

การใช้งาน ThingsBoard IoTs Platform เพื่อสร้างและจัดการระบบอัจฉริยะ ThingsBoard IoTs Platform for smart system

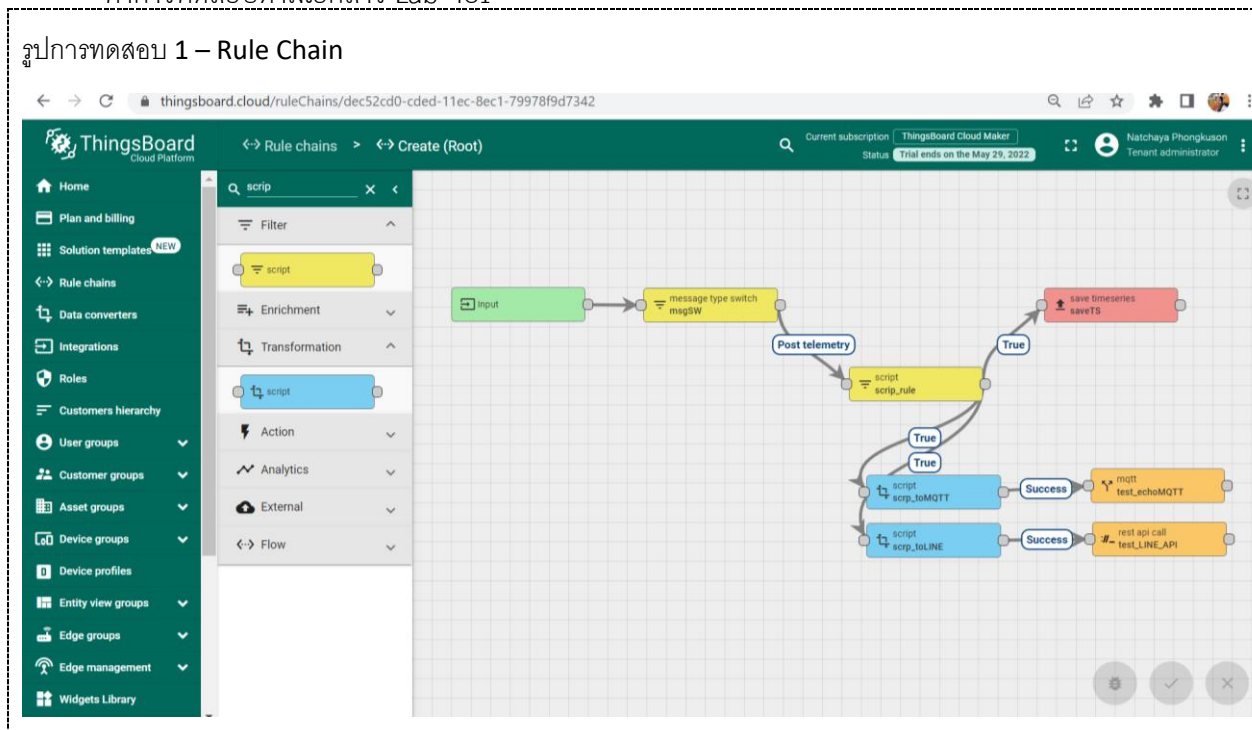
ชื่อ-สกุล : นางสาวณัฐชยา ผ่องกุล B6226718

5/5 -- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_401 – ทดสอบการใช้งาน Rule Chain เพื่อแจ้งเตือนไปยัง LINE (ตาม Lab-401)

- ทำการทดสอบตามเอกสาร Lab-401

รูปการทดสอบ 1 – Rule Chain



รูปการทดสอบ 2 – Dashboard

The screenshot shows the ThingsBoard Cloud Platform interface. The left sidebar contains navigation links: Home, Plan and billing, Solution templates (NEW), Rule chains, Data converters, Integrations, Roles, Customers hierarchy, User groups, Customer groups, Asset groups, Device groups (expanded), Device profiles, and Entity view groups. The main content area is divided into two panels. The left panel, titled 'TP_Thing: Devices', shows a table with columns 'Created time' and 'Name'. It contains one entry: '2022-05-07 15:07:51' and 'TP_Rule'. The right panel, titled 'TP_Rule Device details', shows tabs for Details, Attributes, Latest telemetry, Alarms, Events, Relations, and Audit Logs. The Details tab is active, showing fields for Name (TP_Rule), Device profile (default), Label, Assigned firmware, Assigned software, Is gateway (checkbox), and Description. There are buttons for 'Open details page', 'Manage credentials', 'Delete device', 'Copy device id', and 'Copy access token'.

