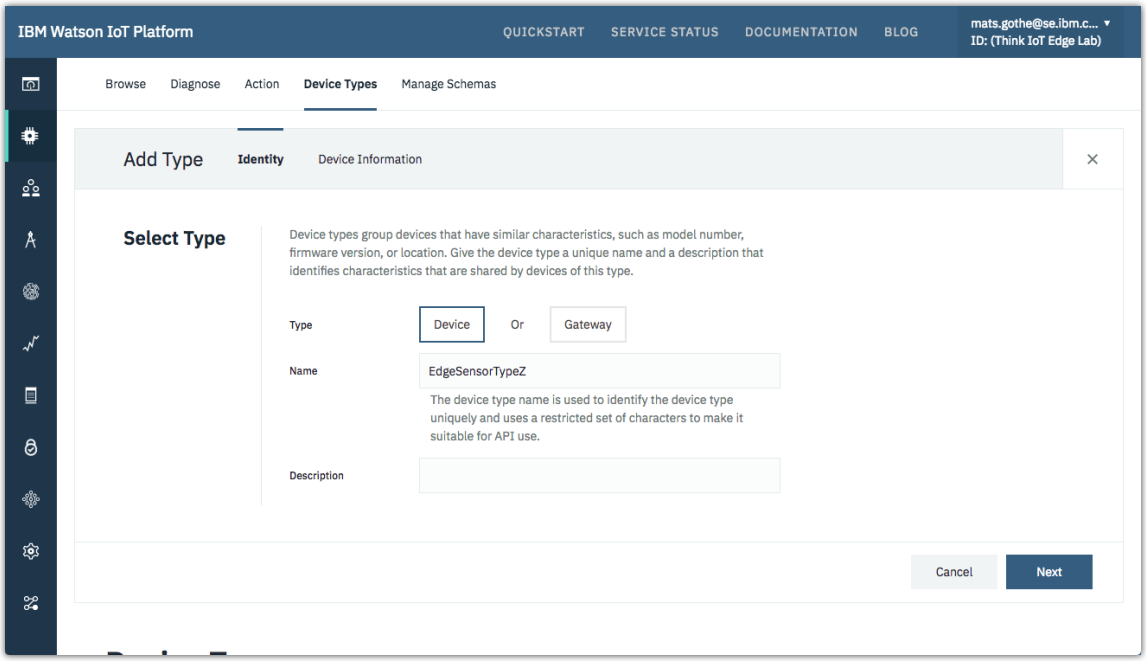
The steps for creating an edge sensor device type and device are similar to the steps we used when creating the edge gateway type and edge gateway. We will provide less detailed instructions in this section. Ask the lab facilitators if you need help.

To create an Edge Sensor device type

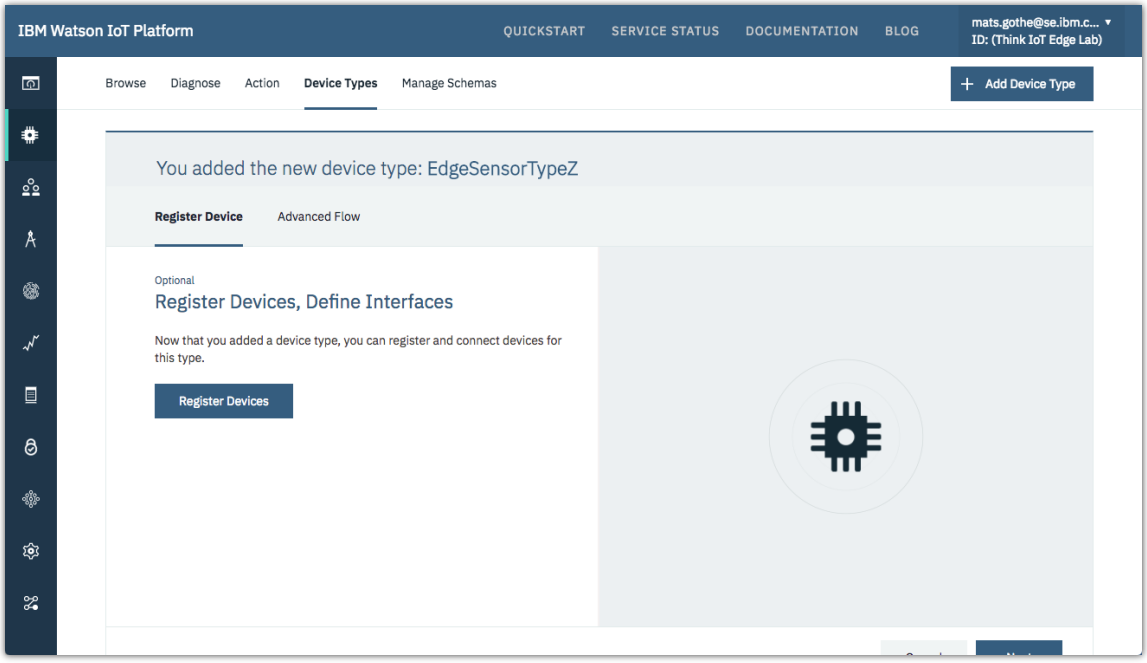
1. On the navigation bar, select the **Devices** section and then the **Device Type** tab



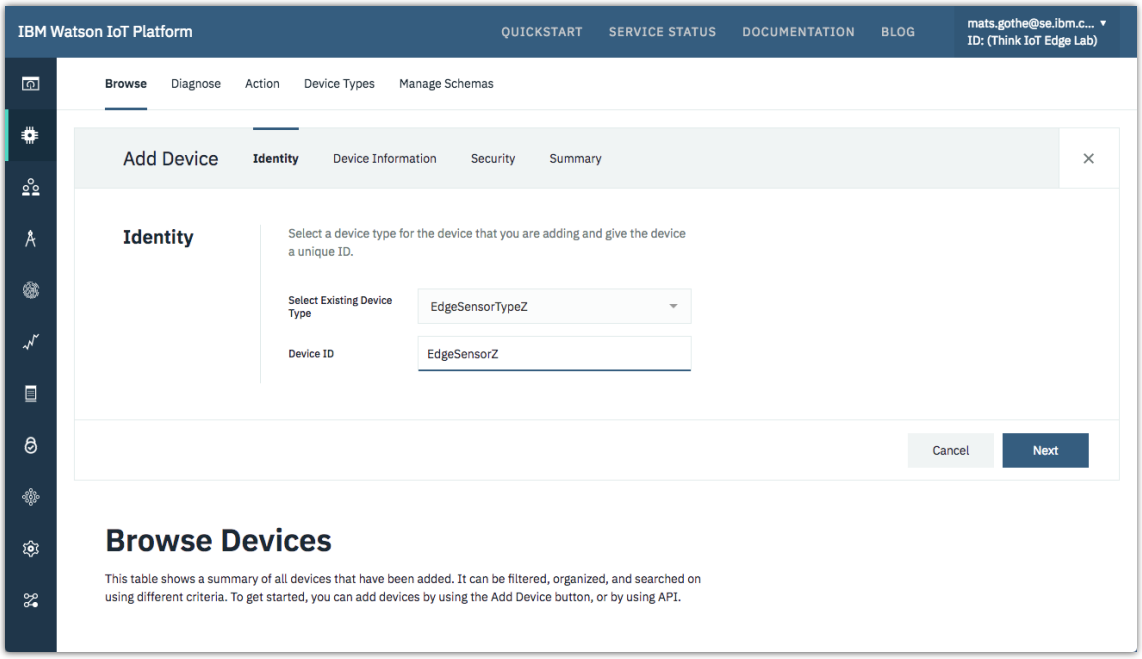
2. Click on the **+ Add Device Type** button
3. Choose the **Device** option
4. Enter a unique **Name**.
   Use the edge sensor type name provided for your workstation.



5. Click **Next** and then click **Done** to finalize the device type definition.
6. Click **Register Devcies** button

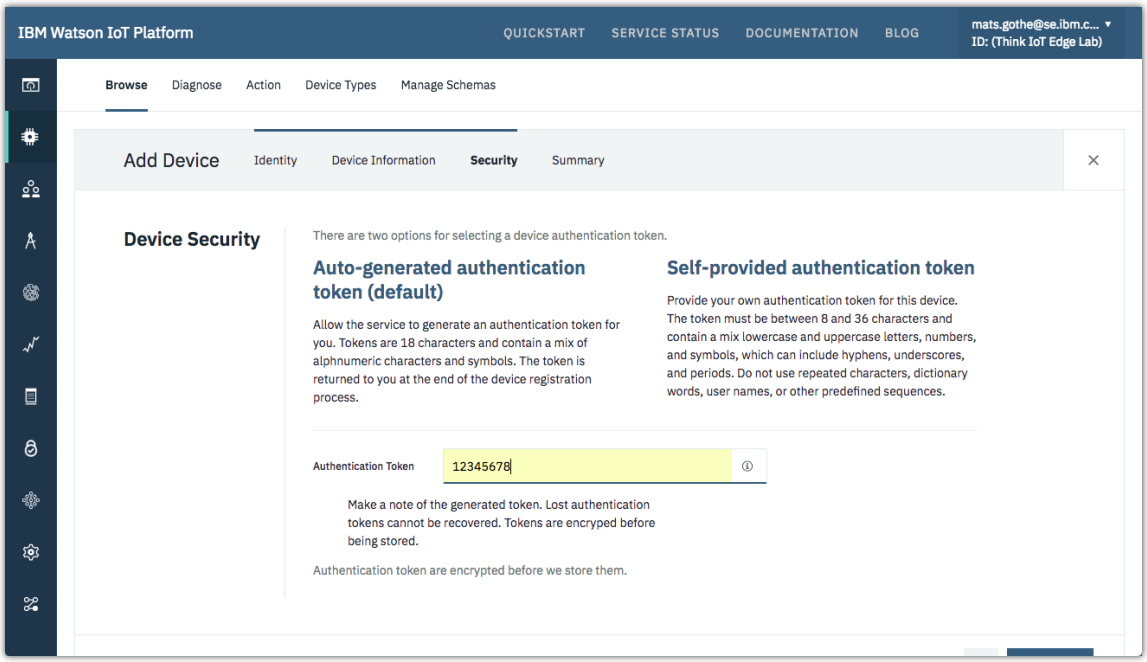


7. On the **Add Device** page, enter a unique **Name**. use the edge sensor devcie name assigned to your workstation.



8. Click **Next** and then click **Next**.

7. On the Devcie Security page, enter a 8 digit token for this device. Use the token **12345678**. Alternatively, make sure you write down the token entered.



8. Click **Next** and then **Done**.

You have now created the Edge Sensor Type and Edge Sensor that will be used to generate sensor values in this lab.

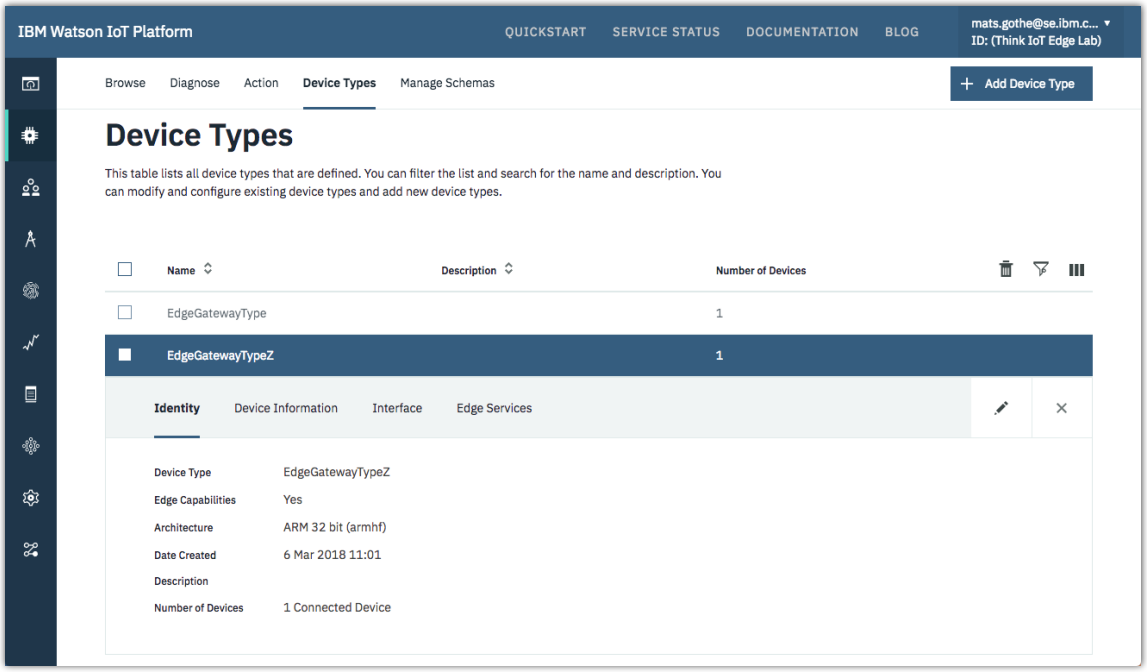
4.4 Finalizing the Edge Gateway Type configuration

In the previous section we created the Edge Sensor Type and the Edge Sensor device. We can now finalize the edge service configuration in the Edge Gateway Type.

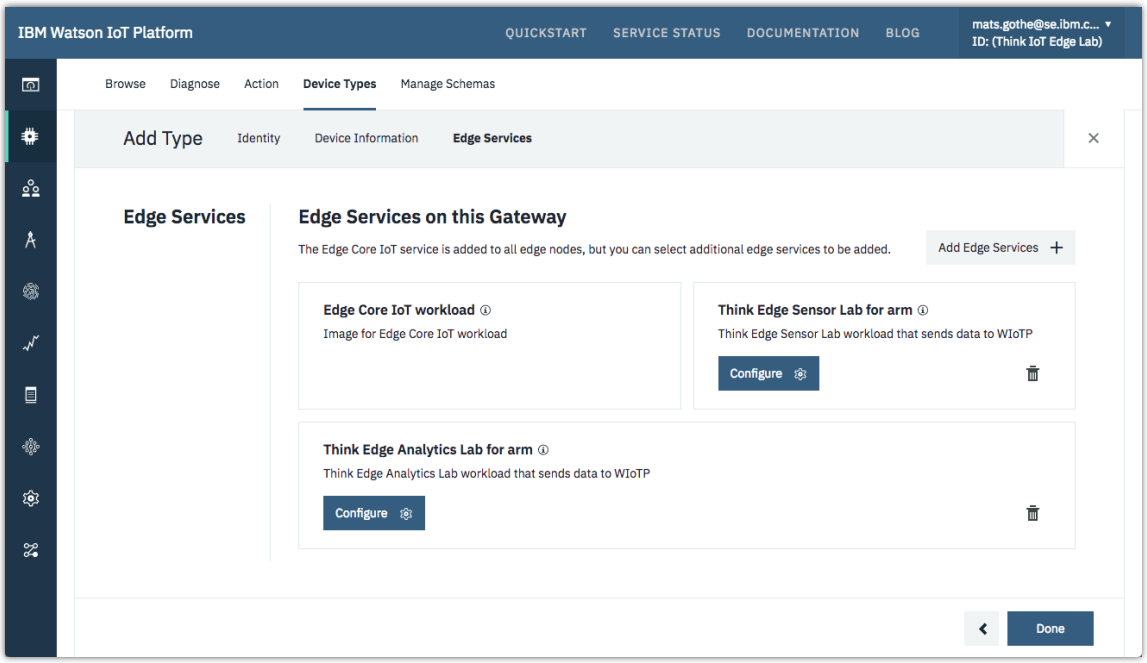
The **Think Edge Sensor Lab for arm** and **Think Edge Analytics Lab for arm** services that we added to the Edge Gateway Type need the names of the Edge Sensor Type and Edge Sensor device to public and subscribe messages. This is achieved by defining environment property values to be set when provisioning the edge services on the edge gateway.

To configure the edge services

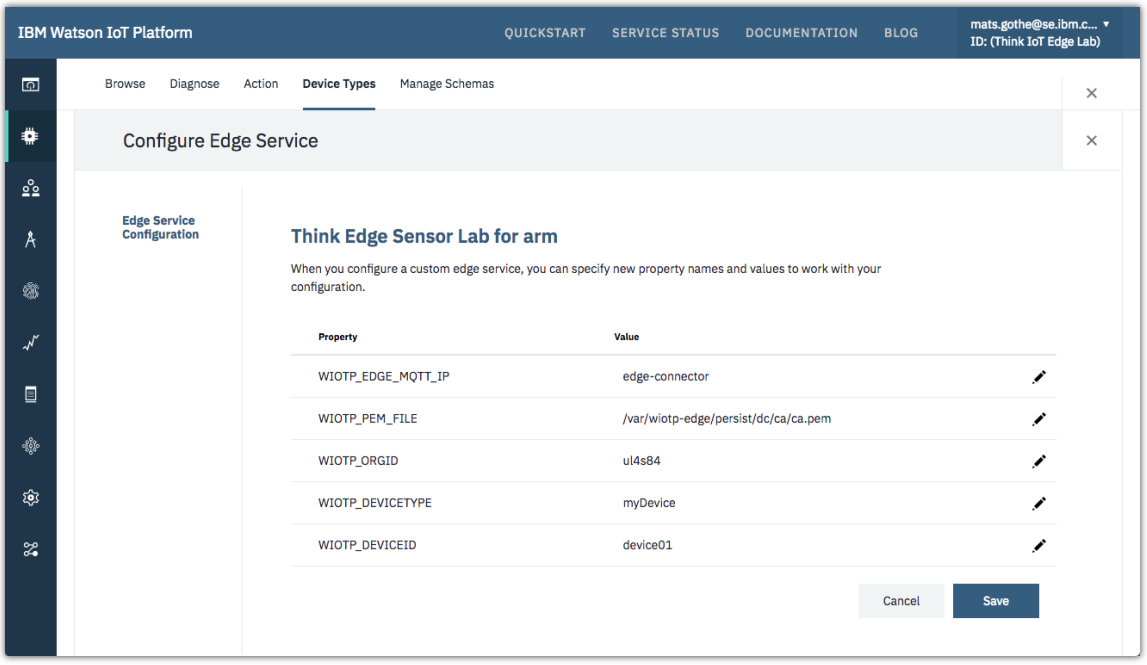
1. On the navigation bar, select **Devices** section and then the **Device Type** tab.
2. Locate and select your Edge Gateway Type



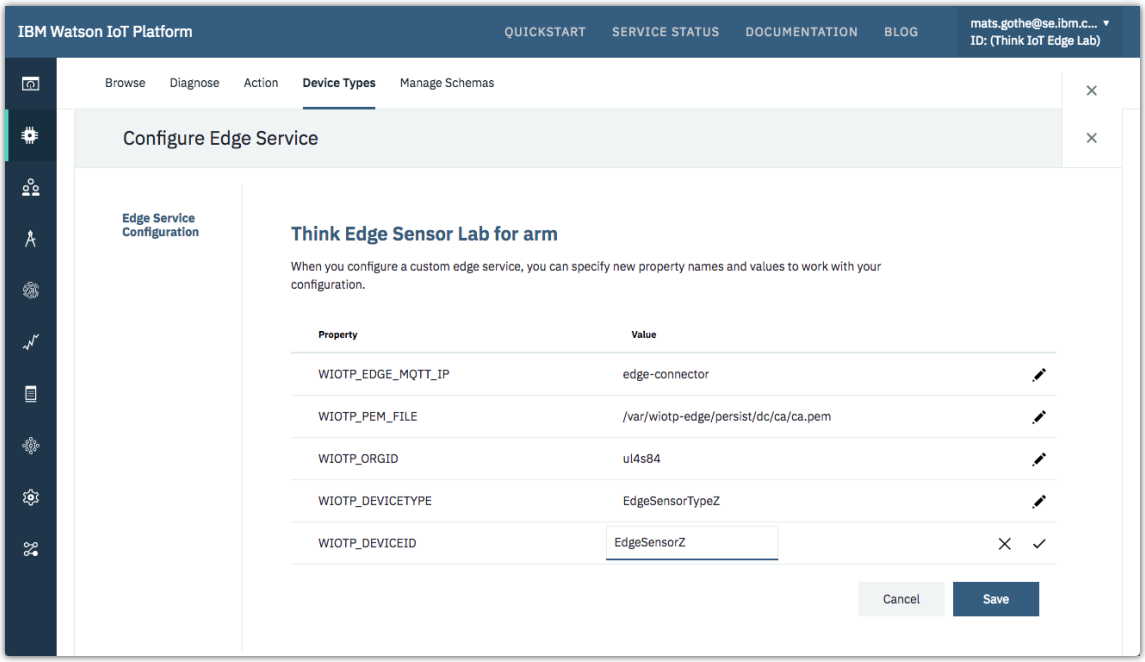
3. Choose the **Edge Services** tab.



