Purpose: The following lab activity will introduce you to the IoT "Ecosystem"; the components that make up a working, end-to-end IoT system. The ecosystem includes remotes (sensors and actuators), dashboards, networks, gateways, analytics, data storage, and security. You will be using the ThingsBoard platform to familiarize yourselves with many of these components, as well as focusing on the basics of a functioning platform and dashboards, sensors, MCUs (microcontrollers) and the MQTT protocol. MQTT is a lightweight publish/subscribe messaging protocol designed for M2M (machine to machine) telemetry in low bandwidth environments.

Note: You are not expected to have any in-depth knowledge or programming skills to complete this lab. The idea is to follow the tutorials to get a better understanding of how an IoT system and its components function. The lab will be broken into multiple tutorials, each building on the knowledge gained from the previous. For this Lab, we'll be using a simple Wi-Fi connection. The SSIDs and Passwords for WPA2-PSK are written on the whiteboard in the front of the room. Although this lab is mostly tutorials, you will use these activities in the second lab to create a system that you are simply given the requirements to complete.

1. **Account Creation**

   Set up an account on the ThingsBoard LiveDemo server at https://demo.thingsboard.io/signup. ThingsBoard can run on several different platforms or in the ThingsBoard cluster in the cloud. The LiveDemo system is cloud based and doesn't require you to set the platform up on a computer, so it is probably the easiest way to setup and learn the platform.

2. **Simulate Sensor Data Flow to Platform**

   The next step will be a demonstration of sending simulated sensor data from a notebook computer into the ThingsBoard cloud system using cURL from your computer. This should help you become familiar with connecting devices, dashboards, widgets, alarms, and customers. The tutorial can be found at https://thingsboard.io/docs/getting-started-guides/helloworld/.

3. **Sensor and MCU Integration and Data Ingestion via MQTT**

   Our next phase will be our first attempt at breadboarding an MCU and a basic temperature and humidity sensor together and sending data via MQTT to your platform. We will be using a NodeMCU ESP8266 DEVKIT along with a DHT22 sensor. NodeMCU is a low-cost open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. The ESP8266 is one of the more common MCUs used for IoT sensor development and deployment. For those of you new to breadboards and circuits, there are lots of good tutorials online, including this one: https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all

   We'll initially use the Arduino IDE for editing the code and uploading to the NodeMCU, though many programming environments including Python or LUA could also be used. The breadboarding instructions will differ slightly from the tutorial since we are using a NodeMCU instead of a bare-bones ESP8266; you will not need to configure the Programming/Flashing schema or Battery Power schema given in the tutorial. The NodeMCU will provide the

programming interface and power via USB connection. You only need to connect the 3 pins of the sensor (with resistor) to the appropriate pins on the NodeMCU. So, just pull up one of the many NodeMCU pinouts found on the Internet for pin references. In other words, where the instructions say GPIO 2 on the 8266, you would use D4 on the NodeMCU, sensor Vcc would connect to a 3.3 V source on the NodeMCU, and sensor ground to a ground on the NodeMCU. The tutorial can be found at **https://thingsboard.io/docs/samples/esp8266/temperature/**

4. Simulated Actuator Control with Remote Procedure Calls over MQTT

Next, let's use the 2-way capabilities of MQTT and RPCs (Remote Procedure Calls) to simulate triggering actuators on another popular MCU used in the IoT development world, the ESP32. The ESP32 is similar to the NodeMCU you used in the last phase, only with a more powerful, dual-core processor. The tutorial can be found at https://thingsboard.io/docs/samples/esp32/gpio-control-pico-kit-dht22-sensor/

Throughout the lab, I recommend you also look at additional ThingsBoard documentation and get familiar with the terms and concepts.

Deliverables:

1. Report: Put together a simple report with what you did to complete each phase and include any "code" that you wrote/modified to accomplish, as well as any modifications you had to do to circuit diagrams that deviated from the tutorial. This would best compare to the 'Methodology' section of a full CIT Lab report. Know this for your checkoffs!
2. Checkoffs: Demonstrate to the instructor that each phase works as intended and be able to answer questions based on your report.