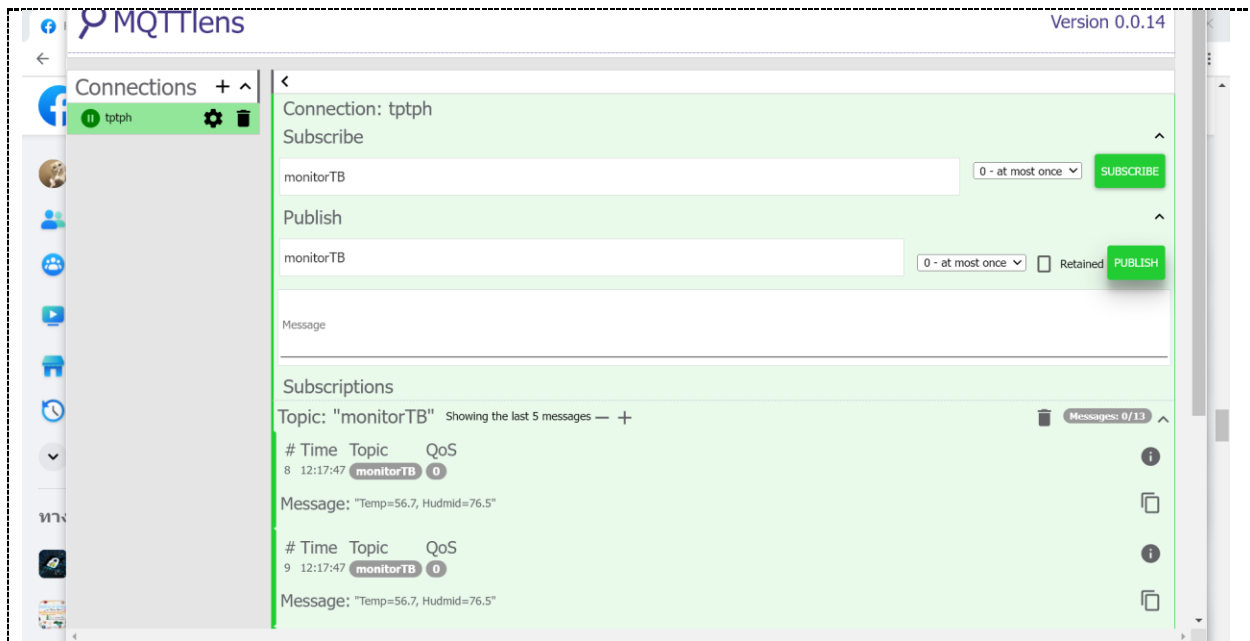
รูปการทดสอบ 3

The screenshot shows the reqbin.com/curl website. The top navigation bar includes links for Curl, Python, JavaScript, PHP, Java, JSON, Contact, Login, and Sign Up. The main content area is divided into two sections. The left section, titled 'EXAMPLES', lists various curl command examples. The right section, titled 'Execute curl commands directly from your browser', shows a curl command being executed. The command is: `curl -v -X POST -d '{"temperature":99}' http://thingsboard.cloud/api/v1/naJA8uBRcIAgD6x9TocV/telemetry --header "Content-Type:application/json"`. The execution results show a status of 200 (), time of 63 ms, and size of 0.00 kb. The output is displayed in a table with columns for Content, Headers (3), Raw (3), and Timings.