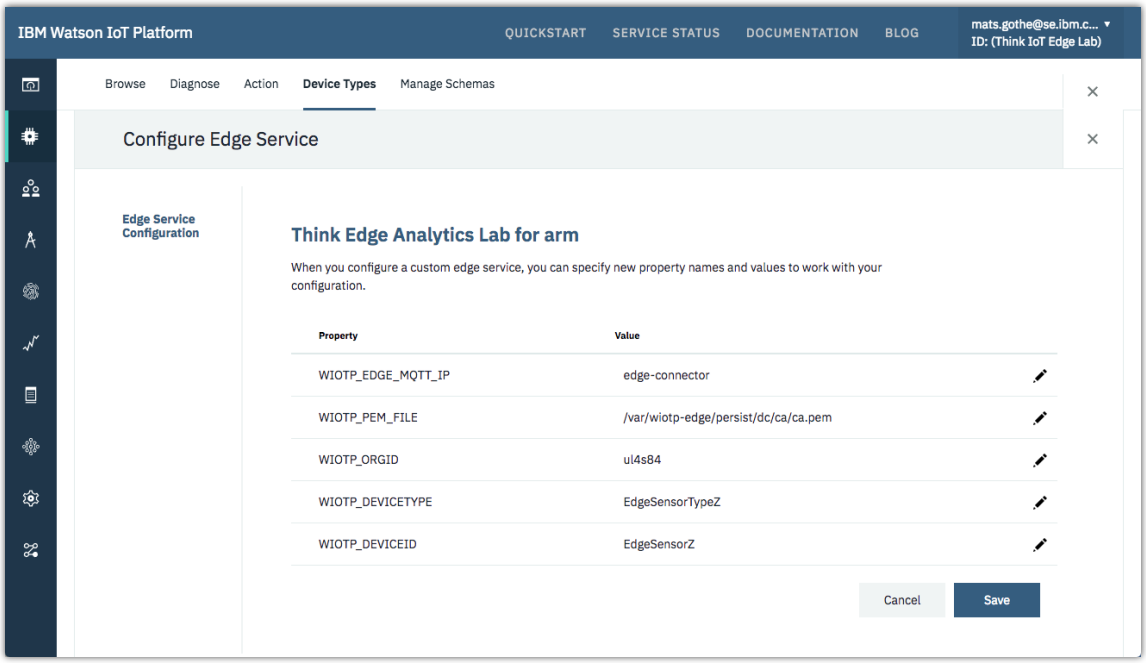
4. Click on the **Configure** button on the **Think Edge Sensor Lab for arm** service.



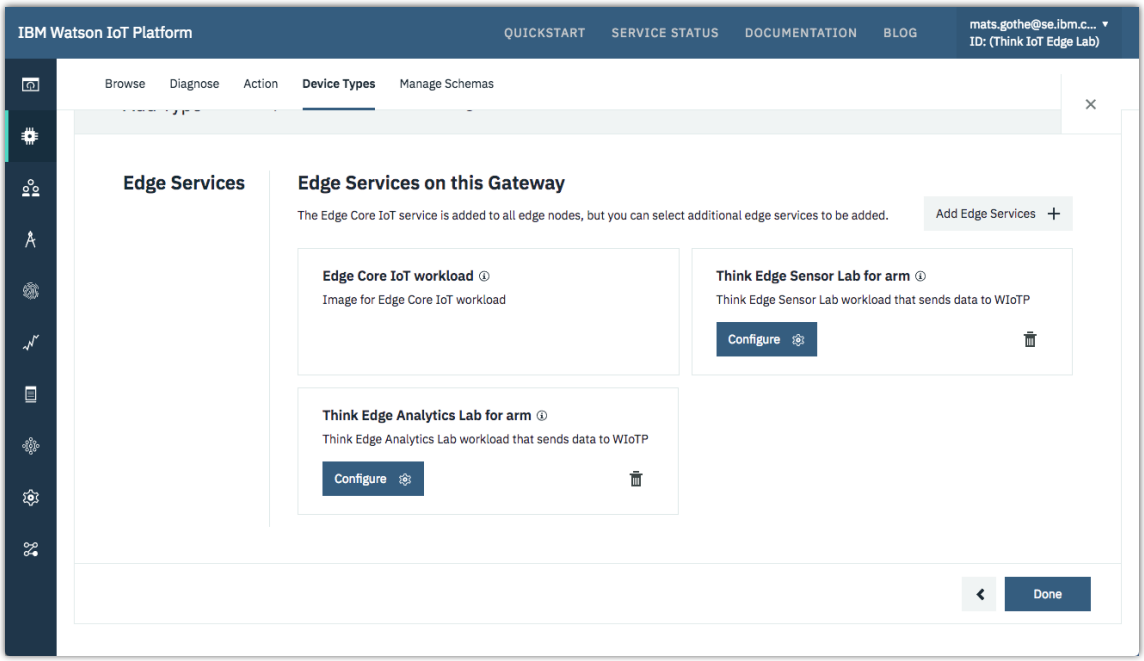
5. On the WIOTP_DEVICETYPE property, click on the pen icon to edit the default value.
6. Enter the name of your Edge Sensor Type device.
   Remember to enter the name and your unique extension.
7. Click the check mark button to confirm your value and exit the editor.



8. Edit the next property, WIOTP_DEVICE, and type in the name of Edge Sensor device. Remember to enter the name with your unique extension.
9. Click **Save**.
10. Proceed and make the exact same edits as above to the **Think Edge Analytics Lab for arm** service.



11. Click **Save** to save the property settings for the **Think Edge Analytics Lab for arm** service.



12. Click Done to complete the final updates on the Edge Gateway Type

We have now completed the configuration of Gateways, Devices and Services in the Watson IoT Platform.

We will now start up the Raspberry Pi and configure it as an Edge Gateway.

5 Configuring the Raspberry Pi / ARM gateway

We will now log into the Raspberry Pi and configure the IoT Edge gateway.
To perform the IoT Edge edge gateway configuration, we need to:

- Power up and boot the Raspberry Pi and the Raspbian OS
- Read the Raspberry Pi IP address on the Sense Hat LED display
- Establish an SSH connection to the Raspberry Pi using the PuTTY program
- Use the PuTTY SSH terminal window to run Linux commands on the Raspberry Pi
- Install the IoT Edge Agent that managed the edge services running on the gateway

Once the IoT Edge Agent is installed, configured and running, it will start the provisioning of the edge services and start the service Docker images.

To complete the edge gateway configuration, we need to:

- Verify the provisioning of the configured Edge Services to the gateway

5.1 Power up the Raspberry Pi

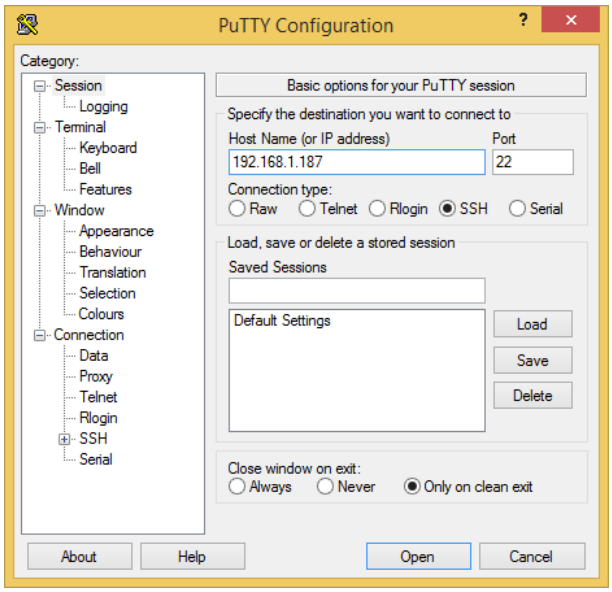
The Raspberry Pi is powered from the workstation using a USB cable. The Raspberry Pi is also connected to the network using a wired network cable.

To log into the Raspberry Pi

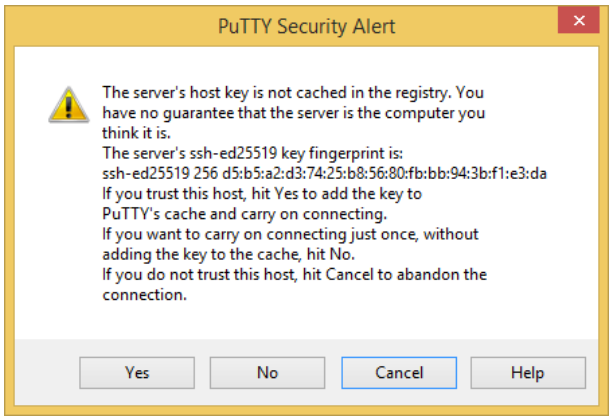
1. Power up the Raspberry Pi by attaching the USB power cable to the workstation
2. Wait a few seconds for the Raspberry Pi to boot.
   View the Raspberry Pi and make a note of the IP address displayed.

   Note: The screen will display the IP address during boot time. Make sure you write down the IP address displayed. If you miss it, just detach and reattach the power to force a reboot.
3. On your Windows workstation, open the File Explorer
4. Start the putty.exe program.

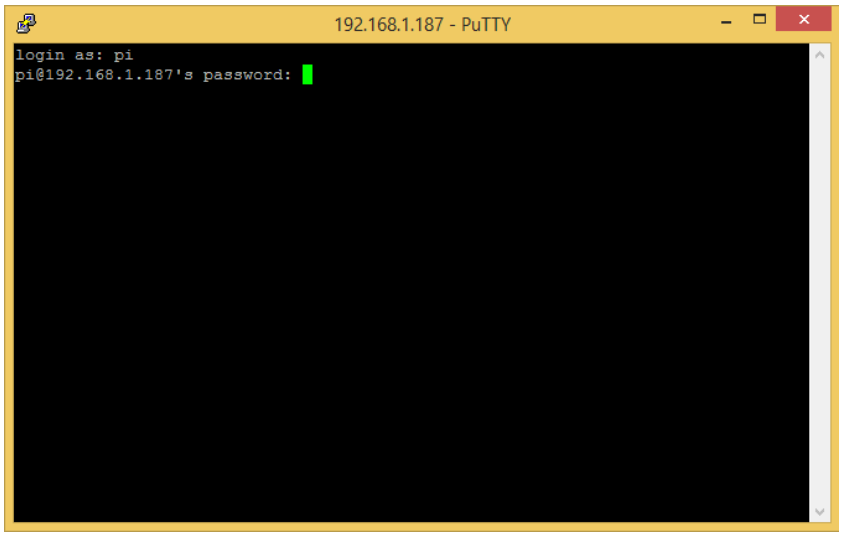
   Note: PuTTY is installed in C:\Program Files\PuTTY



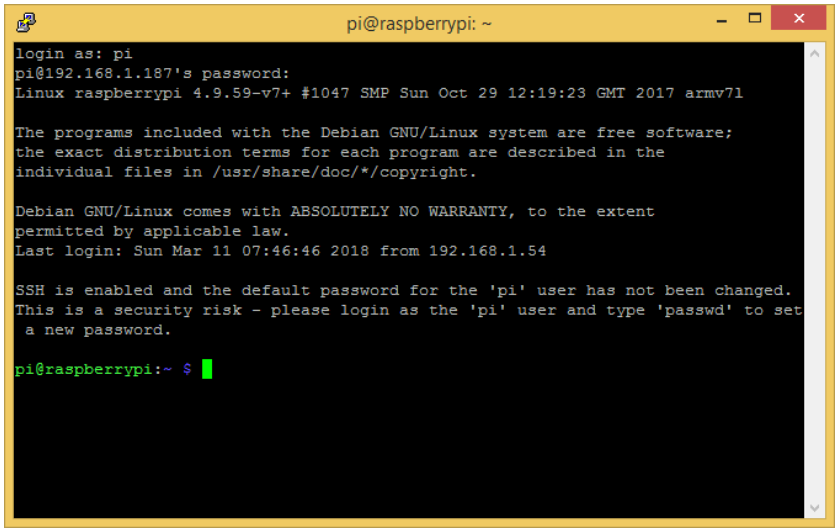
5. Enter the IP address that you Raspberry Pi displayed at boot.
6. Click **Open**
7. In the PuTTY Security Alert dialog, click **Yes**



8. In the PuTTY terminal window, log in using the following account information:
   - Login as: **pi**
   - password: **raspberry**



9. You are now logged into your Raspberry Pi



We have now established a SSH connection to the Raspberry Pi and can start configuring the IoT Edge gateway.

5.2 Installing the IoT Edge Agent

The Raspberry Pi is using an Raspbian OS pre-configured SD-card for this lab.
The following software configurations has been added:

- Raspberry SenseHat Python libraries - https://www.raspberrypi.org/blog/sense-hat-projects/
- Eclipse Paho Python libraries - https://www.eclipse.org/paho/clients/python/
- Docker CE - https://github.com/docker/docker-ce
- The micro editor - https://github.com/zyedidia/micro
- Pure FTP - https://www.pureftpd.org/project/pure-ftpd
- SSH enabled in Raspbian OS

To configure the Edge Gateway

1. Run as root

   ```
   sudo su -
   ```

2. Install the IoT Edge Agent

   ```
   apt-get update && apt-get install -y horizon-wiotp
   ```

3. Edit the following command and update with your gateway configuration details

   ```
   wiotp_agent_setup --org ul4s84 --deviceType EdgeGatewayType --
   deviceId EdgeGateway --deviceToken 12345678 --cloudDisableCertCheck
   true
   ```

   Note: THIS STEP IS REQUIRED TO SUCCESSFULLY CONFIGURING THE EDGE GATEWAY IN THIS LAB. ASK THE LAB FACILITIATORS TO VALIDATE YOUR COMMAND.

   Make sure you update the command parameters, in red above, matching the edge gateway type and edge gateway you have created in the Watson IoT Platform in the previous section 4.

   - --deviceType *your edge gateway type name*
   - --deviceId *your edge gateway id*
   - --deviceToken *your edge gateway token*

4. Run your edited **wiotp_agent_setup** command on your Raspberry Pi command line.

   Wait for the command to complete

You should see the following command reply

```
wiotp_agent_setup --org ul4s84 --deviceType EdgeGatewayType --deviceId EdgeGateway --
deviceToken 12345678 --cloudDisableCertCheck true
WIoTP Horizon agent setup start.
Generating Edge internal certificates ...
Reading input file /etc/wiotp-edge/hznEdgeCoreIoTInput.json...
Horizon Exchange base URL:
https://ul4s84.internetofthings.ibmcloud.com/api/v0002/edgenode
Node ul4s84/g@EdgeGatewayType@EdgeGateway exists in the exchange
Initializing the Horizon node...
Setting global variables...
Setting microservice variables...
Setting workload variables...
Changing Horizon state to configured to register this node with Horizon...
Horizon node is registered. Workload agreement negotiation should begin shortly. Run
'hzn agreement list' to view.
Agent registration complete.
```

Note: If you get the command reply to

```
*** anax.json (Y/I/N/O/D/Z) [default=N] ?
```

Reply with **N** and press **return**

After the wiotp_agent_setup command has completed, the agreement negotiation and docker image provisioning starts between the IoT Edge Agent and the IoT Platform Service Registry.

5. To validate that an agreement has been made, run the command

   ```
   hzn agreement list
   ```

   You should see the following command reply listing the agreements for the three edge services we selected for the edge gateway.


```
hzn agreement list
[
  {
    "name": "Policy for edge-core-iot-microservice merged with
EdgeGatewayType_internetofthings.ibmcloud.com-wiotp-edge-workloads-edge-core-iot-
workload_IBM_arm",
    "current_agreement_id":
"047598d9ba0ec2ff5aee982016d743e84f98826b4315d0a9455df0900bc81521",
    "consumer_id": "IBM/wiotp-agbot-1",
    "agreement_creation_time": "2018-03-12 13:16:50 -0700 PDT",
    "agreement_accepted_time": "2018-03-12 13:17:07 -0700 PDT",
    "agreement_finalized_time": "2018-03-12 13:17:09 -0700 PDT",
    "agreement_execution_start_time": "2018-03-12 13:18:10 -0700 PDT",
    "agreement_data_received_time": "",
    "agreement_protocol": "Basic",
    "workload_to_run": {
      "url": "https://internetofthings.ibmcloud.com/wiotp-edge/workloads/edge-core-
iot-workload",
      "org": "IBM",
      "version": "2.3.1",
      "arch": "arm"
    }
  },
  {
    "name": "Policy for edge-core-iot-microservice merged with
EdgeGatewayType_internetofthings.ibmcloud.com-workloads-edgelab-analytics_ul4s84_arm",
    "current_agreement_id":
"78051fb9e71561b944a1f5b502109292d30db091ab590694debc7d6b77f16e38",
    "consumer_id": "IBM/wiotp-agbot-1",
    "agreement_creation_time": "2018-03-12 13:16:54 -0700 PDT",
    "agreement_accepted_time": "2018-03-12 13:17:12 -0700 PDT",
    "agreement_finalized_time": "2018-03-12 13:17:14 -0700 PDT",
```

```
    "agreement_execution_start_time": "2018-03-12 13:18:27 -0700 PDT",
    "agreement_data_received_time": "",
    "agreement_protocol": "Basic",
    "workload_to_run": {
      "url": "https://internetofthings.ibmcloud.com/workloads/edgelab-analytics",
      "org": "ul4s84",
      "version": "1.0.0",
      "arch": "arm"
    }
  },
  {
    "name": "Policy for edge-core-iot-microservice merged with
EdgeGatewayType_internetofthings.ibmcloud.com-workloads-edgelab-sensor_ul4s84_arm",
    "current_agreement_id":
"ddaa8f913e6457effc8e1ecd2b82b8b5fbbdaed6e8e889d974bc8f0e52d41716",
    "consumer_id": "IBM/wiotp-agbot-1",
    "agreement_creation_time": "2018-03-12 13:16:57 -0700 PDT",
    "agreement_accepted_time": "2018-03-12 13:17:16 -0700 PDT",
    "agreement_finalized_time": "2018-03-12 13:17:19 -0700 PDT",
    "agreement_execution_start_time": "",
    "agreement_data_received_time": "",
    "agreement_protocol": "Basic",
    "workload_to_run": {
      "url": "https://internetofthings.ibmcloud.com/workloads/edgelab-sensor",
      "org": "ul4s84",
      "version": "1.0.0",
      "arch": "arm"
    }
  }
]
```


6. To validate that the edge service Docker images has been downloaded, run the command

   `docker images`

   Note: There may take a few minutes to download the images to your gateway.

   You should see the following command reply


```
docker images
REPOSITORY                                          TAG                 IMAGE ID
CREATED             SIZE
wiotp-connect/edge/armhf/edge-connector             2.4.1               17f451a5e04e
4 days ago          17MB
wiotp-infomgmt/edge/armhf/edge-im                   1.0.15              bdb5bc447c65
5 days ago          48.5MB
ibmiottest/workload_arm_edgelab-sensor              1.0.0               4829111aefb2
5 days ago          392MB
ibmiottest/workload_arm_edgelab-analytics           1.0.0               a80c875178be
5 days ago          392MB
wiotp-connect/edge/armhf/edge-mqttbroker            1.1.3               d1c9d48ff6eb
2 weeks ago         19MB
wiotp-infomgmt/edge/armhf/edge-im                   1.0.12              10bb4d9657d6
2 weeks ago         48.5MB
wiotp-connect/edge/armhf/edge-connector             2.3.2               58cfe54a44fd
4 weeks ago         21.4MB
wiotp-connect/edge/armhf/edge-core-iot-workload     1.0.3               7dfb00cbc477
3 months ago        12.8MB
armhf/ubuntu                                        latest              fa40ea71de37
15 months ago       106MB
```


7. To validate that the edge service Docker containers has been started, run the command

   `docker ps`

You should see the following command reply

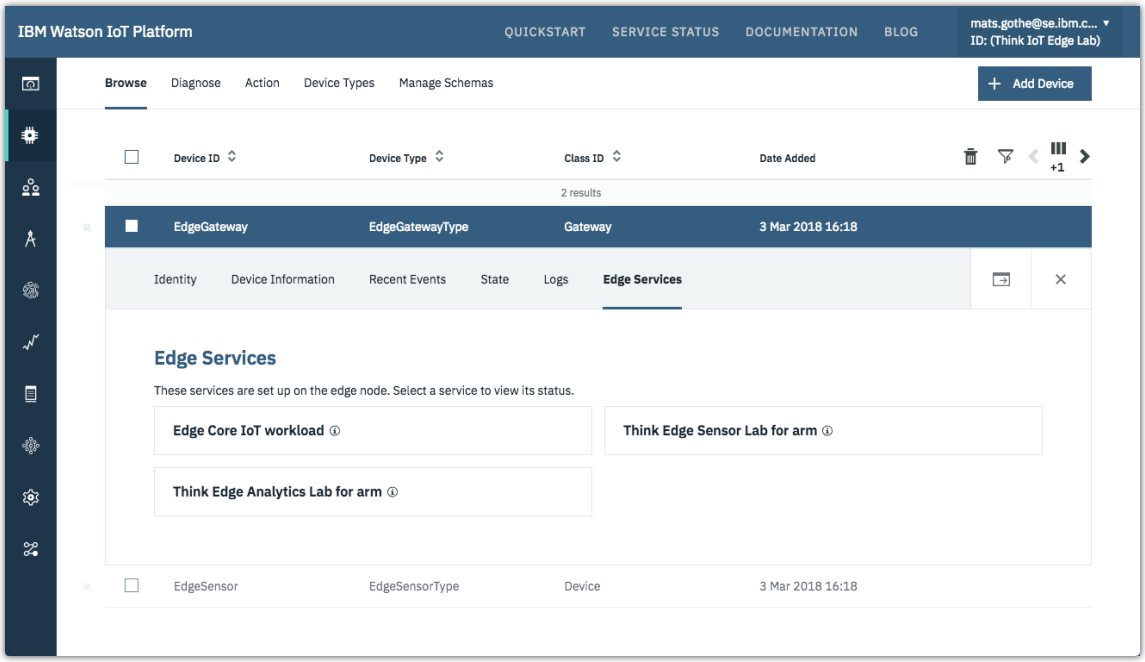
```
docker ps
CONTAINER ID        IMAGE                                                          COMMAND
CREATED             STATUS                              PORTS
NAMES
6861c473c4fd        ibmiottest/workload_arm_edgelab-sensor:1.0.0           "python
/edgelab-sen…"   2 minutes ago       Restarting (1) 44 seconds ago
ddaa8f913e6457effc8e1ecd2b82b8b5fbbdaed6e8e889d974bc8f0e52d41716-edgelab-sensor
97cdb6ab7a3c        ibmiottest/workload_arm_edgelab-analytics:1.0.0        "python
/edgelab-ana…"   3 minutes ago       Up 3 minutes
78051fb9e71561b944a1f5b502109292d30db091ab590694debc7d6b77f16e38-edgelab-analytics
adb4092380fd        wiotp-connect/edge/armhf/edge-core-iot-workload:1.0.3
"/start.sh"              3 minutes ago       Up 3 minutes
047598d9ba0ec2ff5aee982016d743e84f98826b4315d0a9455df0900bc81521-edge-core-iot-
workload
506b89d3c746        wiotp-infomgmt/edge/armhf/edge-im:1.0.15
"/start.sh"              3 minutes ago       Up 3 minutes
internetofthings.ibmcloud.com-wiotp-edge-microservices-edge-core-iot-
microservice_2.4.0_66163459-7692-48b8-8243-1ff8b83adc69-edge-im
911b1c90eda7        wiotp-connect/edge/armhf/edge-connector:2.4.1
"/start.sh"              3 minutes ago       Up 3 minutes
0.0.0.0:1883->1883/tcp, 0.0.0.0:8883->8883/tcp   internetofthings.ibmcloud.com-wiotp-
edge-microservices-edge-core-iot-microservice_2.4.0_66163459-7692-48b8-8243-
1ff8b83adc69-edge-connector
cd06abe0b875        wiotp-connect/edge/armhf/edge-mqttbroker:1.1.3
"/start.sh"              3 minutes ago       Up 3 minutes
internetofthings.ibmcloud.com-wiotp-edge-microservices-edge-core-iot-
microservice_2.4.0_66163459-7692-48b8-8243-1ff8b83adc69-edge-mqttbroker
a0e5a3aea10e        ibmiottest/workload_arm_edgelab-sensor:1.0.0           "python
/edgelab-sen…"   About an hour ago   Restarting (1) About an hour ago
51836e0ff948da8372773d0068325e38e84c5c08433864a932993079adbed125-edgelab-sensor
```

5.3 Monitoring the Edge Gateway status

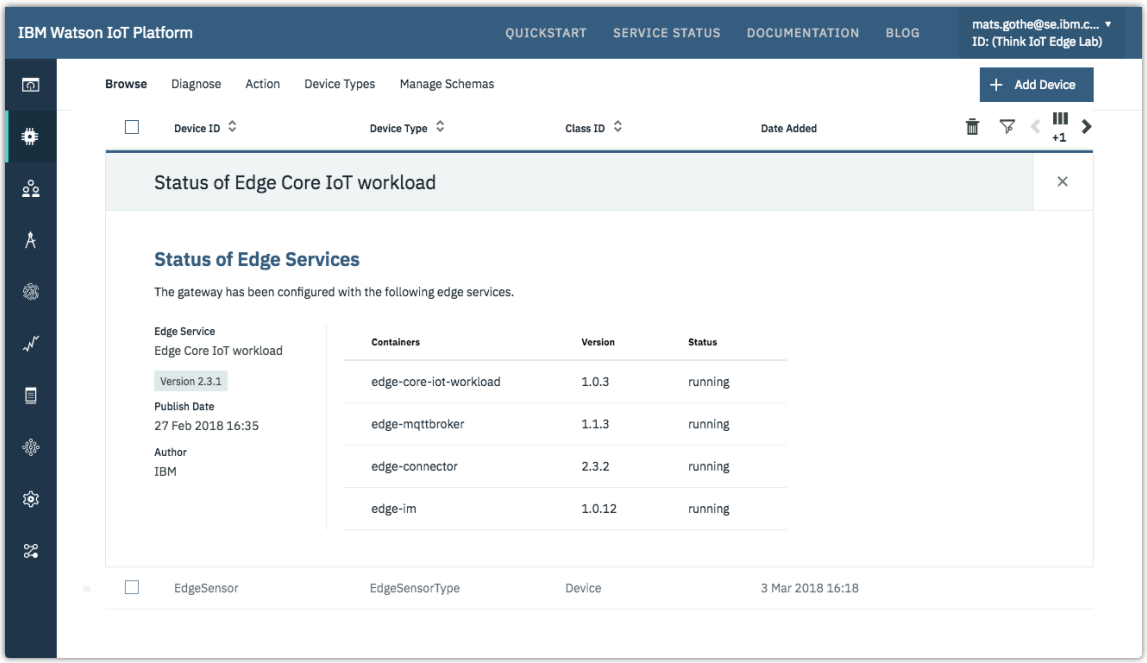
You can also monitor the state of the IoT Edge services and Docker containers running on an edge gateway using the Watson IoT Platform.

To view the services running on an Edge Gateway

1. On the navigation bar, select **Devices** to navigate to the list of gateways and devices.
2. Find, select and expand your Edge Gateway in the list.



3. Select the **Edge Services** tab
4. Select one of the edge services, for example the **Edge Core IoT** service



5. View the state of each Docker container in the service

   Note: All Docker containers should be in state **running**
6. Explore the state of the other edge services on your Edge Gateway

Once you have confirmed that all services have started on the edge gateway you are ready to proceed to next step and start monitoring the device data sent from the gateway to the Watson IoT Platform.

6 Working with IoT Edge Data Management workloads

6.1 Viewing Edge Sensor Data

The Raspberry Pi used in this lab is connected to a Sense Hat board that has multiple sensors integrated to the board.

- Temperature
- Humidity
- Pressure
- 3-axis gyro
- 3-axes accelerometer

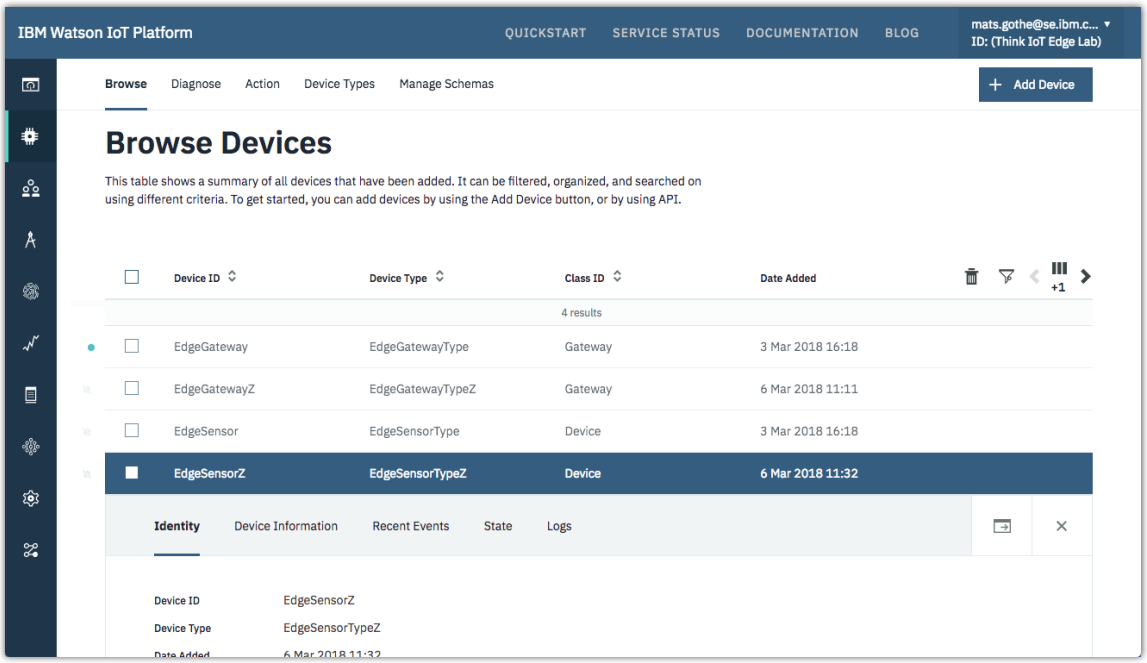
The **Think Edge Sensor Lab for arm** service in your Edge Gateway service configuration is running a edgelab-sensors Docker container that publishes sensor readings.

The sensor data is published to three event topics

- An environment event **env** on the topic **iot-2/evt/env/fmt/json**
- A motion event **mov** on the topic **iot-2/evt/mov/fmt/json**
- An acceleration event **acc** on the topic **iot-2/evt/acc/fmt/json**

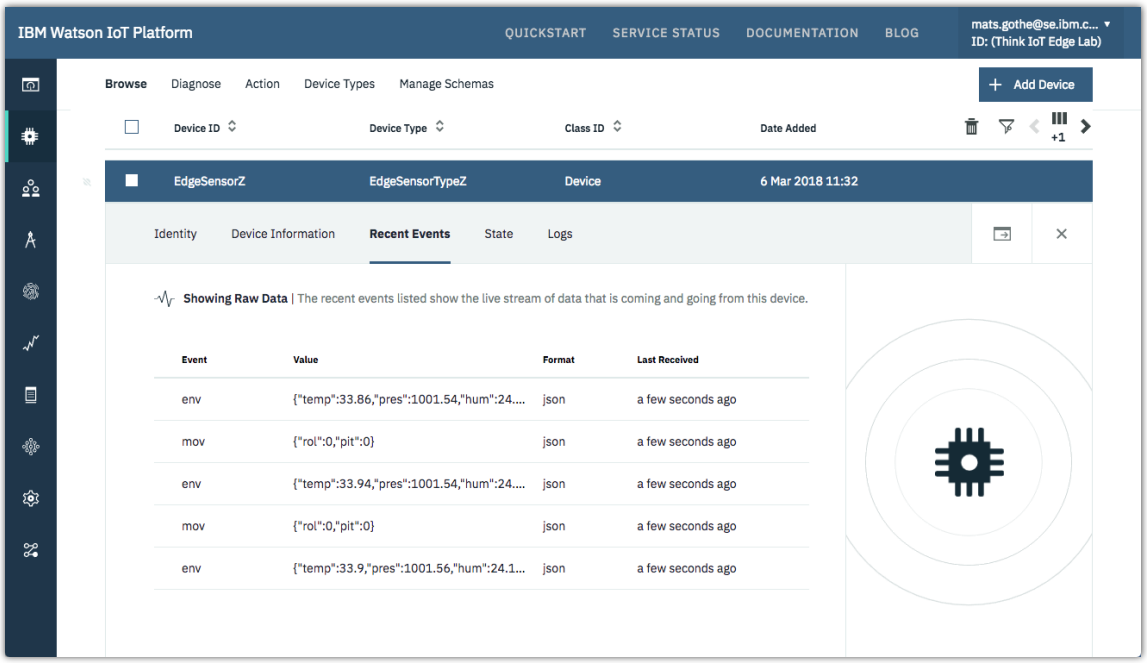
To view device data received on the IoT platform

1. On the navigation bar, select **Devices** to navigate to the list of gateways and devices.
2. Find, select and expand your Edge Sensor device in the list.



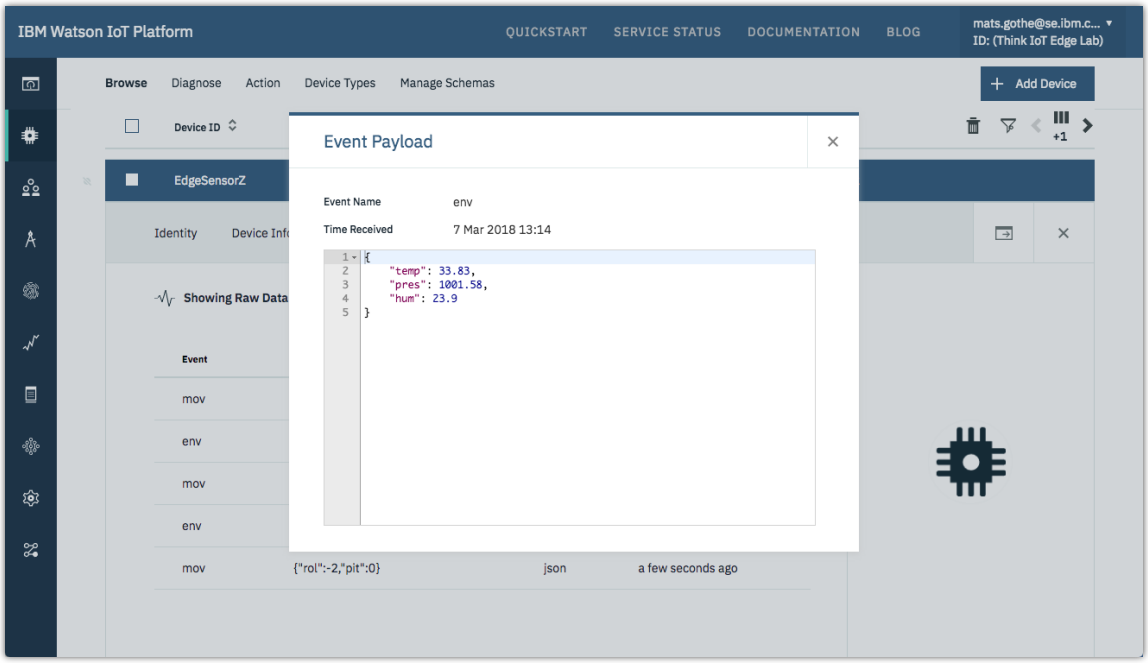
3. Select the **Recent Events** tab. Wait for events to be received from the edge sensors. Events should be published every 15th second.

   Note: The three event types should be showing in the list; **env**, **mov** and **acc**



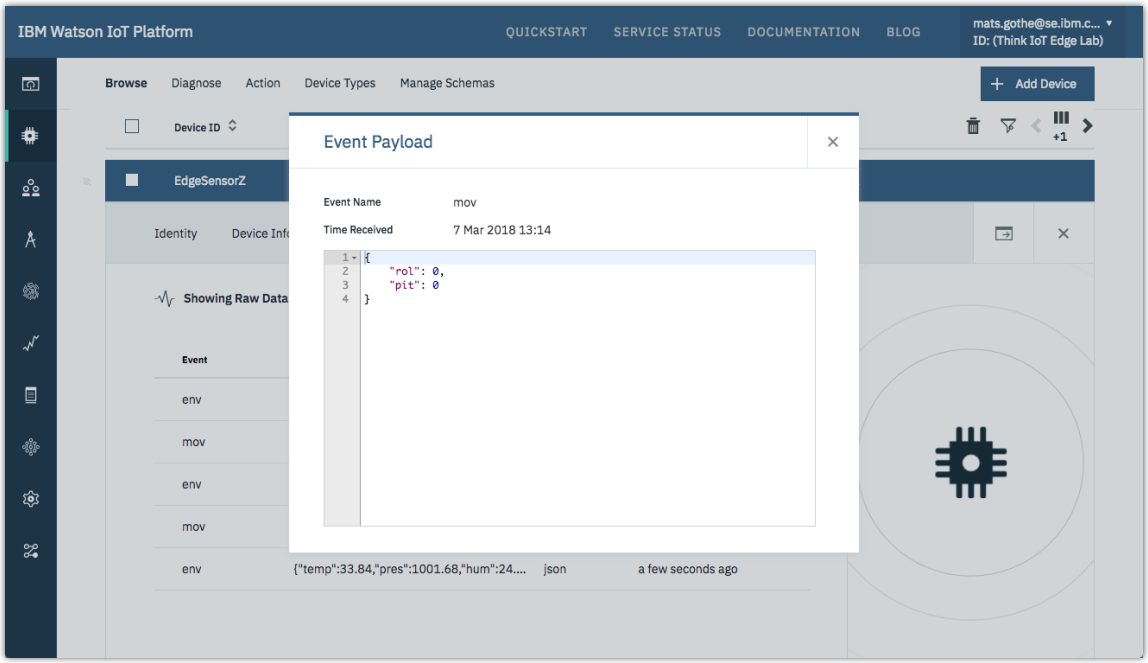
4. Click on one of the event, for example an **env** event.

   Note: The **env** environment event is sending **temp**, **pres** and **hum** data values in the event payload. The data is formatted in JSON.