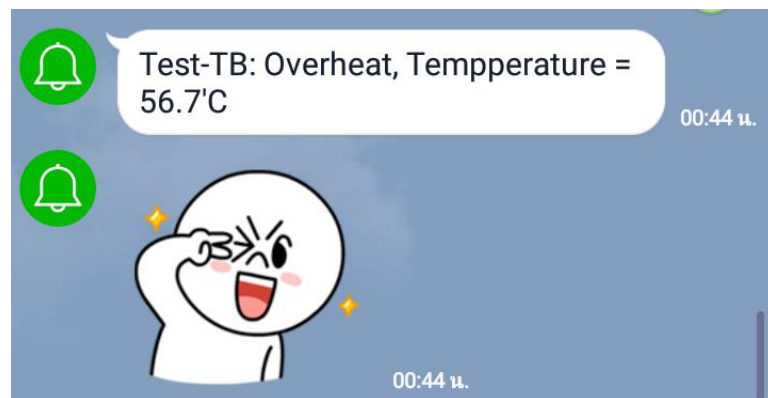
The screenshot displays two overlapping web browser windows. The top window is the ThingsBoard IoT Platform interface, specifically the 'Test script function' page. The URL is `thingsboard.cloud/ruleChains/dec52cd0-cded-11ec-8ec1-79978f9d7342`. The interface is divided into several sections: 'Message type' (Post telemetry), 'Message' (a JSON object with temperature and humidity), 'Metadata' (deviceName, deviceType, ts), 'Filter' (a JavaScript function), and 'Output' (a boolean value). The 'Filter' section contains the following code:

```
function Filter(msg, metadata, msgType) {  
  1 return typeof msg.temperature !== 'undefined'  
  2 || (msg.temperature >= 40 && msg.temperature <= 80);  
}
```

The 'Output' section shows the result of the filter function: `1 false`. The bottom window is the LINE Notify API documentation page, titled 'ออก Access Token (สำหรับผู้พัฒนา)'. It features a modal dialog box for generating a token. The modal has the title 'Token ที่ออก' and displays the generated token: `uepZtYC9OhlwSR5oIT4lpby5TIsjRALDnOkPhJj!`. Below the token, there is a warning message in Thai: 'ถ้าออกจากหน้านี้ ระบบจะไม่แสดง Token ที่ออกใหม่อีกต่อไป โปรดคัดลอก Token ก่อนออกจากหน้านี้'. At the bottom of the modal are two buttons: 'คัดลอก' (Copy) and 'ปิด' (Close).



รูปการทดสอบ 4

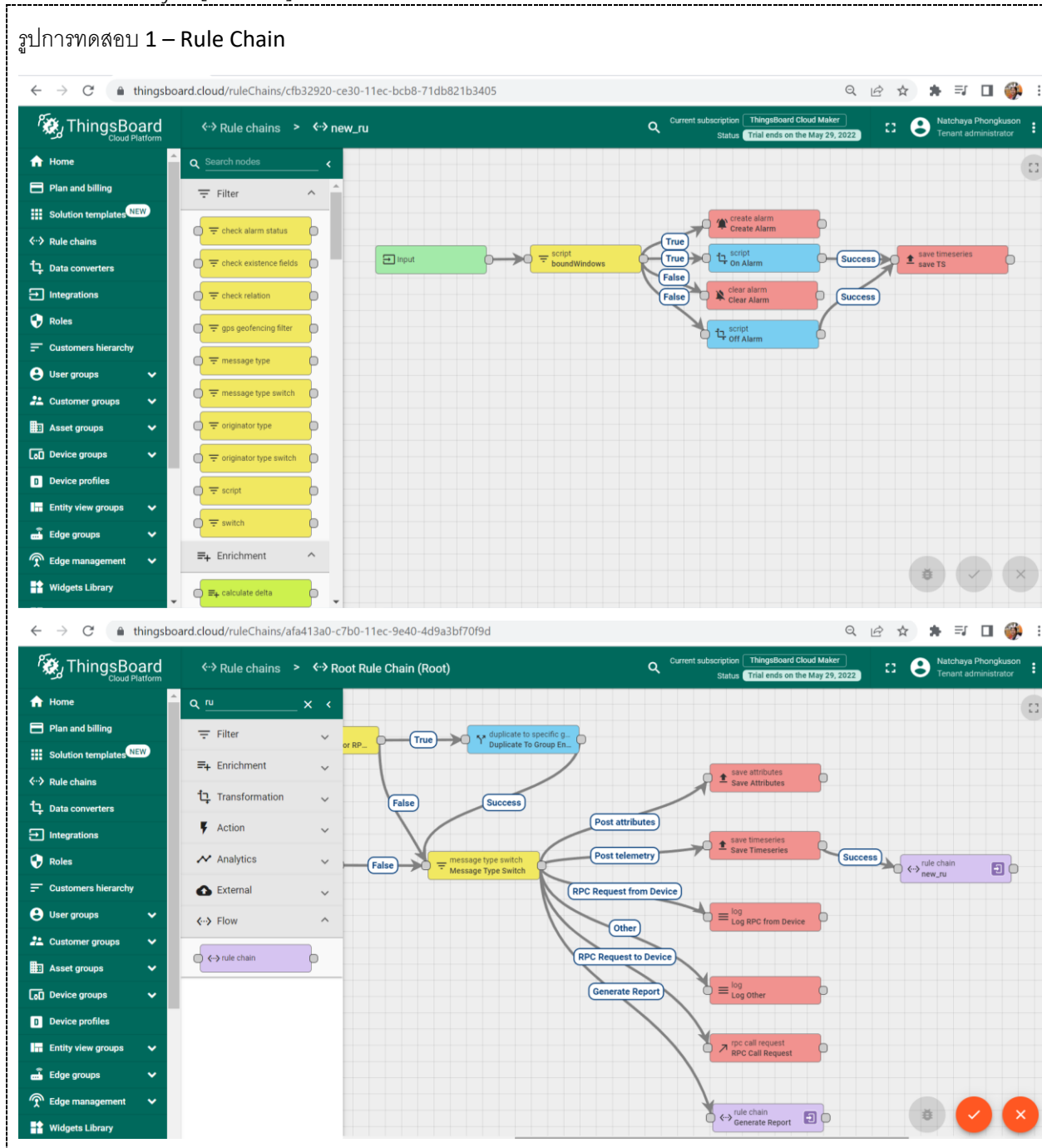


ได้ทดลองทำตามเอกสารแลป มีการปรับเปลี่ยนเล็กน้อยตรง Token และค่า API ต่างๆให้เป็นของตนเอง เมื่อทดสอบการทำงานพบว่าสามารถทำงานได้ตามค่าที่กำหนดไว้

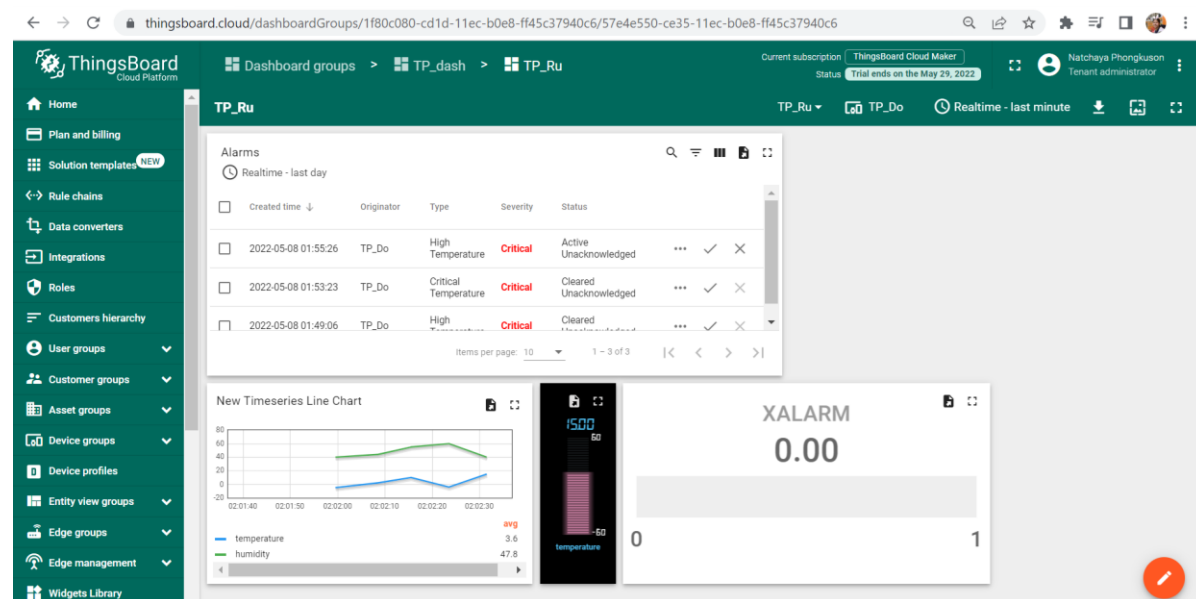
Quiz_402 – ทดสอบการทำงานของ Alarm เมื่ออุณหภูมิอยู่นอกเขตที่กำหนด (ตาม Lab-402)