5. Select and view the **mov** motion event.

   Note: The event is sending **pit** (x-axis pitch) and **rol** (y-axis roll) data values in the event payload.



6. Select and view the **acc** acceperation event.

   Note: The event is sending acceleration data values in the event payload. The acc value is the amplitude of the acceleration vector.

6.2 Filtering and Transforming Edge Sensor Data

IoT devices generally publish sensor data at a regular heartbeat. The frequency depends on the IoT analytics and control use-cases of the application, like every second, every minute, every hour, or once a day.

The network infrastructure may be sensitive to too much data being published by thousands or millions of devices to the cloud. There is a desire to reduce the data sent by the cloud by

- filtering sensor data that is not required by the IoT cloud application
- lowering the frequency of sensor values
- only send changing sensor values
- compute the state of the device at the edge and only send state updates

The IoT platform provides a Data Management capability that allows developers to define abstractions and transformations that filter, abstract and transform event data. We will refer to such transformation as workloads.

The IoT Edge capability provides distribution of Data Management workloads to the edge. That enables filtering and transformation of event data to happen on the edge, without any

need to have data transferred to the cloud. It also enables the compute power of an edge gateway to perform any transformation directly at the source of the data at the edge.

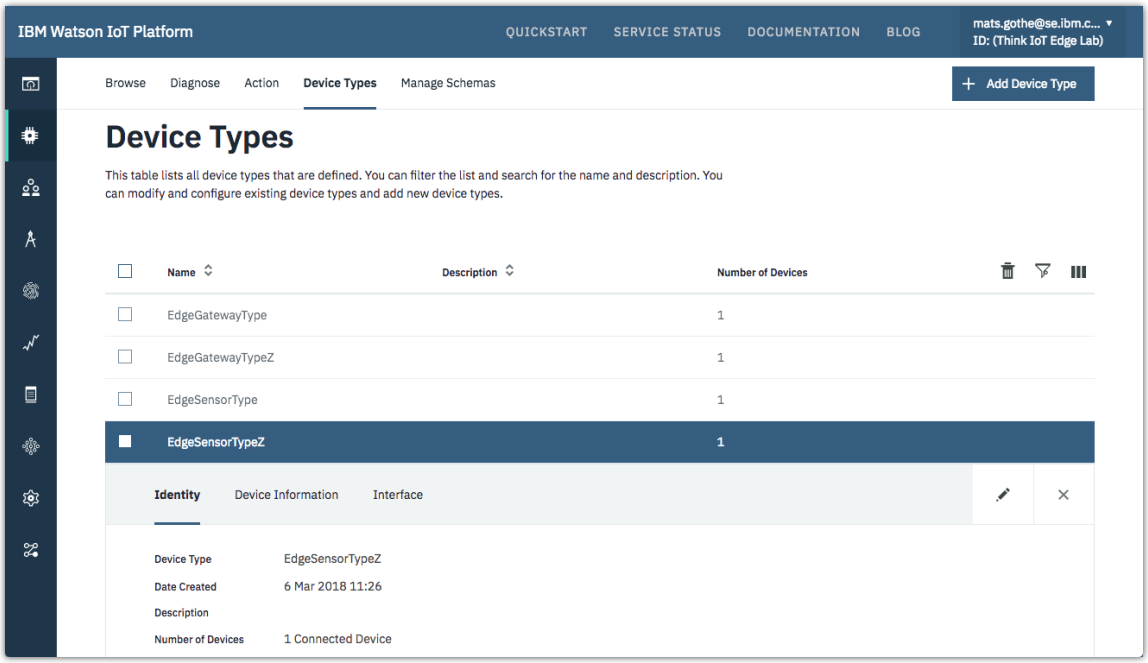
Data Management uses two types of **Interface** definitions added to a device type.

- *Physical Interface* – A physical interface represents the events that a device of a particular device type can send.
- *Logical Interface* – A logical interface represents the state of a device at a point in time. It's a canonical view which can be shared across multiple device types

6.3 Adding Interfaces

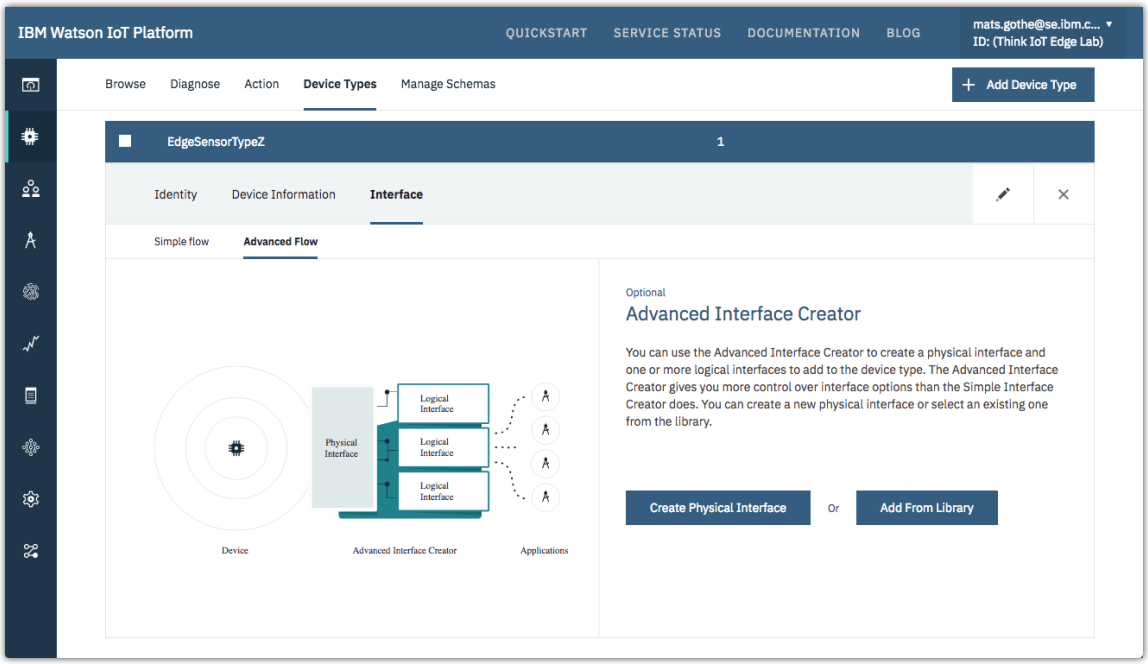
To create Interfaces for the Edge Sensor devices

1. On the navigation bar, select **Devices** section and the **Device Type** tab.
2. Find, select and expand your Edge Sensor Type in the list



3. On the device information card, select the **Interfaces** tab

4. On the Interface page, select the **Advanced Flow**



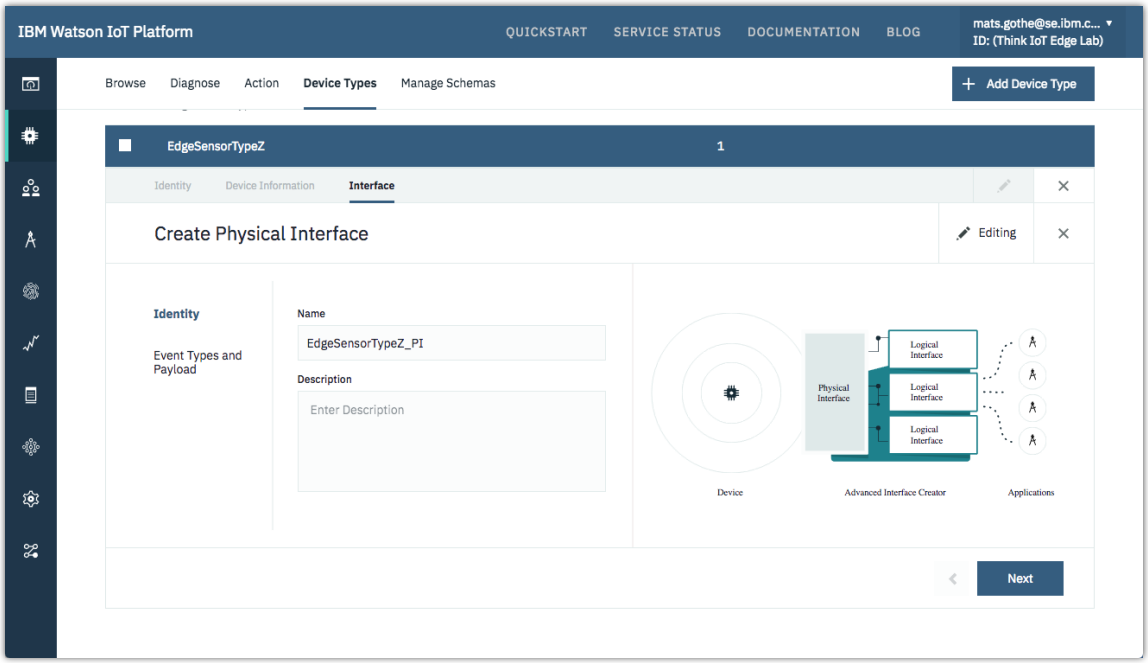
We are now ready to start creating the physical and logical interfaces for the Edge Sensor device type.

6.4 Adding an Edge Sensor Type Physical Interface

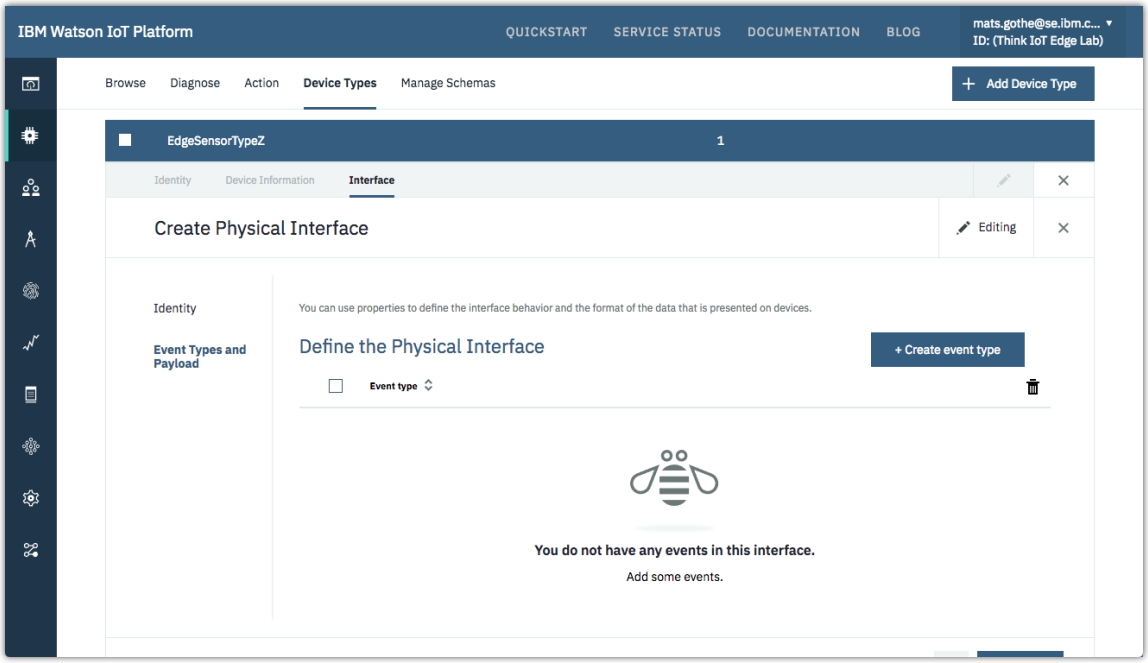
To add a Physical Interface

1. Click **Create Physical Interface** button. The Physical Interface editor page is shown.

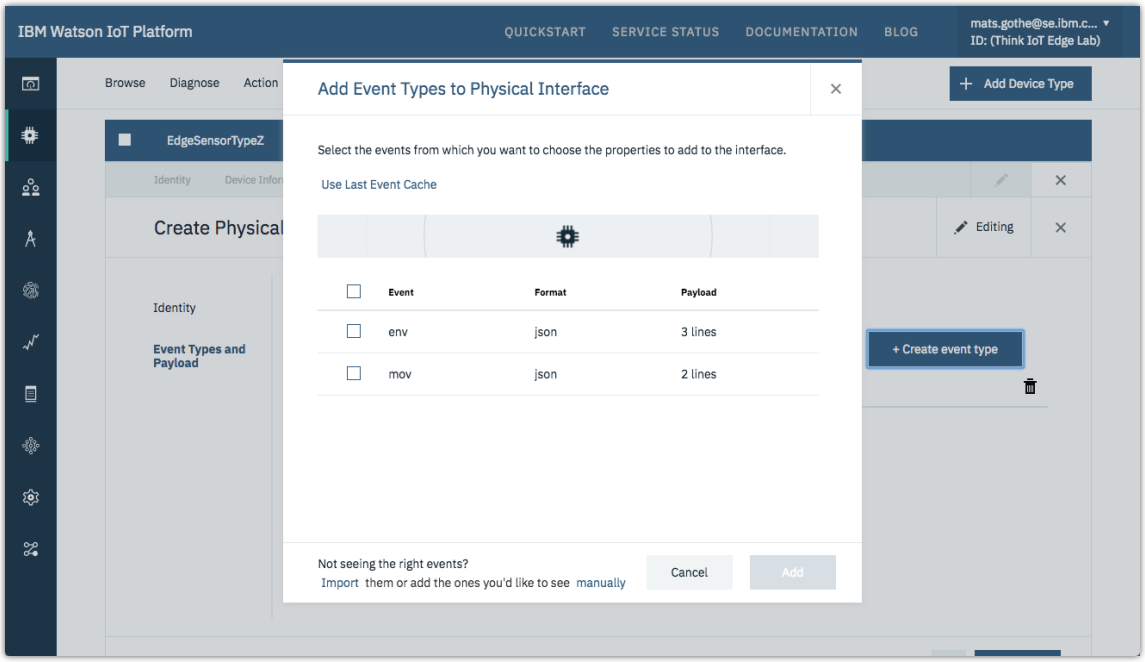
   Note: In most cases its favorable to use a unique name for the physical interface. Keep the default name and click **Next**.



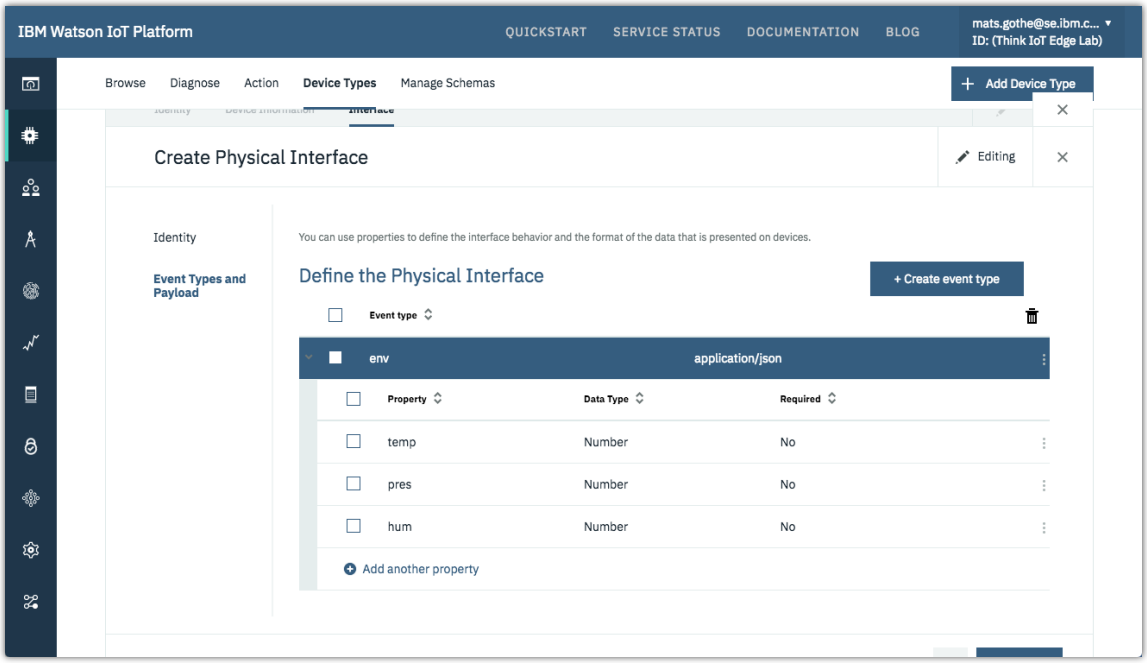
2. On the Event Types and Payload page, click the **+ Create Event Type** button



3. The Add Events Types dialog opens and start to listen for events. Wait for events to be received.
4. Select the **env** event and click **Add**



5. The event type schema for the **env** event is added to the physical interface. Expand the row to see the details of the event type schema.



We have now added the event type schema for one of the event types published by the Edge Sensor device, that is the **env** events. For an IoT application this may be sufficient sensor data required by the application. By omitting any further event types in the Physical Interface will be filter out all other event data from being sent to the further processing and transfers to the cloud.

In this lab you may optionally proceed and add the **acc** and **mov** events type schemas.

6. Optionally, repeat the steps above to add event type schemas for the **acc** and **mov** events.

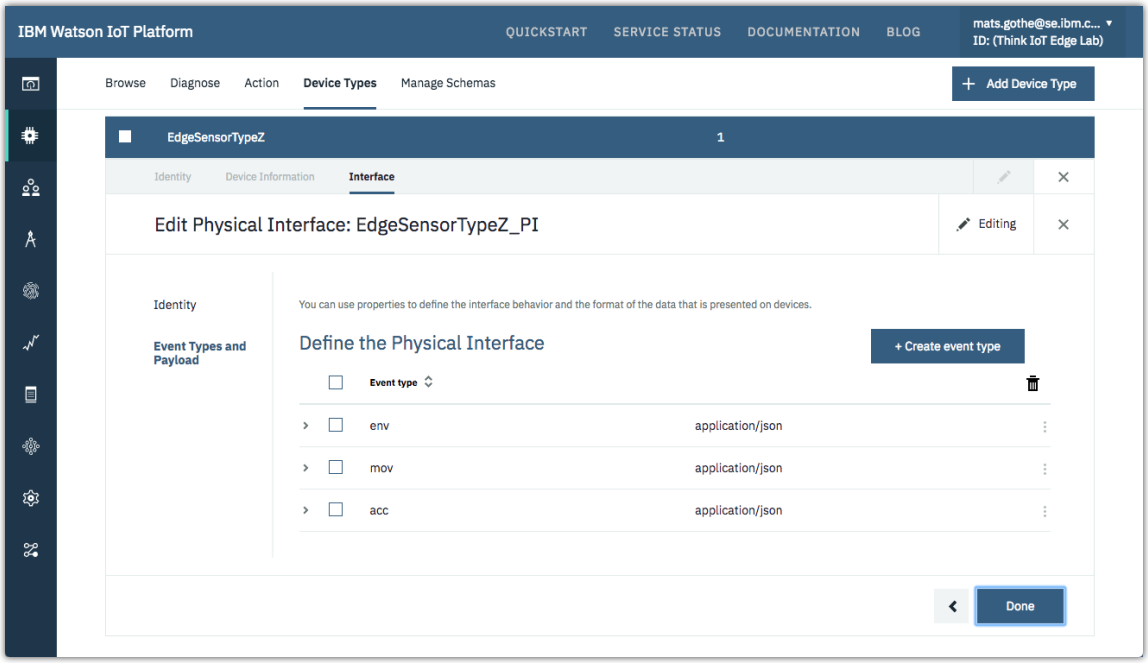
   To add the additional **acc** and **mov** event type schemas

   i. Click the **+ Create Event Type** button

   ii. Wait for the event to be received, select and click **Add**

   iii. Expand the added event type to see its event type schema details

7. Once all events types have been added, click **Done**

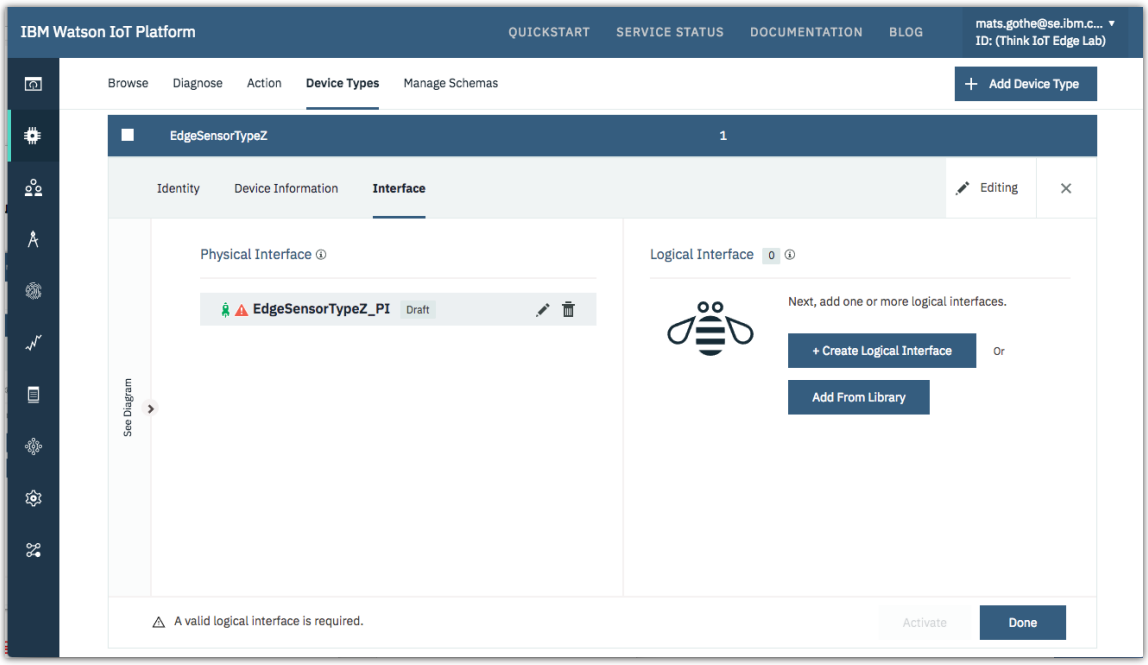


You have now created a Physical Interface for the Edge Sensor device type.

We will now add a Logical Interface for the type.

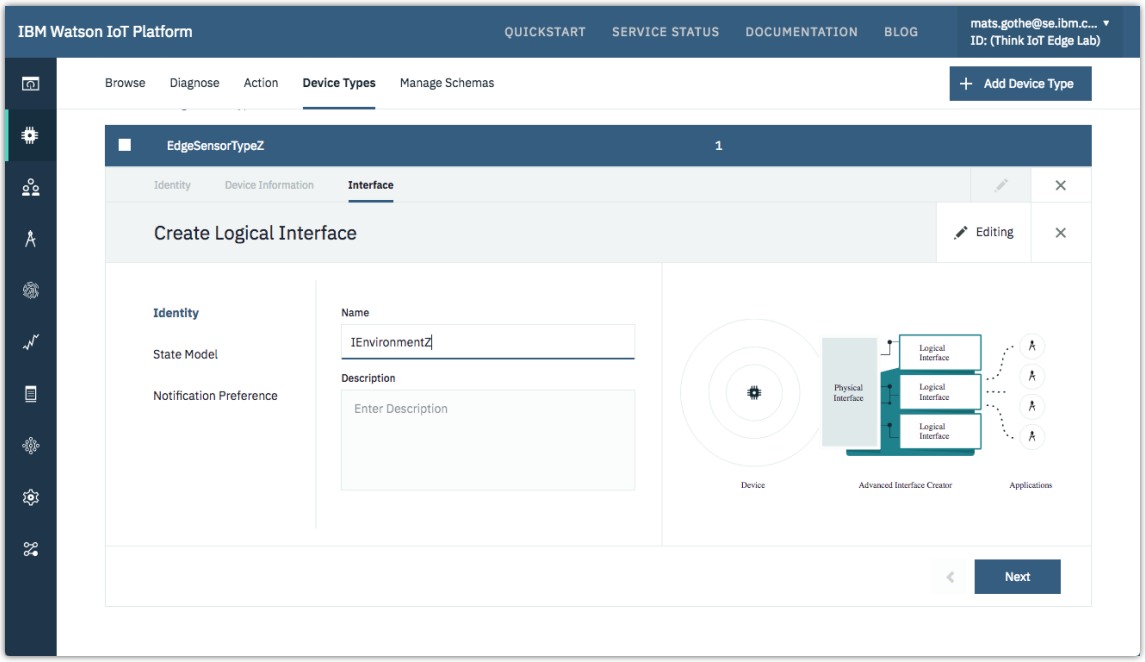
6.5 Adding an IEnvironment Logical Interface

1. On the Interface page, click **Add Logical Interface** button.
   The logical interface editor opens.



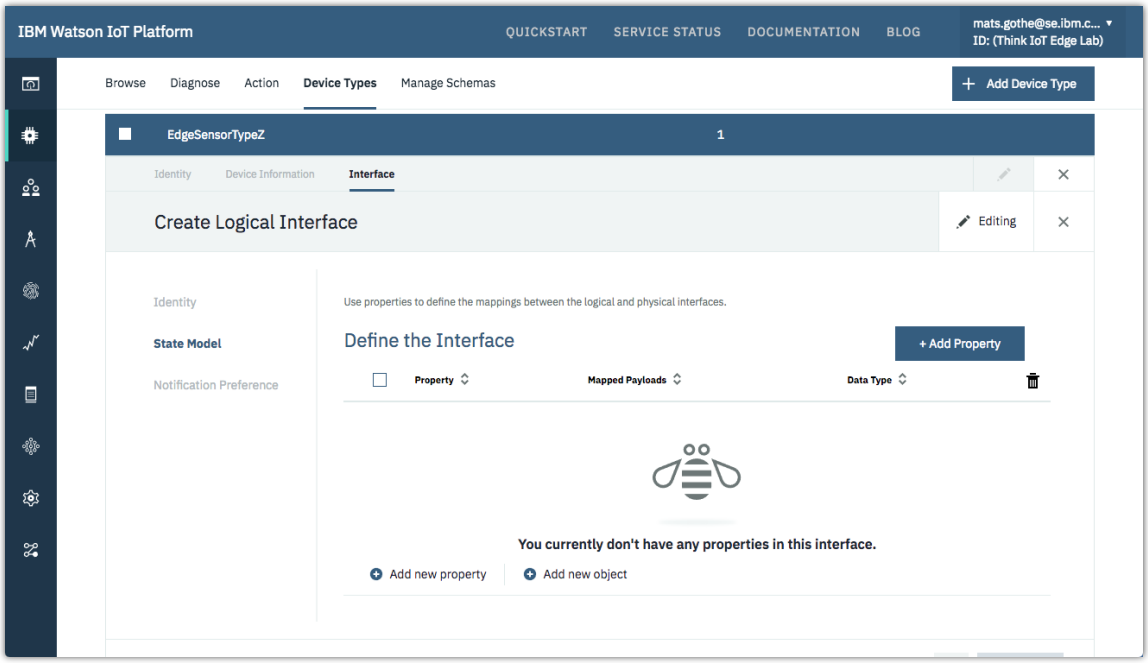
2. Replace the default name with a unique name staring with **IEnvironment**.
   Use the name extension given for your workstation. Click **Next**.

   Note: Logical Interface are a reuse asset. It's important to give the interface a name that corresponds to the abstraction its defining. In this case we will expose the state of environment and the IEnvironment is a suitable name for this abstraction.

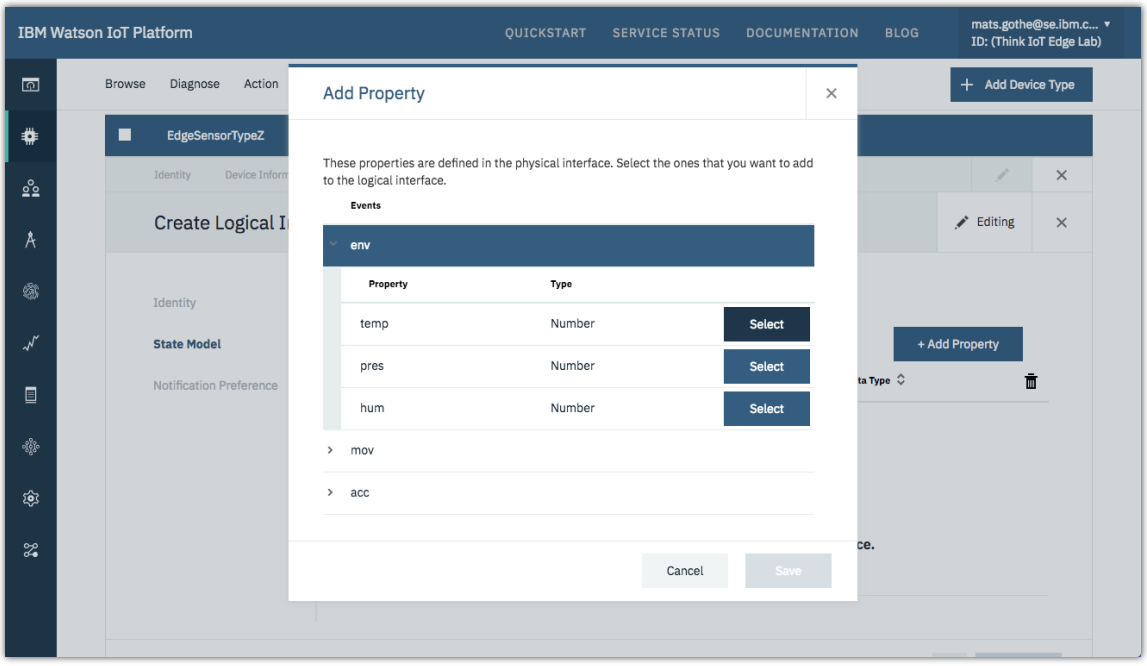


We will now add the state properties of the Logical Interface. The State properties define the abstraction of the interface.

3. On the State Model page, click the **+ Add Property** button

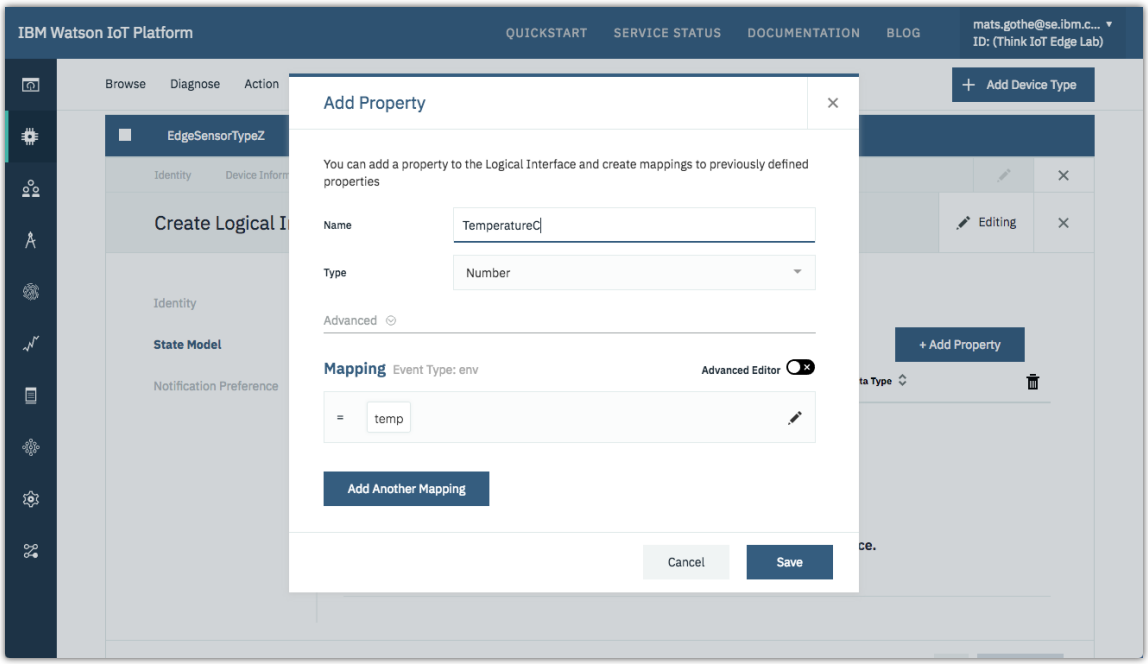


4. In the Add Property dialog, expand the **env** event type, then click **Select** on the **temp** property row.



5. The temp property is now added to the mapping expression for the temperature state property. In the Add property dialog, edit the default name to **TemperatureC**.

   Click **Save**.



6. Repeat the steps above and add a second property.

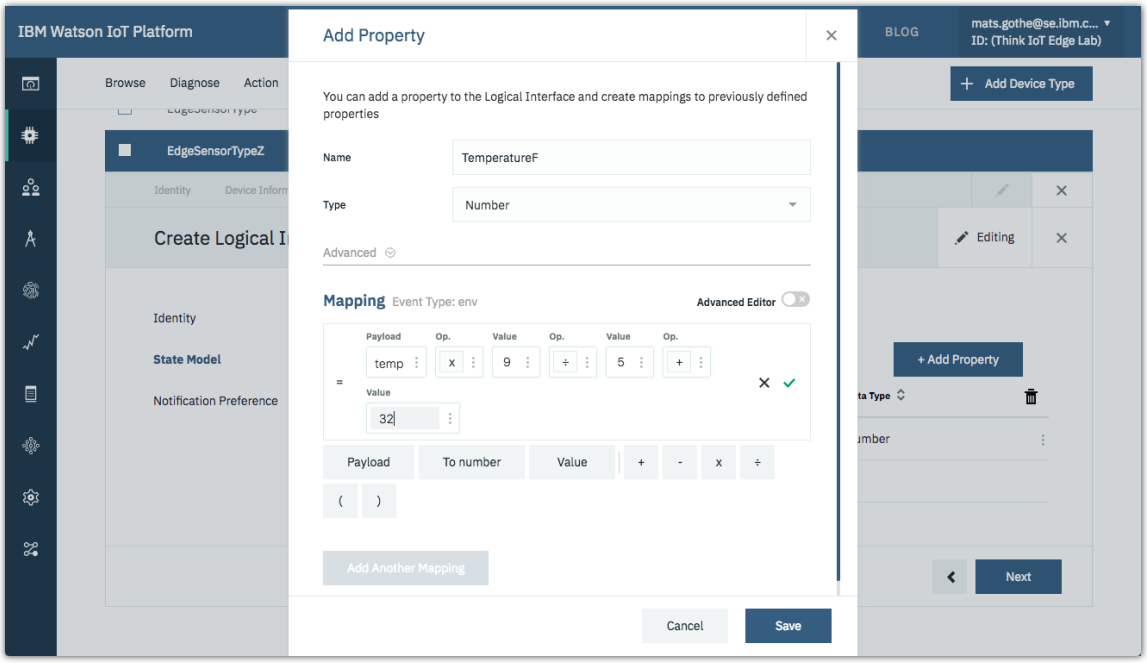
   Add a new property.

   Select the **evt** event **temp** property.

   Rename the property TemperatureF.

7. Click on pen icon in the expression builder to start editing. Then, use the editor buttons to build an expression that converts a Celsius temperature to Fahrenheit.

   °F = (°C × 9/5) + 32



8. Repeat the steps above to add state properties for Pressure and Humidity.

   Note: These two properties do not need any transformations, just create a mapping by selecting the **pres** and **hum** properties in the **env** event type schema.

   After completing the definition of all state properties, click **Next**

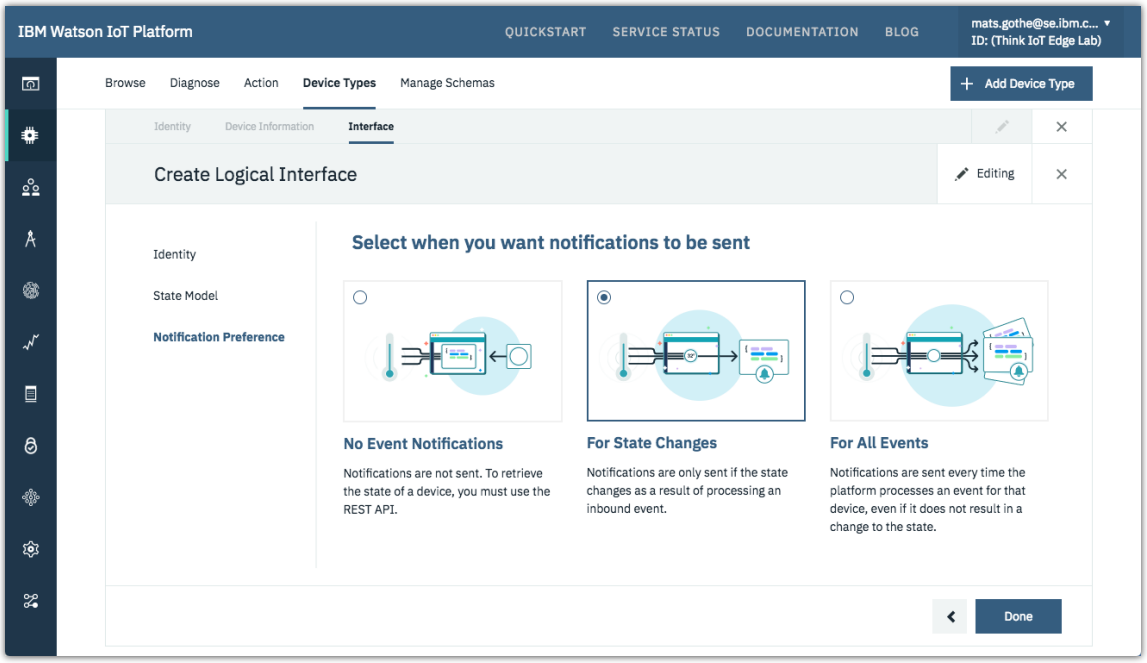
9. On the following Notification page, select how notifications of state changes will be submitted.

   The options, in the dialog, are

   - Send a notification after each device event is received
   - Send a notification only after state has changes,
   - or never.

   In this lab, choose **For State Changes**.

   Click **Done**.