- ทำการทดสอบตามเอกสาร Lab-402 กำหนดเงื่อนไขในช่วงที่ยอมรับ คือ temperature = $[-5,15]$ และ humidity = $[40 - 60]\%$

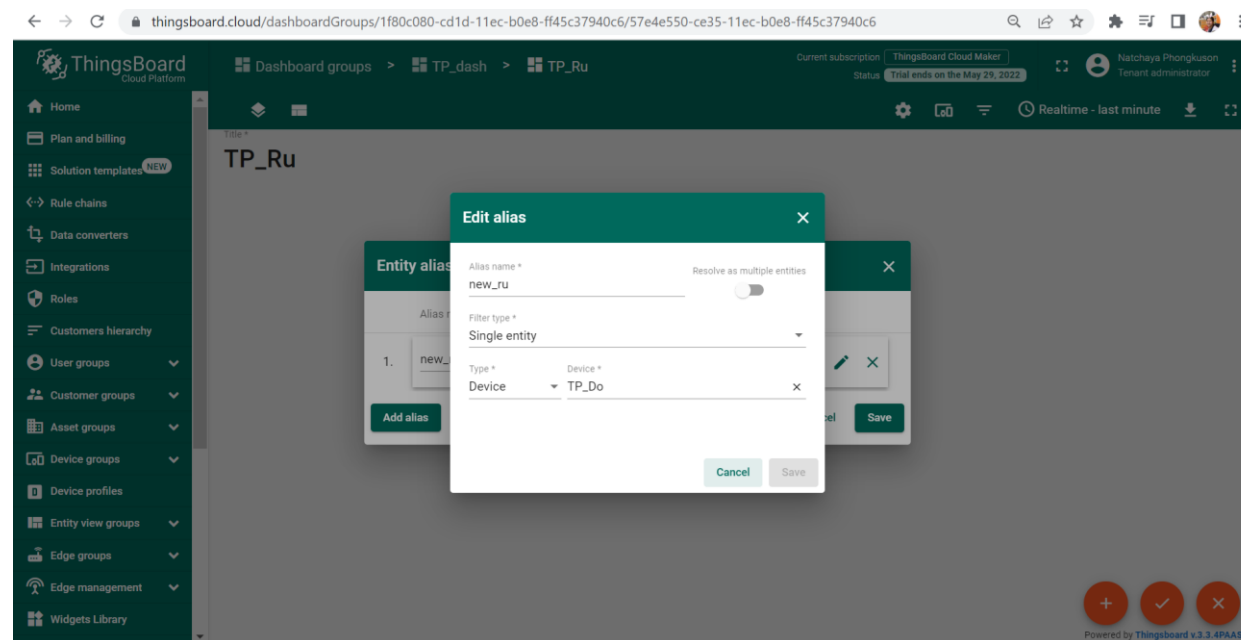
รูปการทดสอบ 1 – Rule Chain



รูปการทดสอบ 2 – Dashboard



รูปการทดสอบ 3



The image displays two screenshots of the ThingsBoard Cloud Platform interface, specifically the Rule Chains configuration page. The top screenshot shows the 'boundWindows' rule chain, and the bottom screenshot shows the 'Create Alarm' rule chain.