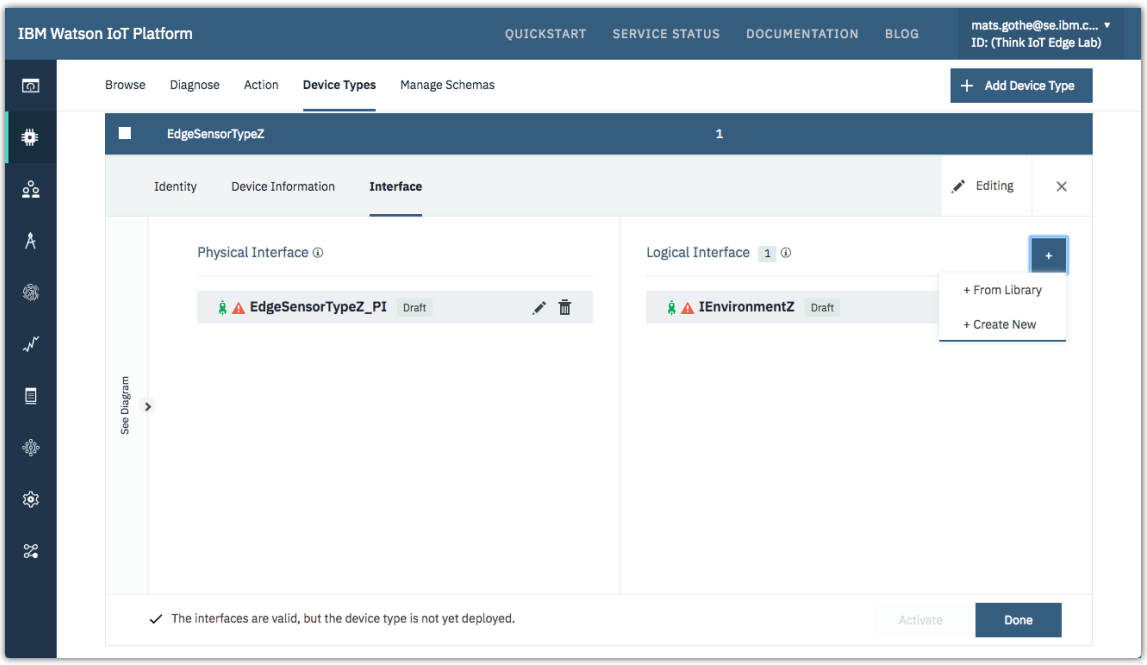
10. The device type page now shows the new logical interface added to the Interface page.

You may now optionally add a second Logical Interface

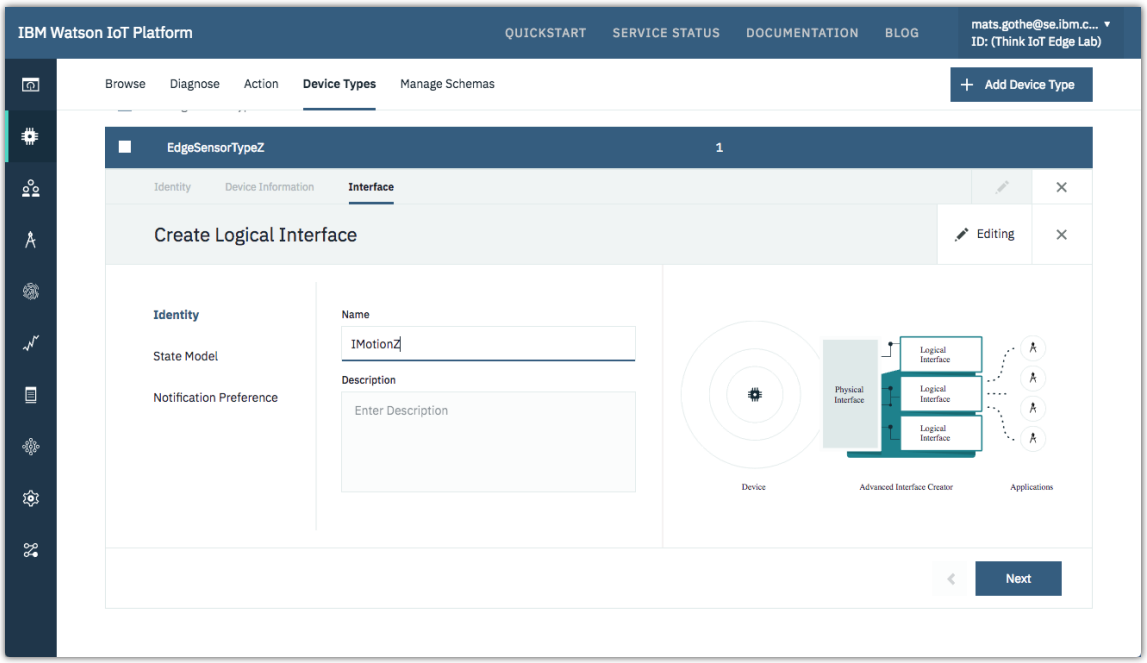
6.6 Optional: Adding an IMotion Logical Interface

In this section you may optionally add a second logical interface that provides the motion state for Edge Sensor devices. This logical interface will demonstrate how to map state properties to multiple event types.

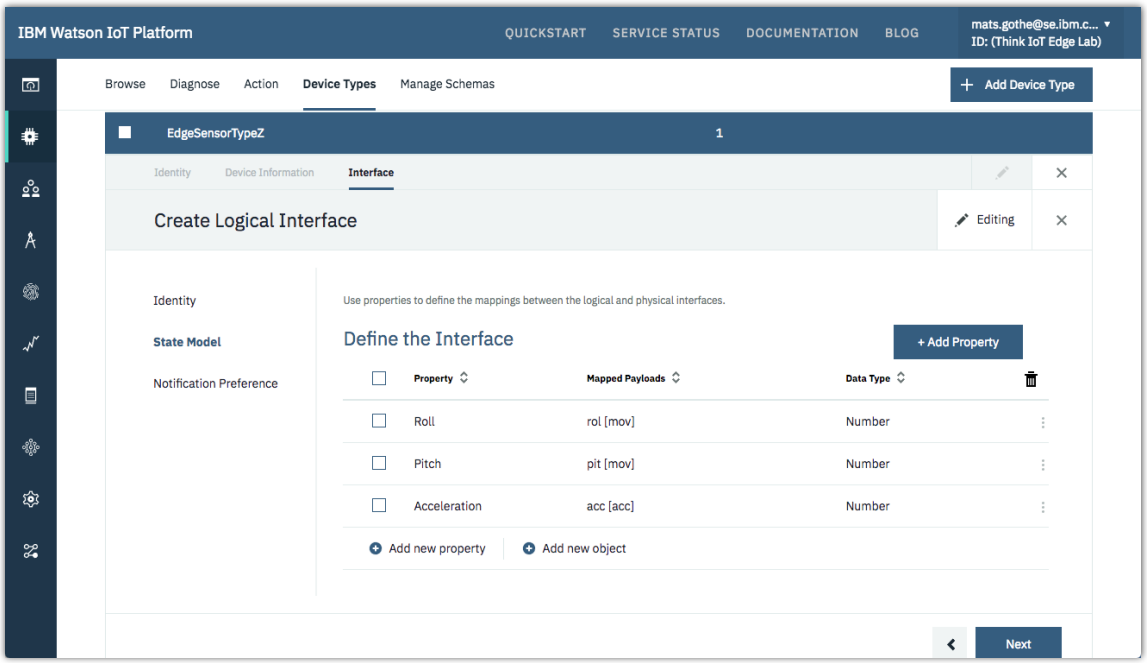
1. Add an additional logical interface by clicking the **+** button and then clicking on the **+ Create New** option.



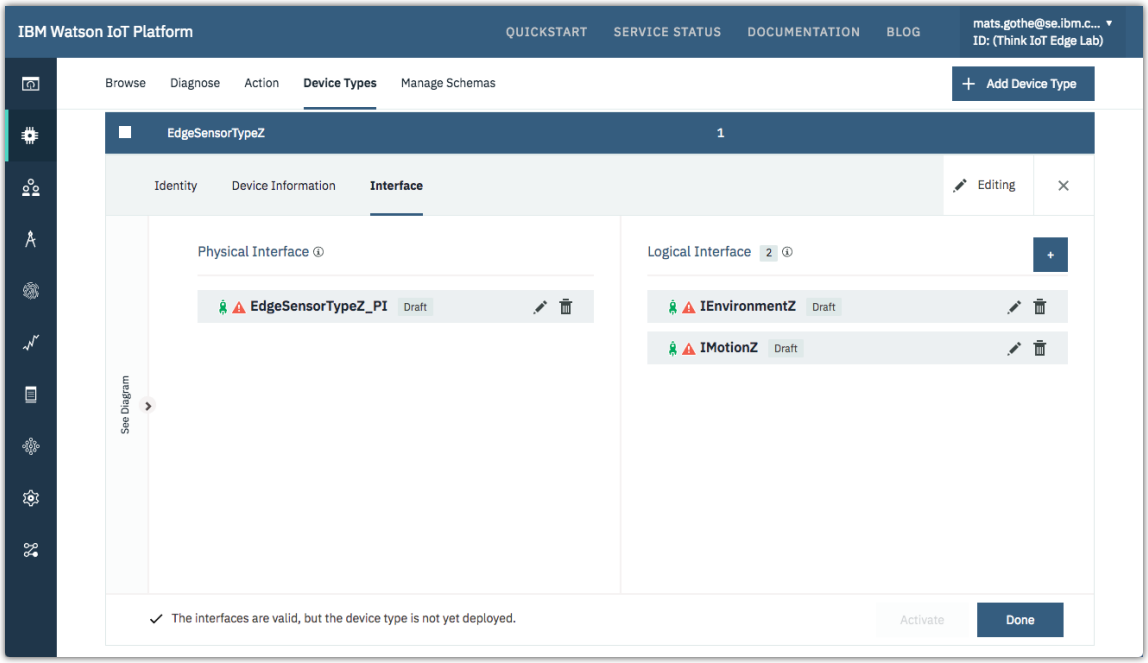
A new logical interface is created.
Give a unique name for this new interface using **IMotion** and your unique name extension.



2. For this interface, add state properties for
    - *Roll*, a number type property mapped to the **rol** property in the **mov** event type
    - *Pitch*, a number type property mapped to the **pit** property in the **mov** event type
    - *Acceleration*, a number type property mapped to the **acc** property in the **acc** event type



3. Click **Next** and then choose **For State Changes** for the notification setting.
4. Click **Done** to complete the logical interface.



5. Click **Done**.

6.7 Activating the Physical and Logical Interfaces

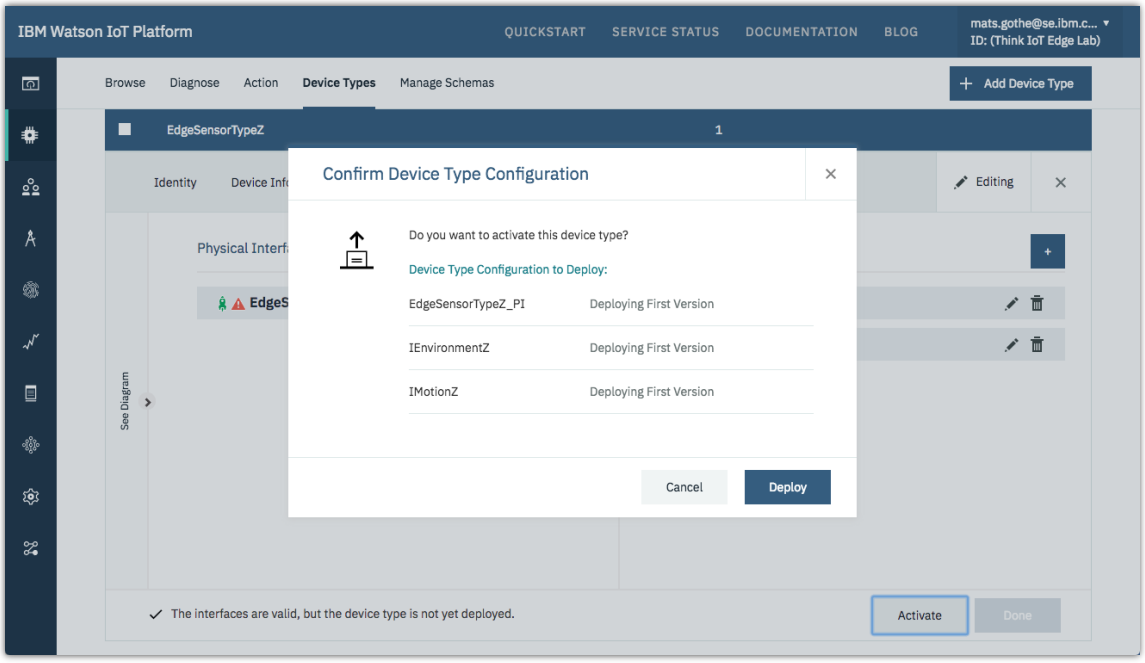
In this section we will activate the interfaces. The activation step will move the interfaces we just created from a DRAFT deployment configuration to an ACTIVE deployment configuration. The Data Management ACTIVE deployment configuration represents the compiled resources deployed into an event processing pipeline in the Watson IoT Platform. After the interfaces has been activated and deployed, updates can be made and saved to the DRAFT deployment configuration. Such changes can then later be activated and updates made to the running ACTIVE deployment configuration.

To activate the physical and logical interfaces

1. On the Interface page, click **Activate** to bring the interfaces into the event processing pipeline.

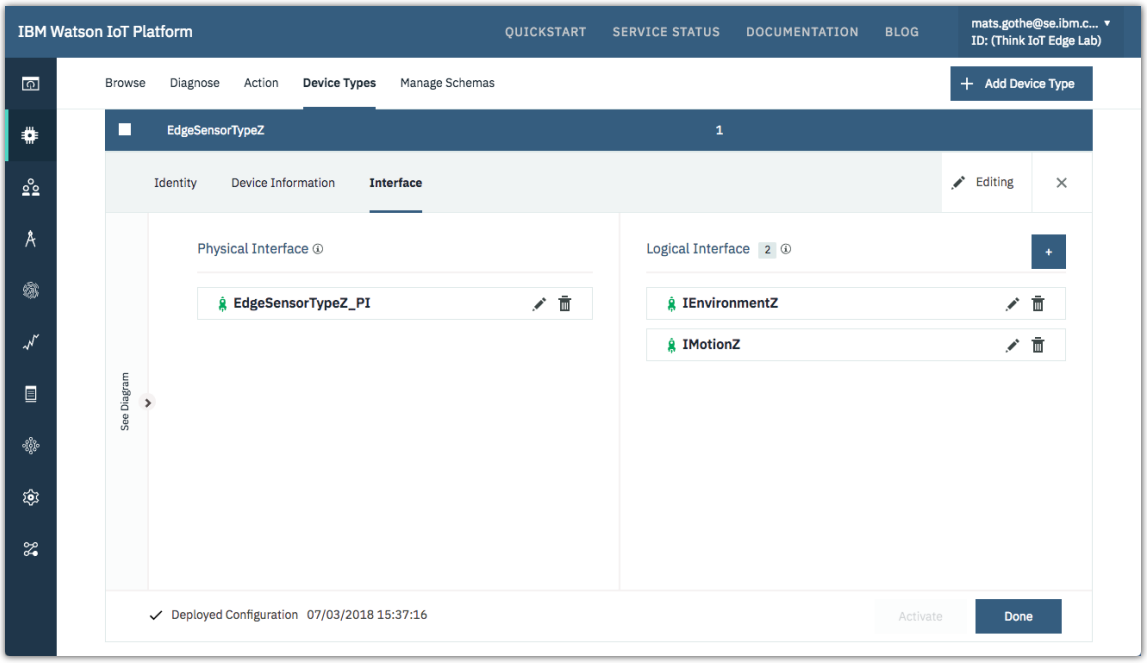
   A confirmation dialog is shown listing all interface resources that will be activated.

2. Click **Deploy**



3. After the activation is completed the Interface page is updated with state information.

   Note: The Interface page shows the activation state and timestamp of the interfaces.



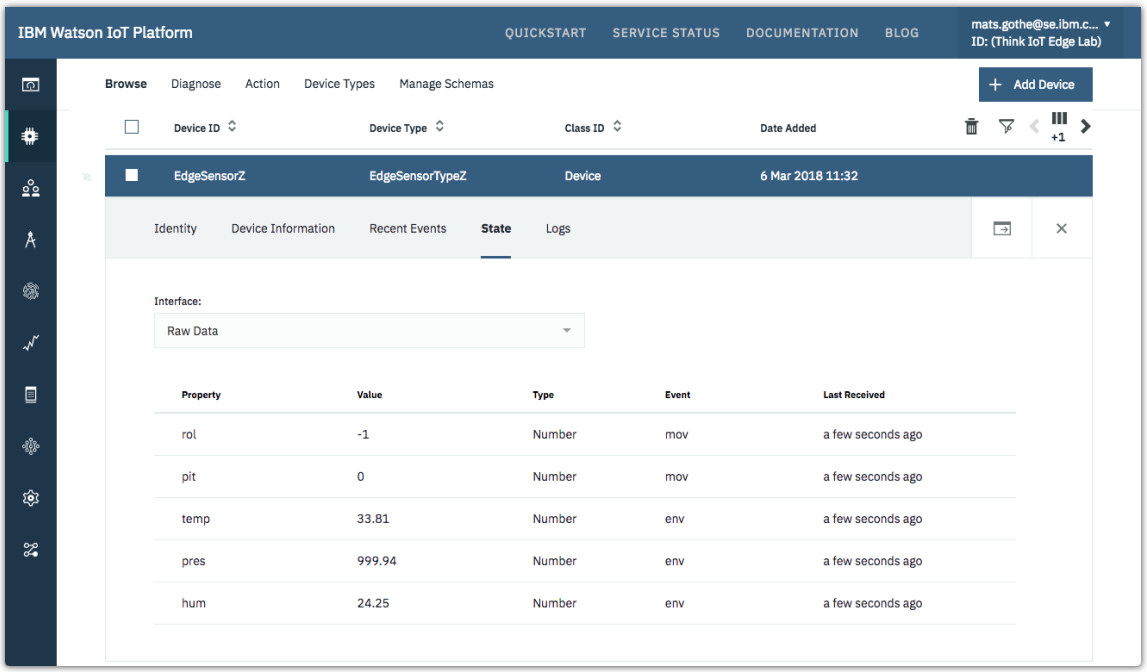
4. Click **Done** to close the page.

6.8 Viewing Device State

Using the Edge Sensor interfaces, we can now view the device state and updates to the state.

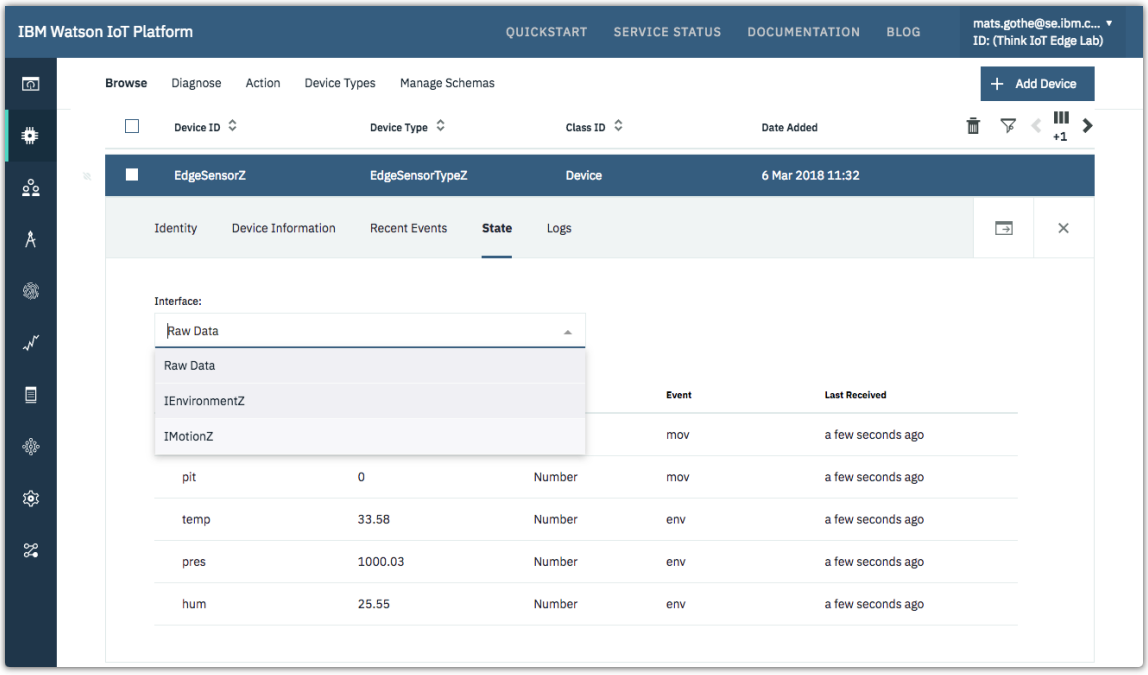
To view device state

1. From the navigation bar, select **Devices** section.
2. Locate your Edge Sensor in the list of devices.
3. Select and expand the device details.
4. Select the **State** tab



5. By default, the **Raw Data** from all event type schemas are shown

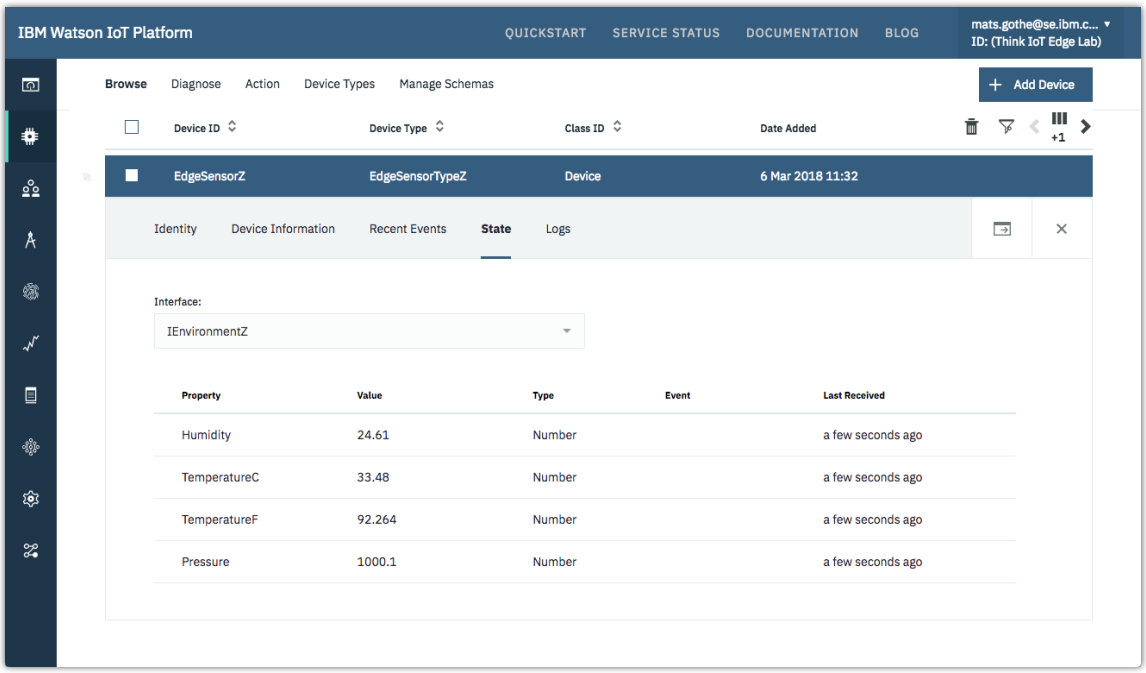
   To view the state of an interface, choose the interface from the **Interface** dropdown.



6. Select your **IEnvironnemt** interface. The list of properties is updated, and the state of the environmental properties are presented.

   Note: The state for the properties are updated as new device events are received.

   Note: The Celsius temperature is received from the device. The Fahrenheit temperature is computed by the interface.



7. Choose the IMotion interface. Move and shake the Raspberry Pi to generate motion events with varying data.

6.9 Conclusions on IoT Edge Data Management workloads

We have now completed the section in this lab that explores IoT Edge with Data Management workloads.

In this section we have:

- Defined the filtering, transformation and abstractions of an Edge Sensor and defined its Environment and Motion characteristics
- Created a Physical Interface and declared the event types schemas for the messages that we will use in the transformations. Other events will be discarded by the physical interface.
- Created three Logical Interfaces for Environment state, Motion state and Analytic state computations.
- We have defined these Data Management resources in the IoT Platform on the cloud. We have discussed that the IoT Edge will distribute these resource from the cloud to the edge gateway and start an event processing pipeline on the edge gateway for any device connecting to the gateway of a type that has interfaces defined.
- We have disuses the value for device developer to define the Data Management resources and have them deployed into edge gateways as needed by the device connection topology.
- We have discussed the value for application developers to be agnostic to the instrumentation of devices.

In the next section of this lab will we now explore the Edge Service workloads for the IoT Edge.

7 Working with IoT Edge Service workloads

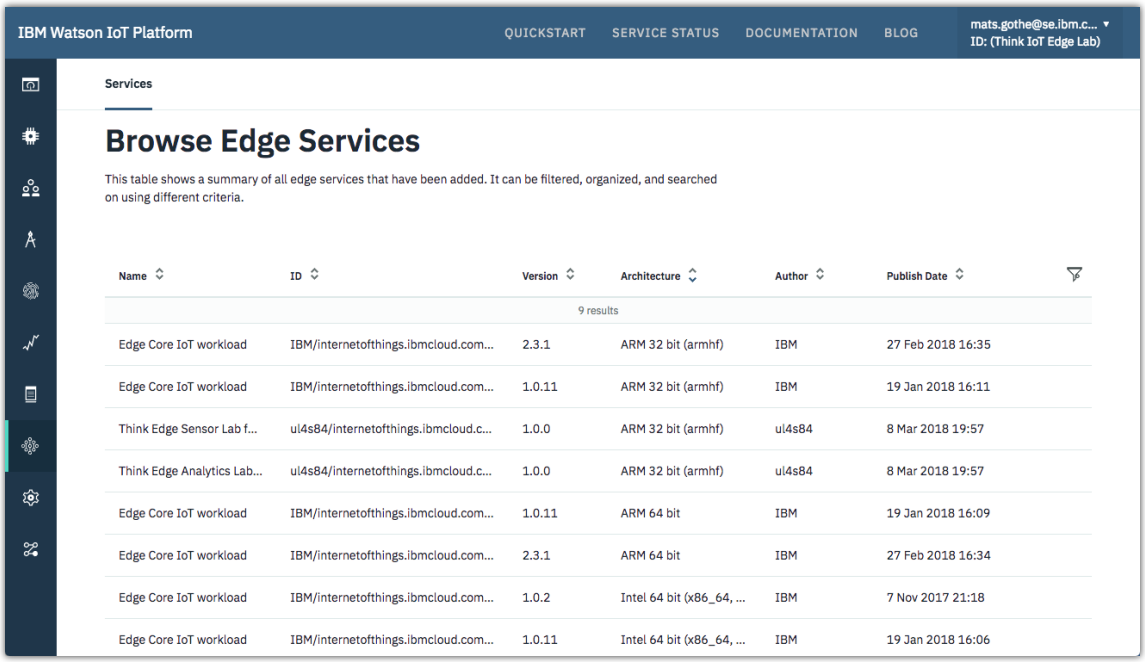
In the initial section of this lab we configured the EdgeGatewayType with the **Think Edge Sensor Lab for arm** and **Think Edge Analytics Lab for arm** edge services.

We will now explore some more details of these services.

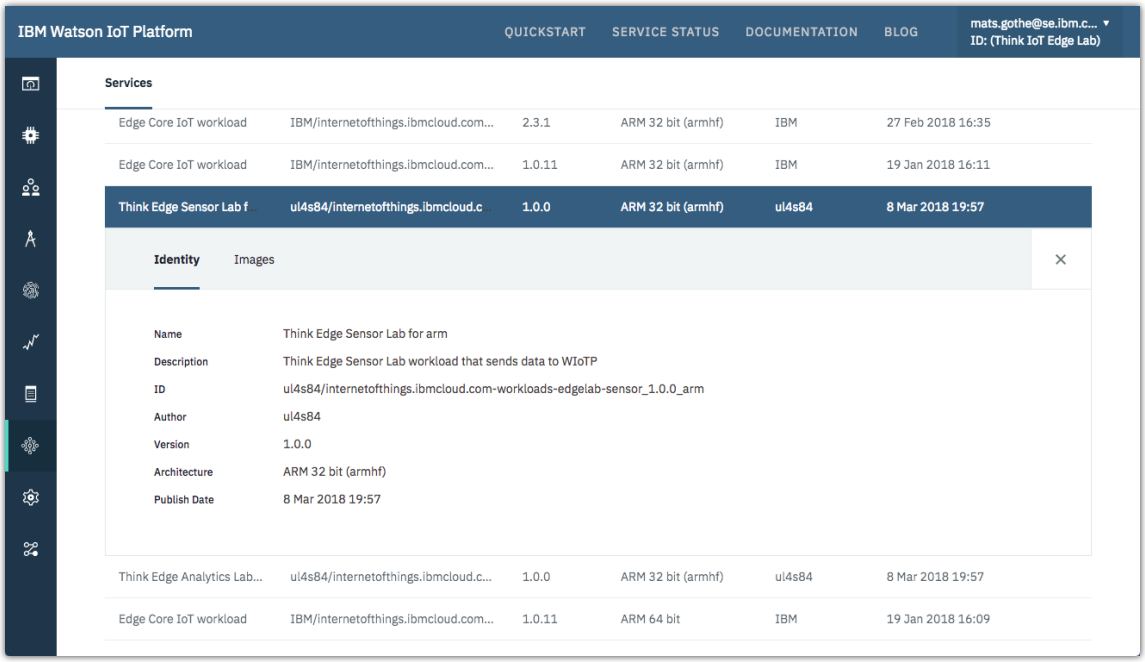
7.1 Exploring the Edge Lab Sensor Workload

To view the edge services and images in the IoT Edge Catalog:

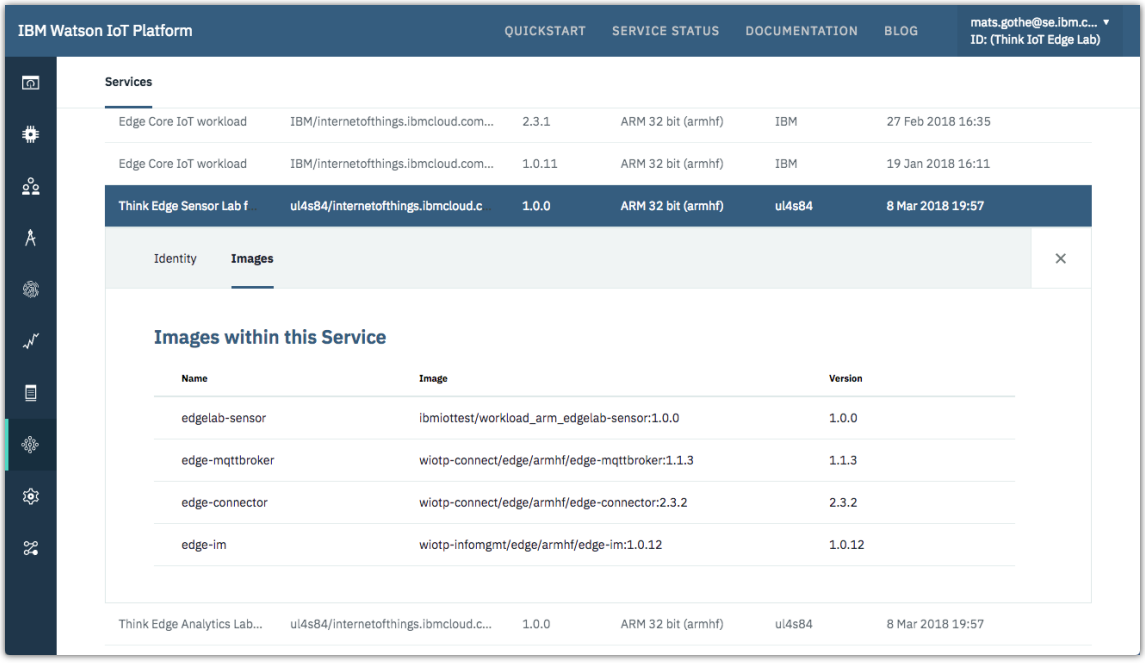
1. From the navigation bar, select **Edge Services** section



2. Locate the **Think Edge Sensor Lab for arm** service in the list.
   Select the row to see the edge service details.



3. In the **Think Edge Sensor Lab for arm** details page, select the **Image** tab to view the Docker images configured in the service.



Note: The Edge Lab Sensors Workload consist of the *edgelab-sensor* image that is the Docker conatiner that we have used in the previous sections in this lab. The *edgelab-sensor* image is written in Python and uses Eclipse Paho Python library to publish and subscribe to local events on the edge gateway. The container is simulating the Edge Sensor device and is submitting events with the env and mov events.

The servcie also references other images used like the MQTT broker image, the edge connector image and the IM image running the Data Management workload.

4. Scroll the list of services and select the **Think Edge Analytics Lab for arm** service.

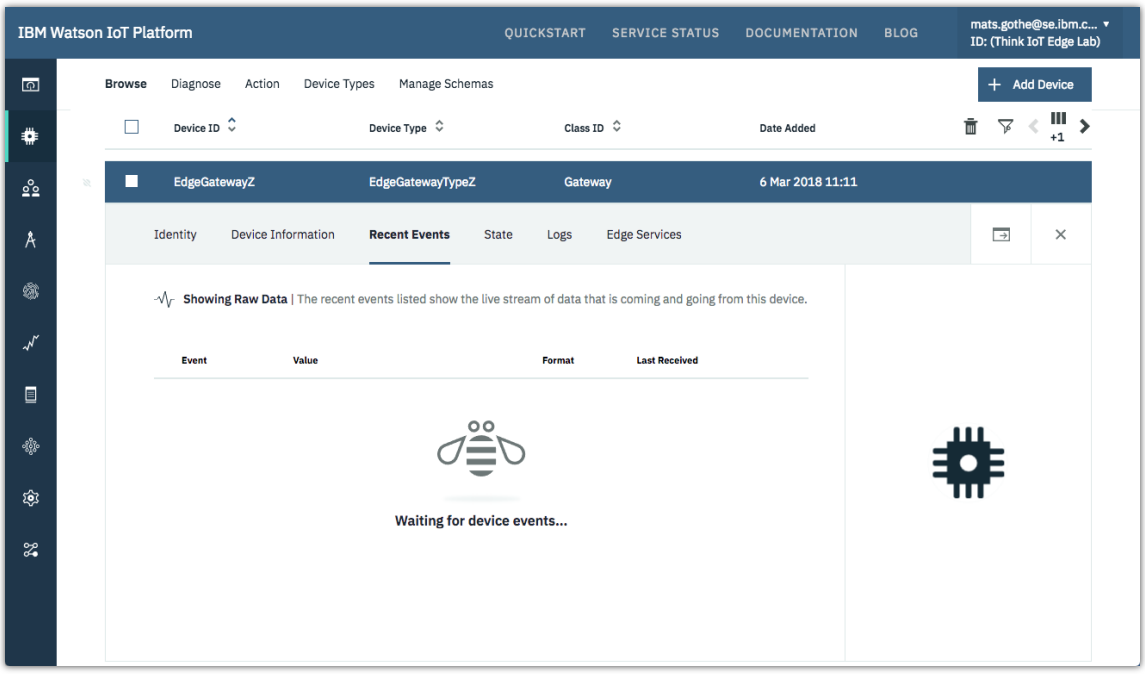
   Click the **Image** tab to view the Docker images configured in the service. This service includes the *edgelab-analytics* image.

We will now explore analytics performed by the edgelab-analytics Docker component.

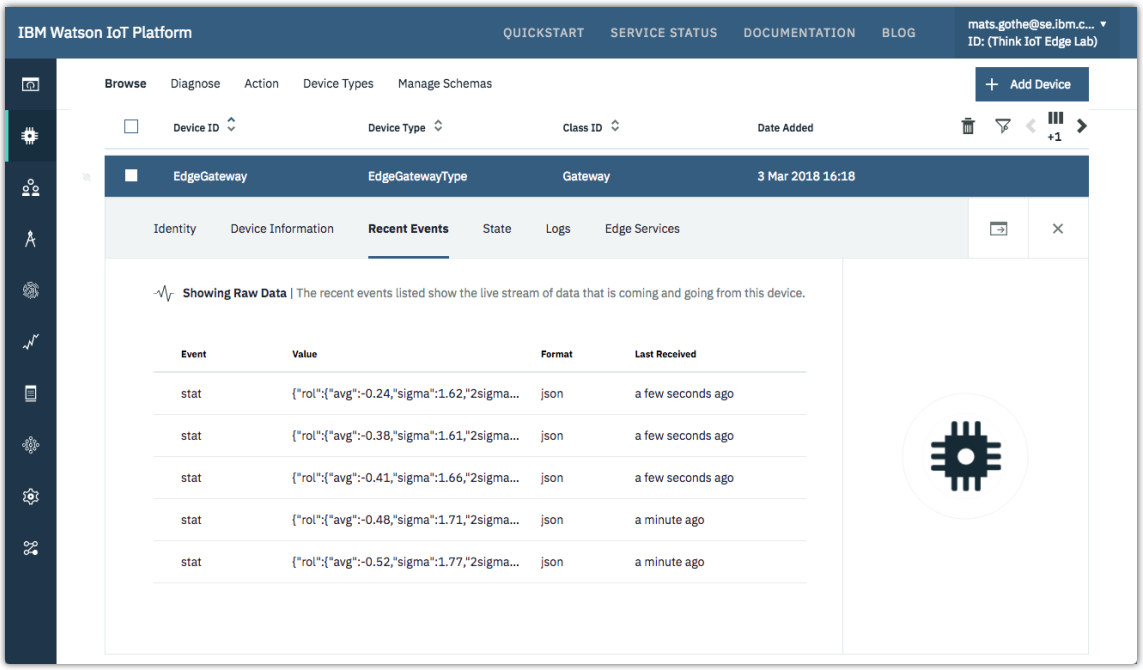
7.2 Viewing the IoT Edge Service Analytics events

To view events from the edgelab-analytics application:

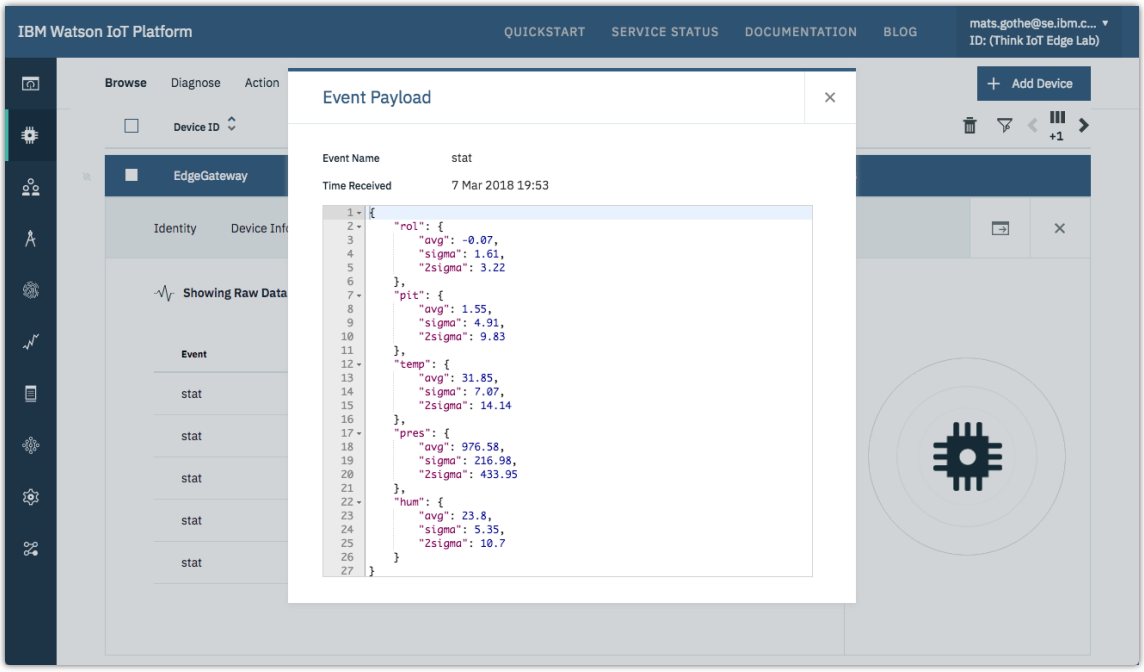
1. From the navigation bar, select **Devices** section
2. Locate the Edge Gateway in the list of devices
3. Select and expand the device details.
4. Select the **Recent Events** tab



5. Wait for application events to be received



6. The edge analytics application is submitting **stat** events every 15th sec. Select one of the **stat** event to view its details.