Top Screenshot: boundWindows Rule Chain

- Page Header:** thingsboard.cloud/ruleChains/cfb32920-ce30-11ec-bcb8-71db821b3405
- Left Sidebar:** Home, Plan and billing, Solution templates (NEW), Rule chains, Data converters, Integrations, Roles, Customers hierarchy, User groups, Customer groups, Asset groups, Device groups, Device profiles, Entity view groups, Edge groups, Edge management, Widgets Library.
- Search Nodes:** Filter
- Rule Chain Nodes:** check alarm status, check existence fields, check relation, geo geofencing filter, message type, message type switch, originator type, originator type switch, script, switch, Enrichment, calculate data.
- Right Panel:**
 - Name:** boundWindows
 - Filter:** function Filter(msg, metadata, msgType) {
1 return msg.temperature < -40 || msg.temperature > 80;
2 }
 - Test filter function** button
 - Description:**

Bottom Screenshot: Create Alarm Rule Chain

- Page Header:** thingsboard.cloud/ruleChains/cfb32920-ce30-11ec-bcb8-71db821b3405
- Left Sidebar:** Same as the top screenshot.
- Search Nodes:** Filter
- Rule Chain Nodes:** Same as the top screenshot.
- Right Panel:**
 - Name:** Create Alarm
 - Use message alarm data:** ☐
 - Alarm details builder:** function Details(msg, metadata, msgType) {
1 var details = {};
2 if (metadata.prevAlarmDetails) {
3 details = JSON.parse(metadata.prevAlarmDetails);
4 //remove prevAlarmDetails from metadata
5 }
6 }
7
8 return details;
9 }
 - Test details function** button
 - Alarm type:** Critical Temperature
 - Hint:** use \${metadataKey} for value from metadata, \${messageKey} for value from message body

The screenshot shows the ThingsBoard Cloud Platform interface. On the left is a sidebar with navigation options: Home, Plan and billing, Solution templates, Rule chains, Data converters, Integrations, Roles, Customers hierarchy, User groups, Customer groups, Asset groups, Device groups, Device profiles, Entity view groups, Edge groups, Edge management, and Widgets Library. The main area displays a rule chain named 'new_ru' with a filter and an 'Input' node. A modal window titled 'Clear Alarm' is open, showing the 'Details' tab. It contains a JavaScript function for building alarm details:

```
function Details(msg, metadata, msgType) {
  1 var details = {};
  2 if (metadata.prevAlarmDetails) {
  3   details = JSON.parse(metadata.prevAlarmDetails);
  4   //remove prevAlarmDetails from metadata
  5   //now metadata is the same as it comes IN this rule node
  6 }
  7
  8 return details;
  9
}
```

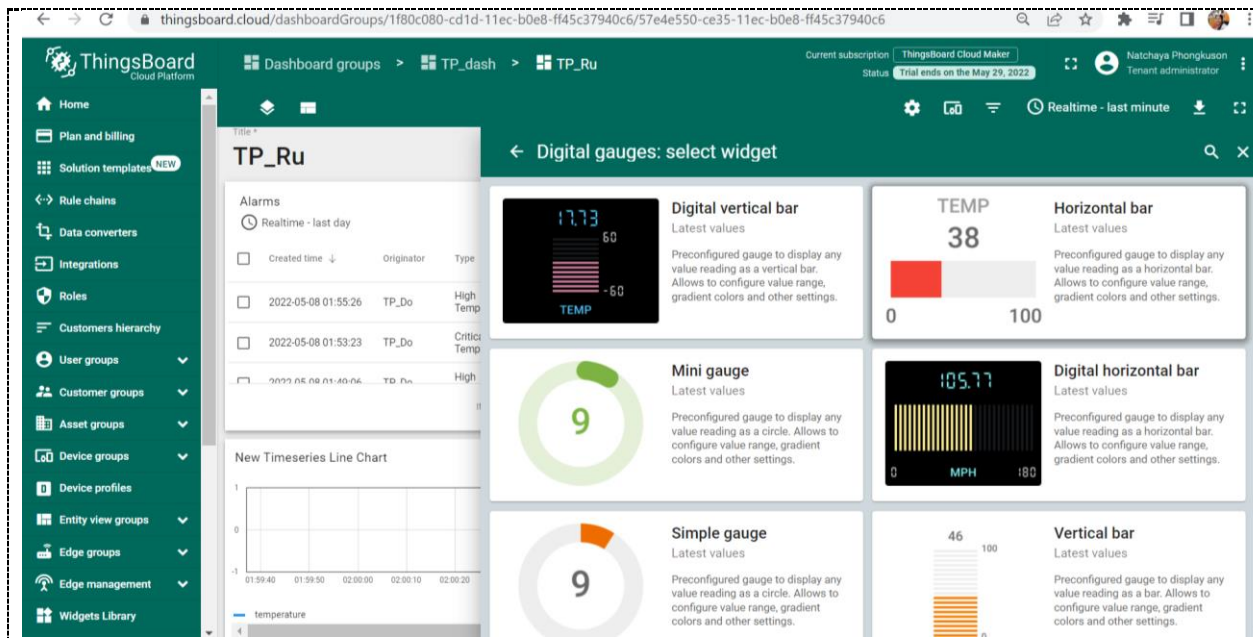
Below the function is a 'Test details function' button. The 'Alarm type' is set to 'Critical Temperature'. A hint indicates: 'Hint: use \${metadataKey} for value from metadata, \${messageKey} for value from message body'. The 'Description' field is empty.

รูปการทดสอบ 4

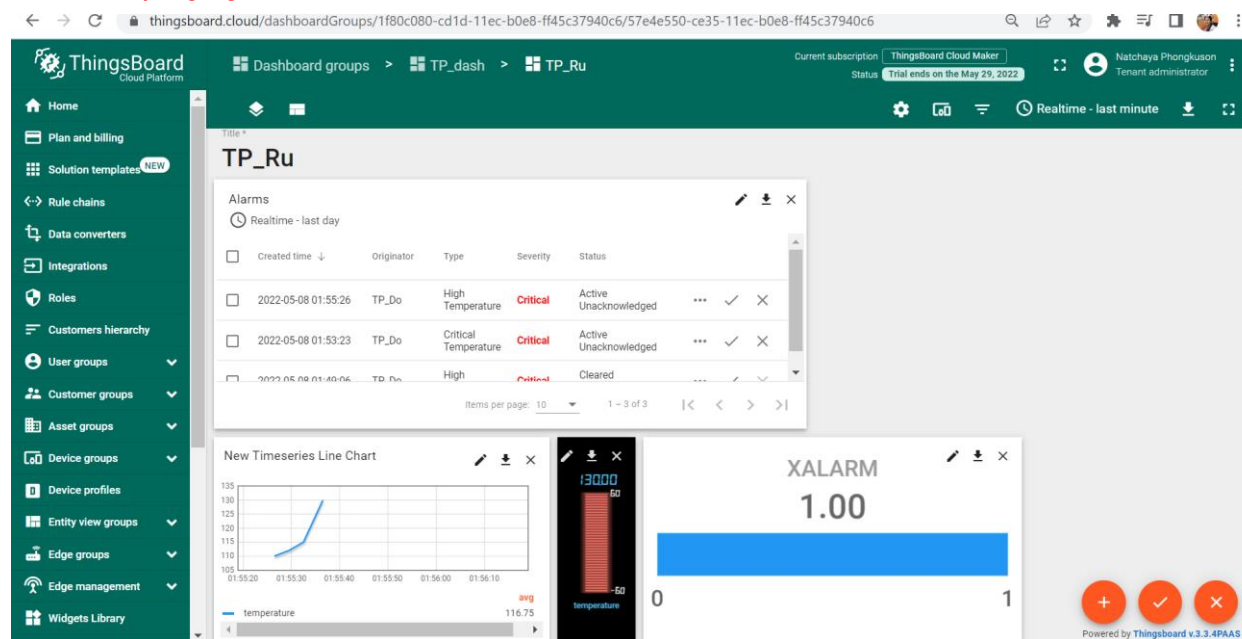
The screenshot shows the Reqbin website interface. The top navigation bar includes links for Curl, Python, JavaScript, PHP, Java, JSON, Contact, Login, and Sign Up. The main area displays a curl command that was executed successfully:

```
curl -v -X POST -d '{"temperature":15,"humidity": 40}'
http://thingsboard.cloud/api/v1/JY0hAKKvRNw4Qk9Q6TM/tele
metry --header "Content-Type:application/json"
```

The response status is 200 (), time is 27 ms, and size is 0.00 kb. The 'Content' tab is selected, showing the response body. The left sidebar lists various curl examples, including POST JSON, GET Request, POST Body, Ignore SSL Errors, Add Header, Download File, HTTPS Request, Basic Auth, POST File, Proxy, Follow Redirect, Bearer Token Auth, GET JSON, POST Form, Content-Type, Set Timeout, XML, SSL Request, HEAD Request, PUT Request, and DELETE Request.