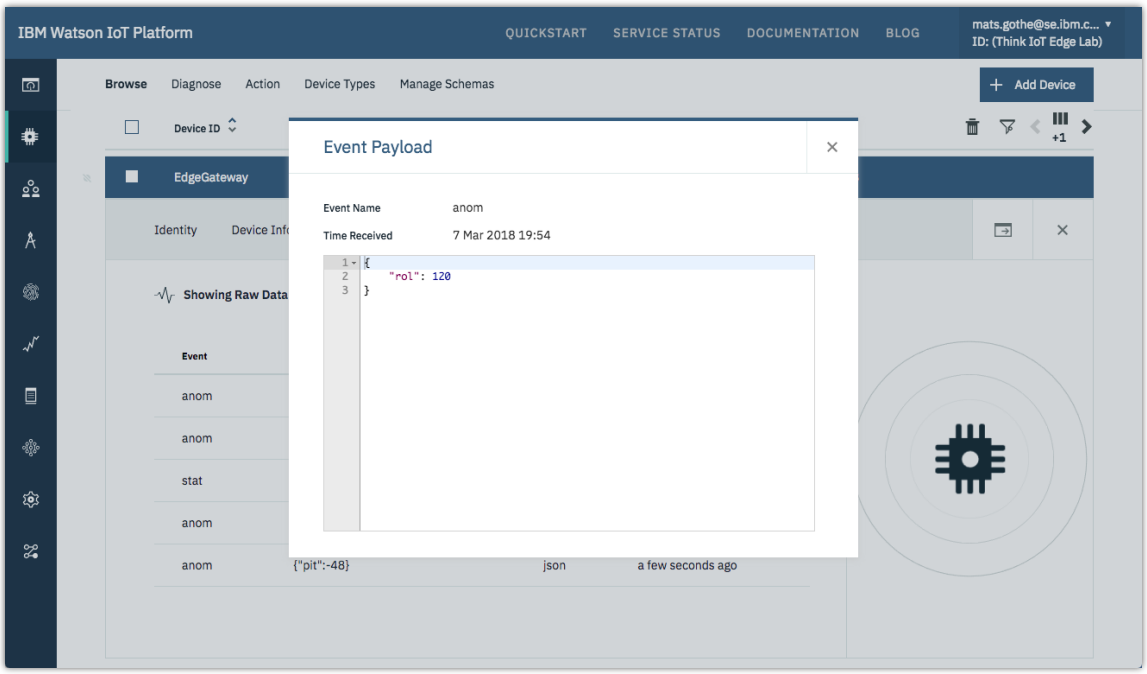
7. The event payload contains average, standard deviation and sigma analytics for each unit if data.

8. The edge analytics application also identifies anomalies in the data and sends an **anom** event when data is larger or smaller than 2sigma.

   Pick up the Raspberry Pi to generate anomalies

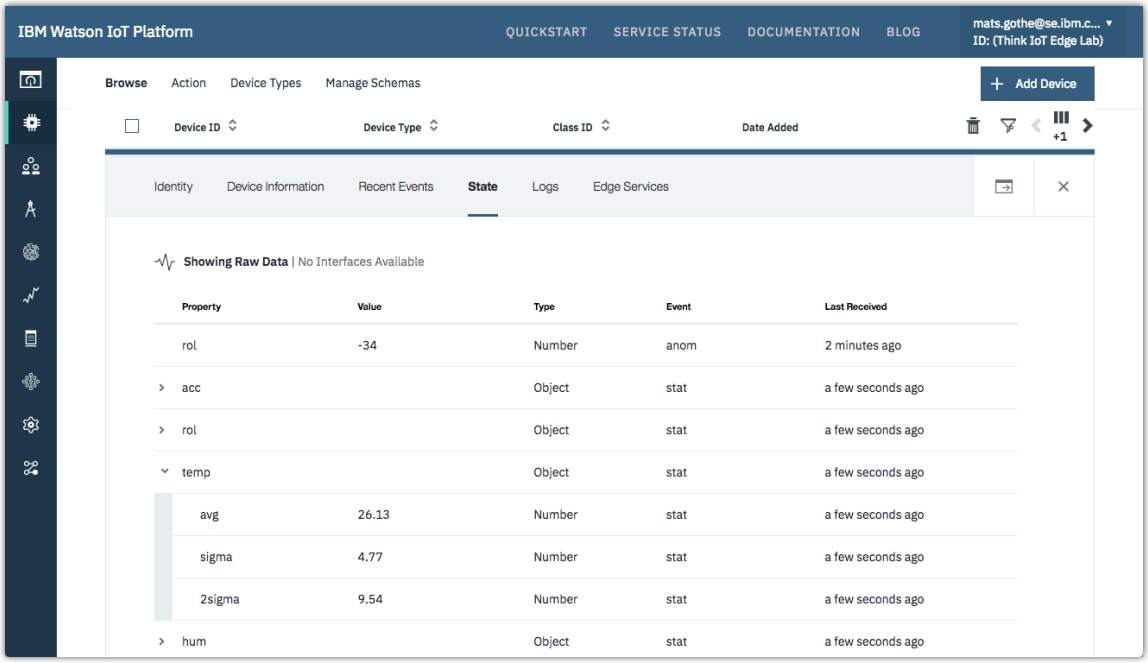
   - Tilt the Raspberry Pi more than 90 deg
   - Shake the Raspberry Pi more than 3G

9. View anomaly events received from the edge application.
10. Click on an event to view details.



The state tab provides a more structured view with live updates of event data.

11. Choose the **State** tab
12. Expand the structure to view the analytics for each sensor value



We have now completed the exploration of the edgelab-analyics container.

7.3 Conclusions on Edge Service Workloads

We have now completed the section in this lab that explores IoT Edge Service workloads.

In this section we have:

- Explored the Edge IoT Core workload and the mandatory images that provides messaging on the gateway
- Explored the Edge Lab Sensors workload that has provided environment and motion data to this lab.
- Viewed data published to the IoT Platform by the Analytics edge application.

In the next section of this lab will we now explore the code of an edge safety application. We will build and run the edge service and a Docker container.

8 Advanced Topics – Safety Edge Application

Many IoT business solutions use the basic concepts that we have explored in this lab. Worker safety in IoT for Insurance is one example for such solutions. Other are medical solutions and home care solutions.

To demonstrate these IoT Edge concepts in the context of Worker Safety we have evolved the Analytics application into a Safety edge application. This application is subscribing to environmental events from Edge Sensors located on a construction site. The application is also subscribing to motion events from Biometric sensors on the worker or on safety wearables on a worker. Analyzing the sensor data, applying machine learning, and detecting anomalies will improve worker safety.

The safety application will detect a number of safety situations and generate alerts based on these conditions.

- A tilt on the sensor more than 90 degrees from the normal working vector will generate a TILT notification.
- If the sensor comes back to an UP direction within 3 sec, the notification is withdrawn.
- If the TILT state remains for more than 3 sec, the safety application will raise a DOWN notification.
- If the sensor then comes back to an UP direction the notification will be withdrawn.
- If the safety application detects an acceleration of more than 3G, a DOWN notification will instantly be sent.

The state of the safety application is shown on the Raspberry Sense Hat for visual feedback.

In this last section you will familiarize yourself with developing and running custom applications as Docker containers on the IoT Edge Gateway.

You may also ask your lab facilitators for a demo of the safety application.

8.1 Exploring the Edge Lab Safety Application

To view the edgelab-safety application

1. Log into the Raspberry Pi
2. On the command line, run the following commands

```
cd /home/pi/edgelab/docker/edgelab-safety
micro edgelab-safety.py
```

3. Browse the edgelab-safety python application

In the edgelab-safety application code notice a few IoT related commands

- `IOT_BROKER = "edge-connector"`
  `client.connect(IOT_BROKER, 8883, 7200)`
  The edgelab-safety python application is connecting to the local edge-connector container and the local MQTT broker using a TLS connection
- `APP_ID = "a:edgelab-safety"`
  The edgelab-safety python application is publishing as an application to the IoT Platform
- `client.subscribe(SENSOR_MOTION_TOPIC)`
  `client.subscribe(SENSOR_ACCELERATION_TOPIC)`
  The edgelab-safety python application is subscribing to motion and acceleration events from the sensor
- `SAFETY_TOPIC = "iot-2/evt/saf/fmt/json"`
  The edgelab-safety python application is to the **saf** event
- ```
  def mqtt(state):
      m = '{"state":"' + state + '"}'
      client.publish(SAFETY_TOPIC, m)
  ```
  The edgelab-safety python application is publishing a state JSON payload with the worker state
  ```

8.2 Building the Edge Lab Safety Application

To build the Edge Lab Safety Application:

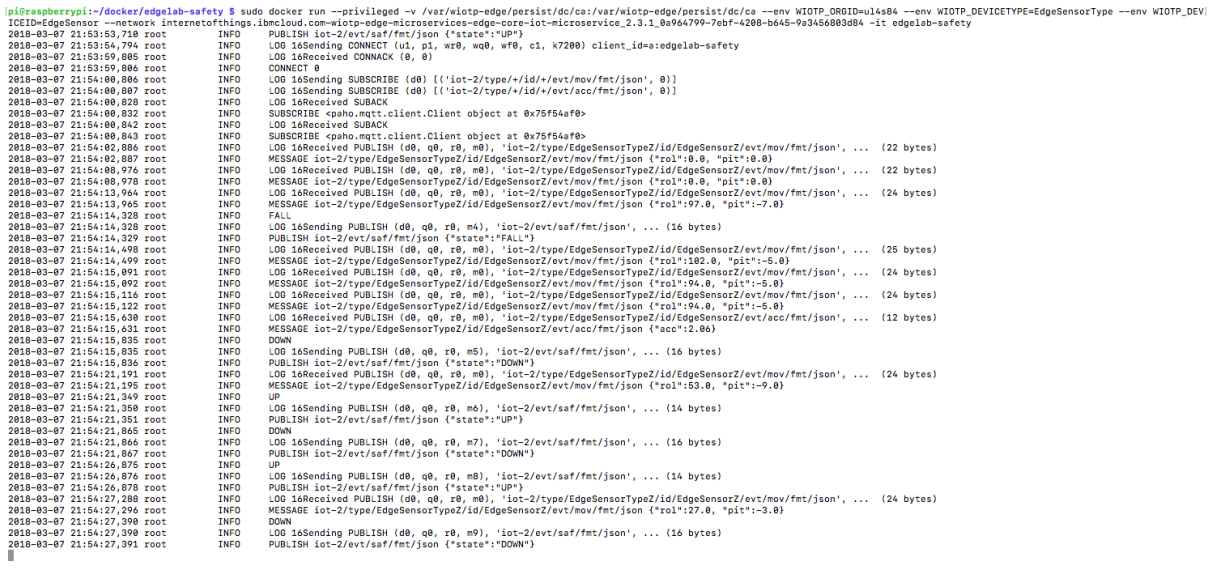
1. Log into the Raspberry Pi
2. On the command line, run the following commands

```
cd /home/pi/edgelab/docker/edgelab-safety
sudo docker build -t edgelab-safety .
```

3. Edit the command below. Replace the sections marked in red with
   - the organization id used in this lab
   - the edge sensor type name
   - the edge sensor name

```
sudo docker run --privileged -v /var/wiotp-
edge/persist/dc/ca:/var/wiotp-edge/persist/dc/ca --env
WIOTP_ORGID=ul4s84 --env WIOTP_DEVICETYPE=EdgeSensorType --env
WIOTP_DEVICEID=EdgeSensor --network $(sudo docker network ls |
grep "edge-core-iot" | awk '{print $2}') -it edgelab-safety:0.0.9
```

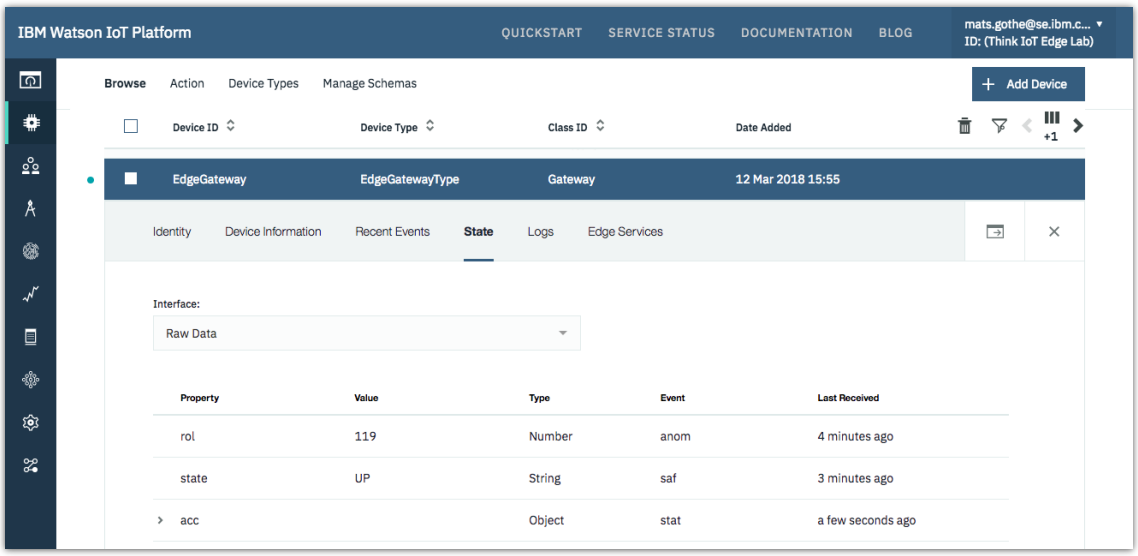
4. Run the edited command
5. The edgelab-safety Docker container now starts on the Raspberry Pi and lightens up the screen with a green up arrow
6. The application will output log information to the screen.
   Note messages published and subscribed by the application.



8.3 Monitoring Worker Safety

To monitor the safety application notifications:

1. From the navigation bar, select **Devices** section
2. Locate and select your Edge Gateway Type in the list
3. Select the **State** tab and monitor state changes on the **saf** event and **state** property.



Play the role as a worker and simulate a fall to the side.

1. Tilt the Raspberry Pi to the side to simulate a fall.
   The display shows an orange warning of fall.
   The IoT platform receives a **saf** event with a **TILT** status
2. Get back up by resuming the up direction of the Raspberry Pi
   The display goes back green
   The IoT platform receives a **saf** event with a **TILT ^ UP** status
3. Now fall again and remain in a fall position.
   The display goes orange and then red indicating a DOWN
   The IoT Platform receives a **saf** event with a **DOWN** status
4. Get back up again after the fall and wait for the display to go back to a green state
5. Shake the Raspberry Pi to simulate a fall with a strong hit to the ground.
   The display should go red instantly indicating a DOWN.
   The IoT platform receives a **saf** event with a worker **DOWN** status
6. When you are done, press **ctrl-c** to stop the edgelab-safety Docker container

You have now completed the exploration of the Safety edge application.

9 Conclusions from this Watson IoT Edge lab

You have now completed this hands-on lab on Getting started with IoT Edge Gateways.

During this lab you have:

- Depend your understanding of the Watson IoT platform and the new IoT Edge capabilities
- You have configured Edge Gateways Types for IoT Edge
- You have configured a Raspberry Pi as an IoT Edge Gateway
- You have explored two kinds of edge workloads:
    - Data Management workloads defining the filtering, transformation and abstractions of edge devices
    - Edge Services running Docker containers on the IoT Edge Gateway using the Edge Core IoT service for local and cloud messaging
- You have finally explored an IoT application that uses these edge capabilities in a Worker Security scenario

We hope that you have found this lab informative, educational, and fun.

9.1 Further reading

To learn more about IoT Edge Preview 2 beta

- Developing for Watson IoT Platform Edge (Preview)
  https://console.bluemix.net/docs/services/IoT/edge/WIoTP_edge_dev.html#edge_dev
- Edge Developer Quickstart Guide
  https://github.com/open-horizon/examples/wiki/Edge-Developer-Quickstart-Guide

To learn more about the Watson IoT Platform

- Watson Internet of Things
  https://www.ibm.com/internet-of-things
- Watson IoT Platform on ibm.com
  https://www.ibm.com/internet-of-things/spotlight/watson-iot-platform
- Watson IoT Platform blog
  https://developer.ibm.com/iotplatform/blog
- Watson IoT Platform on IBM Developer Works
  https://www.ibm.com/developerworks/learn/iot/

To further explore, sign up for a trial account on IBM Cloud, create the IBM Watson IoT Platform service, and start connecting your IoT devices

9.2 We Value Your Feedback!

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

10 Troubleshooting

This hands-on lab is using the IoT Edge Preview 2 beta version. We expect there to be missing capabilities and defects in the software components running in the Watson IoT Platform, the services in the IoT Edge Service catalog, and the IoT Platform Agent and IoT Edge services running on the Edge Gateway.

This section provides a few steps on how to resolve problems

10.1 Removing Edge IoT

To remove the IoT Edge Agent and any edge services from the edge gateway

1. Run the command

   ```
   sudo hzn unregister
   ```

2. Reboot the Raspberry Pi

   ```
   sudo reboot
   ```

10.2 Reinstalling Edge IoT

To reinstall IoT Edge Agent and the services on the edge gateway

1. Unregister

   ```
   sudo hzn unregister
   ```

2. Run the command after updatating the parameters to your names

   ```
   sudo wiotp_agent_setup --org ul4s84 --deviceType EdgeGatewayType --
   deviceId EdgeGateway --deviceToken 12345678 --cloudDisableCertCheck
   true
   ```

3. Wait for the agreements to be settled

   ```
   sudo hzn agreement list
   ```

4. Wait for the docker containers to be downloaded

   ```
   sudo docker images
   ```

5. Wait for the docker containers to run

   ```
   sudo docker ps
   ```

10.3 Manually starting the Edge Lab Services

To start Edge Lab services on the edge gateway

1. Edit the command

   ```
   sudo docker run --privileged -v /var/wiotp-
   edge/persist/dc/ca:/var/wiotp-edge/persist/dc/ca --env
   WIOTP_ORGID=ul4s84 --env WIOTP_DEVICETYPE=EdgeSensorType --env
   WIOTP_DEVICEID=EdgeSensor --network $(sudo docker network ls | grep
   "edge-core-iot" | awk '{print $2}') -it edgelab-sensor
   ```

2. Update `WIOTP_DEVICETYPE` and WIOTP`_DEVICEID` with the values that apply to your system
3. Run the command
4. Update the command and run it with the parameter

   ```
   -it edgelab-analytics
   ```

5. Update the command and run it with the parameter

   ```
   -it edgelab-safety
   ```

10.4 Manually building the Edge Lab Services locally

To build the Edge Lab services on the edge gateway

1. Go to the source directory

   ```
   cd /home/pi/edgelab/docker/edgelab-sensor
   ```

2. View the README.md file

   ```
   more README.md
   ```

3. Run the command

   ```
   sudo docker build -t edgelab-sensor:0.0.9 .
   ```

4. Repeat the steps for the edgelab-analytics and edgelab-safety images

   /home/pi/edgelab/docker/edgelab-analytics

   /home/pi/edgelab/docker/edgelab-safety

think
2018