แดมกรเฝ้าอุณหภูมิสูง



ได้ทำอุณหภูมิในช่วงที่ยอมรับ คือ temperature = [-5,15] และ humidity = [40 – 60]% โดยตั้งค่าตามเอกสารแต่จะเปลี่ยนค่าที่ curl จะพบว่าหน้า dashboard จะเปลี่ยนไปตามค่าที่ได้รับมา และจะมีการ Alarms สามารถ create และ clear ได้ดังรูปด้านบน

Quiz_403 – ให้ตอบคำถาม แสดงแนวคิด อภิปรายในหัวข้อต่อไปนี้

1. ความรู้ที่ได้เพิ่มเติมเกี่ยวกับ IoT

แต่เดิมรู้ว่า IoT คืออุปกรณ์สื่อสารกันผ่าน Internet เมื่อได้ทำตามเอกสารพบว่าเข้าใจ IoT มากขึ้น จากการทดลองพบว่า Internet สามารถช่วยในการสื่อสาร เช่น Thingboard ส่งข้อมูลไปยัง Line และ MQTTlens ได้ โดยการทำงานของมันสามารถกำหนดเงื่อนไขได้ตามผู้ใช้ต้องการ

2. ความรู้ที่ได้เพิ่มเติมเกี่ยวกับ ThingsBoard

ThingsBoard สามารถสื่อสารกับ MQTTlens และ Line ได้ เมื่อได้ลองใช้จึงทราบว่ามันเป็น platform ที่ใช้ฟรีหรือเรียกว่า Opensource ใช้งานได้ค่อนข้างง่าย (แต่เริ่มใช้แรกๆจะยาก)

3. แนวทางการปรับใช้ ThingsBoard IoT Platform กับงานที่รับผิดชอบ

งานที่รับผิดชอบคือการส่งข้อมูลไป MQTT และ Line สามารถนำมาปรับใช้โดยการเอา ThingsBoard มาสื่อสารข้อมูลระหว่างพวกมัน ได้ตั้งค่า device, Dashboard รวมไปถึง rule chain สามารถตั้งคำสั่งเงื่อนไขได้หลากหลาย

4. คำแนะนำ ข้อเสนอแนะ จากผู้เรียน – ประเด็นเนื้อหาที่น่าสนใจ (มากไป, น้อยไป, ลึกไป, อธิบายน้อยไป, เอกสาร, ความเหมาะสมของเวลา)

เอกสารครั้งนี้ไม่ค่อยละเอียด(เมื่อเทียบกับเอกสารหลักสูตรก่อนหน้า) แต่ฟังคลิปลงไปด้วยก็เลยทำได้ไม่ติดปัญหา ปัญหาหลักๆที่เจอคือพิมพ์ผิดจึงทำงานไม่สำเร็จ ก็ใช้เวลาหาอยู่พอสมควร ทั้งนี้ค่อนข้างทำคล่องเพราะเคยผ่านมา ก่อนจึงประเมินว่า เนื้อหาที่น่าสนใจค่อนข้างเยอะ อธิบายดีมาก เอกสารอยากให้ละเอียดขึ้น ใช้เวลาได้เหมาะสมค่ะ

5. คำแนะนำ ข้อเสนอแนะ จากผู้เรียน – ประเด็นเนื้อหาที่อยากให้เสริม หรือเปิดหลักสูตรเพิ่มเติม หรือต้องการให้อบรมแบบเข้าห้องเรียน

ไม่มีเพิ่มเติมค่ะ ตอนนี้อยากฝึกให้คล่องก่อน พอทำหลายๆรอบก็รู้สึกกว้างง่ายขึ้นเยอะ เอาไปใช้ต่อยอดได้ขึ้นอยู่ กับความสนใจและทักษะการประยุกต์ใช้